

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 THINK IN GRAPHS: INFRASTRUCTURE AND BENCH- MARK FOR LARGE LANGUAGE MODEL REASONING FRAMEWORKS

Anonymous authors

Paper under double-blind review

ABSTRACT

Enhancing the reasoning ability of Large Language Models (LLMs) has become a central focus of current research. While approaches based on prompt engineering have significantly improved LLM performance, the increasing complexity of reasoning frameworks has led to higher development costs. Moreover, these frameworks often require extensive redesigns to actually work on different tasks, with their performance heavily dependent on these specific designs. This creates challenges in establishing clear and consistent evaluation benchmarks. To address these issues, we propose a unified infrastructure that represents reasoning processes as graphs, thereby standardizing and structuring the reasoning workflow. This approach enables more consistent and efficient implementation of diverse reasoning frameworks, facilitates objective comparisons, and supports deeper analysis through graph algorithms. Building on this infrastructure, we develop an LLM reasoning benchmark and demonstrate its effectiveness through multiple experiments, enabling more comprehensive evaluation and analysis. *Code and data can be found in <https://anonymous.4open.science/r/210-5DD5/>.*

1 INTRODUCTION

The enhancement of reasoning ability has become a major focus in current research on LLMs (Huang & Chang, 2023). With the emergence of the Chain-of-Thought (CoT) framework (Wei et al., 2022b), prompt-based reasoning optimization methods have gained widespread applications (Hao et al., 2024). Through CoT, LLMs can reason more transparently and better handle complex tasks. Further advancements, such as Tree of Thoughts (ToT) (Yao et al., 2023b) and Graph of Thoughts (GoT) (Besta et al., 2024b), represent the reasoning process in more complex tree-like or graph-like structures, as well as others, facilitating more sophisticated forms of reasoning (Yao et al., 2024; Shin & Kim, 2025; Sel et al., 2024; Zhou et al., 2023b).

However, as more complex and powerful reasoning frameworks are proposed, implementing them incurs increasingly higher costs, including those related to coding and prompt design (McDonald et al., 2024). Furthermore, these frameworks often require frequent redesigns of prompts and program structures to perform effectively across different tasks (Gao et al., 2025). This not only results in development inefficiencies but also creates a strong dependence between the performance of these methods and their specific designs, making it challenging to establish clear and consistent benchmarks.

In fact, logical reasoning is inherently a highly complex and difficult-to-quantify process (Shojaee et al., 2025). Reasoning involves not only the organization and deduction of information but also the integration of information across multiple dimensions and layers. Different reasoning paths may lead to the same conclusion, and whether the reasoning steps within these paths are considered “reasonable” or “correct” often lacks a unified standard. In the philosophy and cognitive science of human thinking, logical reasoning is often viewed as a field filled with ambiguity and uncertainty (Stenning & Van Lambalgen, 2012). Therefore, despite the significant advancements made by LLMs in prompt-based reasoning optimization, **how to avoid the continuous redevelopment of reasoning frameworks and how to objectively and fairly evaluate these reasoning processes remain unresolved challenges.**

To address the above issue, it is necessary to develop a unified framework that can standardize and structure the reasoning process of LLMs. In fact, if the reasoning steps are represented as nodes and the relationships between these steps are represented as edges, all reasoning processes can be expressed as graphs. This is because all reasoning architectures—whether chains, trees, or other forms—are essentially specialized instances of a graph. Moreover, by representing the reasoning process as structured data in the form of a graph, we can make a more objective comparison of these reasoning processes, e.g., using quantifiable graph distances to measure the differences between reasoning procedures. Additionally, if all reasoning processes can be represented as graphs, a unified graph-based infrastructure would also, in turn, enable the implementation of any reasoning framework. Figure 1 provides an intuitive illustration of such viewpoints.

Building on this perspective, we have constructed an infrastructure for LLM reasoning called *Think in Graphs* (TiG). TiG implements different LLM reasoning frameworks in a unified manner. Specifically, all reasoning processes in TiG are specified by a configuration file. Based on such a file, TiG continuously generates new thoughts, which are added to the graph-based reasoning flow, enabling the ongoing progression of reasoning. For users of TiG, the only requirement is to define the configuration file, which eliminates the need to rebuild the entire reasoning framework. Additionally, TiG can save the LLM reasoning process in graph form and use this data structure for unified and objective comparisons. Furthermore, graph-based algorithms, such as our proposed graph kernel (introduced later), can be applied to the analysis of logical reasoning, thereby enabling more diverse and in-depth evaluations. Building on TiG and the collected reasoning tasks of various types, we constructed a prompting-based reasoning benchmark for LLMs. We then conducted extensive experiments with different reasoning frameworks on this benchmark to gain deeper insights and to demonstrate the practicality of the proposed TiG.

Our contributions are as follows:

- We design TiG, a unified and efficient infrastructure that facilitates the rapid implementation of diverse LLM reasoning frameworks, serving as a foundation to support ongoing research on prompt engineering for reasoning.
- With TiG, we design a series of new metrics for analyzing LLM reasoning logic, including a novel graph kernel.
- Based on TiG and a dataset with a variety of test tasks, we construct a benchmark for prompt engineering for reasoning.
- We conduct a series of analytical experiments based on the proposed benchmark and derive the corresponding conclusions.

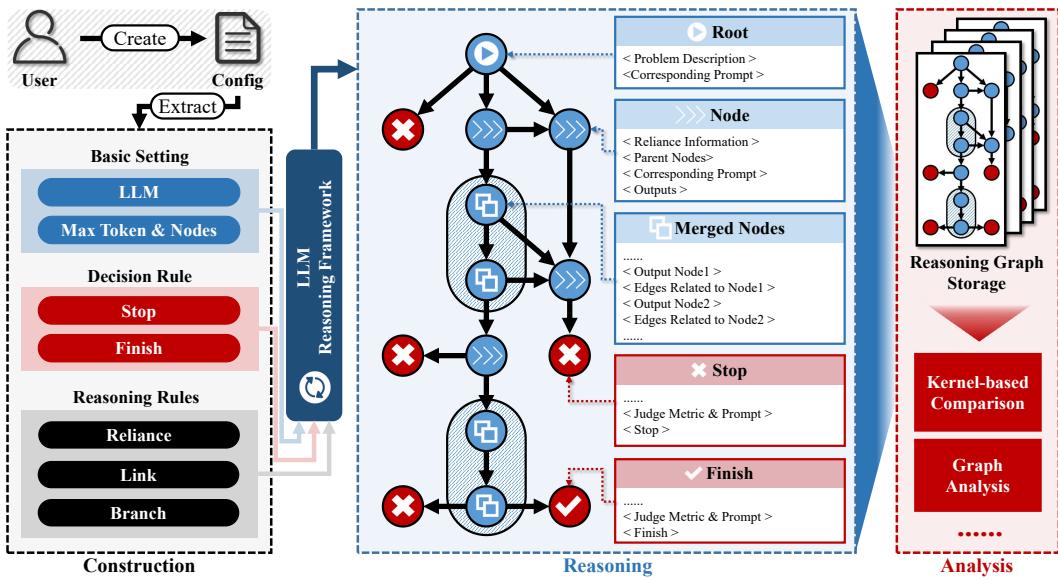
2 RELATED WORKS

2.1 REASONING WITH PROMPTING

Recent research has increasingly focused on designing logically consistent prompts that improve reasoning performance, enabling LLMs to tackle complex tasks more effectively. The CoT framework (Wei et al., 2022a) has inspired advances like Auto-CoT (Zhang et al., 2023), which automates and optimizes reasoning outputs, and LogiCoT (Zhao et al., 2024), which integrates symbolic logic for refinement. Prompt Sketching (Beurer-Kellner et al., 2024) and CCoT (Mitra et al., 2024) improve control over reasoning, while RASC (Wan et al., 2025) boosts consistency and reduces sampling costs. More complex structures like ToT (Yao et al., 2023b), GoT (Besta et al., 2024a), and GoTR (Yao et al., 2024) enhance multi-step and multimodal reasoning. EGoT (Shin & Kim, 2025) optimizes inference paths, and ThoT (Zhou et al., 2023b) and AoT (Sel et al., 2024) structure reasoning hierarchically and algorithmically, enabling more efficient exploration of complex reasoning paths.

108 2.2 BENCHMARKING LLM REASONING
109

110 Recent research on the reasoning abilities of LLMs has led to the development of several benchmarks. MMLU (Hendrycks et al., 2021), BIG-bench (Srivastava et al., 2022), and HELM (Liang
111 et al., 2022) provide comprehensive multi-task evaluations. MT-Bench (Zheng et al., 2023) focuses
112 on multi-turn dialogue reasoning, while OpenAI Eval (OpenAI, 2023b) facilitates benchmark shar-
113 ing. Specific datasets for reasoning include BBH (Suzgun et al., 2022), GSM8K (Cobbe et al.,
114 2021), MATH (Hendrycks et al., 2021), ARC (Clark et al., 2018), and DROP (Dua et al., 2019).
115 ReClor (Yu et al., 2020) evaluates logical reasoning, and MME-CoT (Jiang et al., 2025) focuses on
116 CoT reasoning in LLMs. REVEAL (Greyling, 2024) verifies CoT correctness. We approach the
117 problem from a different perspective, constructing a new, more unified thinking infrastructure based
118 on graphs, then benchmarking LLM reasoning with it.
119

120 3 INFRASTRUCTURE
121142 Figure 2: Architecture of the proposed TiG infrastructure.
143

144 The overall architecture of TiG is illustrated in Figure 2. In TiG, the entire reasoning process is
145 represented as a directed acyclic graph (DAG), denoted as $G^{(t)} = \{\mathcal{V}^{(t)}, \mathcal{E}^{(t)}\}$, where t indicates the
146 number of reasoning iterations. $\mathcal{V}^{(t)}$ represents the set of nodes in $G^{(t)}$, with each node correspond-
147 ing to a reasoning step generated by the LLM. Each node effectively encodes a segment of thought.
148 $\mathcal{E}^{(t)}$ denotes the set of edges, where each edge represents a dependency between nodes. The use of
149 the DAG is motivated by its clear causal structure and computational simplicity (Spirtes et al., 2000).
150 Operations such as backtracking (Besta et al., 2024a), which might otherwise introduce cycles, are
151 also represented within acyclic structures. Specifically, a newly generated node resulting from a
152 backtracking operation can be formalized as a common child of the node requiring backtracking and
153 the node it backtracks to. In other words, as the reasoning process unfolds, $G^{(t)}$ is guaranteed to
154 always remain a DAG.

155 Clearly, the evolution of $G^{(t)}$ with increasing t reflects the ongoing reasoning process of the LLM.
156 TiG constrains and guides this evolutionary process, while also enabling the analysis of $G^{(t)}$. Specif-
157 ically, the workflow of TiG consists of three phases: construction, reasoning, and analysis. (1) In the
158 **construction phase**, the user provides a configuration file to define the intended reasoning process,
159 including three sets of constraint rules that specify the reasoning behavior within the framework.
160 (2) In the **reasoning phase**, the LLM leverages our framework to extract the defined rules from the
161 user’s configuration and execute the reasoning accordingly. (3) In the **analysis phase**, we collect the
final $G^{(t)}$ graph and perform result analysis. These phases will be introduced separately below.

162 3.1 CONSTRUCTION PHASE
163

164 In this phase, the user constructs the configuration file, specifying the basic settings, the decision
165 rule, and the reasoning rules. The approaches for building each of these three components will be
166 introduced in the following subsections. The implementation details of the configuration file can be
167 found in [Appendix C](#), and [Section 3.4](#) provides a running example.

168

169 **Basic settings.** The basic settings include the LLM to be used, as well as the specified limits on
170 the number of tokens and nodes consumed for task execution.

171

172 **Decision rule.** This rule specifies how to determine the next action for a given node $v^{(t)} \in G^{(t)}$.
173 To be concise, we omit the superscript (t) for $v^{(t)}$, because the node v itself does not change over
174 time. The decision rule constrains whether the reasoning process should (1) terminate at v , (2) treat
175 v as the final answer, or (3) continue reasoning based on v . The decision rule is expressed as a
176 textual description.

177

178 **Reasoning rules.** These rules apply to nodes that require further reasoning, and they specify the
179 structure and attributes of the subgraph $G_{(v)}^{\text{sub}}$ to be generated. In essence, $G_{(v)}^{\text{sub}}$ represents the newly
180 generated thoughts together with their relationships to the preceding ones. Specifically, each node
181 v corresponds to one reasoning rule, which is selected based on the specific characteristics of v ,
182 including its position in the graph, its distance from the root node, its attributes, and other contextual
183 features. For the defined set of rules $\Phi = \{\phi_i\}_{i=1}^m$, our infrastructure builds the following mapping:

$$184 \quad i = s(v, G^{(t-1)}, \Phi), \quad i \in \{1, 2, \dots, m\}, \quad (1)$$

186 where i is the index of the specific rule, $s(\cdot)$ denotes the selection function. $s(\cdot)$ is defined by the
187 user with the configuration file, with details in [Appendix C](#).

188 The selected rule determines, based on the features of node v and the current $G^{(t)}$, all possible child
189 nodes of v , their respective parent nodes, the connections between these nodes, and the prompt used
190 for generation. Formally, the child node set $\text{Ch}(v)$ of v can be represented as:

$$192 \quad \text{Ch}(v) = \{u \in \mathcal{V}^{(t+1)} \mid (v, u) \in \mathcal{E}^{(t+1)}\}, \quad (2)$$

194 where $\mathcal{E}^{(t+1)}$ and $\mathcal{V}^{(t+1)}$ are generated by applying the rules. The set of parent nodes is:

$$195 \quad \bigcup_{u \in \text{Ch}(v)} \text{Pa}(u) = \left\{ w \in \mathcal{V}^{(t)} \mid \exists u \in \mathcal{V}^{(t+1)}, (v, u) \in \mathcal{E}^{(t+1)} \wedge (w, u) \in \mathcal{E}^{(t+1)} \right\}. \quad (3)$$

198

199 $\text{Pa}(\cdot)$ denotes the parent nodes. The generated graph substructure corresponding to v is:

$$200 \quad G_{(v)}^{\text{sub}} = \left\{ \text{Ch}(v), \left\{ (u, w) \in \mathcal{E}^{(t+1)} \mid u \in \text{Ch}(v), w \in \text{Pa}(u) \right\} \cap \left\{ (v, u) \in \mathcal{E}^{(t+1)} \mid u \in \text{Ch}(v) \right\} \right\}. \quad (4)$$

203 In the reasoning process, some methods generate only a single reasoning step—i.e., a single node
204 in $G^{(t)}$ —per generation round of LLM, while others may generate multiple reasoning steps at once.
205 Our framework is designed to support both approaches. Specifically, when multiple nodes are
206 generated in a single round, we treat the entire output as a single node during initial processing, and
207 then split it into individual nodes based on predefined delimiters embedded in the infrastructure.

208

209 3.2 REASONING PHASE
210

211 In this phase, the framework processes the nodes and continuously updates the graph until either a
212 final result is obtained or the maximum token (or node) limit for reasoning is reached. At the t -th
213 update step, only the nodes newly generated in step $(t-1)$, i.e., $u \in G^{(t-1)} \setminus G^{(t-2)}$, are selected.
214 This is because all other nodes are either already terminated or are ancestor nodes of the newly
215 generated nodes, and thus no longer represent active reasoning processes. Excluding them helps
reduce computational cost. The detailed reasoning procedure is provided in [Algorithm 1](#).

216 **Algorithm 1** Reasoning Procedure of TiG

217 **Require:** Problem description, decision rule, evolution rule set Φ

218 **Ensure:** Final answer to the problem

219 1: Initialize graph $G^{(0)}$ with a single root node based on the problem description

220 2: Set $t \leftarrow 1$

221 3: **while** Total token usage or node count has not exceeded the maximum limit **do**

222 4: Select newly generated nodes: $\mathcal{X} \leftarrow G^{(t-1)} \setminus G^{(t-2)}$

223 5: **for all** $v \in \mathcal{X}$ **do**

224 6: Apply decision rule to determine whether v should be an answer node or terminated

225 7: **if** v is an answer node **then**

226 8: Output the answer and terminate the reasoning process

227 9: **else if** reasoning based on v should be stopped **then**

228 10: Skip to the next node

229 11: **else**

230 12: Identify evolution rule index: $i \leftarrow s(v, G^{(t-1)}, \Phi)$

231 13: Retrieve ϕ_i from Φ and the structure of $G_{(v)}^{\text{sub}}$

232 14: Generate node features of $G_{(v)}^{\text{sub}}$ via LLM using v , $G^{(t-1)}$, and ϕ_i .

