NoisyICL: A Little Noise in Model Parameters Can Calibrate In-context Learning

趙羽風 [1]　坂井吉弘 [1]　井之上直也 [1,2]
[1]Japan Advanced Institute of Science and Technology, [2]RIKEN
{yfzhao, y.sakai, naoya-i}@jaist.ac.jp

Abstract

In-**C**ontext **L**earning (ICL), where language models learn tasks in a generative form from few-shot demonstrations without parameter update, is emerging while scaling up the language models. Nevertheless, the performance of ICL is still unsatisfactory. Some previous studies suggested that it is due to under-calibration and they fine-tuned language models for better ICL performance with enormous datasets and computing costs. In this paper, we propose NoisyICL, simply perturbing the model parameters by random noises to strive for a calibration. Our experiments on 2 models and 7 downstream task datasets show that NoisyICL helps perform ICL better. Our further analysis indicates that NoisyICL can enable the model to provide more fair predictions, with less unfaithful confidence. So, NoisyICL can be considered as an effective calibration.

1 Introduction

Scaling up language models is beneficial for many emergent abilities [1]. Among them, one of the most noticeable ones is **I**n-**C**ontext **L**earning (ICL), in which language models can learn tasks in a generative form from few-shot input-label demonstrations in natural language without explicit parameter updates. Therefore, ICL has been a highly promising application of language models [2].

Nevertheless, the performance of ICL is still below the pre-training and fine-tuning models [3]. Therefore, there has been some effort in fine-tuning or calibrating language models towards ICL tasks [4, 5, 6]. These works focus on remedying the difference between the pre-training knowledge and the ICL task, and produce significant improvements in the ICL performance, while the computation cost is quite high to fine-tune these enormous language models on the additional data.

We believe that adding noise to model parameters, which is beneficial in the pre-training and fine-tuning paradigm [7, 8], can be a bridge from the pre-training to ICL. In this paper, we propose NoisyICL, simply add noise to language model parameters, and then perform ICL on the modified models.

Our experiments on 2 models and 7 datasets show that an appropriate perturbation can significantly improve the performance of the ICL with low computational complexity, as shown in Fig. 1. Moreover, to verify whether NoisyICL can calibrate language models, we conduct further analysis and point out that: **1.** NoisyICL can neutralize bias among label tokens introduced by the pre-training and **2.** NoisyICL can relent the over- and under-confidence in the prediction, which is considered harmful to the model predictions [9, 10, 11].

Our contribution can be summarized as:

- We propose NoisyICL, simply add noise into the language models and then perform ICL (§2). Our experiment shows that NoisyICL can obtain a better ICL performance (§3.3).
- We show that adding noise can be an effective calibration for language models to reduce the pre-training bias and unfaithful confidence in ICL (§3.4).

2 NoisyICL

Here we introduce the basic form of ICL and our perturbation method named NoisyICL.

In-context Learning. Given a supervised dataset $\mathcal{D} = \{(x_i, y_i)\}, i = 1, \dots, n$, where x_i is an input, and $y_i \in U$ is the corresponding label in the label space U, for each query input x_q to be predicted by the language model, we sample a demos sequence $\{(x_{a_j}, y_{a_j})\}, j = 1, 2, \dots, k$, where the k is the number of demos, and

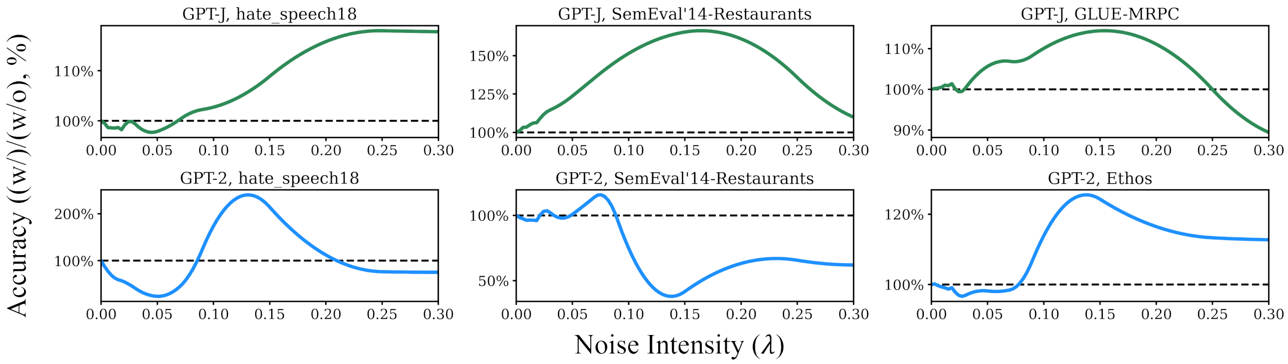

Figure 1 While noise is being added to the model, the ratio of ICL accuracy on the models with NoisyICL to the models without NoisyICL on downstream tasks will reach peaks. This indicates that an appropriate noise perturbation can improve the accuracy of ICL.

