Towards LLMs Robustness to Changes in Prompt Format Styles

Lilian Ngweta[1], Kiran Kate[2], Jason Tsay[2], Yara Rizk[2]

[1]Rensselaer Polytechnic Institute, [2]IBM Research

Correspondence: ngwetl@rpi.edu

Abstract

Large language models (LLMs) have gained popularity in recent years for their utility in various applications. However, they are sensitive to non-semantic changes in prompt formats, where small changes in the prompt format can lead to significant performance fluctuations. In the literature, this problem is commonly referred to as prompt brittleness. Previous research on prompt engineering has focused mainly on developing techniques for identifying the optimal prompt for specific tasks. Some studies have also explored the issue of prompt brittleness and proposed methods to quantify performance variations; however, no simple solution has been found to address this challenge. We propose Mixture of Formats (MOF), a simple and efficient technique for addressing prompt brittleness in LLMs by diversifying the styles used in the prompt few-shot examples. MOF was inspired by computer vision techniques that utilize diverse style datasets to prevent models from associating specific styles with the target variable. Empirical results show that our proposed technique reduces style-induced prompt brittleness in various LLMs while also enhancing overall performance across prompt variations and different datasets.

1 Introduction

Large language models (LLMs) are useful for many applications and tasks i.e., content generation, translation, text analysis, etc. One of the popular techniques for adapting pre-trained LLMs to specific tasks that has emerged in recent years is prompt engineering (Liu et al., 2023; Tonmoy et al., 2024; Chen et al., 2023). Prompt engineering involves carefully crafting task-specific instructions and a few input-output demonstrations (prompts) to guide LLMs without changing their parameters (Sahoo et al., 2024). The popularity of prompt engineering can be attributed to the fact that it does not require labeled data and only needs a few demonstrations in prompts containing few-shot examples (Liu et al., 2023). Prompting is also generally computationally cheaper than supervised fine-tuning techniques since the model parameters are not modified (Sahoo et al., 2024).

Existing prompting techniques include zero-shot prompting (Radford et al., 2019), few-shot prompting (Brown et al., 2020), chain-of-thought (CoT) prompting (Wei et al., 2022), and automatic chain-of-thought (Auto-CoT) prompting (Zhang et al., 2023). Most research on prompting techniques has focused on identifying or designing good prompts for specific tasks (Zhou et al., 2023b; Wan et al., 2023). However, a key problem often overlooked by these techniques is the sensitivity of LLMs to meaning-preserving changes in prompts. Examples of such changes include adding extra spaces, replacing two colons with one, changing the order of few-shot examples, or varying the choice of few-shot examples (He et al., 2024; Sclar et al., 2024; Lu et al., 2022; Wan et al., 2023). This problem is sometimes referred to as prompt brittleness (Zhou et al., 2023a). Prompt brittleness contributes to LLMs being unreliable and prevents their adoption in high-risk domains such as healthcare.

In this work, we focus on style-induced prompt brittleness as illustrated in Figure 1, and propose *Mixture of Formats (MOF)* to address it. MOF is a simple and computationally efficient prompting technique where each few-shot example in the prompt is presented in a distinct style. Furthermore, the model is instructed to rewrite each example using a different style, as shown in Figure 2. MOF was inspired by ideas from computer vision that involve learning from datasets with diverse styles to prevent models from associating styles with the target variable (Arjovsky et al., 2019; Kamath et al., 2021; Yin et al., 2021; Wald et al., 2021; Ngweta et al., 2023; Li et al., 2021). We evaluate the effectiveness of MOF prompting using datasets from var-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 529–537
April 30 - May 1, 2025 ©2025 Association for Computational Linguistics

ious tasks within SuperNaturalInstructions (Wang et al., 2022), comparing its performance against *traditional prompts*. Our experiments focus on few-shot prompting, where a *traditional prompt* refers to a regular few-shot prompt, and a *MOF prompt* is a few-shot prompt that has been converted into the MOF style, as demonstrated in Figure 2.

