Measuring the Instability of Fine-Tuning

Yupei Du and Dong Nguyen
Utrecht University
Utrecht, the Netherlands
{y.du,d.p.nguyen}@uu.nl

Abstract

Fine-tuning pre-trained language models on downstream tasks with varying random seeds has been shown to be unstable, especially on small datasets. Many previous studies have investigated this instability and proposed methods to mitigate it. However, most studies only used the standard deviation of performance scores (SD) as their measure, which is a narrow characterization of instability. In this paper, we analyze SD and six other measures quantifying instability at different levels of granularity. Moreover, we propose a systematic framework to evaluate the validity of these measures. Finally, we analyze the consistency and difference between different measures by reassessing existing instability mitigation methods. We hope our results will inform the development of better measurements of fine-tuning instability.[1]

1 Introduction

Since the introduction of BERT (Devlin et al., 2019), the pre-train-then-fine-tune paradigm has achieved state-of-the-art performance across many NLP benchmarks (Sun et al., 2021; Fedus et al., 2022; Chi et al., 2022). However, despite its wide success, the fine-tuning process, especially when fine-tuning large models on *small datasets*, is shown to be unstable: fine-tuning a given model with varying random seeds can lead to different performance results (Lee et al., 2020; Dodge et al., 2020; Mosbach et al., 2021; Hua et al., 2021). This instability makes the investigation of better architectures and instability mitigation methods (IMMs) challenging (Zhang et al., 2021).

Many previous studies have investigated fine-tuning instability (Dodge et al., 2020; Lee et al., 2020; Mosbach et al., 2021; Zhang et al., 2021). In these studies, the most prevalent instability measure is the *standard deviation of performance* (**SD**, e.g. the standard deviation of F1-scores). However, as we discuss in §3 and §6, SD can only offer very limited assessments. For example, classifiers can obtain the same accuracy score (i.e. zero SD) even when they neither make the same predictions on each example (*prediction instability*) nor have the same hidden representations (*representation instability*). Therefore, it is important to also use other measures that can address the weaknesses of SD.

However, it is difficult to decide which measures to use: because instability is an abstract concept, it is hard to examine *to which extent a measure indeed quantifies what it intends to measure*. This property is called **validity** in measurement theory (William M. K., 2023). For example, using the average accuracy of models as an instability measure would have low validity, because how accurate these models make predictions does not reflect their stability.

To better assess the instability of fine-tuning pre-trained language models (PLMs), we study more measures concerning instability at different granularity levels (Summers and Dinneen, 2021; Khurana et al., 2021; Raghu et al., 2017; Kornblith et al., 2019; Ding et al., 2021) and develop a framework to assess their validity. We focus on BERT and RoBERTa for their popularity, but our framework can also be applied to other PLMs. Concretely,

- First, we discuss six other instability measures at different granularity levels in addition to SD, and categorize them into *prediction measures* and *representation measures* based on the type of instability they focus on (§3).
- Second, we propose a framework to systematically assess two types of validity of these measures, without relying on labelled data (§5).
- Third, we investigate the consistency and differences between different measures by reassessing the effectiveness of existing IMMs, analyzing their correlations (§6.1), and performing bootstrap analyses (§6.2). We find that measures at

[1] Our implementation is available at https://github.com/nlpsoc/instability_measurement.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6209–6230
July 9-14, 2023

different granularity levels do not always produce consistent instability scores with each other and tend to differ more when the models are more stable. Moreover, based on our observations, we offer two suggestions for future studies: (1) use multiple instability measures, especially when models are more stable; (2) use only one prediction and one representation measure when limited computational resources are available (§6.3).

2 Background

2.1 Instability of Fine-tuning

The seminal work of BERT by Devlin et al. (2019) has already shown that fine-tuning PLMs is unstable regarding the choices of random seeds. This observation was further confirmed by other studies on more PLMs, including RoBERTa (Liu et al., 2019; Lan et al., 2020; Phang et al., 2018; Lee et al., 2020; Dodge et al., 2020; Mosbach et al., 2021; Zhang et al., 2021; Sellam et al., 2022). Most of these studies used SD to measure the instability.

Different explanations have been proposed to account for the instability of fine-tuning PLMs on small datasets, including catastrophic forgetting (Lee et al., 2020)[2], the lack of Adam bias correction (Mosbach et al., 2021; Zhang et al., 2021), too few training steps (Mosbach et al., 2021), and task-specific top layers (Zhang et al., 2021).

2.2 Instability Mitigation Methods (IMMs)

Various IMMs have been used to mitigate the instability of fine-tuning PLMs. Following Zhang et al. (2021), we focus on four methods for their popularity. Nevertheless, we acknowledge the existence of other methods, including entropy regularization and co-distillation (Bhojanapalli et al., 2021), and component-wise gradient norm clipping (Yang and Ma, 2022).

Mixout (Lee et al., 2020) is a generalized version of Dropout (Srivastava et al., 2014). It randomly replaces the outputs of neurons with the ones produced by the pre-trained weights by a probability p. In this way, it can mitigate the catastrophic forgetting of pre-trained knowledge which potentially stabilizes fine-tuning.

[2]Although several studies assumed that catastrophic forgetting causes instability (Kirkpatrick et al., 2017; Schwarz et al., 2018; Lee et al., 2020), Mosbach et al. (2021) argued against it.

$\mathbf{WD_{pre}}$ (Li et al., 2018) is a variant of weight decay: after each optimization step, each model weight w will move a step size of λw towards the pre-trained weights, where λ is a hyper-parameter. $\mathsf{WD_{pre}}$ also aims to improve the fine-tuning instability by mitigating catastrophic forgetting.

Layer-wise Learning Rate Decay (Howard and Ruder, 2018, LLRD) assigns decreasing learning rates from the topmost layer to the bottom layer by a constant hyper-parameter discounting factor η. Howard and Ruder (2018) empirically show that models trained using LLRD are more stable, by retaining more generalizable pre-trained knowledge in bottom layers, while forgetting specialized pre-train knowledge in top layers.

Re-init (Zhang et al., 2021) stabilizes fine-tuning by re-initializing the top k layers of PLMs. The underlying intuition is similar to LLRD: top layers of PLMs contain more pre-train task specific knowledge, and transferring it may hurt stability.

3 Instability Measures

Despite its wide usage, *SD only provides a narrow view of the instability of models*. For example, consider fine-tuning two pre-trained models on the same classification task. If one of them makes correct predictions only on half of the test data, while the other model makes correct predictions only on the other half, these two models will both have a 0.5 accuracy score and therefore no instability would be measured using SD. However, they actually make different *predictions* on each data point (i.e. **prediction instability**). Moreover, even if these two models achieve the same accuracy by making identical predictions, due to the over-parameterization of PLMs (Roeder et al., 2021), they can have different sets of hidden *representations* (i.e. **representation instability**).

To better assess these two types of instability, we study six other instability measures at different granularity levels in addition to SD. Furthermore, according to the instability types that these measures intend to quantify, we categorize these measures into two types: *prediction measures* (§3.1) and *representation measures* (§3.2). All these instability measures have a continuous output range 0–1, with higher values indicating lower stability.

It is worth noting that similar categorizations have been used before. For example, Csiszárik et al. (2021) categorized measures as *functional* and

representational. However, they used functional similarity to refer to the function compositions that different components of the models realize. Also, Summers and Dinneen (2021) categorized measures as *performance variability* and *representation diversity*. However, they used *performance variability* to specifically refer to SD and used *representation diversity* to refer to all other measures at different granularity levels that we study here.

Notation Formally, suppose we have a dataset consisting of n data points. We fine-tune m BERT models $\{M_1, M_2, \ldots, M_m\}$, with the same settings except for m different random seeds. We use p_i^k and $\hat{y}_i^k$ to denote the class probability and the prediction of M_i on the k-th test sample.

Assume the l-th layer of M_i consists of e neurons, we use $M_i^l \in \mathbb{R}^{n \times e}$ to denote this layer's *centered* representation, w.r.t. all n data points (all representation measures discussed below require us to center the representations). Representation measures involve computing the distances between the representations derived from the same layer of two different models. We use $d_{i,j}^l$ to represent the distance between M_i^l and M_j^l.

3.1 Prediction Measures

We refer to measures that assess the prediction instability of models as prediction measures. In other words, prediction measures only assess the output of the models (i.e. logits and predictions). In this paper, we study three prediction measures besides SD: *pairwise disagreement*, *Fleiss' Kappa*, and *pairwise Jensen-Shannon divergence (pairwise JSD)*. Among these three measures, both pairwise disagreement and Fleiss' Kappa quantify the instability of the discrete predictions of models, and therefore are at the same granularity level. Pairwise JSD looks at continuous class probabilities and is thus more fine-grained. Nevertheless, they are all more fine-grained than SD, which only considers the overall performance.

