Assessing the Macro and Micro Effects of Random Seeds on Fine-Tuning Large Language Models

Anonymous ACL submission

Abstract

The impact of random seeds in fine-tuning large language models (LLMs) has been largely overlooked despite its potential influence on model performance. In this study, we systematically evaluate the effects of random seeds on LLMs using the GLUE and SuperGLUE benchmarks. We analyze the macro impact through traditional metrics like accuracy and F1, calculating their mean and variance to quantify performance fluctuations. To capture the micro effects, we introduce a novel metric, *consistency*, measuring the stability of individual predictions across runs. Our experiments reveal significant variance at both macro and micro levels, underscoring the need for careful consideration of random seeds in fine-tuning and evaluation.

1 Introduction

The impact of random seeds in neural network training has long been recognized across various domains, such as general machine learning classification and regression tasks (Ganesh et al., 2023; Madhyastha and Jain, 2019), computer vision (Picard, 2021; Åkesson et al., 2024), natural language processing (NLP)(Bethard, 2022; Lucic et al., 2022).

In the field of NLP, large language models (LLMs) have achieved state-of-the-art results on benchmarks like GLUE and SuperGLUE, which are now standard for evaluating language understanding and reasoning. However, pretrained transformers such as BERT (Devlin et al., 2019) and RoBERTa (Liu, 2019) are highly sensitive to random seeds (Risch and Krestel, 2020; Dodge et al., 2020; Mosbach et al., 2020), often leading to significant performance variation that complicates experimental interpretation and benchmarking. While other sources of randomness, such as prompt formatting (He et al., 2024), in-context example selection (Gupta et al., 2023), and how learnable weights are initialized (Hayou et al., 2024), have also been

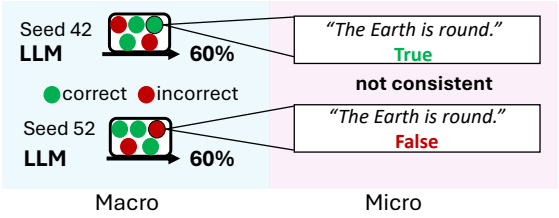

Figure 1: Macro and micro performance. A pretrained LLM is fine-tuned with random seed 42 and 52. The accuracy for both models is 60%, but the overlapping of individual predictions is 20%.

explored, seed variation remains a fundamental and underaddressed issue. A recent analysis of 85 papers from the ACL Anthology (Bethard, 2022) revealed risky practices in the use of random seeds: over 50% of the papers exhibited potential misuse, with 24 using a single fixed random seed. *This highlights that random seeds sensitivity in LLM fine-tuning remains insufficiently understood, motivating the need for more systematic investigation.*

Existing studies examining the impact of random seeds (Ganesh et al., 2023; Madhyastha and Jain, 2019; Picard, 2021) typically evaluate performance variations by measuring the variance of standard metrics, such as accuracy score for classification tasks, or MAE (mean absolute error) for regression tasks, across multiple seeds. These evaluations focus on the **macro** agreement of model performance across the entire test set, offering insights into overall variability. However, *they overlook the **micro** impact of how individual test points are influenced by seed-induced variability*. As shown in Figure 1, model performance is robust to random seed 42 and 52 at the macro level (both achieve 60% accuracy) but lacks consistency at the micro level (only 20% overlapping predictions). This micro inconsistency can have severe consequences in real-world applications, especially in fields where model predictions are highly sensitive to individual test points,

such as medical diagnosis and autonomous driving. Understanding this micro effect is crucial for assessing model robustness at the level of individual predictions, ensuring that specific test samples are not inconsistently predicted due to seed-induced variability. Additionally, it helps pinpoint specific areas where models may exhibit significant instability, such as consistently misclassifying certain types of data points or showing highly variable predictions for similar inputs. Recognizing these areas of instability can guide targeted improvements in both model design and evaluation practices, ensuring that assessments account for seed-induced variability in performance.

Major contributions: To address these gaps, in this work, (1) we analyze the impact of random seeds on pretrained LLMs using the GLUE and SuperGLUE benchmarks, covering both macro and micro variability; (2) We introduce a novel **consistency** metric to assess prediction stability on individual test points, capturing the micro effects of random seeds; (3) Our extensive experiments reveal significant variability in both standard and consistency metrics, underscoring the need to consider seed-induced variability in fine-tuning and evaluation, and incorporate random seeds sensitivity into benchmarking and reporting for more reliable and reproducible results.