233 15: Integrate $G_{(v)}^{\text{sub}}$ into $G^{(t)}$

234 16: **end if**

235 17: **end for**

236 18: Update round index: $t \leftarrow t + 1$

237 19: **end while**

239 3.3 ANALYSIS PHASE

241 We represent the entire reasoning process using the graph obtained at the final time step, denoted
 242 as G . This graph serves as a compact and interpretable abstraction of the sequence of intermediate
 243 reasoning steps. Based on G , we are able to perform more precise and fine-grained analyses. In
 244 particular, we can extract the exact set of reasoning paths that contribute to the final answer—namely,
 245 the union of all directed paths from the root node r to the answer node a , denoted by $\mathcal{P}_{r \rightarrow a}$. This
 246 allows us to compute the proportion of nodes involved in generating the final answer as:

$$247 \quad 248 \quad \lambda = \frac{|\mathcal{P}_{r \rightarrow a}|}{|\mathcal{V}|}.$$

250 Furthermore, the structural properties of G enable us to identify and quantify redundant nodes,
 251 calculate the proportion of redundant tokens, and track the precise number of instances when the
 252 reasoning reaches a dead end. These metrics are thoroughly analyzed in the experimental section.

253 Given this graph-based representation of the LLM’s reasoning trajectory, we further introduce the
 254 concept of a graph kernel to formally measure the similarity between different reasoning processes.
 255 Specifically, we propose an extension to the traditional Weisfeiler-Lehman (WL) kernel (Sher-
 256 vashidze et al., 2011) that incorporates the directionality of G and the answer-contributing ratio
 257 λ . The resulting kernel, termed the *Reasoning Graph Weisfeiler-Lehman (RGWL)* kernel, is defined
 258 as follows:

$$259 \quad 260 \quad \mathbb{K}_{\text{RGWL}}^{(h)}(G, G') = \sum_{i=1}^h \left(\left\langle \eta \left(\tilde{\psi}^{(i)}(\rho(\tau(G))) \right), \eta \left(\tilde{\psi}^{(i)}(\rho(\tau(G'))) \right) \right\rangle \right. \\ 261 \quad 262 \quad \left. + \lambda \lambda' \left\langle \eta \left(\tilde{\psi}^{(i)}(\rho(G)) \right), \eta \left(\tilde{\psi}^{(i)}(\rho(G')) \right) \right\rangle \right), \quad (5)$$

263 where $\tau(\cdot)$ extracts the answer-contributing subgraph $\mathcal{P}_{r \rightarrow a}$, and $\rho(\cdot)$ performs KNN-based node
 264 labeling by clustering node features from both G and G' and assigning labels to each cluster. $\tilde{\psi}^{(i)}(\cdot)$
 265 denotes the i -th round of label propagation as in the WL kernel, but restricted to the direction of
 266 edges. λ' is the answer-contributing ratio of G' . The function $\eta(\cdot)$ computes a histogram of node
 267 labels after each round. Further details are provided in [Appendix E](#).

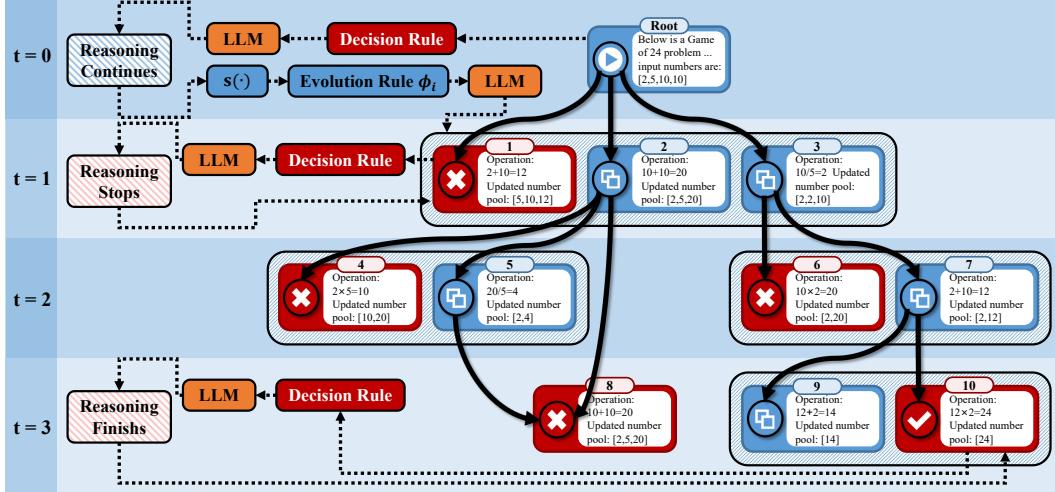
268 We further prove that the RGWL kernel is positive semi-definite, ensuring that it defines a valid
 269 inner product in a Reproducing Kernel Hilbert Space (RKHS). This theoretical guarantee supports

270 the validity of subsequent analyses and allows the RGWL kernel to be broadly employed in a variety
 271 of kernel-based learning algorithms.

272 **Proposition 1.** *The kernel matrix $\mathbf{K}_{RGWL}^{(h)}$ defined by $\mathbb{K}_{RGWL}^{(h)}(\cdot, \cdot)$ is positive semi-definite.*

273 The formal proof of this proposition is provided in [Appendix D](#).

277 3.4 RUNNING EXAMPLE



295 Figure 3: A running example on Game of 24.
 296

297 Here, we use a practical example to demonstrate the infrastructure. The example is a ToT (Yao et al.,
 298 [2023a](#)) framework specifically built for the Game of 24 problem with TiG.

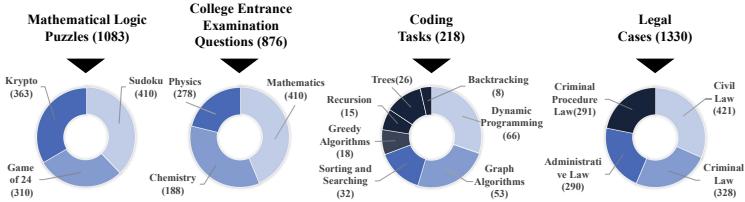
299 The first step is to construct the configuration file that specifies the decision rules and reasoning
 300 rules. The specific content of the configuration file used in this example is provided in [Appendix](#)
 301 [C](#). Within it, the decision rule states that if the answer can be calculated to 24, the result should be
 302 returned. If neither the current node nor any historical nodes can lead to 24, the reasoning based on
 303 the current node should be terminated. The reasoning rules state that: If the node is the root, then
 304 generate three child nodes for further reasoning. If the current node cannot lead to 24, but its parent
 305 node can, then backtrack and generate two child nodes based on that parent node. In all other cases,
 306 generate two child nodes for further reasoning. Additionally, the reasoning rules establish that,
 307 regardless of the number of nodes generated, the generation of each node's child nodes is carried
 308 out through a single interaction with the LLM.

309 Next, reasoning is carried out based on the configuration file. The entire reasoning process is il-
 310 lustrated in Figure 3. Initially, at time $t = 0$, reasoning begins at the root node. The root node
 311 essentially serves as a description of the problem. As shown in the figure, the content of the root
 312 node is evaluated by the decision rule and the LLM to determine whether reasoning should continue.
 313 Once reasoning is confirmed to proceed, the corresponding rule from the reasoning rules is selected.
 314 Since this is the root node, the rule specifically associated with the root is applied, leading to the
 315 generation of three child nodes.

316 At $t = 1$, each newly generated node is evaluated for whether reasoning should continue. As
 317 illustrated in the figure, the reasoning process terminates on Node 1 after applying the decision rule.
 318 In contrast, two subsequent reasoning nodes are generated based on Node 3. At $t = 2$, Node 5,
 319 which matches the backtracking rule, reinitiates the reasoning process together with its parent node.
 320 Specifically, Node 5 generates a subgraph $G_{(5)}^{\text{sub}}$, which includes Nodes 5, 2, and 8. The edges of
 321 the subgraph are represented by the pairs (5, 8) and (2, 8). Ultimately, Node 10 meets the necessary
 322 criteria and produces the final answer.

323 The resulting graph, along with the associated attributes, can subsequently be utilized for kernel-
 324 based graph analysis.

324 **4 BENCHMARK**



334 Figure 4: Category and subcategory distribution of questions in TiG
335 Benchmark.

Statistic	Number
Total questions	3529
Total categories	4
Total subcategories	27
Answer in text form	2313
Answer in function form	1207
Total implemented methods	55
Implemented methods per category	13.75

Table 1: Key statistics of
the TiG Benchmark

337 Based on the proposed TiG, we construct the corresponding TiG Benchmark to analyze prompt-
338 engineering-based LLM reasoning. For test data, we have collected a diverse set of reasoning tasks
339 with varying complexity, ensuring that the evaluation results can be reliably validated. Furthermore,
340 we have implemented different reasoning frameworks for each of these tasks using our proposed in-
341 frastructure. Specifically, we collected a total of four categories of data, namely Mathematical Logic
342 Puzzles, College Entrance Examination Questions, Coding Tasks, and Legal Cases. The detailed
343 composition of these datasets is illustrated in Figure 4. Among them, the problems in Mathematical
344 Logic Puzzles were constructed based on three different types of mathematical games. The prob-
345 lems in the College Entrance Examination Questions were derived from China’s national college
346 entrance examination. The Coding Tasks were collected from various online sources containing
347 programming problems. As for Legal Cases, they consist of publicly released court cases, where the
348 LLM is required to provide appropriate decisions based on the case descriptions. We provide details
349 concerning the collection procedure along with the provided data.

350 Answers for the questions come in two formats: textual and functional. Textual answers consist of
351 written content, such as selections for multiple-choice questions, authoritative judicial rulings from
352 official institutions, and other text-based responses. The functional answers, on the other hand, are
353 correctness-checking functions specifically designed for certain problems, e.g., a validation function
354 to check whether a submitted arithmetic expression evaluates to 24. Furthermore, for the questions,
355 our benchmark implements five state-of-the-art reasoning frameworks: CoT (Wei et al., 2022a), ToT
356 (Yao et al., 2023a), GoT (Besta et al., 2023), AoT (Sel et al., 2024), and EGoT (Shin & Kim, 2025).
357 Table 1 presents the specific statistics related to the benchmark.

358 **5 EXPERIMENTS**

360 **5.1 STATISTICAL EVALUATIONS**

363 Table 2: Results on Mathematical Logic Puzzles. The best performance is highlighted in **bold**, and
364 the second-best is indicated with underline.

Method	Accuracy	Time Cost (s)	Node Redundancy	Thought Redundancy	Count of Invalid Branches	Root-Answer Shortest Path Length
CoT	62.08 ± 2.11	26.87	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 1.51	4.68 ± 1.05
ToT	85.37 ± 7.03	78.16	71.96 ± 3.40	73.01 ± 6.42	6.10 ± 0.68	6.88 ± 0.03
AoT	89.55 ± 2.22	<u>37.42</u>	60.15 ± 5.04	63.31 ± 8.23	<u>3.06 ± 0.08</u>	3.03 ± 0.56
GoT	87.85 ± 8.47	74.65	79.01 ± 4.42	80.31 ± 6.03	5.47 ± 1.10	5.45 ± 0.98
EGoT	89.97 ± 7.67	70.78	<u>51.10 ± 3.01</u>	48.42 ± 7.04	4.99 ± 0.72	5.01 ± 1.43

371 Based on the constructed TiG Benchmark, we conducted a series of analytical experiments on the
372 implemented reasoning frameworks mentioned above, i.e., CoT, ToT, GoT, AoT, and EGoT, in or-
373 der to validate the usability of our infrastructure and further investigate the characteristics of these
374 reasoning paradigms. We first performed statistical evaluations across multiple metrics, including
375 accuracy, redundancy, and the number of invalid branches.

377 Specifically, based on our graph representation of reasoning logic, we propose the following new
378 metrics: node redundancy, thought redundancy, count of invalid branches, and root–answer shortest

378 Table 3: Results on College Entrance Examination Questions. The best performance is highlighted
 379 in **bold**, and the second-best is indicated with underline.
 380

Method	Accuracy	Time Cost (s)	Node Redundancy	Thought Redundancy	Count of Invalid Branches	Root-Answer Shortest Path Length
CoT	48.06 ± 1.25	<u>59.74</u>	0.00 ± 0.00	0.00 ± 0.00	0.17 ± 0.06	3.57 ± 1.46
ToT	62.42 ± 0.96	154.33	78.41 ± 4.10	74.60 ± 4.16	5.08 ± 0.94	5.63 ± 1.01
AoT	65.17 ± 2.11	65.96	53.21 ± 4.13	51.56 ± 4.10	4.10 ± 0.10	4.52 ± 1.34
GoT	67.22 ± 3.31	101.23	69.33 ± 3.31	67.67 ± 4.30	5.96 ± 1.32	5.17 ± 0.21
EGoT	69.89 ± 1.30	135.47	<u>50.51 ± 6.22</u>	<u>50.62 ± 6.10</u>	<u>5.03 ± 1.58</u>	<u>4.06 ± 0.85</u>

386 Table 4: Results on Coding Tasks. The best performance is highlighted in **bold**, and the second-best
 387 is indicated with underline.
 388

Method	Accuracy	Time Cost (s)	Node Redundancy	Thought Redundancy	Count of Invalid Branches	Root-Answer Shortest Path Length
CoT	48.23 ± 7.23	95.44	0.00 ± 0.00	0.00 ± 0.00	0.12 ± 0.08	3.56 ± 0.01
ToT	72.56 ± 9.21	143.92	74.96 ± 5.15	78.47 ± 6.01	3.56 ± 1.11	4.10 ± 0.75
AoT	82.12 ± 1.03	136.48	<u>52.12 ± 3.08</u>	<u>56.34 ± 2.45</u>	3.93 ± 1.23	<u>5.22 ± 1.33</u>
GoT	<u>72.45 ± 2.56</u>	140.01	72.56 ± 7.12	75.31 ± 9.04	<u>3.12 ± 1.14</u>	5.89 ± 0.31
EGoT	82.68 ± 9.32	<u>131.45</u>	69.01 ± 4.25	71.44 ± 4.56	4.15 ± 1.35	4.68 ± 1.10

395 Table 5: Results on Legal Cases. The best performance is highlighted in **bold**, and the second-best
 396 is indicated with underline.
 397

Method	Accuracy	Time Cost (s)	Node Redundancy	Thought Redundancy	Count of Invalid Branches	Root-Answer Shortest Path Length
CoT	43.33 ± 2.05	70.74	0.00 ± 0.00	0.00 ± 0.00	0.21 ± 0.06	3.57 ± 1.46
ToT	<u>58.33 ± 1.02</u>	154.33	71.41 ± 4.10	78.60 ± 4.16	5.08 ± 0.94	5.63 ± 1.01
AoT	61.67 ± 2.12	<u>113.96</u>	<u>65.21 ± 4.13</u>	<u>65.56 ± 4.10</u>	<u>4.10 ± 0.10</u>	4.52 ± 1.34
GoT	63.83 ± 3.31	135.47	76.33 ± 3.31	80.67 ± 3.40	4.96 ± 1.32	5.17 ± 0.21
EGoT	65.01 ± 1.30	129.23	65.51 ± 6.22	70.62 ± 6.10	6.23 ± 1.58	<u>4.16 ± 0.85</u>

405 path length. Node redundancy and thought redundancy respectively represent the percentage of
 406 redundant nodes and tokens relative to the total numbers. Redundant nodes and tokens are defined
 407 as those not lying on any path from the root node to the answer node, as well as the tokens contained
 408 within such nodes. The count of invalid branches measures the number of terminated reasoning
 409 branch that fail to acquire the answer. Root–answer shortest path length is the shortest distance from
 410 the root node to the answer node in the graph.

411 The experimental results are presented in Tables 2–5. As
 412 shown in the tables, the four newly introduced metrics
 413 enable a more in-depth analysis and comparison of the
 414 reasoning process. Based on these metrics, we observe
 415 that more complex reasoning frameworks tend to include
 416 a larger number of redundant nodes; however, the actual
 417 path length from the question to the answer does not vary
 418 significantly. When considered alongside the accuracy of
 419 different methods, this suggests that complex frameworks
 420 explore a wider range of possibilities in order to achieve
 421 higher accuracy. Figure 5 summarizes the average per-
 422 formance of each framework across different datasets. From
 423 the figure, it can be observed that the legal dataset is the
 424 most challenging one, while the coding dataset exhibits
 425 the largest performance gap among models.

