Frustratingly Simple Regularization to Improve Zero-shot Cross-lingual Robustness

Anonymous ACL submission

Abstract

Large-scale multilingual pretrained encoders, such as mBERT and XLM-R, have demonstrated impressive zero-shot cross-lingual transfer capability across multiple NLP tasks. However, as we show in this paper, these models suffer from two major problems: (1) degradation in zero-shot cross-lingual performance after fine-tuning on a single language, and (2) cross-lingual performance sensitivity to fine-tuning hyperparameters. In order to address these issues, we evaluate two techniques during fine-tuning, namely, Elastic Weight Consolidation (EWC) and L2-distance regularization to assist the multilingual models in retaining their cross-lingual ability after being fine-tuned on a single language. We compare zero-shot cross-lingual performance of mBERT with/without regularization on four different tasks: XNLI, PANX, UDPOS and PAWSX and demonstrate that the model fine-tuned with L2-distance regularization performs better than its vanilla fine-tuned counterpart in zero-shot setting across all the tasks by up to 1.64%. Moreover, by fine-tuning mBERT with different hyperparameter settings on the specified tasks, we demonstrate that L2-distance regularization also makes fine-tuning more robust, reducing standard deviation of zero-shot results by up to 87%. Based on our experiments, EWC does not provide consistent improvements across languages. Moreover, to test if additional constraint on the encoder parameters would improve the results further, we compared L2-distance regularization with techniques that freeze most of the encoder parameters during fine-tuning, such as bitfit, soft prompting, and adapter-based methods. However, we observe that L2-distance regularization still performs the best.

1 Introduction

In recent years, we have seen multilingual transformer-based encoders, such as mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), XLM-R(Conneau et al., 2019) supporting 100+ languages, could achieve state-of-the-art zero-shot cross-lingual performance across multiple Natural Language Processing (NLP) tasks outperforming almost all previously developed techniques (Hu et al., 2020). Major efforts have been made to make these models stronger by changing the pretraining objectives, adding parallel data (Conneau and Lample, 2019), and pretraining on a larger corpus (Conneau et al., 2019).

Despite their unprecedented success, when these models are fine-tuned on a downstream task on a single language (e.g., English), it is highly likely that the weights which are important for languages other than the source language are overwritten, especially languages which are quite different from the source language (e.g., *Japanese*, *Chinese*, and *Hindi*). One way to avoid this is by having a constraint on the weights of the encoder so that during fine-tuning the weights do not change drastically (a.k.a. *catastrophic forgetting*). We evaluate two techniques to overcome this issue: (1) L2-distance regularization, which constrains new fine-tuned weights to remain close to the original weights using L2-distance (Daumé III, 2007), and (2) Elastic Weight Consolidation (EWC) which imposes the same L2-distance constraints but weight them using the Fisher information matrix (F) (Kirkpatrick et al., 2017) which acts as an indicator of the importance of different model weights.

We extensively evaluate the specified techniques (by performing 500+ experiments) on four different NLP tasks including natural language inference (XNLI dataset), name entity recognition (Wikiann dataset), part of speech tagging (UDPOS dataset), and paraphrase detection (PAWSX dataset). We demonstrate that L2-distance regularization can improve cross-lingual zero-shot performance by up to 1.64%. EWC also improves zero-shot cross-lingual performance for some languages but overall it under-performs when compared to vanilla fine-tuning. Nevertheless, we show that the Fréchet distance between the Fisher estimates of different languages (needed for EWC) can be a good indicator of an encoder's cross-lingual capability. In particular, the Fréchet distance between English and other languages shows a significant negative correlation with the downstream tasks' performance suggesting that languages which are quite different from English would have lower zero-shot scores and vice-versa.

Motivated by the performance improvement using L2 distance regularization, we decided to experiment with techniques that go one step further and either freeze most of the encoder parameters during fine-tuning, such as bitfit (Zaken et al., 2021), or only train some of the additional parameters, such as soft prompts in the encoder embeddings (Lester et al., 2021) or adapter layers in the existing transformer layers (Houlsby et al., 2019). We tested all three techniques on XNLI and UDPOS tasks. For XNLI, adapter-based model outperformed vanilla fine-tuning but not L2-distance regularization when averaged over three runs. For UDPOS, all three techniques achieved sub-par performance as compared to vanilla fine-tuning and L2-distance regularization.

Another issue with BERT-style encoders is their instability and sensitivity to the initial seed as well as fine-tuning hyperparameters when tuned for different downstream tasks (Zhang et al., 2020; Dodge et al., 2020; Mosbach et al., 2020). In this work, we show that the zero-shot cross-lingual performance of multilingual models on languages other than the source language is quite sensitive to the fine-tuning hyperparameters. *So much so that the same validation score on the source language results in very different zero-shot test scores on a target language.* For example, when mBERT is trained on Wikiann English training data with vanilla fine-tuning using five different hyperparameter settings, we observe a low standard deviation on English test set F1 score (85.29 ± 0.20). However, the average zero-shot performance on other languages exhibits 6 times higher standard deviation (60.99 ± 1.26). We demonstrate that L2-distance regularization is effective in overcoming this instability issue. In particular, it reduces standard deviation in zero-shot performance by at least 60% compared to vanilla fine-tuning. In summary, our main contributions are as follows:

- We compare L2-distance regularization and EWC during fine-tuning multilingual encoders and demonstrate that fine-tuning with L2-distance regularization consistently outperforms the vanilla fine-tuning for multiple NLP tasks (especially for zero-shot performance).
- We show that the zero-shot cross-lingual performance of multilingual models are quite sensitive to the fine-tuning hyperparameters. However, we demonstrate that L2-distance regularization makes cross-lingual models more robust to hyperparameter changes during fine-tuning and makes them more efficient.
- We show that L2-distance regularization achieves better zero-shot cross-lingual performance as compared to even more constraint techniques that freeze most of the encoder parameters, such as bitfit, soft prompting, and adapters.

2 Related Works

There has been a major improvement in unsupervised cross-lingual transfer learning methods in the recent years thanks to introduction of large-scale multilingual transformer encoders such as mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-R (Conneau et al., 2019). These models are able to achieve state-of-the-art zero-shot cross-lingual performance across multiple NLP tasks (Hu et al., 2020), such as cross-lingual dialogue systems (Lin et al., 2020), part of speech tagging (Zeman et al., 2018) and natural language inference (Conneau et al., 2018).