construct a prompt input in natural language form $s = f(x_{a_1}, y_{a_1}, x_{a_2}, y_{a_2}, \ldots, x_{a_k}, y_{a_k}, x_q)$ with a pattern f. Then, we input s into the language model $P_\theta(\cdot)$ with parameters θ and get an output token distribution $P_\theta(s)$. We choose the label token l with the maximum probability **among the label space** as the prediction $\hat{y_q}$, that is:

$$\hat{y_q} = \operatorname*{argmax}_{l \in U} P_\theta(l|s) \tag{1}$$

Notice that we only construct prompts to drive the model to predict labels generatively, without any parameter updates. Such a paradigm is In-context Learning.

NoisyICL. For every parameter matrix θ_i in the language model used for ICL, in this paper, we simply do an interpolation between the parameter matrix and a noise matrix sampled from $N(0, \sigma^2)$ with intensity λ, that is:

$$\theta_i' = (1 - \lambda)\theta_i + \lambda N(0, \sigma^2) \tag{2}$$

the λ and σ are model and task-wise hyperparameters. Then we perform the aforementioned ICL on the modified model. We call this NoisyICL.

3 Experiments and Results

We conduct comprehensive experiments to investigate the effectiveness of NoisyICL. First, we search for the most suitable noise intensity for each task and model (§3.2). Then, we confirm that NoisyICL can improve ICL performance (§3.3). Moreover, we demonstrate that NoisyICL is a kind of model calibration, that is, it can effectively alleviate the model's bias and unfaithful confidence (§3.4).

3.1 Experimental Settings

Here we introduce the datasets, models, and other details of our experiments.

Data. In the experiments, we use 7 downstream task datasets, whose details are shown in Appendix A. Unlike the common methods that only use the training sets for demos and testing sets for queries, we sample the demos and queries from all the labeled data. In detail, for each labeled data in the whole dataset, we treat it as the query and contrast a prompt with the demos sampled from the whole dataset (except the query).

Models. We use GPT-2 [12] and GPT-J [13]. The model checkpoints are loaded from huggingface[1)].

Hyperparameters. We fix the σ, the standard deviation of the normal distribution, to 0.02, which is the same as the initialization of both models. In advance, we search the value of λ, the intensity of noise, as described in §3.2.

Other details. We default to use 4 demos and a very simple template for each prompt as shown in Appendix B. We repeat each experiment 20 times.

3.2 The Intensity of Noise

First, we determine the most suitable noise intensity by a simple search method for each dataset and model. In detail, we use various intensities to test the performance and find the one with the best result as the candidate. Some examples are shown in Fig. 1, and the full results are in Appendix C. The selected intensities are shown in Table 1. These optimal intensities are concentrated in $(0, 0.2]$.

3.3 NoisyICL Can Improve Performance

Then, we test the accuracy and Macro-F1 on the 7 downstream task datasets with and without appropriate-noised NoisyICL. Our experimental results are shown in Table 1.

The results show that NoisyICL has an improvement up

1) huggingface.co/gpt2, and huggingface.co/EleutherAI/gpt-j-6b

Table 1 Accuracy and Macro-F1 results (%, $mean_{std}$, $k = 4$). A better result is in **bold**. λ: The intensity of noise, **Acc.**: Accuracy, **MF1**: Macro-F1; **w/o**: Not using NoisyICL, **w/**: Using NoisyICL; Datasets: **PS**: poem_sentiment, **HS**: hate_speech18, **SE'14R**: SemEval 2014-Task 4 Restaurants, **SE'14L**: SemEval 2014-Task 4 Laptops, **RTE**: GLUE-RTE, **MRPC**: GLUE-MRPC, **Ethos**: ethos.