Pairwise Disagreement Following Summers and Dinneen (2021), we measure the models' instability by averaging the *pairwise disagreement* among models' predictions. Formally,

$$\mathcal{I}_{\text{pwd}} = \frac{2}{nm(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{k=1}^{n} \mathbb{1}(\hat{y}_i^k \neq \hat{y}_j^k),$$

where $\mathbb{1}$ is the indicator function. We note that our definition of pairwise disagreement relates closely to *churn* and *jitter* proposed and used by Milani Fard et al. (2016) and Bhojanapalli et al. (2021); Liu et al. (2022).

Fleiss' Kappa Similar to Khurana et al. (2021), we adopt Fleiss' Kappa, which is a popular measure for inter-rater consistency (Fleiss, 1971), to measure the consistency among different models' predictions. Because Fleiss' Kappa is negatively correlated with models' instability and ranges from 0 to 1, we use its difference with one as the output, to stay consistent with other measures. Formally,

$$\mathcal{I}_{\kappa} = 1 - \frac{p_a - p_\epsilon}{1 - p_\epsilon},$$

where p_a is a term evaluating the consistency of models' predictions on each test sample, and p_ϵ is an error correction term (Details in Appendix B).

Pairwise JSD The previous two measures only look at discrete labels, while continuous class probabilities contain richer information about a model's predictions. Therefore, we average the pairwise JSD of models' class probabilities to obtain a finer-grained evaluation of instability. Formally,

$$\mathcal{I}_{\text{JSD}} = \frac{2}{nm(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \sum_{k=1}^{n} JSD(p_i^k \| p_j^k),$$

where $JSD(\cdot\|\star)$ is the JSD between $\cdot$ and $\star$.

3.2 Representation Measures

We refer to measures that assess the instability of models based on their hidden representations as representation measures. Here, we study three representation measures: *singular vector canonical correlation analysis* (SVCCA, Raghu et al., 2017), *orthogonal Procrustes distance* (OP, Schönemann, 1966), and *linear centered kernel alignment* (Linear-CKA, Kornblith et al., 2019). Because all representation measures look at the hidden representations of models, they are at the same granularity level, which is more fine-grained than prediction measures.

All these three measures are originally developed to compute the distance between a pair of representations (although Linear-CKA is also used by Summers and Dinneen (2021) to study model instability). With these measures, we are able to analyze the behavior of neural networks, going beyond the model predictions alone (Kornblith et al., 2019). To evaluate the instability of all m models

regarding a specific layer l, $\mathcal{I}^l$, we average the distance d of each possible pair of models. Formally,

$$\mathcal{I}^l = \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} d_{i,j}^l.$$

We next describe how to calculate d for each representation measure. We respectively denote the instability score of each measure after aggregating d as $\mathcal{I}_{\text{SVCCA}}$, $\mathcal{I}_{\text{CKA}}$, and $\mathcal{I}_{\text{OP}}$.

SVCCA (Raghu et al., 2017) is developed based on canonical correlation analysis (CCA, Hardoon et al., 2004). For two representations M_i^l and M_j^l, CCA finds e orthogonal bases so that their correlations after projection are maximized.

Formally, for $1 \leq k \leq e$,

$$\rho_k = \max_{\mathbf{w}_i^k, \mathbf{w}_j^k} \text{corr}\left(M_i^l \mathbf{w}_i^k, M_j^l \mathbf{w}_i^k\right),$$
$$\text{s.t. } \forall k_1 < k_2,\ M_i^l \mathbf{w}_i^{k_1} \perp M_i^l \mathbf{w}_i^{k_2} \text{ and } M_j^l \mathbf{w}_j^{k_1} \perp M_j^l \mathbf{w}_j^{k_2},$$

where $\mathbf{w}_i^k, \mathbf{w}_j^k \in \mathbb{R}^{p_1}$. After obtaining ρ, we use the *mean correlation coefficient* to transform ρ into a scalar dissimilarity measure. Formally,

$$d_{\text{CCA}} = 1 - \frac{1}{e} \sum_{k=1}^{e} \rho_k.$$

Raghu et al. (2017) find that meaningful information usually distributes in a lower-dimensional subspace of the neural representations. To avoid overfitting on noise, SVCCA first uses singular-value decomposition to find the most important subspace directions of the representations.[3] The representations are then projected onto these directions, followed by CCA. We again calculate the mean ρ as the d_{SVCCA}.

OP (Ding et al., 2021) consists of computing the minimum Frobenius norm of the difference between M_i^l and M_j^l, after M_i^l being transformed by an orthogonal transformation. Formally,

$$\min_R \|M_j^l - M_i^l R\|_{\text{F}}^2, \quad s.t.\ R^\top R = I.$$

Schönemann (1966) provides a closed-form solution of this problem. To constrain the output range to be between zero and one, we normalize the representations with their Frobenius norms. Formally,

$$d_{\text{OP}}(M_i^l, M_j^l) = 1 - \frac{\left\|M_i^{l\top} M_j^l\right\|_*}{\left\|M_i^{l\top} M_i^l\right\|_F \left\|M_j^{l\top} M_j^l\right\|_F},$$

where $\|\cdot\|_*$ is the nuclear norm.

[3]Following Raghu et al. (2017), we keep directions that explain 99% of the representations.

Linear-CKA measures the representation distance by the similarity between representations' inter-sample similarity $\langle M_i^{l\top} M_i^l, M_j^{l\top} M_j^l \rangle$ (Kornblith et al., 2019). After normalizing the representations with Frobenius norms, we obtain a similarity score between zero and one. We then use its difference with one as the distance measure. Formally,

$$d_{\text{CKA}}(M_i^l, M_j^l) = 1 - \frac{\left\|M_i^{l\top} M_j^l\right\|_F^2}{\left\|M_i^{l\top} M_i^l\right\|_F \left\|M_j^{l\top} M_j^l\right\|_F}.$$

4 Experimental Setup

We study the instability of fine-tuning BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) empirically. Following Lee et al. (2020) and Zhang et al. (2021), we perform our experiments on three small datasets of the GLUE benchmark (Wang et al., 2018): RTE, MRPC (Dolan and Brockett, 2005), and CoLA (Warstadt et al., 2019), because models trained on small datasets are observed to be less stable (Zhang et al., 2021).[4]

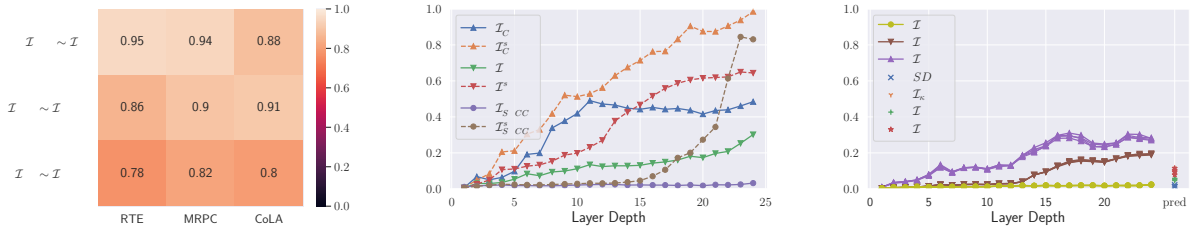
Unless specified, we fine-tune BERT-large and RoBERTa-large models from HuggingFace Transformers (Wolf et al., 2020), with a 16 batch size, a 0.1 Dropout rate, and a 2×10^{-5} learning rate, using de-biased Adam, as well as a linear learning rate warm-up during the first 10% steps followed by a linear decay, following Zhang et al. (2021). Consistent with Mosbach et al. (2021), we train the models for five epochs with 20 random seeds.

Consistent with Zhang et al. (2021), we divide the validation data into two equally sized parts, respectively as new validation and test data, because we have no access to the GLUE test datasets. Moreover, we keep the checkpoint with the highest validation performance and obtain all our results on the test set. More details are provided in Appendix A.

5 Assessing the Validity of Instability Measures

It is not trivial to assess the validity of instability measures, because there is no clear ground truth. Nevertheless, we can still perform validity assessments by building on approaches from measurement theory. Here, we propose a framework to assess two important types of validity (William M. K., 2023), by computing their correlations with

[4]To study whether our findings also generalize to larger datasets, we include a pilot study on SST-2 (8× larger than CoLA) in Appendix C. As expected, we observe higher stability. Furthermore, the behaviors of measures are consistent with those observed on smaller datasets.