2 Macro Metric: Variance

To measure the macro impact of random seeds on LLM performance, we calculate the variance of a standard evaluation metric across multiple seeds. Let $[\zeta_1, \cdots, \zeta_S]$ represent the values of model performance for an LLM fine-tuned with S random seeds, the *variance* is calculated by:

$$\text{VAR}(\zeta) = \sqrt{\frac{1}{S}\sum_{i=1}^{S}(\zeta_i - \bar{\zeta})^2} \tag{1}$$

where $\bar{\zeta} = \frac{1}{S}\sum_{i=1}^{S}\zeta_i$. ζ can be any standard metrics, such as accuracy score for classification tasks or MAE for regression tasks. A smaller VAR indicates less variation in macro performance.

3 Micro Metric: Consistency

'Consistency' can have varying definitions across domains. Building on prior work, Wang et al. (2020) formally defined the *consistency* of a deep learning model as its ability to produce consistent predictions for the same input when periodically retrained with streaming data in deployment settings. Extending this idea, we define the *consistency* of an LLM as its ability to generate consistent predictions for the same input across models fine-tuned with different hyperparameter settings, with *correct-consistency* further specifying its ability to make consistent correct predictions in this context.

More specifically, consider two LLMs A and B, given a dataset $\mathcal{D} = d_1, \cdots, d_N$ of N data points, y_i^A and y_i^B are the prediction of A,B for a data point d_i with ground truth r_i. The *consistency* of A and B can be calculated as follows:

$$\text{CON:}\frac{1}{N}\sum_{t=1}^{N}\pi_{A,B}(t) \tag{2}$$

where $\pi_{A,B}(\cdot)$ is the scoring function that quantifies the alignment between predictions y_t^A and y_t^B, with higher values indicating smaller variations in micro-level predictions; it can be either binary (e.g., 1 for match, 0 for mismatch) or probabilistic based on different NLP tasks. And the *correct-consistency* is calculated by:

$$\text{CCON:}\frac{1}{N}\sum_{t=1}^{N}\pi_{A,B,r}(t) \tag{3}$$

where $\pi_{A,B,r}(\cdot)$ is the scoring function that quantifies the alignment between predictions y_t^A, y_t^B, and ground truth r_t.

In this paper, we focus solely on classification tasks because the GLUE and SuperGLUE benchmarks primarily consist of classification problems. The scoring function $\pi(\cdot)$ is defined as an indicator function that equals 1 if $y_t^A = y_t^B (= r_t)$, otherwise 0. We summarize the standard metric ζ and the possible corresponding scoring functions π for various NLP tasks in Table 2 in the Appendix A.1. We hope this summary offers useful context for interpreting our results and supports future extensions of our evaluation to a broader range of NLP tasks.

While consistency metrics can generally be used for quantifying the agreement of individual predictions from any two LLMs with different architectures, hyperparameters, or training settings, in our study, they are specifically used to serve as metrics to evaluate the micro impact of random seeds on the same pretrained LLM.

4 Experimental Setup

4.1 Benchmarks and pretrained models

In this study, we conduct experiments on a range of NLP tasks including CoLA, SST2, MRPC, QQP,

RoBERTa-large							Llama3.2-3B						
GLUE	**MRPC**	**QNLI**	**QQP**	**SST2**	**RTEG**	**CoLA**	GLUE	**MRPC**	**QNLI**	**QQP**	**SST2**	**RTEG**	**CoLA**
ζ	90.34	94.00	92.00	95.59	85.02	65.61	ζ	84.02	94.20	89.37	96.78	85.92	61.33
VAR	0.89	0.38	0.06	0.55	1.48	1.32	VAR	0.56	0.14	0.09	0.27	3.34	0.88
CON	92.89	95.64	95.57	96.83	91.09	93.95	CON	90.34	96.85	96.63	98.32	88.80	94.86
CCON	86.79	91.80	89.79	94.01	80.55	82.76	CCON	79.19	92.63	87.69	95.95	80.32	81.36
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**	SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**
ζ	83.05	94.64	72.89	76.46	68.57	73.00	ζ	72.49	73.92	68.66	80.14	68.71	84.20
VAR	7.35	3.95	16.67	13.30	2.77	16.83	VAR	1.62	2.03	2.77	0.57	1.58	5.17
CON	86.79	91.50	70.05	74.63	80.36	64.88	CON	81.10	89.82	71.69	86.73	85.86	86.8
CCON	76.37	89.84	64.58	63.77	58.75	55.44	CCON	63.09	70.35	54.51	73.50	61.64	77.6

Table 1: Macro and micro impact of ten random seeds. ζ is the average of 10 values, which is MCC for CoLA and accuracy for the other tasks. VAR is the variance of ζ calculated using Equation 1. CON and CCON are the average of 45 consistency values calculated using Equation 2 and 3. ζ, CON, and CCON are expressed as percentages.