BERT-style encoders have been demonstrated to be very sensitive to hyperparameters and the random seed during fine-tuning (Dodge et al., 2020; Zhang et al., 2020). These studies focus mainly on the sensitivity of monolingual models. Hence, such sensitivity analysis has not been done for cross-lingual models and in particular in zero-shot settings. Several efforts have been made to improve cross-lingual capability of multilingual encoders by using parallel-data (Conneau and Lample, 2019) and larger corpora (Conneau et al., 2019) and to investigate their effectiveness in cross lingual settings (Pires et al., 2019; Kudugunta et al., 2019). However, no significant efforts have been made towards improving the fine-tuning process of these encoders for better zero-shot performance and more robust fine-tuning.

Catasrophic forgetting has been widely studied before for continual learning. These studies suggest multiple approaches to prevent such a forgetting including L2-distance (Daumé III, 2007), EWC (Kirkpatrick et al., 2017), Variational Continual Learning (Nguyen et al., 2018), and Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017). L2-distance and EWC use regularization over weights during training of the second task to retain the knowledge required for good performance on the previous task. GEM on the other hand poses continual learning as a constrained objective function on the previous task's loss to make sure that it does not increase while learning a new task. Variational Continual Learning tries to retain the distribution over model parameters but this approach becomes computationally expensive for large models.

Despite extensive use of these methods in continual learning context, their application in cross-lingual transfer learning has not been explored before. In this work, we mainly focus on EWC and L2-distance since other methods such as GEM require extra model parameters and increase the computational complexity of fine-tuning by a significant amount.

3 Background

Multilingual transformer-based language models (LM) such as mBERT and XLM-R are trained via unsupervised training objectives such as Masked Language Modeling (MLM) on the mix of data extracted from Wikipedia (or other resources) for multiple languages. We refer to the dataset used for each language as D_{lg} where lg denotes the language. The model weights after pretraining are denoted by θ^*. The task-specific loss during fine-tuning is denoted by ℓ_t. We describe the proposed regularization techniques in the following subsections.

3.1 L2-distance Regularization (L2)

In L2-distance regularization, the total loss (L_{L2}) is computed as the summation of the task loss (ℓ_t) and L2-distance between the model's weights (θ) during fine-tuning and the initial pretrained weights (θ^*), as can be seen in Eq. 1. The L2 distance helps constraining the model's weights to not change drastically during the fine-tuning to potentially maintain cross-lingual capability of the encoder.

$$L_{L2} = \ell_t + \frac{\lambda}{2} \sum_i (\theta_i - \theta_i^*)^2 \qquad (1)$$

3.2 Elastic Weight Consolidation (EWC)

Elastic Weight Consolidation (EWC) penalty is similar to L2-distance penalty with the difference that each of the model's weights has an "importance" factor for a particular language that needs to be determined prior to fine-tuning. In particular, the importance of the i^{th} parameter is given by $F_{lg,i}$ where $F_{lg,i}$ is the i^{th} element in the diagonal Fisher information matrix calculated for a particular language (lg). The total loss is therefore computed as:

$$L_{EWC} = \ell_t + \frac{\lambda}{2} \sum_i F_{lg,i} (\theta_i - \theta_i^*)^2 \qquad (2)$$

EWC utilizes the Laplacian approximation to estimate the posterior distribution of θ given data of a downstream task for the *source language* and the unlabeled data of the *target language* used during pretraining. In our setup, this translates into $\log P(\theta|D_{lg}, D_t) \propto \log P(D_t|\theta) + \log P(\theta|D_{lg})$ where D_{lg} refers to the data of language lg which is used for pretraining the model and D_t refers to the downstream task's data. The first term corresponds to ℓ_t in Eq. 2 and the second term is the EWC penalty. Intuitively, the term $\log P(\theta|D_{lg})$ denotes information about the weights θ in the context of the unlabeled data of a particular language (D_{lg}). Kirkpatrick et al. (2017) remarks that the Fisher information refers to the importance of a particular model weights for the previous task, which is the unsupervised pretraining on the target language in our setup.

In (Kirkpatrick et al., 2017), $\log P(\theta|D_{lg})$ is approximated using Laplace approximation (MacKay, 1992). In particular, $\log P(\theta|D_{lg})$ is approximated by a Gaussian distribution with mean θ^* and diagonal precision F_{lg}. In this work, we use Wikipedia corpus of a target language to estimate the Fisher information matrix associated with that language.

4 Experimentation Setup

In this section, we describe the experimental setup as well as the downstream tasks used to test the efficacy of different regularization methods described in the previous section. For our experiments, we mainly use mBERT. We also performed our experiments on XLM-R Large for which the results are provided in Appendix H.

4.1 Downstream Tasks

We pick four different tasks from XTREME benchmark (Hu et al., 2020) to evaluate the zero-shot cross-lingual performance across both sequence and token-level classification tasks.

XNLI: The Cross-lingual Natural Language Inference corpus (Conneau et al., 2018) is a quite famous NLP task which asks for the relationship between a premise statement and a hypothesis statement which could be *entailment, contradiction* or *neutral*. For XNLI, the crowdsourced English data is translated to ten other languages with the help of professional translators and is used for evaluation. For training, the original English data of MultiNLI (Williams et al., 2017) is used.

Wikiann: The named entity recognition (NER) dataset consists of automatically tagged `LOC`, `PER`, and `ORG` tags. These tags were automatically tagged using a combination of knowledge base properties, cross- lingual and anchor links, self-training, and data selection. In adherence to the XTREME benchmark, we also used the balanced train, dev, and test splits (Rahimi et al., 2019) provided in the IOB2 format.

UDPOS: The POS tagging dataset from Universal Dependencies v2.5 treebanks (Nivre, 2018). For each word in a provided sentence, one of the 17 universal POS tags are provided. Similar to other datasets, the model is trained on English training data and is evaluated on the test set of the target languages.

PAWS-X: The Cross-lingual Paraphrase Adversaries from Word Scrambling dataset (Yang et al., 2019). Two sentences are provided and the model has to sepcify whether the sentences are paraphrases or not. A subset of the PAWS (Zhang et al., 2019) dev and test set is translated into six other languages with the help of professional translators and is used for evaluation. The original training set of PAWS (Zhang et al., 2019) is used for training.

4.2 Experimental Setup for Zero-shot Performance Comparison

To assess the zero-shot performance of mBERT on each of the above described downstream tasks, the model is trained on English training data and is then evaluated on other languages' test sets. For fine-tuning, three different setups are used: (1) Vanilla with no regularization on model's weights, (2) L2-distance regularization (see Eq. 1), and (3) EWC (see Eq. 2).

