Zero-to-Strong Generalization: Eliciting Strong Capabilities of Large Language Models Iteratively without Gold Labels

Chaoqun Liu[*12] **Qin Chao**[*12] **Wenxuan Zhang**[†23] **Xiaobao Wu**[1]
Boyang Li[1] **Anh Tuan Luu**[1] **Lidong Bing**[23]
[1]Nanyang Technological University, Singapore; [2]DAMO Academy, Alibaba Group, Singapore
[3]Hupan Lab, 310023, Hangzhou, China;
{chaoqun.liu,qin.chao,saike.zwx,l.bing}@alibaba-inc.com
{xiaobao002,boyang.li,anhtuan.luu}@ntu.edu.sg

Abstract

Large Language Models (LLMs) have demonstrated remarkable performance through supervised fine-tuning or in-context learning using gold labels. However, this paradigm is limited by the availability of gold labels, while in certain scenarios, LLMs may need to perform tasks that are too complex for humans to provide such labels. To tackle this challenge, this study explores whether solely utilizing unlabeled data can elicit strong model capabilities. We propose a new paradigm termed *zero-to-strong generalization*. We iteratively prompt LLMs to annotate unlabeled data and retain high-quality labels by filtering. Surprisingly, we obverse that this iterative process gradually unlocks LLMs' potential on downstream tasks. Our experiments on extensive classification and reasoning tasks confirm the effectiveness of our proposed framework. Our analysis indicates that this paradigm is effective for both in-context learning and fine-tuning, and for various model sizes.

1 Introduction

Pre-trained language models (PLMs) have achieved significant improvements through supervised fine-tuning (Radford et al., 2018; Devlin et al., 2019; Wei et al., 2022; Sanh et al., 2022). However, this paradigm often incurs high data costs and requires careful quality control. There are situations where advanced models need to tackle complex tasks that humans cannot fully comprehend or annotate. To study this problem, Burns et al. (2023) consider the analogy of using weak models to supervise strong models. By fine-tuning the strong models on the labels generated by the weak supervisors, the strong student model consistently outperforms

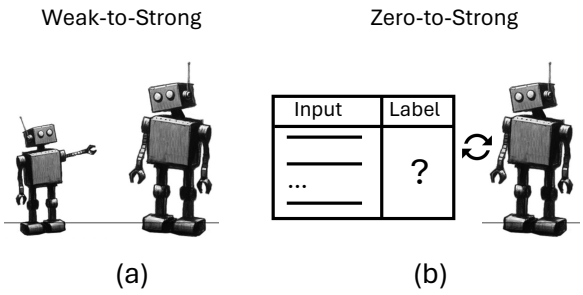


Figure 1: Illustration of (a) weak-to-strong (Burns et al., 2023) and (b) our zero-to-strong analogy. While weak-to-strong uses weak models to supervise strong models, zero-to-strong elicits LLM capabilities without ground-truth labels or weak supervisors.

their weak supervisors, which they call *weak-to-strong generalization*. This phenomenon occurs because strong pretrained models already possess good representations of relevant tasks.

Despite promising, this *weak-to-strong generalization* paradigm has two limitations. Firstly, the student's performance is still constrained by the supervisor's ability to label data, and a weaker supervisor leads to a weaker student. Secondly, the reliance on weak supervisor models restricts its applicability to more scenarios. For example, there may be cases where no weak supervisors are available or humans cannot provide informative supervision in the future.

To address the aforementioned issue, we explore how to harness the capabilities of LLMs without gold (or ground-truth) labels or weak supervisors, a process we refer to as *zero-to-strong generalization*, as illustrated in Figure 1. Previous works have demonstrated that random labels (Min et al., 2022; Yoo et al., 2022) or invalid reasoning paths (Wang et al., 2023a) can also yield good performance, although not as high as with gold labels. Inspired by this, we initially prompt LLMs with random or invalid demonstrations to label the data. We then select a new set of demonstrations based on

[*]Equal contribution. Chaoqun Liu and Qin Chao are under the Joint PhD Program between DAMO Academy and Nanyang Technological University.

[†]Wenxuan Zhang is the corresponding author.

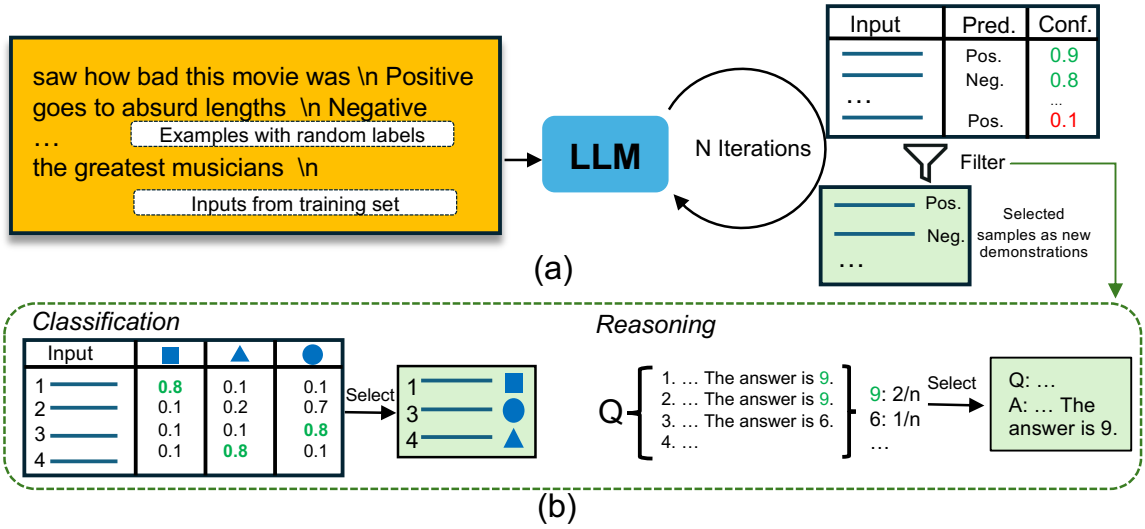


Figure 2: Illustration of (a) zero-to-strong generalization on a sentiment analysis task and (b) the filtering process. For classification tasks, we select demonstrations by ranking the probabilities for each label. For reasoning tasks, we select the most confident answers based on self-consistency (Wang et al., 2023b).

confidence levels and prompt the LLMs again, repeating this process iteratively. This process allows us to achieve strong performance on tasks without needing gold-labeled data or weak supervisors.