QNLI, and RTE from GLUE (Wang et al., 2018) benchmark; RTE, CB, WiC, BoolQ, MultiRC, and COPA from SuperGLUE (Wang et al., 2019) benchmark. STSB, WSC, and MNLI tasks are omitted from our experiment. We specify the reason in Section A.4. All tasks use accuracy as the standard evaluation metric ζ, except for CoLA, which uses Matthews Correlation Coefficient (MCC). In our paper, we use RTEG to denote RTE task from GLUE and RTES for SuperGLUE.

To examine the effects on various scales of LLMs, we experiment with `RoBERTa-large` (~350M trainable parameters), as well as a larger LLM `Llama3.2-3B` (~3.21B trainable parameters) using LoRA (Hu et al., 2022) fine-tuning, enabling us to assess whether our findings generalize across model scales.

4.2 Settings

Our experiment is implemented using Hugging Face Transformers (v4.30.0) and PyTorch (v2.0), conducted on NVIDIA two A100 GPUs with 80GB of memory each. Based on the empirical findings in (Wang et al., 2023; Dodge et al., 2020; Mosbach et al., 2020), 5–10 seeds are sufficient to estimate variance of LLMs in NLP tasks. We perform full fine-tuning for each task with ten randomly chosen seeds: 42, 52, 62, 72, 82, 92, 102, 112, 122, 132 (i.e., $S = 10$). We calculate CON and CCON on each unique pair of seeds and report the average of 45 values as the final consistency score. Our fine-tuning process is based on the PyTorch script `run_glue.py`, and the best previously reported settings were applied unless otherwise specified.

To ensure proper experimental setup and reproducibility, we refer the configurations in (Liu, 2019) and replicate state-of-the-art (SOTA) scores reported using `RoBERTa-large` with full fine-tuning. A comparison of our implementation with the reference SOTA scores and detailed data and learning settings are provided in Appendix Table 3, Table 4, and Table 5. Since we could not find reference configurations or performance reports for `Llama3.2-3B` on the two benchmarks, we use the SOTA scores (Table 3) as a reference and conduct experiments using our own settings, detailed in Table 6.

5 Results and Discussion

5.1 Macro impact

Table 1 presents the averaged accuracy (ζ) and variance (VAR) for all tasks across ten random seeds using two LLMs. Significant VAR is observed in many tasks using `RoBERTa-large`, such as RTES (16.67), COPA (16.83), and MultiRC (13.30), reflecting sensitivity to random seed selection. High variability at the macro level undermines the reliability of single-seed evaluations, emphasizing the need for robust evaluation methods and stability-enhancing techniques. In contrast, tasks like QQP (0.06), QNLI (0.38), SST2 (0.55), and MRPC (0.89) show much greater stability, likely due to their inherent properties such as larger datasets or simpler decision boundaries. This also helps explain why SuperGLUE tasks generally show higher VAR than GLUE tasks.

Compared to `RoBERTa-large`, `Llama3.2-3B` with LoRA fine-tuning exhibits significantly lower VAR across most tasks. This is likely because only a small subset of parameters (~2.3 million) is updated during LoRA fine-tuning, which constrains the variance introduced by random seeds and results in greater stability and robustness to seed-induced fluctuations.

5.2 Micro impact

Table 1 reports consistency (CON) and correct-consistency (CCON) for all tasks over ten random seeds. For `RoBERTa-large`, high CON values in tasks like SST2 (96.83), QNLI (95.64), and QQP (95.57) indicate stable predictions, while lower values in tasks like COPA (64.88) and RTES (70.05) highlight their sensitivity to random seeds, potentially due to smaller training sizes or task complexity. Tasks with large CON–CCON gaps (e.g., WiC with a 21.61 difference and CoLA with 11.19) suggest that *consistent predictions are not always accurate, emphasizing the need to evaluate both stability and correctness*. Furthermore, tasks like WiC, which show low VAR alongside low CON and CCON, demonstrate that *similar macro accuracy can mask underlying instability*, reinforcing the importance of micro-level evaluation beyond traditional metrics. Results from `Llama3.2-3B` show consistent trends, albeit with task-specific variations.