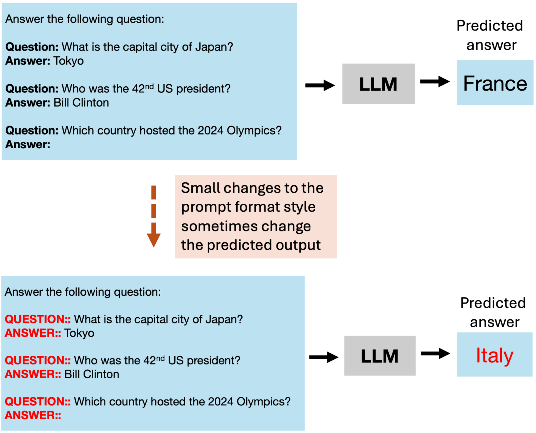

Answer the following question:
Question: What is the capital city of Japan?
Answer: Tokyo
Question: Who was the 42nd US president?
Answer: Bill Clinton
Question: Which country hosted the 2024 Olympics?
Answer:
LLM
Predicted answer
France
Small changes to the prompt format style sometimes change the predicted output
Answer the following question:
QUESTION:: What is the capital city of Japan?
ANSWER:: Tokyo
QUESTION:: Who was the 42nd US president?
ANSWER:: Bill Clinton
QUESTION:: Which country hosted the 2024 Olympics?
ANSWER::
LLM
Predicted answer
Italy

Figure 1: A demonstration of how small changes to the prompt format style can sometimes lead to incorrect predictions in LLMs.

2 Related work

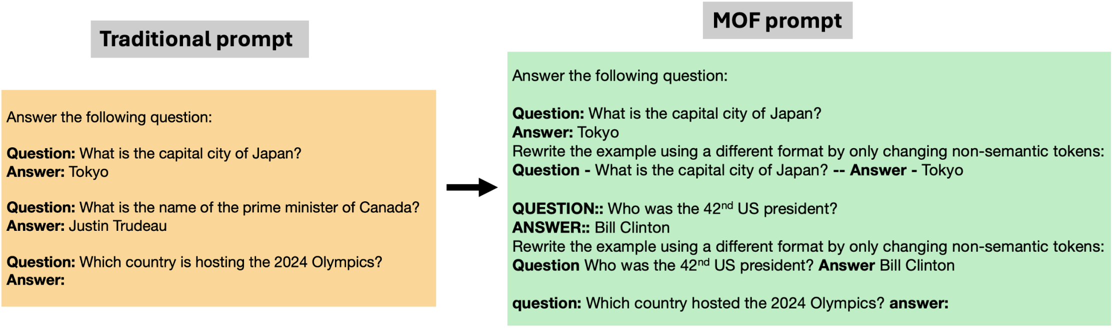
Traditional prompt engineering techniques. Several prompt engineering techniques have been proposed in recent years. Zero-shot prompting is a technique in which a prompt contains a description of the task and no training data is required (Radford et al., 2019). Unlike zero-shot prompting, few-shot prompting adds a few input-output demonstrations to the prompt to further help the model understand the task (Brown et al., 2020). Both zero-shot and few-shot prompting techniques enable the application of LLMs on new tasks without extensive training (Sahoo et al., 2024). For reasoning and logic tasks, prompting techniques that have been proposed include chain-of-thought (CoT) (Wei et al., 2022) and automatic chain-of-thought (Auto-CoT) (Zhang et al., 2023). CoT is a prompting technique that encourages LLMs to do step-by-step reasoning (Wei et al., 2022). Since manually creating CoT examples is time-consuming and not easily scalable, Zhang et al. (2023) proposed Auto-CoT to automatically guide LLMs to generate reasoning steps using a "Let's think step by step" statement in the prompt.

These traditional prompting techniques can be adapted to the MOF format by applying different formatting styles to each prompt example, as demonstrated in Figure 2. In this paper, we focus on the application of MOF to few-shot prompting.