For EWC, the Fisher matrix is computed using the Wikipedia corpus of the target language and used during fine-tuning of the model (for each language separately). For example, for testing the zero-shot performance of the model on *fr*, we fine-tune the model on *en* data with Fisher weights computed using *fr* Wikipedia corpus. Hence, to evaluate zero-shot performance of the model with EWC, we fine-tune the model for each target language separately (with a different loss function). Appendix F provides other ways of computing Fisher weight penalties for EWC fine-tuning (none of the variants performed better that our approach explained here). An illustration of the complete fine-tuning procedure with EWC is illustrated in Appendix A.

For each setup, the model is trained with three different seeds using the best hyperparameters. Their performance averaged over these three runs is then compared against each other. We selected a set of 14 languages for which the zero-shot performance of the models is compared. Out of these 14 languages: fr, bg, hi, ja, and zh are selected by the authors and de, es, sw, vi, ar, el, ru, ur, and tr are randomly selected. The first set of five languages was handpicked to have a diverse set where some languages share some of the language structure with *English (en)* and some which do not.

4.3 Experimental Setup to Assess Cross-lingual Sensitivity

Transformer based language models, such as BERT have shown great performance on multiple NLP tasks and have set new state-of-the-art scores. However, it has been demonstrated that these models' performance are sensitive to the change in fine-tuning hyperparameters for different downstream tasks (Zhang et al., 2020; Dodge et al., 2020) (in monolingual settings).

Here, we similarly demonstrate the sensitivity of multilingual models (e.g., mBERT) to fine-tuning hyperparameters and in particular in zero-shot cross-lingual performance. To assess if L2-distance regularization can help stabilizing the zero-shot cross-lingual performance of multilingual models, we setup an experiment where each model is run with multiple set of hyperparameters with/without regularization and then compare their average zero-shot performance and their standard deviation over all the languages. In the experiments, learning rate, training epochs, and training warmup steps

Model	Fine-tuning	XNLI (Acc.)	Wikiann (F1)	UDPOS (F1)	PAWSX (Acc.)
Cross-lingual zero-shot transfer					
mBERT	Vanilla	64.39	61.72	70.13	82.26
	w/ EWC	63.16 (-1.23)	59.84 (-1.88)	67.80 (-2.33)	78.84 (-3.42)
	w/ L2	**66.03 (+1.64)**	**62.85 (+1.13)**	**70.53 (+0.40)**	**82.55 (+0.29)**

Table 1: Average zero-shot cross-lingual performance using different fine-tuning setups. In all cases, the model is fine-tuned only on the English training data. Each task performance is averaged over the selected 14 languages over 3 runs with different seeds. XNLI test set does not include Japanese. Therefore, the average zero-shot performance is over 13 languages for XNLI. Similarly, average zero-shot performance is calculated for other tasks.

	Fine-tuning	en	ar	bg	de	el	es	fr	
mBERT	vanilla	81.51	63.46	66.89	69.18	65.22	73.67	72.43	
	w/ EWC	81.45	62.54	64.88	67.87	64.16	71.83	74.18	
	w/ L2	**82.36**	**65.20**	**68.81**	**70.82**	**66.82**	**74.60**	**74.25**	
	Fine-tuning	hi	ru	sw	tr	ur	vi	zh	Avg.
mBERT	vanilla	58.73	67.76	47.23	58.25	56.67	69.90	67.64	64.39
	w/ EWC	56.41	64.33	48.09	59.79	54.72	65.70	66.61	63.16
	w/ L2	**60.30**	**68.77**	**50.55**	**61.39**	**58.09**	**70.25**	**68.59**	**66.03**

Table 2: Zero-shot cross-lingual Performance for XNLI task on the selected languages when the model is fine-tuned on English training data using different fine-tuning setups. XNLI test set does not include Japanese.

are changed. We performed the sensitivity experiments over *all available languages* for each task. The five different fine-tuning hyperparameter settings used in these experiments are provided in Appendix C.

5 Experimental Results

In this section, we present the key findings from the experiments described in the previous section.

5.1 L2-Distance Regularization Improves Cross-lingual Zero-shot Performance

As discussed in Section 4.2, for each task the model is trained on the English data and is evaluated on the selected 14 languages test sets (i.e., zero-shot). The overall zero-shot performance of each model is calculated by averaging the zero-shot scores over all languages. For each task, the performance is averaged over three runs with the best hyperparameters setting. The results are provided in Table 1.

XNLI Given that this is a classification task, the accuracy metric is used for the models. As can be seen in Table 1, fine-tuning with L2-distance regularization performs the best overall and improves vanilla fine-tuning by 1.6% whereas when fine-tuned with EWC the model performance deteriorated as compared to the vanilla training technique. Language specific zero-shot performances are shown in Table 2. Fine-tuning w/ EWC helped in improving the performance for certain languages such as *fr*, *tr* and *sw* whereas *fine-tuning w/ L2 consistently performs better for all the languages than vanilla fine-tuning and w/ EWC.*

Wikiann For this task, the $F1$ metric is used since it is an entity recognition task. As observed in Table 1, for Wikiann also fine-tuning w/ L2 performs better than other fine-tuning techniques. Fine-tuning w/ L2 improves over vanilla training by 1.1%. If we look at language specific performances (Appendix B), fine-tuning w/ EWC performs the best for *fr* and performs marginally better than vanilla fine-tuning for *ar* and *el*. Fine-tuning w/ L2 performs the best for all languages, except *fr*.

UDPOS For this task, as well, F1 metric is used for performance comparison. The performance with L2-distance regularization improves the zero-shot cross-lingual performance of the model by 0.40%. Vanilla fine-tuning is still better than fine-tuning with EWC. We can observe from language specific zero-shot performances (refer Appendix B) that fine-tuning w/ L2-distance regularization performs the best for majority of the languages but for some languages, such as *hi, ja* and *ur*, vanilla fine-tuning performs better. Fine-tuning w/ EWC is consistently outperformed by other two techniques.

PAWSX For this task, accuracy is used as an evaluation metric. As observed in Table 1, fine-tuning with L2-distance regularization improves the zero-shot cross-lingual performance of the model by 0.3% whereas when fine-tuned with EWC the model performance decreases by 4.0% compared to the vanilla fine-tuning. The zero-shot performance for each language in provided in Appendix B. The model fine-tuned w/ L2-distance regularization performed better than vanilla for all languages except *es* and *fr*.

