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-level impact through traditional metrics like accuracy and F1, calculating their mean and variance to quantify performance fluctuations. To capture the micro-level 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). Random seeds influence initialization and training dynamics, introducing variability in model outcomes that can lead to significant fluctuations in performance (Bengio, 2012).

In the field of NLP, large language models (LLMs) have achieved remarkable success across a wide range of NLP tasks, setting new state-of-the-art results on benchmarks like GLUE and SuperGLUE. These benchmarks have became the de facto standard for evaluating the capabilities of LLMs in understanding, reasoning, and generating natural language. Despite the success of LLMs, the pretrained transformer architectures, such as BERT (Devlin et al., 2019) and RoBERTa (Liu, 2019), have been found to be particularly sensitive to random seeds (Risch and Krestel, 2020; Dodge et al., 2020; Mosbach et al., 2021). This sensitivity can result in substantial performance variations, complicating the interpretation of experimental results and undermining confidence in benchmarking or state-of-the-art outcomes. 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 the influence of random seeds on LLM performance is still an underexplored area.*

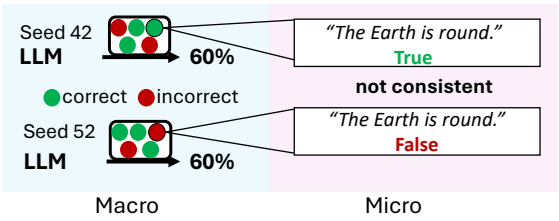

Figure 1: Macro and micro performance. A pretrained LLM is fine-tuned with random seed 42 and 52. The accuracy for both models is 60%, but the overlapping of individual predictions is 20%.

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 and F1 score for classification tasks, or Pearson correlation for regression tasks, across multiple seeds. These evaluations focus on the **macro-level** agreement of model performance across the entire test set, offering insights into overall variability. However, *they overlook the* ***micro-level*** *impact of how individual test points are influenced by random seed variations*. As shown in Figure 1, model performance is robust to random seeds 42 and 52 at the macro level (both achieve 60% accuracy) but lacks consistency at the micro level (only 20% overlapping predictions). This

micro-level 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-level effect is crucial for assessing model robustness at the level of individual predictions, ensuring that specific test samples are not inconsistently misclassified or predicted due to seed-induced variations. 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 variations 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-level variability; (2) We introduce a novel **consistency** metric to assess prediction stability on individual test points, capturing the micro-level effects of random seeds; (3) Our extensive experiments reveal significant variability in both standard and consistency metrics, underscoring the need to consider seed-induced variations in fine-tuning and evaluation, and incorporate random seed sensitivity into benchmarking and reporting for more reliable and reproducible results.

2 Macro Metric: Variance

To measure the macro-level impact of random seeds on LLM performance, we calculate the variance of a standard metric across multiple seeds. Let $[\zeta_1, \cdots, \zeta_S]$ represent the values of a model performance metric for LLMs 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 F1 score for classification tasks or Pearson correlation for regression tasks. A smaller VAR indicates less variation in macro-level 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. For *classification tasks*, we calculate the consistency as follows:

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

where $1_{A,B}(\cdot)$ is the indicator function that equals 1 if $y_t^A = y_t^B$, otherwise 0. And the correct-consistency is calculated by:

$$\text{ACC-CON:}\frac{1}{N}\sum_{t=1}^{N}1_{A,B,r}(t) \tag{3}$$

where $1_{A,B,r}(\cdot)$ is the indicator function that equals 1 if $y_t^A = y_t^B = r_t$, otherwise 0.

For *regression tasks*, we calculate the consistency as the Pearson correlation between two sets of predictions $[y_1^A, ..., y_N^A]$ and $[y_1^B, ..., y_N^B]$:

$$\text{CON-PEAR:}\frac{\sum_{i=1}^{N}\left(y_i^A - \overline{y^A}\right)\left(y_i^B - \overline{y^B}\right)}{\sqrt{\sum_{i=1}^{N}\left(y_i^A - \overline{y^A}\right)^2}\sqrt{\sum_{i=1}^{N}\left(y_i^B - \overline{y^B}\right)^2}} \tag{4}$$

CON and ACC-CON range from [0, 1], while CON-PEAR ranges from [-1, 1], where higher values indicate smaller variations in micro-level predictions. 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-level impact of random seeds on the same pretrained LLM.

4 Experimental Setup

In this section, we will show the details of our experiment settings. More setting details can be

found in Appendix.

