Enhancing Input-Label Mapping in In-Context Learning with Contrastive Decoding

Keqin Peng[1], **Liang Ding**[2*], **Yuanxin Ouyang**[1], **Meng Fang**[3], **Yancheng Yuan**[4], **Dacheng Tao**[5]
[1]Beihang University [2]The University of Sydney [3]University of Liverpool
[4]The Hong Kong Polytechnic University [5]Nanyang Technological University
keqin.peng@buaa.edu.cn, liangding.liam@gmail.com

Abstract

Large language models (LLMs) excel at a range of tasks through in-context learning (ICL), where only a few task examples guide their predictions. However, prior research highlights that LLMs often overlook input-label mapping information in ICL, relying more on their pre-trained knowledge. To address this issue, we introduce In-Context Contrastive Decoding (ICCD), a novel method that emphasizes input-label mapping by contrasting the output distributions between positive and negative in-context examples. Experiments on 7 natural language understanding (NLU) tasks show that our ICCD method brings consistent and significant improvement (up to +1.8 improvement on average) upon 6 different scales of LLMs without requiring additional training. Our approach is versatile, enhancing performance with various demonstration selection methods, demonstrating its broad applicability and effectiveness. The code and scripts are released at https://github.com/Romainpkq/CD_ICL.

1 Introduction

In-context learning (ICL, Brown et al., 2020) is one of the most remarkable emergent capabilities of large language models (LLMs, Achiam et al., 2023; Dubey et al., 2024). By leveraging just a few carefully selected input-output examples, ICL enables models to adapt to new tasks without parameter updating (Dong et al., 2022; Peng et al., 2024). This approach has proven highly effective in unlocking the advanced capabilities of LLMs and has become a standard technique for tackling a spectrum of tasks, like translation, coding, and reasoning (Peng et al., 2023; Wang et al., 2025; Wibisono and Wang, 2024).

Previous studies (Pan et al., 2023; Wei et al., 2023) have identified two critical factors for successful ICL: *task recognition (TR)*, which involves identifying the task from the demonstrations and utilizing prior knowledge to make predictions, and *task learning (TL)*, which focuses on directly learning the input-label mappings from the demonstrations. However, ICL faces challenges in overcoming the biases introduced by pretraining (Kossen et al., 2024), and LLMs tend to underutilize input-label mapping information (Min et al., 2022). For example, in tasks like SST-2 (Socher et al., 2013b), the model may default to using its internal knowledge rather than learning the specific input-label mappings provided in the context.

To address this issue, we propose a simple yet effective method called *in-context contrastive decoding* (ICCD). Our method is inspired by the contrastive decoding technique (Li et al., 2023; Sennrich et al., 2024; Kim et al., 2024; Zhong et al., 2024; Wang et al., 2024), which increases the probability of the desired output by suppressing undesired outputs, and our ICCD enhances the model's attention to input-label mapping during generation. Specifically, we construct negative in-context examples by altering the inputs of the demonstrations, creating incorrect input-label mappings while keeping the labels unchanged. By comparing the output distributions between positive and negative examples, ICCD effectively emphasizes the correct input-label mappings, integrating this information into the original ICL process. Notably, our method works with any pretrained LLMs without requiring additional training.

Experimental results across seven natural language understanding tasks demonstrate that our ICCD strategy consistently and significantly improves performance upon several advanced LLMs, e.g., Llama-3.1, Llama-3.2, and Qwen2, across various datasets and model scales. Moreover, we show that ICCD can be seamlessly integrated with different demonstration selection methods, showcasing its robustness and universal applicability.

