

# Large Language Model Evaluation: From Principles to Applications

LI, Yanyang

A Thesis Submitted in Partial Fulfillment  
of the Requirements for the Degree of  
Doctor of Philosophy  
in  
Computer Science and Engineering

The Chinese University of Hong Kong

August 2025

Thesis Assessment Committee

Professor Farzan FARNIA (Chair)

Professor WANG Liwei (Thesis Supervisor)

Professor LYU Rung Tsong Michael (Thesis Co-supervisor)

Professor DOU Qi (Committee Member)

Professor SONG Yangqiu (External Examiner)

# Abstract

In recent years, the field of natural language processing (NLP) has undergone a significant paradigm shift: researchers no longer train models from scratch but instead leverage large language models (LLMs) to directly perform downstream tasks under few-shot settings. To keep pace with the rapid advancements in LLM performance and to better understand their capabilities, numerous evaluation benchmarks have been proposed. However, this proliferation of benchmarks introduces new challenges: How can researchers develop a comprehensive understanding of the boundaries of model capabilities across so many benchmarks? How can they ensure the reliability of evaluation results?

This dissertation addresses these questions by exploring two key principles for evaluation benchmark design: comprehensiveness and robustness. Comprehensiveness requires benchmarks to provide a holistic assessment of different aspects of model capabilities, while robustness emphasizes that evaluation results should remain stable and unaffected by extraneous factors unrelated to the content being assessed. Based on these principles, this dissertation first introduces CLEVA, a comprehensive evaluation platform for Chinese LLMs. CLEVA is designed with an evaluation framework that spans 31 tasks and 9 major metric dimensions, providing a multifaceted assessment of Chinese LLMs. Building on this, the dissertation presents C<sup>2</sup>LEVA, which enhances the robustness of evaluation results within the comprehensive framework of CLEVA. C<sup>2</sup>LEVA systematically addresses three common factors that affect evaluation stability: format, language, and data contamination.

Furthermore, this dissertation demonstrates how insights gained from evaluation can be translated into practical methodologies through three case studies from different research

domains. In the task of knowledge-grounded conversation, it is observed that LLMs have acquired commonsense and fact summarization abilities from pretraining. This insight inspired the design of PLATO-KAG+, a dialogue system that leverages LLMs as a knowledge base. In multi-hop reasoning, the research finds that the presence of irrelevant information in the context significantly impairs the reasoning capabilities of LLMs. To address this, reasoning with attribution is proposed, enabling LLMs to filter irrelevant information by generating citations to relevant text. Finally, in test-time scaling, traditional methods either lack the ability to effectively retain prior experiences or suffer from low sampling efficiency. To tackle this, the dissertation proposes FTTT, a method that reframes feedback utilization as a test-time training problem, achieving both efficient decoding and the use of model parameters as memory.

# 摘要

近年来，自然语言处理领域经历了深刻的范式变革：研究人员不再从零开始训练模型，而是通过大语言模型在少样本条件下直接完成下游任务。为适应大语言模型性能的快速提升，并深入理解其能力，大量评测基准应运而生。然而，这也带来了新的挑战：研究人员如何在众多基准中构建对模型能力边界的全面认知？如何确保评测结果的可信性？

本论文围绕评测基准设计中的两个核心原则展开探讨：全面性与健壮性。全面性要求评测基准能够全面反映模型的多方面能力；健壮性则强调评测结果的稳定性，不应受到与评测内容无关因素的干扰。基于此，本论文首先提出了CLEVA，一个面向中文大语言模型的综合评测平台。CLEVA设计了涵盖31个任务、9大能力维度的评测体系，从各个角度评估中文大语言模型的表现。在此基础上，本论文进一步提出了C<sup>2</sup>LEVA，其在CLEVA的全面性评测框架上强化了健壮性。C<sup>2</sup>LEVA针对评测结果稳定性面临的三大关键挑战——形式、语言、以及数据污染——提出了系统性解决方案。

此外，本论文通过三个研究领域的实际应用，展示了如何将评测中获得的洞见转化为创新方法。在基于知识的对话生成任务中，研究表明大语言模型已从预训练中掌握了常识与事实总结能力。这一发现促成了PLATO-KAG+的设计，一个将大语言模型作为知识库的对话系统。在多跳推理任务中，研究发现无关信息的增加会显著削弱大语言模型的推理效果。为此，论文提出了基于归因的推理方法，通过生成相关文本的引用过滤上下文中的无关信息。在测试时扩展任务中，传统方法要么缺乏对历史经验的有效记忆，要么采样效率低下。对此，论文提出了FTTT方法，将利用测试时反馈的问题转化为测试时训练任务，实现了既能利用模型参数作为记忆，又具备更高解码效率的解决方案。

# Contents

Abstract . . . . .	iii
Acknowledgments . . . . .	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	4
1.2 Thesis Outline . . . . .	5
<b>2 Literature Review</b>	<b>7</b>
2.1 LLM Evaluation . . . . .	7
2.2 Data Contamination . . . . .	8
2.3 Knowledge-Grounded Conversation . . . . .	10
2.4 Long-Context Reasoning . . . . .	11
2.5 Test-Time Scaling . . . . .	12
<b>3 Comprehensive LLM Evaluation Benchmark: CLEVA</b>	<b>14</b>
3.1 Motivation . . . . .	14
3.2 Preliminaries . . . . .	17
3.3 System Design . . . . .	17
3.3.1 Evaluation Taxonomy . . . . .	18
3.3.2 Leaderboard & Data Contamination . . . . .	22
3.4 Usage Example . . . . .	22
3.5 Evaluation . . . . .	23
3.6 Summary . . . . .	24
<b>4 Robust LLM Evaluation Benchmark: C<sup>2</sup>LEVA</b>	<b>26</b>
4.1 Motivation . . . . .	26
4.2 C <sup>2</sup> LEVA . . . . .	28
4.2.1 Contamination Prevention Overview . . . . .	29
4.2.2 C <sup>2</sup> LEVA Task Taxonomy . . . . .	30
4.2.3 The Solution to Contamination Prevention . . . . .	31
4.3 Experimental Results . . . . .	36

4.3.1	Setup . . . . .	36
4.3.2	Main Results . . . . .	37
4.3.3	Analysis . . . . .	38
4.4	Summary . . . . .	40
<b>5</b>	<b>Applications of LLM Evaluation: Knowledge Grounded Conversation</b>	<b>41</b>
5.1	Motivation . . . . .	41
5.2	Eliciting Knowledge from Large Models . . . . .	42
5.2.1	Methods and Models . . . . .	43
5.2.2	Annotation Tagset . . . . .	45
5.3	Exploiting Generated Knowledge for Conversation . . . . .	46
5.3.1	PLATO-KAG . . . . .	46
5.3.2	Posterior-based Reweighing . . . . .	47
5.3.3	Noisy Training . . . . .	48
5.4	Experimental Setup . . . . .	49
5.4.1	Datasets . . . . .	49
5.4.2	Evaluation Metrics . . . . .	49
5.4.3	Response Generation Baselines . . . . .	50
5.4.4	Implementation Details . . . . .	51
5.5	Results and Analysis . . . . .	53
5.5.1	Knowledge Generation Results . . . . .	53
5.5.2	Response Generation Results . . . . .	55
5.6	Summary . . . . .	57
<b>6</b>	<b>Applications of LLM Evaluation: Multi-Hop Reasoning</b>	<b>58</b>
6.1	Motivation . . . . .	58
6.2	Reasoning with Attributions . . . . .	60
6.2.1	Pilot Study . . . . .	60
6.2.2	Dataset Curation . . . . .	62
6.3	Learning to Attribute in Reasoning . . . . .	64
6.4	Experiments . . . . .	66
6.4.1	Datasets . . . . .	66
6.4.2	Models . . . . .	66
6.4.3	Main Results . . . . .	67
6.4.4	Analysis . . . . .	70
6.5	Summary . . . . .	73

<b>7</b>	<b>Applications of LLM Evaluation: Test-Time Scaling</b>	<b>74</b>
7.1	Motivation . . . . .	74
7.2	Feedback-based Test-Time Training . . . . .	76
7.2.1	The Test-Time Training Task . . . . .	76
7.2.2	Self-Reflected Feedback . . . . .	77
7.3	A Learnable Test-Time Optimizer . . . . .	79
7.3.1	The Learning to Optimize Problem . . . . .	79
7.3.2	A Parameter-Efficient Architecture in The Gradient Space . . . . .	80
7.4	Experiments . . . . .	82
7.4.1	Setup . . . . .	82
7.4.2	Training-Free Results . . . . .	84
7.4.3	Fine-Tuning Results . . . . .	85
7.4.4	Analysis . . . . .	87
7.5	Summary . . . . .	87
<b>8</b>	<b>Conclusion</b>	<b>89</b>
8.1	Produced Publications . . . . .	90
	<b>References</b>	<b>92</b>

## List of Tables

4.1	Comparison between passive and active prevention. . . . .	30
4.2	Summary of contamination prevention strategy adopted in each task of C <sup>2</sup> LEVA. . . . .	32
4.3	Evaluation result distortion of data watermarking (Wei et al., 2024) in the fact completion task. <b>Before</b> is the results before applying data watermarking and <b>After</b> is the results after applying data watermarking. $\Delta$ indicates the performance change, where the performance loss is marked in <span style="background-color: #f08080;">red</span> and <span style="background-color: #90ee90;">green</span> for the performance gain. . . . .	39
5.1	The tagset developed to evaluate the quality of the generated knowledge by human annotators. . . . .	44
5.2	The automatic evaluation result (Unigram F1) of two tuning methods for generating knowledge in WoW seen and unseen test sets (“N/A” means we do not use the pre-trained weights). Results of fine-tuned T5-XXL are missing due to resource constraint. . . . .	51
5.3	The automatic and human evaluation results on WoW seen (upper) and unseen (bottom) test sets. * means this is our implementation results. + means our proposed method is applied. Note that PPL is generally not comparable among baselines, as their vocabularies are different. The best results are in <b>bold</b> and the best baseline results are <u>underlined</u> . . . . .	54
5.4	The evaluation results on Holl-E test set. . . . .	56
5.5	The ablation study on WoW test sets. . . . .	56
6.1	An example of CoT and two of our reasoning with attribution methods: CoC and CoQ. We highlight the differences between these methods, e.g., answers are marked in <span style="color: green;">green</span> , citations are marked in <span style="color: orange;">orange</span> and quotes are marked in <span style="color: blue;">blue</span> . . . . .	61
6.2	Exact-Match (EM) and F1 scores of ChatGPT and Claude-instant with 5-shot prompting on multi-hop reasoning datasets, e.g., MuSiQue, 2WikiMulti-HopQA (2Wiki for short) and HotpotQA. The best results are in <b>bold</b> . AO means models predict answers only. . . . .	62
6.3	Incidence rates of different error types. . . . .	63

6.4	Statistics of MuSiQue-Attribute. . . . .	63
6.5	Exact-Match (EM) results on three multi-hop reasoning datasets. The best small-scale long-context LM results are in <b>bold</b> and the best baseline results are <u>underlined</u> . . . . .	68
6.6	Results on general instruction-following benchmarks. “+ Alpaca Data” is a Vicuna-7B model continued fine-tuning on Alpaca data. . . . .	69
6.7	Ablation study on multi-task learning. . . . .	71
6.8	Ablation study on data augmentation. . . . .	71
6.9	Two examples from MuSiQue test set where Vicuna fails and AttrLoRA succeeds. We manually annotate quotes in <b>blue</b> and citations in <b>orange</b> . Correct predictions are marked by <b>✓</b> and incorrect ones are by <b>✗</b> . . . . .	73
7.1	Comparing the advantages and drawbacks of FTTT and related works. . . . .	79
7.2	Experimental results on four datasets with a budget of 32. For stochastic algorithms, we report the mean of three runs with different random seeds and standard deviation in the subscript. <b>Bold</b> entries are the best results, and <u>underlined</u> entries are the second-best results. . . . .	83
7.3	Fine-tuning results on four datasets with a budget of 32. #Param. denotes the number of trainable parameters. We report the mean of three runs with different random seeds and standard deviation in the subscript. <b>Bold</b> entries are the best results. . . . .	85
7.4	The ablation study of OPTUNE. We report results of Llama-3.1-8B-Instruct on MBPP with a budget of 2. . . . .	87
7.5	Two GSM8K examples with model predictions. <b>✓</b> means correct and <b>✗</b> means incorrect. Sentences marked in <b>orange</b> are incorrect reasoning steps and those marked in <b>blue</b> are the correct counterparts. . . . .	88

## List of Figures

3.1	CLEVA benchmark. . . . .	18
3.2	The mean win rate of 23 models in 31 tasks. The mean win rate is the probability of a model outperforming a random different model on a random task. We exclude toxicity, privacy, and efficiency metrics as all models excel in the former two, and the latter is often paired with other metrics to deliver meaningful comparisons. Since robustness and fairness involve expensive data augmentation, we only evaluate ChatGPT and Claude-instant. . . . .	23
4.1	Contamination prevention overview. Solid lines indicate how data flows within a machine learning model development pipeline. The dotted line indicates where the test data leaks into the training data. . . . .	28
4.2	The task taxonomy of C <sup>2</sup> LEVA. . . . .	31
4.3	The framework of C <sup>2</sup> LEVA for contamination prevention. . . . .	33
4.4	The mean win rate of 15 models in 22 tasks of C <sup>2</sup> LEVA. . . . .	34
4.5	The accuracy standard deviation of different models in different prompt templates from different tasks. . . . .	35
4.6	The English mean win rate of C <sup>2</sup> LEVA scales linearly with style-controlled Chatbot Arena Elo. ● is the observed value. — indicates the linear fit. $\rho$ and $e$ denote the Spearman’s ranking correlation and the root mean square error of the linear fit respectively. . . . .	36
4.7	Mean win rate comparison among models in different task groups. We choose the top-5 best-performing models in each language for visualization. . . . .	37
5.1	Comparison of fine-tuning (Devlin et al., 2019) and prefix-tuning (Li and Liang, 2021). . . . .	43
5.2	Illustration of PLATO-KAG (Huang et al., 2021). . . . .	46
5.3	The human evaluation results of generated knowledge from fine-tuned DialoGPT-large on WoW seen and unseen test sets. . . . .	52
6.1	Comparison of the proposed auxiliary tasks. . . . .	65

6.2	Exact-Match (EM) results of different models under various noise levels in three multi-hop reasoning datasets. Note that all models except our AttrLoRA use 5-shot prompting. A higher noise ratio indicates more distractors, i.e., irrelevant documents, are presented in the context of both the test instance and the demonstrations. . . . .	69
6.3	Multi-hop reasoning performance vs. citation precision and recall of AttrLoRA.	70
6.4	The impact of scaling fine-tuning data size. . . . .	72
7.1	Comparison between sequential revision, parallel sampling, and feedback-based test-time training. <span style="color: red;">■</span> is the failed attempt and <span style="color: green;">■</span> is the successful attempt. $\rightarrow$ indicates the LLM generation with the input on the left of the arrow and the output on the right. $--\rightarrow$ denotes the LLM training, where the left of the arrow is the training data. . . . .	75
7.2	The model architecture of OPTUNE. . . . .	81
7.3	The scaling trends of different methods under varying budgets. The colored area around the line denotes the standard deviation. The first row is the results of Llama-3.1-8B-Instruct and the second row is Mistral-7B-Instruct-v0.3. . . . .	84
7.4	The scaling trends of different fine-tuning methods under varying budgets. We report the mean results of three random trials. The first row is the results of Llama-3.1-8B-Instruct and the second row is Mistral-7B-Instruct-v0.3. . . . .	86

## Acknowledgments

I would like to express my heartfelt gratitude to my thesis committee members for their valuable and constructive feedback throughout my Ph.D. journey. I am especially indebted to my supervisors, Prof. Liwei Wang and Prof. Michael Lyu, for their unwavering guidance, encouragement, and mentorship. They have shown me how to pursue impactful and rigorous research, shaping both my academic career and personal growth.

I am also deeply thankful to my collaborators and co-authors, who have played an integral role in my research journey. Their insightful feedback, technical contributions, and tireless support—whether through writing code, running experiments, or sharing ideas—have been instrumental in publishing my work and advancing my understanding of the field.

To my family, I owe my deepest thanks. My mother and sister have always stood by me, offering unconditional love and encouragement. My fiancée has been my steadfast supporter, constantly believing in my potential and my ability to succeed.

Finally, I dedicate this work to the memory of my father, who sadly passed away during my Ph.D. I hope the contributions I have made to the field of artificial intelligence will honor his memory and make him proud. His love and guidance continue to inspire me, and I carry his legacy with me in all that I do.

# Chapter 1

## Introduction

Recent advances in large language models (LLMs) have revolutionized the field of Natural Language Processing (NLP). These models, equipped with billions of trainable parameters, are trained on massive web-scale, unlabeled datasets using self-supervision (Radford et al., 2018; Devlin et al., 2019). By leveraging such vast amounts of data, LLMs exhibit remarkable capabilities, including multi-tasking and emergent abilities (Wei et al., 2022a), such as in-context learning (Brown et al., 2020) and chain-of-thought reasoning (Wei et al., 2022b). These emergent properties enable LLMs to outperform prior state-of-the-art, task-specific models across a wide range of NLP tasks, including summarization, dialogue generation, and question answering (Liang et al., 2023; Bang et al., 2023), even when prompted with just a few examples at test time. Among the most prominent LLMs developed in recent years are OpenAI's GPT series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023), as well as their reasoning models (OpenAI, 2024, 2025). These models have achieved performance at or beyond the human level on a variety of complex tasks, showcasing their transformative potential.

Evaluation plays a pivotal role in the development and deployment of LLMs, serving as both a barometer of progress and a tool for identifying areas for further improvement. While LLMs have demonstrated immense potential, their black-box nature poses challenges for understanding their strengths, weaknesses, and overall behavior. In this context, evaluation

serves as a critical lens to uncover insights into these models and guide future research.

To better measure and understand LLM capabilities, a variety of benchmarks have been proposed. Comprehensive benchmarks like HELM (Liang et al., 2023) aim to provide broad coverage of LLM abilities, while domain-specific test sets such as MMLU (Hendrycks et al., 2021a) for disciplinary knowledge and GSM8K (Cobbe et al., 2021) for mathematical reasoning focus on in-depth evaluation of specific aspects.

In addition to new benchmarks, the methods used to evaluate LLMs have evolved significantly. Before the rise of LLMs, evaluation typically required fine-tuning or training models from scratch on downstream tasks. However, with the advent of LLMs and their in-context learning capability, where models can execute unseen tasks by being prompted with a few examples in their input, the few-shot evaluation setting has become the standard practice for assessing their performance. Below, we provide a few-shot prompting example for evaluating LLMs on machine translation:

Translate English to French:	→ <i>task description</i>
sea otter ⇒ loutre de mer	} → <i>few-shot examples</i>
peppermint ⇒ menthe poivrée	
lush girafe ⇒ girafe peluche	
cheese ⇒	→ <i>prompt</i>

Despite the significant progress in evaluating LLMs, practitioners often face challenges when working with existing benchmarks. Specifically, the abundance of diverse benchmarks can make it difficult to extract meaningful insights. Moreover, practitioners frequently question the reliability of evaluation results. For instance, widely used benchmarks are prone to data contamination (Brown et al., 2020), where test data overlaps with training data, leading to artificially inflated performance. At the same time, newly proposed benchmarks often lack rigorous validation, raising concerns about their trustworthiness. Finally, translating evaluation outcomes into actionable techniques remains a persistent hurdle.

To address these challenges and ensure the development of systematic and trustworthy benchmarks, this dissertation proposes two fundamental principles for benchmark design:

- **Comprehensiveness:** Benchmarks should provide informative and holistic insights into model performance across a wide range of tasks and metrics.
- **Robustness:** Benchmarks should be reliable and resistant to confounding factors, such as data contamination, to guide future research effectively.

Guided by these principles, we first introduce CLEVA (Li et al., 2023e), a comprehensive benchmark framework that systematically considers both task taxonomy and evaluation metrics. Building on this foundation, we present C<sup>2</sup>LEVA (Li et al., 2024b), a next-generation benchmark that enhances robustness by addressing critical factors such as question formats, multilingual capabilities, and data contamination.

To demonstrate how evaluation can drive technological advancement, this dissertation highlights three key applications where evaluation results informed the development of practical methods, including knowledge grounded conversation (Li et al., 2022), multi-hop reasoning (Li et al., 2024a), and test-time scaling (Li et al., 2025). Each of these works was initially inspired by findings from evaluation results on existing benchmarks. By identifying limitations or opportunities through systematic evaluation, new methods were developed to tackle these challenges or exploit the insights gleaned from the benchmarks.

**Knowledge Grounded Conversation** Recent advances in large-scale pre-training provide large models with the potential to learn knowledge from the raw text. It is thus natural to ask whether it is possible to leverage these large models as knowledge bases for downstream tasks. Here, we answer the aforementioned question in unsupervised knowledge-grounded conversation. We explore various methods that best elicit knowledge from large models. Our human study indicates that, though hallucinations exist, large models post the unique advantage of being able to output common sense and summarize facts that cannot be directly retrieved from the search engine. To better exploit such generated knowledge in dialogue generation, we treat the generated knowledge as a noisy knowledge source and propose the posterior-based reweighing as well as the noisy training strategy. Empirical results on two benchmarks show advantages over the state-of-the-art methods.

**Multi-Hop Reasoning** Recent advancements in long-context modeling have enhanced LLMs for complex tasks across multiple NLP applications. Despite this progress, we find that these models struggle with multi-hop reasoning and exhibit decreased performance in the presence of noisy contexts. To address this challenge, we introduce *Reasoning with Attributions*, a novel approach that prompts LMs to supply attributions for each assertion during their reasoning. We validate our approach through experiments on three multi-hop datasets, employing both proprietary and open-source models, and demonstrate its efficacy and resilience. Furthermore, we explore methods to augment reasoning capabilities via fine-tuning and offer an attribution-annotated dataset and a specialized training strategy. Our fine-tuned model achieves competitive performance on multi-hop reasoning benchmarks, closely paralleling proprietary LLMs such as ChatGPT and Claude-instant

**Test-Time Scaling** Solving complex tasks in a single attempt is challenging for LLMs. Iterative interaction with the environment and feedback is often required to achieve success, making effective feedback utilization a critical topic. Existing approaches either struggle with length generalization or rely on naive retries without leveraging prior information. We introduce FTTT, a novel paradigm that formulates feedback utilization as an optimization problem at test time. Additionally, we propose a learnable test-time optimizer, OPTUNE, to effectively exploit feedback. Experiments on two LLMs across four reasoning datasets demonstrate that FTTT and OPTUNE achieve superior scalability and performance.

## 1.1 Contributions

The contributions of this dissertation are summarized as follows:

- **Comprehensive LLM Evaluation:** We introduce CLEVA (Li et al., 2023e), a platform for evaluating Chinese LLMs. CLEVA features a holistic taxonomy covering 31 tasks and 9 metric dimensions. Using CLEVA, we conduct a large-scale evaluation of 23 Chinese LLMs, demonstrating its efficacy as a comprehensive evaluation framework.
- **Robust LLM Evaluation:** Building on CLEVA, we develop C<sup>2</sup>LEVA (Li et al., 2024b), an

evaluation benchmark designed with systematic data contamination prevention strategies to ensure trustworthy results. C<sup>2</sup>LEVA also includes prompt templates and bilingual test instances, enhancing robustness across diverse formats and languages.

- **Applications of LLM Evaluation:** We showcase the power of effective LLM evaluation in guiding future research in three fields:
  - **Knowledge-Grounded Conversation:** Our human evaluation reveals that LLMs effectively encode factual and commonsense knowledge in their parameters. Leveraging this insight, we develop PLATO-KAG+ (Li et al., 2022), a knowledge-grounded conversational system that uses an LLM as its knowledge base. PLATO-KAG+ is both efficient to train and highly explainable, thanks to the readability of the knowledge generated by the LLM.
  - **Multi-Hop Reasoning:** We identify that LLMs struggle with noisy information in long contexts during multi-hop reasoning. To address this, we propose reasoning with attribution, which encourages LLMs to generate citations within their chain-of-thought to improve denoising (Li et al., 2024a). Additionally, we collect supplementary attribution annotations for existing multi-hop reasoning datasets to support future research in this domain.
  - **Test-Time Scaling:** Through baseline evaluations, we observe that revision-based test-time scaling methods are memory-intensive but inefficient, while sampling-based methods are fast but fail to utilize feedback from prior attempts. To overcome these limitations, we propose Feedback-based Test-Time Training (FTTT) (Li et al., 2025), which formulates test-time feedback utilization as a training problem, embedding feedback into model parameters. We further introduce a learnable test-time optimizer, OPTUNE, to maximize the utility of feedback signals for FTTT.

## 1.2 Thesis Outline

The content of this dissertation is organized as follows:

- **Chapter 2** provides a complete review of related work for this dissertation, such as LLM

evaluation, data contamination, etc.

- **Chapters 3 and 4** cover the principles for delivering trustworthy and informative LLM evaluation outcomes.
  - **Chapter 3** discusses the comprehensiveness of LLM evaluation and presents CLEVA (Li et al., 2023e), a comprehensive LLM evaluation platform that offers a wide range of tasks and metrics that focus on various capabilities and aspects of LLMs.
  - **Chapter 4** discusses the robustness of LLM evaluation and presents C<sup>2</sup>LEVA (Li et al., 2024b), a bilingual LLM evaluation benchmark that explicitly addresses the concerns of data contamination while maintaining the broad coverage of evaluation tasks.
- **Chapters 5, 6 and 7** exemplify the applications of LLM evaluation in three different research fields, including knowledge-grounded conversation, multi-hop reasoning, and test-time scaling.
  - In **Chapter 5**, our human evaluation implies that LLMs possess vast factual knowledge. This finding motivates the development of PLATO-KAG+ (Li et al., 2022), which treats LLMs as knowledge bases for knowledge-grounded conversation.
  - In **Chapter 6**, our preliminary experiments show that LLMs struggle with multi-hop reasoning when noise appears in their context. We thus propose attribution-based reasoning (Li et al., 2024a), a strategy that aims to denoise during reasoning.
  - In **Chapter 7**, evaluation results of test-time scaling baselines reveal that existing approaches either do not utilize test-time feedback well or are inefficient. We present FTTT (Li et al., 2025), a novel paradigm that stores test-time feedback into model parameters for effective memorization and efficient inference. We also introduce OPTUNE, a learnable test-time optimizer that further improves performance.
- **Chapter 8** finally summarizes this dissertation and highlights the contributions.

## Chapter 2

# Literature Review

### 2.1 LLM Evaluation

LLM evaluation is a rapidly developing field in recent years to delineate the boundary of LLM’s capability. In English, various systematic evaluation benchmarks have been proposed. For example, BIG-Bench ([bench authors, 2023](#)) is the largest collection that covers more than 200 tasks. HELM ([Liang et al., 2023](#)) organizes tasks into core scenarios, which focus on use cases, and targeted evaluation, which aims to better understand models. HELM also presents a multi-metric measurement that enables analysis of tradeoffs for each scenario. Recently, AGIEval ([Zhong et al., 2023](#)) is proposed to evaluate LLMs using challenging human exams. PromptBench ([Zhu et al., 2023b](#)), on the other hand, measures the robustness of LLMs to prompts via adversarial attacks. MT-Bench ([Zheng et al., 2023b](#)) collects multi-turn questions and presents the Chatbot Arena platform that treats GPT-4 ([OpenAI, 2023](#)) as the judge.