Unlike the macro impact where `Llama3.2-3B` with LoRA fine-tuning shows significantly lower VAR than `RoBERTa-large`, no such trend is observed for CON and CCON. This suggests that *reducing the number of trainable parameters helps mitigate macro variance but has limited effect on micro consistency.*

5.3 Discussion

Will increasing training data size improve variance and consistency in general?

To answer the question, we present Pearson correlation analysis in Figure 2, showing the relationship between training size, variance, and consistency, with tasks sorted by increasing dataset size. It reveals a weak negative correlation (-0.3918) between training size and VAR, indicating that smaller datasets tend to increase macro variance. However, the effect is not pronounced or consistent across all tasks, as MultiRC exhibits high VAR despite a relatively large dataset. A weak or moderate positive correlation is observed between training size and both CON (0.4257) and CCON (0.4259), suggesting that larger datasets generally improve consistency and prediction stability across random seeds, but with no guarantee. *Increasing training size can reduce macro and micro variability in random seeds, but its effectiveness depends on factors like data quality, task complexity, and label noise (Shahinfar et al., 2020; Althnian et al., 2021; Bailly et al., 2022).*

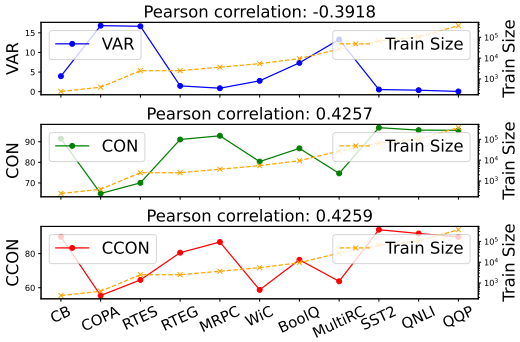

Figure 2: Correlation between training size (log scale), VAR, CON, and CCON while using `RoBERTa-large`. Tasks are arranged in ascending order of training size, with exact sizes detailed in Appendix 4.

Outlook: mitigating seed-induced variability.

Our analysis reveals substantial seed-induced variability in model performance at both the macro and micro levels. Prior studies have proposed several strategies to mitigate macro variability, including model ensembling (Risch and Krestel, 2020; Wang et al., 2023, 2020), stability-aware training (Dodge et al., 2020), and more robust evaluation protocols (Mosbach et al., 2020).

While our findings suggest that reducing the number of trainable parameters can mitigate macro variance but has limited impact on micro consistency. Increasing the training size can reduce both macro and micro variability in random seeds, but it's not a guaranteed solution, especially for complex tasks. An alternative direction is to identify robust data points (i.e., examples consistently predicted correctly across seeds) through micro-level analysis. An additional discussion can be found in A.6. These examples can guide improvements in data collection, preprocessing, prompt design, or synthetic data generation to enhance model robustness. We leave this as a promising direction for future work.

6 Conclusion

In conclusion, this work highlights the significant impact of random seeds on pretrained LLMs, revealing variability at both macro and micro levels. By introducing a novel consistency metric, we emphasize the importance of considering seed-induced variability in individual predictions in model evaluation. Our findings stress the need for incorporating random seed sensitivity into benchmarking for more reliable and reproducible results.

7 Limitations

Our work focuses on classification tasks which are the main consists of GLUE and SuperGLUE benchmarks. While this provides a solid foundation, it may limit the generalizability of our findings to other NLP task types. Incorporating a broader set of benchmark datasets would allow for a more comprehensive evaluation using our proposed macro and micro metrics across diverse task categories (as summarized in Table 2). Experimenting with greater task diversity would better capture variability in model behavior, ultimately enhancing the robustness and applicability of our analysis. We leave this broader exploration as future work.

References

Julius Åkesson, Johannes Töger, and Einar Heiberg. 2024. Random effects during training: Implications for deep learning-based medical image segmentation. *Computers in Biology and Medicine*, 180:108944.

Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. 2021. Impact of dataset size on classification performance: an empirical evaluation in the medical domain. *Applied Sciences*, 11(2):796.