* Corresponding Authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 997–1004
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

2 Methodology

2.1 Background

Given an input query x, the probability of generating the target y using a casual LLM M parameterized by θ can be formulated as follows:

$$y \sim p_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x)), \tag{1}$$

where $\mathcal{T}(\cdot)$ is the template used to wrap up inputs and $c = \mathcal{T}(x_1), \cdots, \mathcal{T}(x_k)$ is the context string concatenating k in-context examples, $p_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x)) = \text{softmax}[\text{ logit}_\theta(y \mid \boldsymbol{c}, \mathcal{T}(x))]$ is the probability for the predicted token. For obtaining the desired y, the regular decoding method is to choose the token with the highest probability (*i.e.*, greedy decoding) or sampling from its distribution (*e.g.*, top-k decoding).

Here, we can observe that there are two kinds of knowledge contributing to model prediction, models' prior knowledge and input-label mapping information in in-context learning. However, LLMs usually prioritize prior knowledge over input-label mapping information (Kossen et al., 2024), leading to ICL's struggle to fully overcome prediction preferences acquired from pre-training.

2.2 In-Context Contrastive Decoding

To mitigate the issue above, we construct negative in-context examples to factor out the input-label mapping from the models' original output distribution contrastively. Specifically, in addition to the origin in-context examples $\boldsymbol{c}$, we construct negative in-context examples $\boldsymbol{c}^-$ with incorrect input-label mapping. We then subtract the negative output $\mathbf{z}_t^-$ from the positive output $\mathbf{z}_t$ to isolate the knowledge of input-label mapping. Finally, we integrate this knowledge with the original in-context learning to reinforce the importance of input-label mapping:

$$y_t \sim \text{softmax}(\mathbf{z}_t + \alpha(\mathbf{z}_t - \mathbf{z}_t^-)), \tag{2}$$

where α is a hyperparameter that governs the importance of input-label mapping information. Equivalently,

$$y_t \sim \tilde{p}_\theta(y|\boldsymbol{c}, \boldsymbol{c}^-, \mathcal{T}(x)) \tag{3}$$

$$\propto p_\theta(y|\boldsymbol{c}, \mathcal{T}(x)) \left(\frac{p_\theta(y|\boldsymbol{c}, \mathcal{T}(x)}{p_\theta(y|\boldsymbol{c}^-, \mathcal{T}(x))} \right)^\alpha. \tag{4}$$

Construction of $\boldsymbol{c}^-$. The negative in-context examples $\boldsymbol{c}^-$ is the key to the success of the in-context contrastive decoding method (ICCD). Considering the label bias (Zhao et al., 2021) of in-context learning, directly altering the labels of demonstrations may introduce a completely different label bias, potentially distorting the input-label mapping information. Hence, we adjust the inputs instead of the labels to change input-label mapping information. Specifically, for each demonstration (x_i, y_i), we first randomly select a different label $y_j (y_j \neq y_i)$ from the label space. Then we randomly choose an input x_j whose label is y_j from the demonstrations pool to construct the negative demonstration (x_j, y_i). We compare the effect of different $\boldsymbol{c}^-$ in Section 5.

3 Experimental Setup

Models and Baselines. We perform experiments across different sizes of models, including Llama-series: Llama3.2-1B (1B), Llama3.2-3B (3B) and Llama3.1-8B (8B) (Dubey et al., 2024) and Qwen2 series: Qwen2-0.5B (0.5B), Qwen2-1.5B (1.5B) and Qwen2-7B (7B) (Yang et al., 2024), which are all widely-used decoder-only dense LMs. We also conduct experiments on extensive alignment models, e.g., Llama3.2-1B-Instruct, Llama3.2-3B-Instruct, and Llama3.1-8B-Instruct (Dubey et al., 2024) to verify the generalizability of our approach. For the baseline, we use the regular decoding methods following prior work (Shi et al., 2024; Zhao et al., 2024).

Demonstration Selection methods. To verify that our method is complementary to different demonstration selection methods, we mainly consider three different demonstration selection methods that do not require additional training.

- **Random** baseline randomly select in context examples for each testing sample.
- **BM25** (Robertson et al., 2009) baseline uses BM25 to calculate the word-overlap similarity between samples and test input and select the high-similarity samples as demonstrations.
- **TopK** (Liu et al., 2022) baseline uses the nearest neighbors of a given test sample as the corresponding in-context examples.

