Hierarchical Prompting Taxonomy: A Universal Evaluation Framework for Large Language Models Aligned with Human Cognitive Principles

Devichand Budagam
Indian Institute of Technology Kharagpur
India

Ashutosh Kumar
Rochester Institute of Technology
USA

Mahsa Khoshnoodi
Researcher, Fatima Fellowship
USA

Sankalp KJ
AI Institute, University of South Carolina
USA

Vinija Jain*
Amazon GenAI
Stanford University
USA

Aman Chadha*
Amazon GenAI
James Silberrad Brown Center for AI, San Diego State University
Stanford University
USA

Abstract

Assessing the effectiveness of large language models (LLMs) in performing different tasks is crucial for understanding their strengths and weaknesses. This paper presents Hierarchical Prompting Taxonomy (HPT), grounded on human cognitive principles and designed to assess LLMs by examining the cognitive demands of various tasks. The HPT utilizes the Hierarchical Prompting Framework (HPF), which structures five unique prompting strategies in a hierarchical order based on their cognitive requirement on LLMs when compared to human mental capabilities. It assesses the complexity of tasks with the Hierarchical Prompting Index (HPI), which demonstrates the cognitive competencies of LLMs across diverse datasets and offers insights into the cognitive demands that datasets place on different LLMs. This approach enables a comprehensive evaluation of LLM's problem-solving abilities and the intricacy of a dataset, offering a standardized metric for task complexity. Extensive experiments with multiple datasets and LLMs show that HPF enhances LLM performance by $2 \rightarrow 63\%$ compared to baseline performance, with GSM8k being the most cognitively complex task among reasoning and coding tasks with an average HPI of 3.20 confirming the effectiveness of HPT. To support future research in this domain, the implementations of HPT and HPF are publicly available[1].

CCS Concepts

• **Computing methodologies** → **Natural language generation**; *Reasoning about belief and knowledge*; **Cognitive science**.

*Work does not relate to position at Amazon.
[1]Code and Experiments

Prompt Optimization KDD 2025, Toronto, ON, Canada

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

Prompting Taxonomy, Cognitive Demands, Prompt Optimization

ACM Reference Format:
Devichand Budagam, Ashutosh Kumar, Mahsa Khoshnoodi, Sankalp KJ, Vinija Jain, and Aman Chadha. 2025. Hierarchical Prompting Taxonomy: A Universal Evaluation Framework for Large Language Models Aligned with Human Cognitive Principles. In . ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing (NLP), enabling significant advancements in a wide range of applications. Conventional evaluation frameworks often apply a standard prompting approach to assess different LLMs, regardless of the complexity of the task, which may result in biased and suboptimal outcomes. Moreover, applying the same prompting approach across all samples within a dataset without considering each sample's relative complexity adds to the unfair situation. To achieve a more balanced evaluation framework, it is essential to account for both the task-solving ability of LLMs and the varying cognitive complexities of the dataset samples. This limitation highlights the need for more sophisticated evaluation methods that can adapt to varying levels of sample task complexity. This study defines *complexity* as the cognitive demands imposed by a task or the cognitive load introduced by a prompting strategy on LLMs. Task complexity in human cognition reflects the mental effort required for processing, analyzing, and synthesizing information. As Sweller [30] noted, complexity increases with greater cognitive resource demands, engaging working memory in reasoning and problem-solving. Similarly, Anderson et al. [2] describes cognitive abilities as a continuum, from basic recall to higher-order thinking, with difficulty rising for tasks requiring analysis, synthesis, and evaluation. By mapping LLM prompting strategies onto this hierarchy, we systematically assess how LLMs handle varying cognitive loads. This framework provides a structured, cognitively grounded method for evaluating model performance across tasks of differing complexity. This study is directed by the following research questions:

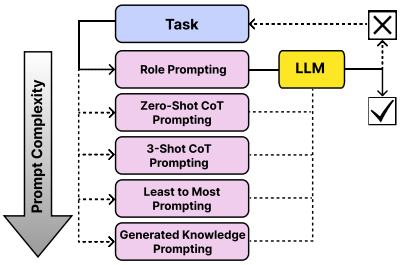

Figure 1: The Hierarchical Prompting Framework includes five distinct prompting strategies, each designed for different levels of task complexity to ensure the appropriate prompt is selected for the given task. A ✓ indicates task completion, while a × signifies task incompletion.

Research Questions:

- **RQ1:** Does cognitively inspired strategic selection of prompts enable small language models (SLMs) to match the performance of LLMs?
- **RQ2:** How can cognitive demand measurements of LLMs guide model selection and deployment decisions beyond traditional metrics?
- **RQ3:** How can we align prompt complexity with task demands to optimize both computational efficiency and performance?

This paper introduces the HPT, a set of rules that maps the human cognitive principles for assessing the complexity of different prompting strategies. It employs the HPF shown in Figure 1, a prompt selection framework that selects the prompt with the optimal cognitive load on LLM required in solving the task. The main contributions of this work are:

- **Hierarchical Prompting Taxonomy (HPT):** The paper introduces HPT, rules mapping prompting strategies to human cognitive principles, enabling a universal measure of LLMs' task complexity.
- **Hierarchical Prompting Framework (HPF):** The HPF framework selects the best prompt from five strategies to optimize LLMs' cognitive load, improving evaluation and performance transparency.
- **Hierarchical Prompting Index (HPI):** HPI[2] quantitatively assesses LLMs' task complexity across datasets, revealing cognitive demands on various LLMs.

HPF can be compared to an *"open book"* exam (see Figure 2), with tasks analogous to questions and prompting strategies akin to textbooks. The exam questions, ranging from basic recall to complex analysis, parallel the cognitive challenges in HPT tasks. Similarly, textbooks offer structured support, much like HPF, which arranges prompts by complexity to assist LLMs. A glossary lookup represents a task with low complexity, whereas solving a multi-step analytical problem indicates high complexity. The effort exerted by a student is similar to HPI, which measures the cognitive demand on LLMs. Just as structured learning materials improve students' performance, carefully crafted hierarchical prompts help LLMs in addressing increasingly complex tasks more effectively.

[2]HPI can be quantitatively assessed to analyze the cognitive abilities of an LLM and the cognitive demands imposed by datasets on LLMs, as both factors are interchangeably related to the complexity of tasks.

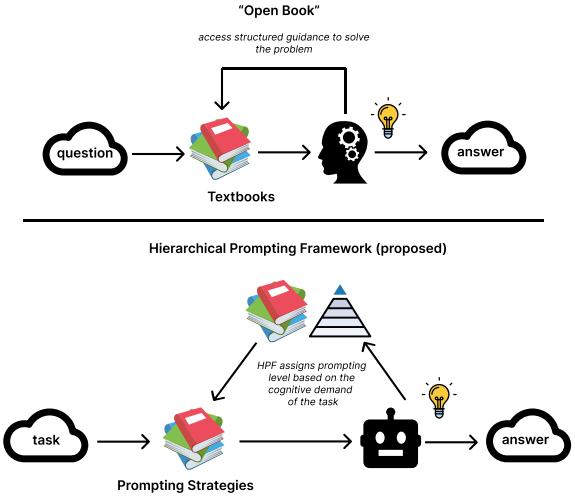

Figure 2: Analogical framework comparing the HPF with "open book" examination methodology. The diagram illustrates how HPF components (below) mirror traditional educational assessment elements (above), with parallel relationships between task complexity levels, resource utilization (prompts/textbooks), and performance metrics (HPI/student effort). This comparison demonstrates how LLM task complexity scales similarly to educational assessment complexity, from simple lookup tasks to complex synthesis problems.

The remainder of the paper is structured as follows: Section 2 reviews the related work on prompting and evaluation in LLMs. Section 3 details the HPT and its associated frameworks. Section 4 outlines the experimental setup, results, and ablation studies. Section 5 concludes the paper. Section 6 discusses the limitations of the work. Section 7 discusses the ethical impact of the work.

2 Related Work

The advent of LLMs has revolutionized NLP by demonstrating significant improvements in few-shot and zero-shot learning capabilities. Brown et al. [6] introduced GPT-3, a 175 billion parameter autoregressive model, showcasing its ability to perform a wide range of tasks such as question-answering, reading comprehension, translation, and natural language inference without fine-tuning. This study highlighted the potential of very large models for in-context learning while also identifying limitations in commonsense reasoning and specific comprehension tasks. Similarly, Liu et al. [23] surveyed prompt-based learning, emphasizing the role of prompt engineering in leveraging pre-trained models for few-shot and zero-shot adaptation to new tasks with minimal labeled data.