Alexandre Bailly, Corentin Blanc, Élie Francis, Thierry Guillotin, Fadi Jamal, Béchara Wakim, and Pascal Roy. 2022. Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 213:106504.

Steven Bethard. 2022. We need to talk about random seeds. *arXiv preprint arXiv:2210.13393*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Noah Smith, and Hannaneh Hajishirzi. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Prakhar Ganesh, Hongyan Chang, Martin Strobel, and Reza Shokri. 2023. On the impact of machine learning randomness on group fairness. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1789–1800.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. *arXiv preprint arXiv:2305.14907*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. The impact of initialization on lora finetuning dynamics. *Advances in Neural Information Processing Systems*, 37:117015–117040.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Ana Lucic, Maurits Bleeker, Samarth Bhargav, Jessica Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, and Robert Stojnic. 2022. Towards reproducible machine learning research in natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 7–11, Dublin, Ireland. Association for Computational Linguistics.

Pranava Madhyastha and Rishabh Jain. 2019. On model stability as a function of random seed. *arXiv preprint arXiv:1909.10447*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

David Picard. 2021. Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Saleh Shahinfar, Paul Meek, and Greg Falzon. 2020. "how many images do i need?" understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics*, 57:101085.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Lijing Wang, Dipanjan Ghosh, Maria Gonzalez Diaz, Ahmed Farahat, Mahbubul Alam, Chetan Gupta, Jiangzhuo Chen, and Madhav Marathe. 2020. Wisdom of the ensemble: Improving consistency of deep learning models. *Advances in Neural Information Processing Systems*, 33:19750–19761.

Lijing Wang, Yingya Li, Timothy Miller, Steven Bethard, and Guergana Savova. 2023. Two-stage fine-tuning for improved bias and variance for large pretrained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15746–15761, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Macro and Micro Metrics for Various NLP Tasks

Given two LLMs A and B, a dataset $\mathcal{D} = d_1, \cdots, d_N$ of N data points, y_i^A and y_i^B are the prediction of A,B for a data point d_i with ground truth r_i. Table 2 summarizes the performance metric ζ and the corresponding scoring functions $\pi(\cdot)$ used in Equations 2 and 3 for various NLP tasks.

A.2 Data Description

Table 4 presents the statistics of the dataset used in our experiments. Each dataset consists of predefined train, dev and test data in CSV format. We use the train and dev sets for training and evaluation. Since the test set does not include gold-standard labels, the dev set also serves as the test set. For datasets where each instance may have multiple correct answers, such as MultiRC, we split the data at the question-answer pair level rather than the passage level. This ensures a more balanced distribution of instances across the train and dev sets. In the COPA dataset, each instance is originally described by six fields—premise, choice1, choice2, question, idx, and label. To adapt these instances into a multiple-choice format, we construct two candidate sequences for every sample. Specifically, for each candidate, we concatenate the premise with the question and the corresponding choice using a dedicated separation token (e.g., "[SEP]") to clearly delineate the different textual components. We then maintain the original label field, converting it from 1/2 to 0/1 to match the 0-based index convention in multiple-choice classification. This preprocessing ensures consistency with other classification tasks and allows the model to effectively learn the relationships between the premise and possible choices.

A.3 Hyperparameter Settings

Table 5 and Table 6 provides the detailed hyperparameter configurations. Unless stated otherwise, we adopt the default hyperparameter values from the Hugging Face framework.

A.4 Replicated SOTA Scores

To ensure the reproducibility of our experiments in SuperGLUE and GLUE tasks, we adhered to the specified settings and reproduced the state-of-the-art (SOTA) accuracy scores reported in: `https://github.com/facebookresearch/fairseq/tree/main/examples/roberta`. Although differences in the fine-tuning script and missing settings from the original authors prevented us from reproducing the exact SOTA scores, our replicated accuracy scores for the GLUE and SuperGLUE tasks, presented in Table 3, are directly comparable and align with those shown in Table 1 of the main paper and Table 7 in Section A.5.

We were unable to replicate the representation (special token extractions) and model settings (unpublished pretrained model) for the WSC and MNLI tasks, so they are omitted from the experiment. STSB is a regression task, thus is omitted.