Overall, fine-tuning with L2-distance regularization outperforms vanilla fine-tuning across all four tasks. EWC performs better than vanilla for some languages but generally deteriorates the overall performance. As described in Section 4.2, EWC regularization requires the model to be fine-tuned for each target language separately. Whereas, fine-tuning with L2-distance regularization is required only once, which is another advantage of using L2-distance regularization over EWC. Language-specific results for Wikiann, UDPOS and PAWSX tasks are provided in Appendix B. We also experimented with an extreme hyperparameter setting, where we increased the learning rate significantly, and L2-distance regularization outperformed vanilla finetuning on zero-shot performance by at least 10% for each task, task-specific results are provided in Appendix E.

We repeated the experiments using XLM-R Large model (Conneau et al., 2019) and observed similar performance gains when using L2-distance regularization. When fine-tuned with L2-distance regularization, the model outperformed vanilla fine-tuning and the reported state-of-the-art results (Hu et al., 2020). All the results for XLM-R Large are provided in Appendix H.

We also studied the relationship between the zero-shot performance and the L2-distance penalty (λ in Eq. 1) which is presented in Fig. 1. We trained mBERT using different L2-distance penalty over 6 different seeds and plotted the average zero-shot performance improvement over vanilla fine-tuning along with the performance on English test set. The figure suggests that fine-tuning multilingual models can benefit from L2-distance regularization as long as it provides the model some flexibility to learn the downstream task while constraining it enough to retain the language information learnt during the pretraining. Even the performance for English test set is better with lower L2-distance regularization.

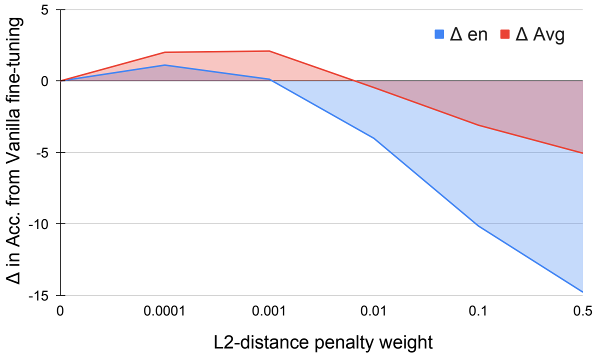

Figure 1: The relationship between the average zero-shot performance (percentage improvement over vanilla fine-tuning denoted by Δ Avg) and the L2-distance penalty weight for the XNLI task.

5.2 L2-distance Regularization Improves Cross Lingual Sensitivity

As discussed in Section 4.3, to study the models sensitivity to hyperparameters, we fine-tune them with five different hyperparameter settings and evaluate them on *all available languages for each task*. On the basis of results from our zero-shot experiments (Section 5.1), we focus the sensitivity experiments on vanilla fine-tuning and fine-tuning w/ L2-distance regularization. Results are provided in Table 3.

Comparing vanilla fine-tuning performance on English to zero-shot shows that in case of multilingual encoders even though the performance on English might be consistent (small standard deviation), the few-shot performance varies a lot (higher standard deviation) with the change in hyperparameters. This demonstrates that the issue of sensitivity to hyperparameters is even more critical in the case of multilingual encoders. However, as can be seen in Table 3, L2-distance regularization is effective in improving this instability issue in zero-shot performance as well.

XNLI When fine-tuned with L2-distance regularization, the performance both on English as well as overall zero-shot performance improves while the standard deviation is significantly dropped. The overall zero-shot performance improved by 2%

Model	Fine-tuning.	Test	XNLI (Acc. ± Std)	Wikiann (F1 ± Std)	UDPOS (F1 ± Std)	PAWSX (Acc. ± Std)
mBERT	Vanilla	*en*	81.60 ± 0.64	85.29 ± 0.20	95.58 ± 0.06	94.15 ± 0.59
	w/ L2	*en*	82.36 ± 0.13	84.72 ± 0.13	95.40 ± 0.02	94.25 ± 0.19
	Vanilla	zero-shot	63.22 ± 1.58	60.99 ± 1.26	71.26 ± 0.27	82.92 ± 1.42
	w/ L2	zero-shot	65.13 ± 0.20	61.87 ± 0.51	72.52 ± 0.11	83.11 ± 0.86

Table 3: Average performance over different fine-tuning hyperparamters. For zero-shot performance, scores on each task is the average over all available languages and the standard deviation is computed over average zero-shot scores over five runs.

		en	ar	bg	de	el	
mBERT	Vanilla	81.6 ± 0.64	63.54 ± 1.34	66.92 ± 0.86	69.29 ± 1.7	65.47 ± 1.68	
	w/ L2	82.36 ± 0.13	65.08 ± 0.54	68.55 ± 0.39	71.19 ± 0.32	66.88 ± 0.44	
		es	fr	hi	ru	sw	
mBERT	Vanilla	73.39 ± 1.29	72.88 ± 1.39	58.62 ± 1.06	67.72 ± 1.2	47.8 ± 2.29	
	w/ L2	74.7 ± 0.29	74.28 ± 0.61	60.44 ± 0.5	68.75 ± 0.31	50.9 ± 0.88	
		th	tr	ur	vi	zh	Average
mBERT	Vanilla	47.65 ± 4.64	58.24 ± 2.12	56.46 ± 1.11	69.2 ± 1.01	67.81 ± 1.22	63.22 ± 1.58
	w/ L2	53.27 ± 0.98	60.92 ± 0.66	58.07 ± 0.61	69.93 ± 0.62	68.78 ± 0.25	65.13 ± 0.2

Table 4: Average performance and its standard deviation for each language for XNLI task over five different hyperparameter settings.

while its standard deviation decreased by 87% compared to vanilla fine-tuning. Table 4 provides the mean and standard deviations of the zero-shot performances on all languages for XNLI task. It can be seen that the standard deviation for all languages is reduced when the model is fine-tuned with L2-distance regularization.

Wikiann We observe that the standard deviation for overall zero-shot performance is almost 6 times the standard deviation of test performance on English for vanilla fine-tuning. L2-distance regularization helps improving the overall zero-shot performance while reducing the standard deviation. The overall zero-shot performance is improved by 1% while fine-tuning with L2-distance regularization. The standard deviation on English reduces by 35% while the standard deviation on overall zero-shot performance reduced by 60%.

UDPOS Similar to other tasks, not only the zero-shot performance is improved with L2-distance regularization but also the standard deviation is reduced by a significant margin. The overall zero-shot performance improved by 1.26% while reducing the standard deviation for English test performance by one-third and reducing the standard deviation for overall zero-shot performance by 59%.