2.1 Prompt Engineering

Prompting plays a vital role in unlocking the full potential of LLMs. By designing specific input prompts, the LLM's responses can be guided, significantly influencing the quality and relevance of the output. Effective prompting strategies have enhanced LLM performance on tasks ranging from simple question-answering to complex reasoning and problem-solving. Recent research has explored various approaches to prompting and reasoning evaluation in LLMs. Chain-of-Thought (CoT) prompting [39] elicits step-by-step reasoning, improving performance on complex tasks. Specializing smaller models [13] and using large models as reasoning teachers [16] have demonstrated the potential for enhancing reasoning capabilities. Emergent abilities in LLMs, which appear suddenly at certain scale thresholds, have also been a topic of interest. Wei et al. [38] examined these abilities in few-shot prompting, discussing the underlying factors and implications for future scaling. Complementing this, Kojima et al. [19] demonstrated that LLMs could exhibit multi-step reasoning capabilities in a zero-shot setting by simply modifying the prompt structure, thus highlighting their potential as general reasoning engines. Yao et al. [40] introduced the Tree-of-Thoughts framework, enabling LLMs to deliberate over coherent text units and perform heuristic searches for complex reasoning tasks. This approach generalizes over chain-of-thought prompting and has shown significant performance improvements in tasks requiring planning and search, such as creative writing and problem-solving games. Kong et al. [20] introduced role-play prompting to improve zero-shot reasoning by constructing role-immersion interactions, which implicitly trigger chain-of-thought processes and enhance performance across diverse reasoning benchmarks. Progressive-hint prompting [41] has been proposed to conceptualize answer generation and guide LLMs toward correct responses. Metacognitive prompting [37] incorporates self-aware evaluations to enhance understanding abilities.

These studies highlight progress in using innovative prompting techniques to improve LLMs' emergent abilities, reasoning, interaction strategies, robustness, and evaluation. Yet, challenges persist in prompt design, managing complex reasoning tasks, and performance evaluation across various scenarios. Although LLMs show promising emergent abilities, they frequently lack predictability and control, and their resistance to misleading prompts is still an issue.

2.2 Prompt Optimization and Selection

The challenge of optimizing prompts for LLMs has been addressed in several key studies, each contributing unique methodologies to enhance model performance and efficiency. Shen et al. [29] introduce PFLAT, a metric utilizing flatness regularization to quantify prompt utility, which leads to improved results in classification tasks. Do et al. [12] propose a structured three-step methodology that contains data clustering, prompt generation, and evaluation, effectively balancing generality and specificity in prompt selection. ProTeGi [27] offers a non-parametric approach inspired by gradient descent, leveraging natural language "gradients" to iteratively refine prompts. Wang et al. [36] present PromISe, which transforms prompt optimization into an explicit chain of thought, employing self-introspection and refinement techniques. Zhou et al. [43] proposed DYNAICL, a framework for efficient prompting that dynamically allocates in-context examples based on a meta-controller's predictions, achieving better performance-efficiency trade-offs compared to uniform example allocation.

These studies seek to automate prompt design, reducing reliance on manual trial-and-error while improving efficiency and scalability across tasks and models. They report performance gains of 5% to 31% across benchmarks, highlighting the growing significance of prompt optimization. Future research directions include exploring theoretical foundations, combining optimization techniques, and differentiating task-specific from general-purpose strategies.

2.3 Evaluation Benchmarks

To facilitate the evaluation and understanding of LLM capabilities, Zhu et al. [44] introduced PromptBench, a unified library encompassing a variety of LLMs, datasets, evaluation protocols, and adversarial prompt attacks. This modular and extensible tool aims to support collaborative research and advance the comprehension of LLM strengths and weaknesses. Further exploring reasoning capabilities, Qiao et al. [28] categorized various prompting methods and evaluated their effectiveness across different model scales and reasoning tasks, identifying key open questions for achieving robust and generalizable reasoning. [35] introduced a multitask benchmark for LLM robustness evaluation, which extends the original GLUE [34] benchmark to assess model robustness against adversarial inputs. It incorporates perturbed versions of existing GLUE tasks, such as paraphrasing, negation, and noise, to test models' abilities with challenging data. The study highlights that despite their success on clean datasets, state-of-the-art models often struggle with adversarial examples, underscoring the importance of robustness evaluations in model development.

3 Hierarchical Prompting Taxonomy

3.1 Governing Rules

Figure 3 illustrates the HPT, a taxonomy that systematically reflects human cognitive functions as outlined in Bloom [4]. Each rule embodies complex cognitive processes based on established principles from learning and psychology.

(1) **Basic Recall and Reproduction**: This reflects the fundamental cognitive process of remembering and reproducing factual information without analysis or interpretation, which involves mere recognition or retrieval of knowledge from memory [2].

(2) **Understanding and Interpretation**: This corresponds to the second cognitive rule of [4], where individuals must not only recall information but also explain it in their own words, summarize key points or clarify the meaning of content. This rule demands an intermediate cognitive load involving information processing rather than retrieving it.

(3) **Analysis and Reasoning**: This aligns with the analysis stage of [4], which involves higher cognitive functions such as comparison, contrast, and deep understanding of the underlying principles. It is more complex than mere understanding because it requires examining structure and identifying patterns and connections.

(4) **Application of Knowledge and Execution**: This mirrors the application and evaluation stages of [4], where individuals must not only understand and analyze but also use knowledge to perform multi-step tasks, solve complex problems, and

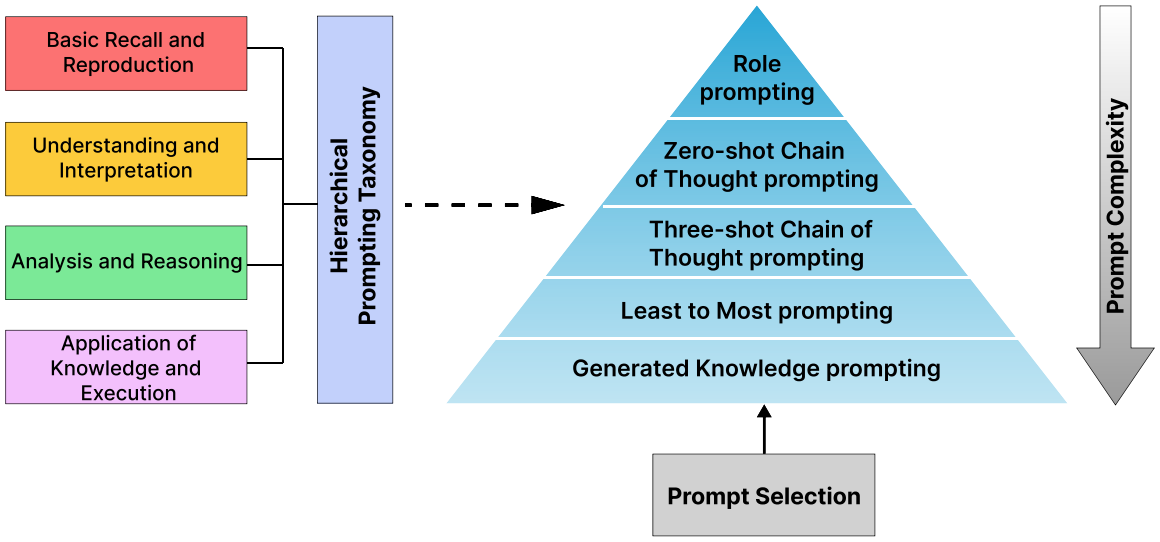

Figure 3: Hierarchical Prompting Taxonomy: A taxonomy designed to assess the complexity of prompting strategies based on the criteria: Basic Recall and Reproduction, Understanding and Interpretation, Analysis and Reasoning, and Application of Knowledge and Reasoning.

execute decisions. It represents the most cognitively complex tasks, which require synthesis of information and practical decision-making, highlighting the critical leap from understanding theory to executing it in practice.

In HPT, the progression from basic recall to application of knowledge reflects increasing cognitive complexity, consistent with educational and cognitive frameworks, where more advanced cognitive processes build on foundational ones, demanding deeper engagement and mental effort.

3.2 Hierarchical Prompting Framework

The HPF consists of five prompting strategies, each assigned a complexity level. These levels are determined by the degree to which the strategies are shaped by the four principles of the HPT. The complexity levels of the prompting strategies are assigned based on human assessment of their relative cognitive loads over a set of 7 different tasks, guaranteeing that the cognitive abilities of LLMs are in harmony with those of humans. This approach enables the assessment of tasks in terms of their complexity and the cognitive load they impose on both humans and LLMs by utilizing HPI. Section 4.5 examines the hierarchical structure of the HPF in conjunction with the LLM-as-a-Judge framework, validating that the cognitive demands on LLMs can be aligned with those of humans.