We conducted experiments on 17 classification tasks, 2 extreme-label classification tasks, and 2 reasoning tasks to demonstrate the effectiveness of our proposed methods. Surprisingly, our method not only achieves performance comparable to but even outperforms in-context learning with gold labels for some tasks. We hypothesize that our method selects more suitable samples for demonstrations over iterations, which leads to high performance. Through careful analysis, we find that zero-to-strong learning is more effective for stronger models and more complex tasks. Additionally, it also works for fine-tuning and with larger models.

Our main contributions are summarized below:

- We propose a simple yet effective framework called zero-to-strong generalization, which elicits the strong capabilities of LLMs iteratively without gold labels.
- We demonstrate the effectiveness of our zero-to-strong learning with extensive experiments on 17 classification tasks, 2 extreme-label classification tasks, and 2 reasoning tasks.
- We analyze the underlying reasons why zero-to-strong learning is effective and discover that its benefits extend to fine-tuning and larger models.

2 Methodology

This section begins with the problem definition, followed by our proposed zero-to-strong learning framework.

2.1 Problem Definition

In our setting, we assume the absence of gold labels, simulating situations where problems are so complex that human annotations are unreliable. However, we still possess minimal information about the problems. For instance, we know the label space $\mathcal{C}$ in a classification problem, and for a generation problem, the output format is defined. Additionally, we have access to a few inputs $x_1, \ldots, x_k$ without gold labels.

2.2 Zero-to-Strong Generalization

Figure 2 illustrates our overall framework, comprising demonstration construction, response generation, sample selection, and iterative evolution.

Demonstration construction. While we lack access to gold labels, we can create demonstrations by randomly sampling from the label space. For classification tasks, labels can be drawn as $\tilde{y} \sim \mathcal{C}$. For reasoning tasks, we can manually generate outputs for a few examples, focusing on maintaining the correct format rather than ensuring complete accuracy.

Response generation. The generated demonstrations are prepended to the input in the training set to form the LLM prompts. By prompting the

LLMs, we generate both pseudo labels and their confidence for the training set samples. For classification tasks, we set the temperature to 0 and predict the labels using $\arg\max_{y \in \mathcal{C}} P(y|x)$, where x is the text input and $\mathcal{C}$ is a limited set of potential labels. We use the normalized probability $P(y|x)$ as the confidence. For reasoning tasks, we set the temperature to 0.7 to sample diverse reasoning paths, selecting the most consistent final answer as the prediction. This method is similar to self-consistency (Wang et al., 2023b), and we further calculate the ratio of consistent paths to the total number of paths as the confidence for each sample.

Sample selection. After generating the responses for all the training samples, we select the k most confident samples for the next iteration. For classification tasks, we uniformly select the top-k most confident samples across the label space. For reasoning tasks, we first identify the top-k questions with the highest confidence. Then, for each question, we randomly select one path from the consistent paths. The selection process is illustrated in Figure 2(b).

Iterative evolution. The selected samples and their predictions will serve as demonstrations for the next round, with this process repeating for several iterations and aiming for progressive performance improvement.

During the evaluation, we set the temperature to 0 and generate final predictions using the same method as in the response generation stage. The zero-to-strong algorithm for classification tasks is detailed in Algorithm 1 in Appendix A.1.

3 Experiments

We evaluate our proposed framework with two pretrained LLMs: Meta-Llama-3-8B (Llama-3-8B) (Dubey et al., 2024) and Mistral-7B-v0.1 (Mistral-7B) (Jiang et al., 2023). All the experiments are conducted on Nvdia A800 GPUs.

3.1 Tasks

We assess our framework's effectiveness through three tasks: standard text classification, extreme-label classification, and reasoning. Despite being a subtype of classification, extreme-label classification is treated separately due to its significantly larger class count.

Classification tasks. Following Yoo et al. (2022), we evaluate 17 widely-used text classification tasks, with dataset details in Table 6 in the Appendix. Evaluations are conducted in 4-shot, 8-shot, and 16-shot, using manual templates from Yoo et al. (2022).

Extreme-label classification tasks. Extreme-label classification poses greater challenges than traditional classification due to the large number of labels (Li et al., 2024). For evaluation, we selected the GoEmotions dataset with 28 classes (Demszky et al., 2020) and banking77 with 77 classes (Casanueva et al., 2020). Due to resource limitations, we sampled 1,000 instances from the training set and 500 from the test set. Dataset details can be found in Table 7 in the Appendix.

Reasoning tasks. We choose GSM8k (Cobbe et al., 2021a) and SVAMP (Patel et al., 2021) for evaluation, as both require multi-step reasoning. Details of the datasets are in Table 8 in the Appendix. We selected up to 1,000 samples from the training set and used the entire test set for our experiment. Additionally, we generated 10 diverse reasoning paths for each sample during response generation.

3.2 Baseline Methods

We compare zero-to-strong with the following baseline methods:

Zero-shot methods. This setting does not use labeled data as demonstrations. For text and extreme-label classification tasks, predictions are made via $\arg\max_{y \in \mathcal{C}} P(y|x)$, where x is the text input and $\mathcal{C}$ is a limited label set. For reasoning tasks, we adopt the Zero-shot-CoT approach (Kojima et al., 2023), prompting LLMs with "Let's think step by step" and concluding with "Therefore, the answer (Arabic numerals) is" to obtain the final result.