Dataset			PS	HS	SE'14R	SE'14L	RTE	MRPC	Ethos	Mean
GPT-J	λ		0.2	0.2	0.1	0.1	0.1	0.2	0.04	—
	Acc.	w/o	**62.24**$_{0.26}$	72.51$_{0.46}$	34.52$_{0.47}$	34.00$_{0.37}$	**49.86**$_{0.87}$	43.08$_{0.45}$	56.31$_{0.75}$	50.36
		w/	52.13$_{6.53}$	**76.12**$_{9.04}$	**52.28**$_{7.13}$	**46.57**$_{2.29}$	49.59$_{0.55}$	**60.86**$_{3.52}$	**56.35**$_{1.30}$	**56.27**
	MF1	w/o	21.18$_{0.50}$	**27.11**$_{0.46}$	31.02$_{0.65}$	33.02$_{0.50}$	47.39$_{0.93}$	42.96$_{0.46}$	55.99$_{0.79}$	36.95
		w/	**22.83**$_{2.07}$	24.39$_{0.70}$	**46.73**$_{4.23}$	**46.34**$_{2.13}$	**48.70**$_{1.03}$	**47.72**$_{2.86}$	**56.00**$_{1.33}$	**41.81**
GPT-2	λ		0.02	0.1	0.08	0.006	0.1	0.08	0.1	—
	Acc.	w/o	52.80$_{0.67}$	37.62$_{0.28}$	41.60$_{0.46}$	40.25$_{0.45}$	50.30$_{0.55}$	**67.30**$_{0.08}$	44.49$_{0.56}$	47.76
		w/	**52.82**$_{1.20}$	**65.40**$_{4.20}$	**47.16**$_{2.07}$	**41.00**$_{0.71}$	**50.36**$_{0.62}$	58.10$_{1.59}$	**50.67**$_{2.03}$	**52.22**
	MF1	w/o	**24.87**$_{0.75}$	17.70$_{0.16}$	**36.55**$_{0.47}$	38.67$_{0.47}$	**49.77**$_{0.57}$	41.01$_{0.16}$	34.80$_{0.65}$	34.77
		w/	24.50$_{1.51}$	**24.12**$_{0.63}$	33.71$_{0.30}$	**39.47**$_{0.77}$	34.75$_{0.62}$	**50.21**$_{0.49}$	**49.53**$_{1.40}$	**36.61**

to 74% and average around 11% to the ICL performance. We infer that the pre-training datasets and objectives are not consistent with the ICL tasks [14], that is, the language models are overfitted on pre-training. And NoisyICL, which adds noise into models, can bridge such a gap.

However, such gains vary depending on the dataset. In some combinations of datasets and models, competitive results cannot be obtained. We speculate that it is due to the difficulty of these datasets, where the models cannot predict these tasks intrinsically, while NoisyICL doesn't provide new knowledge for these tasks.

3.4 NoisyICL Is A Calibration

Some previous studies have proposed calibration on large language models for better ICL performance [4, 5, 6, 15]. These calibrations are mainly aimed at a **1. fairer output distribution** [5, 15], that is, when no valid query is given, the labels should be assigned with the same likelihood. However, in original language models, the output is unfair due to the pre-training bias. Moreover, some researchers pointed out that **2. unfaithful predictions are harmful** [10, 11], and making the model output with more faithful confidence is also a form of calibration [10, 16]. Some scholars also try some demonstration selection methods to obtain outputs with more faithful confidence [9].

In this section, we find that the NoisyICL can also solve both calibrations above. In detail, the model with NoisyICL can not only produce outputs with less bias but also

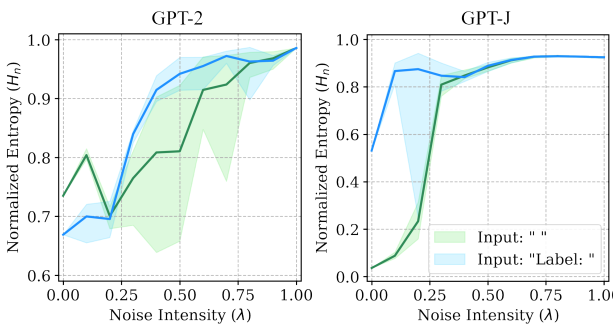

Figure 2 The correlation between the normalized entropy H_n and the noise intensity λ with no query. When the noise gets stronger, the H_n becomes higher, which indicates a fairer output.

with suitable confidence. Therefore, we consider NoisyICL as a kind of calibration with a relatively small time and space cost.