(a) Convergent validity (b) Successful vs. failed runs (c) Consistency among sub-samples

Figure 1: Validity assessment. Figure 1a show Pearson's r between different representation measures for BERT. For Figure 1b and 1c, X-axis and Y-axis are the layer depth and instability scores respectively. Figure 1b shows the differences of representation measures between successful ($*^s$) and failed runs ($*^f$) for BERT on RTE. Figure 1c shows the consistency of different measures for BERT on MRPC across different sub-samples (different lines).

each other (convergent validity, §5.1) and observing their responses to different inputs (concurrent validity, §5.2). Except for SVCCA, all other measures show good validity in our tests, and hence they are suitable for examining fine-tuning instability.[5] Although there are other types of validity (e.g. face, content, discriminative, and predictive validity), we select these two types because of their relevance to our study and our lack of labelled test data. Ding et al. (2021) also provided a framework to evaluate the sensitivity (i.e. responding to important changes) and specificity (i.e. ignoring changes that do not matter) of the representation similarity metrics (i.e. $d_{\text{CKA}}, d_{\text{OP}}, d_{\text{SVCCA}}$). However, their framework was not build on validity theory and they did not consider prediction measures.

5.1 Convergent Validity

In measurement theory, convergent validity refers to validity established by correlations with measures that are theoretically related (Gravetter and Forzano, 2018). In other words, *if two measures aim to quantify the same underlying concept, their measurements should have a high correlation*. It is worth noting that convergent validity usually should be evaluated against established and validated measures (Jackman, 2008). However, in our case, none of the measures have been validated before. Therefore, low convergent validity may have different causes: for example, it can be that only one of the measures is invalid, or that these measures quantify different aspects of the concept.

For representation measures, we have an instability score for each hidden layer. We therefore assess their convergent validity by computing Pearson's r between instability scores that different measures assign to different layers of the same group of models (e.g. BERT fine-tuned on RTE with different random seeds). We show the results on BERT in Figure 1a. All three representation measures correlate highly (> 0.77) with each other, which suggests a good convergent validity. For prediction measures, we only have a single scalar output on each dataset/PLM combination. It is thus not practical to estimate their convergent validity directly because the sample size (i.e. the number of dataset/PLM combinations) is too small. In §6, we offer a detailed discussion and observe that they actually show good convergent validity.

5.2 Concurrent Validity

In measurement theory, concurrent validity refers to the "ability to distinguish between groups that it should theoretically be able to distinguish between" (William M. K., 2023). We therefore test concurrent validity based on the following assumption: a valid instability measure should not only be able to distinguish groups of models with substantial instability differences, but also be unable to distinguish groups of models with trivial instability differences. Concretely, we treat substantial instability differences as the differences *between successful/failed fine-tuning runs*, and define trivial instability differences as *different i.i.d. test datasets*.[6] We accordingly present two analyses.

Differences between successful and failed runs Previous studies have identified failed fine-tuning

[5] We note that passing our tests does not necessarily imply that a measure is perfectly valid: it is also possible that our validity tests/datasets/PLMs are not comprehensive (i.e. mono-method and mono-operation biases, William M. K., 2023). Moreover, as aforementioned in §3, different measures may concern different aspects of instability and should usually be used together. We offer a more extensive discussion in §6.

[6] Our tests are inspired by the concurrent validity definition, rather than strictly following it. See the limitations section.

runs where the training fails to converge (Dodge et al., 2020; Mosbach et al., 2021).[7] In particular, Mosbach et al. (2021) observe that failed runs suffer from vanishing gradients. Because all runs start from the pre-trained weights, and the vanishing gradient makes the models update less intensively, this observation leads to the following assumption: *compared with successful runs, failed runs bear lower representation instability*. In this analysis, we use this assumption to evaluate the concurrent validity of representation measures, by testing whether they are able to distinguish failed from successful runs. Because of this former observation only applies to hidden representations, in this analysis we exclude prediction measures.

Specifically, we train our models using the same 20 random seeds and keep the last checkpoint for each seed. We adopt larger learning rates: 5×10^{-5} for BERT and 3×10^{-5} for RoBERTa, because failed runs occur more frequently with larger learning rates (Mosbach et al., 2021). For each group of models, we obtain 9–13 failed runs out of 20 runs.

We show our results for BERT on RTE in Figure 1b and observe similar patterns on other PLMs/datasets (see Appendix E). Linear-CKA and OP indeed indicate a lower instability in failed runs. This observation is consistent with our expectation, suggesting the concurrent validity of these two measures. However, SVCCA fails to distinguish successful and failed runs based on the representations in the bottom layers, and therefore fails this test. One plausible explanation is that because lower layers of models tend to update less intensively during fine-tuning (the nature of back-propagation), they are likely to be more stable, and SVCCA may ignore these smaller differences.

Differences among test datasets Because we aim to quantify the instability of models themselves, one desideratum of a valid measure is to be independent of the specific data samples used to obtain the predictions and representations of models, as these data samples are not inherent components of these models. Concretely, we expect *a valid measure to produce similar outputs for the same group of models when the instability scores are computed using different i.i.d. datasets*.

To evaluate the input invariance of the measures, we create four sub-samples with half the test dataset size for each task, by uniformly sampling without replacement. We then compute the instability scores using both prediction and representation measures on all samples, and show the results for BERT on MRPC in Figure 1c (we include results for RoBERTa and on MRPC/CoLA in Appendix E). We observe that the variance among different samples is very small, suggesting that all these measures show good concurrent validity in this test.[8]

6 The Need to Use Different Measures

In §5, all measures discussed in §3 except for SVCCA showed good validity in our tests, thus they are capable of measuring fine-tuning instability. However, the following question remains: *when do we need which instability measures*? In this section, we explore this question via two studies. First, we reassess the effectiveness of existing IMMs by comparing the results when using different measures (§6.1). Second, we further analyze the relationship between different measures using bootstrapping analyses (§6.2). We observe that measures at different granularity levels show better consistency when the models are less stable and vice versa. Moreover, based on our findings, we provide two concrete suggestions for selecting instability measures for future studies (§6.3).

6.1 Reassessing IMMs

To study the relationships between different measures, we reassess existing IMMs from §2.2 and compare their instability scores. We include all measures discussed in §3, except for SVCCA because it did not show good validity in our tests.

Experimental setup For each IMM, we train 10 models using different random seeds, with the same hyper-parameters in §4. We also adopt the IMM hyper-parameters from Zhang et al. (2021): Mixout $p = 0.1$, $\text{WD}_{\text{pre}}\lambda = 0.01$, LLRD $\eta = 0.95$, and Re-init $k = 5$. We compare the IMMs against a baseline (*Standard*), which does not use any IMM.[9]

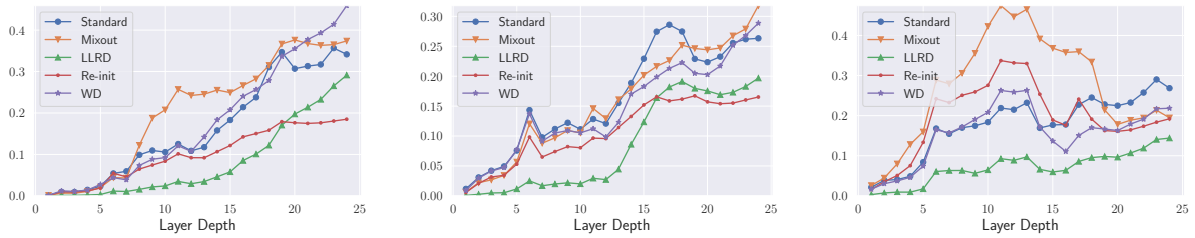
[7] Following Dodge et al. (2020) and Mosbach et al. (2021), we define failed runs as runs whose accuracies at the end of the training $\leq$ a majority classifier (i.e. a classifier that consistently predicts the majority label regardless of the inputs).

[8] To investigate the impact of sample sizes, we also include the results of sample 10% of the test datasets in Appendix D. Despite that the differences between different sub-sampled datasets are larger compared to 50%, we still observe good concurrent validity, especially in lower layers.

[9] Note that our *Standard* baseline is different from the one in Zhang et al. (2021). Ours uses a longer training time (five epochs) and a standard weight decay regularization, because these two choices can be seen as hyper-parameter selections. As observed by Mosbach et al. (2021), these modifications make our baseline stronger.

	RTE				MRPC				CoLA			
	Acc ± SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$	F1 ± SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$	MCC ± SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$
Standard	**71.3** ± 1.8	6.8	13.9	13.8	89.3 ± 1.2	5.1	9.1	9.1	64.5 ± 5.5	4.0	7.9	7.9
Mixout	71.2 ± 3.2	7.9	15.5	15.4	89.6 ± **0.7**	4.8	8.9	8.8	**67.1** ± 1.9	3.6	7.1	7.1
LLRD	69.2 ± 2.8	5.4	13.8	13.7	89.5 ± 1.3	4.0	8.2	8.2	63.9 ± 2.3	**2.8**	**5.3**	**5.3**
Re-init	70.4 ± **1.4**	**4.7**	**10.1**	**10.0**	89.9 ± 0.8	**3.9**	**7.1**	**7.1**	64.2 ± 2.9	3.7	6.8	6.8
WD_{pre}	70.5 ± 5.6	7.4	18.0	17.9	**90.2** ± 1.2	4.7	8.8	8.7	65.6 ± **1.8**	3.6	6.7	6.7

Table 1: Prediction instability scores of BERT after using different IMMs. Higher values indicate higher instability for the instability measures (SD, $\mathcal{I}_{\text{JSD}}$, $\mathcal{I}_{\kappa}$, $\mathcal{I}_{\text{pwd}}$). For better readability, all values are multiplied by 100.