Few-shot with gold labels. For classification and extreme-label classification tasks, we sample k input-label pairs $(x_1, y_1) \dots (x_k, y_k)$ from the training set either randomly or uniformly based on the label space. We then make predictions via $\arg\max_{y \in \mathcal{C}} P(y|x_1, y_1 \dots x_k, y_k, x)$. For reasoning tasks, we use a fixed set of demonstrations $(x_1, r_1, y_1) \dots (x_k, r_k, y_k)$ to prompt LLMs, where r_k represents the reasoning steps, following Wei et al. (2023). The demonstrations are shown in Table 11 in the Appendix. The final answer is extracted using a regular expression.

Task	Setting	Llama-3-8B			Mistral-7B		
		4-shot	8-shot	16-shot	4-shot	8-shot	16-shot
Classification	zero-shot	40.7	40.7	40.7	36.1	36.1	36.1
	random label	42.7	50.3	43.8	45.3	51.0	45.9
	gold label	53.3	56.6	61.1	57.5	57.5	**60.5**
	ours (zero-to-strong)	**57.5**	**63.2**	**61.4**	**61.1**	**62.4**	60.1
Extreme-label Classification	zero-shot	21.4	21.4	21.4	**23.9**	23.9	23.9
	random label	4.5	3.7	2.5	5.3	3.6	2.3
	gold label	21.0	26.5	29.1	17.1	**26.1**	26.4
	ours (zero-to-strong)	**24.6**	**27.2**	**33.4**	21.1	23.3	**32.7**

Table 1: Average Macro-F1 (%) of Llama-3-8B and Mistral-7B on 17 classification and 2 extreme-label classification tasks.

Setting	Llama-3-8B	Mistral-7B
zero-shot	53.5	40.3
invalid	38.9	35.4
gold label	62.2	**53.4**
ours (zero-to-strong)	**64.2**	49.0

Table 2: Average accuracy (%) of Llama-3-8B and Mistral-7B on reasoning tasks.

Few-shot with invalid labels. In classification and extreme-label classification, demonstrations are generated by assigning random labels rather than using the actual data labels. Each x_i ($1 \leq i \leq k$) is paired with a randomly sampled label $\tilde{y}_i$ from $\mathcal{C}$. The sequence $(x_1, \tilde{y}_1) \ldots (x_k, \tilde{y}_k)$ is then used to make a prediction by maximizing $\arg\max_{y \in \mathcal{C}} P(y|x_1, \tilde{y}_1 \ldots x_k, \tilde{y}_k, x)$. For reasoning tasks, we reused demonstrations with the "no coherence" setting (Wang et al., 2023a), meaning the rationales are out of order, as shown in Table 12 in the Appendix.

To ensure reproducibility, we set the evaluation temperature to 0. Results for gold-label, invalid labels, and zero-to-strong are averaged over three seeds to sample the training set for demonstrations. For methods other than zero-shot, initial demonstrations are sampled using two approaches: 1) random initialization — random sampling from the training set, and 2) uniform initialization — sampling an equal number of instances from each class.

3.3 Main Results

Table 1 presents the main results for classification and extreme-label classification tasks. Our zero-to-strong method for Llama-3-8B consistently outperforms other approaches across all shots settings, demonstrating its effectiveness. It also yields the best results with shots lower than 16 for Mistral-7B. We believe this difference stems from Llama-3-8B's superior capabilities, as zero-to-strong performance relies on inherent capabilities gained during pre-training. Overall, extreme-label classification tasks show lower performance compared to standard tasks, emphasizing their increased difficulty. Poor performance in random label settings underlines the necessity of accurate labels for these challenging tasks. Additionally, the number of demonstrations significantly affects extreme-label classification, as performance with gold-label and zero-to-strong settings improves with more demonstrations, while random-label performance declines.

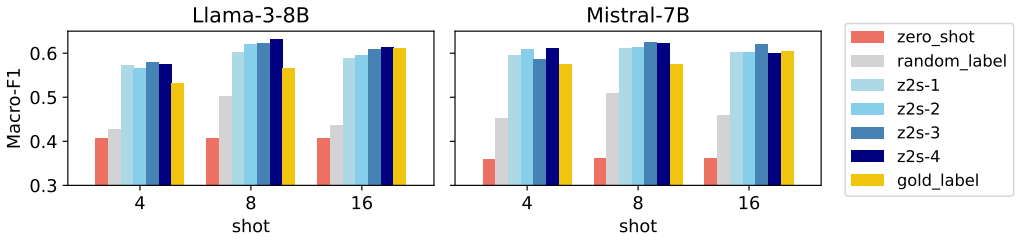
Table 2 presents the average accuracies for the two reasoning tasks. Our zero-to-strong method outperforms other approaches using Llama-3-8B, yet it still lags behind the few-shot method with gold labels using Mistral-7B. This trend aligns with classification and extreme-label classification results, indicating that zero-to-strong is more effective with stronger models. As models continue to improve in the future, our approach may gain even more advantages.

3.4 Analysis

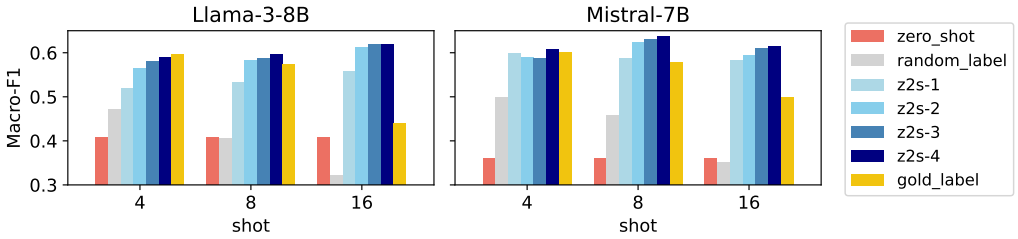
The zero-to-strong performance is promising. To better understand its behavior and underlying reasons, we conduct the following analysis.

3.4.1 How does the performance improve over the iterations?

Classification tasks. The detailed results for 17 classification tasks are shown in Figure 3. It can be seen that for both models, zero-to-strong can achieve comparable or better results than few-shot with gold labels within 4 rounds of iteration. We hypothesize that the zero-to-strong method selects the



(a) Random initialization.



(b) Uniform initialization.