Optimizing for the best prompt. This line of work focuses on optimizing and identifying the most effective prompt for a given task. Zhou et al. (2023b) propose the automatic prompt engineer (APE), an approach that enables the generation and selection of prompt instructions automatically. APE involves analyzing input queries, generating candidate prompt instructions, and then using reinforcement learning to select the best prompt (Zhou et al., 2023b). Similarly, Wan et al. (2023) propose a method where an LLM generates zero-shot outputs for given inputs, followed by selecting high-quality few-shot examples to construct an improved prompt, focusing on consistency, diversity, and repetition. Since automatic prompt optimization (APO) methods focus on optimizing instruction or optimizing few-shot examples, Wan et al. (2024) propose a technique to optimize for both, and compare its performance with the performance of techniques that only optimize instructions or examples. Yang et al. (2024) present Optimization by PROmpting (OPRO), a method that leverages LLMs as optimizers by describing the optimization task in natural language (Yang et al., 2024). Pryzant et al. (2023) propose Prompt Optimization with Textual Gradients (ProTeGi), which employs text gradients guided by beam search and bandit selection techniques for automatic prompt optimization (Pryzant et al., 2023). Additionally, Khattab et al. (2024) introduce DSPy, a framework that replaces hard-coded prompt templates with a systematic approach for building language model pipelines. Other methods for identifying optimal prompts include (Feffer et al., 2024; Sorensen et al., 2022; Yin et al., 2023).

Unlike existing methods in this area that repeatedly search for optimal prompts per task and model, our goal is to reduce style-induced prompt brittleness using an efficient and straightforward recipe illustrated in Figure 2.

Quantifying prompt brittleness in LLMs. Several works have shown that LLMs are sensitive to changes in prompt formats (Sclar et al., 2024; He et al., 2024; Voronov et al., 2024) and to the order of few-shot examples in the prompt (Lu et al., 2022). Sclar et al. (2024) propose FormatSpread, a method to efficiently measure performance variations in LLMs caused by prompt format changes,



Answer the following question:

Question: What is the capital city of Japan?
Answer: Tokyo

Question: What is the name of the prime minister of Canada?
Answer: Justin Trudeau

Question: Which country is hosting the 2024 Olympics?
Answer:

Answer the following question:

Question: What is the capital city of Japan?
Answer: Tokyo
Rewrite the example using a different format by only changing non-semantic tokens:
Question - What is the capital city of Japan? **-- Answer -** Tokyo

QUESTION:: Who was the 42nd US president?
ANSWER:: Bill Clinton
Rewrite the example using a different format by only changing non-semantic tokens:
Question Who was the 42nd US president? **Answer** Bill Clinton

question: Which country hosted the 2024 Olympics? **answer:**

Figure 2: An illustration of how to convert a traditional prompt into a MOF prompt. This example serves as a simple demonstration of the conversion process. In the actual experiments, datasets use various formats such as `Passage:: {} , Answer:: {}` for dataset **task280**, `SYSTEM REFERENCE : {}. ORIGINAL REFERENCE : {}. ANSWER : {}` for dataset **task1186**, and `Tweet:{} , Label:{} , Answer:{}` for dataset **task905**. These formats are generated using FormatSpread (Sclar et al., 2024), as described in Section 3.1. The datasets used are described in Table 3.

by computing the performance difference (*spread*) between the best-performing format and the worst-performing format. Due to the sensitivity of LLMs to prompt format variations, Polo et al. (2024) propose PromptEval, an efficient method for evaluating LLMs on multiple prompts instead of a single prompt. Similarly, Mizrahi et al. (2024) propose metrics for multi-prompt evaluation of LLMs.

While these approaches are valuable tools for quantifying prompt brittleness, our proposed method focuses on mitigating it, particularly the brittleness arising from style variations in prompt formats.

Prompt ensembles. Arora et al. (2022) introduce Ask Me Anything (AMA), a prompting approach that transforms inputs into a question-answering format to encourage open-ended responses. AMA generates multiple imperfect prompts and combines the responses using a weak supervision strategy to produce the final output (Arora et al., 2022). Similarly, Voronov et al. (2024) propose Template Ensembles, an approach that aggregates model predictions across multiple prompt templates. However, both methods are computationally expensive, as they require aggregating predictions from multiple prompts. Furthermore, unlike our proposed method, they do not specifically address prompt brittleness caused by style variations in prompt formats.