PAWSX When using L2-distance regularization, the zero-shot performance improves by 0.2% while its standard deviation is reduced by 40% compared to vanilla fine-tuning. The performance on English also improves slightly while its standard deviation is reduced by 67%.

Overall, L2-distance regularization makes the model more robust to fine-tuning hyperparameters. It improves the overall performance while significantly reducing the standard deviation of the results for both source language and target languages in zero-shot setting by as much as 87%. This makes L2-distance regularization an ideal choice for fine-tuning multi-lingual models for downstream tasks. Detailed language-specific results for Wikiann, UDPOS and PAWSX tasks are provided in Appendix D.

5.3 Fréchet Distance as an Indicator of Encoder's Zero-shot Capability

Fréchet distance (Kirkpatrick et al., 2017), as defined in Eq. 3, can be used to calculate the distance between Fisher matrices of *en* (F_{en}) and another target language (F_{lg}). Where $\hat{F_{en}}$ and $\hat{F_{lg}}$ are the normalized versions of the Fisher matrices.

$$d^2(\hat{F_{en}}, \hat{F_{lg}}) = \frac{1}{2}\text{tr}(\hat{F_{en}} + \hat{F_{lg}} - 2(\hat{F_{en}}, \hat{F_{lg}})^{\frac{1}{2}}) \quad (3)$$

For each task, we calculated the correlation between the Fréchet distance of target languages (lg) from en and their zero-shot performance. For each language, we calculated two set of Fisher matrices using the Wikipedia corpora and XNLI test set.

Corpus	XNLI	Wikiann	UDPOS	PAWSX
Correlation b/w Fréchet distance and Task perf.				
Wiki	-0.677	-0.248	0.190	-0.541
XNLI	-0.814	-0.811	-0.305	-0.953

Table 5: The correlation between Fréchet distances of the source (*en*) and target (*lg*) language's Fisher matrices and zero-shot performance using mBERT.

The negative correlations, shown in Table 5, suggest that larger the Fréchet distance between *en* and *lg*, weaker the zero-shot performance on *lg* when the model is fine-tuned on English training data for a specific task. Hence, Fréchet distance can be used as an approximate indicator of a multi-lingual encoder's cross lingual capability.

6 Comparison with Frozen Encoder Techniques

Motivated by the performance improvement observed by using L2 distance regularization, we decided to experiment with techniques that go even further and freeze most of the encoder parameters during downstream fine-tuning, such as bitfit (Zaken et al., 2021), soft prompting (Lester et al., 2021) and adapters (Houlsby et al., 2019). We tested these techniques on XNLI and UDPOS. For our experiments, we implemented common adapter layers (Houlsby et al., 2019) for all languages as compared to MAD-X (Pfeiffer et al., 2020) which requires language specific adapter layers and is more suited for zero-shot performance when the languages are unseen by the encoder during pre-training.

The performance of different fine-tuning techniques on English test set and overall zero-shot performance, averaged over three runs, is provided in Table 6. For XNLI task, bitfit and soft-prompts based method resulted in sub-par overall zero-short performance as compared to vanilla and L2-distance based fine-tuning. Adapter based fine-tuning outperformed vanilla but not L2-distance regularization. For UDPOS, all three techniques under-performed when compared to vanilla and L2-distance based fine-tuning. All three fine-tuning techniques consistently achieved lower performance on the English test set. This suggests that freezing most of the encoder parameters can result in better zero-shot performance in some cases compared to Vanilla. However, it performs slightly worse on English test set, which is expected given the extreme constraints. Hence, it seems like L2-distance regularization provides the best of both worlds. All the language specific zero-shot performances for both XNLI and UDPOS and the best-hyperparameters for soft-prompts, bitfit and adapter are provided in Appendix G.

	en (test set)		*zero-shot*	
Fine-tuning	XNLI (Acc.)	UDPOS (F1)	XNLI (Acc.)	UDPOS (F1)
Vanilla	81.51	**95.43**	64.39	70.13
w/ L2	**82.36**	95.39	**66.03**	**70.53**
Frozen Encoder Techniques				
w/ soft prompts	73.2	92.45	60.52	65.37
w/ bitfit	76.33	94.80	64.03	69.26
w/ adapter	80.65	95.20	64.96	68.37

Table 6: Performance on *en* test set and average zero-shot cross-lingual performance using different frozen encoder fine-tuning techniques.

7 Conclusion

In this paper, we rigorously compared L2-distance regularization and EWC during multilingual models' fine-tuning and demonstrated that L2-distance regularization outperforms EWC across multiple tasks and languages in improving zero-shot cross-lingual performance. We showed that Fisher information matrices can be used to approximately indicate the cross-lingual capability of a multilingual encoder before fine-tuning it for downstream tasks. We also show that L2-distance regularization outperforms techniques that tend to freeze most of their encoder parameters during training, such as bitfit, soft-prompting and adapters based methods.

Moreover, we demonstrated that the zero-shot cross-lingual performance of multilingual models is quite sensitive to the fine-tuning hyperparameters. However, we showed that using L2-distance regularization during fine-tuning not only improves the zero-shot cross-lingual performance of the model, but also makes it more robust to hyperparameter choices. Based on the results of this paper, we recommend the use of L2-distance regularization during fine-tuning of multilingual models to obtain the best and most robust performance.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proc. NeurIPS'19*, pages 7059–7069.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proc. EMNLP'18*, pages 2475–2485.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proc. ACL'07*, pages 256–263.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL'19*, pages 4171–4186.

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv:2003.11080*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proc. of the National Academy of Sciences*, 114(13):3521–3526.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual nmt representations at scale. In *Proc. EMNLP-IJCNLP'19*, pages 1565–1575.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Zhaojiang Lin, Zihan Liu, Genta Indra Winata, Samuel Cahyawijaya, Andrea Madotto, Yejin Bang, Etsuko Ishii, and Pascale Fung. 2020. Xpersona: Evaluating multilingual personalized chatbot. *arXiv:2003.07568*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Proc. NeurIPS'17*, pages 6467–6476.

David JC MacKay. 1992. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

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

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. 2018. Variational continual learning. In *Proc. ICLR'18*.

Abrams M. Agic Z. Ahrenberg L. Antonsen L. Aranzabe M. J. Arutie G. Asahara M. Ateyah L. Attia M. et al. Nivre, J. 2018. Universal dependencies 2.2. *arXiv:1902.00193*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proc. ACL'19*, pages 4996–5001.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for ner. *arXiv:1902.00193*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv:1704.05426*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. In *Proc. EMNLP'19*, pages 3678–3683.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proc. CoNLL'18*, pages 1–21.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv:2006.05987*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *Proc. NAACL'19*, pages 1298–1308.