Figure 3: Average macro-F1 for 17 classification tasks, using two LLMs and two initialization settings. "z2s-*i*" means the *i*th round of iteration for zero-to-strong method.

most confident samples as demonstrations, which is superior to randomly sampling from gold labels. Zero-to-strong also has a big advantage over few-shot with random labels (please note that few-shot with random labels can be regarded as the 0th round for zero-to-strong). We also notice that for some settings LLMs improve iteration by iteration but the benefits diminish after certain rounds and the performances fluctuate. In addition, the phenomenon exists for all numbers of shots.

Extreme label classification. The results for GoEmotions are shown in Figure 4 and the results for banking77 are shown in Figure 11 in the Appendix. With more demonstrations, few-shots with gold labels perform better with random initialization. It is interesting that when the number of shots is small, few-shot with gold labels underperforms zero-shot setting. We hypothesize that when the number of shots is small, it cannot cover all the labels and make the distribution of the demonstration deviate from the test set. For few-shot with random labels, more demonstrations hurt the performance. This is reasonable as more demonstrations result in more wrong demonstrations, which deteriorate performance. Interestingly, zero-to-strong outperforms few-shot with gold labels in all settings for GoEmotions but the relative performance depends on the initialization settings and the number of shots, which again confirms the effectiveness of zero-to-strong method.

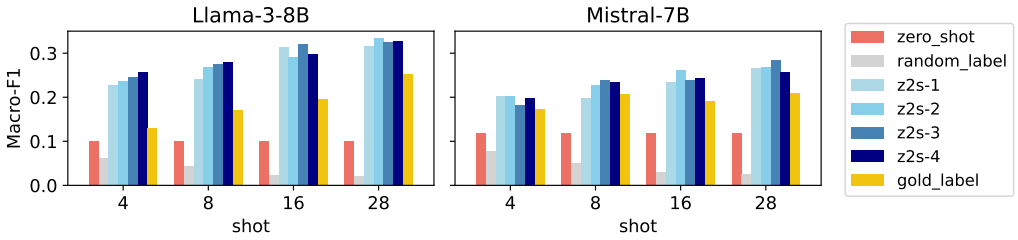
Reasoning tasks. The results for the two reasoning tasks are shown in Figure 5. For GSM8K, zero-to-strong improves performance iteration by iteration and approaches few-shot with gold labels after 4 iterations. For SVAMP, zero-to-strong outperforms few-shot with gold labels after a few iterations. We hypothesize that the initial demonstrations with gold label are not optimal for SVAMP and we can generate better demonstrations for this task with zero-to-strong approach.

3.4.2 What happens during the iterations?

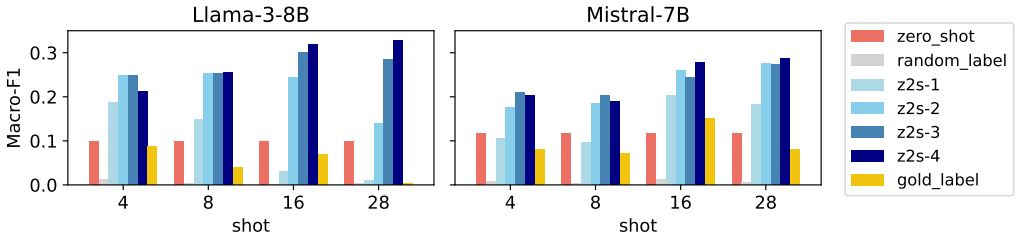
To further understand the mechanics behind zero-to-strong approach, we conduct more analysis on GoEmotions and GSM8K.

Does the confidence correlate with the accuracy? Our sample selection process is based on the hypothesis that predictions with higher confidence will have higher accuracy. To verify this hypothesis, we plot the distributions of the sample confidence and their accuracy in Figure 6. It can be seen that accuracy is highly correlated with confidence. Initially, more samples have low confidence and low accuracy. After several iterations, more samples have higher confidence and higher accuracy. This observation explains why the model performs better and better.

Do more iterations help with the final performance? In Section 3.3, we initially set the maximum number of iterations to 4. In some cases, performance consistently improved with each itera-



(a) Random initialization.



(b) Uniform initialization.

Figure 4: Average macro-F1 for GoEmotions, using two LLMs and two initialization settings.

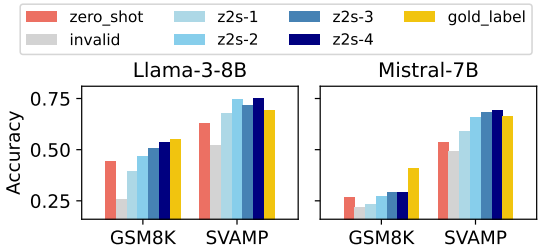


Figure 5: Accuracy for the two reasoning tasks.

Setting	invalid	z2s-1	z2s-2	z2s-3	z2s-4
Invalid Reasoning	48.8	51.6	54.5	50.7	51.9
No coherence	25.9	46.7	51.0	46.6	51.8
No coherence for BOs	43.9	52.6	54.7	54.2	53.8
No coherence for LTs	29.0	47.3	52.8	52.7	50.3
No relevance	3.9	2.7	2.6	2.7	2.8
No relevance for BOs	37.0	53.2	49.6	51.9	51.6
No relevance for LTs	27.8	47.7	49.4	49.5	51.9
Invalid RnA	38.1	46.0	51.4	51.0	50.9

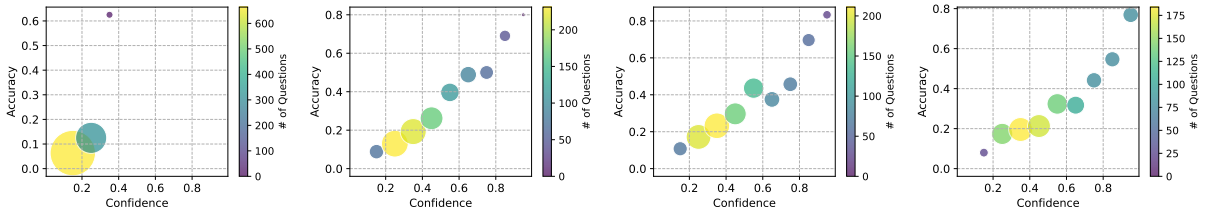
Table 3: GSM8K with different invalid demonstrations for Llama-3-8B. The zero_shot score is 44.3, while the few-shot with gold_label is 55.0. "BO" refers to bridging objects and "LT" refers to "language templates". "RnA" refers to "reasoning and answer".

tion. However, in other cases, performance reached a plateau after a certain number of iterations and subsequently fluctuated. To further explore the models' performance over a greater number of iterations, we extended the total number of iterations to 9. The results, depicted in Figure 7, indicate that performance does not improve beyond a certain point. We hypothesize that once the optimal demonstrations are selected, additional iterations do not contribute to further improvements.