(a) RTE (b) MRPC (c) CoLA

Figure 2: $\mathcal{I}_{\text{CKA}}$ scores of fine-tuned BERT models on all three datasets after applying different IMMs. We observe highly similar trends with RoBERTa and OP. Higher values indicate higher instability.

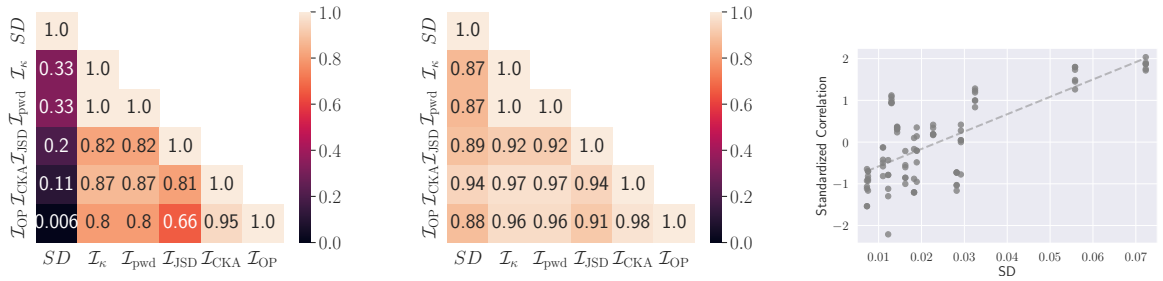
Measures of different granularity levels are not always consistent with each other We show the results of prediction measures and $\mathcal{I}_{\text{CKA}}$ on BERT in Table 1 and Figure 2 (other results show similar trends, see Appendix E). We observe that different measures are not always consistent with each other. For example, when using BERT on MRPC (Table 1), SD shows Mixout to be the most stable training scheme. However, the other three prediction measures and $\mathcal{I}_{\text{CKA}}$ (the top layer, Figure 2b) rank Mixout to be the (second) least stable one.

To better quantify the inconsistency, we calculate Kendall's τ between the rankings of IMMs based on different measures, on each dataset/PLM. We include the full results in Appendix E and make two observations. First, measures of similar granularity level tend to be consistent with each other. For example, $\mathcal{I}_{\text{CKA}} \sim \mathcal{I}_{\text{OP}}$ (both representation measures) and $\mathcal{I}_{\text{pwd}} \sim \mathcal{I}_{\kappa}$ (both based on discrete predictions) show good consistency (i.e. $\tau \geq 0.8$) on each combination of models and datasets. Also, $\mathcal{I}_{\text{pwd}}$ and $\mathcal{I}_{\kappa}$ show better consistency with $\mathcal{I}_{\text{jsd}}$ ($\tau \geq 0.6$) than with SD ($\tau = -0.2$ for BERT on MRPC). Second, the consistency among measures differs across datasets and models. For example, all measures correlate well for BERT on RTE, with a minimum $\tau \approx 0.6$. In contrast, the correlations derived from MRPC are much smaller, with close-to-zero τ values between SD and other measures.

Most IMMs are not always effective Our results also show that most IMMs are not always effective: they sometimes fail to improve stability compared to the *Standard* baseline, which is consistent with the observations of Zhang et al. (2021). In fact, Re-init is the only IMM that consistently improves over *Standard* according to all measures. Also, for BERT on RTE, *Standard* is the third most stable training method according to all prediction (Table 1) and representation (Figure 2) measures. Generally, models trained with WD_{pre} and Mixout are less stable compared to models trained with LLRD and Re-init. Because both WD_{pre} and Mixout aim to stabilize fine-tuning by resolving catastrophic forgetting, these results suggest that catastrophic forgetting may not be the actual or sole cause of instability, which is consistent with the observations of Mosbach et al. (2021).

6.2 Bootstrapping Analyses

In §6.1, we computed Kendall's τ between the rankings of different IMMs obtained using different instability measures. However, because we only have five groups of models (i.e. each group consists of 10 models trained with the same IMM/*Standard* baseline but different random seeds), the results we obtain may be less accurate. To mitigate this issue, in this section we focus on generating more groups for a specific IMM-dataset combination.



(a) *Standard* BERT on MRPC (b) *Standard* BERT on CoLA (c) Correlations on BERT

Figure 3: Figures 3a and 3b show the bootstrapping results for BERT *Standard* baseline on MRPC and CoLA. Figure 3c shows the correlation between 1) the average standardized correlation between each measure and other measures 2) the corresponding SD value, for BERT on each dataset/IMM combination.

Unfortunately, generating more groups is extremely expensive, as each group consists of 10 models trained with different random seeds.[10]

To avoid the high training cost, we instead use **bootstrapping** to *generate more groups of models*. Concretely, for each group of 10 models (i.e. 10 different random seeds), we sample 10 models with replacement for 1000 times to obtain 1000 groups of models. We then compute the Pearson's r between each pair of measures using these groups. We apply the representation measures on the topmost layer to make the results more comparable with the prediction measures.

We show the results for the BERT *Standard* baseline on MRPC and CoLA in Figure 3 and observe similar trends on other datasets/models (see Appendix E). We make two observation. First, consistent with §6.1, we observe that measures at closer granularity levels have higher correlations with each other. For example, SD has correlations of a decreasing strength with other measures on MRPC: from the most similar and coarse-grained Fleiss' Kappa and pairwise disagreement, to the pairwise JSD in between, and finally to the furthest and the most fine-grained representation measures OP and Linear-CKA. Also, the correlations between the two representation measures ($\mathcal{I}_{\text{CKA}}$ and $\mathcal{I}_{\text{OP}}$) are much higher than that between them and other measures. Second, we also observe that correlations obtained from different combinations of dataset/IMM are different from each other, which is expected.

The second observation points to another question: when will different measures be more consistent with each other? Intuitively, *they will be more consistent when the differences between models are large*. In this case, both coarse-grained and fine-grained measures can detect the instability. In contrast, when the differences between models are small, only fine-grained measures can capture these nuances. In other words, *instability measures are more consistent when the models are less stable, and vice versa.*

To quantitatively check this intuition, on each PLM (i.e. BERT and RoBERTa), using the bootstrapping results on each dataset and IMM, we compute the Pearson's r between 1) the average correlations between each measure and other measures and 2) SD values.[11] We observe strong correlations on both BERT ($r = 0.734$, Figure 3c) and RoBERTa ($r = 0.653$, Appendix E), confirming our intuition.

6.3 Implications

In §6.1 and §6.2, we investigated the consistency and differences between different instability measures. Based on our observations, we provide two practical suggestions for future studies.

First, we observed that measures are not always consistent with each other, despite their good validity in §5. This observation suggests that different measures focus on different aspects of instability and therefore should be used together in future studies. Moreover, we observed that different measures tend to be less consistent with each other when the models themselves are more stable. This observa-

[10] For example, the minimum recommended sample size for Pearson's $r = 0.5$ is 99 (Bonett and Wright, 2000). In our case, this means training 990 models (i.e. 99 groups).

[11] Although sharing the same range -1–1, correlations between different measures usually have different scales of values. In other words, some measures are more consistent with other measures, and thus have larger correlations. To balance the weights of different measures, we standardize the correlations for each measure according to the its average correlations with other measures on different datasets/IMMs.

tion further demonstrates the necessity of adopting multiple measures when the instability assessed by one of the measures is low, and that using any measure alone may produce inaccurate conclusions.

Second, we observed measures at similar granularity levels to be more consistent. One can therefore start with SD, and sequentially add more fine-grained measures when previous measures indicate low stability. Because computing fine-grained instability is often slow, only one prediction measure and one representation measure can be used when limited computational resources are available.

7 Conclusion

In this paper, we study measures that quantify the instability of fine-tuning PLMs. In addition to the most commonly used measure, SD, we study six other measures at different granularity levels and propose a framework to evaluate their validity. Using this framework, we show that all these measures except SVCCA have good validity. Moreover, by reassessing existing IMMs, we show that different instability measures are not always consistent, and that they are more consistent when the models are less stable. Finally, based on our observations, we offer two suggestions for selecting instability measures in future studies.

Limitations

Our study leaves room for future work. First, we would like to highlight the difficulty of applying the validity assessment framework from measurement theory to instability measures. For example, in §5.1, our low convergent validity scores may have different interpretations because there are no well-established instability measures. Further, in §5.2, because no previous studies have built theoretical foundations of factors that impact the prediction and representation instability, both our tests do not rigorously follow the concurrent validity definition: our first test of successful and failed runs is based on an assumption derived from observations of Mosbach et al. (2021) rather than theory, and our second test of differences among test datasets examines the consistency between theoretically indistinguishable groups instead of the differences between theoretically distinguishable groups.