While CLEVA shares the same fundamental motivation with HELM ([Liang et al., 2023](#)), to holistically evaluate language learning models in their original languages, CLEVA is far from a mere Chinese replica of HELM. Building on the foundation of HELM’s taxonomy, CLEVA introduces a range of tasks, with particular emphasis on those unique to Chinese, to better assess the capabilities of Chinese LLMs. It offers a new perspective on prompts,

providing abundant prompt templates to standardize evaluation and encourage in-depth exploration of models’ sensitivity. In terms of metrics, CLEVA expands into new areas of diversity and privacy for a more comprehensive evaluation. Finally, CLEVA proactively mitigates train-test contamination by collecting a significant amount of new data, creating unique test sets by sampling, and regularly updating the leaderboard. All of these evaluation designs are neatly packaged in a user-friendly platform to facilitate community usage.

There is also a lot of progress in evaluating Chinese LLMs (Huang et al., 2023; Liu et al., 2023a; Li et al., 2023c; Zhang et al., 2023b; Zeng, 2023). OpenCompass (Contributors, 2023b) and FlagEval (Contributors, 2023a) are two important attempts to evaluate Chinese LLMs. OpenCompass pools 53 public datasets and uses standard accuracy-like metrics as the only measurement for each dataset. FlagEval, with a smaller number of datasets and metrics, still needs further expansion to achieve sufficient coverage. Compared to previous efforts, CLEVA offers Chinese data from 84 datasets, including 33.98% original queries, while employing the broadest range of metrics to promote holistic evaluation. CLEVA standardizes prompts (§ 3.3) and mitigates data contamination issues, pioneering new paths for LLM evaluation in general.

## 2.2 Data Contamination

Existing work in data contamination can be divided into two main categories (Jacovi et al., 2023): *reactive* approaches aim to detect potential contamination risks in the evaluation results of existing benchmarks and models, while *preventative* approaches target preventing contamination before the evaluation.

**Reactive Contamination Detection.** Contamination detection is an application of membership inference attacks (Shokri et al., 2017; Yeom et al., 2018), which aim to determine whether an arbitrary sample is part of a given model’s training data. There are numerous works on contamination detection, typically based on various assumptions: if the training data is available, N-gram matching (Brown et al., 2020; Dodge et al., 2021) is the most popular

approach to report the contamination risk, despite its vulnerability to rephrasing (Yang et al., 2023b). If only a white-box model is available, most methods exploit token probabilities for accurate contamination detection (Shi et al., 2023b; Oren et al., 2023; Zhang et al., 2024). If we can only access a text-generation API, detection techniques like prompting (Golchin and Surdeanu, 2023) and synthetic data (Wei et al., 2023; Duarte et al., 2024) are proposed. In this work, we do not focus on proposing new detection methods but rather treat existing detection methods as a building block in preventative contamination mitigation.

**Preventative Contamination Mitigation.** Hiding the test set completely (Sun et al., 2023a) or partially (Li et al., 2023e) has been a common practice to prevent data contamination. However, this approach faces challenges due to the repurposing of test data and difficulties in maintenance (Jacovi et al., 2023). Recent methods seek to maintain the trustworthiness of evaluation results by continuously updating the data. Some work constructs new test cases from the latest web data (Li et al., 2024c; White et al., 2024). Specifically, for reasoning tasks that can be characterized by rules, various systems have been proposed to synthesize data for evaluation (Zhu et al., 2023a; Lei et al., 2023; Fan et al., 2023). However, these methods are vulnerable if previous test data is repurposed in the newly collected data, fail to guarantee the evaluation trustworthiness of scarce data, or are limited to a small number of tasks. Another line of work (Wang et al., 2024a; Ying et al., 2024; Qian et al., 2024) aims to generate new test data from existing test sets. However, the data quality is constrained by the performance of LLM assistants on those tasks.

Instead of creating new test cases, Jacovi et al. (2023) first propose avoiding unintentional contamination via licensing and encryption if the model developers cooperate. Recent works on copyrighted content protection shed light on alleviating contamination caused by users who intend to cheat on the benchmark (Wei et al., 2024; Zhao et al., 2024b). C<sup>2</sup>LEVA combines the best of both worlds: it not only renews the benchmark data with improved techniques but also employs data protection techniques to secure the released data. These two methods benefit each other: data protection prolongs the effectiveness of renewed scarce test data while new test cases ensure trustworthiness when the protection method is

compromised.

## 2.3 Knowledge-Grounded Conversation

**Knowledge-Grounded Conversation** The dialogue system field has witnessed a growing interest in knowledge-grounded conversation in recent years. Many related benchmarks have been proposed to study this problem [Zhang et al. \(2018\)](#); [Zhou et al. \(2018\)](#); [Dinan et al. \(2019\)](#); [Gopalakrishnan et al. \(2019\)](#); [Komeili et al. \(2022\)](#). Early work [Dinan et al. \(2019\)](#) had harnessed the annotated knowledge for training. Unsupervised approaches become attractive as acquiring these annotations is expensive. [Zhao et al. \(2020\)](#) use reinforcement learning to fine-tune GPT-2 [Radford et al. \(2019\)](#) for unsupervised knowledge selection. [Huang et al. \(2021\)](#) achieve a new state-of-the-art by selecting top-K knowledge when annotations are not available. Another line of research improves the knowledge selection modeling by estimating the posterior, which makes use of the future utterance. [Lian et al. \(2019\)](#) train the knowledge selector as a variational auto-encoder [Kingma and Welling \(2014\)](#). [Kim et al. \(2020\)](#) further model the knowledge selection in multi-turn dialogue as a sequential latent variable. More recently, dialogue model pre-training also attempts to involve knowledge for generating informative responses. [Shuster et al. \(2021\)](#) utilize the pre-trained retriever DPR [Karpukhin et al. \(2020\)](#). [Thoppilan et al. \(2022\)](#) directly access to the search engine to collect relevant knowledge.

**Knowledge in Pre-Trained Models** The LAMA prob [Petroni et al. \(2019\)](#) first study knowledge stored in pre-trained models. They show that pre-trained models contain a certain amount of factual knowledge without any fine-tuning. This finding has motivated a series of work that adopts knowledge from pre-trained models for downstream tasks. [Roberts et al. \(2020\)](#) show that pre-trained models fine-tuned on question-answering datasets without accessing any external knowledge base could obtain a remarkable result. [Wang et al. \(2022\)](#) prob relational structures from pre-trained models for Text-to-SQL parsing. [Liu et al. \(2022b\)](#) further demonstrate that pre-trained models can generate knowledge via

prompting to help in common sense reasoning tasks. Perhaps the most related work is [Liu et al. \(2022c\)](#), where they adapt a large model to knowledge-grounded conversation via multi-stage prompting and which includes an intermediate knowledge generation step. Compared to this work, our work treats large models as a general-purpose knowledge base, then elicits and transfers knowledge from it to improve a small but strong downstream task model with a distinct architecture.

**Knowledge Distillation** Our work also closely resembles knowledge distillation [Hinton et al. \(2015\)](#), as we similarly transfer knowledge from a large pre-trained model to a small downstream task model. Most existing approaches employ continuous vectors to represent knowledge, e.g., logits [Hinton et al. \(2015\)](#), attention distribution [Wang et al. \(2020\)](#), hidden features [Romero et al. \(2015\)](#) or weights [Lin et al. \(2021\)](#), which are not straightforwardly interpretable. In this work, the large model generates discrete, readable sentences to transfer knowledge.

## 2.4 Long-Context Reasoning

**Multi-Hop Reasoning.** Multi-hop reasoning in open-domain question answering requires the synthesis and analysis of disparate facts across various documents to formulate a response. Key datasets in this field include HotpotQA ([Yang et al., 2018b](#)), 2Wiki ([Ho et al., 2020](#)), and MuSiQue ([Trivedi et al., 2022](#)), which predominantly adopt a reading comprehension framework with pre-retrieved documents supplied by the creators. Traditional approaches often utilize a selector-reader model ([Zhang et al., 2023a](#); [Zhu et al., 2021](#)), where the selector is tasked with pinpointing relevant documents from the provided set, and the reader constructs an answer based on these selections.

Recent advances, however, pivot towards a paradigm that leverages long-context LMs ([Khot et al., 2023](#); [Trivedi et al., 2023](#)). In this approach, the role of the selector is phased out, and instead, the entirety of the retrieved documents is processed by a long-context LM, which acts as the reader. Our study aligns with this emergent research trend, particularly

focusing on the use of attributions to enhance the performance of multi-hop reasoning within this long-context LM framework.

**Context Utilization.** The recent advent of long-context LMs has shown promise (Li et al., 2023a; Zheng et al., 2023b; Chen et al., 2023c). However, these models often struggle with noisy contexts. Shi et al. (2023a) demonstrate that superfluous sentences can significantly disrupt mathematical reasoning. Liu et al. (2024) identify a relevant document position bias in multi-document QA. Wu et al. (2024a) show that LMs could be easily distracted by retrieved irrelevant inputs.

To mitigate the impact of irrelevant context, Shi et al. (2023a) prompt models to disregard such information and adopt self-consistency techniques (Wang et al., 2023). Creswell et al. (2023) suggest a two-stage approach that focuses on fact selection prior to reasoning. Echoing this approach, Yu et al. (2023) introduce Chain-of-Note which entails reviewing document relevance before providing an answer. Meanwhile, Yoran et al. (2023) examine automatic data generation for training more robust models. Our research contributes to this domain by investigating the use of attributions as a novel method for effective context utilization.

**Language Models Attribution.** Attribution in language models constitutes a nascent area of study, primarily aimed at identifying and mitigating hallucination (Li et al., 2023b). A line of research concentrates on post-retrieval answering: Models provide responses based on retrieved results with cited attributions (Nakano et al., 2021; Menick et al., 2022; Gao et al., 2023b). Our research emerges from this foundation but diverges in its application; We focus on multi-hop reasoning rather than hallucination reduction. Moreover, we delve into optimizing training methodologies to maximize the efficacy of scarce attribution annotations.

## 2.5 Test-Time Scaling

**Learning from Feedback.** Other than the heuristic binary feedback studied in this work, prior research has explored feedback from various sources, such as humans (Ouyang et al.,

2022), other models (Yang et al., 2022), tools (Schick et al., 2023), and knowledge bases (Gao et al., 2023a). Chapter 7 focuses on demonstrating the effectiveness of the proposed method and other feedback types are beyond the scope of Chapter 7.

**Test-Time Training.** Test-Time Training (TTT) has shown success in the image modality by addressing distribution shifts and enhancing model capacity through self-supervised fine-tuning on each test case (Sun et al., 2020; Liu et al., 2021; Sun et al., 2023b). Recent studies have extended TTT to the text modality (Hardt and Sun, 2024; Wang et al., 2024b). The most relevant work, by Akyürek et al. (2024), uses TTT to enhance the reasoning ability of LLMs. However, their method relies heavily on human scaffolding for self-supervision and does not generalize beyond ARC-AGI (Chollet, 2019). In contrast, FTTT is generally applicable.

**Learning to Optimize.** Learning to Optimize (L2O) trains a network to act as an optimizer for another network (Chen et al., 2022). Early approaches used reinforcement learning to train such optimizers (Li and Malik, 2017; Chen et al., 2017), while recent work focuses on discovering analytical white-box optimizers (Bello et al., 2017; Chen et al., 2023b). The most relevant work, MEND (Mitchell et al., 2022), trains a network to predict weight updates from training gradients. OPTUNE builds on this idea, extending it to learn from test-time feedback with a distinct architecture.

## Chapter 3

# Comprehensive LLM Evaluation

## Benchmark: CLEVA

### 3.1 Motivation

Large language models (LLMs) have fundamentally revolutionized natural language processing. Transformer models with more than 100B parameters have exhibited remarkable generalization ability across diverse tasks without the need for fine-tuning. The success of GPT-4 (OpenAI, 2023) and ChatGPT sparked a trend of training Chinese LLMs, with new models launching almost every week (Zeng et al., 2023; Team, 2023; Chenghao Fan and Tian, 2023; Ji et al., 2023; Cui et al., 2023). These rapid developments aggravate the need for Chinese LLM evaluation.

**Assessing the capacity of LLMs is non-trivial.** Traditional practices that evaluate models on a single task at a time are gradually becoming obsolete, since a single task can hardly characterize a full view of an LLM’s capacity. Instead, to effectively grasp a holistic view of an LLM’s capacity, we need to decompose its capacity into various abilities, evaluate these abilities with numerous corresponding tasks, and measure the competence of each task with multiple metrics. In this sense, HELM (Liang et al., 2023), leads the way in English LLM evaluation, as it conducts an in-depth evaluation of English LLMs on various NLP tasks using

seven metrics. In Chinese, previous attempts have shown limitations, either in task selection or the metrics adopted. C-Eval (Huang et al., 2023), M3KE (Liu et al., 2023a), CMMLU (Li et al., 2023c), GAOKAO-Bench (Zhang et al., 2023b), and MMCU (Zeng, 2023) narrow down to knowledge and reasoning abilities, whose datasets are mostly constructed using Chinese exams. By the time of our submission, OpenCompass (Contributors, 2023b), with around 74K Chinese queries out of 300K total, leans on accuracy as its sole metric, overlooking other important aspects in LLM evaluation. FlagEval (Contributors, 2023a) inherits four out of seven metrics from HELM and 22 existing Chinese datasets, having limited coverage on some significant tasks. A comprehensive Chinese benchmark incorporating diverse metrics to holistically evaluate Chinese LLMs is urgently demanded.

**Prompt-based evaluation in Chinese is largely unstandardized.** Previous evaluations, such as HELM (Liang et al., 2023), do not explicitly optimize prompts, though LLMs’ significant sensitivity to the format of prompt has been observed (Webson and Pavlick, 2022; Abdou et al., 2022; Sanh et al., 2022). Moreover, unlike many English benchmarks that have well-developed prompts (§ 3.2), many Chinese benchmarks are in their early stage and do not enjoy such privileges. Chinese LLMs are evaluated using different prompts, making the results incomparable and hence untrustworthy.

Consuming up to trillions of tokens during pretraining, LLMs are prone to train-test contamination (Brown et al., 2020), which significantly threatens the validity of an evaluation. Previous work (OpenAI, 2023; Liang et al., 2023) approaches this issue more from a consequentialist perspective: They examine the contamination risk, by methods like long n-gram overlap, only after the evaluation has been done. These post-evaluation analyses, though responsibly examining if train-test contamination happens, cannot alleviate the risk of contamination in the first place. A proactive method to mitigate the contamination risk is of great importance.

We present CLEVA, Chinese Language models EVALuation platform that tackles the aforementioned problems with the following features:

- **A comprehensive Chinese benchmark.** Inspired by HELM (Liang et al., 2023), CLEVA

organizes the evaluation tasks into two parts: *ability evaluation*, which gauges specific LLM skills and *application assessment*, which tests how well LLMs apply their skills to real-world applications (§ 3.3.1). Most of the well-accepted Chinese datasets relevant to our ability evaluation or application assessment are organized, standardized, and then adopted by our platform. More importantly, we design new Chinese-specific tasks, e.g., Pinyin transliteration and intent understanding, and collect a substantial amount of new data, accounting for 33.98% of our total data. As for the metrics (§ 3.3.1), we incorporate metrics for diversity and privacy into our system in addition to the seven in HELM. With 370K (over 9 million queries after augmentation) test instances from 84 datasets and 9 metrics, CLEVA, so far, stands as the most extensive Chinese evaluation dataset and possesses the most dimensions, facilitating a holistic evaluation of Chinese LLMs.

- **Standardized prompt-based evaluation methodology.** CLEVA takes full control of key aspects of LLM evaluation, with data and prompts being the most important among them. All data are jointly prepared with unified preprocessing steps, ensuring a level playing field for all LLMs. Meanwhile, CLEVA provides a set of prompts, instead of just one prompt as in prior work, for each task for prompting-based inference (Brown et al., 2020). This prompt design ensures comparable evaluation results by standardizing the prompts used for testing, while also encouraging further analysis of LLMs’ sensitivity to different prompts (Zhu et al., 2023b).
- **An up-to-date and trustworthy leaderboard.** CLEVA advocates a proactive method for securing trustworthy evaluation results. By collecting extensive new data, CLEVA suppresses the leakage of testing data prior to the evaluation. Moreover, we frequently organize new evaluation rounds, sampling a unique test set from 9 million augmented instances. This strategy further mitigates the risk of train-test contamination, improving the trustworthiness and timeliness of the leaderboard.

CLEVA is thoroughly validated by benchmarking 23 Chinese LLMs on our large-scale test sets (§ 3.5). The corresponding leaderboard and all other user-friendly features will

be continuously maintained and improved to accommodate new models and evaluation methods.

## 3.2 Preliminaries

To measure the model performance on a task, a relevant *test set* is constituted from a collection of *instances*. A test instance will contain multiple *input fields* (string typically) and a list of *references*.

We then adopt a *prompt template*, which essentially describes how to assemble the model input, a.k.a, *prompt*, from input fields (Bach et al., 2022). For example, a Chinese paraphrase identification prompt template (and its translation) is:

*Chinese Example:*

“{sentence1}”和“{sentence2}”这两个问题是在问同一件事情吗?

*English Translation:*

Are the questions “{sentence1}” and “{sentence2}” asking the same thing?

where {sentence1} and {sentence2} are two input fields that will be replaced by the two candidate questions in the test instance. The prompt will be fed into a black-box LLM to predict an output string together with its probability.

Finally, all model predictions and the corresponding test instances will be passed into a *metric* to obtain a numerical value that indicates how well the model performs. Following HELM (Liang et al., 2023), a *metric* is an umbrella for a dimension of measures that share similar purposes. For example, the *accuracy* metric corresponds to BLEU for translation and pass@k for code synthesis. We employ nine metrics, foregrounding metrics beyond accuracy and ensuring a holistic evaluation.

## 3.3 System Design

CLEVA aims to deliver the following two key assets to users who try to evaluate their own LLMs:

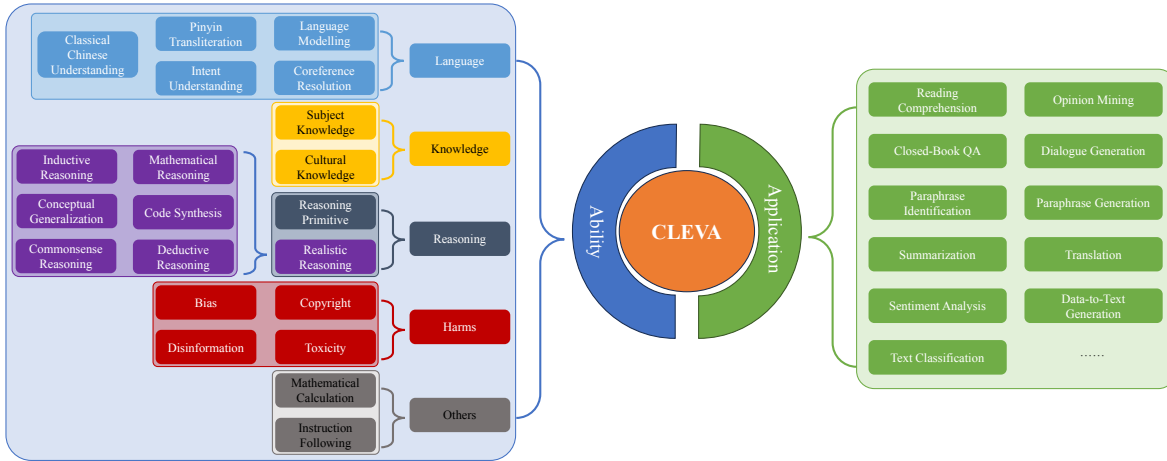


Figure 3.1: CLEVA benchmark.

- A comprehensive and thorough **assessment report** that informs users of the strength and limitations of their models.
- A trustworthy **leaderboard** reflecting the latest advancement of LLMs.

We will discuss our taxonomy that ensures comprehensive evaluations, and challenges like train-test contamination in leaderboards maintenance.

### 3.3.1 Evaluation Taxonomy

Inspired by HELM (Liang et al., 2023), we present a **Tasks**×**Prompts**×**Metrics** evaluation taxonomy for users to evaluate their models. Our evaluation taxonomy carefully designs a Chinese benchmark targeting various LLM abilities, employs a set of diverse prompt templates for each task to characterize the model performance variance, and adopts multiple metrics to comprehensively assess LLMs.

**Tasks.** As shown in Figure 3.1, our Chinese LLM evaluation benchmark consists of two parts: *ability evaluation* and *application assessment*. Each task in ability evaluation focuses on one special skill of LLMs, while application assessment involves real-world NLP tasks that require LLMs to solve practical use cases with their skill sets. Ability evaluation assesses LLM ability from five aspects:

- **Language** measures how well LLMs understand Chinese. In addition to three conventional tasks, we incorporate two tasks specific to Chinese: *Pinyin transliteration* and *classical Chinese understanding*.
- **Knowledge** focuses on assessing the capacity of knowledge acquired by LLMs. We further segment our evaluation into *subject knowledge* and *cultural knowledge* (mainly Chinese culture) based on the source of knowledge. This fine-grained design allows users to closely analyze the model performance across different knowledge categories.
- **Reasoning** evaluates LLMs’ reasoning ability in two settings: *reasoning primitives*, which is independent of language and knowledge background, and *realistic reasoning* that requires reasoning with domain knowledge on practical scenarios. On top of HELM, we additionally include *commonsense reasoning*, *inductive reasoning*, *conceptual generalization*, and *deductive reasoning*.
- **Harms** evaluates the potential risk of LLMs in *copyright*, *disinformation*, *bias*, and *toxicity*.
- **Others** is newly introduced to include crucial yet uncategorized tasks like *mathematical calculation* and *instruction following*.

For application assessment, CLEVA features 11 real-world NLP tasks. In addition to the core scenarios of HELM, we newly include *opinion mining*, *dialogue generation*, *paraphrase generation*, *translation*, *paraphrase identification*, and *data-to-text generation*.

We instantiate the aforementioned tasks in two ways: by directly adopting related public Chinese datasets and by collecting new data. For well-studied tasks, widely-recognized datasets are the best options for forming our benchmark. However, many important tasks, such as *reasoning primitive*, *Pinyin transliteration*, and *disinformation*, lack corresponding Chinese datasets, making the evaluation even more challenging. On these occasions, we either synthesize using sophisticated rule-based scripts (e.g., reasoning primitive) or enlist professional human annotators to construct new test sets. In total, the 31 tasks include 370K test instances from 84 datasets (9M queries in total after applying multiple prompt templates and data augmentation), 33.98% of which are newly collected.

**Prompts.** Ideally, an LLM should be a general interface, capable of understanding prompts with the same semantics, regardless of variations in surface forms. However, LLMs’ notorious sensitivity to prompt templates hinders accurate evaluation (Webson and Pavlick, 2022; Abdou et al., 2022), leading to results that are sometimes incomparable. To better understand an LLM’s sensitivity to plausible human instructions, multiple prompt templates are needed, rather than a single template as in previous work (Contributors, 2023a,b; Liang et al., 2023).

In this work, we manually annotate an average of 3.95 prompt templates for each test set and support all major prompting formats. CLEVA calculates the performance statistics across the entire set of prompts. These statistics do more than just examine the robustness to prompt templates, as reflected by the standard deviation; they also help estimate the upper and lower bounds of an LLM’s performance on a specific test set, as indicated by the minimum and maximum values. Users can benefit from these statistics to select models and to make informed trade-offs between performance and investment in prompt engineering.

**Metrics.** We adopt the 7 metrics from HELM for a holistic evaluation, and, to address recent interest in chatbots and safety concerns, we add two new dimensions: *diversity* and *privacy*.

- **Accuracy.** Accuracy refers to the standard metrics to measure model performance on different tasks, e.g., F1 score for question answering and ROUGE score for summarization.
- **Calibration and uncertainty.** Calibration represents the gap between the model confidence and its actual error rate and is measured mainly by expected calibration error (ECE, (Naeini et al., 2015)).
- **Robustness.** Robustness is the worst-case performance of a model across transformations of test instances. We focus on semantics-preserving perturbations as there are many well-studied data augmentation tools.
- **Fairness.** Similar to robustness, fairness employs perturbations related to social groups to test the disparate treatment and disparate impact of LLMs. We support

4 perturbation strategies: *simplified to traditional conversion*<sup>1</sup>, *Mandarin to Cantonese conversion*<sup>2</sup>, *Chinese name perturbation*, and *gender term perturbation* (Dhole et al., 2021).

- **Bias and stereotypes.** We quantify bias as the disproportionate representation of different social groups. This is gauged through the rates at which these groups are mentioned during model generation. Additionally, we interpret stereotypes as uneven associations between these social groups and certain stereotyped terms, such as occupational roles.
- **Toxicity.** Following HELM (Liang et al., 2023), toxicity is a general term that covers hate speech, abusive language, etc.
- **Efficiency.** Efficiency is a rather broad concept that has many subtleties. It could refer to training or inference efficiency and is measured by energy, carbon, and wall-clock time. As most information could be confidential, we focus only on the inference wall-clock time.
- **Diversity.** Given the popularity of LLM-based chatbots, we incorporate the conventional diversity metric in dialogue systems that evaluates the response surface form diversity (Li et al., 2016). A more diverse response means that the underlying LLM is more creative. Here, we employ the diversity metrics from Miller et al. (2017).
- **Privacy.** In the real-world deployment of LLMs, detecting private information in the generated text, such as Personally Identifiable Information (PII), is a challenging yet important question. We report the portion of PII in the whole test set to make the privacy evaluation generalizable. CLEVA adopts some established tools to smoothly detect PII, and we are working on accommodating more aspects of private content in the near future.

---

<sup>1</sup><https://github.com/BYVoid/OpenCC>

<sup>2</sup><https://justyy.com/tools/chinese-converter/>

### 3.3.2 Leaderboard & Data Contamination

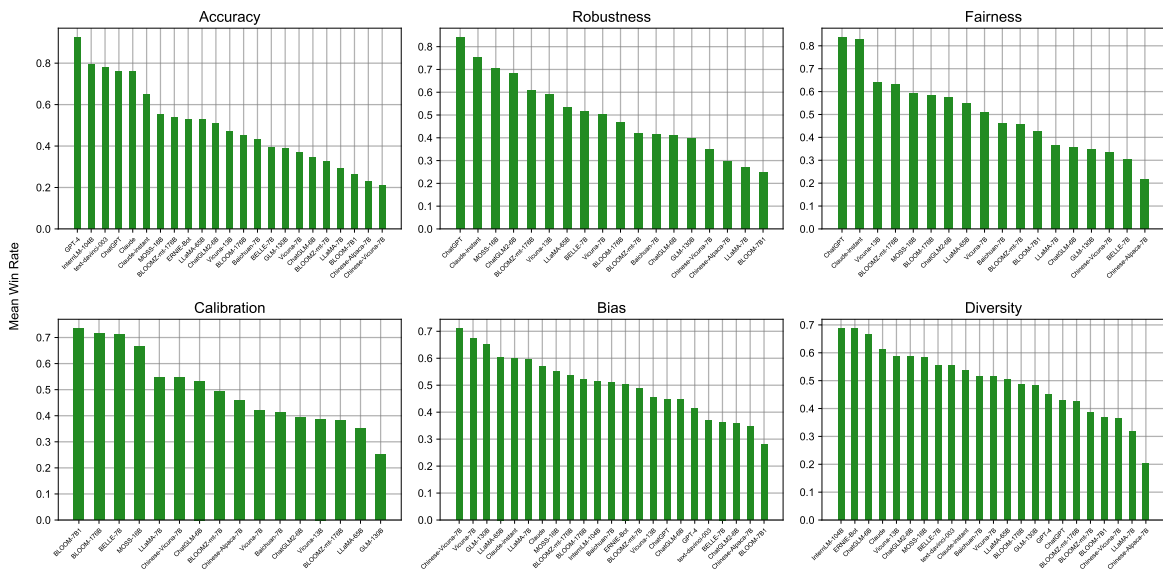
Ensuring fairness, objectivity, and authority is central to maintaining a trustworthy leaderboard. Previous work (Brown et al., 2020) has reported **train-test contamination**, a situation where the test set is included in the training data, leading to unreliable evaluations. Many existing benchmarks, e.g., Huang et al. (2023), conceal the test set labels to avoid data contamination. Given the small scale of their test sets and the large-scale training corpora used by modern LLMs, the risk of unintentional train-test contamination remains high. Sun et al. (2023a) address this problem by making the official test set private and requiring users to submit models’ weights for evaluation. However, this arrangement is unpopular because numerous cutting-edge models consider their weights highly confidential.