A.5 Additional Results

Table 7 and Table 8 present model performance across various metrics, including precision (P), recall (R), F1 score, accuracy, CON, and CCON, with average values and standard deviations (VAR). In Section 5 of the main paper, significant variance in macro-level performance across many tasks highlights sensitivity to random seed selection. Similar patterns in the VAR values for P, R, and F1 further confirm the robustness of our findings across various standard metrics.

A.6 Additional Discussion

Is there a universally superior random seed?

To investigate whether a specific random seed consistently leads to better results across different models or tasks, in Figure 3 we present a heatmap of normalized accuracy for each task across ten random seeds using `RoBERTa-large`. There is no significant difference in color distribution between

each row, indicating that *no discernible pattern or evidence supporting the existence of a universally superior random seed*.

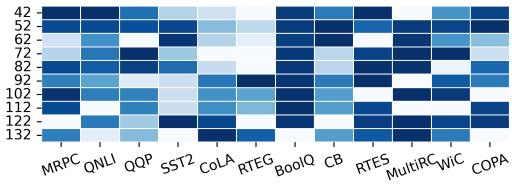

Figure 3: A heatmap of normalized ζ values across tasks and ten random seeds, with a darker color representing a better accuracy.

Will increasing training data size improve variance and consistency in general?

Figure 4 presents the correlation analysis of VAR, CON, and CCON using `Llama3.2-3B`, revealing patterns consistent with the findings discussed in Section 5.3.

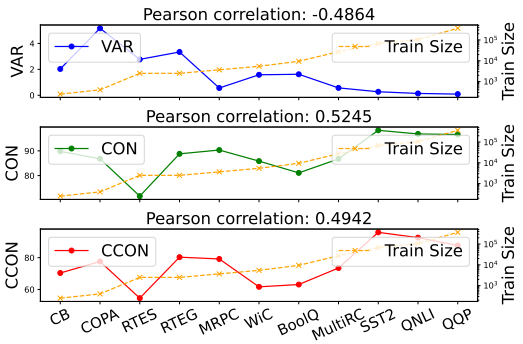

Figure 4: Correlation between training size (log scale), VAR, CON, and CCON while using `Llama3.2-3B`. Tasks are arranged in ascending order of training size, with exact sizes detailed in Appendix 4.

Outlook: mitigating seed-induced variability.

As discussed in Section 5.3, identifying robust data points through micro-level analysis can support more consistent model behavior by informing data collection, preprocessing, prompt design, or synthetic data generation. Based on the metric definitions in Table 2, robust data points are those that optimize the scoring function π at the individual level. For instance, for classification tasks, these are examples consistently predicted correctly across random seeds. For regression, they are points with lower π values (e.g., MAE). For generation tasks, robust points yield higher π scores (e.g., BLEU), indicating greater semantic or lexical consistency. This rationale extends to other task types accordingly.

NLP Task Type	Standard Metric ζ	Scoring Function $\pi(y_i^A, y_i^B)$	Scoring Function $\pi(y_i^A, y_i^B, r_i)$
Classification	Accuracy, F1	$\mathbf{1}(y_i^A = y_i^B)$	$\mathbf{1}(y_i^A = y_i^B = r_i)$
Regression	Pearson/Spearman, MSE, MAE	$\zeta(y^A, y^B)$	$\frac{1}{2}(\zeta(y_i^A, r_i) + \zeta(y_i^B, r_i))$
Text Generation	BLEU, ROUGE, BERTScore	$\zeta(y_i^A, y_i^B)$	$\frac{1}{2}(\zeta(y_i^A, r_i) + \zeta(y_i^B, r_i))$
Sequence Labeling	Token-/Span-level F1	$\frac{1}{T}\sum_t \mathbf{1}(y_{i,t}^A = y_{i,t}^B)$	$\frac{1}{T}\sum_t \mathbf{1}(y_{i,t}^A = y_{i,t}^B = r_{i,t})$
Structured Prediction	UAS, LAS, Exact Match	$\zeta(y_i^A, y_i^B)$	$\frac{1}{2}(\zeta(y_i^A, r_i) + \zeta(y_i^B, r_i))$
Question Answering	Exact Match, F1 (token overlap)	$\zeta(y_i^A, y_i^B)$	$\frac{1}{2}(\zeta(y_i^A, r_i) + \zeta(y_i^B, r_i))$
Retrieval / Ranking	NDCG, MRR, MAP	$\zeta(y_i^A, y_i^B)$	$\frac{1}{2}(\zeta(y_i^A, r_i) + \zeta(y_i^B, r_i))$