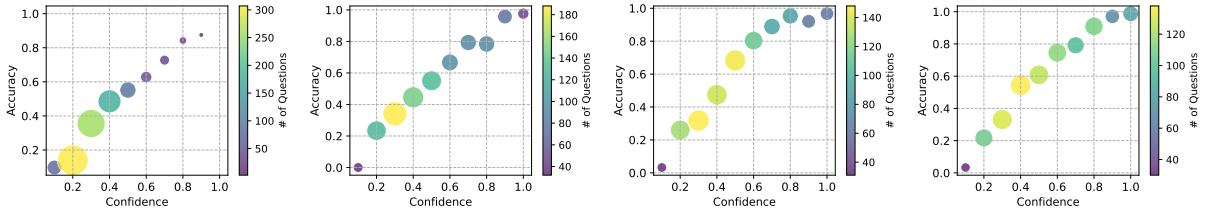
Are the demonstrations more and more confident and accurate over iterations? We select the demonstrations for the next iteration based on confidence. Thus we expect the confidence to increase over iterations. As shown in Figure 8, 9 and 10 (in the Appendix), the confidence for both GoEmotions and GSM8K increases steadily but saturates after a few iterations. For GoEmotions, confidence for the smaller number of shots is larger and saturates faster. This is expected, as it is harder to get more confident samples. It is also interesting that for GoEmotions, random initialization converges faster than uniform initialization, which is also observed in Figure 4. The possible reason is that the training set is not uniform, thus it is better to initialize the demonstrations randomly.

Even though we select the most confident samples for each iteration, we cannot guarantee the accuracy of the selected demonstrations. As shown in Figure 8(b) and 9(b), the accuracy of the demonstrations fluctuates or even decreases after certain iterations. This is a possible reason why the performances on evaluation sets fluctuate after certain iterations.

Does it work with different initial demonstrations for reasoning tasks? In the previous experiments, we used the "no coherence" demonstration for initialization. To evaluate whether our method



(a) Accuracy vs confidence for GoEmotions from iteration 1 (left) to iteration 4 (right).



(b) Accuracy vs confidence for GSM8K from iteration 1 (left) to iteration 4 (right).

Figure 6: The relation between accuracy and confidence of the answers for the training set from iteration 1 to iteration 4. The confidence of GoEmotions and GSM8K is calculated based on the methods described in Section 2.2. After each iteration, more samples are becoming more confident and accurate.

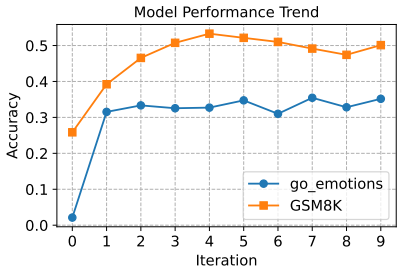
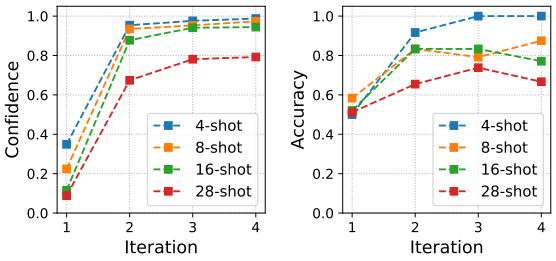


Figure 7: The accuracy for more iterations for zero-to-strong on GSM8K and GoEmotions. The evaluation is on Llama-3-8B.



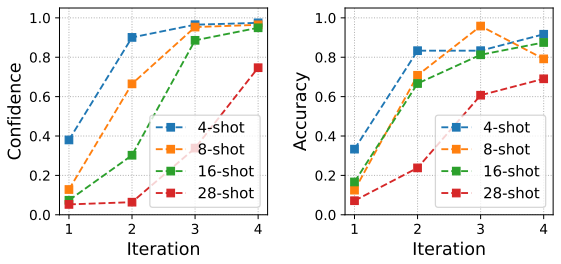
(a) Confidence (b) Accuracy

Figure 8: Confidence and accuracy of demonstrations over iterations for GoEmotions with random initialization.

applies to general incorrect demonstrations, we tested other settings from Wang et al. (2022). Additionally, we manually created a new set of demonstrations featuring invalid reasoning and incorrect final answers but containing relevant bridging objects and language templates, as illustrated in Table 13 in the Appendix. We generate 5 reasoning paths during response generation for this analysis. The results are presented in Table 3. From the results, it is evident that the zero-to-strong method achieves accuracies greater than 50% across all settings, except for the "no relevance" condition. This indicates that providing relevant demonstrations is crucial for the zero-to-strong method to be effective. Fortunately, this requirement is manageable for humans, as providing incorrect but relevant reasoning paths and final answers is not hard.

3.4.3 Does it work for fine-tuning besides in-context learning?

We further investigate the impact of incorporating fine-tuning with LoRA (Hu et al., 2021) into our framework. We first generate the labels for the training set with ICL and demonstrations with random labels. Then we filter the samples and fine-tune the model with the pseudo training set. After that, we generate the new labels with the fine-tuned model in a zero-shot manner. We repeat the above process for several iterations, as detailed in Appendix A.2.1. Optionally, we can fine-tune the model with samples labeled after four rounds of zero-to-strong with ICL. As shown in Table 4, fine-tuning also improves progressively, notably surpassing few-shot results with gold labels for GoEmotions.