426 5.2 DEEPER INSIGHTS

428 Additionally, we performed a similarity analysis between different reasoning approaches based on
 429 the RGWL kernel. First, we compared the similarity between the reasoning processes generated by
 430 a single reasoning framework, GoT, across different problems. The visualization of the comparison
 431 results is shown in Figure 6. It can be observed that for the coding and legal datasets, the consistency
 432 of reasoning between correct answers is higher, as indicated by higher RGWL kernel output. In

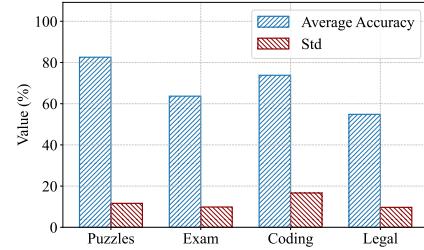


Figure 5: Average accuracy and standard deviation across datasets.

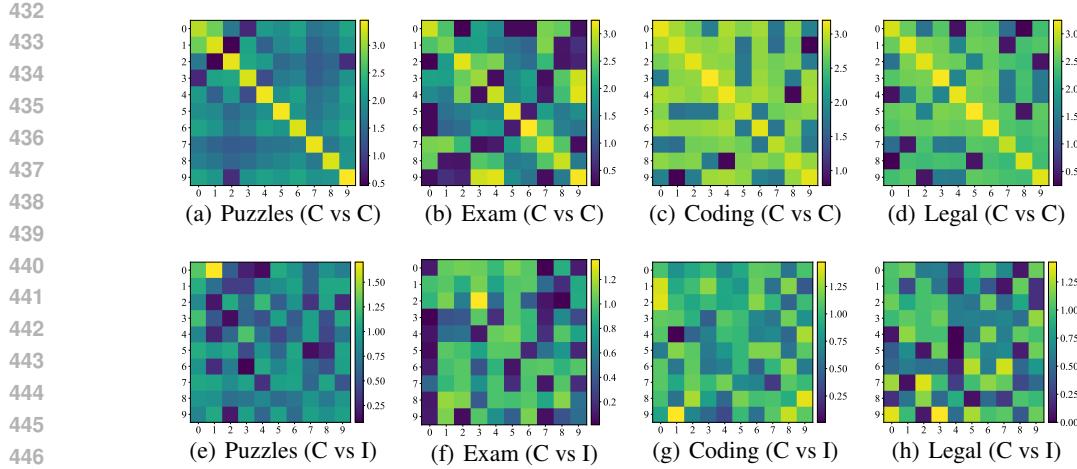


Figure 6: Visualization of RGWL results with GoT: “C vs C” compares ten reasoning graphs with correct answers, while “C vs I” compares ten correct-answer graphs with ten incorrect ones. Brighter colors indicate higher RGWL output and greater similarity.

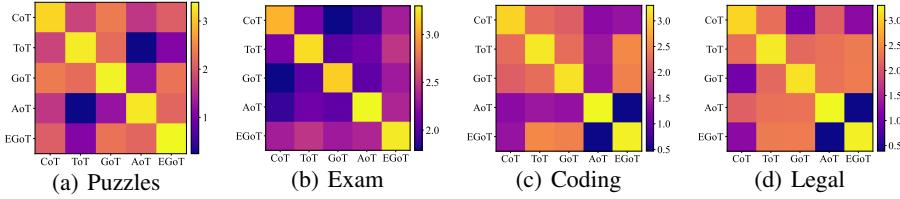


Figure 7: Visualization of RGWL computation results for different reasoning processes with different reasoning frameworks.

contrast, the distance between incorrect reasoning and correct reasoning is significantly greater than the distance between two correct reasoning processes.

Furthermore, we compared the similarity between the reasoning processes generated by different frameworks across various problems, with the results shown in Figure 7. It can be observed that the similarity between reasoning processes produced by different frameworks is highest in the legal dataset, which is consistent with our previous experimental findings. This observation indicates that the knowledge structures and decision rules in the legal domain are relatively stable and standardized, leaving less room for divergent reasoning paths. As a result, even when different reasoning frameworks are applied, they tend to converge on similar reasoning trajectories and final conclusions.

In contrast, datasets such as coding tasks or mathematical puzzles allow multiple solution strategies and a more open-ended reasoning space, leading to lower similarity between frameworks. This demonstrates that our graph-based analysis captures not only outcome accuracy but also the structural consistency and diversity of reasoning processes.

6 CONCLUSION

In this paper, we propose the TiG infrastructure based on graph structures, which enables the construction of diverse prompt-engineering-based LLM reasoning frameworks in a unified and streamlined manner. Building upon TiG, we introduce a broader set of reasoning logic evaluation metrics and develop a benchmark for comparing different reasoning frameworks, on which we conduct a series of experiments and analyses.

486 REPRODUCIBILITY STATEMENT
487488 Our theoretical results have been rigorously proven, and the corresponding proofs are provided in
489 Appendix D. Additionally, our experiments provide both data and code to ensure reproducibility.
490 These resources are included in the anonymous link [https://anonymous.4open.science/](https://anonymous.4open.science/r/210-5DD5/)
491 [r/210-5DD5/](https://anonymous.4open.science/r/210-5DD5/), with further details available in the accompanying README.md file.
492493 REFERENCES
494495 Amazon AGI, Aaron Langford, Aayush Shah, Abhanshu Gupta, Abhimanyu Bhatter, Abhinav
496 Goyal, Abhinav Mathur, Abhinav Mohanty, Abhishek Kumar, Abhishek Sethi, Abi Komma,
497 Abner Pena, Achin Jain, Adam Kunysz, Adam Oprchal, Adarsh Singh, Aditya Rawal, Adok
498 Achar Budihal Prasad, Adrià de Gispert, Agnika Kumar, Aishwarya Aryamane, Ajay Nair, Akilan
499 M, Akshaya Iyengar, Akshaya Vishnu Kudlu Shanbhogue, Alan He, Alessandra Cervone, Alex
500 Loeb, Alex Zhang, Alexander Fu, Alexander Lisnichenko, Alexander Zhipa, Alexandros Potamianos,
501 Ali Kebarighotbi, Aliakbar Daronkolaei, Alok Parmesh, Amanjot Kaur Samra, Ameen
502 Khan, Amer Rez, Amir Saffari, Amit Agarwall, Amit Jhindal, Amith R. Mamidala, Ammar
503 Asmro, Amulya Ballakur, Anand Mishra, Anand Sridharan, Anastasiia Dubinina, Andre Lenz,
504 Andreas Doerr, Andrew Keating, Andrew Leaver, Andrew Smith, Andrew Wirth, Andy Davey,
505 Andy Rosenbaum, Andy Sohn, Angela Chan, Aniket Chakrabarti, Anil Ramakrishna, Anirban
506 Roy, Anita Iyer, Anjali Narayan-Chen, Ankith Yennu, Anna Dabrowska, Anna Gawlowska,
507 Anna Rumshisky, Anna Turek, Anoop Deoras, Anton Bezruchkin, Anup Prasad, Anupam De-
508 wan, Anwith Kiran, Apoorv Gupta, Aram Galstyan, Aravind Manoharan, Arijit Biswas, Arindam
509 Mandal, Arpit Gupta, Arsamkhan Pathan, Arun Nagarajan, Arushan Rajasekaram, Arvind Sun-
510 dararajan, Ashwin Ganesan, Ashwin Swaminathan, Athanasios Mouchtaris, Audrey Cham-
511 peau, Avik Ray, Ayush Jaiswal, Ayush Sharma, Bailey Keefer, Balamurugan Muthiah, Beatriz
512 Leon-Millan, Ben Koopman, Ben Li, Benjamin Biggs, Benjamin Ott, Bhanukiran Vinzamuri,
513 Bharath Venkatesh, and Bhavana Ganesh. The amazon nova family of models: Technical re-
514 port and model card. *CoRR*, abs/2506.12103, 2025. doi: 10.48550/ARXIV.2506.12103. URL
<https://doi.org/10.48550/arXiv.2506.12103>.
515516 Meta AI. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.517 Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Jo-
518 han Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin
519 Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Tim-
520 othy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald
521 Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan
522 Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha
523 Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Dani-
524 helka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati,
525 Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A fam-
526 ily of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ARXIV.
527 2312.11805. URL <https://doi.org/10.48550/arXiv.2312.11805>.
528529 Anthropic. Model card and evaluations for claude 2. *Anthropic*, 2023a. Available at <https://www.anthropic.com>.530 Anthropic. Ai governance and accountability: An analysis of anthropic’s claude. *arXiv preprint*
531 *arXiv:2407.01557*, 2023b.532 Anthropic. The claude 3 model family: Opus, sonnet, haiku. *arXiv preprint arXiv:2404.13813*,
533 2024.534 Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Shikhar Vashishth, Sriram Gana-
535 pathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. Llm augmented llms: Expanding
536 capabilities through composition, 2024. URL <https://arxiv.org/abs/2401.02412>.
537538 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda,
539 Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczek, and Torsten Hoefer.
Graph of thoughts: Solving elaborate problems with large language models, 2023.

540 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Giani-
 541 nazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczek, and Torsten Hoe-
 542 fler. Graph of thoughts: Solving elaborate problems with large language models. In Michael J.
 543 Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Ar-
 544 tificial Intelligence, February 20-27, 2024, Vancouver, Canada*, pp. 17682–17690. AAAI Press,
 545 2024a. doi: 10.1609/AAAI.V38I16.29720. URL <https://doi.org/10.1609/aaai.v38i16.29720>.

546

547 Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Giani-
 548 nazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczek, et al. Graph of
 549 thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI
 550 conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024b.

551

552 Luca Beurer-Kellner, Mark Niklas Müller, Marc Fischer, and Martin T. Vechev. Prompt sketching for
 553 large language models. In *Forty-first International Conference on Machine Learning, ICML 2024,
 554 Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=2Yu5FWdzde>.

555

556 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhari-
 557 wal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal,
 558 Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M.
 559 Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin,
 560 Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford,
 561 Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle,
 562 Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances
 563 in Neural Information Processing Systems 33: Annual Conference on Neural Information Pro-
 564 cessing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

565

566 Wenhui Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting:
 567 Disentangling computation from reasoning for numerical reasoning tasks. *Trans. Mach. Learn.
 568 Res.*, 2023, 2023. URL <https://openreview.net/forum?id=YfZ4ZPt8zd>.

569

570 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam
 571 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh,
 572 Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam
 573 Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James
 574 Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Lev-
 575 skaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin
 576 Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret
 577 Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivan Agrawal, Mark Omernick,
 578 Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica
 579 Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Bren-
 580 nan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas
 581 Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways.
 582 *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL <https://jmlr.org/papers/v24/22-1144.html>.

583

584 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and
 585 Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge,
 586 2018.

587

588 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 589 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 590 Schulman. Training verifiers to solve math word problems, 2021.

591

592 Mengyao Cui et al. Introduction to the k-means clustering algorithm based on the elbow method.
 593 *Accounting, Auditing and Finance*, 1(1):5–8, 2020.

594

595 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep
 596 bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and
 597 Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of
 598 the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*

594 2019, *Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–
 595 4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL
 596 <https://doi.org/10.18653/v1/n19-1423>.

597 Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz,
 598 and Jason Weston. Chain-of-verification reduces hallucination in large language models. In
 599 *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and*
 600 *virtual meeting, August 11-16, 2024*, pp. 3563–3578. Association for Computational Linguistics,
 601 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.212. URL <https://doi.org/10.18653/v1/2024.findings-acl.212>.

602 Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting
 603 with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting*
 604 *of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok,*
 605 *Thailand, August 11-16, 2024*, pp. 1330–1350. Association for Computational Linguistics, 2024.
 606 doi: 10.18653/V1/2024.ACL-LONG.73. URL <https://doi.org/10.18653/v1/2024.acl-long.73>.

607 Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner.
 608 Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs, 2019.

609 Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards re-
 610 vealing the mystery behind chain of thought: A theoretical perspective. In Alice Oh, Tris-
 611 stan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Ad-*
 612 *vances in Neural Information Processing Systems 36: Annual Conference on Neural Infor-*
 613 *mation Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16,*
 614 *2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

615 Hang Gao, Chenhao Zhang, Tie Wang, Junsuo Zhao, Fengge Wu, Changwen Zheng, and Huap-
 616 ing Liu. Learn to think: Bootstrapping LLM reasoning capability through graph represen-
 617 tation learning. *CoRR*, abs/2505.06321, 2025. doi: 10.48550/ARXIV.2505.06321. URL
 618 <https://doi.org/10.48550/arXiv.2505.06321>.

619 Cobus Greyling. A benchmark for verifying chain-of-thought. <https://cobusgreyling.medium.com/a-benchmark-for-verifying-chain-of-thought-904db5ebeef>,
 620 Feb 2024.

621 Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma,
 622 Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhitong Hu. LLM reasoners: New evalua-
 623 tion, library, and analysis of step-by-step reasoning with large language models. *CoRR*, abs/2404.05221,
 624 2024. doi: 10.48550/ARXIV.2404.05221. URL <https://doi.org/10.48550/arXiv.2404.05221>.

625 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob
 626 Steinhardt. Measuring massive multitask language understanding, 2021.

627 Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey.
 628 In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pp. 1049–
 629 1065. Association for Computational Linguistics (ACL), 2023.

630 Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot,
 631 Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,
 632 Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas
 633 Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023.
 634 doi: 10.48550/ARXIV.2310.06825. URL <https://doi.org/10.48550/arXiv.2310.06825>.

635 Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan
 636 Jin, Claire Guo, Shen Yan, Bo Zhang, Chaoyou Fu, Peng Gao, and Hongsheng Li. Mme-cot:
 637 Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness,
 638 and efficiency. *CoRR*, abs/2502.09621, 2025. doi: 10.48550/ARXIV.2502.09621. URL <https://doi.org/10.48550/arXiv.2502.09621>.

648 Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu
 649 Soriciut. ALBERT: A lite BERT for self-supervised learning of language representations. In
 650 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia,*
 651 *April 26-30, 2020.* OpenReview.net, 2020. URL <https://openreview.net/forum?id=H1eA7AEtvS>.

653 Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
 654 Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and
 655 Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo
 656 Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin
 657 (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural*
 658 *Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

659 Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang
 660 Yang, and Xing Xie. Large language models understand and can be enhanced by emotional
 661 stimuli, 2023a. URL <https://arxiv.org/abs/2307.11760>.

663 Jia Li, Ge Li, Yongmin Li, and Zhi Jin. Structured chain-of-thought prompting for code generation,
 664 2023b. URL <https://arxiv.org/abs/2305.06599>.

666 Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga,
 667 Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan,
 668 Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana
 669 Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong,
 670 Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksek-
 671 gonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson,
 672 Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Silei Xu, Stefano Ermon, Tatsunori
 673 Hashimoto, Tianyi Zhang, Tiana Lusha, Tony Z. Zhao, Valentina Pyatkin, Vishaal Chavalier, Wei-
 674 jia Shi, Wenlong Zhao, Yifan Mai, Yuhui Zhang, Yuta Koreeda, Yujin Kim, Yiming Zuo, Ziyi Wu,
 675 Amir Pouran Ben Veyseh, Avanika Narayan, Chelsea Finn, Christopher H. Lin, Ellie Pavlick,
 676 Emily Alsentzer, Ezi O. Young, Fabio Viola, Fernando Pérez, Jean-Philippe Vert, Jiasheng Gu,
 677 John Hewitt, Juraj Juraska, Katherine A. Keith, Kevin K. Yang, Kevin R. McKee, Kyle Richard-
 678 son, Linyong Nan, Mahdi Namazifar, Maura R. Grossman, Michael S. Bernstein, Noah A. Smith,
 679 Noah D. Goodman, Pang Wei Koh, Qinyuan Ye, Robert Frank, Rohan Sikand, Ryan T. Cotterell,
 680 Sanmi Koyejo, Sara Hooker, Sebastian Riedel, Shiori Sagawa, Surya Ganguli, Tatsuki Koyama,
 681 Thomas Icard, Tobias Gerstenberg, William Wang, Yejin Choi, Yoav Artzi, Yushi Hu, and Ziyi
 682 Yang. Holistic evaluation of language models, 2022.

682 Yinhai Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike
 683 Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining
 684 approach. *CoRR*, abs/1907.11692, 2019. URL <http://arxiv.org/abs/1907.11692>.

686 Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. Directgpt: A direct manipulation
 687 interface to interact with large language models. In Florian 'Floyd' Mueller, Penny Kyburz,
 688 Julie R. Williamson, Corina Sas, Max L. Wilson, Phoebe O. Toups Dugas, and Irina Shklovski
 689 (eds.), *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024,*
 690 *Honolulu, HI, USA, May 11-16, 2024*, pp. 975:1–975:16. ACM, 2024. doi: 10.1145/3613904.
 691 3642462. URL <https://doi.org/10.1145/3613904.3642462>.