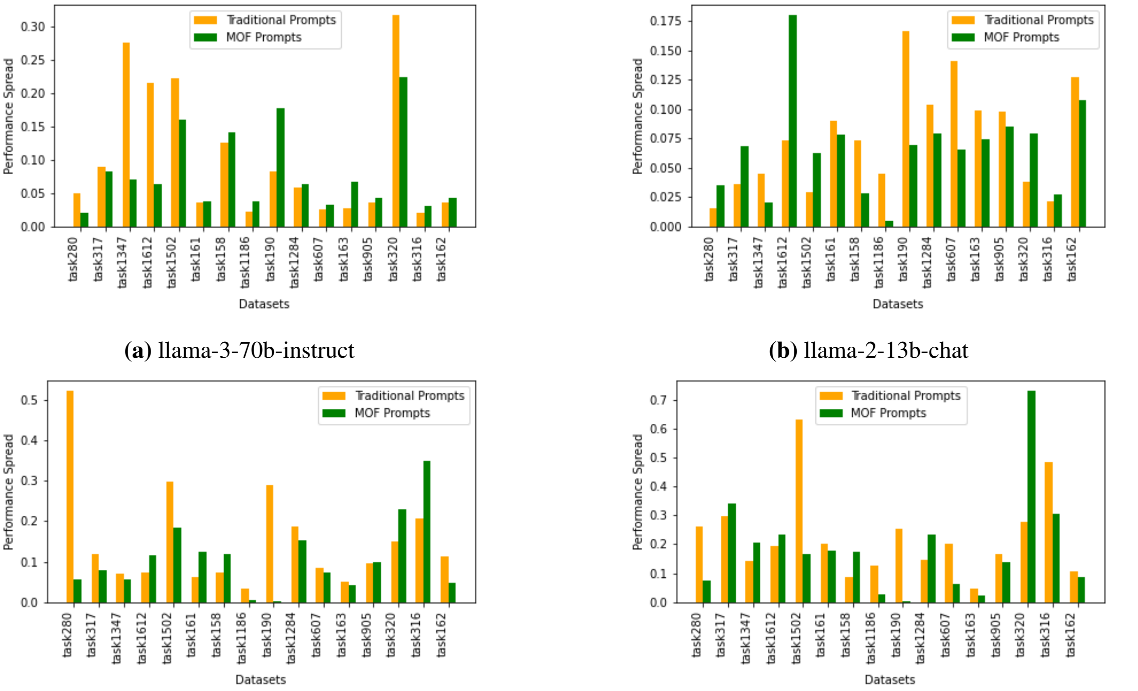
3 Mixture of Formats

Style-induced prompt brittleness in LLMs is similar to problems observed in computer vision, where small changes to an image's style (eg. color or background) can affect the model's ability to make accurate predictions (Nagarajan et al., 2020). In computer vision, various approaches have been developed to address this issue, often involving learning from diverse datasets (Arjovsky et al., 2019; Ngweta et al., 2023; Kamath et al., 2021; Yin et al., 2021; Wald et al., 2021; Li et al., 2021). The underlying idea is that exposure to diverse data points helps the model disassociate styles from the target variable. Drawing inspiration from these techniques, we propose Mixture of Formats (MOF), a novel prompting strategy that deviates from traditional ways of crafting prompts by employing a distinct style format for each few-shot example in the prompt. To further reinforce model understanding, we have the model rewrite the question and answer of each example using a different format style, as illustrated in Figure 2. The effectiveness of this approach is evaluated in the subsequent subsections.

3.1 Experiments

Let X denote input queries for a task, and Y denote the target variable. Given N observations of inputs X and their corresponding targets Y as data $\mathcal{D} = \{X_n, Y_n\}_{n=1}^{N}$, we automatically build a traditional prompt and its MOF prompt version, each containing 5 few-shot examples, and use them for inference with an LLM. The traditional prompt is created using FormatSpread (Sclar et al., 2024), while the MOF prompt is generated by modifying FormatSpread to incorporate diverse formats within the few-shot examples, as illustrated in Figure 2.