4.1 Benchmarks

In this study, we conduct experiments on a range of NLP tasks including COLA (Matthews Correlation Coefficient), SST2 (Accuracy), MRPC (Accuracy), STSB (Pearson correlations), QQP (Accuracy), QNLI (Accuracy), and RTE (Accuracy) from GLUE (Wang et al., 2018) benchmark; RTE (Accuracy), CB (Accuracy), WiC (Accuracy), BoolQ (Accuracy), MultiRC (Accuracy), and COPA (Accuracy) from SuperGLUE (Wang et al., 2019) benchmark. In our paper, we use RTEG to denote RTE task from GLUE and RTES for SuperGLUE.

To ensure our experiments are conducted with proper settings and are reproducible, we followed the configurations and replicated the state-of-the-art (SOTA) scores reported in (Liu, 2019). We choose `RoBERTa-large` as the pretrained LLM. 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.

4.2 Settings

Our experiments were implemented using Hugging Face Transformers (v4.30.0) and PyTorch (v2.0), conducted on NVIDIA A100 GPUs with 40GB of memory across two GPUs. With limited computational resources, we perform fully fine-tuning for each task with five random seeds: 42, 52, 62, 72, 82 (i.e., $S = 5$), which are randomly chosen except 42 (see Section 5.1). To calculate CON, ACC-CON, and CON-PEAR, we compute the metric for each of the 10 random seed pairs (e.g., 42 and 52) and then take the average of these 10 values as the final consistency score. We used the `run_glue.py` PyTorch script for fine-tuning, and default settings were applied unless otherwise specified. Although differences in the fine-tuning script and missing settings from the original authors prevented us from reproducing the exact SOTA scores, our results are close to the reported SOTA. A comparison of our implementation with the reference SOTA scores and detailed data and learning settings are provided in Appendix Table 2, Table 3, and Table 4.

5 Results and Discussion

In this section, we will show experimental results at both macro- and micro-level and discuss key findings.

SuperGLUE	BoolQ	CB	RTES	MultiRC	WiC	COPA
ACC	85.05	98.8	69.6	79.01	68.4	73.2
VAR	0.24	1.1	18.22	12.21	2.83	12.83
CON	94.78	91.61	71.84	76.16	79.09	67.6
ACC-CON	82.44	90.18	56.61	67.10	57.95	57

GLUE	MRPC	QNLI	QQP	SST2	RTEG	COLA	STSB
ACC	90.34	94.53	92.02	95.73	84.04	64.51	92.19
VAR	0.93	0.16	0.07	0.71	0.47	0.64	0.30
CON	92.21	96.84	96.00	98.10	92.78	94.23	0.9853
ACC-CON	52.33	92.95	90.03	94.58	80.43	82.52	-

Table 1: Macro- and micro-impact of five random seeds. ACC is the average of five accuracies (SOTA metrics). VAR is the variance of Accuracy calculated using Equation 1. CON and ACC-CON are the average of 10 values, each derived from Equation 2, 3, or 4. For STSB, we put CON-PEAR value in CON row for concise format. ACC, CON, and ACC-CON are expressed as percentages.

5.1 Macro impact

Table 1 presents the averaged accuracy (ACC) and variance (VAR) for GLUE and SuperGLUE tasks across five random seeds. Significant variance in macro-level performance is observed in many tasks, reflecting sensitivity to random seed selection. Tasks like RTES (VAR = 18.22), COPA (VAR = 12.83), and MultiRC (VAR = 12.21) exhibit high variability in ACC, highlighting *the need for robust evaluation methods and stability-enhancing techniques, such as more robust optimization methods, better hyperparameter tuning, or ensembling across multiple seeds*. In contrast, tasks like QQP (VAR = 0.07) and QNLI (VAR = 0.16) show much greater stability, likely due to their inherent properties such as larger datasets or simpler decision boundaries. High variability undermines the reliability of single-seed evaluations, emphasizing the importance of averaging results and addressing task-specific challenges to improve model robustness.

There is a "common belief" in the machine learning community that random seed 42 may outperform others. To investigate whether a specific random seed consistently leads to better results across different models or tasks, in Figure 2 we present a heatmap of normalized ACC for each task across five random seeds. 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*.

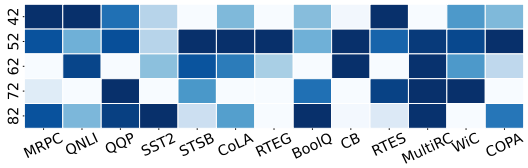

Figure 2: A heatmap of normalized ACC across tasks and five random seeds, with a darker color representing a better accuracy.