We advocate “mutual confidentiality” in LLM evaluation: Users need not expose their model details, and the platform should minimize the risk of disclosing its test set. Instead of model weights, CLEVA only requires API access. We proactively achieve the other half of mutual confidentiality by continuously collecting new data and frequently organizing leaderboard rounds with unique test sets sampling from our full-scale 9 million augmented instances. These strategies not only improve evaluation efficiency but also alleviate train-test contamination from data and temporal perspectives.

To make sure that the sampled subset delivers accurate results, our sampling strategy is not just random sampling: It estimates an acceptable approximation error threshold (i.e., within this threshold, the evaluation results on the sampled set have at least a 70% chance to correctly rank any model pairs), then adjusts the sampling rate for each task according to this threshold, reducing the risk of over-/under-estimating the model performance.

## 3.4 Usage Example

Upon authentication, users are immediately presented with an interactive summary of our evaluation results of 23 LLMs. Users can select from these models, freely exploring the evaluation results from all 9 metrics and 31 tasks.



**Figure 3.2:** The mean win rate of 23 models in 31 tasks. The mean win rate is the probability of a model outperforming a random different model on a random task. We exclude toxicity, privacy, and efficiency metrics as all models excel in the former two, and the latter is often paired with other metrics to deliver meaningful comparisons. Since robustness and fairness involve expensive data augmentation, we only evaluate ChatGPT and Claude-instant.

CLEVA simplifies the evaluation process of new models with minimal coding required. If a user has a model to evaluate, the user only needs a few minutes to finish these three steps: entering the model’s API, selecting relevant tasks from 31 choices, and picking desired metrics from 9 options. CLEVA will autonomously call the user’s model, extract the corresponding responses, and compute the final metrics.

### 3.5 Evaluation

**Setup.** We sample 6.43% of our data to test 23 models that support Chinese. As for the cost, for example, it takes roughly 1600 GPU hours (NVIDIA A100 80G) to evaluate BLOOMZ-176B-mt (Muennighoff et al., 2023).

**Results & Analysis.** Figure 3.2 ranks all models by their mean win rates under different metrics.

- **Accuracy.** It can be seen that GPT-4 (OpenAI, 2023) has the highest winning rate,

followed by other limited-accessed models. This result shows a considerable margin between the performance of open-source models and limited-accessed models. Recent small instruction-following models are better than large LLMs without instruction-tuning, and are even better than some early large instruction-following models, indicating the necessity of effective instruction tuning.

- **Robustness.** The trend on robustness is roughly the same as that of accuracy, with the exception of LLaMA (Touvron et al., 2023a).
- **Fairness.** Most of the model rankings have changed. One possible reason is that fairness involves simplified-to-traditional conversion, and many models have rarely seen traditional Chinese in pretraining.
- **Calibration.** We report  $ECE_{-10}$  (Kumar et al., 2019) following HELM. We find that models with more parameters tend to have higher ECE. For example, GLM-130B (Zeng et al., 2023) and LLaMA-65B rank at the bottom. For BLOOMZ-*mt*-7B vs BLOOMZ-*mt*-176B and BLOOM-7B1 vs BLOOM-176B (Scao et al., 2022), the smaller one wins.
- **Bias.** We focus on gender bias for comparison. GPT-4 and other models, which rank top by other metrics, are at the bottom, while most of the open-source models have low bias. This is because open-source models usually output shorter, resulting in a lower risk of bias.
- **Diversity.** We choose inter-distinct to compare different models. Open-source models generate more diverse and innovative expression than limited-accessed ones, probably due to their fewer safety concerns.

### 3.6 Summary

We present CLEVA, a Chinese LLM evaluation platform. With the largest scale of Chinese instances and broadest metrics, CLEVA provides a comprehensive benchmark to holistically evaluate Chinese LLMs. CLEVA standardizes key components, such as prompt templates, to make evaluation comparable. It also proactively mitigates the contamination issue by collecting large-scale new data, sampling for unique test sets, and regularly updating the

leaderboard.

## Chapter 4

# Robust LLM Evaluation Benchmark: C<sup>2</sup>LEVA

### 4.1 Motivation

Data contamination (Brown et al., 2020; Liang et al., 2023), where test data appears in the training set, has become the central concern in delivering trustworthy evaluations for large language models (LLMs), as these models are typically trained on large-scale corpora that are poorly understood (Dodge et al., 2021). A line of research in data contamination focuses on preventing test data leakage before the evaluation. Common methods include concealing the entire (Sun et al., 2023a) or part (Li et al., 2023e) of the test set during benchmark releases. However, these approaches tend to lose effectiveness over time (Jacovi et al., 2023).

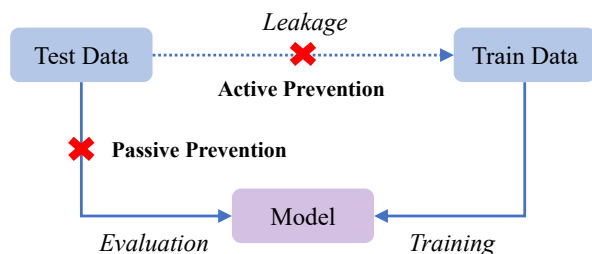
Recently, more promising prevention methods that rely on renewing test data have emerged. These strategies do not depreciate over time. For instance, LatestEval (Li et al., 2024c) and LiveBench (White et al., 2024) gather up-to-date text from the web to create new test cases. DyVal (Zhu et al., 2023a), S3Eval (Lei et al., 2023) and NPHardEval (Fan et al., 2023) synthesize new test data. Additionally, Wang et al. (2024a); Ying et al. (2024); Qian et al. (2024) generate new test cases from existing data. Despite these advances, two challenges persist:

**Missing a comprehensive task taxonomy.** Most prevention methods target only a limited number of tasks. For example, LatestEval (Li et al., 2024c) focuses solely on reading comprehension problems. DyVal (Zhu et al., 2023a), S3Eval (Lei et al., 2023) and NPHardEval (Fan et al., 2023) only evaluate LLMs exclusively on reasoning tasks. LiveBench (White et al., 2024) is relatively comprehensive, covering 18 tasks across 6 categories, but still omits crucial tasks such as those related to harms and practical use cases (Liang et al., 2023). There is a clear need for a comprehensive, contamination-free benchmark to evaluate LLMs holistically.

**Overlooking the contamination risk.** Even though existing prevention methods claim to be free of contamination, they often overlook that “new” does not always imply “unseen”. Users can repurpose open data (Jacovi et al., 2023), making contamination possible even in continuously updated benchmarks. Moreover, these prevention approaches are typically “passive”, lacking control over the released data. Some users may intentionally train their models on the test set and cheat on the benchmark inconspicuously until new data is released. For tasks involving expensive or scarce data sources, such as those related to human values or knowledge, existing methods may soon lose their effectiveness.

We present C<sup>2</sup>LEVA, a benchmark toward **Comprehensive and Contamination-free Language model EVALuation** that addresses the aforementioned issues with the following features:

- **Systematic Contamination Prevention.** C<sup>2</sup>LEVA systematically prevents data contamination from both the *passive* and *active* perspectives: the passive solution aligns with existing work by updating benchmark data to ensure uncontaminated evaluation, specifically addressing repurposing attacks through contamination detection and data scarcity via data augmentation. The active solution minimizes unauthorized use of test data by implementing data protection techniques (Wei et al., 2024; Zhao et al., 2024b) during benchmark release, thereby prolonging the effectiveness of the passive solution. To the best of our knowledge, we are the first to propose active prevention for data contamination, instantiated with data protection.
- **A Comprehensive and Contamination-Free Task Taxonomy.** To ensure extensive



**Figure 4.1:** Contamination prevention overview. Solid lines indicate how data flows within a machine learning model development pipeline. The dotted line indicates where the test data leaks into the training data.

coverage, we follow the task taxonomy of Li et al. (2023e) and apply our contamination prevention techniques to its critical tasks. C<sup>2</sup>LEVA contains 22 tasks for *application assessment* and *ability evaluation*: application assessment encompasses core scenarios of Liang et al. (2023), while ability evaluation gauges LLM capabilities across four aspects: language, knowledge, reasoning, and harms. Additionally, C<sup>2</sup>LEVA provides at least 5 prompt templates for each task to mitigate prompt sensitivities (Zhu et al., 2023b), contributing to a robust evaluation. Furthermore, C<sup>2</sup>LEVA offers data in both English and Simplified Chinese, facilitating the understanding of cross-lingual transfer in LLMs.

C<sup>2</sup>LEVA is thoroughly validated through a large-scale evaluation of 15 open-source and proprietary LLMs. The corresponding leaderboard will be continuously maintained and updated with new evaluation results for emerging models and data. Our experiments also reveal the limitations of the current data protection method in preventing data contamination, underscoring the need for improved approaches in this new research area.

## 4.2 C<sup>2</sup>LEVA

In this section, we first present a systematic discussion of contamination prevention. Then we introduce the task taxonomy adopted in C<sup>2</sup>LEVA and demonstrate how contamination prevention can be applied to tasks within this taxonomy.

### 4.2.1 Contamination Prevention Overview

We first revisit the machine learning model development pipeline, which consists of training the model on training data and evaluating the trained model on test data. As illustrated in Figure 4.1, contamination occurs when 1) test data appears in the training data and 2) developers reuse this leaked test data.

To prevent contamination, two possible actions can be taken: either use another unseen test set for evaluation or avoid the inclusion of test data in the training set. The former is a “passive” approach, as it reacts to existing test data compromise, while the latter is an “active” approach, aiming to prevent data leakage from the outset. Table 4.1 outlines the assumptions, strengths, and weaknesses of these two strategies. Notably, most weaknesses arise directly from violating the assumptions.

Table 4.1 shows that both prevention strategies complement each other: active prevention safeguards tasks where passive prevention is ineffective, such as tasks with scarce or hard-to-collect data. Conversely, passive prevention can renew test data to maintain uncontaminated evaluation results when active prevention is compromised. C<sup>2</sup>LEVA leverages this complementary relationship to achieve systematic prevention across a comprehensive benchmark covering various tasks.

Moreover, data contamination is frequently considered as a threat model for evaluation [Bowen et al. \(2024\)](#). We delineate two specific threat models based on the attacker type, i.e., **intentional** model developers who deliberately train on the test data ([Zhou et al., 2023](#)) and **unintentional** model developers who inadvertently do so ([Brown et al., 2020](#)). Both passive and active prevention are effective against these threat models. However, while passive prevention methods are generally applicable, active prevention strategies require customization based on the attacker type (see § 4.2.3).

**Discussion.** Passive prevention has been extensively explored (see § 2.2). Despite considerable discussion on passive prevention, many methods overlook that new data is not necessarily “unseen” as required. Previous test data can be repurposed as new data, leading

	Passive	Active
Assumption	<ul style="list-style-type: none"> <li>• Data is renewable and unseen.</li> <li>• Task data creation can be <u>automated</u>.</li> </ul>	<ul style="list-style-type: none"> <li>• Evaluation is intact.</li> <li>• <u>Stealthy</u> such that adversaries can not cheat easily.</li> </ul>
Strengths	<ul style="list-style-type: none"> <li>• Always effective as long as unseen data is available</li> <li>• Accurate and trustworthy results.</li> </ul>	<ul style="list-style-type: none"> <li>• No long-term maintenance is required and applied once.</li> <li>• Applicable to many tasks without priors.</li> </ul>
Weaknesses	<ul style="list-style-type: none"> <li>• Costly to build and maintain.</li> <li>• Not applicable to scarce data.</li> <li>• Only available to tasks that can be automated.</li> </ul>	<ul style="list-style-type: none"> <li>• Potential distortion in results.</li> <li>• Invalid once compromised.</li> <li>• Defense strategy depends on attacker type.</li> </ul>

**Table 4.1:** Comparison between passive and active prevention.

to contamination. We later demonstrate how *contamination detection* methods can alleviate this issue. Additionally, some task data may be renewed slowly, such as tasks involving human values and knowledge. We propose *data augmentation* to mitigate data scarcity.

Active prevention, though promising, is rarely explored (Jacovi et al., 2023). Our work pioneers active prevention with data protection (Wei et al., 2024; Zhao et al., 2024b), and explores the effectiveness of these algorithms and the *trustworthiness* of the corresponding evaluation results. Our findings open up a new research topic for data contamination prevention.

#### 4.2.2 C<sup>2</sup>LEVA Task Taxonomy

C<sup>2</sup>LEVA adopts the task taxonomy from Li et al. (2023e). As illustrated in Figure 4.2, tasks are organized into two primary categories: *application assessment*, which targets practical LLM use cases, and *ability evaluation*, which aims to understand the various capabilities of LLMs.

In the **application assessment** category, we focus on *summarization*, *sentiment analysis*, and *text classification*. These tasks are core scenarios in HELM (Liang et al., 2023).

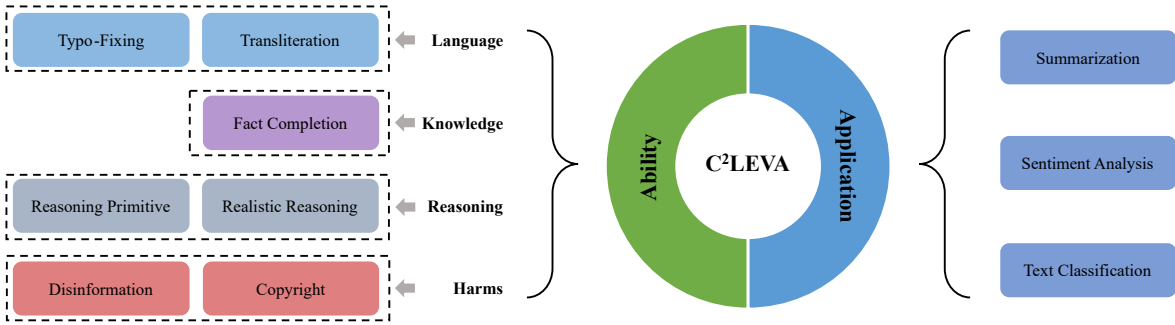


Figure 4.2: The task taxonomy of C<sup>2</sup>LEVA.

In the **ability evaluation** category, we include tasks from four different aspects:

- **Language:** This aspect gauges the LLMs’ proficiency in specific languages. Since C<sup>2</sup>LEVA is bilingual, we select tasks common to both English and Chinese to ensure comparable results. We implement *typo-fixing* (White et al., 2024), where models correct common typos, and *transliteration* (bench authors, 2023; Li et al., 2023e), which assesses knowledge of language pronunciation.
- **Knowledge:** This aspect investigates the LLMs’ understanding of factual knowledge. We require LLMs to answer *fact completion* questions with entities (Petroni et al., 2019).
- **Reasoning:** This aspect evaluates various crucial reasoning abilities and their application to realistic problems. We consider four *reasoning primitive* tasks (Liang et al., 2023) that measure three specific abstract reasoning skills, and seven *realistic reasoning* tasks (Zhu et al., 2023a) that cover three types of practical reasoning problems.
- **Harms:** This aspect measures the potential legal and societal risks posed by LLMs. We primarily investigate *copyright* issues (Liang et al., 2023), evaluating how likely LLMs are to memorize copyrighted content, and *disinformation* (Buchanan et al., 2021), assessing the capability of LLMs to mislead public opinion.

### 4.2.3 The Solution to Contamination Prevention

This section provides detailed descriptions of our contamination prevention solutions.

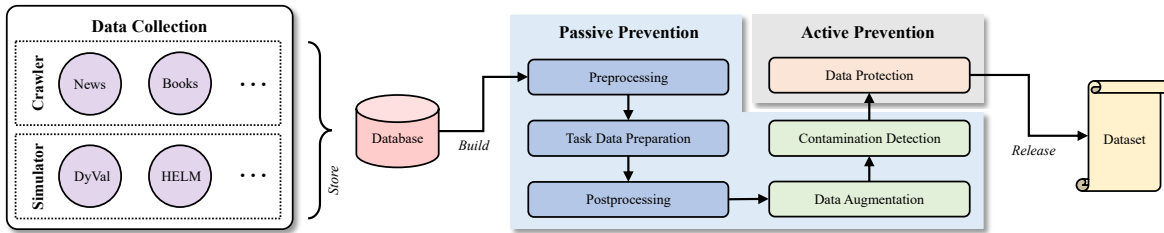
Category	Task	Prevention Strategy
Application	Summarization	Crawling + Contamination Detection
	Sentiment Analysis	Crawling + Contamination Detection
	Text Classification	Crawling + Contamination Detection
Language	Typo-Fixing	Crawling + Rule-based Systems + Contamination Detection
	Transliteration	Crawling + Rule-based Systems + Contamination Detection
Knowledge	Fact Completion	Crawling + Data Augmentation + Contamination Detection + Data Watermarking
Reasoning	Reasoning Primitive	Rule-based Systems
	Realistic Reasoning	Rule-based Systems
Harms	Copyright	Crawling
	Disinformation	Crawling + LLM Assistants + Contamination Detection

**Table 4.2:** Summary of contamination prevention strategy adopted in each task of C<sup>2</sup>LEVA.

**Passive Prevention.** Building on existing work, we devise three basic methods for automating test set construction:

- **Crawling** (White et al., 2024): This method collects task inputs and labels directly from the recent content of appropriate data sources. To ensure new data is unseen, we apply contamination detection to filter out test cases with contamination risks exceeding a predetermined threshold.
- **Rule-based Systems** (Zhu et al., 2023a; Lei et al., 2023; Fan et al., 2023): This method synthesizes new test cases based on predefined complexities. The contamination risk of this method is guaranteed by an extremely low collision probability (Zhu et al., 2023a).
- **LLM Assistants** (Wang et al., 2024a; Ying et al., 2024): This method generates new test cases from existing human-annotated data. Since LLMs may generate their training data (Carlini et al., 2021), we similarly apply contamination detection to exclude risky test data as in *crawling*.

In Table 4.2, we outline the prevention strategies for each task. Some tasks within C<sup>2</sup>LEVA necessitate a combination of the three aforementioned methods for construction (Li et al., 2024c). For instance, in typo-fixing, electronic books are crawled to obtain task labels (*crawling*), followed by the use of `butter-finger` augmentation (Dhole et al., 2021) to generate task inputs (*rule-based systems*). Contamination detection is applied to the test data



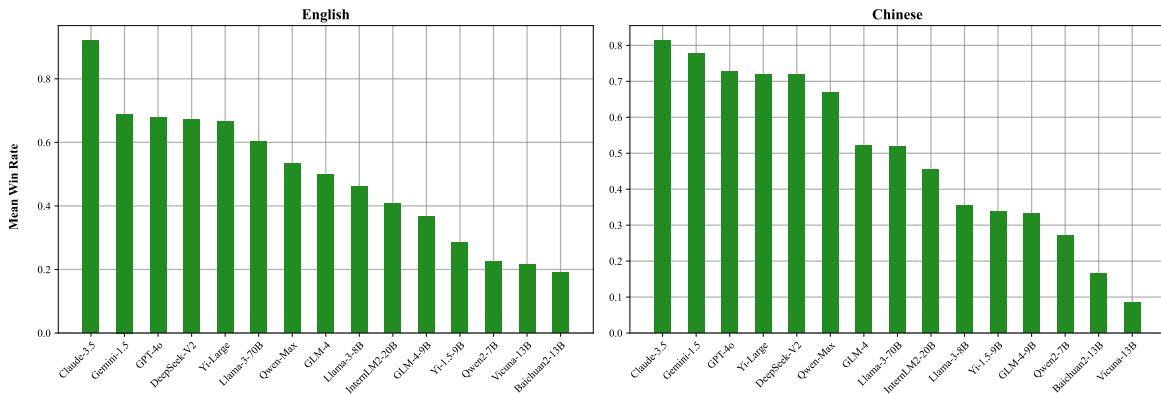
**Figure 4.3:** The framework of C<sup>2</sup>LEVA for contamination prevention.

generated through the integration of these basic construction methods.

For contamination detection, we select Min-K% (Shi et al., 2023b), which provides per-instance contamination risk estimates based on LLM token probabilities. This helps effectively identify contaminated test cases. Although N-gram matching (Brown et al., 2020; Dodge et al., 2021) is another option, it is impractical for us to collect, maintain, and perform N-gram matching on web-scale data. Since we cannot predict which LLMs will be tested before constructing the benchmark, we use Llama-3-8B (AI@Meta, 2024), trained on 15T tokens, as a representative model for others trained on web data.

As stated in Section 4.2.1, passive prevention can be vulnerable if data is scarce. To address this, we introduce semantic-preserving perturbations to augment existing test cases, thereby increasing the number of available test cases without compromising their effectiveness. We prefer a more mechanical approach over LLM-based rephrasing (Wei et al., 2023; Wang et al., 2024a; Ying et al., 2024), as it is transparent and well-understood. Besides, LLMs could generate training data, introducing the contamination risk in data augmentation (Carlini et al., 2021). For practical implementation, we choose synonym substitution (Dhole et al., 2021).

**Active Prevention.** Unlike passive prevention methods, no existing defense can simultaneously prevent attacks from both intentional and unintentional model developers. To address this, we implement tailored defense strategies for each type of attacker as a comprehensive solution.



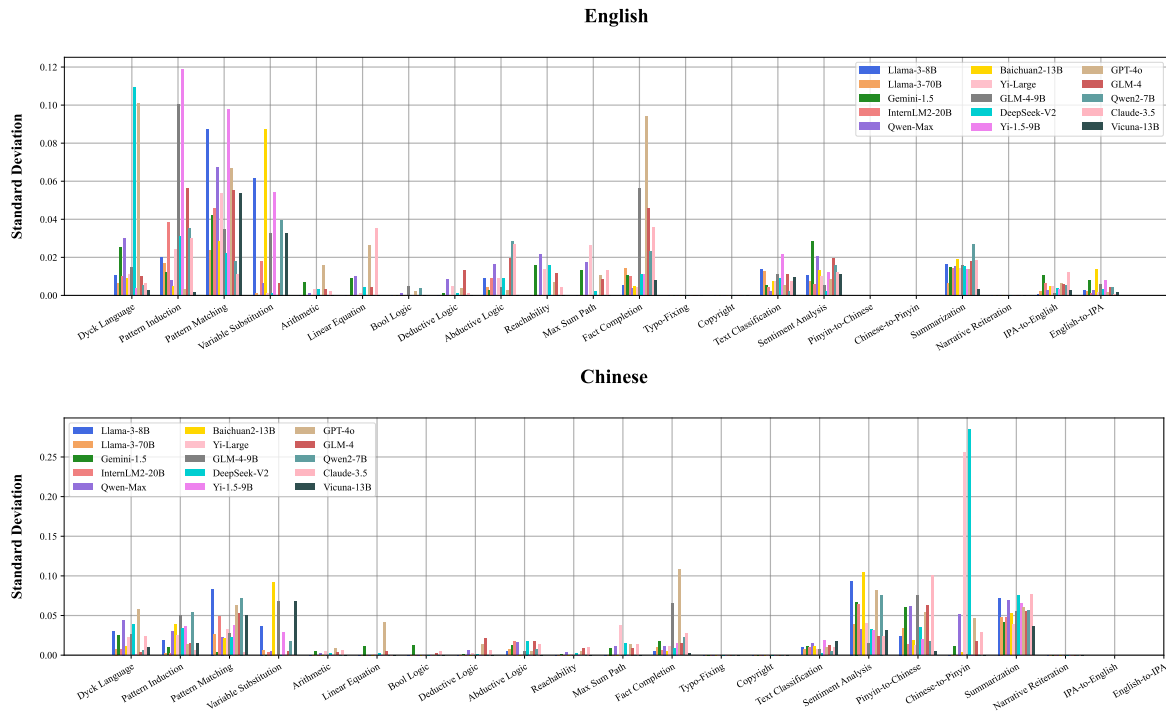
**Figure 4.4:** The mean win rate of 15 models in 22 tasks of C<sup>2</sup>LEVA.

Unintentional contamination often occurs when test data is not excluded during the collection of training data. [Jacovi et al. \(2023\)](#) suggest that properly licensing and encrypting the test data archive can effectively prevent data contamination, assuming model developers are cooperative. Accordingly, we license the test sets under CC BY-NC-ND 4.0 and encrypt the data using ZipCrypto.

For intentional attackers, we use established data protection techniques that facilitate membership inference by embedding stealthy signals into the data ([Hu et al., 2022b](#); [Wei et al., 2024](#); [Zhao et al., 2024b](#)). We choose data watermarking ([Wei et al., 2024](#)) for its provable detection capability. Specifically, we use the random sequence watermark as it is language-agnostic. However, data watermarking can deteriorate model performance and lead to inaccurate evaluations (see § 4.3). Therefore, we apply it to only a random subset of test inputs before licensing and encryption. These modified test cases are designed to ensure a maximum performance loss of 5% while still achieving statistically significant detection with a  $p$ -value of approximately 0.05.

**The Framework.** Figure 4.3 presents the overall framework for constructing C<sup>2</sup>LEVA, consisting of two main stages: *data collection* and *prevention*.

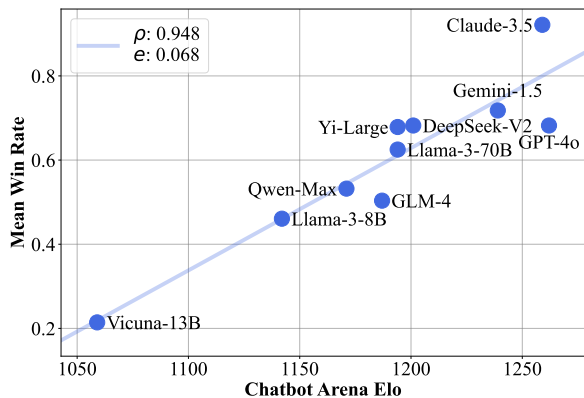
In the data collection stage, crawlers periodically retrieve the latest text data from selected high-quality sources and store it in a centralized database. This stage also runs



**Figure 4.5:** The accuracy standard deviation of different models in different prompt templates from different tasks.

simulators, which are rule-based systems for data synthesis.

In the prevention stage, raw data from the database is accessed to generate a test set. The passive prevention process creates a draft test set through three steps: preprocessing (e.g., filtering out incomplete data), task data preparation (e.g., generating task inputs and labels), and postprocessing (e.g., removing duplicate test cases). Once the draft test set is complete, active prevention applies data protection measures to part of the data and encrypts the release archive with a chosen license.