The five prompting strategies were selected to ensure comprehensive coverage of cognitive demands rather than maximizing the number of strategies (see Appendix A). This makes HPF adaptable, allowing for replication or expansion with similar strategies. The strategies, ordered by increasing complexity, are:

(1) **Role Prompting** [20]: Specifies the LLM's role in task resolution, exerting minimal influence from HPT principles.

(2) **Zero-Shot Chain-of-Thought Prompting (Zero-CoT)** [19]: Uses "Let's think step by step" to encourage reasoning, moderately influenced by rule 3.

(3) **Three-Shot Chain-of-Thought Prompting (3-CoT)** [39]: Provides three examples to guide reasoning, strongly influenced by rules 1 and 2, with moderate influence from rule 3.

(4) **Least-to-Most Prompting** [42]: Breaks tasks into sub-problems, requiring recall, interpretation, and analysis, exerting strong influence from rules 1, 2, & 3.

(5) **Generated Knowledge Prompting (GKP)** [22]: Integrates external knowledge, demanding correlation, application, and analysis, making it the most cognitively complex (rules 2, 3, and 4). Llama-3 8B generates the external knowledge in experiments.

3.3 Hierarchical Prompting Index

HPI is an evaluation metric for assessing the task complexity of LLMs over different datasets, which is influenced by the HPT rules. A lower HPI for a dataset suggests that the corresponding LLM is more adept at solving the task with fewer cognitive processes. For each dataset instance, we begin with the least complex prompting strategy and progressively move through the HPF prompting strategies until the instance is resolved. The HPI corresponds to the complexity level of the prompting strategy where the LLM first tackles the instance.

Algorithm 1 illustrates the process for determining HPI, with m indicating the total levels within the HPF and n representing the number of samples in the evaluation dataset. $\texttt{HPI}_{Dataset}$ denotes the penalty that human evaluations impose on the framework. Additional information regarding human annotation is provided in Appendix A.

Algorithm 1 HPI Computation

```
HPI_List = [ ]
for sample i in the evaluation dataset do
    for level x in the HPF do
        if LLM resolves the task then
            HPI_List[i] = x
            break
        end if
    end for
    if LLM failed to resolve the task then
        HPI_List[i] = m + HPI_Dataset
    end if
end for
HPI = (1/n) Σ_{j=1}^{n} HPI_List[j]
```

4 Results

4.1 Experimental Setup

Datasets

We evaluated the framework on diverse datasets spanning reasoning, coding, mathematics, question-answering, summarization, and machine translation. For dataset sizes, see Appendix A.

Reasoning: MMLU [15] (57 subjects, multiple-choice), CSQA [31] (12K commonsense questions).

Coding: HumanEval [8] (164 function-based coding tasks).

Mathematics: GSM8k [11] (8.5K multi-step math problems).

Question-Answering: BoolQ [10] (16K True/False questions from Wikipedia).

Summarization: SamSum [14] (16K human-annotated dialogue summaries).

Machine Translation: IWSLT-2017 en-fr [7] (TED Talk parallel corpus).

Large Language Models: We tested LLMs ranging from 7B to 12B parameters across open-source and proprietary models.

Proprietary LLMs: GPT-4o [25], Claude 3.5 Sonnet [3].

SLMs: Gemma 7B [32], Mistral 7B [17], Llama-3 8B [1], Gemma-2 9B [33], Mistral-Nemo 12B [24].

Additional Evaluation Metrics

Coding: Pass@k [9] estimates the probability of at least one correct solution among the top k outputs for code generation.

Summarization: ROUGE-L [21] measures sequence-level similarity via the longest common subsequence.

Machine Translation: BLEU [26] evaluates n-gram precision against reference texts.

Summarization and translation tasks used thresholds of 0.15 and 0.20, respectively, to define task completion at each HPF complexity level, enabling iterative refinement of prompting strategies.

4.2 Results on Standard Benchmarks: MMLU, GSM8K, and Humaneval

The evaluation of HPF effectiveness as shown in Figure 4 spans three standard benchmarks: MMLU, GSM8k, and HumanEval. On the MMLU benchmark, which tests general knowledge across multiple domains, all models showed notable improvements over their

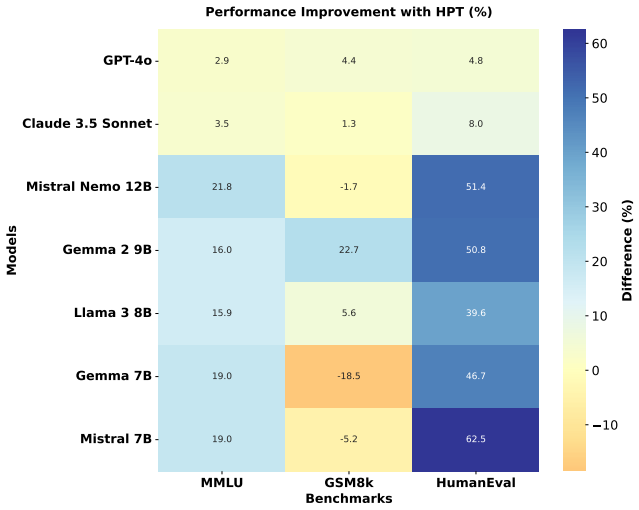

Figure 4: Performance Comparison of HPT-based Evaluation vs. Standard Evaluation: Performance improvements (in %) when using HPT-based evaluation compared to standard evaluation across three benchmarks: MMLU, GSM8k, and HumanEval. Positive values indicate performance gains with HPT, while negative values indicate performance decreases. The baseline standard evaluation scores are sourced from Hugging Face leaderboard and official research reports.

baseline performance. Mistral-Nemo 12B demonstrated the most substantial MMLU enhancement (+21.8%), while Claude 3.5 Sonnet achieved a consistent improvement of 3.5%. In mathematical reasoning, assessed through GSM8k, the results revealed a correlation with the model scale. Larger models like GPT-4 and Claude 3.5 Sonnet showed modest gains (+4.4% and +1.3% respectively), while smaller models exhibited more variable performance. The HumanEval benchmark, which assesses code generation capabilities, revealed the most dramatic improvements across all models. Mistral 7B achieved an exception 62.5% improvement in HumanEval scores, followed by Mistral-Nemo 12B with an impressive 51.4% improvement, and Gemma-2 9B with a 50.8% enhancement. The results suggest that HPF enhances performance on all benchmarks for the majority of SLMs and achieves similar performance to LLMs such as GPT-4o and Claude 3.5 Sonnet, thereby addressing **RQ1**, its impact is particularly pronounced in programming tasks, suggesting that the technique may be especially valuable for enhancing code-related capabilities.

Table 1 highlights the improved performance of various LLMs on MMLU, with all models showing an HPI index below three. This indicates that reasoning over most MMLU samples requires minimal cognitive effort for these models, compared to baseline multi-shot CoT methods (5 shot), which typically require more than five examples and are more cognitively demanding according to HPT. Interestingly, while Claude 3.5 Sonnet achieves the highest MMLU accuracy, GPT-4o records the best HPI score, showing that minimal cognitive effort does not necessarily equate to the best performance addressing **RQ2**. The enhancement in GSM8k is relatively smaller compared to MMLU, with decreased performances for both Mistral

Table 1: HPI (lower is better) and accuracy of LLMs across MMLU, GSM8K, BoolQ, and CSQA datasets. Blue indicates datasets where the LLM with the best HPI does not achieve the best performance. Green indicates the LLM with the best performance over the maximum number of datasets.

DATASETS	MMLU		GSM8k		BoolQ		CSQA	
Models	**HPI**	**Accuracy**	**HPI**	**Accuracy**	**HPI**	**Accuracy**	**HPI**	**Accuracy**
GPT-4o	1.81	91.61	1.71	96.43	1.32	96.82	**1.65**	**92.54**
Claude 3.5 Sonnet	1.84	92.16	**1.35**	**97.72**	1.20	99.81	2.01	86.15
Mistral-Nemo 12B	2.45	89.75	3.01	86.80	1.75	99.87	2.06	90.17
Gemma-2 9B	2.34	87.28	2.17	91.28	1.30	98.28	1.94	88.86
Llama-3 8B	2.84	82.63	2.34	86.20	1.37	99.30	2.43	84.76
Gemma 7B	2.93	83.31	6.70	27.88	1.45	99.42	2.50	83.78
Mistral 7B	2.89	81.45	5.11	46.93	1.41	98.07	2.49	82.06

7B and Gemma 7B. The high HPI values for Gemma 7B and Mistral 7B indicate that none of the five prompting strategies in HPF posed significant cognitive challenges for these LLMs, i.e more cognitively demanding prompting strategies are needed, highlighting a limitation of the HPF. As shown in Table 2, Claude 3.5 Sonnet achieves a perfect pass@1 of 1.00 with low HPI values, outperforming GPT-4o, which scores 0.95 but has a higher HPI. Gemma 7B struggles with the lowest pass@1 of 0.79 and the highest HPI of 3.71, indicating a need for a more complex prompting strategy.