A Illustration of Fine-tuning using EWC

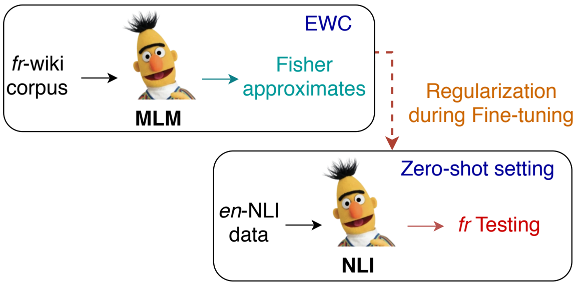

Figure 2: Steps for fine-tuning mBERT with EWC loss. For example, in XNLI task the model is trained on English NLI data with additional EWC loss with Fisher weights estimated from French Wikipedia corpus (see Eq. 2). The model is then evaluated in zero-shot settings on French NLI test data.

B Per Language Zero-shot Results

Language specific zero-shot performances using different fine-tuning setups for Wikiann, UDPOS, and PAWS-X are provided in Tables 7, 8, and 9, respectively.

	Training Tech.	en	fr	bg	de	es	sw	hi	vi
mBERT	Vanilla	84.30	79.10	76.87	78.21	73.68	69.43	63.78	71.34
	w/ EWC	83.96	**79.93**	75.93	77.85	73.20	65.80	62.30	67.58
	w/ L2	**84.78**	79.64	**78.06**	**78.60**	**74.50**	**70.67**	**65.93**	**72.32**
	Training Tech.	ja	zh	ar	el	ru	ur	tr	Avg.
mBERT	Vanilla	27.29	41.65	39.82	70.96	65.00	34.84	72.10	61.72
	w/ EWC	21.40	39.77	39.90	71.09	64.51	26.48	72.04	59.84
	w/ L2	**28.08**	**43.20**	**41.03**	**72.39**	**65.87**	**36.10**	**73.47**	**62.85**

Table 7: Zero-shot cross-lingual Performance for Wikiann on the selected languages when the model is fine-tuned on English training data using different fine-tuning setups.

	Fine-tuning	en	fr	bg	de	es	hi	vi	
mBERT	vanilla	**95.43**	82.96	85.23	85.45	86.40	**65.05**	53.67	
	ewc	95.36	82.68	84.50	84.46	85.22	60.67	52.40	
	l2	95.39	**85.39**	**85.95**	**86.03**	**87.41**	63.38	**54.45**	
	Fine-tuning	ja	zh	ar	el	ru	ur	tr	Avg.
mBERT	vanilla	**45.47**	61.63	52.92	81.23	85.78	**56.87**	68.98	70.13
	ewc	43.07	60.28	52.55	79.08	83.46	46.90	66.10	67.80
	l2	43.27	**62.49**	**53.48**	**82.76**	**86.43**	55.72	**70.17**	**70.53**

Table 8: Zero-shot cross-lingual Performance for UDPOS on the selected languages when the model is fine-tuned on English training data using different fine-tuning setups.

	Fine-tuning	en	de	es	fr	ja	zh	Avg.
mBERT	Vanilla	93.68	85.49	**87.96**	**87.56**	72.76	77.54	82.26
	w/ EWC	93.02	80.99	84.07	85.87	69.35	73.96	78.85
	w/ L2	**94.00**	**86.24**	87.77	87.37	**73.56**	**77.79**	**82.55**

Table 9: Zero-shot cross-lingual Performance for PAWS-X on the selected languages when the model is fine-tuned on English training data using different fine-tuning setups.

C Hyper-parameters for Sensitivity Experiments

Different fine-tuning hyper-parameter settings used for sensitivity analysis in Section 5.2 for all 4 tasks are provided in Table 10.

Wikiann				UDPOS			
Fine-tuning	LR	Warmup steps	Epochs	Fine-tuning	LR	Warmup steps	Epochs
Vanilla	3e-5	0	15	Vanilla	2e-5	0	10
	2e-5	200	15		2e-5	200	15
	3e-5	0	10		3e-5	0	10
	4e-5	500	15		4e-5	500	15
	4e-5	0	10		4e-5	0	10
w/ L2	3e-5	0	15	w/ L2	2e-5	0	15
	2e-5	200	15		2e-5	200	15
	3e-5	0	10		3e-5	0	10
	4e-5	500	15		4e-5	500	15
	4e-5	0	10		4e-5	0	10
XNLI				PANX			
Fine-tuning	LR	Warmup steps	Epochs	Fine-tuning	LR	Warmup steps	Epochs
Vanilla	5e-5	3600	3	Vanilla	2e-5	0	5
	4e-5	5000	3		2e-5	200	8
	2e-5	5000	4		3e-5	0	5
	7e-5	1500	3		4e-5	500	8
	5e-5	0	3		4e-5	0	5
w/ L2	5e-5	4000	3	w/ L2	2e-5	0	5
	4e-5	5000	3		2e-5	200	8
	2e-5	5000	4		3e-5	0	5
	7e-5	1500	3		4e-5	500	8
	5e-5	0	3		4e-5	0	5

Table 10: Different fine-tuning hyper-parameter settings for Wikiann, UDPOS, XNLI and PANX use for sensitivity analysis.

D Per Language Sensitivity Results

Language specific sensitivity experiments comparing vanilla fine-tuning to fine-tuning with L2-distance regularization for Wikiann, UDPOS, and PAWSX are provided in Tables 11, 12, and 13, respectively.

E Extreme Hyperparameter Sensitivity Results

We also experimented with an extreme hyperparameter setting where we set the learning rate to 1e-4 for all four tasks during fine-tuning. This extreme hyperparameter setting was selected to be quite different from the best set of hyperparameters. The models fine-tuned with L2-distance regularization significantly outperformed the vanilla fine-tuned models. This further demonstrates that fine-tuning with L2-distance regularization is more stable and helps the model to learn the downstream task even in extreme hyperparameter settings.