692 Tyler McDonald, Anthony Colosimo, Yifeng Li, and Ali Emami. Can we afford the perfect
 693 prompt? balancing cost and accuracy with the economical prompting index. *arXiv preprint*
 694 *arXiv:2412.01690*, 2024.

695 AI Meta. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation.
 696 <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>, checked on, 4(7):2025, 2025.

697

698 Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-
 699 thought prompting for large multimodal models. In *IEEE/CVF Conference on Computer Vision*
 700 *and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 14420–14431.
 701 IEEE, 2024. doi: 10.1109/CVPR52733.2024.01367. URL <https://doi.org/10.1109/CVPR52733.2024.01367>.

702 Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model
 703 for english tweets. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference*
 704 *on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020*
 705 *- Demos, Online, November 16-20, 2020*, pp. 9–14. Association for Computational Linguistics,
 706 2020. doi: 10.18653/V1/2020.EMNLP-DEMOS.2. URL <https://doi.org/10.18653/v1/2020.emnlp-demos.2>.

707

708 OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023a. doi: 10.48550/ARXIV.2303.
 709 08774. URL <https://doi.org/10.48550/arXiv.2303.08774>.

710

711 OpenAI. Openai evals. <https://github.com/openai/evals>, Mar 2023b.

712

713 Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli,
 714 Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refined-
 715 web dataset for falcon LLM: outperforming curated corpora with web data only. In Alice Oh,
 716 Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Ad-*
 717 *vances in Neural Information Processing Systems 36: Annual Conference on Neural Information*
 718 *Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

719

720 Alec Radford. Improving language understanding by generative pre-training. 2018.

721

722 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language
 723 models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

724

725 Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-
 726 networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of*
 727 *the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th Inter-*
 728 *national Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong,*
 729 *China, November 3-7, 2019*, pp. 3980–3990. Association for Computational Linguistics, 2019.
 730 doi: 10.18653/V1/D19-1410. URL <https://doi.org/10.18653/v1/D19-1410>.

731

732 Tal Ridnik, Dedy Kredo, and Itamar Friedman. Code generation with alphacodium: From prompt
 733 engineering to flow engineering, 2024. URL <https://arxiv.org/abs/2401.08500>.

734

735 Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi
 736 Adi, Jingyu Liu, Tal Remez, Jérémie Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton,
 737 Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez,
 738 Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and
 739 Gabriel Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023.
 740 doi: 10.48550/ARXIV.2308.12950. URL <https://doi.org/10.48550/arXiv.2308.12950>.

741

742 Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si,
 743 Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav
 744 Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao,
 745 Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Gonçarencio,
 746 Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat,
 747 Alexander Hoyle, and Philip Resnik. The prompt report: A systematic survey of prompt engi-
 748 neering techniques, 2025. URL <https://arxiv.org/abs/2406.06608>.

749

750 Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of thoughts:
 751 Enhancing exploration of ideas in large language models. In *Forty-first International Conference*
 752 *on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
 753 URL <https://openreview.net/forum?id=KJL2b6BthC>.

754

755 Chirag Shah. From prompt engineering to prompt science with humans in the loop. *Communications*
 756 *of the ACM*, May 2025. doi: 10.1145/3709599. URL <https://dl.acm.org/doi/10.1145/3709599>.

757

758 Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borg-
 759 wardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12:2539–2561, 2011. doi: 10.5555/1953048.2078187. URL <https://dl.acm.org/doi/10.5555/1953048.2078187>.

756 Sunguk Shin and Youngjoon Kim. Enhancing graph of thought: Enhancing prompts with LLM
 757 rationales and dynamic temperature control. In *The Thirteenth International Conference on Learn-
 758 ing Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL
 759 <https://openreview.net/forum?id=132IrJtpOP>.

760 Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad
 761 Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning
 762 models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

763

764 Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second
 765 Edition*. Adaptive computation and machine learning. MIT Press, 2000. ISBN 978-0-262-19440-
 766 2.

767 Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoaib, Adam Fisch,
 768 Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor
 769 Lewkowycz, Akshay Neelakantan, Alan Schelten, Aleksandra Piktus, Alex Ray, Alex Warstadt,
 770 Alexander W. Kocurek, Ali Safaya, Ali Tazary, Alice Xiang, Alicia Parrish, Alon Talmor, Aman
 771 Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S.
 772 Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai,
 773 Andrew LaFlair, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Ani-
 774 mesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa
 775 Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mollokandov, Ashish Sabharwal, Austin
 776 Herrick, Avia Efrat, Aviral Kumar, Ayla Karakaş, B. Ryan Roberts, Barret Zoph, Bartłomiej
 777 Bojanowski, Batuhan Özürt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno
 778 Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Cameron Diao, Cameron Dour, Catherine
 779 Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin
 780 Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris DuVuono, Chris La Ré, Christo-
 781 pher D. Manning, Christopher Potts, Christopher A. Choquette-Choo, Chu-Cheng Lin, Clarissa
 782 Casimiro, Colin Raffel, Collins Aghaular, Connor Coley, Conrad Ko, Cristina Garbacea, Damien
 783 Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi,
 784 Daniel Levy, Daniel M. Ziegler, Daniel Roberts, Danny Hernandez, Danqi Chen, David Dohan,
 785 David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko,
 786 Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimi-
 787 tri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Se-
 788 gal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola,
 789 Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan
 790 Jerzak, Ethan Kim, Eunice Engefu Manyasi, Eva Rotenberg, Eyal Ben-David, Eyal Lupu, Fanyue
 791 Xia, Fatemeh Siar, Fernando Diaz, Francis Ferraro, Frank Zipcode, Frieda Rong, Gaurav Mishra,
 792 Genta Indra Winata, Gerard de Melo, Germán Kruszewski Ghazi, Girish Sastry, Giovanni De
 793 Toni, Giovanni Piloto, Gloria Wang, Gonzalo Jaimovich-López, Gregor Betz, Guy Gur-Ari, Hana
 794 Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar,
 795 Henry Shevlin, Hinrich Schütze, Hiromu Yakra, Hongming Zhang, Hugh Meehan, Hyung Won
 796 Chung, Ian Ng, Igor Krawczuk, Ipek Ensari, Isaac Caswell, Isaac Ha, Ishaan Gulrajani, Itay
 797 Levy, Ivan Vulic, Jacob Austin, Jacob Bieganek, Jacob Eisenstein, Jacob Hart, Jacob Devlin,
 798 Jakub Semeniuk, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Jared
 799 Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Baldridge, Jason Grossman, Jason Rute,
 800 Jason Yosinski, Jaspreet Singh, Javier González-Abad, Jelle Bosscher, Jennifer Marsh, Jeremy
 801 Kim, Jeroen Taal, Jesse Engel, Jesse Levinson, Jessica Wang, Jiaming Luo, Jiao Sun, Jifan Chen,
 802 Jina Suh, Jinfeng Rao, Jiyuan Zhong, Joan Waweru, John Burden, John Miller, John U. Balis,
 803 Jonathan Berant, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Josie, Joy Hsu, Juha Ala-
 804 Rantala, Julia Rayz, Julian Eisenschlos, Justin Wong, Yacine Jernite, Yallow Uri, Yaron Singer,
 805 Yejin Choi, Yichi Zhang, Yiding Hao, Yifu Chen, Yifan Xu, Yilun Zhao, Yoav Artzi, Yoav Gold-
 806 berg, Young-Suk Lee, Yuntao Bai, Yuta Takahashi, Zachary Kenton, Zanele Mthembu, Zeqiu
 807 Wu, Zhaofeng Wu, and Zonglin Li. Beyond the imitation game: Quantifying and extrapolating
 808 the capabilities of language models, 2022.

809 Keith Stenning and Michiel Van Lambalgen. *Human reasoning and cognitive science*. MIT Press,
 810 2012.

811 Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu,
 812 Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang,

810 Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng
 811 Wang. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and
 812 generation. *CoRR*, abs/2107.02137, 2021. URL <https://arxiv.org/abs/2107.02137>.

813
 814 Mirac Suzgun, Nathan Scales, Nathanael Schärlí, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
 815 Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-
 816 bench tasks and whether chain-of-thought can solve them, 2022.

817 Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al.
 818 Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

819 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,
 820 Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von
 821 Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman
 822 Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on
 823 Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp.
 824 5998–6008, 2017.

825 Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning Aware Self-Consistency: Leveraging
 826 Reasoning Paths for Efficient LLM Sampling. In *Proceedings of the 2025 Conference of the
 827 North American Chapter of the Association for Computational Linguistics: Human Language
 828 Technologies (NAACL 2025)*. Association for Computational Linguistics, 2025. URL <https://aclanthology.org/2025.naacl-long.184>.

829
 830 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi,
 831 Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language
 832 models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.),
 833 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Infor-
 834 mation Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December
 835 9, 2022*, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

836
 837 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
 838 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in
 839 neural information processing systems*, 35:24824–24837, 2022b.

840
 841 An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li,
 842 Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang,
 843 Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren
 844 Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang,
 845 Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin,
 846 Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong
 847 Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu,
 848 Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru
 849 Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024.
 850 doi: 10.48550/ARXIV.2407.10671. URL <https://doi.org/10.48550/arXiv.2407.10671>.

851
 852 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik
 853 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a.

854
 855 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik
 856 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In
 857 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine
 858 (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neu-
 859 ral Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December
 860 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

861
 862 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan
 863 Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International
 864 Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenRe-
 865 view.net, 2023c. URL https://openreview.net/forum?id=WE_vluYUL-X.

864 Yao Yao, Zuchao Li, and Hai Zhao. Got: Effective graph-of-thought reasoning in lan-
 865 guage models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Find-
 866 ings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico,
 867 June 16-21, 2024*, pp. 2901–2921. Association for Computational Linguistics, 2024. doi:
 868 10.18653/V1/2024.FINDINGS-NAACL.183. URL <https://doi.org/10.18653/v1/2024.findings-naacl.183>.

870 Weihao Yu, Zihang Jiang, Yanfei Dong, Jiashi Feng, Nesreen K. Ahmed, Dan Roth, and Xiaojun
 871 Quan. Reclor: A reading comprehension dataset requiring logical reasoning, 2020.

872 Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. Chain-
 873 of-note: Enhancing robustness in retrieval-augmented language models. *CoRR*, abs/2311.09210,
 874 2023. doi: 10.48550/ARXIV.2311.09210. URL <https://doi.org/10.48550/arXiv.2311.09210>.

875 Weizhe Yuan, Richard Yuzhong Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu,
 876 and Jason Weston. Self-rewarding language models, 2025. URL <https://arxiv.org/abs/2401.10020>.

877 Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago
 878 Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird:
 879 Transformers for longer sequences. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell,
 880 Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing
 881 Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020,
 882 December 6-12, 2020, virtual*, 2020.

883 Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompt-
 884 ing in large language models. In *The Eleventh International Conference on Learning Repre-
 885 sentations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/forum?id=5NTt8GFjUHkr>.

886 Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter.
 887 Enhancing zero-shot chain-of-thought reasoning in large language models through logic. In Nico-
 888 letta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nian-
 889 wen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Lin-
 890 guistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino,
 891 Italy*, pp. 6144–6166. ELRA and ICCL, 2024. URL <https://aclanthology.org/2024.lrec-main.543>.

892 Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,
 893 Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica.
 894 Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

895 Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and
 896 Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh Inter-
 897 national Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
 898 OpenReview.net, 2023a. URL <https://openreview.net/forum?id=92gwk82DE->.

899 Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and
 900 Jianbing Shen. Thread of thought unraveling chaotic contexts. *CoRR*, abs/2311.08734, 2023b.
 901 doi: 10.48550/ARXIV.2311.08734. URL <https://doi.org/10.48550/arXiv.2311.08734>.

902
 903
 904
 905
 906
 907
 908
 909
 910
 911
 912
 913
 914
 915
 916
 917

918 A USAGE OF LARGE LANGUAGE MODEL
919920 In our paper, we used LLMs to assist with polishing the writing, including correcting grammatical
921 errors and making the sentences more consistent with academic English writing conventions.
922923 B EXTENDED RELATED WORKS
924925 B.1 LARGE LANGUAGE MODELS
926928 Since the introduction of the Transformer architecture (Vaswani et al., 2017), LLMs have under-
929 gone rapid and transformative development. The Transformer, with its self-attention mechanism
930 and strong parallelism capabilities, quickly became the foundational architecture for modern neural
931 language models, setting a universal paradigm for subsequent designs.932 Based on this architecture, OpenAI released the initial versions of the GPT series in 2018 (Radford,
933 2018), leveraging autoregressive language modeling to achieve notable performance in text gen-
934 eration. This was followed by GPT-2 (Radford et al., 2019), which significantly expanded model
935 capacity and demonstrated strong generalization across diverse tasks. GPT-3, introduced in 2020
936 (Brown et al., 2020), scaled to 175 billion parameters and marked a major leap in LLM capabilities.
937 GPT-3.5 further optimized inference efficiency and improved contextual understanding, especially
938 in dialogue-oriented tasks. GPT-4 (OpenAI, 2023a) introduced substantial advances in logical rea-
939 soning, knowledge integration, and alignment with human values, enabling its application to more
940 complex and multi-modal tasks. The most recent iteration, GPT-4o, focuses on enhancing safety, ro-
941 bustness, and ethical alignment, making it particularly well-suited for high-stakes decision-making
942 scenarios.943 In parallel with the GPT lineage, BERT (Devlin et al., 2019) was introduced in 2019, pione-
944 ring bidirectional contextual learning through masked language modeling (MLM), which marked a
945 departure from the limitations of unidirectional models. Building on BERT, numerous improved
946 variants have been proposed: RoBERTa (Liu et al., 2019) removed the next sentence prediction ob-
947 jective and used more extensive pretraining data; ALBERT (Lan et al., 2020) introduced parameter
948 sharing and factorized embedding layers to reduce redundancy; BERTweet (Nguyen et al., 2020)
949 targeted social media text processing; and BigBird (Zaheer et al., 2020) employed sparse attention
950 mechanisms to effectively handle longer input sequences.951 More recently, a wave of novel LLMs has emerged, further diversifying the research landscape.
952 Meta’s LLaMA series (Touvron et al., 2023; Rozière et al., 2023; AI, 2024) has contributed signifi-
953 cantly to the development of efficient, open-source, and lightweight models. Subsequently, LLaMA
954 4 was introduced as an open-source large language model series featuring a Mixture-of-Experts ar-
955 chitecture, native multimodal capabilities, industry-leading extended context length, and enhanced
956 multilingual support, achieving significant breakthroughs in both performance and efficiency (Meta,
957 2025). Meanwhile, Anthropic’s Claude series (Anthropic, 2023b;a; 2024) emphasizes model align-
958 ment, safety, and controllability, proposing new mechanisms for responsible AI deployment. In
959 addition, several other notable models have been introduced in recent years, including Google’s
960 Gemini (Anil et al., 2023), Alibaba’s Tongyi Qianwen (Yang et al., 2024), Baidu’s ERNIE (Sun
961 et al., 2021), Amazon Nova (AGI et al., 2025), Mistral (Jiang et al., 2023), Falcon (Penedo et al.,
962 2023), and PaLM (Chowdhery et al., 2023). Each of these models introduces distinct innovations
963 in architectural design, code generation, multilingual capability, training efficiency, or open-access
964 availability, collectively advancing the capabilities and diversity of the LLM ecosystem.965 B.2 PROMPT ENGINEERING
966967 With the widespread deployment of LLMs, their capabilities in natural language understanding and
968 generation have continued to surpass expectations. However, effectively guiding these models to
969 produce accurate, logically coherent, and structurally consistent outputs remains a key challenge.
970 Prompt Engineering has emerged as a critical solution to this issue and has rapidly evolved in recent
971 years, forming a systematic framework encompassing strategies such as reasoning enhancement,
972 hallucination mitigation, structured task adaptation, and interactive optimization.

To improve the reasoning capabilities of LLMs, researchers initially proposed the Chain of Thought (CoT) approach, which significantly enhances the model’s deductive reasoning in complex tasks—such as mathematical problem solving and textual inference—by prompting it to generate explicit intermediate reasoning steps (Wei et al., 2022a). Building upon this foundation, methods such as Program of Thoughts (PoT) and Structured Chain-of-Thought (SCoT) further modularize the reasoning process, making them particularly effective for code generation, logic programming, and multi-stage computation tasks (Chen et al., 2023; Li et al., 2023b). Additionally, techniques like Flow Engineering have been proposed to improve the semantic consistency and execution fidelity of code-related prompts, thereby expanding the design space for structured prompt generation (Ridnik et al., 2024).