1. NoisyICL alleviates pre-trained bias. We calculate the normalized entropy H_n of the model output distribution when no valid query is given. In detail, for language model P_θ with a vocabulary size $|V|$, we construct a semantic-less input x_0 (such as a space, or "Label: "), and calculate the H_n as:

$$H_n = \frac{\sum_{i=1}^{|V|} P_\theta(i|x_0) \ln P_\theta(i|x_0)}{\ln |V|} \tag{3}$$

The H_n is higher on a fairer output and $H_n = 1$ on a random output.

We test H_n for both models with 2 different x_0 and various noise intensities. The results are shown in Fig. 2. While the noise is getting stronger, the normalized entropy is getting larger, which means the model is giving a fairer

Table 2 The ECE_1 results (↓, %, $mean_{std}$, $k = 4$).

Dataset	GPT-J w/o	GPT-J **w/**	GPT-2 w/o	GPT-2 **w/**
PS	$15.22_{0.47}$	$\mathbf{12.39}_{2.09}$	$7.25_{0.68}$	$\mathbf{6.22}_{0.90}$
HS	$14.86_{1.89}$	$\mathbf{8.71}_{1.95}$	$37.48_{0.23}$	$\mathbf{11.92}_{4.96}$
SE'14R	$31.12_{1.08}$	$\mathbf{15.49}_{9.81}$	$17.32_{0.82}$	$\mathbf{15.98}_{1.15}$
SE'14L	$35.74_{1.47}$	$\mathbf{14.58}_{9.33}$	$14.03_{0.47}$	$\mathbf{13.88}_{0.66}$
RTE	$\mathbf{29.49}_{1.31}$	$32.24_{2.02}$	$\mathbf{31.83}_{0.61}$	$44.83_{1.12}$
MRPC	$29.08_{0.67}$	$\mathbf{17.60}_{9.00}$	$\mathbf{20.56}_{0.23}$	$21.22_{0.67}$
Ethos	$12.15_{0.99}$	$\mathbf{11.95}_{1.05}$	$45.61_{0.39}$	$\mathbf{28.21}_{0.86}$
Mean	23.95	**16.14**	24.87	**20.32**

output.

2. NoisyICL promotes faithful confidence. The **E**xpected **C**alibration **E**rror (ECE_p) [17] is a widely-used indicator for faithfulness of model confidence:

$$ECE_p = \mathbf{E}(|\max(\hat{z}) - \mathbf{E}(1_{y=\underset{i}{\operatorname{argmax}}\,\hat{z}_i})|^p)^{\frac{1}{p}} \tag{4}$$

where the $\hat{z}$ is the predicted probability vector by a classification model, and the final prediction ($\underset{i}{\operatorname{argmax}}\,\hat{z}_i$) can be obtained with a confidence ($\max \hat{z}$), and the true label is y.

Let the $p = 1$, we use the ECE_1 to investigate the over- and under-confidence of the ICL output. A lower ECE_1 means more faithful confidence, and better calibration, that is, the confidence becomes a prediction of accuracy [18]. We test both models with and without the appropriate-noised NoisyICL for ECE_1 on the 7 datasets, the results are shown in Table 2.

In most situations, the ECE_1 is lower with NoisyICL than the unperturbed one, meaning the confidence is more faithful with NoisyICL. This suggests that NoisyICL can make the model output with more faithful confidence, that is, less over-confidence in wrong predictions, and less under-confidence in correct predictions.

Such results suggest that NoisyICL can be considered as a kind of calibration.

3.5 NoisyICL Furtherance Correct ICL

Moreover, we find that in some cases, unperturbed ICL can't benefit correctly from scaling the number of demos, while, the NoisyICL can help the model correct this issue, as shown in Fig. 3. These unperturbed models exhibit an overfitting-like phenomenon and also low accuracies, while NoisyICL can relieve it.