Notably, HPF noticeably boosted the performance of the majority of LLMs on three benchmark datasets, despite the HPI difference being less than 1 compared to the top-performing LLMs. This suggests that even with a minimal number of inferences, utilizing HPF can achieve optimal performance, unlike multi-shot prompting and prompt optimization strategies, thereby addressing **RQ3**. This highlights that tailoring the prompting strategy to align with the complexity of each dataset instance can lead to substantial improvements, achieving performance levels comparable to state-of-the-art LLMs such as GPT-4o and Claude 3.5 Sonnet on these benchmarks.

Table 2: HPI (lower is better) and Pass@1 of LLMs on the HumanEval dataset. Blue indicates datasets where the LLM with the best HPI does not achieve the best performance. Green indicates the LLMs with the best performance over the dataset.

DATASET	HumanEval	
Models	**HPI**	**Pass@1**
GPT-4o	2.25	0.95
Claude 3.5 Sonnet	1.04	1.00
Mistral-Nemo 12B	2.07	0.96
Gemma-2 9B	1.01	0.91
Llama-3 8B	1.03	1.00
Gemma 7B	3.71	0.79
Mistral 7B	1.10	0.93

4.3 Results on Other Datasets

Table 1 presents LLM performance on the BoolQ and CSQA datasets. While no significant insights emerge, an unexpected result is GPT-4o's poor performance, which deviates from its typical trend. With most LLMs achieving near-perfect scores, BoolQ appears insufficiently complex to serve as an effective benchmark for modern LLMs, as they excel even with minimal cognitive prompting. This highlights HPF's value in assessing dataset complexity relative to LLM capabilities, providing researchers with insights for designing more challenging and robust benchmarks.

Table 3 presents the performance of LLMs on IWSLT and SamSum datasets at varying thresholds. GPT-4o consistently achieved the highest scores across all thresholds, while most models, except Gemma 7B, performed similarly. Interestingly, Claude 3.5 Sonnet, which excelled in reasoning tasks, did not perform as strongly in summarization and translation tasks. The threshold selection is guided by the observed performance plateau across most LLMs as the threshold increases.

4.4 Threshold Selection for SamSum and IWSLT

In addition to the 0.15 and 0.20 thresholds presented in the main experiments, extended evaluations were conducted on the IWSLT and SamSum datasets using thresholds of 0.25 and 0.30 with GPT-4o, Mistral-Nemo 12B, and Llama-3 to assess the impact of varying thresholds on LLM performance.

SamSum Dataset: In the summarization task, increasing the threshold evaluates an LLM's ability to condense content while retaining key information. Higher thresholds like 0.25 and 0.30 reveal the trade-offs between conciseness and informativeness. However, as shown in Figure 5, there was no significant improvement in ROUGE-L, except for a slight increase with GPT-4o. The experiments showed a sharp rise in HPI, reflecting the increased task complexity. These results suggest that LLM performance has plateaued, with no further gains at higher thresholds. This validates that the use of 0.15 and 0.20 thresholds are sufficient for optimal LLM performance.

IWSLT Dataset: In machine translation, higher thresholds (0.25 and 0.30) impose stricter evaluations, assessing how well models capture the nuances of the source text. Lower thresholds (0.15 and 0.20) focus on general adequacy, while higher ones test performance under more challenging conditions. As shown in Figure 6, no BLEU improvements were observed across any LLMs, with models either reaching saturation or showing decreased performance alongside a rapid rise in HPI. This validates the selection of 0.15 and 0.20 thresholds are sufficient for optimal LLM performance.

Table 3: HPI (lower is better), BLEU score for IWSLT, and ROUGE-L score for SamSum, of LLMs with thresholds.

DATASETS	IWSLT				SamSum			
	HPI		BLEU		HPI		ROUGE-L	
Models	**0.15**	**0.20**	**0.15**	**0.20**	**0.15**	**0.20**	**0.15**	**0.20**
GPT-4o	**2.66**	**3.08**	**0.32**	**0.32**	**1.11**	**1.21**	**0.30**	**0.29**
Claude 3.5 Sonnet	4.63	4.87	0.20	0.20	1.25	1.60	0.23	0.23
Mistral-Nemo 12B	2.87	3.40	0.27	0.27	1.19	1.47	0.23	0.24
Gemma-2 9B	4.40	4.75	0.21	0.20	1.30	1.86	0.22	0.22
Llama-3 8B	3.40	3.92	0.24	0.23	1.30	1.72	0.22	0.22
Gemma 7B	5.39	5.84	0.08	0.09	3.31	5.03	0.11	0.10
Mistral 7B	3.52	4.14	0.22	0.22	1.26	1.68	0.21	0.22

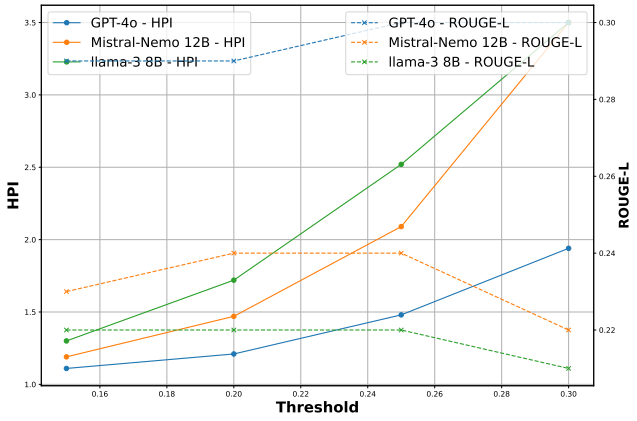

Figure 5: Comparison of HPI and ROUGE-L scores across different threshold values on SamSum dataset.

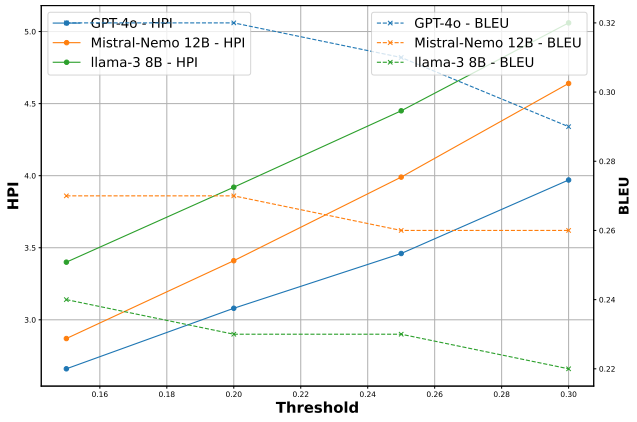

Figure 6: Comparison of HPI and BLEU score across different threshold values in the translation task.

4.5 Complexity Levels with LLM-as-a-Judge

This study evaluated prompting strategies by assessing how GPT-4o, as the LLM judge, replicates the hierarchical complexity levels of these strategies using a systematic scoring approach across tasks. Figure 7 shows a consistent hierarchy with less variability than human judges, indicating a strong alignment between LLM and human judgment. These results validate the proposed framework and demonstrate the correspondence between human cognitive principles and LLM behavior. Figure 8 shows the scoring distribution across the four HPT rules for each strategy. Further details related to evaluation dataset specifications and scoring method are in Appendix B.

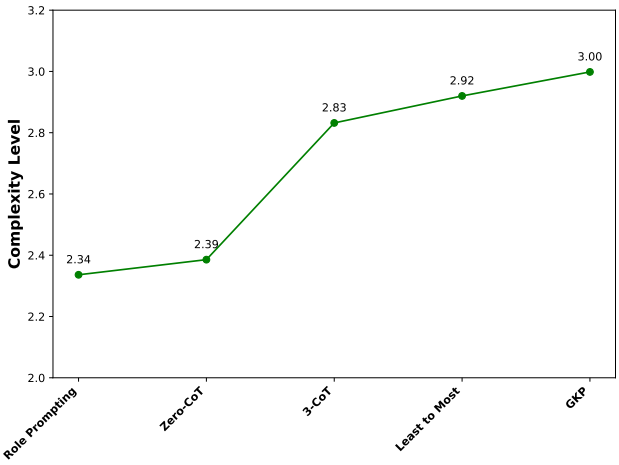

Figure 7: Hierarchy of prompting strategies with LLM-as-a-Judge framework with GPT-4o as the judge.