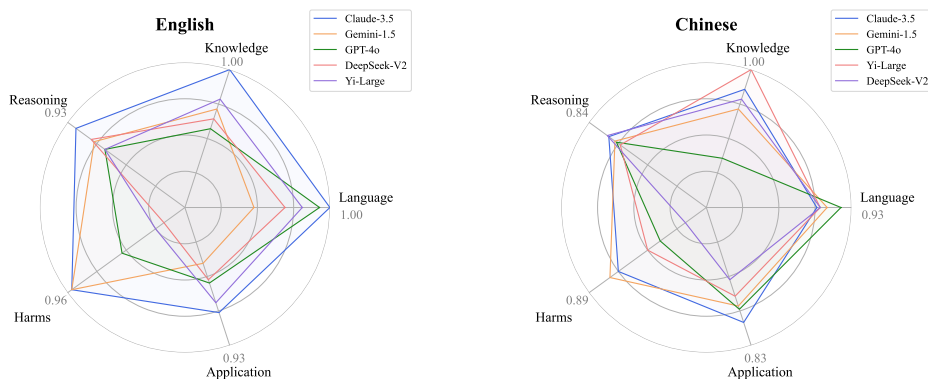
**Figure 4.6:** The English mean win rate of C<sup>2</sup>LEVA scales linearly with style-controlled Chatbot Arena Elo. ● is the observed value. — indicates the linear fit.  $\rho$  and  $e$  denote the Spearman’s ranking correlation and the root mean square error of the linear fit respectively.

## 4.3 Experimental Results

### 4.3.1 Setup

C<sup>2</sup>LEVA encompasses 16,115 test instances, with 8,989 in English and 7,126 in Chinese. 15 LLMs from 11 organizations are evaluated, including GPT-4o (OpenAI, 2023), Claude-3.5 (Anthropic, 2024), Gemini-1.5 (Reid et al., 2024), GLM-4, GLM-4-9B (Zeng et al., 2024), Yi-Large, Yi-1.5-9B (Young et al., 2024), Qwen-Max, Qwen2-7B (Yang et al., 2024), DeepSeek-v2 (DeepSeek-AI et al., 2024), Llama-3-8/70B (AI@Meta, 2024), InternLM2-20B (Cai et al., 2024), Vicuna-13B (Zheng et al., 2023b), and Baichuan2-13B (Yang et al., 2023a). For evaluation, we employ 5-shot prompting. We use automatic metrics for evaluation in each task, except for narrative reiteration, which is assessed through human evaluation<sup>1</sup>. We report the average performance across a set of prompt templates, with a minimum of 5 templates per task, except for specific tasks like copyright. Experiments with open-source models were conducted using 8 NVIDIA A100 80G GPUs over approximately two weeks. The cost of accessing proprietary LLMs’ APIs was approximately \$2134.

<sup>1</sup><https://worksheets.codalab.org/worksheets/0x5f534427e2ae40c4af683b2457676a39>



**Figure 4.7:** Mean win rate comparison among models in different task groups. We choose the top-5 best-performing models in each language for visualization.

### 4.3.2 Main Results

Figure 4.4 presents the rankings of models based on their mean win rate (Liang et al., 2023) across all tasks in C<sup>2</sup>LEVA. The mean win rate indicates the likelihood of a model outperforming a random model on a random task. This metric is used because tasks vary in metrics, complicating direct result comparison. As expected, proprietary models generally surpass open-source ones, and larger models tend to outperform smaller ones.

Notably, the leading models in both languages are Claude-3.5, Gemini-1.5, and GPT-4o, despite their developers being in English-speaking countries. Claude-3.5 significantly outperforms the second-best model in English, attributed to its advanced reasoning capabilities. Since 31.8% of the tasks in C<sup>2</sup>LEVA involve reasoning, and Claude-3.5 excels in other reasoning benchmarks (Anthropic, 2024), this performance is expected. Figure 4.7 supports this observation.

Interestingly, some large models, such as Baichuan2-13B and Vicuna-13B, underperform compared to smaller models like Qwen2-7B and GLM-4-9B. This may be due to the earlier release of these larger models, while recent smaller models utilize advanced techniques and better data.

Performance on each task is averaged across multiple prompts in C<sup>2</sup>LEVA. We also examined how performance varies with different prompts for the same task. Figure 4.5

shows the variability in model performance, measured by standard deviation. Most models exhibit low variability, but many show significant “spikes” on specific tasks, particularly in reasoning primitive. Even in realistic reasoning tasks, many LLMs show a moderate level of variance. This suggests limited robustness in reasoning abilities among the models we examined. In general, smaller models experience more performance spikes. However, even strong models like Claude-3.5 display instability in tasks such as Linear Equation and Max Sum Path. This highlights that prompt sensitivity (Zhu et al., 2023b) remains challenging in evaluating LLMs.

### 4.3.3 Analysis

**Benchmark Effectiveness.** Evaluating the effectiveness of our results is crucial, particularly regarding whether C<sup>2</sup>LEVA is truly comprehensive and free from data contamination. A viable method is to measure the correlation between the C<sup>2</sup>LEVA ranking, based on the mean win rate, and a “ground-truth” ranking (Ni et al., 2024). Such a ground-truth ranking could be the Chatbot Arena Elo (Chiang et al., 2024), derived from millions of user votes for preferred models. These online anonymous votes are based on user-generated queries and judgments, making the leaderboard resistant to manipulation and thus considered contamination-free. A strong correlation between C<sup>2</sup>LEVA and Chatbot Arena Elo indicates effective mitigation of data contamination. Additionally, recent work (Ni et al., 2024) suggests that queries in Chatbot Arena Elo align with web data distribution, implying that high correlation also reflects benchmark comprehensiveness.

Figure 4.6 demonstrates that the mean win rate of C<sup>2</sup>LEVA scales linearly with Chatbot Arena Elo. The Spearman’s rank correlation is 0.948 with  $p < 0.05$ . This supports the conclusion that C<sup>2</sup>LEVA is comprehensive and mitigates data contamination.

**Skill Proficiency.** Figure 4.7 illustrates the proficiency of the top-performing models across various skill sets. We categorize the mean win rate of each model into five task groups: one for application assessment and four for different aspects of ability evaluation.

Model	English			Chinese			$\Delta$ Avg.
	Before	After	$\Delta$	Before	After	$\Delta$	
Claude-3.5	36.72%	28.12%	-23.40%↓	43.75%	26.95%	-38.39%↓	-30.90%↓
Qwen2-7B	21.48%	17.97%	-16.36%↓	32.42%	25.00%	-22.89%↓	-19.63%↓
GLM-4	22.27%	17.97%	-19.30%↓	21.88%	18.36%	-16.07%↓	-17.68%↓
Llama-3-8B	26.17%	25.39%	-2.99%↓	27.73%	19.92%	-28.17%↓	-15.58%↓
InternLM2-20B	31.25%	27.34%	-12.50%↓	35.94%	30.08%	-16.30%↓	-14.40%↓
Qwen-Max	30.47%	30.47%	0.00%↑	37.50%	28.52%	-23.96%↓	-11.98%↓
DeepSeek-V2	24.22%	22.66%	-6.45%↓	34.77%	28.91%	-16.85%↓	-11.65%↓
Vicuna-13B	24.61%	26.17%	6.35%↑	21.09%	14.84%	-29.63%↓	-11.64%↓
Gemini-1.5	26.95%	26.95%	0.00%↑	38.67%	30.86%	-20.20%↓	-10.10%↓
Yi-Large	32.81%	32.42%	-1.19%↓	39.45%	33.20%	-15.84%↓	-8.52%↓
Yi-1.5-9B	17.97%	18.36%	2.17%↑	32.03%	26.95%	-15.85%↓	-6.84%↓
GLM-4-9B	17.97%	12.50%	-30.43%↓	10.55%	12.50%	18.52%↑	-5.96%↓
Baichuan2-13B	18.36%	17.58%	-4.26%↓	35.16%	33.20%	-5.56%↓	-4.91%↓
GPT-4o	28.52%	28.52%	0.00%↑	28.52%	26.95%	-5.48%↓	-2.74%↓
Llama-3-70B	33.98%	35.16%	3.45%↑	39.06%	36.72%	-6.00%↓	-1.28%↓

**Table 4.3:** Evaluation result distortion of data watermarking (Wei et al., 2024) in the fact completion task. **Before** is the results before applying data watermarking and **After** is the results after applying data watermarking.  $\Delta$  indicates the performance change, where the performance loss is marked in red and green for the performance gain.

Within each group, Claude-3.5, the top performer in English, consistently surpasses other models, especially in knowledge and reasoning tasks. This is consistent with their technical report (Anthropic, 2024), which emphasizes Claude-3.5’s excellence in knowledge-intensive tasks like MMLU (Hendrycks et al., 2021a) and reasoning tasks such as GSM8K (Cobbe et al., 2021). However, in Chinese, Claude-3.5 does not maintain its English advantage. Yi-Large excels in knowledge tasks, and DeepSeek-V2 outperforms Claude-3.5 in reasoning tasks. This indicates significant potential for improvement in the cross-lingual transfer capabilities of top-tier LLMs.

**Data Protection.** This study examines the impact of data protection methods, specifically data watermarking, which inherently alters the data (Wei et al., 2024). It is crucial to evaluate the effects of these methods on performance, as previous research indicates that input noise can significantly impair the performance of LLMs (Shi et al., 2023a; Liu et al., 2024). Consequently, we measured performance changes before and after applying watermarking. Table 4.3 details the performance change for each model across two languages in the fact

completion task. Overall, data watermarking results in a performance decline in 76.67% of cases, with an average loss of approximately 11.59% across all models. This effect is particularly noticeable in Chinese, where the average loss is 16.18%, with only the GLM-4-9B model maintaining robust performance. Additionally, proprietary models tend to be more affected by data protection measures; notably, 4 out of 5 models with the least performance drop are open-sourced and smaller in scale. This may be due to open-source models being trained on noisy instruction data. These findings underscore the necessity for enhancements in data protection techniques to reduce their adverse effects on evaluation. To minimize distortion during evaluation, we applied watermarking to only a small portion of the data in practice.

## 4.4 Summary

In this work, we present C<sup>2</sup>LEVA, a comprehensive and contamination-free bilingual benchmark. C<sup>2</sup>LEVA features a systematic contamination prevention strategy, which improves existing passive prevention methods and proposes a novel active prevention solution. Large-scale evaluation of 15 LLMs has been conducted on C<sup>2</sup>LEVA.

## Chapter 5

# Applications of LLM Evaluation: Knowledge Grounded Conversation

### 5.1 Motivation

Knowledge-grounded conversation (Dinan et al., 2019; Moghe et al., 2018) is the task where the model could reply to a dialogue history based on extra knowledge. Compared to standard conversational modeling, this extra knowledge prevents the model from generating generic and non-informative responses (Li et al., 2016). Typically, at each turn of the conversation, a pool of knowledge candidates will be retrieved from a knowledge base like unstructured documents (e.g., Wikipedia) (Dinan et al., 2019) or a structured knowledge graph (Dziri et al., 2021). The model then learns to select the most related knowledge from this pool, in an unsupervised manner, to generate its response.

However, constructing and maintaining knowledge bases are time-consuming and expensive. Recent studies have shown that large pre-trained models are capable of grasping knowledge from unsupervised text corpora and memorizing facts to their weights (Petroni et al., 2019; Roberts et al., 2020; Lewis et al., 2021; Wang et al., 2021; Liu et al., 2022b). These large models can even perform reasoning implicitly (Wei et al., 2022b). In light of this remarkable capacity of large models, we explore the possibility of leveraging large models

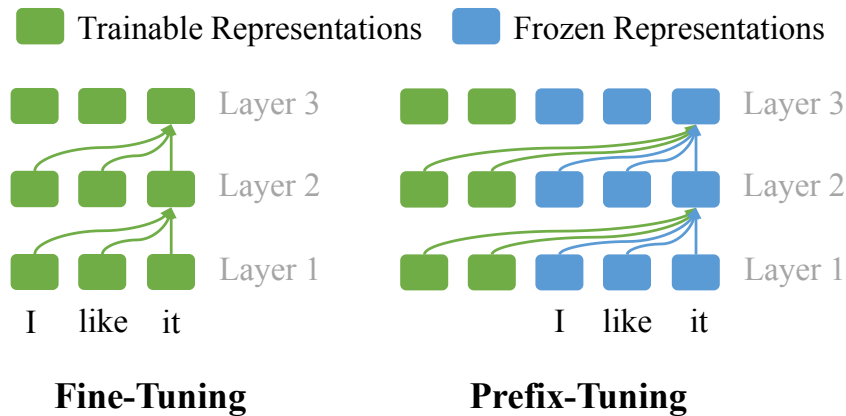
as a new knowledge source for unsupervised knowledge-grounded conversation.

In this work, we first investigate the quality of knowledge generated by large models. We examine which tuning method, including the conventional fine-tuning (Devlin et al., 2019; Zhang et al., 2021) and the recently proposed prefix-tuning (Li and Liang, 2021), best prompts knowledge from large models for a given dialogue history. We then design a human evaluation protocol and conduct an extensive quality assessment of the generated knowledge. Despite some extent of hallucinations (plausible statements with factual errors) persist (Maynez et al., 2020), large models can mostly generate related and correct knowledge for the future development of dialogue. Moreover, some of this knowledge is not simply paraphrased or copied from web pages: they summarize scattered facts on the Internet (See Section 5.5.1). These observations advocate the unique value of employing large models as knowledge bases.

Owing to the hallucinations, it is risky to put generated knowledge directly into the dialogue system as the misinformation could contaminate the response. We instead consider the generated knowledge as a noisy knowledge source and use it to aid the knowledge selection process. Specifically, we measure its similarity to each knowledge candidate and refine the knowledge selection accordingly (See Section 5.3.2). We further estimate the posterior of the refined knowledge selection distribution, inspired by the fact that the posterior detangles the one-to-many relation between dialogue context and knowledge selection (Kim et al., 2020). In addition, we propose a noisy training strategy to strengthen the model’s ability on handling noisy knowledge (See Section 5.3.3). All these strategies significantly elevate the performance of the existing state-of-the-art model to a new level on two widely-adopted benchmarks, Wizard of Wikipedia (Dinan et al., 2019) and Holl-E (Moghe et al., 2018).

## 5.2 Eliciting Knowledge from Large Models

In this section, we first introduce the tuning methods and large pre-trained models we used to generate knowledge for a given dialogue history. Then we show the tagset developed for



**Figure 5.1:** Comparison of fine-tuning (Devlin et al., 2019) and prefix-tuning (Li and Liang, 2021).

evaluating the generated knowledge.

### 5.2.1 Methods and Models

Since the objective function of large pre-trained models is to predict possible words instead of knowledge given the context (Devlin et al., 2019; Radford et al., 2019), tuning these large models on annotated data is necessary. Here we focus on two tuning methods, as shown in Figure 5.1:

- **Fine-Tuning** (Devlin et al., 2019; Zhang et al., 2021) which updates all weights in the model.
- **Prefix-Tuning** (Li and Liang, 2021) which freezes the pre-trained weights and tunes only a small set of parameters that are added as the prefix of the model’s input.

Fine-tuning remains the standard approach for leveraging pre-trained models in downstream tasks, while prefix-tuning has a comparable performance but avoids the risk of catastrophic forgetting (Goodfellow et al., 2013), which is desirable in our task.

Another challenge is selecting large models. Because our task requires large models to understand the dialogue history and then recommend a related knowledge piece for the user to follow up, we study two types of large models:

- **Pre-trained Language Models (PLMs)** that are trained on web documents with access

	Tag	Definition
<i>Context Understanding:</i>		
1	Related	The generated output discusses facts that are related to the conversation.
2	Unrelated	The generated output does not discuss facts that are related to the conversation.
<i>Tuning Effectiveness:</i>		
3	Non-Verifiable	The generated output does not contain facts that could be verified.
4	Verifiable	The generated output contains facts that could be verified.
<i>Fact-Checking:</i>		
5	Supported	One can find evidence from the knowledge base to validate the factual information in the generated output.
6	Explicit Supported	One only needs to find one evidence from the knowledge base for validation.
7	Implicit Supported	One needs to find multiple evidences from the knowledge base for validation.
8	Refuted	One can find evidence from the knowledge base to contradict the factual information in the generated output.
9	Not Enough Information	The factual information in the generated output could not be validated.
10	Reasonable NEI	Though not validated by the knowledge base, the factual information matches common sense.
11	Unreasonable NEI	Though not validated by the knowledge base, the factual information does not match common sense.
12	Hard NEI	The factual information could not be validated by either the knowledge base or common sense.

**Table 5.1:** The tagset developed to evaluate the quality of the generated knowledge by human annotators.

to abundant knowledge during pre-training.

- **Pre-trained Dialogue Models (PDMs)** that are trained on dialogue data to better understand the dialogue history.

We choose T5 (Raffel et al., 2020) as the representative of PLMs and DialoGPT (Zhang et al., 2020) for PDMs, because they release a series of checkpoints with different model sizes.

Besides, we experiment with various decoding methods to see which of them best suits each type of large models, including greedy decoding, beam search and top-K sampling (Fan et al., 2018). We find that PDMs work best with top-K sampling and beam search for PLMs.

## 5.2.2 Annotation Tagset

To assess the quality of the generated knowledge, we develop an annotation tagset for human evaluation in Table 5.1. Each generated knowledge along with its associated dialogue history will be annotated by at least two tags, each from a different category: *context understanding*, *tuning effectiveness* (and *fact-checking* if outputs contain facts).

**Context Understanding** *Related* and *Unrelated* in rows 1-2 of Table 5.1 measure whether large pre-trained models understand the conversation and generate related knowledge. Although we can use automatic metrics like the F1 score that measures the distance between the generated knowledge and the ground truth knowledge as an alternative, a single reference only captures one possible future direction of the dialogue. In this sense, human evaluation provides a more comprehensive assessment.

**Tuning Effectiveness** *Non-Verifiable* (e.g., chitchat) and *Verifiable* in rows 3-4 indicate the reliability of the tuning methods for eliciting knowledge from large models. If a tuning method is effective, models should generate outputs that contain *Verifiable* facts.

**Fact-Checking** Among those *Verifiable* outputs, we classify them into *Supported* (facts is supported by evidence), *Refuted* (facts is refuted by evidence) and *Not Enough Information* (NEI, evidence is not found), as shown in rows 5-12 of Table 5.1. These tags are mainly adapted from Gupta et al. (2022). Annotators will gather trustworthy evidence via search engines to determine the label.

To better understand the detailed behavior of large models, we divide *Supported* into *Explicit Supported* and *Implicit Supported*. The former means that large models memorize existing documents, while the latter implies that they do more than memorization, e.g., summarization. We also let the annotators check whether *NEI* outputs could be validated by common sense. If common sense could be used for validation, these *NEI* outputs will be further classified into *Reasonable NEI* (facts match common sense) or *Unreasonable NEI* (facts contradict common sense), and *Hard NEI* if common sense

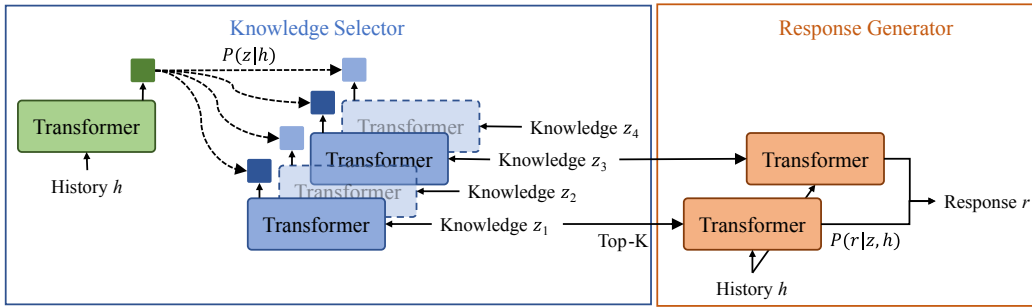


Figure 5.2: Illustration of PLATO-KAG (Huang et al., 2021).

is not applicable.

### 5.3 Exploiting Generated Knowledge for Conversation

In this section, we first review the state-of-the-art approach - PLATO-KAG (Huang et al., 2021). Then we develop our method on top of PLATO-KAG to exploit generated knowledge.

#### 5.3.1 PLATO-KAG

As shown in Figure 5.2, PLATO-KAG is a model consisting of two modules: a knowledge selector which selects top-K most relevant knowledge to the dialogue history from a pool of retrieved knowledge candidates, and a response generator that generates the response based on the dialogue history and the selected knowledge.

**Knowledge Selector** The knowledge selector adopts a dual encoder with shared parameters to extract features. The dialogue history  $h$  and a knowledge candidate  $z$  will pass to this encoder independently to get their own representations. Then it estimates the relevance between the dialogue history  $h$  and a knowledge candidate  $z$  by:

$$f(h, z) = (W_h E(h))^T (W_z E(z)) \quad (5.1)$$

where  $E(\cdot)$  is the fixed-length vector representation of the input, i.e., the encoder's output on the [CLS] token.  $W_h$  and  $W_z$  are two linear projections.

To select the top-K knowledge candidates, the knowledge selector computes the relevance between  $h$  and all possible  $z$ . Then only the top-K most related knowledge  $Z = \{z_1, \dots, z_K\}$  is retained to construct the knowledge selection distribution  $P(z|h)$  as follows:

$$P(z|h) = \frac{e^{f(h,z)}}{\sum_{z' \in Z} e^{f(h,z')}} \quad (5.2)$$

**Response Generator** After the knowledge selection, the response generator will predict the probability of the response  $r$  by:

$$P(r|h) = \sum_{z \in Z} P(z|h)P(r|h, z) \quad (5.3)$$

where  $P(r|h, z) = \prod_i P(r_i|h, z, r_{<i})$  is a decoder that generates response  $r$  given the dialogue history  $h$  and one knowledge candidate  $z$ .

### 5.3.2 Posterior-based Reweighing

**Reweighing** Generated knowledge contains hallucinated facts, as later shown in Section 5.5.1. It is thus not viable to take generated knowledge  $g$  as the direct input of the model. Instead, we interpret  $g$  as noisy ground truth and define a refined knowledge selection distribution  $P^*(z|h)$ :

$$P^*(z|h) = P(z|h)P(z|g) \quad (5.4)$$

where  $P(z|h)$  is the original knowledge selection distribution and  $P(z|g)$  measures the similarity between  $g$  and  $z$ . This refined distribution  $P^*(z|h)$  will score  $z$  high only if  $z$  is close to the history  $h$  as well as the noisy ground truth  $g$ .

Similar to Eqs. 5.1-5.2, we define  $P(z|g)$  to measure the closeness between  $g$  and each  $z$ :

$$f(g, z) = (W_h E(g))^T (W_z E(z)) \quad (5.5)$$

$$P(z|g) = \frac{e^{f(g,z)/\alpha}}{\sum_{z' \in Z} e^{f(g,z')/\alpha}} \quad (5.6)$$

where  $\alpha$  is a hyper-parameter that controls the sharpness of  $P(z|g)$ .

**Posterior** Kim et al. (2020) suggests that the posterior selection distribution  $P(z|h, r)$  could

select a more appropriate knowledge than the prior selection distribution  $P(z|h)$ , because the presence of future response  $r$  narrows down the scope of all possible  $z$ . We drive the posterior of the refined selection distribution  $P^*(z|h)$  via the Bayes rule:

$$P(z|h, r) = \frac{P(r|h, z)P^*(z|h)}{\sum_{z' \in Z} P(r|h, z')P^*(z'|h)} \quad (5.7)$$

where the denominator is tractable as there are only a small number of  $z$  and  $P(r|h, z)$  is exactly the response generator.

The main challenge is how to estimate  $P(r|h, z)$  when  $r$  is not yet observed. We let the response generator greedily decode a most likely response  $\bar{r}$  for a given  $z$ . However, different  $z$  might result in  $\bar{r}$  with various lengths. A long  $\bar{r}$  tends to have a lower probability and is not competitive with the short one (Yang et al., 2018a). We therefore use the mean token probability as the approximation of  $P(r|h, z) \approx \frac{1}{N} \sum_{i=1}^N P(\bar{r}_i|h, z, \bar{r}_{<i})$ , where  $N$  is the length of  $\bar{r}$ .

In the end, we add a hyper-parameter  $\beta$  to control the sharpness of the posterior  $P^*(z|h, r) \propto P(z|h, r)^\beta$ . Since we only apply the Bayes rule once to obtain the posterior, adjusting the sharpness help to amplify or diminish the impact of updating.

### 5.3.3 Noisy Training

Note that Eq. 5.7 is only applicable when the response generator  $P(r|h, z)$  is able to denoise, i.e., the model should give the likelihood estimate of a low value if  $z$  is not appropriate. In this case, the Bayes rule will update the posterior by lowering the chance of this inappropriate  $z$  being selected. Since the knowledge selector always presents its most confident selection to the response generator and the knowledge selector performs much better in the training set (a top-K accuracy of 90.3% for the baseline) than in the test set (a top-K accuracy of 68.1%), such a bias will lead to a training-inference discrepancy and therefore the response generator is not resilience to noisy knowledge during testing.

To alleviate this issue, we employ the Gumbel-TopK trick (Kool et al., 2019), which adds noise to the top-K operation in the knowledge selector during training. Specifically, we

sample noise from the Gumbel distribution with location  $\mu = 0$  and scale  $\phi = 1$ . This noise will add to  $f(h, z)$  in Eq. 5.1 to permute the ranking of knowledge candidates and perturb the selection distribution  $P(z|h)$ .

## 5.4 Experimental Setup

### 5.4.1 Datasets

We conduct experiments on two popular benchmarks: Wizard of Wikipedia (Dinan et al., 2019) (WoW), and Holl-E (Moghe et al., 2018). The WoW dataset covers a wide range of topics (1,365 in total). Each conversation in WoW happens between a wizard who has access to knowledge from Wikipedia about a specific topic and an apprentice who learns from the wizard about the topic. Specifically, for our knowledge generation task in Section 5.2.1, the input is the dialogue history and the target is the ground truth knowledge that the wizard used to generate his response. There are 18,340/1,948/1,933 dialogues in the training/validation/test set. The validation and test sets are split into two categories: *Seen* which contains new dialogues with topics that appeared in the training set and *Unseen* whose dialogues have topics that never appear in the training set. We follow Dinan et al. (2019)’s scripts to preprocess the data.

Compared to WoW, conversations in Holl-E happened between two participants discussing a specific movie, where a single document about that movie is given as knowledge. There are 7,228/930/913 dialogues in the training/validation/test split. We follow Kim et al. (2020)’s scripts for data preprocessing.

### 5.4.2 Evaluation Metrics

We assess all results (generated knowledge and responses) via both the automatic metric and human evaluation.

**Knowledge Generation Assessment** In automatic evaluation, we compute the unigram F1 between the generated knowledge and the ground truth knowledge. In human evaluation, we recruit three well-trained annotators who are fluent in English to evaluate 100 random samples from the seen and unseen test sets each, according to the scheme we proposed in Section 5.2.2. The tag of an example is determined by the majority vote of the three annotators. The agreement among the annotators is measured via Fleiss’ kappa (Fleiss, 1971).

**Response Generation Assessment** In the automatic evaluation, we report the perplexity (PPL) and Unigram F1 of ground truth responses. We also collect the top-1 knowledge accuracy (P@1) statistics, which evaluate the performance of the knowledge selector. In the human evaluation, 100 random examples from WoW seen and unseen test sets are distributed to three annotators respectively. They will evaluate these samples in four aspects, following Huang et al. (2021):

- **Coherent** measures whether the response is consistent with the dialogue history.
- **Informativeness** evaluates whether the response is generic and non-informative or not.
- **Engagingness** assesses how likely the annotator is willing to continue the discussion.
- **Hallucination** checks the correctness of the contained factual information.