(a) Confidence (b) Accuracy

Figure 9: Confidence and accuracy of demonstrations over iterations for GoEmotions with uniform initialization.

tasks	ZS	ft1	ft2	ft3	ft4	z2s-4+ft	GL
GoE	9.9	25.3	26.6	26.7	26.0	31.7	17.2
GSM8K	44.3	30.2	49.7	51.1	50.3	50.3	55.9

Table 4: Fine-tuning performance for Llama-3-8B. "GoE" refers to "GoEmotions". Results are averaged over 3 seeds. "ZS" refers to "zero-shot". "ft" stands for "fine-tuning". "GL" refers to "gold label".

model	ZS	INV	z2s-1	z2s-2	z2s-3	z2s-4	GL
Llama-3-70B	73.7	30.3	60.7	76.7	80.1	80.7	82.3
Mixtral-8x22B	61.0	19.3	56.7	71.2	72.4	69.8	67.9

Table 5: Accuracies on GSM8K with larger models. "ZS" refers to "zero-shot". "INV" refers to "invalid". "GL" refers to "gold label".

3.4.4 Does it work for larger models?

Even though smaller LLMs are more computationally efficient, larger models normally have better performances. To assess the effectiveness of our approach on larger models, we evaluated it on two larger models: Meta-Llama-3-70B (Llama-3-70B) (Dubey et al., 2024) and Mixtral-8x22B-v0.1 (Mixtral-8x22B) (Jiang et al., 2024) on GSM8K. As shown in Table 5, zero-to-strong with the two models outperforms the zero-shot setting and achieves comparable performance with few-shot with gold labels, which is consistent with that observed on smaller models, suggesting that our method generalizes well across models of varying sizes.

4 Related Work

Weak-to-strong generalization. In the future, advanced models will handle complex tasks with only weak human supervision. To study this, Burns et al. (2023) proposed using weak supervisor models to elicit the capabilities of stronger student models. Their findings revealed that, after fine-tuning, the strong student models consistently outperformed the weak supervisor models, a phenomenon they term *weak-to-strong generalization*. In contrast to transferring knowledge from strong models to models (Meng et al., 2022; Ye et al., 2022), this learning paradigm is a specific type of weakly-supervised learning (Bach et al., 2017), where models are trained with noisy or biased labels (Bellamy et al., 2019; Song et al., 2022; Liu et al., 2023). Our work eliminates the necessity of weak models or weak labels for supervision. Instead, we utilize minimal supervision, such as the label space or incorrect initial demonstrations, to elicit the capabilities of large language models. Other research has proposed self-improvement of LLMs using labeled or unlabeled data (Huang et al., 2022; Li and Qiu, 2023; Zelikman et al., 2022) for reasoning tasks. In contrast, we aim to propose a general framework for learning new tasks without labeled data.

Understanding In-context learning. In-context learning (ICL) (Brown et al., 2020) can effectively learn new tasks with a few demonstrations, but its mechanism is still under discussion. Previous research (Lu et al., 2021; Zhao et al., 2021; Su et al., 2022) found that ICL is sensitive to the demonstration samples, their order, and their diversity. Studies by Min et al. (2022) and Wang et al. (2023a) discovered that even random labels for classification or invalid demonstrations for reasoning tasks can yield good performance, suggesting that gold labels are not always necessary. However, Yoo et al. (2022) showed that correct input-label mappings can have varying impacts through extensive experiments. Recently, Wang et al. (2024) found that learning to retrieve in-context examples helps improve the performance, but the gold labels are needed. In contrast, our work achieves strong performance with random or invalid labels and further improves iteratively to attain even better results.

5 Conclusion

In this work, we propose a new framework called *zero-to-strong generalization*. Without gold label data or weaker supervisors, we can elicit the capabilities of LLMs iteratively through prompting and filtering. Experiments on classification and reasoning tasks demonstrate the effectiveness of this framework. Further analysis shows that by selecting the most confident samples as demonstra-

tions for the next iteration, we also select more accurate and more suitable demonstrations. This framework also generalizes well to fine-tuning and larger models. Our work demonstrates the feasibility of eliciting the capabilities of LLMs with minimal supervision. In the future, we plan to explore *zero-to-strong generalization* in more diverse and challenging tasks.

Limitations

Our framework is restricted to tasks with a single definitive correct answer. For instance, sentences in glue-sst2 (Socher et al., 2013) can be either positive or negative, and the final answer in GSM8k (Cobbe et al., 2021b) must be a single number. This uniqueness of the final answer allows us to calculate the confidence of the generated responses. However, for open-ended tasks like story writing, our method is not applicable, as we cannot determine the confidence level of the generated content and leave this as future work.

Acknowledgements

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL). Chaoqun Liu extends his gratitude to Interdisciplinary Graduate Programme and College of Computing and Data Science of NTU, for their support.

References

Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *International Conference on Machine Learning*, pages 273–282. PMLR.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. 2019. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1.

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. *arXiv preprint*. ArXiv:2005.14165 [cs].

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023. Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision. *arXiv preprint*. ArXiv:2312.09390 [cs].

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

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. 2021a. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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. 2021b. Training Verifiers to Solve Math Word Problems. *arXiv preprint*. ArXiv:2110.14168 [cs].

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi.

2020. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint*. ArXiv:1810.04805 [cs].

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint*. ArXiv:2106.09685 [cs].

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

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

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *arXiv preprint*. ArXiv:2401.04088 [cs].

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. *arXiv preprint*. ArXiv:2205.11916 [cs].

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context LLMs Struggle with Long In-context Learning. *arXiv preprint*. ArXiv:2404.02060 [cs].

Xiaonan Li and Xipeng Qiu. 2023. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374.