4.6 Parallels with System 1 and System 2 Thinking

HPF parallels dual-process cognitive theories' System 1 and System 2 thinking [5, 18]. HPT classifies tasks, and HPF designs prompts based on cognitive complexity, reflecting human cognitive resource allocation. For tasks with low cognitive demands, HPF uses simple prompts akin to System 1 thinking, like fact recall or basic classification, enabling quick LLM responses with minimal reasoning. Conversely, tasks with high cognitive demands require prompts for complex reasoning and problem-solving, similar to System 2

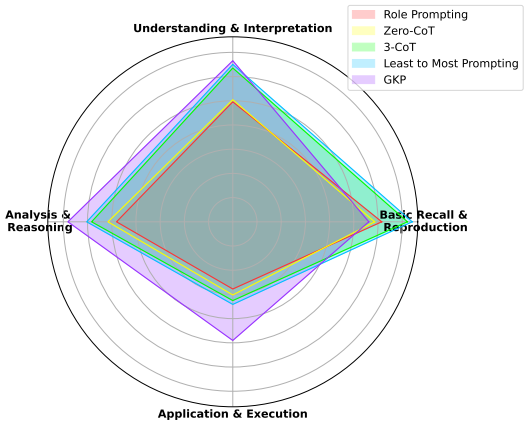

Figure 8: Scoring distribution for each of the four rules of the HPT for the prompting strategies in the HPF.

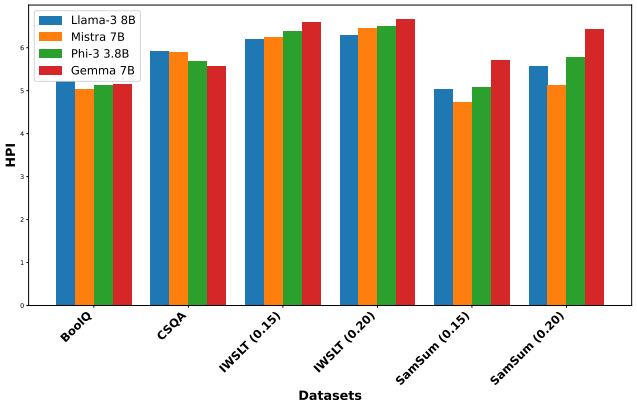

Figure 9: HPI of datasets for LLMs in Adaptive HPF.

thinking, involving logical arguments or intricate problems needing deliberate processing. Elevated HPF levels are used for tasks demanding deep analysis.

HPF explicitly measures this transition with HPI, assessing the cognitive load required for each task. By tailoring prompting strategies to task complexity, HPF optimizes LLM performance, much like humans adaptively switch between System 1 and System 2 based on the situation. This parallel highlights how HPT bridges computational strategies with human-like cognitive models, enabling more nuanced task evaluation and resource allocation.

4.7 Adaptive HPF

The Adaptive HPF automates the selection of the optimal complexity level in the HPF using a *prompt-selector*, Llama-3 8B in a zero-shot setting, bypassing iterative steps. Figure 9 shows that Adaptive HPF yields higher HPI but lower evaluation scores than the standard HPF. This suggests that Adaptive HPF struggles to select the optimal complexity level, likely due to hallucinations by the *prompt-selector* when choosing the prompting strategy. For more results and ablation studies, see Appendix C.

The *prompt-selector* can dynamically select the most suitable prompting strategy for a given task's complexity from the HPF's hierarchy of complexity levels. To determine the most effective prompting strategy to complete the task, the *prompt-selector* was given a maximum number of iterations equivalent to the number of levels in the manual HPF. The score for ith iteration is $i + x$, where x is the complexity level by the *prompt-selector*. If the LLM fails to complete the task after all iterations, it is assigned a penalty, $\texttt{HPI}_{Dataset}$. Algorithm 2 demonstrates the calculation of HPI for an adaptive HPF, where x denotes the HPF level chosen by the *prompt-selector* at the ith iteration as the task is being tackled. Here, m indicates the total number of HPF levels, and n signifies the total quantity of samples in the evaluation set.

Algorithm 2 HPI Computation for Adaptive HPF

```
HPI_List = []
for sample j in the evaluation dataset do
    solved = False
    for iteration i = 1 to m do
        Select prompting strategy at level x
        if LLM completes the task at iteration i then
            HPI_List[j] = x + i
            solved = True
            break
        end if
    end for
    if solved = False then
        HPI_List[j] = m + HPI_Dataset
    end if
end for
HPI_Adaptive = (1/n) Σ_{j=1}^{n} HPI_List[j]
```

5 Conclusion

The HPT offers an efficient way to evaluate LLMs by focusing on task cognitive demands. It shows that cognitively inspired selection of prompting strategies enhances LLM performance across various datasets. This method offers insights into LLM problem-solving and improves evaluation methods based on human cognition, supporting better in-context learning strategies for assessing LLMs.

6 Limitations

Human Annotation Constraints: A limitation of this study is the reliance on human evaluation for inducing the $\texttt{HPI}_{Dataset}$ penalty into the HPF. While this study assessed 5% of the datasets, expanding coverage would offer a more comprehensive analysis. However, due to constraints in human resources for manual annotation, we could not include a larger portion. Future work could address this by increasing manpower or automating parts of the evaluation process.

HPF Optimization: The effectiveness of the HPF heavily relies on the quality of the prompts used at each level of the taxonomy. Crafting high-quality prompts that accurately reflect the subtleties of

each level demands considerable expertise and repeated refinement. This study only investigated a limited set of prompting strategies within the HPF, indicating a need for further research into creating diverse structural frameworks and incorporating additional prompting strategies.

Zero-shot Prompt Selection: HPF dynamically determines the optimal cognitive complexity level by iterating through the framework's levels, which leads to increased inference time. While this study investigated Adaptive HPF for zero-shot prompt selection, it faced considerable hallucinations. Future research should focus on automating HPF using fine-tuning or reinforcement learning-based approaches to select the appropriate complexity level without manual iteration. This strategy would optimize inference time and improve overall performance.

7 Ethical Statement

The $\texttt{HPI}_{Dataset}$ assigned by experts to MMLU, GSM8k, HumanEval, BoolQ, CSQA, IWSLT, and SamSum may introduce bias due to the subjective nature of expert scoring, influenced by individual experience and perspective. However, these publicly available, widely recognized datasets help mitigate unforeseen ethical concerns. Acknowledging potential scoring bias remains essential for transparency and integrity in the analysis.

References

[1] AI@Meta. 2024. Llama 3 Model Card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[2] L.W. Anderson, D. Krathwohl, K. Cruikshank, P. Airasian, J. Raths, P. Pintrich, R. Mayer, and M. Wittrock. 2014. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's*. Pearson. https://books.google.com/books?id=d0gxngEACAAJ

[3] Anthropic. 2024. Claude 3.5 Sonnet. https://www.anthropic.com/claude-3-5-sonnet. Accessed: 2024-09-16.

[4] B.S. Bloom. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Number v. 1 in Taxonomy of Educational Objectives: The Classification of Educational Goals. Longmans, Green. https://books.google.co.in/books?id=hos6AAAAIAAJ

[5] Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andreas Loreggia, Keerthiram Murgesan, Nicholas Mattei, Francesca Rossi, et al. 2021. Thinking fast and slow in AI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15042–15046.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*. International Workshop on Spoken Language Translation, Tokyo, Japan, 2–14. https://aclanthology.org/2017.iwslt-1.1

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs.LG] https://arxiv.org/abs/2107.03374

[10] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 2924–2936. doi:10.18653/v1/N19-1300

[11] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168* (2021).

[12] Viet-Tung Do, Van-Khanh Hoang, Duy-Hung Nguyen, Shahab Sabahi, Jeff Yang, Hajime Hotta, Minh-Tien Nguyen, and Hung Le. 2024. Automatic Prompt Selection for Large Language Models. arXiv:2404.02717 [cs.CL] https://arxiv.org/abs/2404.02717

[13] Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing Smaller Language Models towards Multi-Step Reasoning. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 10421–10430. https://proceedings.mlr.press/v202/fu23d.html

[14] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (Eds.). Association for Computational Linguistics, Hong Kong, China, 70–79. doi:10.18653/v1/D19-5409

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).

[16] Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071* (2022).

[17] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825

[18] Daniel Kahneman. 2011. *Thinking, fast and slow*. macmillan.