Coherence, informativeness and engagingness are in the range of  $[0, 1, 2]$ . A higher value implies a better result. Hallucination is in the range of  $[0, 1]$ , where 0 means the response is factually correct and 1 means the response contains hallucinated facts. We refer the readers to Huang et al. (2021) for more details. The final score of each sample is determined through majority voting.

### 5.4.3 Response Generation Baselines

**TMN** is the baseline released along with the WoW dataset (Dinan et al., 2019). It stores knowledge candidates’ features in the memory for selection. We include the released

PLM Type	Model	#Params	Fine-Tuning		Prefix-Tuning	
			Test Seen	Test Unseen	Test Seen	Test Unseen
N/A	T5-large	737M	0.2530	0.1044	0.1312	0.1245
PLM	T5-small	60M	0.2521	0.1796	0.2138	0.1720
	T5-base	222M	0.2679	0.1807	0.2494	0.1735
	T5-large	737M	0.2624	0.1943	0.2575	0.1579
	T5-XL	3B	0.2684	0.2053	0.2629	0.1808
	T5-XXL	11B	-	-	0.2652	0.1874
PDM	DialoGPT-small	124M	0.2357	0.1588	0.3041	0.1456
	DialoGPT-medium	355M	0.3216	0.1663	0.3173	0.1598
	DialoGPT-large	774M	0.3217	0.1705	0.3209	0.1613

**Table 5.2:** The automatic evaluation result (Unigram F1) of two tuning methods for generating knowledge in WoW seen and unseen test sets (“N/A” means we do not use the pre-trained weights). Results of fine-tuned T5-XXL are missing due to resource constraint.

unsupervised trained checkpoint in our experiments<sup>1</sup>.

**SKT** models the knowledge selection process in multi-turn dialogue generation as a sequential latent variable model (Kim et al., 2020). We use their open-sourced models in our experiments<sup>2</sup>.

**KnowledGPT** fine-tunes a GPT-2 (Radford et al., 2019) and leverages the reinforcement learning approach to train an unsupervised sequential knowledge selector (Zhao et al., 2020). We adopt their released model for experiments<sup>3</sup>.

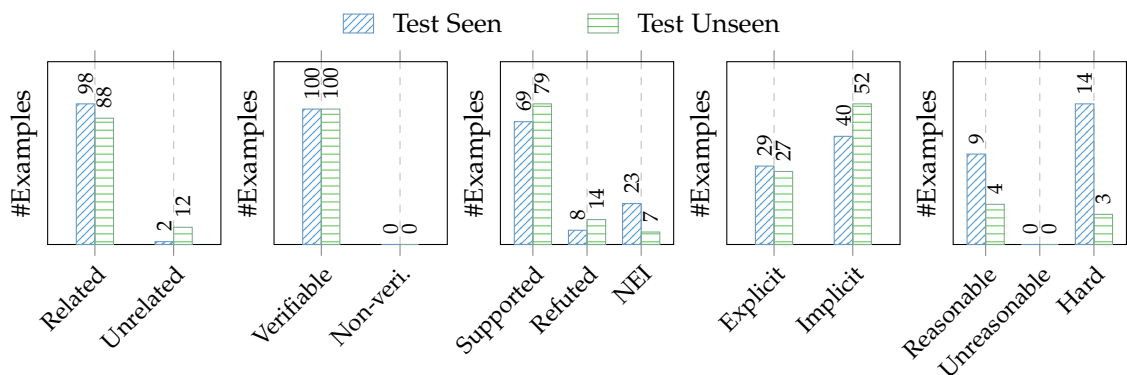
#### 5.4.4 Implementation Details

**Knowledge Generation** For fine-tuning, all models use a batch size of 64, a learning rate of 5e-5, and the inverse square root learning rate scheduler (Vaswani et al., 2017) with 1000 warmup steps. We validate the model on the validation set every 1000 steps and early stop the training if the performance does not improve after 15 validations. For prefix-tuning, the

<sup>1</sup>[https://parl.ai/projects/wizard\\_of\\_wikipedia](https://parl.ai/projects/wizard_of_wikipedia)

<sup>2</sup><https://github.com/bckim92/sequential-knowledge-transformer>

<sup>3</sup><https://github.com/zhaoxlpku/KnowledGPT>



**Figure 5.3:** The human evaluation results of generated knowledge from fine-tuned DialoGPT-large on WoW seen and unseen test sets.

prefix length is set to 5 as in Li and Liang (2021). Other hyper-parameters are almost the same as in fine-tuning, except that the learning rate is kept constant and reduced by 1/10 only if the validation set performance does not improve after 10 validations. At inference, DialoGPT is decoded with top-K sampling where K is 10 and the beam size is 20. For T5, we use beam search with a beam size of 10.

**Response Generation** Since Huang et al. (2021) did not release their codes and models before we start the experiments, we reimplement their approach in ParlAI (Miller et al., 2017) and report our own results as well. We follow Huang et al. (2021)’s hyper-parameters settings in our experiments. For our proposed reweighing method, we perform a grid search on the validation set ( $\alpha \in [1, 10], \beta \in (0, 1)$ ) and set  $\alpha = 5, \beta = 0.4$ . According to Table 5.2, we choose the generated knowledge of DialoGPT-large for our experiments, as it performs the best on average. All experiments are conducted on 8 NVIDIA A100 80G. It takes roughly one day to train one model.

## 5.5 Results and Analysis

### 5.5.1 Knowledge Generation Results

We conduct a case study of eliciting knowledge from large models on the WoW dataset and present the evaluation results.

#### Automatic Evaluation Results

Table 5.2 shows F1 scores of various large models tuned by different methods on the seen and unseen test sets. Results of fine-tuning T5-XXL are missing because we do not have enough resources to train this model. The first row of Table 5.2 is the baseline result of tuning a randomly initialized T5-large model. We observe that nearly all large models perform better than this baseline, especially on the unseen test set. This observation indicates the pre-trained weights do store a lot of factual information as they make a non-trivial improvement.

We also see that PDMs perform much better than PLMs on data with a seen topic, while PLMs are better on the unseen topic in most cases. This might be the consequence that PLMs are trained on diverse text data, which allows them to generalize better on unseen topics. PDMs, on the other hand, are trained on dialogue data only and have a smaller discrepancy between pre-training and fine-tuning. Thus PDMs perform better on seen topics. We also find that the results of fine-tuning are much better than prefix-tuning in general. But this gap is closed when the model gets larger, which is aligned with the conclusion in [Lester et al. \(2021\)](#).

Interestingly, large models scale poorly on our task. On the unseen test set, the performance increases only around 3 points while the model size is 50× larger (0.1796 for fine-tuned T5-small with 60M parameters vs. 0.2053 for fine-tuned T5-XL with 3B parameters).

Test Seen	Automatic Evaluation			Human Evaluation			
	PPL↓	P@1↑	Unigram F1↑	Coherence↑	Informativeness↑	Engagingness↑	Hallucination↓
TMN	61.21	0.220	0.172	0.4757	0.3883	0.4175	0.0777
SKT	57.27	0.258	0.187	0.9806	0.7767	0.6990	0.0680
KnowledGPT	19.60	0.262	<u>0.209</u>	1.0000	<u>1.2330</u>	1.0874	<u>0.0097</u>
PLATO-KAG	9.767	0.253	0.188	-	-	-	-
PLATO-KAG*	11.51	<b>0.266</b>	0.207	<u>1.4757</u>	1.1748	<b>1.2816</b>	0.0388
PLATO-KAG <sup>+</sup>	12.37	0.254	<b>0.211</b>	<b>1.4951</b>	1.1845	1.2718	0.0291

Test Unseen	Automatic Evaluation			Human Evaluation			
	PPL↓	P@1↑	Unigram F1↑	Coherence↑	Informativeness↑	Engagingness↑	Hallucination↓
TMN	103.1	0.112	0.150	0.5000	0.2788	0.3173	0.1058
SKT	87.93	0.177	0.157	0.7019	0.5000	0.5385	0.0673
KnowledGPT	22.85	0.238	<u>0.196</u>	0.9712	0.9904	0.7692	<u>0.0096</u>
PLATO-KAG	11.46	<b>0.253</b>	0.181	-	-	-	-
PLATO-KAG*	12.75	0.233	<u>0.196</u>	<b>1.4327</b>	<b>1.2019</b>	<b>1.2019</b>	0.0962
PLATO-KAG <sup>+</sup>	13.77	0.231	<b>0.203</b>	1.2596	1.0192	1.0096	0.0385

**Table 5.3:** The automatic and human evaluation results on WoW seen (upper) and unseen (bottom) test sets. \* means this is our implementation results. + means our proposed method is applied. Note that PPL is generally not comparable among baselines, as their vocabularies are different. The best results are in **bold** and the best baseline results are underlined.

## Human Evaluation Results

The human evaluation results are presented in Figure 5.3. This evaluation has a kappa value of 1 for the context understanding dimension and 0.698 for the remaining dimensions<sup>4</sup>. Here we study the outputs of fine-tuned DialoGPT-large, as it performs the best on average.

**Context Understanding** From the first subplot of Figure 5.3, large models can reliably generate related knowledge for a given dialogue history, where around 90% tags are Related in both test sets.

**Tuning Effectiveness** The second diagram in Figure 5.3 shows that large models exhibit desirable behaviors after fine-tuning: it generates knowledge (Verifiable) in all cases.

**Fact-Checking** The rightmost three panels of Figure 5.3 demonstrate the factual correctness of the generated knowledge. As shown in the third panel, large models generate factually

<sup>4</sup>Since *fact-checking* is a fine-grained category of Verifiable in *tuning effectiveness*, we merge these two categories and compute the kappa value jointly.

correct (Supported) statements in most cases, though there is still around 10% of the chance to produce hallucinated information (Refuted). Among all the factually correct knowledge (the second last panel), more than 50% of them are Implicit Supported. This is exciting as large models are able to assemble multiple facts in their outputs, which cannot be substituted by simple search engine retrieval. This ability to summarize justifies the value of large models in serving as knowledge bases.

The last panel of Figure 5.3 checks whether NEI claims could be verified by common sense. There is a certain amount (39~57%) of NEI claims that are common sense. This observation advocates another advantage of utilizing large models as knowledge bases: they can provide common-sense information that lies behind the human mind, with no need for humans to explicitly write them down.

## 5.5.2 Response Generation Results

### Main Results

Table 5.3 is the response generation results of the WoW test sets. We can see that applying our proposed method to PLATO-KAG obtains the highest F1 score, even if our reimplemented PLATO-KAG baseline already performs much better than reported in the paper. On the other hand, our proposed method seems to lower the top-1 knowledge accuracy, i.e., P@1 drops from 0.266 to 0.254 in the seen test set and from 0.233 to 0.231 in the unseen test set. Note that PLATO-KAG is a model whose input consists of K knowledge candidates. If the ground truth knowledge is not ranked in the first place but presented in the top-K results, the model can still use the ground truth for the generation. In this case, the top-K knowledge accuracy is a more important metric for evaluating knowledge selection. Though not presented in Table 5.3, P@K increases from 0.681 to 0.690 in the seen test set and from 0.645 to 0.656 in the unseen test set. Table 5.4 displays the automatic evaluation results in Holl-E datasets. Similar to the results of WoW, our proposed method significantly outperforms the baseline systems in terms of the F1 score.

Table 5.3 also reports the human evaluation results of WoW. The kappa value of this

System	PPL↓	P@1↑	Unigram F1↑
SKT	52.02	<b>0.303</b>	0.295
PLATO-KAG	10.22	0.271	0.300
PLATO-KAG*	<u>5.816</u>	0.250	<u>0.310</u>
PLATO-KAG <sup>+</sup>	<b>5.495</b>	0.272	<b>0.320</b>

**Table 5.4:** *The evaluation results on Holl-E test set.*

System	Test Seen		Test Unseen	
	Unigram F1↑	Know. F1↑	Unigram F1↑	Know. F1↑
PLATO-KAG*	0.208	0.193	0.196	0.183
+ Noisy Train.	0.209	0.192	0.203	0.188
+ Post. Reweigh	0.211	0.200	0.203	0.193

**Table 5.5:** *The ablation study on WoW test sets.*

human evaluation is 0.415. In the seen test set, our strategy improves over baselines in nearly all metrics. However, our method degrades the performance of the unseen test set. In Section 5.5.2, we will show that our method put significantly more ground truth knowledge into the responses. In spite of that more knowledge helps to reduce hallucinations (from 0.0962 to 0.0385 as shown in Table 5.3), this could also lead to a degenerated result in human evaluation (Huang et al., 2021), as the knowledge makes the response far less interesting.

## Analysis

We conduct an ablation study in Table 5.5 for a better understanding of our proposed method. We additionally report Knowledge F1, the F1 score between the generated response and the ground truth knowledge (Lian et al., 2019; Shuster et al., 2021), which indicates how much ground truth knowledge is embedded into the response.

As shown in Table 5.5, all steps in our proposed method, including noisy training and posterior-based reweighing, contribute to the final performance. In particular, reweighing greatly improves Knowledge F1, which implies that it helps to select and incorporate ground truth knowledge into the response generation.

## 5.6 Summary

In this work, we show that large pre-trained models could serve as knowledge bases for unsupervised knowledge-grounded conversation. The study on the generated knowledge of large models has the following observations:

- Fine-tuning better elicits knowledge from large models than prefix-tuning.
- Knowledge pieces generated by pre-trained language models have a higher quality on unseen topics, while those from pre-trained dialogue models are better on seen topics.
- Large pre-trained models can synthesize common sense and summarize facts scattered on the web.

We also propose posterior-based reweighing and noisy training, which helps to incorporate the generated knowledge into the dialogue system. These simple strategies show a promising result over the strong baselines.

## Chapter 6

# Applications of LLM Evaluation: Multi-Hop Reasoning

### 6.1 Motivation

The field of long-context modeling has garnered significant attention due to its importance in applications that demand extensive comprehension and generation capabilities (Lewis et al., 2020; Liu et al., 2023b). Techniques for long-context modeling (Chen et al., 2023a; Peng et al., 2023; Chen et al., 2023c) have been proposed with encouraging results on established benchmarks (An et al., 2023; Bai et al., 2023).

Nevertheless, we have identified a gap in the performance of these models when it comes to multi-hop reasoning tasks, where a model must navigate and synthesize information from disparate sources to answer complex questions. Evidence from key benchmarks such as LongBench (Bai et al., 2023), as well as our experimental results in Section 6.4.3, indicate that these long-context LMs underperform compared to leading multi-hop reasoning systems (Zhang et al., 2023a). The reasons for this shortfall in multi-hop reasoning effectiveness are not yet fully understood.

We contend that the limitations in multi-hop reasoning observed in long-context LMs stem from two main issues: The inability to discern pertinent information within noisy

contexts (Liu et al., 2024) and the struggle to incorporate knowledge within the context effectively, particularly for smaller-scale models (Zheng et al., 2023a). To address these challenges, we introduce *Reasoning with Attributions*, a methodology that compels LMs to substantiate their reasoning by linking assertions to relevant context segments, such as citations (Gao et al., 2023b) or direct quotations (Menick et al., 2022). This approach not only guides LMs to perform targeted information retrieval to identify the position of relevant contexts, thereby reducing noise, but also ensures their responses are well-grounded in the source material. Our preliminary and comprehensive experimental findings, detailed in Sections 6.2.1 and 6.4.3, confirm the efficacy and resilience of this method across various multi-hop reasoning benchmarks.

Despite these advancements, smaller long-context LMs exhibit continued difficulties in reasoning. We explore the potential for these models to improve through learning to reason and attribute simultaneously. Utilizing ChatGPT (Brown et al., 2020) to annotate the multi-hop reasoning dataset MuSiQue (Trivedi et al., 2022), we create a specialized dataset *MuSiQue-Attribute* for fine-tuning models in this dual capacity. We propose a potent learning strategy that leverages multi-task learning and data augmentation to fully exploit these annotations. Our experiments with five long-context LMs across three multi-hop reasoning datasets and two general instruction-following datasets reveal that our fine-tuned Vicuna-7B model (Zheng et al., 2023b) surpasses similar-scale baselines by a substantial margin, i.e., more than 20 points on average, and even outperforms ChatGPT and Claude-instant on MuSiQue, albeit with a slight trade-off in other capabilities. This study illuminates a promising avenue to enhance the multi-hop reasoning capabilities of long-context LMs through a focus on attributions.

Our contributions can be summarized as follows:

- We introduce *Reasoning with Attributions*, an innovative reasoning paradigm that enhances both the performance and robustness of long-context LMs in multi-hop reasoning tasks.
- We provide *MuSiQue-Attribute*, an attribution-annotated multi-hop reasoning dataset

to support further research in this domain.

- We develop a bespoke learning strategy that incorporates novel auxiliary tasks for multi-task learning and employs tailored data augmentation techniques.
- Through rigorous testing on three multi-hop reasoning datasets and two general instruction-following benchmarks with both proprietary and open-source models, we demonstrate that our fine-tuned Vicuna-7B model achieves comparable multi-hop reasoning performance to ChatGPT with minimal impact on other capabilities. Additionally, our model shows resilience against varying degrees of contextual noise, underscoring the effectiveness of our methods.

## 6.2 Reasoning with Attributions

### 6.2.1 Pilot Study

The challenge of large language models becoming mired in irrelevant contexts, known as the “Lost in the Middle” phenomenon, has been documented across various NLP tasks, such as multi-document QA (Liu et al., 2024) and mathematical reasoning (Shi et al., 2023a). This issue is also apparent in multi-hop reasoning, which we illustrate later in Figure 6.2. Prior research has noted this problem but has not decoded the underlying mechanisms. For example, while Liu et al. (2024) found that introducing the query before the context can aid in better information retrieval from the context, they did not achieve an improvement in QA performance using this query-aware approach. As we suggest in Section 6.1, the reasons might extend beyond mere retrieval challenges to include complications in effectively applying the retrieved knowledge.

To tackle the issues outlined earlier, we introduce *Reasoning with Attributions*, a strategy that mandates language models to link the claims made during reasoning to specific sections of the provided context. This implicit requirement effectively decomposes a complex multi-hop question into two more manageable tasks: Pinpointing pertinent information within the context and constructing well-founded claims based on that information.

---

**Instruction:** Write an accurate and concise answer for ...

<Retrieve for the question>

Document [1](Title: David Myles (musician)): ...

Document [2](Title: Jamal Plays Jamal): ...

Document [3](Title: Top and Bottom Brass): ...

(Other retrieved documents are omitted.)

**Question:** What is the genre of the record label of the band that performed on the Crush Tour?

**Answer:**

---

*CoT:*

The Crush Tour is performed by the band Bon Jovi. The record label of Bon Jovi is Island Records. The genre of Island Records is jazz. The answer is: jazz ✓

---

*CoC:*

The Crush Tour is performed by the band Bon Jovi [8]. The record label of Bon Jovi is Island Records [17]. The genre of Island Records is jazz [19]. The answer is: jazz ✓

---

*CoQ:*

The Crush Tour is performed by the band Bon Jovi (“The Crush Tour is a third concert” [8]). The record label of Bon Jovi is Island Records (“Bounce is the eighth studio album by American” [17]). The genre of Island Records is jazz (“The Antidote is the debut album by English jazz” [19]). The answer is: jazz ✓

---

**Table 6.1:** An example of CoT and two of our reasoning with attribution methods: CoC and CoQ. We highlight the differences between these methods, e.g., answers are marked in green, citations are marked in orange and quotes are marked in blue.

We adapt the concept of Chain-of-Thought (CoT) (Wei et al., 2022b) reasoning to create two distinct variants aligned with our attribution-based approach: **Chain-of-Citation** (CoC) and **Chain-of-Quote** (CoQ). In CoC, models are prompted to reference citations corresponding to each step of the reasoning chain. CoQ goes further by requiring models to include direct quotations from the cited material for each reasoning step. An illustrative example highlighting the nuances between these methods is provided in Table 6.1.

The results of our preliminary study (Please refer to Section 6.4 for the setup), detailed in Table 6.2, compare the efficacy of CoT, CoC, and CoQ when applied to two proprietary long-context LMs: ChatGPT (Brown et al., 2020) and Claude-instant (Bai et al., 2022).

Model	MuSiQue		2Wiki		HotpotQA	
	EM	F1	EM	F1	EM	F1
<i>ChatGPT (gpt-3.5-turbo-1106)</i>						
+ AO	15.8	26.9	46.2	57.2	51.0	65.4
+ CoT	36.2	50.1	55.2	70.1	56.8	71.2
+ CoC	<b>37.0</b>	51.0	<b>55.4</b>	<b>71.1</b>	<b>58.6</b>	<b>73.4</b>
+ CoQ	36.4	<b>51.3</b>	54.0	68.7	55.4	70.2
<i>Claude-instant (claude-instant-1.2)</i>						
+ AO	26.2	39.4	47.0	57.5	<b>54.4</b>	<b>68.4</b>
+ CoT	26.0	37.9	40.8	52.3	20.2	26.3
+ CoC	<b>32.2</b>	<b>46.2</b>	<b>53.4</b>	<b>67.0</b>	54.2	68.3
+ CoQ	30.2	45.9	49.8	62.1	50.8	65.0

**Table 6.2:** Exact-Match (EM) and F1 scores of ChatGPT and Claude-instant with 5-shot prompting on multi-hop reasoning datasets, e.g., MuSiQue, 2WikiMultiHopQA (2Wiki for short) and HotpotQA. The best results are in **bold**. AO means models predict answers only.

Without further notice, ChatGPT always refers to `gpt-3.5-turbo-1106` and `claude-instant-1.2` for Claude-instant in this work. The findings suggest that both CoC and CoQ generally yield improvements over CoT, indicating that attribution-based reasoning enhances the precision and coherence of the models’ reasoning processes. CoQ appears to slightly underperform CoC, likely due to the increased complexity of producing exact quotations.

It is noteworthy that even in instances where CoT reduces the Answer Only (AO) performance, CoC is able to not only mitigate this decline but also surpass the AO baseline. This demonstrates the potential of CoC as a robust reasoning method. The success of our approach with various open-sourced models is further elaborated upon in Section 6.4.3. Based on these insights, we adopt CoC as our primary reasoning format in subsequent sections.

### 6.2.2 Dataset Curation

Our analysis, evidenced by the data in Tables 6.2 and 6.5, confirms that while reasoning with attributions holds promise, smaller open-source long-context language models significantly underperform compared to their proprietary counterparts in multi-hop reasoning tasks. To

Error Type	Portion
Incorrect Answer	58.44%
Non-Existent Attributions	12.56%
Incorrect Citations	9.80%
Repeated Citations	6.35%
Extreme Quotes	10.55%

**Table 6.3:** Incidence rates of different error types.

Entry	Value
#Max Words per Sample	3385
#Mean Words per Sample	1809.10
#Averaged Words per CoT Step	11.64
#Averaged Words per Quote	16.60
#Totoal Samples	1358
2-Hop Samples [%]	82.18%
3-Hop Samples [%]	14.06%
4-Hop Samples [%]	3.76%

**Table 6.4:** Statistics of MuSiQue-Attribute.

address this, we investigate whether training these models to perform attributions can boost their reasoning capabilities.

A hurdle in this process is the lack of attribution annotations within existing multi-hop reasoning benchmarks. To bridge this gap, we have generated new annotations by prompting ChatGPT with 5-shot CoQ. This has been done to create CoT with attributions for 5,000 instances randomly selected from the answerable training set of the MuSiQue dataset (Trivedi et al., 2022). Although CoC generally outperforms CoQ, we chose CoQ for annotation because it provides more detailed information. This richness is beneficial not only for evaluating the quality of the annotations but also proves advantageous for the fine-tuning processes discussed in Section 6.3.

After generating the annotations, we implemented a filtering process to exclude annotations with any of the following errors:

- **Incorrect Answer:** The model’s predicted answer does not align with the reference answer, which typically indicates an erroneous CoT.
- **Non-Existent Attributions:** Fabricated citations or quotes that do not correspond to

the actual context are indicative of model hallucination.

- **Incorrect Citations:** Citations do not match the manually identified supporting facts, suggesting flawed attributions.
- **Repeated Citations:** Redundant citations contravene the multi-hop requirement of sourcing from multiple documents.
- **Extreme Quotes:** Quotes that are either too terse (under five words) or excessively lengthy (spanning an entire document) lack utility.

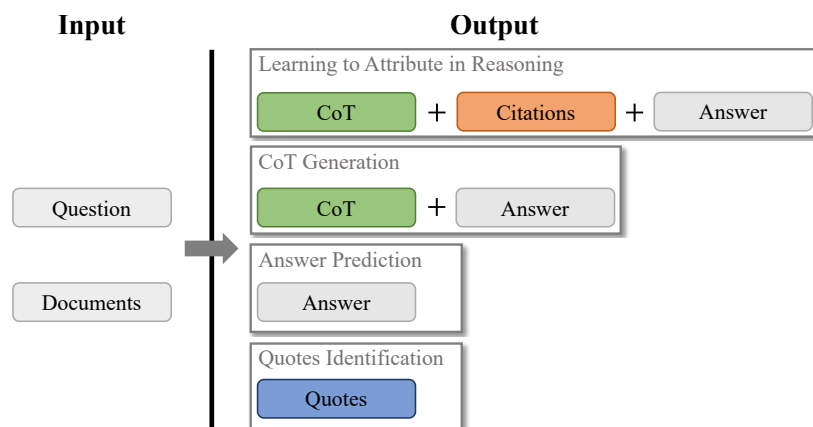
Table 6.3 presents the substantial incidence rates of each error type, which could negatively impact fine-tuning effectiveness. After filtering, we obtain a training dataset of 1,358 samples, referred to as *MuSiQue-Attribute*. The statistics of the *MuSiQue-Attribute* training set are outlined in Table 6.4. It is important to note that the hop distribution in *MuSiQue-Attribute* is skewed. This skewness arises both because generated CoT for questions with more hops is more prone to errors and because such questions represent a smaller fraction of the original *MuSiQue* training set.

### 6.3 Learning to Attribute in Reasoning

One intuitive approach to enhancing the multi-hop reasoning capabilities of LMs is to fine-tune them on our curated *MuSiQue-Attribute*, thereby teaching them to integrate attribution into their reasoning processes, specifically to generate CoC. Despite the simplicity of this method, our subsequent analysis in Section 6.4.4 demonstrates that this direct approach fails to produce robust results.

**Multi-Task Learning.** Beyond simply fine-tuning LMs on the *MuSiQue-Attribute* to learn to attribute in reasoning (denoted as **LA**), we propose three auxiliary tasks that serve as simplified analogs of LA. These tasks are designed to train LMs in conjunction with LA to enhance their proficiency in attribution-based reasoning:

- **Answer Prediction (AP for short):** This task focuses on direct answer prediction without the need for an explicit reasoning process. AP is intended to help LMs



**Figure 6.1:** Comparison of the proposed auxiliary tasks.

internalize the reasoning needed for straightforward questions where CoT is not required.

- **CoT Generation (CG for short):** In the CG task, models are trained to generate a CoT before providing an answer. This is aimed at developing LMs’ abilities to reason explicitly and methodically across multiple pieces of information for complex questions.
- **Quotes Identification (QI for short):** This task trains models to pinpoint critical quotes for reasoning. QI is designed to fine-tune the ability of LMs to filter out irrelevant details and zero in on the pertinent segments of text, thereby sharpening the accuracy of reasoning.