Chaoqun Liu, Wenxuan Zhang, Guizhen Chen, Xiaobao Wu, Anh Tuan Luu, Chip Hong Chang, and Lidong Bing. 2023. Zero-shot text classification via self-supervised tuning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1743–1761, Toronto, Canada. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*, 65(4):782–796.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Clara H. McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: Matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3458–3465, New York, NY, USA. Association for Computing Machinery.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. *arXiv preprint*. ArXiv:2202.04538 [cs].

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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal 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 Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv preprint*. ArXiv:2110.08207 [cs].

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system. In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 93–106, Barcelona, Spain (Online). Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to Retrieve In-Context Examples for Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards process-oriented, modular, and versatile question generation that meets educational needs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 291–302, Seattle, United States. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint*. ArXiv:2203.11171 [cs].

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. *arXiv preprint*. ArXiv:2109.01652 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint*. ArXiv:2201.11903 [cs].

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. *arXiv preprint*. ArXiv:2202.07922 [cs].

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations. *arXiv preprint*. ArXiv:2205.12685 [cs].

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. STaR: Bootstrapping Reasoning With Reasoning. *arXiv preprint*. ArXiv:2203.14465 [cs].

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

A Appendix

A.1 Methodology

The algorithm for zero-to-strong on classification tasks is shown in Algorithm 1.

Algorithm 1 Zero-to-Strong

Require: A LLM with $\Pr(y|x)$ accessible.
Require: Input data X, and the label space $\mathcal{C}$
Require: Max iterations M, number of demos K
Initial state: D_0, contains K random labelled demonstrations from X
while Iter $t < M$ **do**
  Calculate $\hat{y} = \arg\max_{y \in \mathcal{C}} P(y|D_{t-1}; x)$;
  Sort the $\hat{Y} = \{\hat{y_1}, ..., \hat{y_i}\}$ in descending order of probability;
  $D_t = \{\}$
  while $|D_t| < K$ **do**
    if $\hat{y_i} \notin D_{t-i}$ **then**
      $D_t = D_t \cup \hat{y}$;
      $i = i + 1$;
    end if
  end while
end while
return $\hat{Y}$

A.2 Experiment Setup

The list and statistics of 17 classification tasks, 2 extreme-label classification tasks, and 2 reasoning tasks are shown in Table 6, 7 and 8, respectively. The 17 text classification datasets span a variety of tasks such as sentiment analysis, paraphrase detection, natural language inference, and hate speech detection. GoEmotions is an emotion classification task and banking77 is an intent classification task.

Dataset	#Train	#Test	#C
glue-sst2 (Socher et al., 2013)	67,349	872	2
glue-rte (Dagan et al., 2005)	2,490	277	2
glue-mrpc (Dolan and Brockett, 2005)	3,668	408	2
glue-wnli (Levesque et al., 2012)	635	71	2
super_glue-cb (de Marneffe et al., 2019)	250	56	3
trec (Voorhees and Tice, 2000)	5,452	500	5
financial_phrasebank (Malo et al., 2014)	1,181	453	3
poem_sentiment (Sheng and Uthus, 2020)	843	105	3
medical_questions_pairs (McCreery et al., 2020)	2,438	610	2
sick (Marelli et al., 2014)	4,439	495	3
hate_speech18 (de Gibert et al., 2018)	8,562	2,141	4
ethos-national_origin (Mollas et al., 2022)	346	87	2
ethos-race (Mollas et al., 2022)	346	87	2
ethos-religion (Mollas et al., 2022)	346	87	2
tweet_eval-hate (Barbieri et al., 2020)	9,000	1,000	2
tweet_eval-stance_atheism (Barbieri et al., 2020)	461	52	3
tweet_eval-stance_feminist (Barbieri et al., 2020)	597	67	3

Table 6: Data splits of the 17 classification tasks (#C means number of classes.)

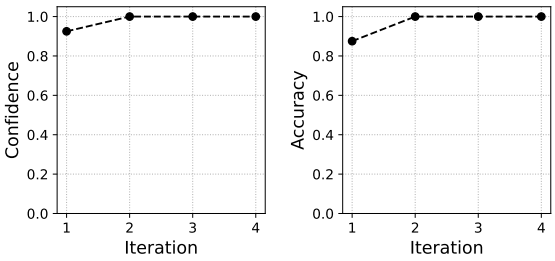
Dataset	#Train	#Test	#Classes
GoEmotions (Demszky et al., 2020)	36308	4590	28
banking77 (Casanueva et al., 2020)	10003	3080	77

Table 7: Data splits of the 2 extreme-label classification tasks.

For the 17 classification tasks, we adopt the manual templates and verbalizers from Yoo et al., 2022 if possible. Examples for some tasks are shown in Table 9. The templates for the two extreme-classification tasks are shown in Table 10. The newly created template for "invalid reasoning and answer" is shown in Table 13. We keep all the questions the same and modify the reasoning paths and the final answer to make sure they are wrong.

Dataset	# Train	# Test
GSM8k (Cobbe et al., 2021a)	7473	1319
SVAMP (Patel et al., 2021)	700	300

Table 8: Data splits of the 2 reasoning tasks.