[19] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[20] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better Zero-Shot Reasoning with Role-Play Prompting. arXiv:2308.07702

[21] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[22] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 3154–3169. doi:10.18653/v1/2022.acl-long.225

[23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv:2107.13586

[24] Mistral AI and NVIDIA. 2024. Mistral NeMo 12B. https://mistral.ai/news/mistral-nemo/. Accessed: 2024-09-16.

[25] OpenAI. 2024. GPT-4o. https://openai.com/gpt-4. Accessed: 2024-09-16.

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, Pennsylvania) *(ACL '02)*. Association for Computational Linguistics, USA, 311–318. doi:10.3115/1073083.1073135

[27] Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search.

arXiv preprint arXiv:2305.03495 (2023).

[28] Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with Language Model Prompting: A Survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 5368–5393. doi:10.18653/v1/2023.acl-long.294

[29] Lingfeng Shen, Weiting Tan, Boyuan Zheng, and Daniel Khashabi. 2023. Flatness-aware prompt selection improves accuracy and sample efficiency. *arXiv preprint arXiv:2305.10713* (2023).

[30] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science* 12, 2 (1988), 257–285. doi:10.1016/0364-0213(88)90023-7

[31] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. arXiv:1811.00937

[32] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295

[33] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118* (2024).

[34] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (Eds.). Association for Computational Linguistics, Brussels, Belgium, 353–355. doi:10.18653/v1/W18-5446

[35] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models. In *Advances in Neural Information Processing Systems*.

[36] Minzheng Wang, Nan Xu, Jiahao Zhao, Yin Luo, and Wenji Mao. 2024. PromISe: Releasing the Capabilities of LLMs with Prompt Introspective Search. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 13120–13130. https://aclanthology.org/2024.lrec-main.1149

[37] Yuqing Wang and Yun Zhao. 2024. Metacognitive Prompting Improves Understanding in Large Language Models. arXiv:2308.05342

[38] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=yzkSU5zdwD Survey Certification.

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[40] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=5Xc1ecxO1h

[41] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-Hint Prompting Improves Reasoning in Large Language Models. arXiv:2304.09797

[42] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WZH7099tgfM

[43] Wangchunshu Zhou, Yuchen Eleanor Jiang, Ryan Cotterell, and Mrinmaya Sachan. 2023. Efficient Prompting via Dynamic In-Context Learning. arXiv:2305.11170

[44] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. PromptBench: A Unified Library for Evaluation of Large Language Models. arXiv:2312.07910

A Human Annotation and Judgement Policy

A.1 Human Annotation Policy

$\texttt{HPI}_{Dataset}$ is introduced to penalize the HPI of tasks or samples unsolvable by the LLM, aligning the framework more closely with human cognitive demands and enhancing its comprehensiveness. We implemented a rigorous human annotation process to ensure the quality of $\texttt{HPI}_{Dataset}$ scored by human experts for the datasets. Human annotators were tasked with calculating the HPI for each sample in a given dataset. The HPI quantifies the cognitive demands imposed on human expert proficiency in completing a task, based on the HPT, where higher values indicate greater cognitive demands. Each sample was scored on a scale from *1* (lowest complexity level) to *5* (highest complexity level) for the following criteria:

(1) **Basic Understanding and Reproduction**: This criterion evaluates the annotator's ability to comprehend and accurately reproduce the content.
(2) **Understanding and Interpretation**: This criterion assesses the annotator's depth of understanding and the ability to interpret the information correctly.
(3) **Analysis and Reasoning**: This criterion measures the annotator's ability to analyze the information and apply logical reasoning.
(4) **Application of Knowledge and Execution**: This criterion evaluates the annotator's practical application of knowledge and the execution of tasks based on the relevant knowledge.

Higher scores for the four rules signify a stronger influence of the respective rules, indicating that completing the task requires greater cognitive effort. The $\texttt{HPI}_{Dataset}$ for each dataset, as shown in Table 4, was calculated by taking the mean of the values from these four criteria, acknowledging the challenge of estimating or computing the individual weights of the influence of each rule.

The `Representative Set Size` in Table 4 refers to the subset of the dataset evaluated by human annotators, ensuring that the assessment reflects the overall task. Human annotation, while time-consuming and costly, provides a gold standard for calibrating the evaluation process of this paper. Selecting 5% of the dataset as the representative set size balances quality assessment and feasibility, capturing the dataset's diversity and ensuring that human annotations encompass a broad range of cases without needing to annotate every sample.

A.2 Human Judgement Policy

To populate the HPF with relevant prompting strategies across a wide range of strategies, human annotators who adhered to the annotation policy for assessing $\texttt{HPI}_{Dataset}$ were instructed to follow a judgment policy for a predefined set of prompting strategies. They

Table 4: $\texttt{HPI}_{Dataset}$ scores across datasets evaluated by human annotators. The table lists the evaluation set size, representative set size, and $\texttt{HPI}_{Dataset}$ for various datasets. $\texttt{HPI}_{Dataset}$ scores provide a measure of task complexity relative to human annotators.

Dataset	Evaluation Set Size	Representative Set Size	$\texttt{HPI}_{Dataset}$
MMLU	14500	725	3.03
GSM8k	1320	66	2.14
Humaneval	160	8	4.68
BoolQ	3270	162	1.71
CSQA	1221	60	2.52
IWSLT	890	45	1.92
SamSum	819	40	2.23

were instructed to evaluate the influence of the four rules of the HPT on solving the annotated tasks using each prompting strategy, rating their influence as High (H), Moderate (M), or Low (L). It's important to note that a high rating on rule 4 has a greater influence than a high rating on rule 3, and similarly for the other two rules. Considering the rating as shown in Table 5 and varying influences of these rules, five prompting strategies that prioritize comprehensive coverage of cognitive demands while ensuring the set optimally widens the variation across complexity levels were selected for populating the HPF.

Prompting Strategy	Rule 1	Rule 2	Rule 3	Rule 4
Role Prompting	L	L	L	L
Emotion Prompting	L	L	M	L
Zero-shot CoT	L	L	M	L
Meta Prompting	M	H	M	L
Three-shot CoT	H	H	M	L
Five-shot CoT	H	H	H	L
Chain-of-Verification	H	H	H	H
Least-to-Most Prompting	H	H	H	L
Self-Consistency	H	H	H	M
GKP	L	H	H	H

Table 5: Human judgment of influence of the rules of taxonomy on different prompting strategies in solving the tasks of the representative set. The ratings are provided based on a voting system involving all human annotators. Green represents the prompting strategies selected for populating the complexity levels of the HPF.

B LLM-as-a-Judge

B.1 Scoring Prompt Template

The system prompt is designed to guide the LLM judge in evaluating different prompting strategies based on four specific criteria: Basic Recall and Reproduction, Understanding and Interpretation, Analysis and Reasoning, and Application of Knowledge and Execution. Each criterion is scored on a scale of 1-5. The evaluation uses GPT-4o as a judge, with the following system prompt:

```
You are a judge evaluating different prompting strategies
and you need to score these prompting strategies based
on pre-defined criteria. Different prompting strategies
leverage varied amounts of intelligence from the model
to achieve the required answer. So, assign the scores
very carefully based on your analysis of the prompt and
its effect on your intelligence to achieve the given
answer as well as the number of multi-step prompts which
increases the complexity of execution.

score1: Basic Recall and Reproduction
Definition: The need of the model to remember and
reproduce factual information without interpretation
or analysis to answer the prompt
Range: 1-5

score2: Understanding and Interpretation
Definition: The need of the model to comprehend and
explain the meaning of information, summarizing or
clarifying content to answer the prompt
Range: 1-5

score3: Analysis and Reasoning
Definition: The need for the model to break down complex
information, understand relationships, and solve problems
using logical reasoning to answer the prompt
Range: 1-5

score4: Application of Knowledge and Execution
Definition: The need for the model to apply knowledge
in practical situations, execute multi-step processes,
and solve complex tasks to answer the prompt
Range: 1-5
```

B.2 Hybrid Dataset

The hybrid dataset is composed of 1106 samples uniformly distributed over seven different task-specific datasets, covering a wide range of language understanding and generation tasks. This diversity allows for a comprehensive evaluation of the prompting strategies across various problem types. The evaluation uses a hybrid dataset composed of samples from various task-specific datasets and each dataset contributes specific types of tasks:

(1) MMLU (Massive Multitask Language Understanding)
(2) HumanEval (Code Generation and Completion)
(3) GSM8K (Grade School Math 8K)
(4) BoolQ (Boolean Questions)
(5) CSQA (Commonsense Question Answering)
(6) IWSLT (International Workshop on Spoken Language Translation)

(7) SamSum (Dialogue Summarization)

B.3 Scoring Method

For each prompting strategy (Role Prompting, Zero-shot CoT, Three-shot CoT, Least to Most Prompting, Generated Knowledge Prompting), the system:

(1) Applies the prompting strategy to each sample in the hybrid dataset
(2) Generates an answer using GPT-4o
(3) Presents the prompt, generated answer, and correct answer to the LLM judge
(4) Collects scores for each of the four criteria and the system calculates average scores for each criterion across all tasks and datasets.