Hallucination mitigation constitutes another central focus of prompt engineering. Classical approaches such as Retrieval-Augmented Generation (RAG) integrate external knowledge sources into the generation process, providing factual grounding and improving the factual accuracy of model outputs from the outset (Lewis et al., 2020). Meanwhile, post-hoc verification methods—such as Chain-of-Verification (CoVe), Chain-of-Note (CoN), and Chain-of-Knowledge (CoK)—introduce layered review mechanisms, filtering out false or inconsistent content through multi-stage validation, citation checking, and cross-examination (Dhuliawala et al., 2024; Yu et al., 2023).

As LLMs are increasingly deployed in open-domain environments, enhancing their interactivity and understanding of user intent has become a key extension of prompt engineering. Interactive Question Answering (Interactive QA) frameworks allow models to obtain real-time feedback through multi-turn dialogue, enabling dynamic adjustment of responses (Yao et al., 2023c; Masson et al., 2024). Concurrently, research has focused on the automation and personalization of prompt selection processes—for example, through techniques that match prompt templates to task-specific contexts (Zhou et al., 2023a)—as well as on modeling user intent for tasks involving emotion control and stylistic adaptation (Diao et al., 2024; Li et al., 2023a).

Currently, prompt engineering is undergoing a transition from empirically driven practices to a more theory-guided scientific paradigm. Scholars have proposed systematic frameworks that integrate diverse prompting techniques (Schulhoff et al., 2025), while also exploring human-in-the-loop methodologies to enable more controllable and robust generation systems (Shah, 2025). Moreover, emerging methods such as Self-Rewarding Language Models and LLM-Augmented LLMs aim to build prompt learning systems that possess self-evaluation and cooperative expansion capabilities, signaling a broader shift toward modular, self-optimizing prompt engineering paradigms (Yuan et al., 2025; Bansal et al., 2024).

B.3 REASONING WITHIN PROMPTING

Recent research has increasingly focused on designing logically consistent prompts that improve reasoning performance, enabling LLMs to tackle complex tasks more effectively. Following the Chain-of-Thought (CoT) framework (Wei et al., 2022a), Auto-CoT (Zhang et al., 2023) introduces an automated pipeline that samples diverse problems and utilizes zero-shot CoT outputs, followed by post-processing to filter and optimize reasoning chains. LogiCoT (Zhao et al., 2024) integrates the principle of Reductio ad Absurdum from symbolic logic to iteratively verify and correct the reasoning process, thereby reinforcing logical rigor.

In parallel, Prompt Sketching (Beurer-Kellner et al., 2024) proposes a structured prompt template to steer the model’s reasoning within a predefined format, achieving more controllable logical pathways. Compositional Chain-of-Thought (CCoT) (Mitra et al., 2024) further extends the CoT framework to multi-modal scenarios, promoting cross-modal reasoning. Other studies (Feng et al., 2023) have also explored the theoretical and empirical performance of CoT on mathematical reasoning tasks. To improve output robustness, the Reasoning-Aware Self-Consistency (RASC) framework (Wan et al., 2025) augments traditional self-consistency mechanisms with a dynamic evaluation of the coherence between each reasoning trace and its final answer. By integrating score-driven stopping strategies and weighted voting, RASC not only reduces sampling cost by approximately 70% but also improves predictive accuracy and reasoning fidelity.

Beyond linear reasoning chains, recent efforts have explored more complex topological representations of thought to enhance LLMs’ logical modeling capabilities. The Tree of Thoughts (ToT) (Yao et al., 2023b) introduces a branching structure that enables models to search and evaluate multiple

1026 reasoning paths. The Graph of Thoughts (GoT) (Besta et al., 2024a) models reasoning as a graph-based process, well-suited for complex, multi-step, and multi-source reasoning tasks. Similarly, Graph-of-Thought Reasoning (GoTR) (Yao et al., 2024) explicitly encodes inter-node relationships within reasoning paths, particularly effective for integrating heterogeneous information in multi-modal contexts. Building on these, Enhancing Graph of Thoughts (EGoT) (Shin & Kim, 2025) further optimizes the design of inference paths within graph structures, enhancing both reasoning efficiency and consistency for complex tasks.

1033 Additionally, several studies have proposed structurally organized reasoning paradigms. Thread of Thought (ThoT) (Zhou et al., 2023b) advocates decomposing complex problems into hierarchical 1034 and sequential “threads of thought” to facilitate more systematic reasoning. Meanwhile, Algorithm 1035 of Thoughts (AoT) (Sel et al., 2024) embeds algorithmic reasoning structures into the prompt 1036 context, guiding the model to emulate procedural execution. This strategy leverages the recursive 1037 dynamics of LLMs, enabling the construction and exploration of sophisticated reasoning paths with 1038 minimal interaction.

1041 B.4 BENCHMARKING LLM REASONING

1043 In recent years, research on the reasoning abilities of LLMs has been growing rapidly. As a result, 1044 evaluating the reasoning capabilities of LLMs has become a hot topic in research. MMLU 1045 (Hendrycks et al., 2021) provides a large-scale multi-task language understanding benchmark that 1046 covers tasks from 57 different domains. Similarly, BIG-bench (Srivastava et al., 2022) includes 1047 over 204 diverse tasks aimed at comprehensively testing the model’s abilities. HELM (Liang et al., 1048 2022) emphasizes a holistic perspective on model evaluation, looking at both the scenarios and the 1049 metrics to gain a thorough understanding of a model’s capabilities. MT-Bench (Zheng et al., 2023) 1050 is a multi-turn dialogue reasoning benchmark that focuses on the reasoning abilities in multi-turn 1051 dialogue scenarios. OpenAI Eval (OpenAI, 2023b) is a general evaluation framework designed 1052 to facilitate the development and sharing of evaluation benchmarks for LLMs by the community. 1053 For reasoning evaluation, BBH (Suzgun et al., 2022) is a set of the 23 most challenging sub-tasks 1054 from BIG-bench, specifically assessing models’ performance on complex reasoning tasks. GSM8K 1055 (Cobbe et al., 2021) is a dataset containing 8,500 elementary school math application problems, 1056 used to evaluate the model’s mathematical reasoning abilities. The MATH (Hendrycks et al., 2021) 1057 dataset includes 12,500 high school math competition questions, primarily testing the model’s 1058 reasoning abilities on advanced math problems. ARC (Clark et al., 2018) is a dataset designed to 1059 evaluate LLMs’ abilities in scientific question answering and common-sense reasoning. DROP (Dua 1060 et al., 2019) is a reading comprehension dataset used to assess LLMs’ abilities in discrete reasoning. 1061 ReClor (Yu et al., 2020) focuses on evaluating LLMs’ abilities in logical reasoning. MME-CoT 1062 (Jiang et al., 2025) is a benchmark specifically designed to evaluate CoT reasoning performance 1063 in large multimodal models (LMMs). REVEAL (Greyling, 2024) is a benchmark for verifying the 1064 correctness of CoT reasoning chains.

1065 C DETAILS CONCERNING CONSTRUCTION PHASE

1066 As described in the main text, in our framework, users only need to prepare a single configuration 1067 file to complete the relevant setup. The structure and content of this configuration file are detailed 1068 below.

1069 The configuration file primarily consists of three parts: basic Information, decision rules, and reasoning 1070 rules.

1073 **Basic Information.** This section covers general settings related to the underlying LLM, specifically 1074 including:

- 1076 • **LLM Base Model Information:** Specifies the LLM API or the locally deployed LLM 1077 model to be used. The corresponding JSON object is as follows:

```
1 "llm_base_model": {  
2     "model_type": "local",
```

```

1080
1081      3     "model_config_file_path": "/model/Llama-3-8B-
1082      4     Instruct/config.py"
1083

```

1084 Here, `model_type` constrains whether the model used is a local model or an API call.
 1085 `local` represents the use of a local model, and `api` represents using an online LLM API.
 1086 The `model_config_file_path` points to the LLM configuration file.

- 1087 • **Maximum Token Count:** Sets the maximum number of tokens available for task execution. The corresponding JSON object is as follows:

```

1090
1091      1     "token_limits": {
1092      2     "max_token_count": 4096
1093      3     }

```

1094 `max_token_count` refers to the maximum number of tokens allowed; exceeding this
 1095 count will terminate the inference process.

- 1096 • **Maximum Node Count:** Limits the maximum number of nodes allowed during the task
 1097 execution process. The corresponding JSON object is as follows:

```

1098
1099      1     "structure_limits": {
1100      2     "max_node_count": 500
1101      3     }

```

1102 Here, `max_node_count` refers to the maximum number of nodes allowed; exceeding this
 1103 count will also terminate the inference process.

1104 **Decision Rule.** This rule is used to determine how to decide the next operation given a specific
 1105 node, specifically including:

- 1106 • **Stop Judgment:** Defines the conditions under which the LLM's reasoning process should
 1107 be terminated. The corresponding JSON object is as follows:

```

1111
1112      1     "stop_judgment": {
1113      2     "condition": "Based on the information from the
1114      3     current node or the parent node, it is no longer
1115      4     possible to derive a calculation result of Game of 24,
1116      5     or the reasoning has entered a cycle."
1117      6     }

```

- 1118 • **Answer Judgment:** Specifies the conditions under which the content of the current node
 1119 can be output as the final answer. The corresponding JSON object is as follows:

```

1120
1121      1     "answer_judgment": {
1122      2     "condition": "The current method can calculate and
1123      3     obtain 24."
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

```

1125 **Reasoning Rules.** Define the generation and connection rules to be followed if a node requires
 1126 further expansion in reasoning. Each rule specifically includes:

- 1127 • **Topological Judgment:** Includes the topological conditions that a node must satisfy for
 1128 this rule to apply, such as the distance to the root node, node in-degree, the required sub-
 1129 graph structure, and a corresponding textual description. The corresponding JSON object
 1130 is as follows:

```

1131
1132      1     "topological_judgment": {
1133      2     "distance_from_root": [1, 3],
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2958
2959
```

```

1134     3     "allowed_in_degree": [1,3],
1135     4     "allowed_out_degree": [0,3],
1136     5     "required_subgraph_structure": "It needs to have
1137     6     descendant nodes with a distance greater than 2."
1138 }
```

1139
1140 max_distance_from_root constrains the range of the distance from the current node to
1141 the root node. [1, 3] represents the nodes that satisfy the current rule, where the distance
1142 to the root node (i.e., the shortest path to the root node) is between 1 and 3. If no such con-
1143 straint is applied, the value is [-1, -1]. allowed_in_degree represents the allowed
1144 in-degree of the current node. Similarly, if there is no restriction, the value is [-1, -1].
1145 allowed_out_degree follows the same logic. required_subgraph_structure
1146 represents the textual description of other possible graph structures required, which will be
1147 evaluated using the LLM for graph structure determination. If there is no restriction, the
1148 value is N/A.

- **Semantic Judgment:** Describes in textual form the semantic features that an applicable node should possess. The corresponding JSON object is as follows:

```

1     "semantic_judgment": {
2         "description": "Contain information about AI
3         technology."
4     }
```

1155
1156 The description provides the semantic features that are used to determine whether this
1157 rule should be applied.

- **G_{sub} Structure:** Specifies the composition of the generated set of child nodes $Ch(v)$ and its related set of parent nodes $Pa(Ch(v))$. If multiple nodes are generated using a single round of LLM interaction, the system will first generate one node. This node is then split based on predefined separation identifiers, and the associations between nodes are determined with the aid of the LLM to construct the edge structure. The corresponding JSON object is as follows:

```

1     "G_sub_structure": {
2         "child_nodes": {
3             "strategy": "single_round_multi_node",
4             "num_of_child_nodes": 1
5         },
6         "parent_nodes": {
7             "shortest_path_to_root": "include",
8             "sibling_node": "include",
9             "search_prompt": "N/A"
10        }
11    }
```

1174 In child_nodes, the value of strategy can be either single_round_multi_node
1175 or single_round_single_node, which indicates whether a single round of LLM in-
1176 teraction generates all child nodes or just a single child node. num_of_child_nodes
1177 defines the number of child nodes to be generated. If single_round_multi_node is
1178 selected, this value remains 1, and the framework itself will handle the splitting of child
1179 nodes in subsequent steps.

1180 In parent_nodes, shortest_path_to_root represents all the nodes in the shortest
1181 path from the current node to the root node. sibling_node indicates whether sibling
1182 nodes are included, with options include and exclude. If search_prompt is N/A,
1183 no search is conducted; otherwise, the parent nodes are searched according to the content
1184 of search_prompt.

- **Reasoning Prompt:** Provides the specific prompt text used.

1185
1186 Note that multiple reasoning rules will be defined, and for any given node requiring further reason-
1187 ing, exactly one rule applies.

1188
 1189 To provide a concrete example, we present the corresponding configuration file for the running
 1190 example in Section 3.4. The content of this configuration file is shown below:
 1191

```

1  {
2      "framework": "ToT for Game of 24",
3      "configuration": {
4          "basic_information": {
5              "llm_base_model": {
6                  "model_type": "local",
7                  "model_config_file_path": "/model/Llama-3-8B-Instruct/
8 config.py"
9              },
10             "token_limits": {
11                 "max_token_count": 4096
12             },
13             "structure_limits": {
14                 "max_node_count": 500
15             }
16         },
17         "decision_rules": {
18             "stop_judgment": {
19                 "condition": "Based on the information from the current
20 node or the parent node, it is no longer possible to derive a
21 calculation result of Game of 24, or the reasoning has entered
22 a cycle."
23             },
24             "answer_judgment": {
25                 "condition": "The current method can calculate and obtain
26 24."
27             }
28         },
29         "reasoning_rules": [
30             {
31                 "rule_id": "rule_for_root",
32                 "topological_judgment": {
33                     "max_distance_from_root": [0,0],
34                     "allowed_in_degree_range": [-1,-1],
35                     "required_subgraph_structure": "N/A"
36                 },
37                 "semantic_judgment": {
38                     "description": "N/A"
39                 },
40                 "G_sub_structure": {
41                     "child_nodes": {
42                         "strategy": "single_round_multi_node",
43                         "num_of_child_nodes": 1
44                     },
45                     "parent_nodes": {
46                         "shortest_path_to_root": "include",
47                         "sibling_node": "exclude",
48                         "search_prompt": "N/A"
49                     }
50                 },
51                 "reasoning_prompt": {
52                     "prompt_text": "You are a Game of 24 game expert. Please
53 solve the given problem, provide 3 different reasoning nodes.
54 Consider the most effective solving strategies. Note that
55 each approach should only advance one step, meaning only
56 compute one additional number. The current computed number
57 pool is: []. Add two numbers to this pool. Output only the
58 combinations. Follow the format: option: [], updated number
59 pool: []. Do not output any other content."
60                 }
61             }
62         }
63     }
64 }
```

```

1242      50  {
1243      51  "rule_id": "rule_backtrack",
1244      52  "topological_judgment": {
1245      53  "max_distance_from_root": [-1,-1],
1246      54  "allowed_in_degree_range": [-1,-1]],
1247      55  "required_subgraph_structure": "N/A"
1248      56  },
1249      57  "semantic_judgment": {
1250      58  "description": "N/A"
1251      59  },
1252      60  "G_sub_structure": {
1253      61  "child_nodes": {
1254      62  "strategy": "single_round_multi_node",
1255      63  "num_of_child_nodes": 1
1256      64  },
1257      65  "parent_nodes": {
1258      66  "shortest_path_to_root": "include",
1259      67  "sibling_node": "exclude",
1260      68  "search_prompt": "N/A"
1261      69  }
1262      70  },
1263      71  "reasoning_prompt": {
1264      72  "prompt_text": "You are a Game of 24 game expert. Please
1265      73  solve the given problem, provide 2 different reasoning nodes.
1266      74  Solve the problem from the parent node of the current node.
1267      75  Consider the most effective solving strategies. Note that each
1268      76  approach should only advance one step, meaning only compute
1269      77  one additional number. The current computed number pool is: [
1270      78  ]. Add two numbers to this pool. Output only the combinations.
1271      79  Follow the format: option: [ ], updated number pool: [ ]. Do
1272      80  not output any other content."
1273      81  }
1274      82  },
1275      83  {
1276      84  "rule_id": "rule_default",
1277      85  "topological_judgment": {
1278      86  "max_distance_from_root": [-1,-1],
1279      87  "allowed_in_degree_range": [-1,-1],
1280      88  "required_subgraph_structure": "N/A"
1281      89  },
1282      90  "semantic_judgment": {
1283      91  "semantic_description": "N/A"
1284      92  },
1285      93  "G_sub_structure": {
1286      94  "child_nodes": {
1287      95  "strategy": "single_round_multi_node",
1288      96  "num_of_child_nodes": 1
1289      97  },
1290      98  "parent_nodes": {
1291      99  "shortest_path_to_root": "include",
1292      "sibling_node": "exclude",
1293      "search_prompt": "N/A"
1294      },
1295      "reasoning_prompt": {
1296      "prompt_text": "You are a Game of 24 game expert. Please
1297      solve the given problem, provide 2 different reasoning nodes.
1298      Consider the most effective solving strategies. Note that
1299      each approach should only advance one step, meaning only
1299      compute one additional number. The current computed number
1299      pool is: [ ]. Add two numbers to this pool. Output only the
1299      combinations. Follow the format: option: [ ], updated number
1299      pool: [ ]. Do not output any other content."
1299  }

```

1296

1297 100]
1298 101 }
1299 102 }

1300

1301 **D PROOF OF PROPOSITION 1**

1302

1303 **Proposition 1** Kernel matrix $\mathbf{K}_{RGWL}^{(h)}$ of $\mathbb{K}_{RGWL}^{(h)}(\cdot, \cdot)$ is positive semi-definite (p.s.d.).