Figure 6.1 illustrates the distinctions between our primary LA task and the three auxiliary tasks.

**Data Augmentation.** A recognized limitation of direct fine-tuning on our MuSiQue-Attribute is the potential for models to develop biases, such as favoring certain locations of relevant documents (Liu et al., 2024), sensitive to a fixed number of documents, or accommodating only a narrow range of noise levels. To counteract these biases, we have devised the following data augmentation strategies:

- **Distractor Sampling:** By randomly selecting a varying number of irrelevant documents, we modify the positioning of relevant documents and the total document count within the context. This approach also mimics the fluctuating noise levels encountered in real-world scenarios, training language models to cope with noisy contexts effectively.
- **Document Shuffling:** Reordering the documents helps to remove any superficial positional cues that could lead to reasoning bias. For example, this ensures that models do not learn to associate the sequence of relevant documents with a fixed reasoning chain sequence.

These data augmentation strategies are applied in sequence for each training instance.

## 6.4 Experiments

### 6.4.1 Datasets

Our method’s effectiveness in multi-hop reasoning is assessed on the following datasets: **HotpotQA** (Yang et al., 2018b), **2WikiMultiHopQA** (**2Wiki** for short) (Ho et al., 2020), and **MuSiQue** (Trivedi et al., 2022). For each question, we provide a context composed of shuffled relevant and irrelevant documents. These irrelevant documents are the official retrieved distractor documents. We adopt the development and test sets from Trivedi et al. (2023) for evaluation, which contains 100 and 500 examples respectively. The results we present are the mean values from three separate trials, each with a distinct random seed.

To understand the broader impact of enhancing multi-hop reasoning on LMs’ overall capabilities, we also conduct evaluations on general instruction-following benchmarks, namely **MT-Bench** (Zheng et al., 2023b) and **AlpacaEval** (Li et al., 2023d).

### 6.4.2 Models

The following long-context baselines are chosen in our experiments.

- **ChatGPT** (Brown et al., 2020). We choose `gpt-3.5-turbo-1106`, which supports a

window size of 16K tokens.

- **Claude-instant (Bai et al., 2022)**. We choose `claude-instant-1.2`, which has a window size of 100K tokens.
- **LongChat (Li et al., 2023a)**. We use `longchat-7b-16k`, a 7B fine-tuned LLaMA model (Touvron et al., 2023a). It has a window size of 16K tokens.
- **LongLoRA (Chen et al., 2023c)**. We use `LongAlpaca-7B-16k`, which has a window size of 16K.
- **Vicuna (Zheng et al., 2023b)**. We use `vicuna-7b-v1.5-16k`, a 7B fine-tuned LLaMA-2 model (Touvron et al., 2023b). It supports a window size of 16K tokens.

We prompt all models with 5-shot to evaluate their multi-hop reasoning performance. These 5 demonstrations are randomly sampled from the 20 annotated training examples provided by Trivedi et al. (2023). If the input length exceeds the window size, we drop the last demonstration until the input length fits.

Our model **AttrLoRA** is fine-tuned on `vicuna-7b-v1.5-16k` with LoRA (Hu et al., 2022a), following hyper-parameters used in FastChat (Zheng et al., 2023b). For its training data, we perform augmentations to double the training data for all tasks in Section 6.3 except for the QI task. Note that we subsample same-sized instruction-tuning data from the Alpaca dataset (Taori et al., 2023) and mix it with the reasoning data. These instruction-tuning data serve the purpose of minimizing the risk of hampering other abilities Vicuna already possesses before fine-tuning.

### 6.4.3 Main Results

**Effectiveness and Robustness of Reasoning with Attributions.** The results in Table 6.5 underscore the efficacy of our CoC prompting across three multi-hop reasoning datasets, benchmarked against five baselines. In 77% of the evaluated cases (disregarding instances of near-zero model performance) CoC outperforms CoT. Notably, Claude-instant exhibits strong results with AO, and its performance diminishes when CoT is used. However, CoC not only mitigates this decline but also attains results on par with AO, demonstrating the

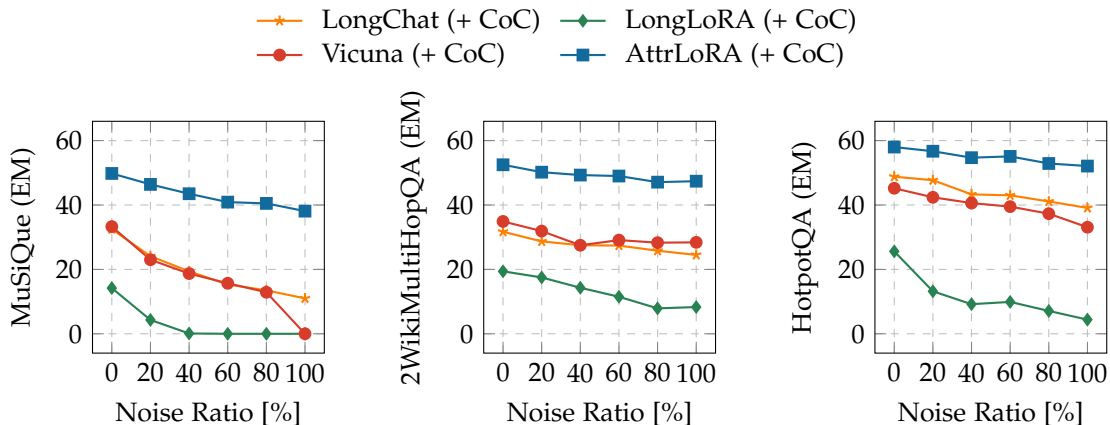
Model	MuSiQue				2Wiki					HotpotQA		
	Overall	2-Hop	3-Hop	4-Hop	Overall	Compositional	Inference	Comparison	Bridge-Comparison	Overall	Bridge	Comparison
ChatGPT												
+ AO	15.8	16.1	14.3	17.4	46.2	19.0	68.1	30.5	71.4	51.0	49.0	60.2
+ CoT	36.2	34.6	38.3	37.0	55.2	24.1	89.1	28.4	90.5	56.8	56.1	60.2
+ CoC	<u>37.0</u>	37.0	37.0	37.0	<u>55.4</u>	27.8	89.1	28.4	88.6	<u>58.6</u>	57.5	63.6
Claude-instant												
+ AO	26.2	25.2	27.3	27.2	47.0	19.0	68.9	32.5	70.5	54.4	52.7	62.5
+ CoT	26.0	27.2	27.9	19.6	40.8	20.3	80.7	15.7	58.1	20.2	23.5	4.5
+ CoC	32.2	32.7	30.5	33.7	53.4	36.7	90.8	22.3	81.9	54.2	55.3	48.9
LongChat												
+ AO	6.7	7.0	4.1	10.1	26.8	12.0	3.4	48.7	47.3	32.3	34.0	24.6
+ CoT	9.7	12.1	6.7	8.3	27.1	17.8	7.2	43.7	40.6	38.5	38.2	40.2
+ CoC	11.0	13.3	8.7	8.7	24.5	19.0	5.9	42.9	27.9	39.1	38.9	39.8
LongLoRA												
+ AO	0.2	0.4	0.0	0.0	7.7	9.0	4.6	13.2	1.3	16.9	16.3	19.3
+ CoT	0.0	0.0	0.0	0.0	15.1	5.9	0.8	28.9	27.3	11.4	11.0	13.3
+ CoC	0.0	0.0	0.0	0.0	8.3	3.2	1.3	15.7	14.6	4.4	4.0	6.4
Vicuna												
+ AO	0.1	0.1	0.0	0.0	20.5	5.2	5.1	31.9	47.6	22.3	24.2	13.6
+ CoT	0.0	0.0	0.0	0.0	27.7	14.7	7.6	48.5	43.5	30.5	32.0	23.1
+ CoC	0.0	0.0	0.0	0.0	28.4	20.6	7.2	49.3	35.2	33.1	34.7	25.8
AttrLoRA												
+ AO	32.9	35.7	27.1	34.8	<b>49.2</b>	48.7	28.7	54.9	59.0	51.5	51.0	54.2
+ CoT	37.9	41.6	35.5	31.9	46.8	48.4	25.3	53.5	52.4	50.9	51.2	49.2
+ CoC	<b>38.1</b>	42.7	35.7	29.7	47.4	46.0	27.4	57.7	53.3	<b>52.1</b>	52.6	49.6

**Table 6.5:** Exact-Match (EM) results on three multi-hop reasoning datasets. The best small-scale long-context LM results are in **bold** and the best baseline results are underlined.

robustness of attribution-based reasoning.

**Performance of AttrLoRA Against Proprietary Models.** Table 6.5 presents a zero-shot performance comparison between our AttrLoRA and five-shot outcomes from various baselines. AttrLoRA surpasses baselines of comparable scale by an average margin of over 20 points. It exceeds the performance of two notable proprietary models on MuSiQue and delivers closely competitive results on the other two benchmarks.

In particular, AttrLoRA with AO achieves superior results on 2Wiki and surpasses CoT on HotpotQA. This can be attributed to the relative simplicity of these datasets, where explicit reasoning does not significantly enhance performance. For instance, CoT’s advantage is noticeably smaller on these datasets compared to MuSiQue for both the ChatGPT and Claude-instant. Additionally, according to [Jiang and Bansal \(2019\)](#), over half of the “bridge-type” questions in HotpotQA contain shortcuts, which can locate the answer by keyword matching, circumventing the need for the intended two-hop reasoning. Similarly, 2Wiki’s predictable nature, due to its question construction from a limited set of rules, simplifies the task for LMs. Another contributing factor is that AttrLoRA is trained on MuSiQue-Attribute,



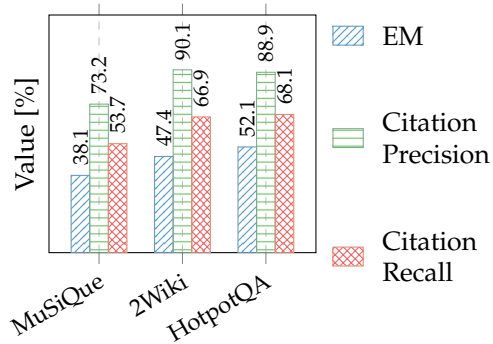
**Figure 6.2:** Exact-Match (EM) results of different models under various noise levels in three multi-hop reasoning datasets. Note that all models except our AttrLoRA use 5-shot prompting. A higher noise ratio indicates more distractors, i.e., irrelevant documents, are presented in the context of both the test instance and the demonstrations.

Model	MT-Bench (Score)	AlpacaEval (Win Rate)
ChatGPT	8.245	9.178%
Claude-instant	8.131	15.664%
Vicuna	6.068	5.415%
+ Alpaca Data	4.850	3.287%
AttrLoRA	4.978	3.106%

**Table 6.6:** Results on general instruction-following benchmarks. “+ Alpaca Data” is a Vicuna-7B model continued fine-tuning on Alpaca data.

which does not encompass the full range of question types found in 2Wiki and HotpotQA, such as “comparison-type” questions.

**Resilience of AttrLoRA to Noisy Contexts.** A key aspect of AttrLoRA is its robustness to contextual noise. To investigate this, Figure 6.2 illustrates AttrLoRA’s performance against varying degrees of synthesized noise. This synthesized noise is implemented by adding varied numbers of random irrelevant documents to the context. The data indicates that while the performance of baseline models markedly declines with increased noise, e.g., Vicuna drops by over 30 points on MuSiQue, AttrLoRA shows greater resilience, with a reduction of only about 10 points.



**Figure 6.3:** Multi-hop reasoning performance vs. citation precision and recall of AttrLoRA.

**Impact of Attribution Learning on General Abilities.** Our investigation extends beyond multi-hop reasoning to examine how attribution learning affects AttrLoRA’s general instruction-following capabilities post-fine-tuning, as compared to the Vicuna baseline. The results in Table 6.6 from two instruction-following benchmarks reveal that fine-tuning slightly compromises abilities beyond multi-hop reasoning in a 7B model due to capacity constraints. However, a closer analysis reveals that over 98% of the performance decrease is attributed to fine-tuning with Alpaca data (“+ Alpaca Data”), while multi-hop reasoning data incurs less than a 2% detriment. This is because the quality of Alpaca data is inferior to Vicuna’s, with the former being single-turn GPT-3 synthesized and the latter comprising multi-turn human-bot conversations.

#### 6.4.4 Analysis

**Attribution Quality of AttrLoRA.** Drawing from the insights of Gao et al. (2023b), we scrutinize the citation precision and recall for AttrLoRA, as presented in Figure 6.3. The model demonstrates high precision, indicating its proficiency in correctly attributing statements to pertinent documents. Nonetheless, the moderate recall highlights that AttrLoRA does not consistently identify all relevant documents, a potential consequence of the disconnected reasoning patterns observed in MuSiQue-Attribute.

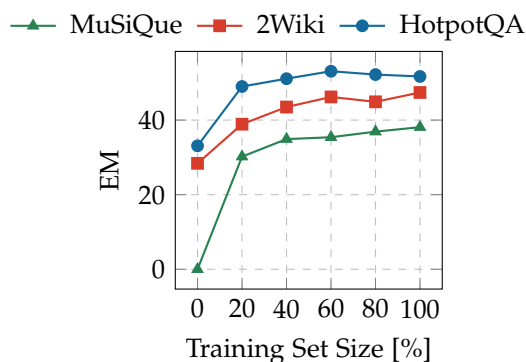
Model	MuSiQue	2Wiki	HotpotQA
Vicuna (5-Shot)	0.00	28.4	33.1
+ AP	32.3	47.8	52.1
+ CG	37.3	45.3	50.7
+ LA	37.3	46.9	52.1
+ QI	38.1	47.4	52.1

**Table 6.7:** Ablation study on multi-task learning.

Model	MuSiQue	2Wiki	HotpotQA
Vicuna (5-Shot)	0.00	28.4	33.1
+ AP	27.4	44.9	50.9
+ Augmentation	32.3	47.8	52.1
+ CG	28.1	37.7	49.4
+ Augmentation	30.5	37.3	50.4
+ LA	29.1	38.8	49.0
+ Augmentation	30.1	37.1	49.9

**Table 6.8:** Ablation study on data augmentation.

**The Effectiveness of Multi-Task Learning.** Our ablation study in Table 6.7 assesses our multi-task learning approach. Results indicate a marked enhancement in Vicuna’s reasoning capabilities upon fine-tuning with our dataset (“+ AP”). However, explicitly training Vicuna to generate CoT (“+ CG”) yields mixed outcomes: It benefits performance on MuSiQue but adversely affects results on 2Wiki and HotpotQA. This discrepancy can be attributed to the relative ease of the latter datasets, where simpler questions and shortcuts reduce the effectiveness of complex reasoning strategies, as discussed in Section 6.4.3. Importantly, integrating the LA task (“+ LA”) mitigates the performance drops associated with CoT and notably boosts MuSiQue scores. This implies that attributions are instrumental in enabling the model to reason over complicated questions without compromising its ability to handle simpler queries. Finally, the addition of the QI task (“+ QI”) appears to further refine the model’s multi-hop reasoning proficiency, underscoring the value of our multi-task learning framework.



**Figure 6.4:** *The impact of scaling fine-tuning data size.*

**The Effectiveness of Data Augmentation.** We explore the impact of our data augmentation strategy, intentionally omitting the QI task, as it alone is insufficient for training models to conduct multi-hop reasoning. The data in Table 6.8 demonstrates that including augmented data generally enhances model performance across various datasets. However, augmenting CG and LA data does not yield improvements on 2Wiki. In this case, the model readily learns from a limited amount of annotated data due to the simplicity of the automatically generated questions within 2Wiki. Conversely, on MuSiQue and HotpotQA, which feature more complex and varied human-crafted questions, the model benefits from exposure to a larger dataset to accommodate the diversity of question formulations.

**The Effectiveness of Scaling Fine-Tuning Data.** In Figure 6.4, we investigate how the expansion of fine-tuning data influences model performance. It is evident that incorporating additional data steadily enhances performance on MuSiQue and 2Wiki, while optimal results are attained with just 60% of our data for HotpotQA. This fact suggests that more complex question answering, involving additional reasoning steps like MuSiQue and 2Wiki, demands a larger dataset. An intriguing discovery is that using a mere 20% of our data achieves approximately 85% of the peak performance. This highlights the efficiency of fine-tuning: Even a modest subset of multi-hop reasoning examples can significantly boost the model’s reasoning capabilities.

<p><b>Question:</b> Who is the mascot of the university related to Randy Conrads?</p> <p><b>Document</b> [4](Title: Benny Beaver): <a href="#">Benny Beaver is the official mascot of Oregon State University</a> and winner of the 2011 Capital One Mascot of the Year write - in campaign. ...</p> <p><b>Document</b> [7](Title: Randy Conrads): <a href="#">Randy Conrads attended Oregon State University</a>, graduating in 1972 with a bachelor’s degree in industrial engineering. ... (Other irrelevant documents are omitted.)</p> <p><b>Vicuna:</b> <a href="#">the university.</a> ✘</p> <p><b>AttrLoRA:</b> Randy Conrads attended Oregon State University [7]. The mascot of Oregon State University is Benny Beaver [4]. The answer is: <a href="#">Benny Beaver</a> ✔</p>	<p><b>Question:</b> What is the record label of the co-writer and recording artist of Permission to Fly?</p> <p><b>Document</b> [8](Title: No Ordinary Girl): <a href="#">No Ordinary Girl is the debut album by Jordan Pruitt, released in the United States on February 6, 2007 by Hollywood Records.</a> The album debuted and peaked at number sixty-four ...</p> <p><b>Document</b> [19](Title: Permission to Fly): <a href="#">Permission to Fly is the second and final studio album by American singer-songwriter Jordan Pruitt.</a> On July 22, 2008, the album was released ... (Other irrelevant documents are omitted.)</p> <p><b>Vicuna:</b> The answer is: <a href="#">The answer is a question.</a> ✘</p> <p><b>AttrLoRA:</b> The co-writer and recording artist of “Permission to Fly” is Jordan Pruitt [19]. The record label of Jordan Pruitt is Hollywood Records [8]. The answer is: <a href="#">Hollywood Records</a> ✔</p>
--	--

**Table 6.9:** Two examples from MuSiQue test set where Vicuna fails and AttrLoRA succeeds. We manually annotate quotes in blue and citations in orange. Correct predictions are marked by ✔ and incorrect ones are by ✘.

**Case Study.** Table 6.9 presents a comparative case study where Vicuna and AttrLoRA are both prompted to generate CoC. Within the provided examples, AttrLoRA successfully produces coherent CoT and precisely attributes each claim. In contrast, Vicuna yields answers without engaging in an explicit reasoning process.

## 6.5 Summary

This study demonstrates that long-context LMs face challenges with multi-hop reasoning within noisy contexts. We introduce a reasoning paradigm that incorporates attributions, which significantly improves the reasoning capabilities of long-context LMs. Alongside, we contribute a new dataset annotated with attributions and study training strategies tailored for multi-hop reasoning. Our comprehensive experiments across five models and five benchmarks validate the superiority of our approach in enhancing multi-hop reasoning performance.

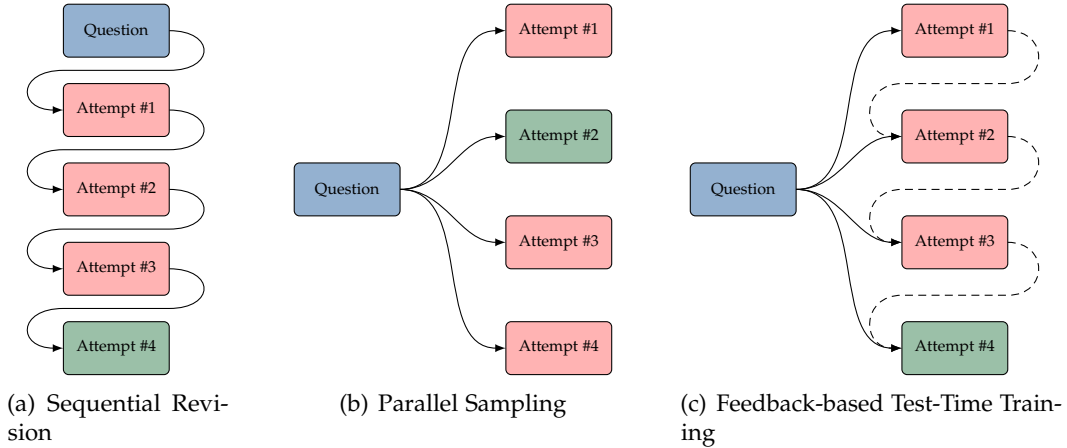
## Chapter 7

# Applications of LLM Evaluation: Test-Time Scaling

### 7.1 Motivation

Leveraging external feedback from interactions with the environment during test time has emerged as a promising approach for large language models (LLMs). This includes applications such as LLM-based agents (Yao et al., 2023; Shinn et al., 2023) and, more recently, test-time scaling (Wu et al., 2024b; Snell et al., 2024; Liu et al., 2025). Such methods further enhance the potential of LLMs to solve challenging tasks, e.g., Olympiad-level math problems (Guan et al., 2025) and competitive programming (OpenAI et al., 2025).

Significant progress in this area typically falls into two categories (Snell et al., 2024), as illustrated in Figure 7.1: sequential revision and parallel sampling. Sequential revision methods (Shinn et al., 2023; Madaan et al., 2023) incorporate previous attempts into the LLM’s context, while parallel sampling methods (Brown et al., 2024; Xie et al., 2023) generate new attempts independently of prior failures. However, both approaches have notable limitations. Sequential revision is computationally expensive due to long context lengths and faces challenges (Muennighoff et al., 2025), such as position bias (Liu et al., 2024) and attention noise (Ye et al., 2024b). In contrast, parallel sampling, while efficient, fails to



**Figure 7.1:** Comparison between sequential revision, parallel sampling, and feedback-based test-time training. ■ is the failed attempt and ■ is the successful attempt.  $\rightarrow$  indicates the LLM generation with the input on the left of the arrow and the output on the right.  $\dashrightarrow$  denotes the LLM training, where the left of the arrow is the training data.

learn from previous errors (Brown et al., 2024). Unlike these paradigms, human reasoning follows a different pattern: humans store recent experiences in “fast weights” (Ba et al., 2016), enabling them to neither revisit past errors explicitly nor start each attempt without any prior knowledge. Recent research suggests that the weights of neural networks could serve as a natural memory mechanism during test time (Wang et al., 2024b).

Building on these observations, we propose a novel paradigm that leverages Test-Time Training (TTT) (Sun et al., 2020, 2023b) to store past experiences in model weights rather than in the context. This approach bridges the gap between sequential revision and parallel sampling by indirectly incorporating knowledge into the LLM without disrupting in-context reasoning. Specifically, we introduce Feedback-based Test-Time Training (FTTT), which employs a carefully designed TTT task enriched with feedback through self-reflection. We demonstrate that FTTT improves test-time computation scalability on two mathematical reasoning and two code generation datasets, using Llama-3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023).

Inspired by advancements in learning to optimize (Chen et al., 2022), we explore training learnable test-time optimizers to yield Pareto-optimal cost-performance tradeoffs. Our

proposed learnable optimizer, OPTUNE, is a lightweight neural network that predicts weight updates from the gradients of the previous attempt. Unlike traditional parameter-efficient fine-tuning (PEFT) methods, OPTUNE works on the gradient rather than the activation space. Experiments on three reasoning datasets and two different LLMs demonstrate the effectiveness of OPTUNE, outperforming five widely used PEFT baselines.

## 7.2 Feedback-based Test-Time Training

### 7.2.1 The Test-Time Training Task

The problem of exploiting test-time feedback is as (Shinn et al., 2023): given a question  $Q$ , a model  $M$  attempts to solve  $Q$  within a budget of  $N$  attempts. A verifier  $V$  evaluates each attempt, such as the  $n$ -th attempt  $A_n$ , and provides feedback  $V(A_n)$ . This work focuses on binary verifiers, which determine whether  $A_n$  is correct. These verifiers are well-established, rule-based systems that are both cost-effective and efficient to evaluate.

When the model generates attempts sequentially, our goal is to enable  $M$  to learn from previous attempts to improve subsequent ones. To achieve this, we frame learning from previous attempts as a training problem: at each step  $n$ , we optimize  $M$  using  $Q$ ,  $A_n$ , and  $V(A_n)$ , aiming for  $M$  to generate a better  $A_{n+1}$ . This way internalizes the past attempts into weights for efficient inference of  $A_{n+1}$ . As a result, the sequence of attempts can be viewed as an  $N$ -step optimization process.

A key challenge is designing an effective supervised task using  $Q$ ,  $A_n$ , and  $V(A_n)$  to improve the model’s ability to solve  $Q$ . We build on the intuition that *a model capable of judging the correctness of a solution should also be able to solve the question itself*. Concretely, given  $Q$  and  $A_n$ , we train  $M$  to predict verbal feedback  $F$  that aligns with  $V(A_n)$ . This leads to our FTTT loss:

$$\mathcal{L}_{\text{FTTT}}(Q, A_n) = -\frac{1}{l_0} \log M_{n-1}(F | Q, A_n) \quad (7.1)$$

where  $l_0$  is the length of  $F$  and  $M_0$  denotes the raw LLM. In this work,  $F$  is set to “Your answer is incorrect.” when  $V(A_n)$  implies an incorrect  $A_n$ .

## 7.2.2 Self-Reflected Feedback

Since we are working with a binary verifier, the learning signal is limited at each interaction. Previous research suggests that LLMs can self-correct errors when provided with external signals (Huang et al., 2024). Inspired by this, we aim to enhance the learning signal by leveraging the model to generate silver-standard training labels.

We first sample a reflection  $R_n$  from the model given  $Q, A_n, F$  and the instruction  $P$ :

$$R_n \sim M_0(R | Q, A_n, F, P) \quad (7.2)$$

In practice, we use  $M_0$  to generate  $R_n$  to mitigate the risk of degraded self-reflection ability after training. The auxiliary loss is then defined as:

$$\mathcal{L}_{\text{aux}}(Q, A_n, R_n) = -\frac{1}{l_n} \log M_{n-1}(R_n | Q, A_n, F) \quad (7.3)$$

where  $l_n$  is the length of  $R_n$ . Eq. 7.3 can be interpreted as a sequence-level distillation loss (Kim and Rush, 2016), where knowledge from the raw model  $M_0$  is distilled into the trained model  $M_{n-1}$  to prevent overfitting. Finally, the overall loss is as:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{FTTT}} + \mathcal{L}_{\text{aux}} \quad (7.4)$$

Below is a training example with self-reflection, where underlined sentences are the training target:

Training Example with Self-Reflection
<p><b>User:</b> Solve the following math problem ...</p> <p><b>Assistant:</b> ... the final answer is: ...</p> <p><b>User:</b> <u>Your answer is incorrect. Here is the summary of the mistakes in the previous solution ...</u></p>

The underlined sentence corresponds to  $F$  in Eq. 7.1 and the wave-underlined sentence represents  $R_n$  for Eq. 7.3. Algorithm 1 summarizes our FTTT.