Datasets and Metrics. We conduct a systematic study across 7 NLU tasks, including binary, multi-class classification tasks (SST-2, SST-5 (Socher et al., 2013a), CR, Subj (Wang et al., 2018)) and natural language inference tasks: MNLI (Williams

et al., 2018) and QNLI (Wang et al., 2018). We will report the accuracy to show the performance.

Experimental Details. Our method introduces a hyperparameter α to control the input-label mapping information. For simplicity, we set $\alpha = 1$ for all models and settings. We ran all experiments 3 times with different random seeds and reported the average accuracies. We use 16-shot ICL for all models. Without a special statement, we report the results of the random selection method.

4 Main Results

We demonstrate the effectiveness of our method in 7 NLU tasks described in the Datasets and Metrics section. We summarize the results in Table 1, Table 2, Tabel 3, and Figure 1. Based on the results, we can find that:

Our method brings gain across different tasks and model scales. Results on Table 2 show that our method can achieve consistently better performance across the majority of tasks under different model scales than the regular decoding method. Specifically, our method brings over 1.0 improvements (in accuracy) in all Llama-series models and Qwen2-series models. It's worth highlighting that ICCD brings +2.3 gains on average in the Qwen2-1.5B model. Furthermore, it is noteworthy that our approach can achieve more significant improvements in challenging tasks with the increase of model scale, such as QNLI and MNLI tasks, respectively bringing 5.1% (1.4%) and 1.8% (1.2%) gains compared to regular decoding in Llama3.1-8B (Qwen2-7B), demonstrating the effectiveness and universality of our method.

Our method consistently improves the performance with different in-context examples selection methods. Table 1 lists the average performance and standard deviation of different models with different demonstration selection methods. Clearly, our method can achieve better and stable performance with different demonstration selection methods. When the model scale increases, our method can achieve more improvement gains compared to the regular decoding method, +0.5 and +1.1 with BM25 method under Llama3.2-3B and Llama3.1-8B, respectively. These results prove that ICCD can be complementary with different demonstration selection methods.

Model	Decoding	Random		BM25		TopK	
		avg.	std.	avg.	std.	avg.	std.
Llama3.2-1B	Regular	66.1	-	72.5	-	73.6	-
	Ours	68.3	0.19	72.9	0.11	73.4	0.17
Llama3.2-3B	Regular	72.9	-	76.6	-	76.7	-
	Ours	74.6	0.47	77.1	0.28	76.9	0.19
Llama3.1-8B	Regular	77.6	-	79.7	-	80.2	-
	Ours	79.4	0.19	80.8	0.15	80.9	0.05

Table 1: **Average performance and standard deviation of 7 Natural Language Understanding (NLU) tasks with different in-context example selection methods.** **Red** results indicate that our method brings improvement over the regular decoding, while **Green** results denote no improvement.

Our method works for aligned chat models. To verify the effectiveness of our method for the chat LLMs, we conducted experiments on different instruction-tuned and RLHF-tuned LLMs. Figure 1 show that our method can achieve consistent improvement in different chat models, demonstrating that our method also works for instruction-tuned and safety-enhanced models.

Our method works for a larger number of target classes. To verify the effectiveness of our method for a larger number of target classes, we conducted experiments on datasets TREC (6 target classes) and Dbpedia (14 target classes) with the random selection method. Results on Table 3 show that our method can achieve remarkable improvement, demonstrating the effectiveness of our method in larger target classes.

5 Analysis

To further explore the impact of different factors on the effectiveness of our method, we conduct further analysis with the Llama3.2-8B models.