1304

1305 *Proof.* According to the proposition, for $\mathbf{c} \in \mathbb{R}^n$, we have:

1306

$$\begin{aligned}
 \mathbf{c}^\top \mathbf{K}_{RGWL}^{(h)} \mathbf{c} &= \sum_{i=1}^M \sum_{j=1}^M c_i c_j \mathbb{K}_{RGWL}^{(h)}(G_i, G_j) \\
 &= \sum_{i=1}^M \sum_{j=1}^M c_i c_j \left(\sum_{i=1}^h \left(\langle \eta(\tilde{\psi}^{(i)}(\rho(\tau(G_i)))) , \eta(\tilde{\psi}^{(j)}(\rho(\tau(G_j)))) \rangle \right. \right. \\
 &\quad \left. \left. + \lambda_i \lambda_j \langle \eta(\tilde{\psi}^{(i)}(\rho(G_i))) , \eta(\tilde{\psi}^{(j)}(\rho(G_j))) \rangle \right) \right) \\
 &= \sum_{i=1}^M \sum_{j=1}^M c_i c_j \left(\sum_{i=1}^h \left(\langle \eta(\tilde{\psi}^{(i)}(\rho(\tau(G_i)))) , \eta(\tilde{\psi}^{(i)}(\rho(\tau(G_j)))) \rangle \right. \right. \\
 &\quad \left. \left. + \lambda_i \lambda_j \sum_{i=1}^h \langle \eta(\tilde{\psi}^{(i)}(\rho(G_i))) , \eta(\tilde{\psi}^{(i)}(\rho(G_j))) \rangle \right) \right), \tag{6}
 \end{aligned}$$

1318

1319 where λ_i denotes the very value of λ according to G_i and G_j . As $\rho(\cdot)$ only modify graph
 1320 node features into labels, $\langle \eta(\tilde{\psi}^{(i)}(\rho(\tau(G_i)))) , \eta(\tilde{\psi}^{(i)}(\rho(\tau(G_j)))) \rangle$ can be denoted as the
 1321 inner product of certain vector α_i and α_j , $\alpha_i \in \mathbb{N}_0^h$ and $\alpha_j \in \mathbb{N}_0^h$. Similarly, we denote
 1322 $\langle \eta(\tilde{\psi}^{(i)}(\rho(G_i))) , \eta(\tilde{\psi}^{(i)}(\rho(G_j))) \rangle$ as the inner product of β_i and β_j , $\beta_i \in \mathbb{N}_0^h$ and $\beta_j \in \mathbb{N}_0^h$.
 1323 Therefore, we have:

1324

$$\begin{aligned}
 \mathbf{c}^\top \mathbf{K}_{RGWL}^{(h)} \mathbf{c} &= \sum_{i=1}^M \sum_{j=1}^M c_i c_j \left(\langle \alpha_i, \alpha_j \rangle + \lambda_i \lambda_j \langle \beta_i, \beta_j \rangle \right) \\
 &= \sum_{i=1}^M \sum_{j=1}^M \left(c_i c_j \langle \alpha_i, \alpha_j \rangle \right) + \sum_{i=1}^M \sum_{j=1}^M \left(c_i c_j \lambda_i \lambda_j \langle \beta_i, \beta_j \rangle \right) \\
 &= \left\langle \sum_{i=1}^M c_i \alpha_i, \sum_{j=1}^M c_j \alpha_j \right\rangle + \left\langle \sum_{i=1}^M c_i \lambda_i \alpha_i, \sum_{j=1}^M c_j \lambda_j \alpha_j \right\rangle \\
 &= \left\| \sum_{i=1}^M c_i \alpha_i \right\|^2 + \left\| \sum_{i=1}^M c_i \lambda_i \alpha_i \right\|^2 \geq 0. \tag{7}
 \end{aligned}$$

1340

Based on the definition of positive semidefinite matrices, the proposition is proven. \square

1341

1342

E DETIAIL IMPLEMENTATION OF $\tau(\cdot)$ AND $\rho(\cdot)$

1343

1344

1345

1346

1347

1348

1349

The function $\tau(\cdot)$ can be simply identified and obtained from the paths in the graph. Specifically, we use a graph search algorithm to find all paths from the root to the answer node, and then we save all the nodes along these paths, along with the corresponding inference content.

$\rho(\cdot)$ characterizes the KNN-based graph node labeling. Initially, a large language model is employed to describe all relevant ideas in a consistent format. Subsequently, the textual responses associated with each node are embedded into feature vectors using a language model (Reimers & Gurevych,

2019). Following this, the Elbow Method (Cui et al., 2020) is applied to cluster the feature vector sets. The clustering process terminates once the Sum of Squared Errors (SSE) falls below a predefined threshold, denoted as δ , which is treated as a hyperparameter. Nodes within the same cluster are then assigned the same label. Algorithm 2 demonstrates the procedure formally.

Algorithm 2 KNN-based Graph Node Labeling with Elbow Method for Clustering

- 1: **Input:** Graph G and G' with nodes, large language model LM , predefined threshold δ , feature embedding language model $f^{LM}(\cdot)$.
- 2: **Output:** Node labels
- 3: **1.** Use the large language model to describe all relevant ideas associated with each node v within G and G' in a consistent format.
- 4: **2.** Embed the textual responses of each node v into feature vectors using $f^{LM}(\cdot)$.
- 5: **3.** Apply the Elbow Method to cluster the feature vector sets:
- 6: For each $k = 1, 2, \dots$, number of nodes:
- 7: Compute the Sum of Squared Errors (SSE) for each clustering result.
- 8: Terminate clustering when the SSE falls below the threshold δ .
- 9: **5.** Assign the same label to all nodes within the same cluster.
- 10: **6.** Return the node labels.

F MORE REASONING FRAMEWORK IMPLEMENTATIONS AND WORKFLOWS

In this section, we provide more reasoning examples to better demonstrate TiG. We first present the configuration file and reasoning process used with the ToT framework on Legal Cases. The configuration file is as follows:

```
1  {
2      "framework": "ToT for Legal Case",
3      "configuration": {
4          "basic_information": {
5              "llm_base_model": {
6                  "model_type": "local",
7                  "model_config_file_path": "/model/gpt-4o/config.py"
8              },
9              "token_limits": {
10                  "max_token_count": 4096
11              },
12              "structure_limits": {
13                  "max_node_count": 500
14              }
15          },
16
17          "decision_rules": {
18              "stop_judgment": {
19                  "condition": "Stop if the current node (or its parent) can
20                  no longer advance the theft vs. fraud distinction, the amount
21                  attribution is already fixed for the active branch, the path
22                  enters a loop (semantic or structural repetition), or the node
23                  repeats previously concluded legal reasoning without adding
24                  new statutory or factual analysis."
25              },
26              "answer_judgment": {
27                  "condition": "The path reaches a final legal conclusion,
28                  the reasoning explicitly distinguishes the secret
29                  appropriation phase from the later deceitful transfer, and
30                  clarifies amount attribution for each offense."
31              }
32          },
33
34          "reasoning_rules": [
35      }
```

```

1404 28      {
1405 29          "rule_id": "rule_for_root",
1406 30          "topological_judgment": {
1407 31              "max_distance_from_root": [0, 0],
1408 32              "allowed_in_degree_range": [0, 0],
1409 33              "allowed_out_degree_range": [1, 3],
1410 34              "required_subgraph_structure": "N/A"
1411 35      },
1412 36          "semantic_judgment": {
1413 37              "description": "Applies only at the root question node
of this case."
1414 38      },
1415 39          "G_sub_structure": {
1416 40              "child_nodes": {
1417 41                  "strategy": "single_round_multi_node",
1418 42                  "num_of_child_nodes": 1
1419 43          },
1420 44              "parent_nodes": {
1421 45                  "shortest_path_to_root": "include",
1422 46                  "sibling_node": "exclude",
1423 47                  "search_prompt": "N/A"
1424 48          }
1425 49      },
1426 50          "reasoning_prompt": {
1427 51              "prompt_text": "You are a legal expert in Chinese
Criminal Law. Instruction: do not assume fixed statute numbers
. Instead, identify and retrieve the most relevant provisions
of the Criminal Law by analyzing the conduct. Output strictly
in the following format:\n reasoning: (one-sentence
preliminary conclusion)."
1428 52      }
1429 53  },
1430 54  },
1431 55  {
1432 56      "rule_id": "rule_theft_analysis",
1433 57      "topological_judgment": {
1434 58          "max_distance_from_root": [1, 2],
1435 59          "allowed_in_degree_range": [1, 3],
1436 60          "allowed_out_degree_range": [0, 3],
1437 61          "required_subgraph_structure": "The path to root
includes a node that flagged D1 (theft analysis) as an active
direction."
1438 62      },
1439 63          "semantic_judgment": {
1440 64              "description": "Node focuses on secret taking via
impersonation; evaluates whether conduct fits theft-related
statutory elements (secret appropriation of another's property
)."
1441 65      },
1442 66          "G_sub_structure": {
1443 67              "child_nodes": {
1444 68                  "strategy": "single_round_multi_node",
1445 69                  "num_of_child_nodes": 1
1446 70          },
1447 71              "parent_nodes": {
1448 72                  "shortest_path_to_root": "include",
1449 73                  "sibling_node": "exclude",
1450 74                  "search_prompt": "exclude"
1451 75          }
1452 76      },
1453 77          "reasoning_prompt": {
1454 78              "prompt_text": "Advance ONE step on theft analysis for
the secret impersonation/loan-obtaining phase. Identify
relevant statutory elements without assuming specific article
numbers. Map facts to elements: (i) secrecy, (ii)

```

```

1458
1459 appropriation, (iii) object = other's property, (iv) intent to
1460 unlawfully possess. Then produce sub-conclusions depending on
1461 money ownership at the moment of appropriation (bank vs
1462 victim). Output strictly: \n reasoning: (element-wise mapping
1463 in one sentence).\n (No extra text.)"
1464     }
1465     },
1466     {
1467         "rule_id": "rule_fraud_analysis",
1468         "topological_judgment": {
1469             "max_distance_from_root": [1, 2],
1470             "allowed_in_degree_range": [1, 3],
1471             "allowed_out_degree_range": [0, 3],
1472             "required_subgraph_structure": "The path to root
1473 includes a node that flagged D2 (fraud analysis) as an active
1474 direction."
1475         },
1476         "semantic_judgment": {
1477             "description": "Node focuses on deceit-induced transfer;
1478 evaluates whether conduct fits fraud-related statutory
1479 elements (obtaining property by deception)."
1480         },
1481         "G_sub_structure": {
1482             "child_nodes": {
1483                 "strategy": "single_round_multi_node",
1484                 "num_of_child_nodes": 1
1485             },
1486             "parent_nodes": {
1487                 "shortest_path_to_root": "include",
1488                 "sibling_node": "exclude",
1489                 "search_prompt": "exclude"
1490             },
1491             "reasoning_prompt": {
1492                 "prompt_text": "Advance ONE step on fraud analysis for
1493 the later inducement/transfer phase. Identify relevant
1494 statutory elements without assuming specific article numbers.
1495 Map facts to elements: (i) false representation/concealment, (ii)
1496 victim's disposal of property, (iii) causal link, (iv)
1497 unlawful possession. Then produce sub-conclusions depending on
1498 the victim's disposal awareness (fully misled vs partially
1499 aware). Output strictly:\n reasoning: (element-wise mapping in
1500 one sentence).\n (No extra text.)"
1501         }
1502     },
1503     {
1504         "rule_id": "rule_amount_attribution_and_concurrence",
1505         "topological_judgment": {
1506             "max_distance_from_root": [2, 4],
1507             "allowed_in_degree_range": [1, 5],
1508             "allowed_out_degree_range": [0, 3],
1509             "required_subgraph_structure": "The ancestor chain must
1510 already contain at least one theft-focused node and one fraud-
1511 focused node."
1512         },
1513         "semantic_judgment": {
1514             "description": "Node determines the ownership/
1515 attribution of the loan at each timepoint and whether theft
1516 and fraud should be punished cumulatively."
1517         },
1518         "G_sub_structure": {
1519             "child_nodes": {
1520                 "strategy": "single_round_single_node",
1521             }
1522         }