This study ensured that both the human judge and the LLM judge utilized the same scoring methodology to eliminate any potential bias in the comparison.

C Hallucination in Adaptive HPF

Hallucinations in *prompt-selector* refer to instances where the LLM generates incorrect or misleading prompting levels or nonsensical information that is not supported by the HPF. These hallucinations can occur across various tasks, including question answering, multiple-choice questions, translation, and summarization.

For the BoolQ task, the *prompt-selector* initially struggles, indicated by the iterations where it reaches Level 4 with hallucinations. However, by the fourth iteration, Llama-3 8B manages to answer correctly at Level 2. For the CSQA task, *prompt-selector* exhibits hallucinations initially, shown by Level 4 and Level 0 (not included in HPF) responses. Eventually, it corrects itself by the third iteration, providing the correct answer at Level 2. For the IWSLT task, *prompt-selector* demonstrates a consistent pattern of hallucinations across multiple iterations. Even though Llama-3 8B attempts the translation at Level 2 multiple times, it ultimately fails to provide a correct translation, indicating a persistent hallucination. For the SamSum task, *prompt-selector* shows initial hallucinations in the first three iterations (Level 4). However, by the fourth and fifth iterations, the *prompt-selector* starts producing lower levels. Finally, Llama-3 8B achieves the correct answer at Level 2 in the last iteration .

The results in Table 6 and Table 7 indicate that the *prompt-selector* exhibits hallucinations in selecting complexity levels across various tasks and iterations resulting in higher HPI for Adaptive HPF, with performance varying significantly. While the LLM can eventually produce correct answers, as seen in the BoolQ and SamSum tasks, it often requires multiple attempts and may still fail in tasks like IWSLT translation.

C.1 Prompt Template for Prompt-Selector

The *prompt-selector* in adaptive HPF selects the prompting level based on the task complexity to address the task. Llama-3 8B serves as the *prompt-selector* in the experiments. The prompt template was meticulously designed to ensure maximum clarity, aiming to reduce hallucinations and select the most effective prompting strategy.

Prompt Template: Choose the most effective prompting strategy among five available strategies for the task. Begin with the lowest indexed strategy and progress to higher indexed strategies if the earlier ones are not effective. For a given task, the prompting strategies are:

- **Role Prompting:** Defines a role for the model in solving the task.
- **Zero-shot Chain of Thought prompting:** Stimulates reasoning and problem-solving by including the phrase 'Let's think step by step' without offering previous examples related to the task.
- **Three-shot Chain of Thought prompting:** Offers three examples related to the task to guide the model's reasoning process.
- **Least-to-most prompting:** Uses a sequential method to derive essential insights from the task to solve it.
- **Generated Knowledge Prompting:** Integration and application of external knowledge to accomplish the task. The external knowledge is generated using some other model based on the task.

Select only the index (do not provide the name) of the most effective prompting strategy.

D Computational Budget

All evaluation experiments and ablation studies were conducted on V100 GPUs (16GB and 32GB variants), utilizing a total of around 9,000 computation hours, this equates to approximately 1.125 petaflop-hours of computational resources.

E Large Language Models Used for Evaluation

The HPF supports leading open source and proprietary LLMs and includes mechanisms for optimizing performance through advanced quantization techniques. The experiments were conducted on the following instruction-tuned LLMs, and the model description and licenses are discussed in Table 8.

The LLMs were loaded in 4-bit precision format, with a maximum generation limit of 1024 tokens per run to ensure concise outputs. The temperature was set to 0.6 to control prediction randomness, while top-p sampling (p=0.9) enabled the exploration of diverse continuations. Additionally, a repetition penalty was applied to discourage the generation of repeated phrases, promoting coherent and varied text output.

F Prompt Templates

F.1 Level 1: Role Prompting

Role prompting represents the most basic interaction with an LLM, assigning it a specific role or task without additional context or examples. This approach relies solely on the initial instruction to guide responses. For instance, asking the LLM to *"act as a translator"* prompts it to translate text based on its training data. While straightforward, this method may lack depth, resulting in less accurate or nuanced outputs. Table 9 shows the prompt templates used for role prompting across all datasets in the experiments.

Table 6: HPI (lower is better) of LLMs across datasets (with thresholds) for Adaptive HPF.

Model	BoolQ	CSQA	IWSLT (0.15)	IWSLT (0.20)	SamSum (0.15)	SamSum (0.20)
Llama-3 8B	5.2173	5.9136	**6.2006**	**6.2841**	5.0316	5.5756
Mistra 7B	**5.0483**	5.9073	6.2478	6.4604	**4.7423**	**5.1336**
Phi-3 3.8B	5.1386	**5.6793**	6.3955	6.4936	5.0961	5.7778
Gemma 7B	5.1514	5.5771	6.5947	6.6605	5.7229	6.4347

Table 7: Performance scores of LLMs across datasetsfor Adaptive HPF.

Dataset	Metric	Threshold	Llama-3 8B	Phi-3 3.8B	Mistral 7B	Gemma 7B
BoolQ	Accuracy	-	0.88577	0.91115	**0.91752**	0.91166
CSQA	Accuracy	-	0.59451	0.68019	0.60111	**0.68549**
IWSLT	BLEU	0.15	**0.21140**	0.15557	0.20000	0.08447
		0.2	**0.21146**	0.15354	0.20568	0.07730
SamSum	ROUGE-1	0.15	0.24407	0.20586	**0.26910**	0.16023
		0.2	0.24981	0.21580	**0.28335**	0.16179

Table 8: License information for LLMs used in the experiments.

Model	License Type	Usage Restrictions
GPT-4o	Proprietary	Commercial use requires paid API access, subject to OpenAI's terms of service
Claude 3.5 Sonnet	Proprietary	Commercial use requires paid API access, subject to Anthropic's terms of service
Mistral-Nemo 12B	Proprietary	Usage likely restricted to authorized partners or specific use cases
Gemma-2 9B	Research License	Non-commercial use only, research purposes
Llama-3 8B	Research License	Specific restrictions may apply, typically for non-commercial research use
Mistral 7B	Open-source	Broad use allowed, must include original license and notices
Gemma 7B	Open-source/Research	Depending on the license, may have non-commercial restrictions or broad use allowed
Phi-3 3.8B	Open-source	Broad use allowed, must include original license and notices

Table 9: Prompt templates of different datasets for Role Prompting.

Dataset	Prompt
BoolQ	`Based on the passage: "passage", answer True/False to the question: "question" as an Omniscient person.`
CSQA	`Choose the answer: "question",A. "option 1",B. "option 2",C. "option 3",D. "option 4",E. "option 5" as a critical thinker.`
IWSLT	`Translate "english text" to french as a Translator.`
SamSum	`Summarize the Dialogue: "dialogue" as a Summarizer.`
GSM8k	`Based on the question: "question", calculate the numerical answer to the question as an expert mathematician.`
HumanEval	`Complete the given code based on the mentioned constraints: "code" as an expert programmer.`
MMLU	`Choose the answer: "question",A. "option 1",B. "option 2",C. "option 3",D. "option 4" as a critical thinker.`

F.2 Level 2: Zero-shot Chain-of-Thought Prompting

Zero-shot Chain-of-Thought (CoT) prompting enhances basic role prompting by requiring the LLM to generate a reasoning process for a task, despite not being explicitly trained on similar examples. This method encourages the LLM to break down the problem and solve it step-by-step using its internal knowledge, improving response quality through logical progression and coherence. Table 10 displays the prompt templates used for Zero-CoT across all datasets in the experiments.

F.3 Level 3: Three-Shot Chain-of-Thought Prompting

Three-shot Chain-of-Thought (CoT) prompting builds on the zero-shot approach by providing the LLM with three task examples, including the reasoning steps used to reach the solution. These examples help the LLM grasp the required structure and logic, enabling it to better replicate the problem-solving process and produce more accurate, contextually relevant responses. Table 11 shows the prompt templates used for 3-CoT across all datasets in the experiments.

F.4 Level 4: Least-to-Most Prompting

Least-to-most prompting is an advanced technique that gradually increases prompt complexity, starting with simpler tasks and progressing to more complex challenges. This method allows the LLM to build confidence and leverage insights from easier prompts to tackle harder ones, enhancing its ability to generalize from straightforward examples to intricate scenarios. Table 12 displays the prompt templates used for Least-to-Most Prompting across all datasets in the experiments.