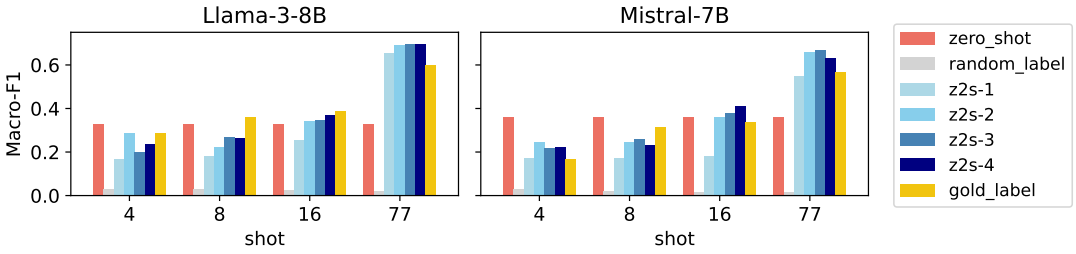


(a) Confidence (b) Accuracy

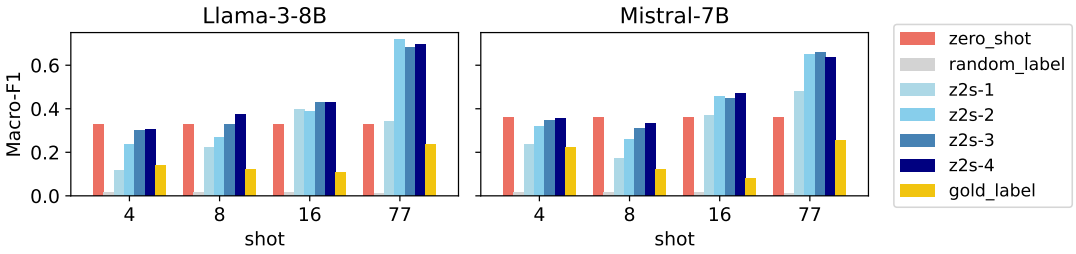
Figure 10: Confidence and accuracy of demonstrations over iterations for GSM8K.

A.2.1 Fine-tuning setup

For the fine-tuning experiments, as mentioned in Section 3.4.3, we filter out low-quality training data before each iteration of fine-tuning. For GoEmotions, according to the probability, we retain only the top $\frac{1}{|C|}$ data points for each class from the label space C. This results in duplicated records with different labels. These labels are noisy but still useful for our fine-tuning process. For GSM8K, we generate 5 paths for each training data and use self-consistency to select confident paths. In all fine-tuning experiments, we set the learning rate to $2e-5$ and train for 3 epochs.



(a) Random initialization.



(b) Uniform initialization.

Figure 11: Average macro-F1 for banking77, using two LLMs and two initialization settings.

Dataset	Manual Template	Verbalizer
glue-sst2	Review: the greatest musicians Sentiment:	negative, positive
glue-wnli	I stuck a pin through a carrot. When I pulled the pin out, it had a hole. The question is: The carrot had a hole. True or False? answer:	True, False
super_glue-cb	That was then, and then's gone. It's now now. I don't mean I 've done a sudden transformation. The question is: she has done a sudden transformation True or False? answer:	True, False, Not sure
trec	Question: What films featured the character Popeye Doyle ? Type:	description, entity, expression, human, number, location
sick	A brown dog is attacking another animal in front of the man in pants The question is: Two dogs are wrestling and hugging True or False? answer:	True, Not sure, False
tweet_eval-hate	Tweet: When cuffin season is finally over Sentiment:	favor, against

Table 9: Examples of templates for classification tasks. Texts in blue are templates.

Dataset	Template	Verbalizer
GoEmotions	comment: This shirt IS a problem. Get rid of it. emotion category:	admiration, amusement, anger, annoyance...
banking77	service query: When did you send me my new card? intent category:	activate my card, age limit, apple pay or google pay...

Table 10: Templates for the 2 extreme-label classification tasks. Texts in blue are the templates.

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There are originally 3 cars. 2 more cars arrive. 3 + 2 = 5. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Originally, Leah had 32 chocolates. Her sister had 42. So in total they had 32 + 42 = 74. After eating 35, they had 74 - 35 = 39. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he gave Denny 20 - 12 = 8. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. 5 + 4 = 9. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. After losing 2 more, he had 35 - 2 = 33 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8.

Table 11: Demonstrations for gold label for reasoning tasks.

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: Then there were 21 - 15 = 6 trees after the Grove workers planted some more. So there must have been 15 trees that were planted. There are 21 trees originally. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: Then 3 + 2 = 5 more cars arrive. Now 3 cars are in the parking lot. There are originally 2 cars. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: After eating 32 + 42 = 74, they had 32 pieces left in total. Originally, Leah had 74 - 35 = 39 chocolates and her sister had 35. So in total they had 42. The answer is 39.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Then he had 20 - 12 = 8 after giving some to Denny. So he gave Denny 20 lollipops. Jason had 12 lollipops originally. The answer is 8.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Now he has 4 toys. So he got 5 + 4 = 9 more toys. Shawn started with 5 toys. He then got 2 * 2 = 4 toys each from his mom and dad. The answer is 9.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: So 5 computers were added. Now 4 * 5 = 20 computers are now in the server room. There were originally 9 + 20 = 29 computers. For each day from monday to thursday, 9 more computers were installed. The answer is 29.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: So he had 2 at the end of Tuesday, and 23 at the end of wednesday. He lost 35 - 2 = 33 on Tuesday, and lost 58 more on wednesday. Michael started with 58 - 23 = 35 golf balls. The answer is 33.

Q: Olivia has $23. She bought five bagels for 3 each. How much money does she have left?
A: Now she has 5 * 3 = 15 dollars left. So she spent 5 dollars. Olivia had 23 - 15 = 8 dollars. She bought 3 bagels for 23 dollars each. The answer is 8.

Table 12: Demonstrations for "no coherence" for reasoning tasks.

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?
A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 + 15 = 36. The answer is 36.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?
A: There are originally 3 cars. 2 more cars arrive. 3 * 2 = 6. The answer is 6.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?
A: Originally, Leah had 32 chocolates. Her sister had 42. So her sister had 42 - 32 = 10 more chocolates. After eating 35, they had 10 + 35 = 45. The answer is 45.

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A: Jason started with 20 lollipops. Then he had 12 after giving some to Denny. So he has 20 + 12 = 32. The answer is 32.

Q: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?
A: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 2 more toys. 5 + 2 = 7. The answer is 7.

Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?
A: There were originally 9 computers. 5 more computers were added. So 9 + 5 is 14. The answer is 14.

Q: Michael had 58 golf balls. On tuesday, he lost 23 golf balls. On wednesday, he lost 2 more. How many golf balls did he have at the end of wednesday?
A: Michael started with 58 golf balls. After losing 23 on tuesday, he had 58 - 23 = 35. The answer is 35.

Q: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?
A: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 3 dollars. So she has 23 - 3 dollars left. 23 - 3 is 20. The answer is 20.

Table 13: Demonstrations for "invalid reasoning and answer" for reasoning tasks.