5.2 Micro impact

Table 1 reports consistency (CON) and correct-consistency (ACC-CON) for GLUE and SuperGLUE tasks across five random seeds. High CON values in tasks like SST2 (98.1%), QNLI (96.84%), and QQP (96%) indicate stable predictions, while lower values for RTES (71.84%) and COPA (67.7%) highlight their sensitivity to random seeds, potentially due to smaller training sizes or task complexity. High ACC-CON in SST2 (94.58%) and QNLI (92.95%) suggest stable correct predictions, whereas low ACC-CON in RTES (56.6%) and MRPC (52.33%) reveal that *consistent predictions are not always accurate, emphasizing the need to evaluate both stability and correctness*. Additionally, MRPC's low VAR (0.93) value demonstrates that *similar macro-level accuracy does not necessarily imply true reproducibility, underscoring the importance of micro-level analysis beyond macro-level metrics*.

Identifying robust data points—those consistently predicted correctly through micro-level analysis—and leveraging them to enhance data collection, preprocessing, prompt engineering, or synthetic data generation offer a potential solution for mitigating seed-induced variability and improving LLM robustness.

5.3 Training size impact

Training size significantly influences a model's predictive performance, with larger datasets generally improving accuracy, though this is not guaranteed due to factors like task complexity and label noise (Shahinfar et al., 2020; Althnian et al., 2021; Bailly et al., 2022). Will increasing training data size improve variance and consistency in general? To answer the question, we show Pearson correlation analysis between training size, variance, and consistency in Figure 3. It reveals a weak negative correlation (-0.25) between training size and VAR, indicating that smaller datasets tends to increase performance variance, as seen in RTES (highest VAR of 18.22 with relatively small training size). However, the effect is not pronounced or consistent across all tasks, as MultiRC and WiC exhibit high VAR despite a relatively large dataset. A weak or moderate positive correlation is observed between training size and both CON (0.41) and ACC-CON (0.43), suggesting larger datasets generally improve consistency and prediction stability across random seeds, but with no guarantee.

Increasing training size can reduce both macro and micro variability to random seeds, but its effectiveness depends on factors like data quality, task complexity, and label noise. Alternatively, as discussed in Section 5.2, identifying robust data points and augmenting the training data with data points having similar robust patterns (either real data or generated synthetic data) provide a more targeted strategy to mitigate seed-induced variability and improve LLM robustness.

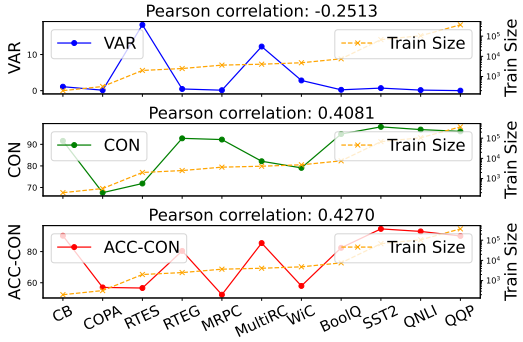

Figure 3: Correlation between training size (log scale), VAR, CON, and ACC-CON. Tasks are arranged in ascending order of training size, with exact sizes detailed in Appendix 3.

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 variations 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

Due to limited computing resources, our experiments were conducted with only five random seeds, which may not be sufficient for drawing

broader generalizations of the findings and implications. Additionally, the reference SOTA scores for GLUE and SuperGLUE tasks were obtained using the pretrained LLM `RoBERTa-large`, and therefore, we conducted experiments solely on `RoBERTa-large`. Expanding the experiments to include various LLMs, particularly larger-scale models, would strengthen our findings and conclusions. Furthermore, incorporating more NLP benchmark datasets would provide a more comprehensive evaluation, as diverse datasets would better capture variability across tasks, domains, and data distributions, ultimately enhancing the robustness and applicability of our analysis. Additionally, our findings and implications are more suited for classification tasks, as only 1 out of the 13 tasks in our experiments is a regression task. Therefore, more comprehensive experiments should be conducted specifically on various regression tasks. The ACC-CON metric, which is not directly applicable to regression tasks, hinders the ability to evaluate correct consistency in this context.

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.

Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade: Second edition*, pages 437–478. Springer.

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.

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*.

Marian Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. 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.

A Appendix

A.1 Data Description

Table 3 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.2 Hyperparameter Settings

Table 4 provides the detailed hyperparameter configurations. Unless stated otherwise, we adopt the default hyperparameter values from the Hugging Face framework.