Table 2: Common evaluation metrics and scoring functions for various NLP task types. $\mathbf{1}(\cdot)$ denotes the indicator function that equals 1 or 0. For generation and ranking tasks, similarity and ranking correlation metrics (like BERTScore or Kendall Tau) are more appropriate. Note that this table includes only a subset of commonly used ζ metrics; many other relevant metrics are omitted due to space limitations.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	CoLA
Reference	90.9	94.7	92.2	96.4	86.6	68.0
Replicated	91.2	94.7	92.1	96.9	84.8	65.3
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**
Reference	86.9	98.2	89.5	85.7	75.6	94.0
Replicated	85.4	100	86.3	84.9	71.2	90.0

Table 3: Reference and replicated scores on the GLUE and SuperGLUE tasks. These scores are obtained by training on the train set, validating and testing on the dev set.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	CoLA
Classes	2	2	2	2	2	2
Train samples	3668	104743	363846	67349	2490	8551
Dev samples	408	5463	40430	872	277	1043
Test samples	1725	5463	390965	1821	3000	1063
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**
Classes	2	3	2	2	2	2
Train samples	9427	250	2490	27243	5428	400
Dev samples	3270	56	277	4848	638	100
Test samples	3245	250	3000	9693	1400	500

Table 4: Data statistics for GLUE and SuperGLUE.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	CoLA
Random seed	42	72	42	52	52	72
Batch size	10	10	10	10	10	10
Epoch	8	6	8	7	10	8
Learning rate	2e-5	2e-5	1e-5	2e-5	1e-5	1e-5
Learning rate schedule type	*linear*	*linear*	*linear*	*linear*	*linear*	*linear*
Max sequence length	512	512	512	512	512	512
Gradient accumulation steps	2	2	2	2	2	2
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**
Random seed	62	52	72	72	42	52
Batch size	10	10	10	10	10	10
Epoch	8	7	10	6	8	9
Learning rate	1e-5	2e-5	2e-5	2e-5	1e-5	3e-5
Learning rate schedule type	*linear*	*linear*	*linear*	*linear*	*linear*	*linear*
Max sequence length	512	512	512	512	512	256
Gradient accumulation steps	2	2	2	2	2	2

Table 5: The hyperparameter settings for GLUE and SuperGLUE tasks to replicate the reference performance in Table 3 using `RoBERTa-large`.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	CoLA
Batch size	10	10	32	10	10	10
Epoch	20	6	2	6	20	6
Learning rate	3e-5	3e-5	2e-5	3e-5	3e-5	1e-5
Learning rate schedule type	linear	linear	linear	linear	linear	linear
Max sequence length	512	512	512	512	512	512
Gradient accumulation steps	2	2	2	2	2	2
LoRA r_q, r_v	8	8	8	8	8	8
LoRA α	16	16	16	16	16	16
LoRA Dropout	0.1	0.1	0.1	0.1	0.1	0.1
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**
Batch size	10	10	10	10	10	10
Epoch	10	20	10	10	10	10
Learning rate	1e-5	5e-5	1e-5	2e-5	1e-5	3e-5
Learning rate schedule type	linear	linear	linear	linear	linear	linear
Max sequence length	512	512	512	512	512	256
Gradient accumulation steps	2	2	2	2	2	2
LoRA r_q, r_v	8	8	8	8	8	8
LoRA α	16	16	16	16	16	16
LoRA Dropout	0.1	0.1	0.1	0.1	0.1	0.1

Table 6: The hyperparameters settings for GLUE and SuperGLUE tasks for `Llama3.2-3B`. For the COPA dataset we implement `LlamaForMultipleChoice` by ourselves.