F EWC Variants

We evaluated two more ways of applying EWC regularization during fine-tuning: (1) Applying multiple EWC penalties along with the task loss, and (2) applying the EWC penalty using the Fisher estimates from a mixed corpus of source and target language. For the first approach, we considered *fr* and *en* penalties together which did not improve the zero-shot performance on French test set compared to using just *fr*

		ar	he	vi	id	jv	ms
mBERT	Vanilla	40.2 ± 1.92	55.04 ± 1.18	70.9 ± 1.12	63.2 ± 5.85	62.4 ± 5.35	71.58 ± 1.14
	w/ L2	42.33 ± 1.96	56.42 ± 0.25	72.19 ± 0.46	61.63 ± 3.13	65.75 ± 1.55	69.98 ± 1.27
		tl	eu	ml	ta	te	af
mBERT	Vanilla	75.7 ± 0.56	62.85 ± 2.18	48.41 ± 2.43	52.16 ± 2.46	46.32 ± 2.02	76.05 ± 0.97
	w/ L2	74.53 ± 1.31	64.31 ± 3.52	54.47 ± 1.18	55.07 ± 0.21	48.82 ± 0.88	77.26 ± 0.18
		nl	en	de	el	bn	hi
mBERT	Vanilla	81.7 ± 0.62	85.29 ± 0.2	77.75 ± 0.46	70.88 ± 1.63	70.14 ± 1.41	63.24 ± 1.43
	w/ L2	82.35 ± 0.28	84.72 ± 0.13	78.28 ± 0.24	71.6 ± 0.82	71.6 ± 0.82	65.75 ± 0.74
		mr	ur	fa	fr	it	pt
mBERT	Vanilla	54.84 ± 2.25	35.38 ± 4.27	41.17 ± 3.68	79.42 ± 0.81	80.94 ± 0.32	80.39 ± 0.8
	w/ L2	57.96 ± 0.55	31.99 ± 1.76	40.27 ± 3.16	79.96 ± 0.4	81.64 ± 0.27	80.3 ± 0.46
		es	bg	ru	ja	ka	ko
mBERT	Vanilla	74.71 ± 2.98	78.07 ± 0.87	65.38 ± 0.56	27.58 ± 0.94	63.82 ± 1.57	59.14 ± 1.3
	w/ L2	74.6 ± 2.6	79 ± 0.46	66.32 ± 0.51	27.5 ± 0.6	66.24 ± 0.82	60.56 ± 0.94
		th	sw	yo	my	zh	kk
mBERT	Vanilla	0.55 ± 0.25	67.86 ± 0.67	39.48 ± 6.47	49.23 ± 2.4	41.8 ± 1.6	47.82 ± 3.44
	w/ L2	0.37 ± 0.11	68.38 ± 2.82	42.56 ± 1.49	50.6 ± 2.45	40.68 ± 0.76	46.13 ± 1.99
		tr	et	fi	hu	Average	
mBERT	Vanilla	73.4 ± 1.34	75.57 ± 1.1	76.83 ± 0.74	76.67 ± 1.25	60.99 ± 1.26	
	w/ L2	74.33 ± 0.65	77.32 ± 0.52	77.64 ± 0.16	76.28 ± 0.83	61.87 ± 0.51	

Table 11: Average Zero-shot performance and its standard deviation on each language for Wikiann over five different hyperparameter settings.

		af	ar	bg	de	el	en
mBERT	Vanilla	85.27 ± 0.15	45.93 ± 0.45	84.33 ± 0.27	85.49 ± 0.35	79.43 ± 0.28	95.58 ± 0.06
	w/ L2	87.05 ± 0.34	46.26 ± 0.3	86.15 ± 0.24	86.31 ± 0.35	82.36 ± 0.73	95.4 ± 0.02
		es	et	eu	fa	fi	fr
mBERT	Vanilla	86.37 ± 0.25	80.13 ± 0.63	59.77 ± 0.39	64.49 ± 0.33	76.67 ± 0.54	84.63 ± 0.85
	w/ L2	88.01 ± 0.15	82.31 ± 0.12	61.8 ± 0.7	67.49 ± 0.35	78.5 ± 0.24	86.53 ± 0.32
		he	hi	hu	id	it	ja
mBERT	Vanilla	58.77 ± 0.81	63.59 ± 1.61	76.7 ± 0.33	80.01 ± 0.33	87.37 ± 0.76	45.46 ± 1.14
	w/ L2	59.77 ± 0.43	60.72 ± 0.62	77.92 ± 0.28	80.54 ± 0.33	88.96 ± 0.33	45.93 ± 1.01
		ko	mr	nl	pt	ru	ta
mBERT	Vanilla	48.39 ± 0.36	66.31 ± 1.32	89.28 ± 0.25	87.37 ± 0.42	85.4 ± 0.11	64.03 ± 0.67
	w/ L2	49.26 ± 0.12	68.47 ± 0.29	89.88 ± 0.11	88.51 ± 0.38	86.71 ± 0.49	64.68 ± 0.51
		te	tr	ur	vi	zh	Average
mBERT	Vanilla	77.67 ± 1.42	67.82 ± 0.88	55.05 ± 1.36	51.8 ± 0.48	57.68 ± 0.34	71.26 ± 0.27
	w/ L2	80.53 ± 0.76	69.36 ± 0.1	54.33 ± 0.96	52.75 ± 0.1	59.57 ± 0.43	72.52 ± 0.11

Table 12: Average Zero-shot performance and its standard deviation on each language for UDPOS over five different hyperparameter settings.

		en	de	es	fr	ja	zh	Avg.
mBERT	Vanilla	94.15 ± 0.59	86.1 ± 1.06	88.33 ± 1.04	87.73 ± 0.81	74.28 ± 2.41	78.14 ± 1.97	82.92 ± 1.42
	w/ L2	94.25 ± 0.19	86.38 ± 0.62	88.65 ± 0.7	87.82 ± 0.97	74.31 ± 1.82	78.39 ± 1	83.11 ± 0.86

Table 13: Average Zero-shot performance and its standard deviation on each language for PAWSX over five different hyperparameter settings.

		XNLI	WIkiann	UDPOS	PAWSX
Cross-lingual zero shot performance					
mBERT	Vanilla	33.33	44.36	59.56	55.21
	w/ L2	43.78	60.05	70.8	78.92

Table 14: Average zero-shot performance on each task when the model is fine-tuned using extremely different hyperparameters compared to the best hyperparameter setting.

penalty during EWC fine-tuning. For the second approach, we mixed English and French Wikipedia data and estimated the Fisher matrix using the mixed corpus. This approach also did not perform better than using EWC with just *fr* Fisher estimates. The results using both of these approaches are provided in Table 15.

Model	Fine-tuning	en	fr
mBERT	Vanilla	81.51	72.43
	w/ EWC (*fr*)	81.45	74.18
	w/ EWC (*fr+en*)	82.25	73.65
	w/ EWC (*mix fr-en*)	82.21	73.13
	w/ L2	**82.36**	**74.25**

Table 15: Zero-shot cross-lingual performance for XNLI using vanilla fine-tuning, with L2-distance regularization, and different EWC regularization variants.