Second, we only experimented with a limited number of tasks, instability measures, PLMs, and validity types. Future work can use our framework to further validate the generalizability of our observations. For example, to apply our validity testing framework to larger datasets, to include other measures (e.g. functional similarity measures, Csiszárik et al., 2021 and jitter, Liu et al., 2022), to study generative PLMs (e.g. T5, Raffel et al., 2020 and OPT, Zhang et al., 2022), and to test other types and validity (e.g. discriminative and predictive validity).

Third, we focused on general text classification tasks in this paper. One promising direction is to investigate which measures to use for specific settings. For example, to extend our framework to more recent generative models (e.g. BART, Lewis et al., 2020 and GPT-3, (Brown et al., 2020)). However, in this case, because our prediction measures in §3 are only useful for classification, new prediction measures should be developed, and our tests should be adjusted accordingly.

Acknowledgements

This work is part of the research programme Veni with project number VI.Veni.192.130, which is (partly) financed by the Dutch Research Council (NWO).

References

Srinadh Bhojanapalli, Kimberly Wilber, Andreas Veit, Ankit Singh Rawat, Seungyeon Kim, Aditya Krishna Menon, and Sanjiv Kumar. 2021. On the reproducibility of neural network predictions. *CoRR*, abs/2102.03349.

Douglas G Bonett and Thomas A Wright. 2000. Sample size requirements for estimating pearson, kendall and spearman correlations. *Psychometrika*, 65(1):23–28.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training

via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182, Dublin, Ireland. Association for Computational Linguistics.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. 2021. Similarity and matching of neural network representations. In *Advances in Neural Information Processing Systems*.

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.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. 2021. Grounding representation similarity through statistical testing. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

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

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

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Frederick J Gravetter and Lori-Ann B Forzano. 2018. *Research methods for the behavioral sciences*. Cengage learning.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Hang Hua, Xingjian Li, Dejing Dou, Chengzhong Xu, and Jiebo Luo. 2021. Noise stability regularization for improving BERT fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3229–3241, Online. Association for Computational Linguistics.

Simon Jackman. 2008. 119 Measurement. In *The Oxford Handbook of Political Methodology*. Oxford University Press.

Urja Khurana, Eric Nalisnick, and Antske Fokkens. 2021. How emotionally stable is ALBERT? testing robustness with stochastic weight averaging on a sentiment analysis task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 16–31, Punta Cana, Dominican Republic. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis,

Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuhong Li, Yves Grandvalet, and Franck Davoine. 2018. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2825–2834. PMLR.

Huiting Liu, Avinesh P.V.S, Siddharth Patwardhan, Peter Grasch, and Sachin Agarwal. 2022. Model stability with continuous data updates.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mahdi Milani Fard, Quentin Cormier, Kevin Canini, and Maya Gupta. 2016. Launch and iterate: Reducing prediction churn. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6076–6085. Curran Associates, Inc.

Geoffrey Roeder, Luke Metz, and Durk Kingma. 2021. On linear identifiability of learned representations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9030–9039. PMLR.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress and compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4528–4537. PMLR.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. 2022. The multiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Cecilia Summers and Michael J. Dinneen. 2021. Nondeterminism and instability in neural network optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9913–9922. PMLR.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi

Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Trochim William M. K. 2023. The research methods knowledge base.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chenghao Yang and Xuezhe Ma. 2022. Improving stability of fine-tuning pretrained language models via component-wise gradient norm clipping. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4854–4859, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*.

A Experimental Setup

Running Environment All models are trained using a single NVIDIA RTX 6000 graphics card, with Python 3.7, PyTorch 1.10.1 (Paszke et al., 2019), Hugging Face Transformers 4.14.1 (Wolf et al., 2020), and CUDA 10.2. The total training time is approximately 70 GPU hours. We calculate the results of the instability measures on Intel Xeon E5-2699 CPUs, with Python 3.7 and Numpy 1.19.5 (Harris et al., 2020), taking approximately 24 CPU hours.

	RTE		MRPC		CoLA	
	Dev	Test	Dev	Test	Dev	Test
Positive	59	72	142	137	375	346
Negative	79	67	62	67	146	176

Table 2: Statistics of the test-validation split

Validation/Test Split The GLUE benchmark is one of the most popular benchmarks in NLP research (Wang et al., 2018), which consists of 11 different tasks, including RTE, MRPC, and CoLA. We download and process the datasets using Hugging Face Datasets (Lhoest et al., 2021). Following Zhang et al. (2021), we split the original validation dataset into two parts of (almost) equal sizes, because we have no access to the test data. We then use one part as the new validation data to select checkpoints with the best performance, and we use the other part as the new test data to compute all instability measures. We provide the statistics of the splits in Table 2.

B Details of Fleiss' Kappa

Consider a k-class classification task, m different models, and a test dataset size of n. We denote the number of models which predict the i-th data point as the j-th class as x_{ij}. Clearly, we have $\sum_{j=1}^{k} x_{ij} = m$, because each of the m models will make a prediction on x_i.

We estimate the proportion of *pairs of models* that agree on the i-th data point by

$$p_i = \frac{\sum_{j=1}^{k} C\left(x_{ij}, 2\right)}{C(m,2)} = \frac{\sum_{j=1}^{k} x_{ij}^2 - m}{m(m-1)},$$

where C means the combination. We can then calculate the mean value of p_i as

$$\begin{aligned} p_a &= \frac{1}{n}\sum_{i=1}^{n} p_i \\ &= \frac{1}{mn(m-1)}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} x_{ij}^2 - mn\right]. \end{aligned}$$

Moreover, we estimate the error term as

$$p_\epsilon = \sum_{j=1}^{k}\left(\frac{1}{nm}\sum_{i=1}^{n} x_{ij}\right)^2.$$

After obtaining p_a and p_ϵ, we can calculate Fleiss' Kappa as

$$\kappa = \frac{p_a - p_\epsilon}{1 - p_\epsilon}.$$

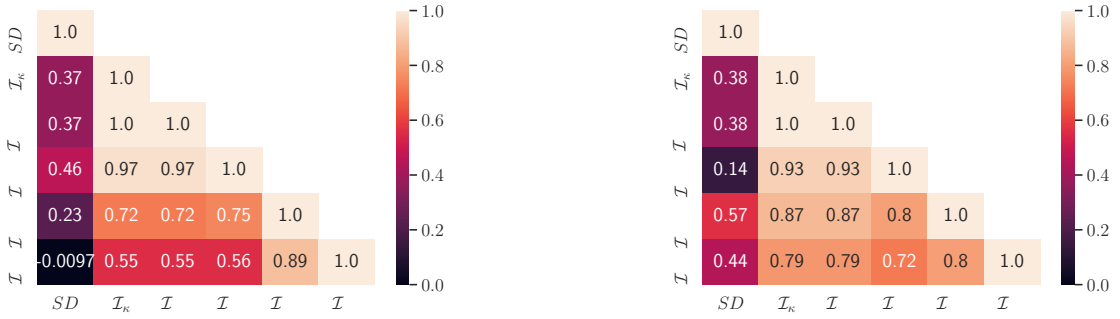
	BERT	RoBERTa
$\mathcal{I}_{\text{CKA}} \sim \mathcal{I}_{\text{OP}}$	0.78	0.94
$\mathcal{I}_{\text{CKA}} \sim \mathcal{I}_{\text{SVCCA}}$	-0.24	0.41
$\mathcal{I}_{\text{OP}} \sim \mathcal{I}_{\text{SVCCA}}$	0.14	0.51
Acc $\pm$ SD	92.6 $\pm$ 0.8	94.4 $\pm$ 0.9
$\mathcal{I}_{\text{JSD}}$	2.3	2.5
$\mathcal{I}_{\kappa}$	4.5	4.3
$\mathcal{I}_{\text{pwd}}$	4.5	4.3

Table 3: Correlations between representation measures, and instability scores computed by different prediction measures, on SST-2.

C Impact of Dataset Size

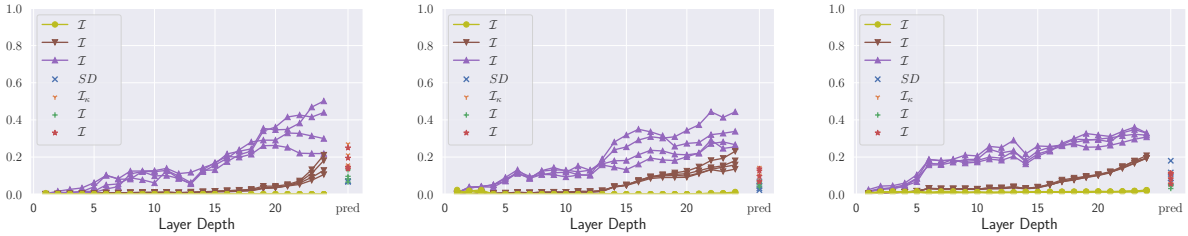
To better understand the impact of using only small train datasets, we perform a preliminary study on SST-2 (Socher et al., 2013), which consists of 67,000 training samples, around eight times larger than the size of CoLA (8,000). We use the same hyper-parameter settings as in Section 4, namely a 16 batch size, a 0.1 Dropout rate, a 2×10^{-5} learning rate, using 20 different random seeds and de-biased Adam, without IMMs.