---

**Algorithm 1** FEEDBACK-BASED TTT

---

**Require:** The question  $Q$ , the model  $M_0$ , the verifier  $V$ , the budget  $N$ , the verbal feedback  $F$ , the instruction  $P$

```
1:  $n \leftarrow 1$ 
2: while  $n \leq N$  do
3:    $A_n \sim M_{n-1}(A | Q)$ 
4:   if  $V(A_n)$  is passed then
5:     return  $A_n$ 
6:   else
7:     Compute the loss  $\mathcal{L}$  using Eq. 7.1
8:     if enable self-reflection then
9:        $R_n \sim M_0(R | Q, A_n, F, P)$ 
10:      Compute the loss  $\mathcal{L}_{\text{aux}}$  using Eq. 7.3
11:       $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\text{aux}}$ 
12:    end if
13:    Update  $M_{n-1}$  using  $\mathcal{L}$  to get  $M_n$ 
14:  end if
15:   $n \leftarrow n + 1$ 
16: end while
17: return  $A_N$ 
```

---

**Discussion.** As shown in Table 7.1, FTTT combines the strengths of sequential revision and parallel sampling. Like sequential revision, it leverages memory (i.e., weights) to store past failed attempts, while avoiding the length generalization issues as in parallel sampling. Figure 7.1 highlights the advantages of FTTT from a probabilistic graphical model perspective, where both sequential revision and parallel sampling are special cases of FTTT with additional independence assumptions. In particular, sequential revision assumes that attempts form a Markov process, whereas parallel sampling treats each attempt as independent.

In terms of efficiency, FTTT is highly optimized, requiring one additional backward propagation computed in parallel for all tokens in one attempt, whose cost is negligible. The dominant overhead of FTTT is generating  $R_n$ . However, FTTT is still much faster than sequential revision (see Section 7.4.2), as  $R_n$  is typically short.

Moreover, FTTT closely resembles vanilla TTT (Sun et al., 2020), but with additional inputs beyond  $Q$ , such as  $A_n$ ,  $V(A_n)$ , and  $R_n$ , as defined in the problem. Consequently, it inherits the same convergence guarantees as TTT.

Method	Self Reflection	Memory	Length Generalization
Revision (Snell et al., 2024)	✗	✓	✗
Self-Refine (Madaan et al., 2023)	✓	✓	✗
Best-of-N (Brown et al., 2024)	✗	✗	✓
Beam Search (Ow and Morton, 1988)	✗	✗	✓
Guided Beam Search (Xie et al., 2023)	✓	✗	✓
<b>FTTT (ours)</b>	✓	✓	✓

Table 7.1: Comparing the advantages and drawbacks of FTTT and related works.

## 7.3 A Learnable Test-Time Optimizer

### 7.3.1 The Learning to Optimize Problem

Although FTTT achieves success (see Section 7.4.2), it simply accumulates the gradients of the feedback received so far to update the weights. This raises the question: can we design a better test-time optimizer that more effectively exploits feedback?

Motivated by learning to optimize (Chen et al., 2022), we adopt a neural network as the test-time optimizer. Concretely, this learnable test-time optimizer is formulated as  $f_{\theta}(Q, \{A_i, V(A_i)\}_{i=1}^n)$ , which predicts updates for all LLM weights based on the previous  $n$  attempts, and  $\theta$  is the optimizer parameter. However, this direct formulation leads to prohibitively large networks due to high-dimensional input and output spaces. For a maximum number of  $m$  tokens per attempt and an  $l$ -layer LLM, the input space grows to  $n \times m \times l$ , even when updating only a scalar (we exclude the token count of  $Q$ , as it is significantly smaller than  $m$ ). Since updates for all weight matrices across all layers are predicted jointly, the dimensionality of the input and output spaces becomes unmanageable. We therefore simplify  $f_{\theta}$  by introducing the following assumptions:

- (A1): **Markov Property:** The latest attempt captures all relevant information from previous attempts.
- (A2): **Independent Update:** The optimizer predicts updates for each parameter independently, similar to conventional optimizers.

**A1** eliminates the dependency on  $n$  and **A2** enables updates to be predicted independently for each weight, significantly reducing the size of the output space. The learnable test-time optimizer now becomes  $\tilde{\nabla}_{W_i} = f_{\theta_{W_i}}(Q, A_n, V(A_n))$ , where it predicts the update for the weight  $W_i$  in the  $i$ -th layer based on  $Q$  and the latest attempt  $A_n$ .

To train all  $f$ , we define the following loss:

$$\mathcal{L}_{\text{meta}} = -\log M(\hat{A} \mid Q, \{W + \tilde{\nabla}_W \mid \forall W \in \mathcal{W}\}) \quad (7.5)$$

where  $\hat{A}$  is the ground-truth for  $Q$  and  $\mathcal{W}$  is the set of LLM weights. Eq. 7.5 encourages  $f$  to predict updates that increase the likelihood of generating the correct answer after applying the updates.

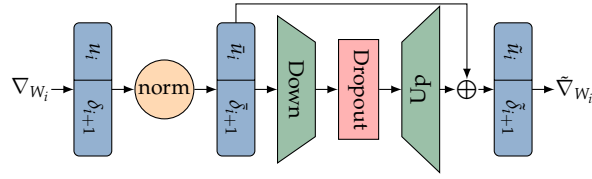
### 7.3.2 A Parameter-Efficient Architecture in The Gradient Space

Given the limited learning signal at test time, we design the learnable optimizer to be parameter-efficient to alleviate overfitting. However, the input and output spaces of  $f_{\theta_{W_i}}(Q, A_n, V(A_n))$  are large due to their lengths, making even a simple linear projection parameter-intensive. Additionally,  $V(A_n)$  may be heterogeneous to  $Q$  and  $A_n$ , e.g., a scalar, posing challenges for modeling.

Inspired by the success of FTTT in Section 7.2 and recent works (Mitchell et al., 2022; Wang et al., 2024b), we propose a parameter-efficient architecture in the gradient space as the learnable optimizer.

**Gradient-based Input Compression.** Instead of directly inputting  $Q$ ,  $A_n$ , and  $V(A_n)$ , we first project them into the gradient space, since recent work suggests that long context can be effectively compressed by gradients (Wang et al., 2024b). This way reduces the token count  $m$  in  $A_n$  to a constant and unifies the spaces of  $Q$ ,  $A_n$ , and  $V(A_n)$  to ease the modeling. To compress  $Q$  and  $A_n$ , we use the next token prediction loss, while for  $V(A_n)$ , we include  $\mathcal{L}_{\text{FTTT}}$  in Eq. 7.1. The final loss for compressing the optimizer input is:

$$\mathcal{L}_{\text{compress}} = -\frac{1}{m} \log M(A_n \mid Q) + \mathcal{L}_{\text{FTTT}} \quad (7.6)$$



**Figure 7.2:** The model architecture of OPTUNE.

The input of  $f_{\theta_{W_i}}$  to predict the update of  $W_i$  now is the gradient  $\nabla_{W_i}$  of  $\mathcal{L}_{\text{compress}}$  w.r.t.  $W_i$ . Consequently,  $f_{\theta_{W_i}}$  receives a fixed-size tensor as input rather than a variable-length sequence.

**Gradient Decomposition.** Although  $f_{\theta_{W_i}}$  operates on a smaller space after compression, the dimensionality of the gradient space remains large for direct processing. We utilize the observation that  $\nabla_{W_i} \in \mathbb{R}^{d \times d}$  (assuming  $W_i \in \mathbb{R}^{d \times d}$ ) can be decomposed into two vectors to further reduce the dimensionalities: the input to a linear projection with weight  $W_i$ ,  $u_i \in \mathbb{R}^{d \times 1}$ , and the gradient of  $\mathcal{L}_{\text{compress}}$  w.r.t. the output of the projection,  $\delta_{i+1} \in \mathbb{R}^{d \times 1}$  (Mitchell et al., 2022). In this framework,  $f_{\theta_{W_i}}$  takes the decomposed  $u_i$  and  $\delta_{i+1}$  as its input and predicts  $\tilde{u}_i$  and  $\tilde{\delta}_{i+1}$ . The update is then reconstructed as  $\tilde{\nabla}_{W_i} = \tilde{\delta}_{i+1} \tilde{u}_i^T$ . This approach reduces the dimension from  $d^2$  to  $2d$ .

**Model Architecture.** The architecture of  $f_{\theta_{W_i}}(u_i, \delta_{i+1})$ , named OPTUNE, is shown in Figure 7.2 and defined as follows:

$$[\tilde{u}_i, \tilde{\delta}_{i+1}] = \text{Norm}([u_i, \delta_{i+1}]) \quad (7.7)$$

$$h_i = \theta_2 \text{Dropout}(\theta_1 [\tilde{u}_i, \tilde{\delta}_{i+1}]) \quad (7.8)$$

$$[\tilde{u}_i, \tilde{\delta}_{i+1}] = h_i + [\tilde{u}_i, \tilde{\delta}_{i+1}] \quad (7.9)$$

where  $\theta_1 \in \mathbb{R}^{r \times 2d}$  and  $\theta_2 \in \mathbb{R}^{2d \times r}$  are the optimizer parameters with  $r \ll d$ .  $[\cdot]$  denotes the vector concatenation. Norm normalizes  $u_i$  and  $\delta_{i+1}$  to have zero mean and unit variance separately. Dropout is the dropout regularization (Srivastava et al., 2014). In practice,  $\theta_1$  and  $\theta_2$  are shared across all weights with the same shape. OPTUNE is similar to the Bottleneck

Adapter (Houlsby et al., 2019), with the key difference that its input is gradients and its output is the weight update. As such, OPTUNE can also be regarded as a specialized PEFT technique tailored to reasoning.

## 7.4 Experiments

### 7.4.1 Setup

**Datasets.** We evaluate both baselines and our method on math and coding reasoning tasks: (a) Mathematical reasoning: MATH (Hendrycks et al., 2021b) and GSM8K (Cobbe et al., 2021), using the test split from Lightman et al. (2024) for MATH. (b) Code generation: MBPP (Austin et al., 2021) and HumanEval (Chen et al., 2021). For all datasets, we report results on subsets where models fail with greedy decoding. We use Exact Match as the evaluation metric as well as the verifier for math tasks and Pass@1 for code.

**Models.** We conduct experiments with Llama-3.1-8B-Instruct (Dubey et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). We evaluate both models with zero-shot prompting and follow the official instructions when evaluating the model on each dataset<sup>1</sup>.

**Baselines.** We compare FTTT with the following test-time scaling methods:

- **Beam Search** (Ow and Morton, 1988) is a popular search algorithm that approximates the most confident model prediction.
- **Self-Consistency** (Wang et al., 2023) samples multiple predictions and selects the most frequent answer.
- **Best-of-N** (Brown et al., 2024) samples  $N$  predictions independently and picks the best one based on external feedback.

---

<sup>1</sup><https://huggingface.co/datasets/meta-llama/Llama-3.1-8B-Instruct-evals>

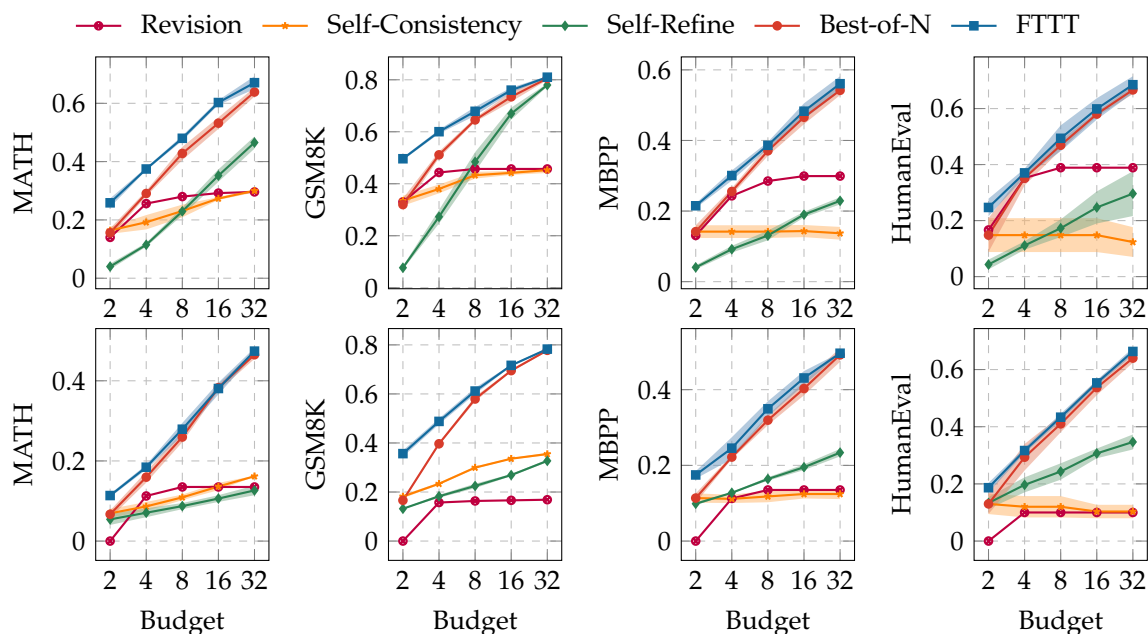
Method	MATH	GSM8K	MBPP	HumanEval	Avg.
Llama-3.1-8B-Instruct					
Revision (Snell et al., 2024)	0.2960	0.4570	0.2991	0.3889	0.3603
Beam Search (Ow and Morton, 1988)	0.2560	0.4842	0.1869	0.2407	0.2920
Self-Consistency (Wang et al., 2023)	0.3000 <sub>0.0057</sub>	0.4525 <sub>0.0111</sub>	0.1371 <sub>0.0180</sub>	0.1235 <sub>0.0531</sub>	0.2533
Self-Refine (Madaan et al., 2023)	0.4693 <sub>0.0207</sub>	0.7828 <sub>0.0064</sub>	0.2305 <sub>0.0088</sub>	0.2963 <sub>0.0800</sub>	0.4447
Best-of-N (Brown et al., 2024)	0.6427 <sub>0.0154</sub>	<u>0.8069</u> <sub>0.0043</sub>	<u>0.5452</u> <sub>0.0154</sub>	0.6728 <sub>0.0087</sub>	0.6669
FTTT	<u>0.6707</u> <sub>0.0222</sub>	<b>0.8100</b> <sub>0.0037</sub>	<b>0.5607</b> <sub>0.0212</sub>	<b>0.6852</b> <sub>0.0302</sub>	0.6817
+ w/o Self-Reflected Feedback	<b>0.6720</b> <sub>0.0113</sub>	0.8054 <sub>0.0133</sub>	0.5405 <sub>0.0117</sub>	<u>0.6790</u> <sub>0.0087</sub>	0.6742
Mistral-7B-Instruct-v0.3					
Revision (Snell et al., 2024)	0.0497	0.1686	0.1351	0.1000	0.1134
Beam Search (Ow and Morton, 1988)	0.1783	0.4537	0.1318	0.1600	0.2310
Self-Consistency (Wang et al., 2023)	0.1618 <sub>0.0011</sub>	0.3549 <sub>0.0052</sub>	0.1243 <sub>0.0135</sub>	0.1033 <sub>0.0236</sub>	0.1861
Self-Refine (Madaan et al., 2023)	0.1287 <sub>0.0139</sub>	0.3312 <sub>0.0078</sub>	0.2347 <sub>0.0091</sub>	0.3533 <sub>0.0309</sub>	0.2620
Best-of-N (Brown et al., 2024)	0.4688 <sub>0.0138</sub>	0.7807 <sub>0.0055</sub>	<b>0.4962</b> <sub>0.0171</sub>	0.6500 <sub>0.0141</sub>	0.5989
FTTT	<u>0.4733</u> <sub>0.0087</sub>	<u>0.7820</u> <sub>0.0045</sub>	<b>0.4962</b> <sub>0.0015</sub>	<u>0.6633</u> <sub>0.0125</sub>	0.6037
+ w/o Self-Reflected Feedback	<b>0.4876</b> <sub>0.0133</sub>	<b>0.7858</b> <sub>0.0021</sub>	<u>0.4941</u> <sub>0.0040</sub>	<b>0.6833</b> <sub>0.0205</sub>	0.6127

**Table 7.2:** Experimental results on four datasets with a budget of 32. For stochastic algorithms, we report the mean of three runs with different random seeds and standard deviation in the subscript. **Bold** entries are the best results, and underlined entries are the second-best results.

- **Revision** (Snell et al., 2024) iteratively refines answers by conditioning the model on previous attempts.
- **Self-Refine** (Madaan et al., 2023) alternates between self-critique and refinement. We select the best solution based on feedback.

For sampling-based methods, we use nucleus sampling (Holtzman et al., 2020) with a temperature of 0.6 and  $p = 0.95$ , following Brown et al. (2024). All methods are allocated a budget of 32. For FTTT, we fine-tune the model with LoRA (Hu et al., 2022a), using a rank of 4 and a dropout ratio of 0.05. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of  $1e-5$ , except for Mistral-7B-Instruct-v0.3 in coding tasks, where we use  $2e-5$ .

As OPTUNE is a specialized PEFT method, we compare it with the following PEFT approaches: **Adapter** (Houlsby et al., 2019), **(IA)**<sup>3</sup> (Liu et al., 2022a), **LoRA** (Hu et al., 2022a) and **LN-Tuning** (Zhao et al., 2024a). We also include **full fine-tuning** that updates all LLM weights. For OPTUNE,  $r = 16$  and the dropout ratio is 0.1. We only apply OPTUNE to the



**Figure 7.3:** The scaling trends of different methods under varying budgets. The colored area around the line denotes the standard deviation. The first row is the results of *Llama-3.1-8B-Instruct* and the second row is *Mistral-7B-Instruct-v0.3*.

query and value projections in the last two layers of the LLM.

#### 7.4.2 Training-Free Results

Table 7.2 compares FTTT with various baselines across four reasoning datasets. FTTT, both with and without self-reflected feedback, outperforms conventional test-time scaling methods on average. This success is partially explained by the findings of [Ye et al. \(2024a\)](#), which show that training with error-correction data enhances reasoning capabilities and models do not retry during inference. FTTT is also efficient. For instance, the inference time of *Llama-3.1-8B-Instruct* on GSM8K with a budget of 32 is 3 GPU hours for the best parallel sampling method (Best-of-N) and 20 GPU hours for the best sequential revision method (Self-Refine). In contrast, FTTT achieves inference times of approximately 3 GPU hours without self-reflected feedback and 4 GPU hours with self-reflected feedback.

Notably, self-reflected feedback does not always improve results. Its effectiveness appears to depend on the LLM’s self-reflection ability. To test this, we computed the

Method	#Param.	MATH	GSM8K	MBPP	Avg.
Llama-3.1-8B-Instruct					
Adapter (Houlsby et al., 2019)	134M	0.5933 <sub>0.0151</sub>	0.7979 <sub>0.0056</sub>	0.2632 <sub>0.0058</sub>	0.5515
(IA) <sup>3</sup> (Liu et al., 2022a)	524K	0.6187 <sub>0.0105</sub>	0.8929 <sub>0.0107</sub>	0.5685 <sub>0.0022</sub>	0.6934
LoRA (Hu et al., 2022a)	1.7M	0.6387 <sub>0.0136</sub>	0.9186 <sub>0.0037</sub>	0.5639 <sub>0.0242</sub>	0.7071
LN-Tuning (Zhao et al., 2024a)	266K	0.6280 <sub>0.0113</sub>	0.8899 <sub>0.0056</sub>	0.5748 <sub>0.0175</sub>	0.6976
Full Fine-Tuning	8B	0.6027 <sub>0.0136</sub>	0.7722 <sub>0.0056</sub>	0.4034 <sub>0.0096</sub>	0.5928
OP-TUNE	439K	<b>0.7013</b> <sub>0.0050</sub>	<b>0.9246</b> <sub>0.0056</sub>	<b>0.6184</b> <sub>0.0159</sub>	0.7481
Mistral-7B-Instruct-v0.3					
Adapter (Houlsby et al., 2019)	134M	<b>0.5418</b> <sub>0.0111</sub>	0.8264 <sub>0.0021</sub>	0.2763 <sub>0.0076</sub>	0.5482
(IA) <sup>3</sup> (Liu et al., 2022a)	524K	0.5041 <sub>0.0056</sub>	0.8686 <sub>0.0060</sub>	0.4914 <sub>0.0185</sub>	0.6214
LoRA (Hu et al., 2022a)	1.7M	0.5117 <sub>0.0091</sub>	0.8686 <sub>0.0016</sub>	0.4968 <sub>0.0046</sub>	0.6257
LN-Tuning (Zhao et al., 2024a)	266K	0.4357 <sub>0.0115</sub>	0.8259 <sub>0.0051</sub>	0.4065 <sub>0.0095</sub>	0.5560
Full Fine-Tuning	7B	0.5388 <sub>0.0157</sub>	0.7355 <sub>0.0016</sub>	0.2548 <sub>0.0095</sub>	0.5097
OP-TUNE	439K	0.4891 <sub>0.0111</sub>	<b>0.9003</b> <sub>0.0039</sub>	<b>0.5194</b> <sub>0.0070</sub>	0.6363

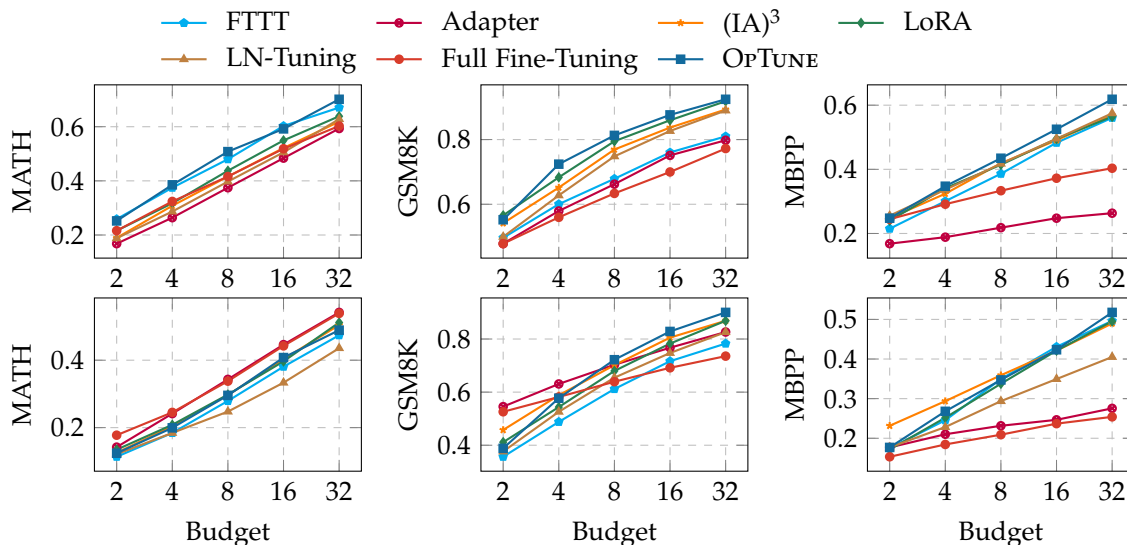
**Table 7.3:** Fine-tuning results on four datasets with a budget of 32. #Param. denotes the number of trainable parameters. We report the mean of three runs with different random seeds and standard deviation in the subscript. **Bold** entries are the best results.

Spearman rank correlation between FTTT and Self-Refine, a self-reflection-based algorithm. The Spearman coefficient ( $r = 0.8656$ ,  $p \leq 0.05$ ) indicates a strong positive correlation, supporting our hypothesis. We also observe that Self-Consistency performs poorly on code tasks because sampled code snippets rarely match exactly, making majority voting akin to random selection.

Figure 7.3 illustrates performance for FTTT and baselines under varying budgets. FTTT consistently outperforms baselines, with greater gains under constrained budgets. In contrast, Revision and Self-Consistency do not scale well. Revision struggles with long-context reasoning due to length generalization issues (Li et al., 2024a), while Self-Consistency fails to leverage feedback, often discarding correct answers during majority voting due to long-tailed distributions of correct answers (Brown et al., 2024).

### 7.4.3 Fine-Tuning Results

We present the results of PEFT baselines and OP-TUNE with a budget of 32 in Table 7.3. Best-of-N is applied to PEFT baselines to exploit test-time feedback. HumanEval is excluded



**Figure 7.4:** The scaling trends of different fine-tuning methods under varying budgets. We report the mean results of three random trials. The first row is the results of *Llama-3.1-8B-Instruct* and the second row is *Mistral-7B-Instruct-v0.3*.

as it lacks a training set. Table 7.3 highlights the effectiveness of OPTUNE, outperforming all PEFT baselines by at least 2.58% on average. OPTUNE is also parameter-efficient, with 439K trainable parameters that are comparable to the most lightweight PEFT method (LN-Tuning, 266K parameters), while surpassing the best PEFT method (LoRA, 1.7M parameters) with an order of magnitude fewer parameters. However, OPTUNE shows suboptimal performance on MATH for *Mistral-7B-Instruct-v0.3*, which is consistent with other PEFT methods with few trainable parameters (e.g., (IA)<sup>3</sup>, LoRA, LN-Tuning). This is likely due to *Mistral-7B-Instruct-v0.3*'s limited mathematical reasoning capabilities, requiring significant parameter updates to improve performance in this domain.

OPTUNE incurs negligible inference overhead. For example, on GSM8K with *Llama-3.1-8B-Instruct* and a budget of 32, the best test-time scaling baseline (FTTT) requires 4 GPU hours, whereas OPTUNE uses only 1.5 GPU hours, benefiting from shorter yet accurate predictions.

Finally, Figure 7.4 examines the scaling behavior of PEFT baselines and OPTUNE. Initially, OPTUNE underperforms compared to other PEFT methods and FTTT, but it mostly achieves

Method	#Params.	Pass@1
<b>OpTune</b>	439K	0.2600
+ w/o Dropout	439K	0.2243
+ w/o Norm	439K	0.2150
+ Not Share $\theta_1$ & $\theta_2$	878K	0.2243

**Table 7.4:** The ablation study of OPTUNE. We report results of *Llama-3.1-8B-Instruct* on MBPP with a budget of 2.

superior results when the budget exceeds 2. OPTUNE’s weaker performance with smaller budgets arises from its reliance on an initial attempt sampled from the raw LLM to initiate the process. This initial attempt often fails but is still counted as a valid attempt, making OPTUNE less competitive in low-budget settings.

#### 7.4.4 Analysis

**Ablation Study.** Table 7.4 presents an ablation study on the architecture design of OPTUNE. The results demonstrate that all components are essential, as removing any of them significantly degrades performance. Notably, normalization is the most critical component, as it addresses the varying gradient scales of different weights.

**Case Study.** Table 7.5 provides two examples on GSM8K where the leading PEFT method, LoRA, fails, but OPTUNE succeeds using *Llama-3.1-8B-Instruct*. These examples highlight OPTUNE’s superior ability to correctly interpret and reason through questions, unlike LoRA.

## 7.5 Summary

In Chapter 7, we propose a novel paradigm that leverages optimization to address the challenge of exploiting test-time feedback, resulting in improved scaling performance. We further present a learnable test-time optimizer, OPTUNE, which surpasses various PEFT baselines. Both FTTT and OPTUNE are efficient in terms of speed and trainable parameter count.

---

**Question:** Blake and Kelly are having a contest to see who can run the most in 15 minutes. They decide to do it on a football field that is 100 yards long. Blake runs back and forth 15 times. Kelly runs back and forth once, and then decides that she doesn't want to run next to Blake, so she starts to run to the 40-yard line and back. She does this 34 times. How much farther does the winner run than the loser?