G Other Frozen Encoder Techniques

The best hyper-parameters for all three techniques are provided in Table 16. The results for zero-shot cross lingual performance for the XNLI task are provided in Table 17 and UDPOS in Table 18.

		# Prompts	LR	Epochs	Size	Seeds
XNLI	soft prompts	20	0.01	3	-	42,10,20
	bitfit		0.001	3	-	42,10,20
	adapter		0.001	3	64	42,10,20
		# Prompts	LR	Epochs	Size	Seeds
UDPOS	soft prompts	20	0.001	15	-	42,10,20
	bitfit	-	0.001	15	-	42,10,20
	adapter	-	0.001	15	64	42,10,20

Table 16: Best hyperparameters for bitfit, soft prompting and adapters for XNLI and UDPOS.

		en	ar	bg	de	el	es	fr	hi	ru	sw	tr	ur	vi	zh	Avg.
mBERT	soft prompts (Lester et al., 2021)	73.2	59.75	63.62	62.89	61	65.55	66.47	56.18	62	48	59.31	54.76	64.3	62.97	60.52
	bitfit (Zaken et al., 2021)	76.33	63.13	66.28	68.07	63.98	71.58	69.87	**59.36**	66	49.43	**61.44**	57.87	67.81	67.52	64.03
	adapter (Houlsby et al., 2019)	**80.65**	**64.28**	**67.45**	**69.59**	**66.32**	**72.84**	**72.24**	58.84	**67.24**	**49.74**	61.39	**57.9**	**68.83**	**67.87**	**64.96**

Table 17: Zero-shot cross lingual performance for XNLI task on the selected languages when the model is fine-tuned using soft prompting, bitfit technique and using adapters.

	Fine-tuning	en	fr	bg	de	es	hi	vi	ja	zh	ar	el	ru	ur	tr	Avg.
	soft prompts (Lester et al., 2021)	92.45	73.36	79.08	80.44	79.66	61.59	50.75	49.74	57.68	56.31	72.93	79.46	49.51	59.25	65.37
	bitfit (Zaken et al., 2021)	94.80	85.66	86.60	85.22	87.25	56.85	51.97	43.10	62.58	53.95	84.26	87.39	48.52	67.06	69.26
mBERT	adapter (Houlsby et al., 2019)	95.20	80.40	84.04	84.60	84.15	64.02	50.05	46.39	60.73	52.93	78.48	85.09	54.66	63.27	68.37

Table 18: Zero-shot cross lingual performance for UDPOS task on the selected languages when the model is fine-tuned using soft prompting, bitfit technique and using adapters.

H Experimental Results using XLM-R Large

We repeated the L2-distance regularization experiments with XLM-R Large since it provides the best zero-shot cross-lingual performance as reported by (Hu et al., 2020). These experiments were run on a single seed. The results for XNLI, Wikiann, UDPOS, and PAWS are provided in Tables 19, 20, 21, and 22, respectively. The results are consistent with our earlier findings indicating fine-tuning with L2-distance regularization performs better than the vanilla fine-tuning. For UDPOS, however, the average zero-shot performance for vanilla and L2-distance regularized fine-tuning are quite close to each other.

		en	ar	bg	de	el	es	fr	hi	ru	sw	tr	ur	vi	zh	Avg.
XLM-R$_{\text{Large}}$	(Hu et al., 2020)	**88.70**	77.20	83	**82.5**	80.8	83.7	82.2	75.6	79.1	71.2	78.00	71.7	**79.30**	78.2	78.65
	Vanilla	86.21	73.77	80.56	78.08	77.52	80.86	80.06	71.46	77.33	57.49	70.96	66.71	76.85	76.73	74.49
	w/ L2	88.34	**78.20**	**83.01**	82.18	**81.78**	**84.13**	**83.05**	**75.87**	**79.56**	**71.26**	**78.6**	**71.78**	79.16	**78.84**	**79.03**

Table 19: Zero-shot cross-lingual performance on XNLI using vanilla fine-tuning, w/ L2-distance regularization, and as reported by (Hu et al., 2020)

		en	fr	bg	de	es	sw	hi	vi	ja	zh	ar	el	ru	ur	tr	Avg.
XLM-R$_{\text{Large}}$	(Hu et al., 2020)	**84.70**	**80.5**	81.4	78.8	**79.60**	70.5	**73.00**	**79.4**	**23.20**	**33.10**	53.00	79.5	69.1	56.4	76.1	66.69
	Vanilla	84.55	79.88	81.91	79.3	70.85	71.06	71.1	78.97	20.23	28.93	49.47	78.79	70.39	53.97	77.98	65.2
	w/ L2	84.09	80.36	**84.97**	**79.46**	71.01	**71.21**	72.06	76.86	20.63	28.18	**57.69**	**80.39**	**73.65**	**69.18**	**80.29**	**67.57**

Table 20: Zero-shot cross-lingual performance on Wikiann using vanilla fine-tuning, w/ L2-distance regularization, and as reported by (Hu et al., 2020).

		en	fr	bg	de	es	hi	vi	ja	zh	ar	el	ru	ur	tr	Avg.
XLM-R$_{\text{Large}}$	(Hu et al., 2020)	96.10	45.1	88.1	88.5	88.3	**76.40**	56.8	15.9	25.7	**67.50**	86.3	89.5	**70.30**	76.3	67.28
	Vanilla	**96.16**	45.18	**88.64**	**88.59**	89.29	74.00	**57.25**	36.54	**46.64**	67.14	86.62	89.74	65.43	75.85	**70.07**
	w/ L2	96.13	**45.43**	88.2	88.58	**89.65**	73.9	56.28	**31.13**	38.62	67.01	**86.96**	**89.99**	70.00	**76.74**	69.42

Table 21: Zero-shot cross-lingual performance on UDPOS using vanilla fine-tuning, w/ L2-distance regularization, and as reported by (Hu et al., 2020)

		en	de	es	fr	ja	zh	Avg.
XLM-R$_{\text{Large}}$	(Hu et al., 2020)	94.7	89.7	90.1	90.4	78.7	82.3	86.24
	Vanilla	**95.45**	89.99	89.64	**91.85**	**81.49**	83.49	87.29
	w/ L2	95.10	**90.65**	**90.65**	91.60	81.29	**83.69**	**87.57**

Table 22: Zero-shot cross-lingual performance on PAWSX using vanilla fine-tuning, w/ L2-distance regularization, and as reported by (Hu et al., 2020).