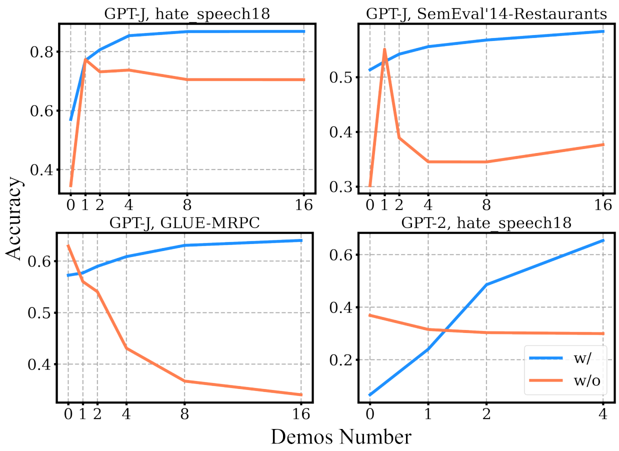

Figure 3 The impact of demos quantity on accuracy. NoisyICL can make the model learn from the demos correctly.

We speculate the reason is the mismatch between the pre-training knowledge and ICL inputs. This leads to a decrease in the model's in-context task learning [19] ability, while NoisyICL reduces such a gap between pre-training data and ICL style data, which makes models extract information from ICL inputs better.

4 Conclusion

In this paper, we propose NoisyICL, simply adds random noise to the parameters of language models to build a bridge between the pre-training knowledge and the ICL. We show that NoisyICL can not only improve the ICL performance but also calibrate the model for fairer outputs and more faithful confidence.

Limitations. Unlike the fine-tuning on additional ICL-style datasets [4, 5, 6], NoisyICL does not provide new knowledge for the model, so the calibrated model can not discover tasks that are not potentially included in the pre-training data [20]. Meanwhile, a simple search for the noise intensity is not efficient and satisfactory.

Future Works. Besides fixing the limits, future works can focus on where and how the noise should be introduced. In Transformer-based models, different layers have different abilities [21, 22]. So, treating these layers differently may be an effective improvement of NoisyICL. Noise sampling methods also should be discussed.

Moreover, adding noise to model parameters can be a rollback of pre-training [23], so, the search for λ is the search for the best pre-training checkpoints. With these checkpoints, we can determine [24, 25] which data is disadvantageous to ICL, to better reveal the essence of ICL.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 19K20332. The authors would like to thank Mr. Daichi Haraguchi at JAIST and MSc. Tongyuan Wei at the University of Cambridge, for their constructive criticism.

References

[1] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. **arXiv preprint arXiv:2206.07682**, 2022.

[2] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. **arXiv preprint arXiv:2301.00234**, 2022.

[3] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. **arXiv preprint arXiv:2305.16938**, 2023.

[4] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. In **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 2791–2809, 2022.

[5] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In **International Conference on Machine Learning**, pp. 12697–12706. PMLR, 2021.

[6] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In **International Conference on Learning Representations**, 2021.

[7] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Noisytune: A little noise can help you finetune pretrained language models better. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 680–685, 2022.

[8] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. In **International Conference on Learning Representations**, 2020.

[9] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 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)**, pp. 8086–8098, 2022.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In **International conference on machine learning**, pp. 1321–1330. PMLR, 2017.

[11] Julia Grabinski, Paul Gavrikov, Janis Keuper, and Margret Keuper. Robust models are less over-confident. **Advances in Neural Information Processing Systems**, Vol. 35, pp. 39059–39075, 2022.

[12] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[13] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[14] Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. On the effect of pretraining corpora on in-context learning by a large-scale language model. **arXiv preprint arXiv:2204.13509**, 2022.

[15] Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. Symbol tuning improves in-context learning in language models. **arXiv preprint arXiv:2305.08298**, 2023.

[16] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. **arXiv preprint arXiv:2305.14975**, 2023.

[17] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In **Proceedings of the AAAI conference on artificial intelligence**, Vol. 29, 2015.

[18] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. **Advances in Neural Information Processing Systems**, Vol. 32, , 2019.

[19] Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning" learns" in-context: Disentangling task recognition and task learning. **arXiv preprint arXiv:2305.09731**, 2023.

[20] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Pre-training to learn in context. **arXiv preprint arXiv:2305.09137**, 2023.

[21] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 9840–9855, Singapore, December 2023. Association for Computational Linguistics.

[22] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In **ACL 2019-57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[23] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In **The Eleventh International Conference on Learning Representations**, 2022.

[24] Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 12660–12673, 2023.

[25] Xiaochuang Han and Yulia Tsvetkov. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. **arXiv preprint arXiv:2205.12600**, 2022.

[26] Emily Sheng and David Uthus. Investigating societal biases in a poetry composition system, 2020.

[27] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate Speech Dataset from a White Supremacy Forum. In **Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)**, pp. 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[28] Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. Ethos: an online hate speech detection dataset, 2020.