```

```

1512     123         "num_of_child_nodes": 1
1513     124     },
1514     125     "parent_nodes": {
1515     126         "shortest_path_to_root": "include",
1516     127         "sibling_node": "include",
1517     128         "search_prompt": "Summon the two closest analysis
1518 nodes (theft and fraud) on the shortest path for joint
1519 synthesis."
1520     129     }
1521     130     },
1522     131     "reasoning_prompt": {
1523     132         "prompt_text": "Synthesize the results of theft and
1524 fraud analyses. Decide (i) at impersonation moment the loaned
1525 money is owned by bank or victim, (ii) the later induced
1526 transfer disposes of <victim>'s property by deception, and (iii)
1527 whether the offenses concur and shall be combined for
1528 punishment. Output strictly:\n reasoning: (one-sentence
1529 rationale linking timepoints to ownership) \n (No extra text
1530 .)"
1531     133     }
1532     134     },
1533     135     {
1534     136         "rule_id": "rule_finalize",
1535     137         "topological_judgment": {
1536     138             "max_distance_from_root": [2, 6],
1537     139             "allowed_in_degree_range": [1, 10],
1538     140             "allowed_out_degree_range": [0, 1],
1539     141             "required_subgraph_structure": "Upstream nodes already
1540     142 fixed both: (i) theft elements satisfied for impersonation
1541     143 stage; (ii) fraud elements satisfied for inducement stage; and
1542     144 amount attribution coherent."
1543     145     },
1544     146     "semantic_judgment": {
1545     147         "description": "All material elements established."
1546     148     },
1547     149     "G_sub_structure": {
1548     150         "child_nodes": {
1549     151             "strategy": "single_round_single_node",
1550     152             "num_of_child_nodes": 1
1551     153         },
1552     154         "parent_nodes": {
1553     155             "shortest_path_to_root": "include",
1554     156             "sibling_node": "exclude",
1555     157             "search_prompt": "N/A"
1556     158         }
1557     159     },
1558     160     "reasoning_prompt": {
1559     161         "prompt_text": "Issue the final conclusion exactly as
1560     162 the higher court maintained: theft + fraud, combined
1561     163 punishment. Also state in one sentence the amount attribution
1562     164 logic (impersonation stage vs later induced transfer). Output
1563     165 strictly:\n final: (theft + fraud; sentences combined True)\n
rationale: (one-sentence amount-attribution and concurrence
explanation)\n (No extra text.)"
1564     166     }
1565     167   ]
1566     168 }
1567     169 }
1568     170 }
1569     171 }
1570     172 }
1571     173 }
1572     174 }
1573     175 }
1574     176 }
1575     177 }
1576     178 }
1577     179 }
1578     180 }
1579     181 }
1580     182 }
1581     183 }
1582     184 }
1583     185 }
1584     186 }
1585     187 }
1586     188 }
1587     189 }
1588     190 }
1589     191 }
1590     192 }
1591     193 }
1592     194 }
1593     195 }
1594     196 }
1595     197 }
1596     198 }
1597     199 }
1598     200 }
1599     201 }
1600     202 }
1601     203 }
1602     204 }
1603     205 }
1604     206 }
1605     207 }
1606     208 }
1607     209 }
1608     210 }
1609     211 }
1610     212 }
1611     213 }
1612     214 }
1613     215 }
1614     216 }
1615     217 }
1616     218 }
1617     219 }
1618     220 }
1619     221 }
1620     222 }
1621     223 }
1622     224 }
1623     225 }
1624     226 }
1625     227 }
1626     228 }
1627     229 }
1628     230 }
1629     231 }
1630     232 }
1631     233 }
1632     234 }
1633     235 }
1634     236 }
1635     237 }
1636     238 }
1637     239 }
1638     240 }
1639     241 }
1640     242 }
1641     243 }
1642     244 }
1643     245 }
1644     246 }
1645     247 }
1646     248 }
1647     249 }
1648     250 }
1649     251 }
1650     252 }
1651     253 }
1652     254 }
1653     255 }
1654     256 }
1655     257 }
1656     258 }
1657     259 }
1658     260 }
1659     261 }
1660     262 }
1661     263 }
1662     264 }
1663     265 }
1664     266 }
1665     267 }
1666     268 }
1667     269 }
1668     270 }
1669     271 }
1670     272 }
1671     273 }
1672     274 }
1673     275 }
1674     276 }
1675     277 }
1676     278 }
1677     279 }
1678     280 }
1679     281 }
1680     282 }
1681     283 }
1682     284 }
1683     285 }
1684     286 }
1685     287 }
1686     288 }
1687     289 }
1688     290 }
1689     291 }
1690     292 }
1691     293 }
1692     294 }
1693     295 }
1694     296 }
1695     297 }
1696     298 }
1697     299 }
1698     300 }
1699     301 }
1700     302 }
1701     303 }
1702     304 }
1703     305 }
1704     306 }
1705     307 }
1706     308 }
1707     309 }
1708     310 }
1709     311 }
1710     312 }
1711     313 }
1712     314 }
1713     315 }
1714     316 }
1715     317 }
1716     318 }
1717     319 }
1718     320 }
1719     321 }
1720     322 }
1721     323 }
1722     324 }
1723     325 }
1724     326 }
1725     327 }
1726     328 }
1727     329 }
1728     330 }
1729     331 }
1730     332 }
1731     333 }
1732     334 }
1733     335 }
1734     336 }
1735     337 }
1736     338 }
1737     339 }
1738     340 }
1739     341 }
1740     342 }
1741     343 }
1742     344 }
1743     345 }
1744     346 }
1745     347 }
1746     348 }
1747     349 }
1748     350 }
1749     351 }
1750     352 }
1751     353 }
1752     354 }
1753     355 }
1754     356 }
1755     357 }
1756     358 }
1757     359 }
1758     360 }
1759     361 }
1760     362 }
1761     363 }
1762     364 }
1763     365 }
1764     366 }
1765     367 }
1766     368 }
1767     369 }
1768     370 }
1769     371 }
1770     372 }
1771     373 }
1772     374 }
1773     375 }
1774     376 }
1775     377 }
1776     378 }
1777     379 }
1778     380 }
1779     381 }
1780     382 }
1781     383 }
1782     384 }
1783     385 }
1784     386 }
1785     387 }
1786     388 }
1787     389 }
1788     390 }
1789     391 }
1790     392 }
1791     393 }
1792     394 }
1793     395 }
1794     396 }
1795     397 }
1796     398 }
1797     399 }
1798     400 }
1799     401 }
1800     402 }
1801     403 }
1802     404 }
1803     405 }
1804     406 }
1805     407 }
1806     408 }
1807     409 }
1808     410 }
1809     411 }
1810     412 }
1811     413 }
1812     414 }
1813     415 }
1814     416 }
1815     417 }
1816     418 }
1817     419 }
1818     420 }
1819     421 }
1820     422 }
1821     423 }
1822     424 }
1823     425 }
1824     426 }
1825     427 }
1826     428 }
1827     429 }
1828     430 }
1829     431 }
1830     432 }
1831     433 }
1832     434 }
1833     435 }
1834     436 }
1835     437 }
1836     438 }
1837     439 }
1838     440 }
1839     441 }
1840     442 }
1841     443 }
1842     444 }
1843     445 }
1844     446 }
1845     447 }
1846     448 }
1847     449 }
1848     450 }
1849     451 }
1850     452 }
1851     453 }
1852     454 }
1853     455 }
1854     456 }
1855     457 }
1856     458 }
1857     459 }
1858     460 }
1859     461 }
1860     462 }
1861     463 }
1862     464 }
1863     465 }
1864     466 }
1865     467 }
1866     468 }
1867     469 }
1868     470 }
1869     471 }
1870     472 }
1871     473 }
1872     474 }
1873     475 }
1874     476 }
1875     477 }
1876     478 }
1877     479 }
1878     480 }
1879     481 }
1880     482 }
1881     483 }
1882     484 }
1883     485 }
1884     486 }
1885     487 }
1886     488 }
1887     489 }
1888     490 }
1889     491 }
1890     492 }
1891     493 }
1892     494 }
1893     495 }
1894     496 }
1895     497 }
1896     498 }
1897     499 }
1898     500 }
1899     501 }
1900     502 }
1901     503 }
1902     504 }
1903     505 }
1904     506 }
1905     507 }
1906     508 }
1907     509 }
1908     510 }
1909     511 }
1910     512 }
1911     513 }
1912     514 }
1913     515 }
1914     516 }
1915     517 }
1916     518 }
1917     519 }
1918     520 }
1919     521 }
1920     522 }
1921     523 }
1922     524 }
1923     525 }
1924     526 }
1925     527 }
1926     528 }
1927     529 }
1928     530 }
1929     531 }
1930     532 }
1931     533 }
1932     534 }
1933     535 }
1934     536 }
1935     537 }
1936     538 }
1937     539 }
1938     540 }
1939     541 }
1940     542 }
1941     543 }
1942     544 }
1943     545 }
1944     546 }
1945     547 }
1946     548 }
1947     549 }
1948     550 }
1949     551 }
1950     552 }
1951     553 }
1952     554 }
1953     555 }
1954     556 }
1955     557 }
1956     558 }
1957     559 }
1958     560 }
1959     561 }
1960     562 }
1961     563 }
1962     564 }
1963     565 }
1964     566 }
1965     567 }
1966     568 }
1967     569 }
1968     570 }
1969     571 }
1970     572 }
1971     573 }
1972     574 }
1973     575 }
1974     576 }
1975     577 }
1976     578 }
1977     579 }
1978     580 }
1979     581 }
1980     582 }
1981     583 }
1982     584 }
1983     585 }
1984     586 }
1985     587 }
1986     588 }
1987     589 }
1988     590 }
1989     591 }
1990     592 }
1991     593 }
1992     594 }
1993     595 }
1994     596 }
1995     597 }
1996     598 }
1997     599 }
1998     600 }
1999     601 }
2000     602 }
2001     603 }
2002     604 }
2003     605 }
2004     606 }
2005     607 }
2006     608 }
2007     609 }
2008     610 }
2009     611 }
2010     612 }
2011     613 }
2012     614 }
2013     615 }
2014     616 }
2015     617 }
2016     618 }
2017     619 }
2018     620 }
2019     621 }
2020     622 }
2021     623 }
2022     624 }
2023     625 }
2024     626 }
2025     627 }
2026     628 }
2027     629 }
2028     630 }
2029     631 }
2030     632 }
2031     633 }
2032     634 }
2033     635 }
2034     636 }
2035     637 }
2036     638 }
2037     639 }
2038     640 }
2039     641 }
2040     642 }
2041     643 }
2042     644 }
2043     645 }
2044     646 }
2045     647 }
2046     648 }
2047     649 }
2048     650 }
2049     651 }
2050     652 }
2051     653 }
2052     654 }
2053     655 }
2054     656 }
2055     657 }
2056     658 }
2057     659 }
2058     660 }
2059     661 }
2060     662 }
2061     663 }
2062     664 }
2063     665 }
2064     666 }
2065     667 }
2066     668 }
2067     669 }
2068     670 }
2069     671 }
2070     672 }
2071     673 }
2072     674 }
2073     675 }
2074     676 }
2075     677 }
2076     678 }
2077     679 }
2078     680 }
2079     681 }
2080     682 }
2081     683 }
2082     684 }
2083     685 }
2084     686 }
2085     687 }
2086     688 }
2087     689 }
2088     690 }
2089     691 }
2090     692 }
2091     693 }
2092     694 }
2093     695 }
2094     696 }
2095     697 }
2096     698 }
2097     699 }
2098     700 }
2099     701 }
2100     702 }
2101     703 }
2102     704 }
2103     705 }
2104     706 }
2105     707 }
2106     708 }
2107     709 }
2108     710 }
2109     711 }
2110     712 }
2111     713 }
2112     714 }
2113     715 }
2114     716 }
2115     717 }
2116     718 }
2117     719 }
2118     720 }
2119     721 }
2120     722 }
2121     723 }
2122     724 }
2123     725 }
2124     726 }
2125     727 }
2126     728 }
2127     729 }
2128     730 }
2129     731 }
2130     732 }
2131     733 }
2132     734 }
2133     735 }
2134     736 }
2135     737 }
2136     738 }
2137     739 }
2138     740 }
2139     741 }
2140     742 }
2141     743 }
2142     744 }
2143     745 }
2144     746 }
2145     747 }
2146     748 }
2147     749 }
2148     750 }
2149     751 }
2150     752 }
2151     753 }
2152     754 }
2153     755 }
2154     756 }
2155     757 }
2156     758 }
2157     759 }
2158     760 }
2159     761 }
2160     762 }
2161     763 }
2162     764 }
2163     765 }
2164     766 }
2165     767 }
2166     768 }
2167     769 }
2168     770 }
2169     771 }
2170     772 }
2171     773 }
2172     774 }
2173     775 }
2174     776 }
2175     777 }
2176     778 }
2177     779 }
2178     780 }
2179     781 }
2180     782 }
2181     783 }
2182     784 }
2183     785 }
2184     786 }
2185     787 }
2186     788 }
2187     789 }
2188     790 }
2189     791 }
2190     792 }
2191     793 }
2192     794 }
2193     795 }
2194     796 }
2195     797 }
2196     798 }
2197     799 }
2198     800 }
2199     801 }
2200     802 }
2201     803 }
2202     804 }
2203     805 }
2204     806 }
2205     807 }
2206     808 }
2207     809 }
2208     810 }
2209     811 }
2210     812 }
2211     813 }
2212     814 }
2213     815 }
2214     816 }
2215     817 }
2216     818 }
2217     819 }
2218     820 }
2219     821 }
2220     822 }
2221     823 }
2222     824 }
2223     825 }
2224     826 }
2225     827 }
2226     828 }
2227     829 }
2228     830 }
2229     831 }
2230     832 }
2231     833 }
2232     834 }
2233     835 }
2234     836 }
2235     837 }
2236     838 }
2237     839 }
2238     840 }
2239     841 }
2240     842 }
2241     843 }
2242     844 }
2243     845 }
2244     846 }
2245     847 }
2246     848 }
2247     849 }
2248     850 }
2249     851 }
2250     852 }
2251     853 }
2252     854 }
2253     855 }
2254     856 }
2255     857 }
2256     858 }
2257     859 }
2258     860 }
2259     861 }
2260     862 }
2261     863 }
2262     864 }
2263     865 }
2264     866 }
2265     867 }
2266     868 }
2267     869 }
2268     870 }
2269     871 }
2270     872 }
2271     873 }
2272     874 }
2273     875 }
2274     876 }
2275     877 }
2276     878 }
2277     879 }
2278     880 }
2279     881 }
2280     882 }
2281     883 }
2282     884 }
2283     885 }
2284     886 }
2285     887 }
2286     888 }
2287     889 }
2288     890 }
2289     891 }
2290     892 }
2291     893 }
2292     894 }
2293     895 }
2294     896 }
2295     897 }
2296     898 }
2297     899 }
2298     900 }
2299     901 }
2300     902 }
2301     903 }
2302     904 }
2303     905 }
2304     906 }
2305     907 }
2306     908 }
2307     909 }
2308     910 }
2309     911 }
2310     912 }
2311     913 }
2312     914 }
2313     915 }
2314     916 }
2315     917 }
2316     918 }
2317     919 }
2318     920 }
2319     921 }
2320     922 }
2321     923 }
2322     924 }
2323     925 }
2324     926 }
2325     927 }
2326     928 }
2327     929 }
2328     930 }
2329     931 }
2330     932 }
2331     933 }
2332     934 }
2333     935 }
2334     936 }
2335     937 }
2336     938 }
2337     939 }
2338     940 }
2339     941 }
2340     942 }
2341     943 }
2342     944 }
2343     945 }
2344     946 }
2345     947 }
2346     948 }
2347     949 }
2348     950 }
2349     951 }
2350     952 }
2351     953 }
2352     954 }
2353     955 }
2354     956 }
2355     957 }
2356     958 }
2357     959 }
2358     960 }
2359     961 }
2360     962 }
2361     963 }
2362     964 }
2363     965 }
2364     966 }
2365     967 }
2366     968 }
2367     969 }
2368     970 }
2369     971 }
2370     972 }
2371     973 }
2372     974 }
2373     975 }
2374     976 }
2375     977 }
2376     978 }
2377     979 }
2378     980 }
2379     981 }
2380     982 }
2381     983 }
2382     984 }
2383     985 }
2384     986 }
2385     987 }
2386     988 }
2387     989 }
2388     990 }
2389     991 }
2390     992 }
2391     993 }
2392     994 }
2393     995 }
2394     996 }
2395     997 }
2396     998 }
2397     999 }
2398     1000 }
2399     1001 }
2400     1002 }
2401     1003 }
2402     1004 }
2403     1005 }
2404     1006 }
2405     1007 }
2406     1008 }
2407     1009 }
2408     1010 }
2409     1011 }
2410     1012 }
2411     1013 }
2412     1014 }
2413     1015 }
2414     1016 }
2415     1017 }
2416     1018 }
2417     1019 }
2418     1020 }
2419     1021 }
2420     1022 }
2421     1023 }
2422     1024 }
2423     1025 }
2424     1026 }
2425     1027 }
2426     1028 }
2427     1029 }
2428     1030 }
2429     1031 }
2430     1032 }
2431     1033 }
2432     1034 }
2433     1035 }
2434     1036 }
2435     1037 }
2436     1038 }
2437     1039 }
2438     1040 }
2439     1041 }
2440     1042 }
2441     1043 }
2442     1044 }
2443     1045 }
2444     1046 }
2445     1047 }
2446     1048 }
2447     1049 }
2448     1050 }
2449     1051 }
2450     1052 }
2451     1053 }
2452     1054 }
2453     1055 }
2454     1056 }
2455     1057 }
2456     1058 }
2457     1059 }
2458     1060 }
2459     1061 }
2460     1062 }
2461     1063 }
2462     1064 }
2463     1065 }
2464     1066 }
2465     1067 }
2466     1068 }
2467     1069 }
2468     1070 }
2469     1071 }
2470     1072 }
2471     1073 }
2472     1074 }
2473     1075 }
2474     1076 }
2475     1077 }
2476     1078 }
2477     1079 }
2478     1080 }
2479     1081 }
2480     1082 }
2481     1083 }
2482     1084 }
2483     1085 }
2484     1086 }
2485     1087 }
2486     1088 }
2487     1089 }
2488     1090 }
2489     1091 }
2490     1092 }
2491     1093 }
2492     1094 }
2493     1095 }
2494     1096 }
2495     1097 }
2496     1098 }
2497     1099 }
2498     1100 }
2499     1101 }
2500     1102 }
2501     1103 }
2502     1104 }
2503     1105 }
2504     1106 }
2505     1107 }
2506     1108 }
2507     1109 }
2508     1110 }
2509     1111 }
2510     1112 }
2511     1113 }
2512     1114 }
2513     1115 }
2514     1116 }
2515     1117 }
2516     1118 }
2517     1119 }
2518     1120 }
2519     1121 }
2520     1122 }
2521     1123 }
2522     1124 }
2523     1125 }
2524     1126 }
2525     1127 }
2526     1128 }
2527     1129 }
2528     1130 }
2529     1131 }
2530     1132 }
2531     1133 }
2532     1134 }
2533     1135 }
2534     1136 }
2535     1137 }
2536     1138 }
2537     1139 }
2538     1140 }
2539     1141 }
2540     1142 }
2541     1143 }
2542     1144 }
2543     1145 }
2544     1146 }
2545     1147 }
2546     1148 }
2547     1149 }
2548     1150 }
2549     1151 }
2550     1152 }
2551     1153 }
2552     1154 }
2553     1155 }
2554     1156 }
2555     1157 }
2556     1158 }
2557     1159 }
2558     1160 }
2559     1161 }
2560     1162 }
2561     1163 }
2562     1164 }
2563     1165 }
2564     1166 }
2565     1167 }
2566     1168 }
2567     1169 }
2568     1170 }
2569     1171 }
2570     1172 }
2571     1173 }
2572     1174 }
2573     1175 }
2574     1176 }
2575     1177 }
2576     1178 }
2577     1179 }
2578     1180 }
2579     1181 }
2580     1182 }
2581     1183 }
2582     1184 }
2583     1185 }
2584     1186 }
2585     1187 }
2586     
```

1566
 1567 Next, we present the specific outputs of the model, including textual descriptions of the reasoning
 1568 graphs and the content of each reasoning node.
 1569

1570 **Example Output**

1571 **Graph structure:**

1573 Nodes:
 1574 [t=0] :root 0
 1575 [t=1] :node 1 node 2 node 3
 1576 [t=2] :node 4 node 5 node 6 node 7 node 8 node 9
 1577 [t=3] :node 10 node 11 node 12 node 13

1578 Edges:
 1579 root 0 → node 1
 1580 root 0 → node 2
 1581 root 0 → node 3
 1582 node 1 → node 4
 1583 node 1 → node 5
 1584 node 2 → node 6
 1585 node 2 → node 7
 1586 node 3 → node 8
 1587 node 3 → node 9
 1588 node 5 → node 10
 1589 node 5 → node 11
 1590 node 7 → node 12
 1591 node 9 → node 13

1592 Terminated nodes:
 1593 node 4, node 6, node 8, node 10, node 11, node 12
 1594

1595 Answer node:
 1596 node 13

1598 Answer path:
 1599 root 0 → node 3 → node 9 → node 13

1601 **Reasoning Procedure:**

1602 [t=1] Nodes:

1603 node 1:

1604 reasoning: Secret impersonation of the victim at the loan stage may constitute theft-related
 1605 statutory elements. next_check: Ownership of loan funds (bank or victim).