Effects of Different Negative In-context Examples. As mentioned in Section 2.2, the choice of negative in-context examples is important to the performance of our methods. Here, we conduct contrastive experiments to analyze the impact of different negative examples. Specifically, we refer to the selected negative examples as **Input**, if the input-label mapping is altered by modifying the inputs of the demonstrations. Additionally, we construct another variant, **Label**, in which the labels of the demonstrations are changed. For comparison, we also include **NULL**, which does not use any negative demonstrations, similar to Shi et al. (2024). The results in Table 4 show that **Input** out-

Model	Decoding	SST2	CR	SST5	Subj	QNLI	MNLI	AG_NEWS	*Avg.*
Llama3.2-1B	Regular	89.8	83.0	43.7	72.8	53.5	36.6	83.3	66.1
	Ours	**91.1**	**83.7**	**43.3**	**83.0**	**53.8**	**39.2**	**84.1**	**68.3 (+2.1)**
Llama3.2-3B	Regular	93.7	87.2	46.2	86.0	54.2	56.9	86.4	72.9
	Ours	**94.0**	**88.1**	**46.5**	**92.1**	**57.2**	**57.0**	**86.9**	**74.6 (+1.7)**
Llama3.1-8B	Regular	96.7	92.3	48.0	94.0	60.3	65.3	86.7	77.6
	Ours	**96.5**	**93.2**	**49.3**	**96.1**	**65.4**	**67.5**	**87.6**	**79.4 (+1.8)**
Qwen2-0.5B	Regular	87.9	89.4	34.5	62.2	52.5	47.6	78.1	64.6
	Ours	**89.2**	**89.6**	**33.9**	**68.1**	**53.2**	**47.6**	**78.7**	**65.8 (+1.2)**
Qwen2-1.5B	Regular	95.2	91.0	49.0	72.3	60.2	61.8	76.7	72.3
	Ours	**95.1**	**91.3**	**48.3**	**81.5**	**61.8**	**65.2**	**79.1**	**74.6 (+2.3)**
Qwen2-7B	Regular	96.0	91.5	51.9	82.3	71.4	78.7	83.8	79.4
	Ours	**96.3**	**91.7**	**52.9**	**90.4**	**72.8**	**79.9**	**85.0**	**81.3 (+1.9)**

Table 2: **Performance of different models across 7 Natural Language Understanding (NLU) tasks.** **Red** results indicate our method brings improvement over the regular decoding, while **Green** denote no improvement.

Model	Decoding	TREC	Dbpedia
Llama3.2-1B	**Regular**	40.0	85.6
	Ours	**46.2 (+6.2)**	**90.5 (+4.9)**
Llama3.2-3B	**Regular**	44.4	83.1
	Ours	**49.6 (+5.2)**	**91.4 (+8.3)**
Llama3.1-8B	**Regular**	41.0	87.5
	Ours	**46.6 (+5.6)**	**93.8 (6.3)**

Table 3: **Average performance of two datasets with larger target classes.** **Red** results indicate that our method brings improvement over the regular decoding.

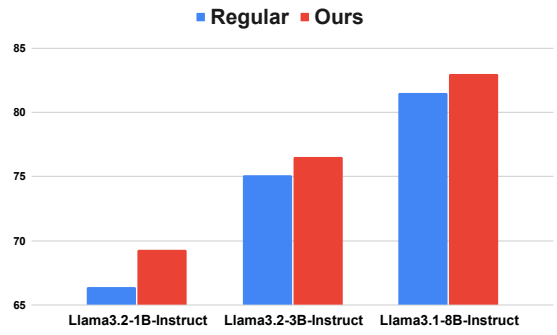

Figure 1: **Performance with different chat models.**

Method	Selection Method		
	Random	**BM25**	**TopK**
Regular Decoding	77.6	79.7	80.2
Equipped with our method			
+NULL	**73.0**	**75.8**	76.5
+Label	**77.3**	**79.5**	**80.0**
+Input	**79.4**	**80.8**	**80.9**

Table 4: **Average performance with different negative in-context examples.** **Red** results indicate that our method brings improvement over the regular decoding, while **Green** results denote no improvement.