F.5 Level 5: Generated Knowledge Prompting

Generated Knowledge prompting is one of the most complex techniques in HPF, where the LLM not only addresses the task but also integrates relevant additional information to enhance its response. This method prompts another LLM to produce auxiliary knowledge, creating a richer context for understanding and solving the problem. By leveraging self-generated insights, the LLM can deliver more detailed, accurate, and nuanced answers. Table 13 shows the prompt templates used for Generated Knowledge Prompting across all datasets in the experiments.

Table 10: Prompt templates of different datasets for Zero-shot Chain-of-Thought Prompting.

Dataset	Prompt
BoolQ	Based on the passage: “**passage**”, answer True/False to the question: “**question**”. Let’s think step by step.
CSQA	Choose the answer: A. “**option 1**”,B. “**option 2**”,C. “**option 3**”,D. “**option 4**”,E. “**option 5**”. Let’s think step by step.
IWSLT	Translate “**english text**” to french. Let’s think step by step.
SamSum	Summarize the Dialogue: “**dialogue**”. Let’s think step by step.
GSM8k	Based on the question: "**question**", calculate the numerical answer to the question. Let’s think step by step.
HumanEval	Complete the given code based on the mentioned constraints: "**code**". Let’s think step by step.
MMLU	Choose the answer: “**question**”,A. “**option 1**”,B. “**option 2**”,C. “**option 3**”,D. “**option 4**”. Let’s think step by step.

Table 11: Prompt templates of different datasets for Three-Shot Chain-of-Thought Prompting.

Dataset	Prompt
BoolQ	Based on the passage: "**passage1**", answer True/False to the question: "**question1**". Answer: "**answer1**". Explanation: "**explaination1**". Based on the passage: "**passage2**", Answer True/False to the question: "**question2**". Answer: "**answer2**". Explanation: "**explaination2**". Based on the passage: "**passage3**", Answer True/False to the question: "**question3**". Answer: "**answer3**". Explanation: "**explaination3**". Based on the passage: "**passage**", Answer True/False to the question: "**question**".
CSQA	Choose the answer: "**question1**",A. "**option1-1**",B. "**option2-1**",C. "**option3-1**",D. "**option4-1**",E. "**option5-1**", Answer: "**answer1**", Explanation: "**explaination1**". Choose the answer: "**question2**", A. "**option1-2**",B. "**option2-2**",C. "**option3-2**",D. "**option4-2**",E. "**option5-2**", Answer: "**answer2**", Explanation: "**explainatio n2**". Choose the answer: "**question3**", A. "**option1-3**",B. "**option2-3**",C. "**option3-3**",D. "**option4-3**",E. "**option5-3**", Answer: "**answer3**", Explanation: "**explaination3**". Choose the answer: "**question**", "**question**",A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**",E. "**option 5**".
IWSLT	Translate "**english text1**" to French. French: "**french text1**". Translate "**english text2**" to French. French: "**french text2**". Translate "**english text3**" to French. French: "**french text3**". Translate "**english text**" to French.
SamSum	Summarize the Dialogue: "**dialogue1**". Summary: "**summary1**". Summarize the Dialogue: "**dialogue2**". Summary: "**summary2**". Summarize the Dialogue: "**dialogue3**". Summary: "**summary3**". Summarize the Dialogue: "**dialogue**".
GSM8k	Based on the question:"**gsm8k_question1**", calculate the numerical answer to the question.Answer: "**gsm8k_ans1**".Based on the question:"**gsm8k_question2**", calculate the numerical answer to the question.Answer: "**gsm8k_ans2**". Based on the question:"**gsm8k_question3**", calculate the numerical answer to the question.Answer: "**gsm8k_ans3**". Based on the question: "**question**", calculate the numerical answer to the question.
HumanEval	Complete the given code based on the mentioned constraints: "**humaneval_code1**", Code: "**humaneval_sol1**". Complete the given code based on the mentioned constraints: "**humaneval_code2**", Code: "**humaneval_sol1**".Complete the given "**code**" based on the mentioned constraints: "**humaneval_code3**", Code: "**humaneval_sol3**".
MMLU	Choose the answer for the question: "**mmlu_ques1**" A. [AND, NOT] B. [NOT, OR] C. [AND, OR] D. [NAND] Answer: C. Explanation: "**mmlu_exp1**". Choose the answer for the question "**mmlu_ques2**" A. The defendant's statement was involuntary. B. The defendant's statement was voluntary. C. The defendant was not in custody when the statement was made. D. The statement was not made in response to a known police interrogation. Answer: A, Explanation: "**mmlu_exp2**". Choose the answer for the question: "**mmlu_ques3**" . A. Wrong, Wrong. B. Wrong, Not wrong C. Not wrong, Wrong D. Not wrong, Not wrong. Answer: B Explanation: "**mmlu_exp3**". Choose the answer."**question**" "**question**",A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**".

Table 12: Prompt templates of different datasets for Least-to-Most Prompting.

Dataset	Prompt
BoolQ	**prompt 1**: Summarize the main points of this passage: "**passage**". **prompt 2**: Analyze this question to identify its key components: "**question**". **prompt 3**: Find the part of the passage that relates to this question: "**question**", Passage: "**passage**". **prompt 4**: Based on the passage, what is the answer to this question: "**question**", Relevant Information: "**previous response**".
CSQA	**prompt 1**: Analyze this question: "**question**". **prompt 2**: Elaborate about each option for the question: "**question**", options: A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**",E. "**option 5**". **prompt 3**: Based on the analysis: "**previous response**", discard wrong answers among the options: A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**",E. "**option 5**". **prompt 4**: Choose the correct answer from the options: A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**",E. "**option 5**".
IWSLT	**prompt 1**: What is the main idea or theme of this text? "**english text**". **prompt 2**: Identify and list the key phrases or terms in this text: "**english text**". **prompt 3**: Translate the following key phrases into French: "**previous response**". **prompt 4**: Translate "**english text**" into French, incorporating the translations of the key phrases: "**previous response**".
SamSum	**prompt 1**: List the main points or key ideas present in this dialogue: "**dialogue**". **prompt 2**: Elaborate on the following key points, providing additional details or context: "**previous response**". **prompt 3**: Using the listed key points and their elaborations, draft a concise summary of this text: "**dialogue**". **prompt 4**: Refine this draft summary to make it more concise and coherent, ensuring it captures the essence of the text: "**dialogue**".
GSM8k	Analyze the question: "**question**". Break the question into sub-problems: "**question**". Calculate answers for the subproblems of the question: "**pred**". Calculate the numerical answer to this question: "**question**" based on the previous calculations: "**pred**"
HumanEval	Analyze the code: "**code**". Break the question into sub-problems: "**code**". Complete code for the subproblems of the code: "**pred**". Complete the code based on the mentioned constraints: "**code**" based on the previous calculations: "**pred**"
MMLU	Analyze the question: "**question**". Elaborate about each option for the question: "**question**", options: A. "**option 1**" B. "**option 2**" C. "**option 3**" D. "**option 4**". Based on the analysis: "**question**", Discard wrong answers among the options: A. "**option 1**" B. "**option 2**" C. "**option 3**" D. "**option 4**".

Table 13: Prompt templates of different datasets for Generated Knowledge Prompting.

Dataset	Prompt
BoolQ	**inference prompt**: Based on the passage:"**passage**", answer True/False to the question: '**question**' using knowledge of the passage:"**knowledge**" **knowledge generation prompt**: Generate Knowledge about the passage: "**passage**".
CSQA	**inference prompt**: Choose the answer:"**question**", A. "**option 1**",B. "**option 2**",C. "**option 3**",D. "**option 4**",E. "**option 5**" using knowledge of the question:"**knowledge**" **knowledge generation prompt**: Generate Knowledge about the question: "**question**".
IWSLT	**inference prompt**: Translate "**english text**": to French using definitions of the keywords:"**knowledge**" **knowledge generation prompt**: Generate definitions in french of each word in the text: "**english text**".
SamSum	**inference prompt**: Summarize the Dialogue: "**dialogue**" using the interpretation of the dialogue:"**knowledge**" **knowledge generation prompt**: Generate interpretation about the dialogue: "**dialogue**".
GSM8k	Based on the question:"**question**", calculate the numerical answer to the question using an interpretation of the question:"**pred**"
HumanEval	Complete the code based on the mentioned constraints:"**code**" using knowledge of the constraints:"**pred**"
MMLU	Choose the answer. "**question**", options: A. "**option 1**" B. "**option 2**" C. "**option 3**" D. "**option 4**" using knowledge of the question:"**pred**"