1606 Rule applied: rule_for_root.

1607 Why proceed: This branch explores theft-related analysis.

1608 node 2:

1609 reasoning: Inducing the victim to transfer funds later may constitute fraud-related statutory
 1610 elements.

1611 next_check: Victim's awareness and property disposal.

1612 Rule applied: rule_for_root.

1613 Why proceed: This branch explores fraud-related analysis.

1614 node 3:

1615 reasoning: Determination of ownership of loan funds is crucial for deciding theft vs. fraud
 1616 and concurrence.

```

1620
1621 next_check: Attribution of property and concurrence rules. Rule applied: rule_for_root.
1622 Why proceed: This branch explores amount attribution and concurrence.
1623
1624 [t=2] Nodes:
1625
1626 node 4 (terminated False):
1627 reasoning: If loan funds are considered victim-owned, appropriation = theft True element
1628 mapping.
1629 branch: victim-owned
1630 next_check: N/A
1631 Rule applied: rule_theft_analysis.
1632 Termination reason: No further statutory issues to check once victim-ownership path is ex-
1633 hausted. Marked False (terminated).
1634
1635 node 5:
1636 reasoning: If loan funds are considered bank-owned, appropriation analysis continues.
1637 branch: bank-owned
1638 next_check: Further confirmation of ownership and appropriation.
1639 Rule applied: rule_theft_analysis.
1640 Why proceed: Ownership requires deeper analysis, not terminated.
1641
1642 node 6 (terminated False):
1643 reasoning: If the victim is fully deceived and disposes of property, fraud-related elements are
1644 satisfied. branch: fully-misled next_check: N/A Rule applied: rule_fraud_analysis. Termina-
1645 tion reason: Complete satisfaction of fraud elements. Marked False.
1646
1647 node 7:
1648 reasoning: If the victim is partially aware but still induced to transfer funds, fraud requires
1649 concurrence analysis.
1650 branch: partially-aware
1651 next_check: Offense concurrence and attribution.
1652 Rule applied: rule_fraud_analysis.
1653 Why proceed: Requires deeper concurrence analysis.
1654
1655 node 8 (terminated False):
1656 reasoning: If loan funds entirely belong to the victim, only theft is applicable.
1657 conclusion: single offense False
1658 next_check: N/A
1659 Rule applied: rule_amount_attribution_and_concurrence.
1660 Termination reason: This conclusion contradicts the higher court (single offense). Ter-
1661 minated False.
1662
1663 node 9:
1664 reasoning: If loan funds initially belong to the bank, later deceitful transfer is fraud, leading
1665 to combined punishment.
1666 conclusion: pending final check
1667 next_check: Final concurrence confirmation.
1668 Rule applied: rule_amount_attribution_and_concurrence (from config).
1669 Why proceed: This aligns with the higher court's reasoning, must continue.
1670
1671 [t=3] Nodes:
1672
1673 node 10 (terminated False):
1674 reasoning: Bank retains ownership; impersonation = theft.
1675 conclusion: theft established True
1676 next_check: N/A
1677 Rule applied: rule_theft_analysis.

```

```

1674
1675 Termination reason: Stopped at theft-only branch, contradicts combined judgment → False.
1676
1677 node 11 (terminated False):
1678 reasoning: Loan as debt only, not property; theft not established.
1679 conclusion: theft not False
1680 next_check: N/A
1681 Rule applied: rule_theft_analysis.
1682 Termination reason: Theft elements negated → branch ends False.
1683
1684 node 12 (terminated False):
1685 reasoning: Victim partially aware, inducement insufficient → fraud weakened.
1686 conclusion: fraud not False
1687 next_check: N/A
1688 Rule applied: rule_fraud_analysis (from config).
1689 Termination reason: Fraud elements not fully satisfied → branch ends False.
1690
1691 node 13 (True final answer):
1692 final: theft + fraud; sentences combined True
1693 rationale: At impersonation stage, appropriation = theft against bank's property; at inducement stage, transfer = fraud against victim's property; ownership attribution supports two offenses in concurrence.
1694 Rule applied: rule_finalize (from config).
1695 Why valid: Matches higher court judgment; distinguishes phases and attributes amounts correctly → True
1696
1697

```

1698 Next, we present an example using the GoT framework on College Entrance Examination Questions.
1699 The configuration file is as follows:

```

1700 {
1701     "framework": "GoT for College Entrance Examination Question",
1702     "configuration": {
1703         "basic_information": {
1704             "llm_base_model": {
1705                 "model_type": "local",
1706                 "model_config_file_path": "/model/gpt-4o/config.py"
1707             },
1708             "token_limits": {
1709                 "max_token_count": 4096
1710             },
1711             "structure_limits": {
1712                 "max_node_count": 500
1713             }
1714         },
1715         "decision_rules": {
1716             "stop_judgment": {
1717                 "condition": "Stop if the current node (or its parent) cannot further advance the mathematical reasoning, the branch yields contradictions, or all solution cases are exhausted."
1718             },
1719             "answer_judgment": {
1720                 "condition": "The path reaches a final conclusion consistent with the expected solution form (numeric range, interval, or multiple-choice)."
1721             }
1722         },
1723         "reasoning_rules": [
1724             {
1725                 "rule_id": "rule_for_root",
1726             }
1727         ]
1728     }
1729 }

```

```

1728      30      "topological_judgment": {
1729      31          "max_distance_from_root": [0, 0],
1730      32          "allowed_in_degree_range": [0, 0],
1731      33          "allowed_out_degree_range": [1, 3],
1732      34          "required_subgraph_structure": "N/A"
1733      35      },
1734      36      "semantic_judgment": {
1735      37          "description": "Root node: classifies the math problem
and generates initial reasoning directions."
1736      38      },
1737      39      "G_sub_structure": {
1738      40          "child_nodes": {
1739      41              "strategy": "single_round_multi_node",
1740      42              "num_of_child_nodes": 3
1741      43      },
1742      44      "parent_nodes": {
1743      45          "shortest_path_to_root": "include",
1744      46          "sibling_node": "exclude",
1745      47          "search_prompt": "N/A"
1746      48      },
1747      49      "reasoning_prompt": {
1748      50          "prompt_text": "You are a math expert. Generate high-
level reasoning direction advancing ONE step: Output strictly:
\n reasoning: (one-sentence preliminary step)."
1749      51      },
1750      52      },
1751      53  },
1752      54  },
1753      55  {
1754      56      "rule_id": "rule_equation_analysis",
1755      57      "topological_judgment": {
1756      58          "max_distance_from_root": [1, 3],
1757      59          "allowed_in_degree_range": [1, 3],
1758      60          "allowed_out_degree_range": [0, 3],
1759      61          "required_subgraph_structure": "The path to root
includes a node flagged D1."
1760      62      },
1761      63      "semantic_judgment": {
1762      64          "description": "Analyzes quadratic equations (
discriminant, factorization, or root conditions)."
1763      65      },
1764      66      "G_sub_structure": {
1765      67          "child_nodes": {
1766      68              "strategy": "single_round_multi_node",
1767      69              "num_of_child_nodes": 2
1768      70      },
1769      71          "parent_nodes": {
1770      72              "shortest_path_to_root": "include",
1771      73              "sibling_node": "exclude",
1772      74              "search_prompt": "Retrieve ancestor node introducing
equation analysis."
1773      75      },
1774      76      "reasoning_prompt": {
1775      77          "prompt_text": "Advance ONE step in equation/roots
analysis. Output strictly:\n reasoning: (root/condition
analysis in one sentence)."
1776      78      },
1777      79      },
1778      80  },
1779      81  },
1780      82  {
1781      83      "rule_id": "rule_set_operations",
1782      84      "topological_judgment": {
1783      85          "max_distance_from_root": [1, 3],

```

```

1782     87         "allowed_in_degree_range": [1, 3],
1783     88         "allowed_out_degree_range": [0, 3],
1784     89         "required_subgraph_structure": "The path to root
1785 includes a node flagged D2."
1786     90     },
1787     91     "semantic_judgment": {
1788     92         "description": "Analyzes subset and inclusion relations
between sets."
1789     93     },
1790     94     "G_sub_structure": {
1791     95         "child_nodes": {
1792     96             "strategy": "single_round_multi_node",
1793     97             "num_of_child_nodes": 2
1794     98         },
1795     99         "parent_nodes": {
1800    100             "shortest_path_to_root": "include",
1801    101             "sibling_node": "exclude",
1802    102             "search_prompt": "Retrieve ancestor node introducing
set operation analysis."
1803    103         }
1804    104     },
1805    105     "reasoning_prompt": {
1806    106         "prompt_text": "Advance ONE step in analysis. Output
strictly:\n reasoning: (analysis). "
1807    107     }
1808    108 },
1809    109 },
1810    110     {
1811    111         "rule_id": "rule_backtrack",
1812    112         "topological_judgment": {
1813    113             "max_distance_from_root": [-1, -1],
1814    114             "allowed_in_degree_range": [1, 10],
1815    115             "allowed_out_degree_range": [0, 2],
1816    116             "required_subgraph_structure": "N/A"
1817    117         },
1818    118         "semantic_judgment": {
1819    119             "description": "The branch stalls or contradicts."
1820    120         },
1821    121         "G_sub_structure": {
1822    122             "child_nodes": {
1823    123                 "strategy": "single_round_single_node",
1824    124                 "num_of_child_nodes": 1
1825    125             },
1826    126             "parent_nodes": {
1827    127                 "shortest_path_to_root": "include",
1828    128                 "sibling_node": "exclude",
1829    129                 "search_prompt": "Find the closest ancestor
introducing the conflicting assumption."
1830    130             }
1831    131         },
1832    132         "reasoning_prompt": {
1833    133             "prompt_text": "Backtrack ONE step to the nearest
ancestor with a wrong or incomplete assumption. Propose a
minimally revised condition. Output strictly:\n reasoning: ("
1834    134             "short_fix)."
1835    135         }
1836    136     },
1837    137     {
1838    138         "rule_id": "rule_finalize",
1839    139         "topological_judgment": {
1840    140             "max_distance_from_root": [2, 6],
1841    141             "allowed_in_degree_range": [1, 10],
1842    142             "allowed_out_degree_range": [0, 1],

```

```

1836    143      "required_subgraph_structure": "Upstream nodes already
1837    143      fixed both: (i) equation root conditions, (ii) set-inclusion
1838    143      conditions."
1839    144      },
1840    145      "semantic_judgment": {
1841    146      "description": "All relevant conditions are satisfied;
1842    147      issue the final solution."
1843    148      },
1844    149      "G_sub_structure": {
1845    150      "child_nodes": {
1846    151      "strategy": "single_round_single_node",
1847    152      "num_of_child_nodes": 1
1848    153      },
1849    154      "parent_nodes": {
1850    155      "shortest_path_to_root": "include",
1851    156      "sibling_node": "exclude",
1852    157      "search_prompt": "N/A"
1853    158      },
1854    159      "reasoning_prompt": {
1855    160      "prompt_text": "Issue the final answer strictly in the
1856    160      problem's required format. Output strictly:\n final: (answer)
1857    161      \n rationale: (one-sentence justification) \n (No extra text.)"
1858    162      }
1859    163      ],
1860    164      }
1861    165      }
1862    166  }

```

1863 Next, we present the specific outputs of the model, including textual descriptions of the reasoning
1864 graphs and the content of each reasoning node.

1865

1866

Example Output

1867

1868

Graph structure:

1869

1870

Nodes:

1871

[t=0] :root 0

1872

[t=1] :node 1 node 2 node 3

1873

[t=2] :node 4 node 5 node 6 node 7

1874

[t=3] :node 8 node 9 node 10

1875

Edges:

1876

root 0 → node 1

1877

root 0 → node 2

1878

root 0 → node 3

1879

node 2 → node 4

1880

node 2 → node 5

1881

node 3 → node 8

1882

node 8 → node 3 (backtracking)

1883

node 3 → node 9

1884

node 4 → node 8

1885

node 5 → node 9

1886

node 7 → node 10

1887

Terminated nodes:

1888

node 1, node 8, node 9

1889

```

1890
1891     Answer node:
1892     node 10
1893
1894     Answer path:
1895     root 0 → node 3 → node 7 → node 10
1896
1897     Reasoning Procedure:
1898
1899     node 1 (terminated False):
1900     reasoning: Solve roots of  $A$  and  $B$ ; check discriminant of  $B$ .
1901     next_check: Whether roots of  $B$  lie inside  $\{1, 2\}$ .
1902     Rule applied: rule_for_root.
1903     Termination reason: Incomplete, does not yet impose subset condition → False.
1904
1905     node 2:
1906     reasoning: For the condition  $A \cup B = A$ , require  $B \subseteq A$ .
1907     next_check: Compare roots of  $B$  with elements of  $A$ .
1908     Rule applied: rule_for_root.
1909     Why proceed: Core set-inclusion condition.
1910
1911     node 3:
1912     reasoning: Consistency check: verify parameter  $a$  effect on roots of  $B$ .
1913     next_check: Analyze  $a$  values to ensure  $B \subseteq A$ .
1914     Rule applied: rule_for_root.
1915     Why proceed: Directly links to final inclusion condition.
1916
1917     node 4:
1918     reasoning: Roots of  $B$ :  $x = 1$  and  $x = a - 1$ .
1919     branch: case1: both roots in  $A$  — case2: one root outside  $A$ 
1920     next_check: Whether  $a - 1 \in \{1, 2\}$ .
1921     Rule applied: rule_equation_analysis.
1922
1923     node 8 (terminated False):
1924     reasoning: If  $a - 1 \notin \{1, 2\}$ , then  $B$  contains elements not in  $A$ .
1925     conclusion: False
1926     Rule applied: rule_equation_analysis.
1927     Termination reason: Contradicts requirement  $B \subseteq A \rightarrow$  False.
1928
1929     node 5:
1930     reasoning: Require  $a - 1$  equals 1 or 2, so  $a = 2$  or  $a = 3$ .
1931     branch: case1:  $a = 2$  — case2:  $a = 3$ 
1932     next_check: Check consistency with discriminant.
1933     Rule applied: rule_set_operations.
1934
1935     node 9 (terminated False):
1936     reasoning: If  $a = 3$ , discriminant fails, contradiction.
1937     conclusion: False
1938     Rule applied: rule_set_operations.
1939     Termination reason: Inconsistent with quadratic constraints → False.
1940
1941     node 6 (backtrack):
1942     reasoning: Reconsider condition  $a - 1 \in \{1, 2\}$ . Correct range:  $1 \leq a \leq 2$ .
1943     backtrack_to: node 3
1944     next_check: Re-evaluate subset condition with corrected parameter range.
1945     Rule applied: rule_backtrack.
1946
1947     node 7:
1948

```

```

1944
1945 reasoning: Within  $1 \leq a \leq 2$ , roots of  $B$  are in  $\{1, 2\}$ , so  $B \subseteq A$ .
1946 conclusion: supports condition
1947 next_check: Finalize answer.
1948 Rule applied: rule_set_operations.

1949 node 10 (True final answer):
1950 final:  $1 \leq a \leq 2$ 
1951 rationale: Within  $1 \leq a \leq 2$ ,  $B \subseteq A$  holds, hence  $A \cup B = A$ .
1952 Rule applied: rule_finalize.
1953 Why valid: Matches expected answer → True
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997

```