We computed the instability scores using different prediction measures. We also computed the correlations between representation measures and performed bootstrapping analyses. We show the results in Table 3 and Figure 4. We make three observations. First, as expected, we observe lower instability from models trained on SST-2 compared with models trained on the three small datasets we used in the main text. Second, consistent with our observations in Section 6, we observe that the correlations between different measures on SST-2 are



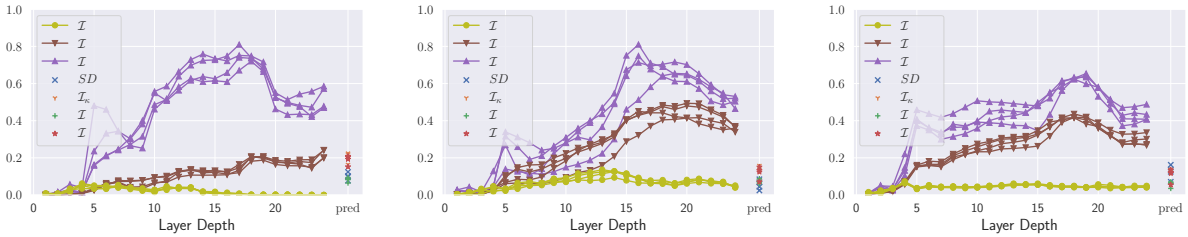
(a) BERT (b) RoBERTa

Figure 4: Bootstrapping analyses on SST-2.



(a) RTE (b) MRPC (c) CoLA

Figure 5: Consistency among sub-samples on BERT, sample rate 0.1.



(a) RTE (b) MRPC (c) CoLA

Figure 6: Consistency among sub-samples on RoBERTa, sample rate 0.1.

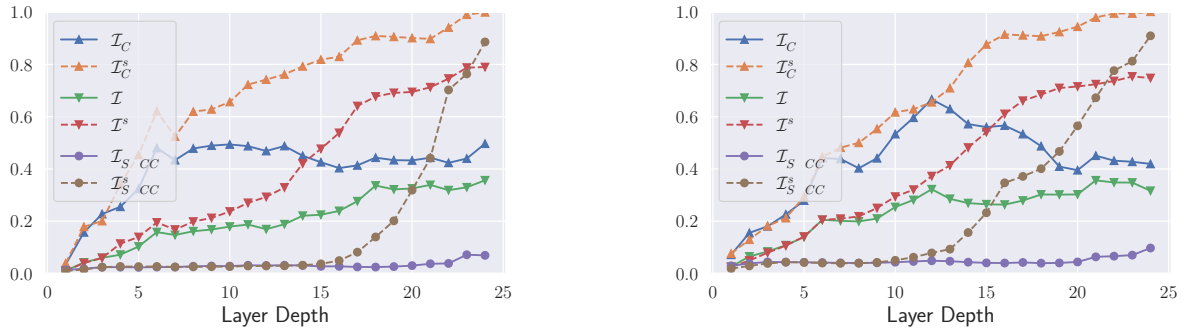
lower, because the models are more stable. Third, also consistent with our observations in Section 6, Figure 4 shows that measures at similar granularity levels are more consistent with each other. Our results on SST-2 suggest that our previous observations are generalizable to larger datasets.

D Impact of Subsample Size

To investigate the impact of sample sizes regarding the differences among different i.i.d. datasets, we also experimented with sampling only 10% of the test samples. We show the results in Figure 5–6. Sampling only 10% of the test samples does bring larger variances (compared with sampling 50%), but results on different samples are mostly still consistent, especially in the lower layers.

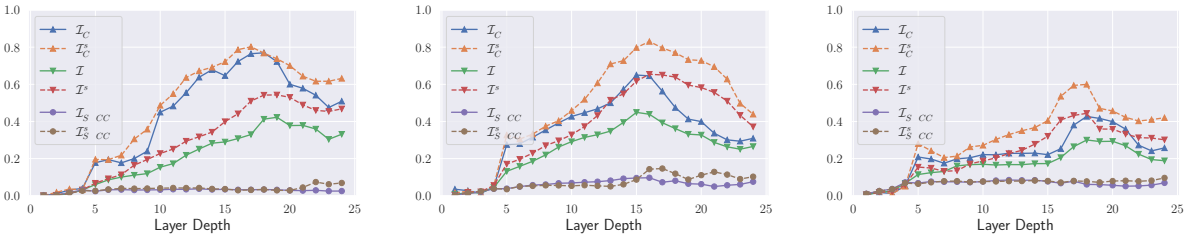
E Additional Results

Figures are on the next page. In Figures 7 – 13, the Y-axis refers to the instability scores computed by different measures.



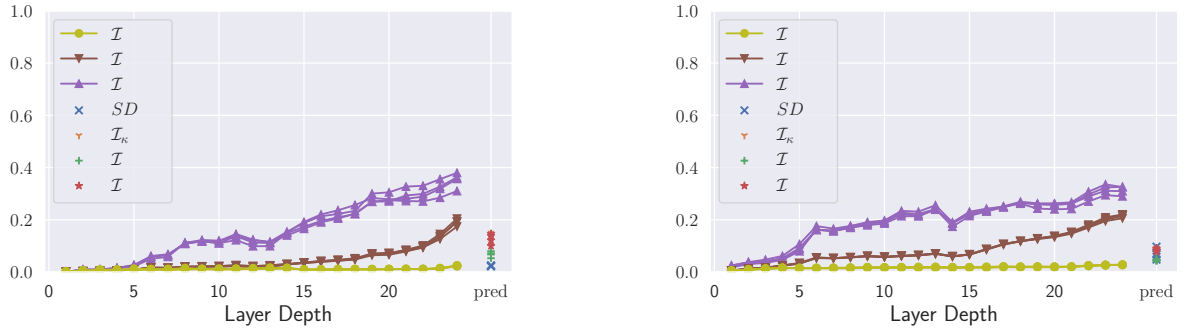
(a) MRPC (b) CoLA

Figure 7: Results of successful vs. failed runs, BERT on MRPC and CoLA.



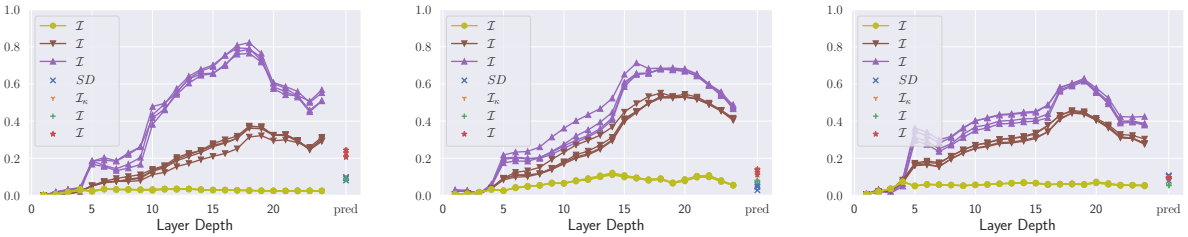
(a) RTE (b) MRPC (c) CoLA

Figure 8: Results of successful vs. failed runs, RoBERTa on MRPC and CoLA.



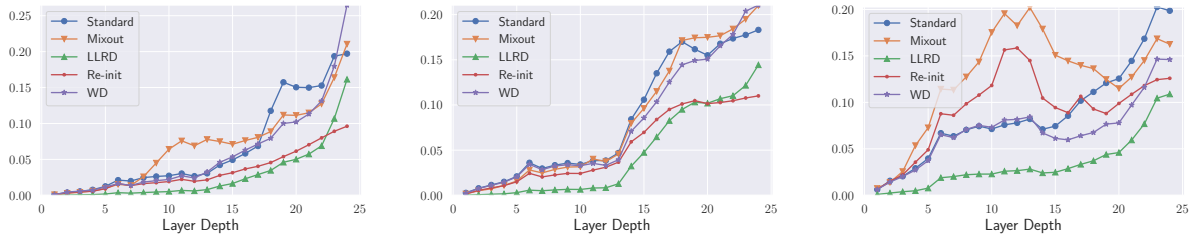
(a) RTE (b) MRPC

Figure 9: Consistency among sub-samples on BERT (on RTE and CoLA), sample rate 0.5.