	SST2	CR	SST5	Subj	QNLI	MNLI	AGNEWS
KL_divergence	0.64	0.48	0.43	0.49	0.04	0.27	0.79

Table 5: **The average KL divergence** between the normalized output distributions with positive and negative in-context examples with Llama3.2-8B.

performs the other counterparts, thus leaving it as our default setting in this work.

Differences between the positive and negative examples. To verify whether our proposed method can truly lead to models to contrast the positive and negative in-context examples, we calculate the average KL divergence between the output distributions and report the results in Table 5, we can notice that our method can get large KL divergence in most datasets, which means that the output distributions of positive and negative in-context examples are notably different. This demonstrates that our method can truly lead to models to contrast the positive and negative in-context examples.

Effects of Different number of shots. We gradually increase the number of in-context examples (denoted as N) from 1 to 16 to verify the influence of the number of shots in our method. Figure 2 reports the average performance of 7 NLU tasks and the different task QNLI. We see that our method can consistently outperform the regular decoding method with a different number of shots on aver-

Dataset	Method	α				
		0.0	0.5	1.0	1.5	2.0
SST5	**Random**	48.0	49.1	49.3	49.3	**49.5**
	BM25	53.0	**53.6**	53.3	53.2	53.1
	TopK	53.0	53.2	**53.2**	52.9	52.5
MNLI	**Random**	65.3	66.8	67.5	67.8	**68.1**
	BM25	65.8	66.6	67.1	67.4	**67.6**
	TopK	65.9	67.0	67.4	**67.7**	67.7

Table 6: **The SST5 and MNLI performance with different α.**

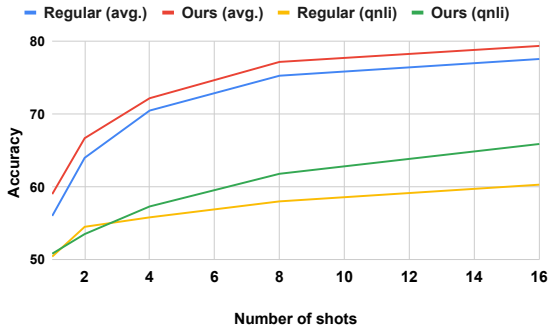

Figure 2: **The performance with different shots.**

age. For the task QNLI, as the number of shots increases, the performance gains of our method also improve. We attribute this to the model acquiring more input-label mapping information from the demonstrations, which aligns with previous findings (Pan et al., 2023).

Effects of α. The factor α in Eq. 2, which controls the importance of input-label mapping information, is an important hyper-parameter. In this part, we analyze its influence by evaluating the performance on SST5 and MNLI varying α from 0 to 2. The results on Table 6 show that: 1) the performance improves with the increase of α, and it becomes stable when $\alpha \geq 1.0$, we set $\alpha = 1$ as default; 2) For advanced demonstration selection methods(e.g.TopK), too large positive α values lead to performance degradation.

6 Conclusion

Large language models suffer from insufficient attention to the input-label mapping compared to their prior knowledge in in-context learning, leading to an unfaithful generation of the input query. In this work, we present a simple yet effective in-context contrastive decoding method that highlights input-label mapping by contrasting positive and negative in-context examples. Our experiments across various datasets and model architectures demonstrate the effectiveness and broad applicability of our approach, confirming its potential to enhance in-context learning.

Limitations

While the results presented in this paper demonstrate the effectiveness of our In-Context Contrastive Decoding (ICCD) method, there are a few limitations that warrant future exploration. First, our experiments were conducted on models up to 8B parameters, primarily due to computational limitations. Extending our method to even larger models (e.g., 70B parameters) could provide further insights into its scalability and effectiveness. Second, while our method shows promise across various Natural Language Understanding (NLU) tasks, its performance in specialized domains, such as legal or medical texts, has yet to be thoroughly examined. Third, our method requires additional forward passes to compute contrastive distributions. Although these passes are executed in parallel, the overall inference time may still increase. Future work will explore the generalizability of ICCD to these domains, as well as investigate its interaction with domain-specific datasets. Additionally, while we focused on classification tasks, other NLP tasks like text generation, machine translation, and summarization remain unexplored.