Using FormatSpread, we create 10 traditional



(a) llama-3-70b-instruct

(b) llama-2-13b-chat

(c) llama-2-13b

(d) falcon-11B

Figure 3: Comparing the performance *spread* of traditional prompts and MOF prompts. *Spread* is a metric for quantifying style-induced prompt brittleness and it is obtained by taking the difference between the best performing prompt (maximum accuracy) and the worst performing prompt (minimum accuracy). MOF prompts perform comparably or outperform traditional prompts in most datasets and in some datasets, traditional prompts have better performance.

prompt variations and 10 MOF prompt variations. From the 10 prompt variations, for both traditional and MOF prompts, we compute performance accuracies for each prompt format across various tasks. The goal is to compare the style-induced prompt brittleness between traditional prompts and MOF prompts. As in Sclar et al. (2024), we measure brittleness by calculating the performance *spread*, defined as the accuracy difference between the best-performing and worst-performing prompt formats. The evaluation pipelines for traditional and MOF prompts are summarized in Algorithm 1 and Algorithm 2, respectively.

Datasets We perform experiments on datasets covering various tasks from SuperNaturalInstructions (Mishra et al., 2022; Wang et al., 2022). Due to limited computational resources, we randomly selected 16 datasets and for each dataset we use 1000 samples and a batch size of 100. The datasets used are described in Table 3.

Baselines, metrics, and LLMs used In our experiments, we use traditional few-shot prompts as our baselines, where we compare the performance of LLMs when using traditional prompts versus MOF prompts. A primary focus of this work is to determine whether MOF prompting can minimize performance variations (*spread*) in LLMs when prompt format styles change. The performance *spread* is obtained by taking the difference between the highest performing prompt (denoted as "Max Accuracy" in the results tables) and the minimum performing prompt (denoted as "Min Accuracy"). The *spread* value ranges from 0.0 to 1.0, where values closer to 0.0 indicate that the LLM is more robust and less sensitive to style changes, while values closer to 1.0 suggest that the LLM is highly sensitive to these changes. Additionally, for both traditional and MOF prompts, we compute the average accuracy across all 10 prompt variations to assess the overall performance of MOF prompts relative to traditional prompts. We use four LLMs in our experiments: `falcon-11B`, `Llama-2-13b-hf`, `Llama-2-13b-chat-hf`, and `llama-3-70b-instruct`.

We emphasize that while MOF prompting can be applied and compared with other existing traditional prompting techniques, such as automatic

Table 1: Best performing format (*Max Accuracy*) and worst performing format (*Min Accuracy*) results for both traditional prompts and MOF prompts for `llama-3-70b-instruct`. MOF prompts improve the *Min Accuracy* and the *Max Accuracy* over traditional prompts in most cases.

Task	Traditional Prompts		MOF Prompts	
	Min Accuracy	Max Accuracy	Min Accuracy	Max Accuracy
task280	0.811	0.860	**0.880**	**0.900**
task317	0.139	0.229	**0.712**	**0.795**
task1347	0.248	0.524	**0.464**	**0.535**
task1612	0.624	0.839	**0.787**	**0.851**
task1502	0.443	**0.666**	**0.479**	0.639
task161	0.472	0.507	**0.475**	**0.512**

chain-of-thought (Auto-CoT) (Zhang et al., 2023) and the automatic prompt engineer (APE) (Zhou et al., 2023b), this paper focuses on applying MOF prompting to regular few-shot prompting and comparing their performances, due to limited computational resources.

Generating responses for evaluation To generate a response for a given question, a traditional or MOF prompt is combined with the question and then passed to an LLM to generate the response. The generated response is then compared to the ground-truth answer to calculate the model's accuracy.

3.2 Results

We perform experiments to evaluate whether MOF prompts reduce prompt brittleness in LLMs by comparing their *spread* with traditional prompts. We also assess improvements by analyzing the best (Max Accuracy) and worst (Min Accuracy) performing prompts. Finally, we evaluate overall performance by comparing the mean accuracies across all 10 prompt variations for both prompt types.