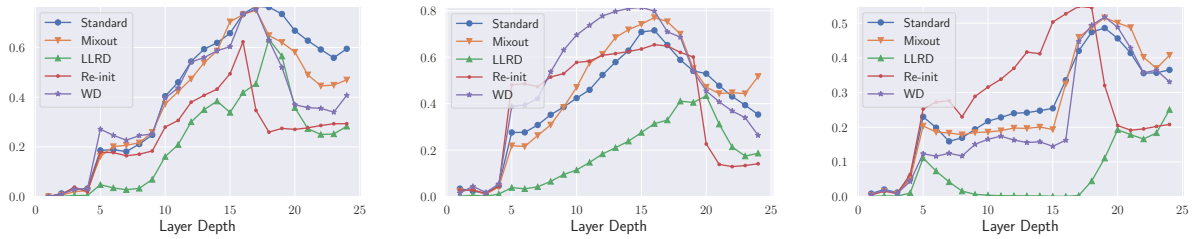
(a) RTE (b) MRPC (c) CoLA

Figure 10: Consistency among sub-samples on RoBERTa, sample rate 0.1.



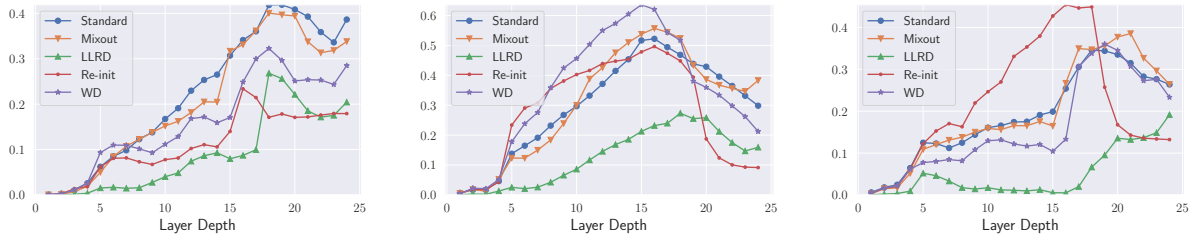
(a) RTE (b) MRPC (c) CoLA

Figure 11: Representation instability for BERT after applying different instability mitigation methods on all three datasets, measured by OP.



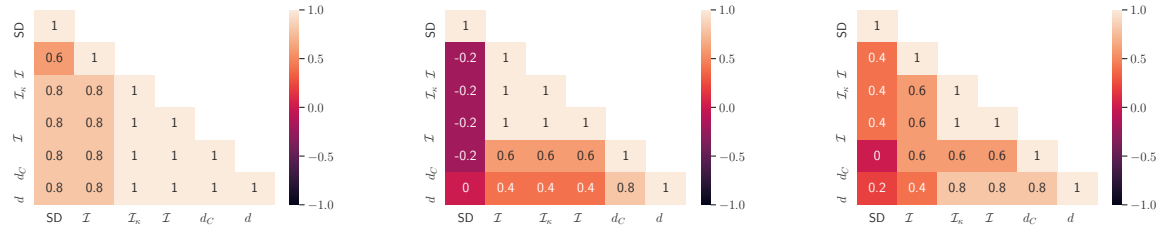
(a) RTE (b) MRPC (c) CoLA

Figure 12: Representation instability for RoBERTa after applying different instability mitigation methods on all three datasets, measured by Linear-CKA.



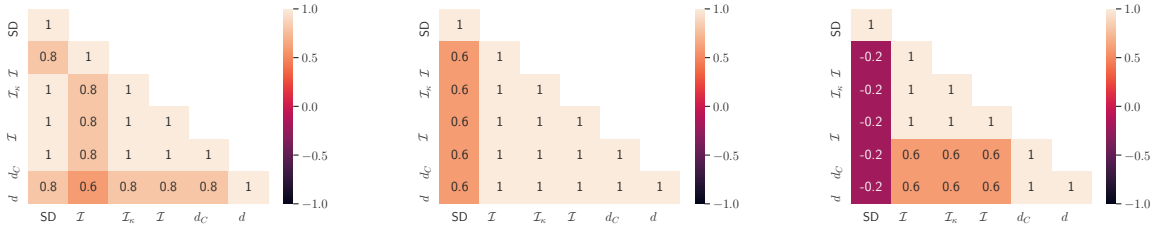
(a) RTE (b) MRPC (c) CoLA

Figure 13: Representation instability for RoBERTa after applying different instability mitigation methods on all three datasets, measured by OP.



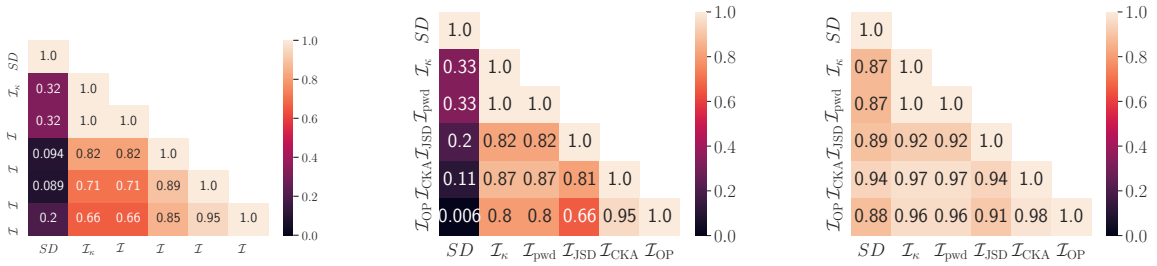
(a) RTE (b) MRPC (c) CoLA

Figure 14: Kendall's τ values after applying different instability mitigation methods on BERT, between each pair of measures. For representation measures, we take the value of the topmost layer.



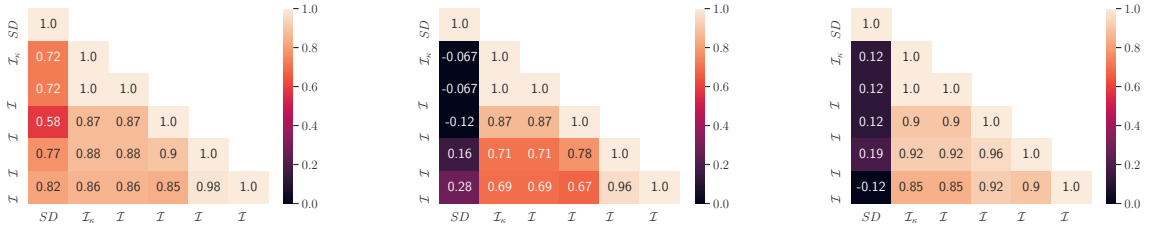
(a) RTE (b) MRPC (c) CoLA

Figure 15: Kendall's τ values after applying different instability mitigation methods on RoBERTa, between each pair of measures. For representation measures, we take the value of the topmost layer.



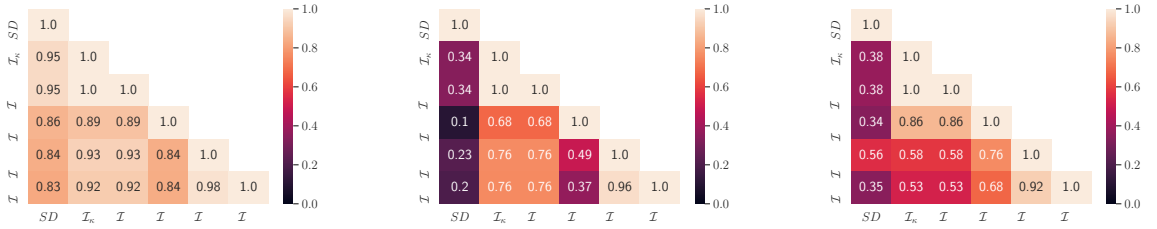
(a) RTE (b) MRPC (c) CoLA

Figure 16: Bootstrapping results of *Standard* BERT.



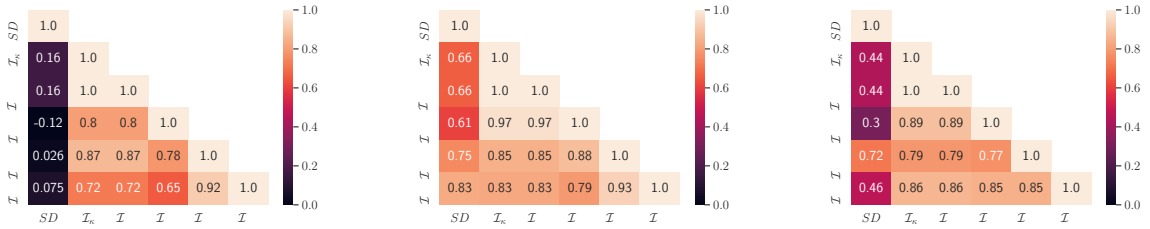
(a) RTE (b) MRPC (c) CoLA

Figure 17: Bootstrapping results of Mixout BERT.