Acknowledges

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work is supported by the National Natural Science Foundation of China (No. 62377002). This project is supported by the National Research Foundation, Singapore, under its NRF Professorship Award No. NRF-P2024-001.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint*.

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 *NeurIPS*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint*.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2024. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. In *ICLR*.

Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. In-context learning learns label relationships but is not conventional learning. In *ICLR*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *ACL*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *DeeLIO*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *ACL Findings*.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. In *ACL*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. In *Findings of EMNLP*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*.

Rico Sennrich, Jannis Vamvas, and Alireza Mohammadshahi. 2024. Mitigating hallucinations and off-target machine translation with source-contrastive and language-contrastive decoding. In *EACL*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *NAACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013a. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP*.

Shuai Wang, Liang Ding, Li Shen, Yong Luo, Zheng He, Wei Yu, and Dacheng Tao. 2024. Uscd: Improving code generation of llms by uncertainty-aware selective contrastive decoding. *arXiv preprint*.

Shuai Wang, Liang Ding, Yibing Zhan, Yong Luo, Zheng He, and Dapeng Tao. 2025. Leveraging metamemory mechanisms for enhanced data-free code generation in llms. *arXiv preprint*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint*.

Kevin Christian Wibisono and Yixin Wang. 2024. In-context learning from training on unstructured data: The role of co-occurrence, positional information, and training data structure. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*.

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.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. In *NAACL*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2024. ROSE doesn't do that: Boosting the safety of instruction-tuned large language models with reverse prompt contrastive decoding. In *Findings of ACL*.

A Datasets

Natural Language Understanding (NLU) Dataset information is detailed in Table 7. All NLU datasets are loaded from the HuggingFace Hub. For most NLU datasets, we report the results on the test set; while for the datasets MNLI and QNLI, we report the results on the validation set due to restricted access to their test sets.

B Templates

The templates of NLU tasks used in this paper are detailed in Table 8.

Dataset	Task	# of Classes	Data Split
SST-2	Sentiment Classification	2	6920/872/1821
SST-5	Sentiment Classification	5	8544/1101/2210
CR	Sentiment Classification	2	3394/0/376
Subj	Subjectivity Analysis	2	8000/0/2000
AgNews	Topic Classification	4	120000/0/7600
MNLI	Natural Language Inference	3	392702/19647/19643
QNLI	Natural Language Inference	2	104743/5463/5463

Table 7: **Details of NLU datasets.**

Task	Prompt	Class
SST-2	Review: "<X>" Sentiment: positive	positive
	Review: "<X>" Sentiment: negative	negative
SST-5	Review: "<X>" Sentiment: terrible	terrible
	Review: "<X>" Sentiment: bad	bad
	Review: "<X>" Sentiment: okay	okay
	Review: "<X>" Sentiment: good	good
	Review: "<X>" Sentiment: great	great
Subj	Input: "<X>" Type: objective	objective
	Input: "<X>" Type: subjective	subjective
CR	Review: "<X>" Sentiment: positive	positive
	Review: "<X>" Sentiment: negative	negative
AgNews	Input: "<X>" Type: world	World
	Input: "<X>" Type: sports	Sports
	Input: "<X>" Type: business	Business
	Input: "<X>" Type: technology	Sci/Tech
MNLI	Premise: <C> Hypothesis: <X> Prediction: entailment	Entailment
	Premise: <C> Hypothesis: <X> Prediction: neutral	Neutral
	Premise: <C> Hypothesis: <X>? Prediction: contradiction	Contradiction
QNLI	<C> Can we know <X>? Yes.	Entailment
	<C> Can we know <X>? No.	Contradiction

Table 8: **Templates of NLU tasks.** Placeholders (e.g., <X> and <C>) will be replaced by real inputs.