---

**LoRA:** Blake runs  $15 * 100 * 2 = 3000$  yards.  
Kelly runs  $100 * 2 = 200$  yards for each run to the 40-yard line and back.  
*Kelly runs  $200 * 34 = 6800$  yards.*  
Kelly runs  $6800 - 3000 = 3800$  yards more than Blake.  
The final answer is **3800 ✗**.

---

**OpTune:** Blake runs back and forth 15 times, and each time, he runs 100 yards twice because he runs to one end of the field and back. So, Blake's total running distance is  $15 \times 200 = 3000$  yards.  
Kelly runs back and forth once, which is 200 yards, and then runs to the 40-yard line and back, which is  $40 \times 2 = 80$  yards, and does this 34 times. *Kelly's total running distance is  $200 + (34 \times 80) = 200 + 2720 = 2920$  yards.*  
Therefore, Blake runs  $3000 - 2920 = 80$  yards farther than Kelly.  
The final answer is **80 ✓**.

---

**Question:** Josh decides to take up juggling to perform at the school talent show a month in the future. He starts off practicing juggling 3 balls, and slowly gets better adding 1 ball to his juggling act each week. After the end of the fourth week the talent show begins, but when Josh walks on stage he slips and drops three of his balls. 2 of them are caught by people in the crowd as they roll off the stage, but one gets lost completely since the auditorium is dark. With a sigh, Josh starts to juggle on stage with how many balls?

---

**LoRA:** Josh started with 3 balls and added 1 ball each week for 4 weeks, so he had  $3 + 4 = 7$  balls.  
*He dropped 3 balls, but 2 were caught, so he lost  $3 - 2 = 1$  ball.*  
He was left with  $7 - 1 = 6$  balls.  
The final answer is **6 ✗**.

---

**OpTune:** Josh started with juggling 3 balls. After the end of the fourth week, Josh would have added 4 balls, making it a total of  $3+4=7$  balls.  
*Josh had 3 balls left after 2 of the balls were caught by people in the crowd, so he was left with  $7-3=4$  balls.*  
The final answer is **4 ✓**.

---

**Table 7.5:** Two GSM8K examples with model predictions. ✓ means correct and ✗ means incorrect. Sentences marked in orange are incorrect reasoning steps and those marked in blue are the correct counterparts.

## Chapter 8

# Conclusion

This dissertation addresses two critical challenges in the evaluation of LLMs in the era of their rapid advancement: the lack of systematic evaluation benchmarks (particularly for Chinese) and the unreliability of existing evaluation results.

To tackle these challenges, this dissertation first introduces CLEVA, a comprehensive Chinese LLM evaluation benchmark. CLEVA features a holistic taxonomy covering 31 tasks and 9 dimensions per task, offering detailed insights into LLM capabilities. Through CLEVA, we uncover emergent abilities in Chinese LLMs and reveal intrinsic correlations across various abilities.

Building upon CLEVA, this dissertation proposes C<sup>2</sup>LEVA, which ensures robust and trustworthy evaluation. C<sup>2</sup>LEVA addresses three key issues that compromise evaluation reliability: (1) *Format*: By providing multiple prompt templates per task, C<sup>2</sup>LEVA ensures evaluation results are consistent across different question formats. (2) *Language*: As a bilingual benchmark (English and Simplified Chinese), C<sup>2</sup>LEVA assesses cross-lingual transfer in LLMs. (3) *Data Contamination*: Systematic strategy prevent data contamination, safeguarding the integrity of evaluation results.

Furthermore, this dissertation demonstrates how evaluation benchmarks can drive innovation through three practical research applications:

- **Knowledge-Grounded Conversation**: Human evaluation reveals that LLMs possess

knowledge that goes beyond what is easily retrievable via search engines, such as commonsense and composite facts. Leveraging this, we propose PLATO-KAG+, a knowledge-grounded conversation system that uses LLMs as a knowledge base. This approach not only reduces training costs but also enhances interpretability.

- **Multi-Hop Reasoning:** Evaluation of existing LLMs on multi-hop reasoning tasks highlights performance degradation when irrelevant information is present. To address this, we propose reasoning with attribution, where LLMs cite specific passages to support their chain-of-thought reasoning. We also collect attribution annotations on existing multi-hop reasoning datasets to support and advance future research.
- **Test-Time Scaling:** Experiments with test-time scaling approaches expose limitations, such as sampling-based methods lacking memory of past failures and revision-based methods being latency-intensive. To resolve these issues, we introduce FTTT, which uses model parameters as memory to improve efficiency and effectiveness. Additionally, we propose OPTUNE, a learnable test-time optimizer designed to enhance FTTT.

In summary, this dissertation establishes two guiding principles for designing practical LLM evaluation benchmarks: comprehensiveness and robustness. These principles are embodied in the development of CLEVA and C<sup>2</sup>LEVA. Moreover, the demonstrated applications in knowledge-grounded conversation, multi-hop reasoning, and test-time scaling illustrate how evaluation can serve as a catalyst for advancing LLM capabilities and practical use cases.

## 8.1 Produced Publications

The research work of this dissertation has produced some direct and indirect publications as listed below:

- [1] Jianqiao Zhao<sup>1</sup>, Yanyang Li<sup>1</sup>, Wanyu Du<sup>1</sup>, Yangfeng Ji, Dong Yu, Michael Lyu, Liwei Wang (2022). FlowEval: A Consensus-Based Dialogue Evaluation Framework Using

---

<sup>1</sup>Equal Contribution.

Segment Act Flows. *EMNLP*.

- [2] Yanyang Li, Jianqiao Zhao, Michael Lyu, Liwei Wang (2022). Eliciting Knowledge from Large Pre-Trained Models for Unsupervised Knowledge-Grounded Conversation. *EMNLP*.
- [3] Yongfeng Huang, Yanyang Li, Yichong Xu, Lin Zhang, Ruyi Gan, Jiaying Zhang, Liwei Wang (2023). MVP-Tuning: Multi-View Knowledge Retrieval with Prompt Tuning for Commonsense Reasoning. *ACL*.
- [4] Zi-Yuan Hu, Yanyang Li, Michael Lyu, Liwei Wang (2023). VL-PET: Vision-and-Language Parameter-Efficient Tuning via Granularity Control. *ICCV*.
- [5] Yanyang Li<sup>1</sup>, Jianqiao Zhao<sup>1</sup>, Duo Zheng, Zi-Yuan Hu, Zhi Chen, Xiaohui Su, Yongfeng Huang, Shijia Huang, Dahua Lin, Michael Lyu, Liwei Wang (2023). CLEVA: Chinese Language Models EVALuation Platform. *EMNLP Demo*.
- [6] Shijia Huang, Jianqiao Zhao, Yanyang Li, Liwei Wang (2023). Learning Preference Model for LLMs via Automatic Preference Data Generation. *EMNLP*.
- [7] Yanyang Li, Shuo Liang, Michael Lyu, Liwei Wang (2024). Making Long-Context Language Models Better Multi-Hop Reasoners. *ACL*.
- [8] Yanyang Li, Tin Long Wong, Cheung To Hung, Jianqiao Zhao, Duo Zheng, Ka Wai Liu, Michael Lyu, Liwei Wang (2025). C<sup>2</sup>LEVA: Toward Comprehensive and Contamination-Free Language Model Evaluation. *Findings of ACL*.
- [9] Yanyang Li, Michael Lyu, Liwei Wang (2025). Learning to Reason from Feedback at Test-Time. *ACL*.

# References

- Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. [Word order does matter and shuffled language models know it](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6907–6919. Association for Computational Linguistics.
- AI@Meta. 2024. [Llama 3 model card](#).
- Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. 2024. [The surprising effectiveness of test-time training for abstract reasoning](#).
- Chenxin An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. [L-eval: Instituting standardized evaluation for long context language models](#). *CoRR*, abs/2307.11088.
- Anthropic. 2024. [Claude 3.5 sonnet model card addendum](#).
- Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. [Program synthesis with large language models](#). *CoRR*, abs/2108.07732.
- BIG bench authors. 2023. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *Transactions on Machine Learning Research*.
- Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. 2016. [Using fast weights to attend to the recent past](#). In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4331–4339.
- Stephen H. Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-David, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Alan Fries, Maged S. Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Xiangru Tang, Mike Tian-Jian Jiang, and Alexander M. Rush. 2022. [Promptsources: An integrated development environment and repository for natural language prompts](#).
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav

- Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *CoRR*, abs/2204.05862.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. [Longbench: A bilingual, multitask benchmark for long context understanding](#). *CoRR*, abs/2308.14508.
- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. [A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.
- Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V. Le. 2017. [Neural optimizer search with reinforcement learning](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 459–468. PMLR.
- Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. 2024. [Scaling laws for data poisoning in llms](#). *CoRR*, abs/2408.02946.
- Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Ben Buchanan, Andrew Lohn, Micah Musser, and Katerina Sedova. 2021. [Truth, lies, and automation: How language models could change disinformation](#).
- Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu,

- Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, and et al. 2024. [Internlm2 technical report](#). *CoRR*, abs/2403.17297.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. [Extracting training data from large language models](#). In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidi 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, Joshua 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](#). *CoRR*, abs/2107.03374.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023a. [Extending context window of large language models via positional interpolation](#). *CoRR*, abs/2306.15595.
- Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. 2022. [Learning to optimize: A primer and A benchmark](#). *J. Mach. Learn. Res.*, 23:189:1–189:59.
- Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. 2023b. [Symbolic discovery of optimization algorithms](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023c. [Longlora: Efficient fine-tuning of long-context large language models](#).
- Yutian Chen, Matthew W. Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Timothy P. Lillicrap, Matthew M. Botvinick, and Nando de Freitas. 2017. [Learning to learn without gradient descent by gradient descent](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 748–756. PMLR.

- Zhenyi Lu Chenghao Fan and Jie Tian. 2023. [Chinese-vicuna: A chinese instruction-following llama-based model](#).
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. [Chatbot arena: An open platform for evaluating llms by human preference](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- François Chollet. 2019. [On the measure of intelligence](#).
- 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](#). *CoRR*, abs/2110.14168.
- FlagEval Contributors. 2023a. Flageval. <https://github.com/FlagOpen/FlagEval>.
- OpenCompass Contributors. 2023b. Opencompass: A universal evaluation platform for foundation models. <https://github.com/InternLM/OpenCompass>.
- Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. [Selection-inference: Exploiting large language models for interpretable logical reasoning](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Yiming Cui, Ziqing Yang, and Xin Yao. 2023. [Efficient and effective text encoding for chinese llama and alpaca](#). *CoRR*, abs/2304.08177.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, Tao Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, and Xiaowen Sun. 2024. [Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model](#). *CoRR*, abs/2405.04434.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaustubh D. Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, Tongshuang Wu, Jascha Sohl-Dickstein, Jinho D. Choi, Eduard Hovy, Ondrej Dusek, Sebastian Ruder, Sajant Anand, Nagender Aneja, Rabin Banjade, Lisa Barthe, Hanna Behnke, Ian Berlot-Attwell, Connor Boyle, Caroline Brun, Marco Antonio Sobrevilla Cabezudo, Samuel Cahyawijaya, Emile Chapuis, Wanxiang Che, Mukund Choudhary, Christian Clauss, Pierre Colombo, Filip Cornell, Gautier Dagan, Mayukh Das, Tanay Dixit, Thomas Dopierre, Paul-Alexis Dray, Suchitra Dubey, Tatiana Ekeinhor, Marco Di Giovanni, Rishabh Gupta, Rishabh Gupta, Louanes Hamla, Sang Han, Fabrice Harel-Canada, Antoine Honore, Ishan Jindal, Przemyslaw K. Joniak, Denis Kleyko, Venelin Kovatchev, Kalpesh Krishna, Ashutosh Kumar, Stefan Langer, Seungjae Ryan Lee, Corey James Levinson, Hualou Liang, Kaizhao Liang, Zhexiong Liu, Andrey Lukyanenko, Vukosi Marivate, Gerard de Melo, Simon Meoni, Maxime Meyer, Afnan Mir, Nafise Sadat Moosavi, Niklas Muennighoff, Timothy Sum Hon Mun, Kenton Murray, Marcin Namysl, Maria Obedkova, Priti Oli, Nivranshu Pasricha, Jan Pfister, Richard Plant, Vinay Prabhu, Vasile Pais, Libo Qin, Shahab Raji, Pawan Kumar Rajpoot, Vikas Raunak, Roy Rinberg, Nicolas Roberts, Juan Diego Rodriguez, Claude Roux, Vasconcellos P. H. S., Ananya B. Sai, Robin M. Schmidt, Thomas Scialom, Tshephisho Sefara, Saqib N. Shamsi, Xudong Shen, Haoyue Shi, Yiwen Shi, Anna Shvets, Nick Siegel, Damien Sileo, Jamie Simon, Chandan Singh, Roman Sitelew, Priyank Soni, Taylor Sorensen, William Soto, Aman Srivastava, KV Aditya Srivatsa, Tony Sun, Mukund Varma T, A Tabassum, Fiona Anting Tan, Ryan Teehan, Mo Tiwari, Marie Tolkiehn, Athena Wang, Zijian Wang, Gloria Wang, Zijie J. Wang, Fuxuan Wei, Bryan Wilie, Genta Indra Winata, Xinyi Wu, Witold Wydmański, Tianbao Xie, Usama Yaseen, M. Yee, Jing Zhang, and Yue Zhang. 2021. [NL-augmenter: A framework for task-sensitive natural language augmentation](#).

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. [Wizard of wikipedia: Knowledge-powered conversational agents](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Jesse Dodge, Maarten Sap, Ana Marasovic, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1286–1305. Association for Computational Linguistics.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. [DE-COP: detecting copyrighted content in language models training data](#). *CoRR*, abs/2402.09910.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev,

- Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. [The llama 3 herd of models](#). *CoRR*, abs/2407.21783.
- Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. 2021. [Neural path hunter: Reducing hallucination in dialogue systems via path grounding](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2197–2214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, and Yongfeng Zhang. 2023. [Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes](#). *CoRR*, abs/2312.14890.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. [RARR: Researching and revising what language models say, using language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. [Enabling large language models to generate text with citations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.
- Shahriar Golchin and Mihai Surdeanu. 2023. [Time travel in llms: Tracing data contamination in large language models](#). *CoRR*, abs/2308.08493.

- Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. [Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations](#). In *Proc. Interspeech 2019*, pages 1891–1895.
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. [rstar-math: Small llms can master math reasoning with self-evolved deep thinking](#).
- Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. [DialFact: A benchmark for fact-checking in dialogue](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.
- Moritz Hardt and Yu Sun. 2024. [Test-time training on nearest neighbors for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#). *CoRR*, abs/1503.02531.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Hongsheng Hu, Zoran Salcic, Gillian Dobbie, Jinjun Chen, Lichao Sun, and Xuyun Zhang. 2022b. [Membership inference via backdooring](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3832–3838. ijcai.org.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. [Large language models cannot self-correct reasoning yet](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Xinxian Huang, Huang He, Siqi Bao, Fan Wang, Hua Wu, and Haifeng Wang. 2021. [PLATO-KAG: Unsupervised knowledge-grounded conversation via joint modeling](#). In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 143–154, Online. Association for Computational Linguistics.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. [C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models](#). *CoRR*, abs/2305.08322.
- Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. [Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5075–5084. Association for Computational Linguistics.
- Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023. [Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases](#). *CoRR*, abs/2303.14742.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Yichen Jiang and Mohit Bansal. 2019. [Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. [Decomposed prompting: A modular approach for solving complex tasks](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. [Sequential latent knowledge selection for knowledge-grounded dialogue](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Diederik P. Kingma and Max Welling. 2014. [Auto-encoding variational bayes](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. [Internet-augmented dialogue generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.
- Wouter Kool, Herke van Hoof, and Max Welling. 2019. [Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3499–3508. PMLR.
- Ananya Kumar, Percy Liang, and Tengyu Ma. 2019. [Verified uncertainty calibration](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3787–3798.
- Fangyu Lei, Qian Liu, Yiming Huang, Shizhu He, Jun Zhao, and Kang Liu. 2023. [S3eval: A synthetic, scalable, systematic evaluation suite for large language models](#). CoRR, abs/2310.15147.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. [Question and answer test-train overlap in open-domain question answering datasets](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023a. [How long can open-source llms truly promise on context length?](#)
- Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023b. [A survey of large language models attribution](#).
- Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023c. [CMMLU: measuring massive multitask language understanding in chinese](#). *CoRR*, abs/2306.09212.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Ke Li and Jitendra Malik. 2017. [Learning to optimize](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023d. AlpacaEval: An automatic evaluator of instruction-following models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval).
- Yanyang Li, Shuo Liang, Michael Lyu, and Liwei Wang. 2024a. [Making long-context language models better multi-hop reasoners](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2475, Bangkok, Thailand. Association for Computational Linguistics.
- Yanyang Li, Michael Lyu, and Liwei Wang. 2025. [Learning to reason from feedback at test-time](#). *CoRR*, abs/2502.15771.
- Yanyang Li, Tin Long Wong, Cheung To Hung, Jianqiao Zhao, Duo Zheng, Ka Wai Liu, Michael R. Lyu, and Liwei Wang. 2024b. [C<sup>2</sup>Leva: Toward comprehensive and contamination-free language model evaluation](#). *CoRR*, abs/2412.04947.

- Yanyang Li, Jianqiao Zhao, Michael Lyu, and Liwei Wang. 2022. [Eliciting knowledge from large pre-trained models for unsupervised knowledge-grounded conversation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10551–10564, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yanyang Li, Jianqiao Zhao, Duo Zheng, Zi-Yuan Hu, Zhi Chen, Xiaohui Su, Yongfeng Huang, Shijia Huang, Dahua Lin, Michael R. Lyu, and Liwei Wang. 2023e. [CLEVA: chinese language models evaluation platform](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 186–217. Association for Computational Linguistics.
- Yucheng Li, Frank Guerin, and Chenghua Lin. 2024c. [Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 18600–18607. AAAI Press.
- Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. [Learning to select knowledge for response generation in dialog systems](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5081–5087. ijcai.org.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yuksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. [Holistic evaluation of language models](#). *Trans. Mach. Learn. Res.*, 2023.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Ye Lin, Yanyang Li, Ziyang Wang, Bei Li, Quan Du, Tong Xiao, and Jingbo Zhu. 2021. [Weight distillation: Transferring the knowledge in neural network parameters](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2076–2088, Online. Association for Computational Linguistics.
- Chuang Liu, Renren Jin, Yuqi Ren, Linhao Yu, Tianyu Dong, Xiaohan Peng, Shuting Zhang, Jianxiang Peng, Peiyi Zhang, Qingqing Lyu, Xiaowen Su, Qun Liu, and Deyi Xiong. 2023a.

- M3KE: A massive multi-level multi-subject knowledge evaluation benchmark for chinese large language models. *CoRR*, abs/2305.10263.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. **Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning**. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022b. **Generated knowledge prompting for commonsense reasoning**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. **Lost in the middle: How language models use long contexts**. *Trans. Assoc. Comput. Linguistics*, 12:157–173.
- Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. **Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling**.
- Tianyang Liu, Canwen Xu, and Julian J. McAuley. 2023b. **Repobench: Benchmarking repository-level code auto-completion systems**. *CoRR*, abs/2306.03091.
- Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. **TTT++: when does self-supervised test-time training fail or thrive?** In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21808–21820.
- Zihan Liu, Mostofa Patwary, Ryan Prenger, Shrimai Prabhumoye, Wei Ping, Mohammad Shoeybi, and Bryan Catanzaro. 2022c. **Multi-stage prompting for knowledgeable dialogue generation**. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1317–1337, Dublin, Ireland. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. **Self-refine: Iterative refinement with self-feedback**. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. **On faithfulness and factuality in abstractive summarization**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

- Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, H. Francis Song, Martin J. Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. 2022. [Teaching language models to support answers with verified quotes](#). *CoRR*, abs/2203.11147.
- Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. [ParLAI: A dialog research software platform](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. [Fast model editing at scale](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. [Towards exploiting background knowledge for building conversation systems](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. [Crosslingual generalization through multitask finetuning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15991–16111. Association for Computational Linguistics.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#).
- Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. [Obtaining well calibrated probabilities using bayesian binning](#). In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. [Webgpt: Browser-assisted question-answering with human feedback](#). *CoRR*, abs/2112.09332.
- Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. [Mixeval: Deriving wisdom of the crowd from LLM benchmark mixtures](#). *CoRR*, abs/2406.06565.
- OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry

- Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. 2025. [Competitive programming with large reasoning models](#).
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- OpenAI. 2024. [Openai o1 system card](#).
- OpenAI. 2025. [Openai o3-mini system card](#).
- Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. 2023. [Proving test set contamination in black box language models](#). *CoRR*, abs/2310.17623.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *NeurIPS*.
- Peng Si Ow and Thomas E Morton. 1988. Filtered beam search in scheduling. *The International Journal Of Production Research*, 26(1):35–62.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. [Yarn: Efficient context window extension of large language models](#). *CoRR*, abs/2309.00071.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Kun Qian, Shunji Wan, Claudia Tang, Youzhi Wang, Xuanming Zhang, Maximillian Chen, and Zhou Yu. 2024. [VarBench: Robust language model benchmarking through dynamic variable perturbation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16131–16161, Miami, Florida, USA. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican,

- Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *CoRR*, abs/2403.05530.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. [How much knowledge can you pack into the parameters of a language model?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. [Fitnets: Hints for thin deep nets](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. [Multitask prompted training enables zero-shot task generalization](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. [BLOOM: A 176b-parameter open-access multilingual language model](#). *CoRR*, abs/2211.05100.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023a. [Large language models can be easily distracted by irrelevant context](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.
- Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023b. [Detecting pretraining data from large language models](#). *CoRR*, abs/2310.16789.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. [Membership inference attacks against machine learning models](#). In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 3–18. IEEE Computer Society.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. [Retrieval augmentation reduces hallucination in conversation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. [Scaling LLM test-time compute optimally can be more effective than scaling model parameters](#). *CoRR*, abs/2408.03314.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: a simple way to prevent neural networks from overfitting](#). *J. Mach. Learn. Res.*, 15(1):1929–1958.
- Hao Sun, Zhixin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023a. [Safety assessment of chinese large language models](#). *CoRR*, abs/2304.10436.
- Yu Sun, Xinhao Li, Karan Dalal, Chloe Hsu, Sanmi Koyejo, Carlos Guestrin, Xiaolong Wang, Tatsunori Hashimoto, and Xinlei Chen. 2023b. [Learning to \(learn at test time\)](#). *CoRR*, abs/2310.13807.
- Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. 2020. [Test-time training with self-supervision for generalization under distribution shifts](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9229–9248. PMLR.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. <https://github.com/InternLM/InternLM>.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agueria-Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. 2022. [Lamda: Language models for dialog applications](#). *CoRR*, abs/2201.08239.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Thimothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [MuSiQue: Multihop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural*

*Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021. [Can generative pre-trained language models serve as knowledge bases for closed-book QA?](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3241–3251, Online. Association for Computational Linguistics.

Lihan Wang, Bowen Qin, Binyuan Hui, Bowen Li, Min Yang, Bailin Wang, Binhua Li, Jian Sun, Fei Huang, Luo Si, and Yongbin Li. 2022. [Proton: Probing schema linking information from pre-trained language models for text-to-sql parsing.](#) In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1889–1898. ACM.

Siyuan Wang, Zhuohan Long, Zhihao Fan, Zhongyu Wei, and Xuanjing Huang. 2024a. [Benchmark self-evolving: A multi-agent framework for dynamic LLM evaluation.](#) *CoRR*, abs/2402.11443.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. [Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.](#) In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models.](#) In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yan Wang, D. Ma, and Deng Cai. 2024b. [With greater text comes greater necessity: Inference-time training helps long text generation.](#) *CoRR*, abs/2401.11504.

Albert Webson and Ellie Pavlick. 2022. [Do prompt-based models really understand the meaning of their prompts?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2300–2344. Association for Computational Linguistics.

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. 2022a. [Emergent abilities of large language models.](#) *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. [Chain-of-thought prompting elicits reasoning in large language models.](#) In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

- Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. 2024. [Proving membership in LLM pretraining data via data watermarks](#). *CoRR*, abs/2402.10892.
- Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. 2023. [Skywork: A more open bilingual foundation model](#). *CoRR*, abs/2310.19341.
- Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2024. [Livebench: A challenging, contamination-free llm benchmark](#).
- Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024a. [How easily do irrelevant inputs skew the responses of large language models?](#)
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024b. [Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models](#).
- Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. [Self-evaluation guided beam search for reasoning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. [Baichuan 2: Open large-scale language models](#). *CoRR*, abs/2309.10305.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. [Qwen2 technical report](#).

- Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. [Re3: Generating longer stories with recursive reprompting and revision](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023b. [Rethinking benchmark and contamination for language models with rephrased samples](#). *CoRR*, abs/2311.04850.
- Yilin Yang, Liang Huang, and Mingbo Ma. 2018a. [Breaking the beam search curse: A study of \(re-\)scoring methods and stopping criteria for neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2024a. [Physics of language models: Part 2.2, how to learn from mistakes on grade-school math problems](#).
- Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. 2024b. [Differential transformer](#).
- Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. [Privacy risk in machine learning: Analyzing the connection to overfitting](#). In *31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018*, pages 268–282. IEEE Computer Society.
- Jiahao Ying, Yixin Cao, Bo Wang, Wei Tang, Yizhe Yang, and Shuicheng Yan. 2024. [Have seen me before? automating dataset updates towards reliable and timely evaluation](#). *CoRR*, abs/2402.11894.
- Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. [Making retrieval-augmented language models robust to irrelevant context](#). *CoRR*, abs/2310.01558.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. [Yi: Open foundation models by 01.ai](#). *CoRR*, abs/2403.04652.

- Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. [Chain-of-note: Enhancing robustness in retrieval-augmented language models](#). *CoRR*, abs/2311.09210.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. [GLM-130B: an open bilingual pre-trained model](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. [Chatglm: A family of large language models from GLM-130B to GLM-4 all tools](#). *CoRR*, abs/2406.12793.
- Hui Zeng. 2023. [Measuring massive multitask chinese understanding](#). *CoRR*, abs/2304.12986.
- Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2023a. [Beam retrieval: General end-to-end retrieval for multi-hop question answering](#).
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. [Personalizing dialogue agents: I have a dog, do you have pets too?](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. 2021. [Revisiting few-sample BERT fine-tuning](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. [Pretraining data detection for large language models: A divergence-based calibration method](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5263–5274, Miami, Florida, USA. Association for Computational Linguistics.
- Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023b. [Evaluating the performance of large language models on GAOKAO benchmark](#). *CoRR*, abs/2305.12474.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the*

- Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. 2024a. [Tuning layernorm in attention: Towards efficient multi-modal LLM finetuning](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Shuai Zhao, Linchao Zhu, Ruijie Quan, and Yi Yang. 2024b. [Ghost sentence: A tool for everyday users to copyright data from large language models](#). *CoRR*, abs/2403.15740.
- Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. [Knowledge-grounded dialogue generation with pre-trained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.
- Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023a. [Can we edit factual knowledge by in-context learning?](#) In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4862–4876. Association for Computational Linguistics.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. [Agieval: A human-centric benchmark for evaluating foundation models](#). *CoRR*, abs/2304.06364.
- Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. [A dataset for document grounded conversations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.
- Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. [Don't make your LLM an evaluation benchmark cheater](#). *CoRR*, abs/2311.01964.
- Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. [Retrieving and reading: A comprehensive survey on open-domain question answering](#).
- Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. 2023a. [Dyval: Graph-informed dynamic evaluation of large language models](#). *CoRR*, abs/2309.17167.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, and Xing Xie. 2023b. [Promptbench: Towards evaluating the robustness of large language models on adversarial prompts](#). *CoRR*, abs/2306.04528.