GLUE							SuperGLUE						
Tasks	P	R	F1	Accuracy	CON	CCON	**Tasks**	P	R	F1	Accuracy	CON	CCON
MRPC	89.62	87.83	88.59	90.34	92.89	86.79	BoolQ	79.09	80.78	79.71	83.05	86.79	76.37
	(±1.39)	(±1.15)	(±1.02)	(± 0.89)	(± 1.18)	(± 0.91)		(±16.87)	(±10.82)	(±14.54)	(± 7.35)	(± 12.21)	(± 10.76)
QNLI	93.98	93.98	93.97	94.00	95.64	91.80	CB	89.43	90.81	89.61	94.64	91.50	89.84
	(±0.36)	(±0.36)	(±0.36)	(± 0.38)	(± 0.52)	(± 0.46)		(± 6.15)	(± 6.69)	(± 6.11)	(± 3.95)	(± 3.21)	(± 3.40)
QQP	91.22	91.73	91.46	92.00	95.57	89.79	RTES	71.71	76.49	72.99	72.89	70.05	64.58
	(±0.07)	(±0.17)	(±0.07)	(± 0.06)	(± 0.30)	(± 0.16)		(±25.46)	(±14.21)	(±21.72)	(± 16.67)	(± 18.36)	(±17.43)
SST2	95.48	95.45	95.46	95.59	96.83	94.01	MultiRC	67.62	74.32	70.05	76.46	74.63	63.77
	(±0.42)	(±0.43)	(±0.43)	(± 0.55)	(± 0.57)	(± 0.51)		(±26.94)	(±16.79)	(±23.24)	(±13.30)	(±18.94)	(±15.65)
RTEG	85.41	84.74	84.86	85.02	91.09	80.55	WiC	0.7053	0.6857	0.6778	68.57	80.36	58.75
	(±1.45)	(±1.54)	(±1.52)	(± 1.48)	(± 1.74)	(± 1.34)		(±2.68)	(±2.77)	(±3.28)	(± 2.77)	(± 6.07)	(± 4.60)
CoLA	-	-	-	65.61	93.95	82.76	COPA	73.03	72.85	72.78	73.00	64.88	55.44
	-	-	-	(± 1.32)	(± 0.71)	(± 0.51)		(± 16.69)	(± 16.59)	(± 16.81)	(± 16.83)	(± 17.35)	(± 11.98)

Table 7: Additional evaluation performance using `RoBERTa-large`. P - Precision, R - Recall, F1, Accuracy, CON - consistency, CCON - correct consistency. CoLA employs Matthew's correlation coefficient as the accuracy.

GLUE							SuperGLUE						
Tasks	P	R	F1	Accuracy	CON	CCON	**Tasks**	P	R	F1	Accuracy	CON	CCON
MRPC	81.98	80.31	81.03	84.02	90.34	79.19	BoolQ	70.78	69.29	69.74	72.49	81.10	63.09
	(± 0.57)	(±1.01)	(±0.79)	(± 0.56)	(±1.66)	(±0.70)		(±1.77)	(±2.07)	(±2.07)	(± 1.62)	(±1.67)	(±1.58)
QNLI	94.22	94.22	94.21	94.20	96.85	92.63	CB	56.33	56.23	55.32	73.92	89.82	70.35
	(±0.15)	(±0.15)	(±0.14)	(± 0.14)	(±0.24)	(±0.18)		(±10.07)	(±3.77)	(±5.78)	(± 2.03)	(±2.92)	(±2.55)
QQP	88.36	89.08	88.69	89.37	96.63	87.69	RTES	68.83	68.82	68.64	68.66	71.69	54.51
	(±0.07)	(±0.21)	(±0.12)	(± 0.09)	(±0.20)	(±0.09)		(±2.84)	(±2.83)	(±2.78)	(± 2.77)	(±3.82)	(±3.38)
SST2	96.78	96.79	96.78	96.78	98.32	95.95	MultiRC	79.72	80.02	79.83	80.14	86.73	73.50
	(±0.27)	(±0.27)	(±0.27)	(±0.27)	(±0.63)	(±0.44)		(±0.57)	(±0.55)	(±0.57)	(±0.57)	(±1.06)	(±0.84)
RTEG	85.92	85.91	85.88	85.92	88.80	80.32	WiC	68.99	68.71	68.60	68.71	85.86	61.64
	(±3.36)	(±3.34)	(±3.35)	(± 3.34)	(±3.05)	(±3.40)		(±1.64)	(±1.58)	(±1.58)	(± 1.58)	(±1.79)	(±1.08)
CoLA	-	-	-	61.33	94.86	81.36	COPA	84.08	84.05	84.03	84.20	86.8	77.6
	-	-	-	(± 0.88)	(±0.56)	(± 0.33)		(±5.25)	(±5.28)	(±5.24)	(±5.17)	(± 3.99)	(± 4.83)

Table 8: Additional evaluation performance using `Llama3.2-3B`. P - Precision, R - Recall, F1, Accuracy, CON - consistency, CCON - correct consistency. CoLA employs Matthew's correlation coefficient as the accuracy.