[29] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In Preslav Nakov and Torsten Zesch, editors, **Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)**, pp. 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics.

[30] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In **International Conference on Learning Representations**, 2018.

A Datasets

The datasets used in this paper are shown in the Table 3

Table 3 Datasets used in this paper.

Dataset	Data#	Label#
single-sentence classification:		
poem_sentiment[26]	1101	4
hate_speech18[27]	10944	4
Ethos*[28]	980	2
aspect-based sentiment classification:		
SemEval 2014-Task 4 Restaurants[29]	4722	3
SemEval 2014-Task 4 Laptops[29]	2951	3
double-sentence classification:		
GLUE-RTE[30]	2767	2
GLUE-MRPC[30]	4076	2

*To construct inputs of appropriate length, we remove data points with lengths exceeding 500 from the Ethos, and the number of the remaining data is 980.

B Prompt Patterns

In this paper, we use a minimum prompt template. For each task, we design various templates as shown below.

For single-sentence classification datasets (x, y), we use:

```
Input: <x>, Label: <y> \n
...
Input: <x>, Label:
```

For aspect-based sentiment classification datasets $((x, a), y)$, we use:

```
Input: <x>, Aspect: <a>, Label: <y> \n
...
Input: <x>, Aspect: <a>, Label:
```

For double-sentence classification datasets $((x_1, x_2), y)$. we use:

```
Input: <x1>, Text 2: <x2>, Label: <y> \n
...
Input: <x1>, Text 2: <x2>, Label:
```

C Full Results: λ - Accuracy

The rest of the results in 3.2 and Fig. 1 are shown in Fig. 4.

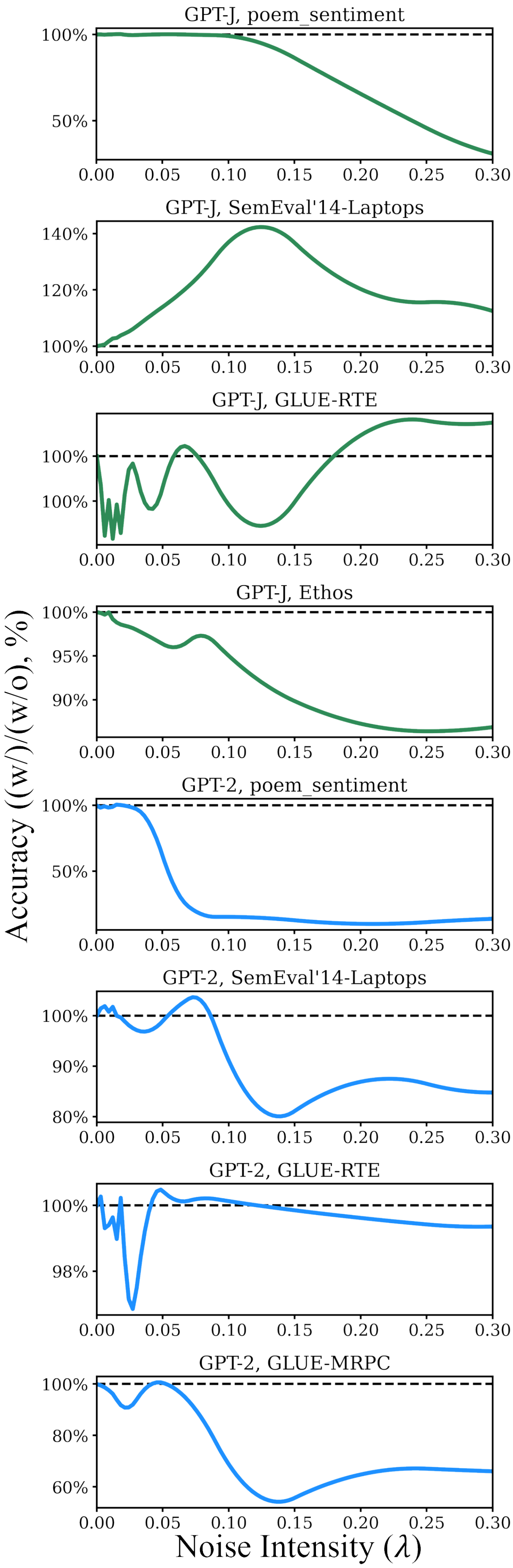

Figure 4 The rest of the results in 3.2 and Fig. 1.