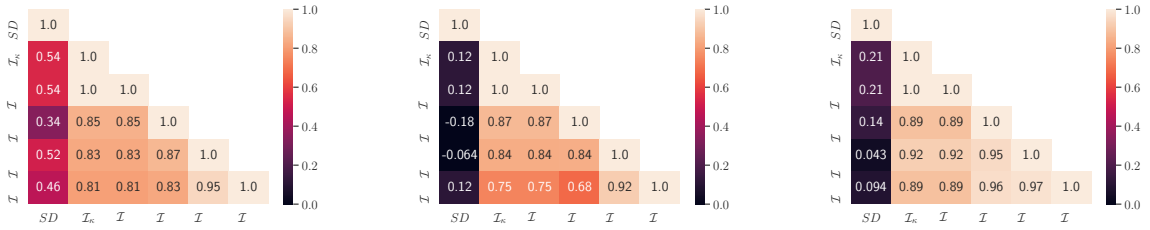
(a) RTE (b) MRPC (c) CoLA

Figure 18: Bootstrapping results of WD_{pre} BERT.



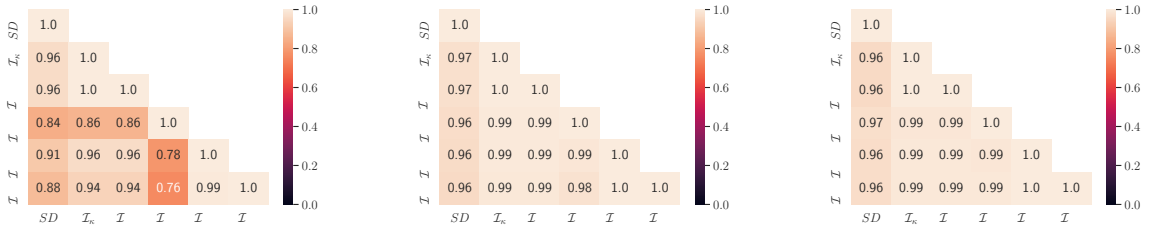
(a) RTE (b) MRPC (c) CoLA

Figure 19: Bootstrapping results of LLRD BERT.



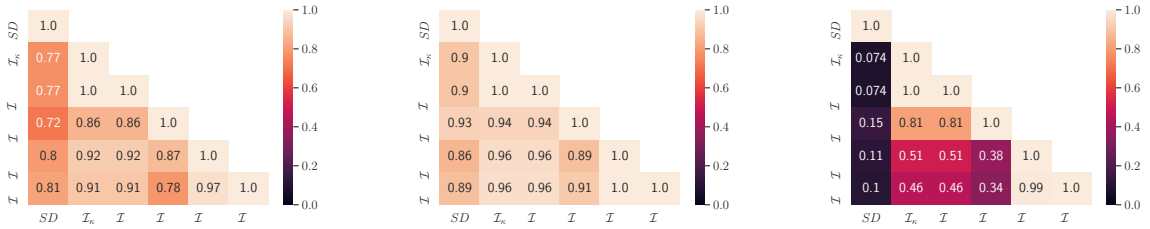
(a) RTE (b) MRPC (c) CoLA

Figure 20: Bootstrapping results of Re-init BERT.



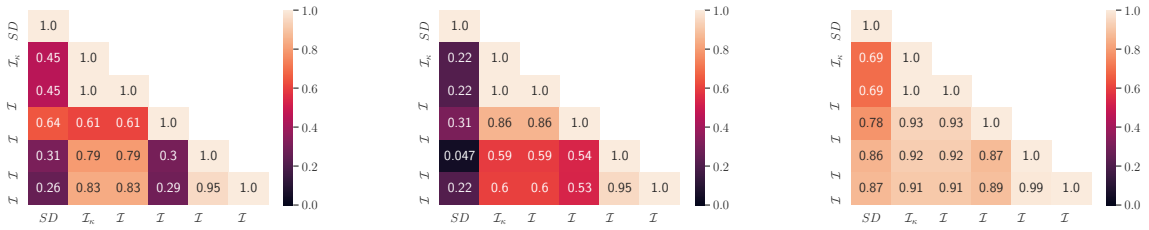
(a) RTE (b) MRPC (c) CoLA

Figure 21: Bootstrapping results of *Standard* RoBERTa.



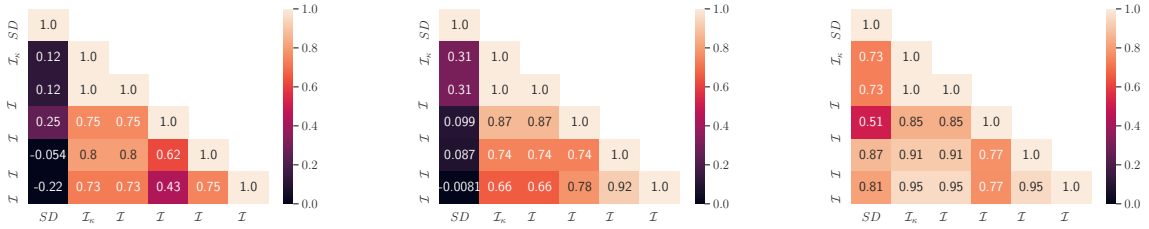
(a) RTE (b) MRPC (c) CoLA

Figure 22: Bootstrapping results of Mixout RoBERTa.



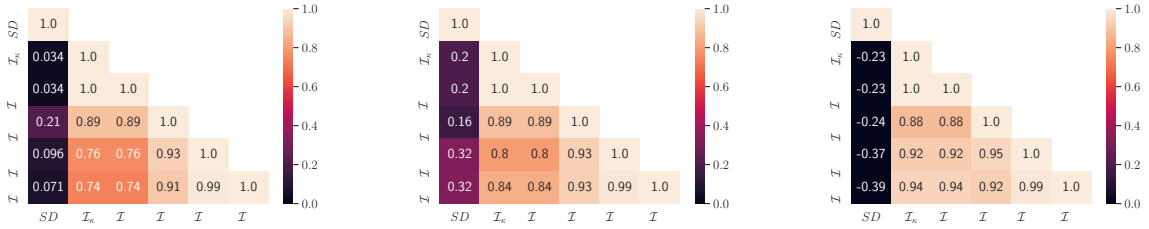
(a) RTE (b) MRPC (c) CoLA

Figure 23: Bootstrapping results of $\mathsf{WD}_{\mathsf{pre}}$ RoBERTa.



(a) RTE (b) MRPC (c) CoLA

Figure 24: Bootstrapping results of LLRD RoBERTa.



(a) RTE (b) MRPC (c) CoLA

Figure 25: Bootstrapping results of Re-init RoBERTa.

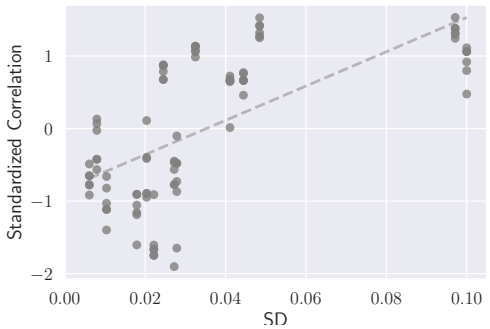


Figure 26: Correlation between 1) the average standardized correlation between each measure and other measures 2) the corresponding SD value, for RoBERTa on each dataset/IMM combination. Pearson's $r = 0.653$.

	RTE				MRPC				CoLA			
	Acc $\pm$ SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$	F1 $\pm$ SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$	MCC $\pm$ SD	$\mathcal{I}_{\text{JSD}}$	$\mathcal{I}_{\kappa}$	$\mathcal{I}_{\text{pwd}}$
Standard	74.4 $\pm$ 12.2	9.8	25.6	25.4	90.8 $\pm$ 3.6	5.9	10.2	10.1	65.6 $\pm$ 7.8	4.9	8.9	8.9
Mixout	79.3 $\pm$ 4.4	9.3	16.7	16.6	89.4 $\pm$ 3.2	6.2	13.2	13.1	68.1 $\pm$ **2.2**	4.6	8.7	8.7
LLRD	**81.3** $\pm$ **1.8**	**5.7**	**11.2**	**11.2**	91.3 $\pm$ **0.6**	3.3	6.2	6.2	**69.7** $\pm$ 4.1	**3.0**	**6.2**	**6.2**
Re-init	79.6 $\pm$ 2.0	7.2	12.7	12.6	**92.5** $\pm$ 0.8	**3.0**	**5.3**	**5.3**	69.2 $\pm$ 2.7	3.8	7.1	7.1
WD_{pre}	**81.3** $\pm$ 2.8	6.6	13.0	12.9	92.0 $\pm$ 1.0	3.6	6.7	6.6	66.6 $\pm$ 2.5	4.4	8.4	8.4

Table 4: Prediction instability scores of RoBERTa after applying different IMMs. To obtain better readability, all values shown here are multiplied by 100. Higher values indicate higher instability for instability measures (i.e. except for performance metrics Acc, F1, and MCC).

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.