Minimizing prompt brittleness Figure 3 shows that MOF prompting effectively reduces style-induced prompt brittleness across several datasets and LLMs, with a notable 46% reduction in `task280` using `Llama-2-13b`. While MOF prompts generally perform as well or better than traditional prompts, exceptions occur in `task190` (`llama-3-70b-instruct`), `task1612` (`llama-2-13b-chat`), and `task320` (`falcon-11B`), where traditional prompts perform better. Investigating why MOF fails on these datasets is an important future direction.

Best and worst performing prompts Results for the best-performing prompt (Max Accuracy) and worst-performing prompt (Min Accuracy) for both traditional and MOF prompting are reported in Table 1. We observe that MOF prompting not only reduces spread but also improves both minimum and maximum accuracies. Average accuracy results across all 10 prompt variations for both traditional and MOF prompts are discussed in Appendix A.

4 Conclusion and future work

Addressing prompt brittleness remains a challenge, particularly when caused by changes in prompt format styles. In this work, we introduce a simple and efficient prompting technique, MOF, and evaluate its effectiveness in addressing style-induced prompt brittleness. The preliminary results are promising, with significant improvements over traditional prompting in many datasets, as shown in Figure 3.

Future directions include integrating MOF with techniques like chain-of-thought (CoT) and automatic prompt engineer (APE), comparing its performance with methods that aggregate results from multiple prompts such as AMA (Arora et al., 2022) and Template Ensembles (Voronov et al., 2024), and conducting experiments with larger LLMs like GPT-4, Claude 3.5 Sonnet, Falcon 40B, and Llama 3.1 405B. Additionally, analyzing MOF's failures on certain datasets is a crucial area for further exploration.

We hope this work will inspire further research into addressing prompt brittleness in LLMs, and the code for this project is publicly available on GitHub.[1]

[1] Code: github.com/lilianngweta/mof.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. 2022. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*.

Michael Feffer, Ronald Xu, Yuekai Sun, and Mikhail Yurochkin. 2024. Prompt exploration with prompt regression. *arXiv preprint arXiv:2405.11083*.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? *arXiv preprint arXiv:2411.10541*.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. 2021. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pages 4069–4077. PMLR.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Shuangning Li, Matteo Sesia, Yaniv Romano, Emmanuel Candès, and Chiara Sabatti. 2021. Searching for consistent associations with a multi-environment knockoff filter. *arXiv preprint arXiv:2106.04118*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. 2020. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*.

Lilian Ngweta, Subha Maity, Alex Gittens, Yuekai Sun, and Mikhail Yurochkin. 2023. Simple disentanglement of style and content in visual representations. In *International Conference on Machine Learning*, pages 26063–26086. PMLR.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. 2024. Efficient multi-prompt evaluation of llms. *arXiv preprint arXiv:2405.17202*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. 2021. On calibration and out-of-domain generalization. *Advances in Neural Information Processing Systems*, 34.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, and Sercan O Arik. 2024. Teach better or show smarter? on instructions and exemplars in automatic prompt optimization. *arXiv preprint arXiv:2406.15708*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

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

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. *Preprint*, arXiv:2309.03409.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079, Toronto, Canada. Association for Computational Linguistics.

Mingzhang Yin, Yixin Wang, and David M Blei. 2021. Optimization-based causal estimation from heterogenous environments. *arXiv preprint arXiv:2109.11990*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2023a. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023b. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Appendix

Average accuracy across all 10 prompt variations Up to this point, we have examined the performance in minimizing prompt brittleness, as well as the performance of the best and worst performing prompts. In this section, we focus on the performance of traditional and MOF prompts across all 10 prompt variations for each. The average accuracy across these 10 prompt variations for both traditional and MOF prompts is reported in Table 2. For all LLMs, we find that MOF prompts perform nearly as well as traditional prompts, with MOF prompts generally leading to significant overall mean accuracy improvements.