A.3 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. Our replicated accuracy scores for the GLUE and SuperGLUE tasks, presented in Table 2, are directly comparable and align with those shown in Table 1 of the main paper and Table 5 in Section A.4.

A.4 Additional Results

Table 5 presents model performance across various metrics, including precision (P), recall (R), F1 score, accuracy, CON, and ACC-CON, 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.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	COLA	STSB
Reference	90.9	94.7	92.2	96.4	86.6	68.0	92.4
Replicated	91.2	94.7	92.1	96.9	84.8	65.3	92.5
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 2: 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	STSB
Classes	2	2	2	2	2	2	-
Train samples	3668	104743	363846	67349	2490	8551	5749
Dev samples	408	5463	40430	872	277	1043	1500
Test samples	1725	5463	39096	1821	3000	1063	1379
SuperGLUE	**BoolQ**	**CB**	**RTES**	**MultiRC**	**WiC**	**COPA**	
Classes	2	3	2	2	2	2	
Train samples	9427	250	2500	27243	5428	400	
Dev samples	1886	50	500	4848	1200	100	
Test samples	3270	57	278	953	638	500	

Table 3: Data statistics for GLUE and SuperGLUE.

GLUE	MRPC	QNLI	QQP	SST2	RTEG	COLA	STSB
Random seed	42	72	42	52	52	72	42
Batch size	10	10	10	10	10	10	32
Epoch	8	6	8	7	10	8	3
Learning rate	2e-5	2e-5	1e-5	2e-5	1e-5	1e-5	4e-5
Learning rate schedule type	*linear*	*linear*	*linear*	*linear*	*linear*	*linear*	*linear*
Max sequence length	512	512	512	512	512	512	512
Gradient accumulation steps	2	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 4: The hyperparameter settings for GLUE and SuperGLUE tasks to replicate the reference performance in Table 2.

GLUE							**SuperGLUE**						
Tasks	P	R	F1	Accuracy	CON	ACC-CON	**Tasks**	P	R	F1	Accuracy	CON	ACC-CON
MRPC	91.67	94.48	93.04	90.34	92.21	52.33	BoolQ	87.69	88.36	88.03	85.05	94.78	82.44
	(±0.47)	(±1.9)	(±0.75)	(±0.93)	(±0.98)	(±1.19)		(±0.39)	(±0.32)	(±0.18)	(±0.24)	(±0.28)	(±0.24)
QNLI	95.47	93.62	94.53	94.53	96.84	92.95	CB	99.13	98.26	98.67	98.8	91.61	90.18
	(±0.28)	(±0.31)	(±0.17)	(±0.16)	(±0.3)	(±0.21)		(±1.95)	(±2.38)	(±1.22)	(±1.1)	(±2.53)	(±2.95)
QQP	87.67	91.17	89.38	92.02	96.00	90.03	RTES	76.31	65.34	76.34	69.6	71.84	56.61
	(±0.36)	(±0.53)	(±0.11)	(±0.07)	(±0.07)	(±0.04)		(±19.45)	(±38.38)	(±9.12)	(±18.22)	(±17.63)	(±14.4)
SST2	95.56	95.68	95.6	95.73	98.1	94.58	MultiRC	79.81	68.24	82.46	79.01	76.16	67.09
	(±0.57)	(±0.89)	(±0.36)	(±0.71)	(±0.27)	(±0.27)		(±0.66)	(±38.17)	(±0.65)	(±12.21)	(±18.87)	(±16.48)
RTEG	87.57	77.25	82.08	84.04	92.78	80.43	WiC	65.12	79.06	71.28	68.4	79.09	57.95
	(±1.28)	(±1.13)	(±0.52)	(±0.47)	(±1.09)	(±0.74)		(±1.29)	(±8.29)	(±4.17)	(±2.83)	(±7.00)	(±4.99)
COLA	-	-	-	64.51	94.23	82.52	COPA	70.20	72.20	71.00	73.20	67.60	57.00
	-	-	-	(±0.64)	(±0.64)	(±0.35)		(±14.46)	(±13.79)	(±13.55)	(±12.83)	(±10.29)	(±11.98)
STSB	-	-	-	92.19	98.53	-							
	-	-	-	(±0.30)	(±0.16)	-							

Table 5: Evaluation metrics used in this study. Accuracy is employed for all tasks except STSB and CoLA, where Pearson correlation and Matthew's correlation coefficient are used, respectively. CON - consistency, ACC-CON - correct consistency.