Algorithm 1 Traditional prompts evaluation pipeline

```
Input: Data $\mathcal{D}$
Create 10 variations of traditional prompts using FormatSpread (Sclar et al., 2024).
Use the created traditional prompt variations to generate responses.
Evaluate each of the 10 traditional prompts and save results.
Compute the average accuracy across all 10 traditional prompt variations.
Identify the best performing prompt, the worst performing prompt, and compute the spread.
Output: Return accuracies for the best performing prompt (max accuracy), worst performing prompt
  (min accuracy), the spread, and the average accuracy across all 10 traditional prompt variations.
```

Algorithm 2 MOF prompts evaluation pipeline

```
Input: Data $\mathcal{D}$
Create 10 variations of MOF prompts using a modified FormatSpread (Sclar et al., 2024) that
  incorporates diverse styles in the few-shot examples as illustrated in Figure 2.
Use the created MOF prompt variations to generate responses.
Evaluate each of the 10 MOF prompts and save results.
Compute the average accuracy across all 10 MOF prompt variations.
Identify the best performing prompt, worst performing prompt, and compute the spread.
Output: Return accuracies for the best performing prompt (max accuracy), worst performing prompt
  (min accuracy), the spread, and the average accuracy across all 10 MOF prompt variations.
```

Table 2: Average accuracy results across 10 prompt variations for traditional prompts (denoted as *Trad Mean Acc*) and MOF prompts (denoted as *MOF Mean Acc*). For all LLMs, MOF prompts perform comparable and in most cases have a higher overall average accuracy than traditional prompts.

(a) Llama-2-13b-chat

Task	Trad Mean Acc	MOF Mean Acc
task280	**0.853**	0.841
task317	0.578	**0.749**
task1612	0.471	**0.490**
task1502	**0.596**	0.579
task161	0.199	**0.278**

(b) Llama-2-13b

Task	Trad Mean Acc	MOF Mean Acc
task280	0.635	**0.842**
task317	0.564	**0.725**
task1612	**0.564**	0.505
task1502	**0.489**	0.485
task161	0.245	**0.371**

(c) falcon-11B

task	Trad Mean acc	MOF Mean acc
task280	0.727	**0.802**
task317	0.501	**0.672**
task1612	**0.638**	0.553
task1502	0.305	**0.493**
task161	**0.390**	0.387

(d) llama-3-70b-instruct

task	Trad Mean acc	MOF Mean acc
task280	0.836	**0.890**
task317	0.154	**0.770**
task1612	0.800	**0.821**
task1502	**0.600**	0.593
task161	**0.496**	0.492

Table 3: Datasets from SuperNaturalInstructions (Mishra et al., 2022; Wang et al., 2022) that we used in our experiments.

Dataset ID	Dataset Description
task280	A text categorization dataset that involves classifying sentences into four types of stereotypes: gender, profession, race, and religion.
task317	A stereotype detection dataset that involves classifying sentences into various types of stereotypes.
task1347	A text matching dataset that involves classifying the semantic similarity of two sentences on a scale of 0 - 5.
task1612	A textual entailment dataset derived from the SICK dataset, that involves accurately classifying labels to show the relationship between two sentences.
task1502	A toxic language detection dataset that involves classifying the type of tweet in HateXplain.
task161	A dataset focused on counting the words in a sentence that contain a specified letter.
task158	A dataset that involves counting the number of times a word occurs in a sentence.
task1186	A text quality evaluation dataset that involves evaluating the naturalness of system generated reference.
task190	A textual entailment dataset that involves choosing whether two given sentences agree, disagree, or neither with each other.
task1284	A text quality evaluation dataset that involves evaluating the informativeness of system generated reference.
task607	A toxic language detection that involves determining whether or not the post is intentionally offensive.
task163	A dataset that involves counting the number of words in the sentence that end with a specified letter.
task905	A toxic language detection dataset that involves determining whether the given category of a tweet is true or false.
task320	A stereotype detection dataset that involves determining whether a given target pertaining to race in two sentences is a stereotype.
task316	A stereotype detection dataset that involves classifying whether a sentence is stereotype or anti-stereotype.
task162	A dataset that involves counting the words in a sentence that begin with a specified letter.