DEFT-X: Denoised Sparse Fine-Tuning for Zero-Shot Cross-Lingual Transfer

Anonymous ACL submission

Abstract

Effective cross-lingual transfer remains a critical challenge in scaling the benefits of large language models from high-resource to low-resource languages. Towards this goal, prior studies have explored many approaches to combine *task knowledge* from task-specific data in a (high-resource) source language and *language knowledge* from unlabeled text in a (low-resource) target language. One notable approach proposed composable sparse fine-tuning (SFT) for cross-lingual transfer that learns task-specific and language-specific sparse masks to select a subset of the pretrained model's parameters that are further fine-tuned. These sparse fine-tuned vectors (SFTs) are subsequently composed with the pretrained model to facilitate zero-shot cross-lingual transfer to a task in a target language, using only task-specific data from a source language. These sparse masks for SFTs were identified using a simple magnitude-based pruning. In our work, we introduce DEFT-X, a novel composable SFT approach that denoises the weight matrices of a pretrained model before magnitude pruning using singular value decomposition, thus yielding more robust SFTs. We evaluate DEFT-X on a diverse set of extremely low-resource languages for sentiment classification (NusaX) and natural language inference (AmericasNLI) and demonstrate that it performs at par or outperforms SFT and other prominent cross-lingual transfer baselines.

1 Introduction

Pretrained language models (LMs) are the de-facto choice for NLP, achieving state-of-the-art results across diverse benchmarks. However, effectively adapting these models to specific tasks remains a challenge owing to their large model sizes and the substantial training costs incurred during full fine-tuning. Furthermore, full fine-tuning approaches are prone to issues such as catastrophic forgetting and negative interference when adapted to multiple tasks. To mitigate these challenges, parameter-efficient fine-tuning (PEFT) techniques are a popular choice (Pfeiffer et al., 2024). These approaches include sparse fine-tuning (SFT) that refers to identifying a sparse subnetwork of the full model to train, and adapter-based methods that insert additional trainable modules while keeping the original model parameters fixed (Pfeiffer et al., 2020; Hu et al., 2022).

Multilingual NLP introduces an additional layer of complexity, especially in the context of low-resource languages. A key objective of recent work in multilingual NLP has been to facilitate cross-lingual transfer by leveraging high-resource language data to improve performance on low-resource languages. Zero-shot cross-lingual transfer refers to the more constrained setting of having access only to task-specific labeled data in a high-resource (source) language and no labeled data in a low-resource (target) language. PEFT-based approaches such as MAD-X (Pfeiffer et al., 2020) and LT-SFT (Ansell et al., 2022) are designed to support zero-shot cross-lingual transfer. In MAD-X, a task adapter is learned using labeled data in a source language and a language adapter is learned using unlabeled data in a target language. To achieve zero-shot cross-lingual transfer, the task adapter in the source language is combined with the language adapter in the target language. LT-SFT (Lottery Ticket Sparse Fine-Tuning) removes the need for new trainable modules as in adapters by first identifying sparse subnetworks (a.k.a. "lottery tickets") within the model using magnitude pruning. Fine-tuning these subnetworks while keeping the rest of the model frozen yields sparse vectors; these vectors can be estimated independently to learn task-specific and language-specific knowledge. These vectors are composed via simple addition to obtain the final model for zero-shot cross-lingual transfer.

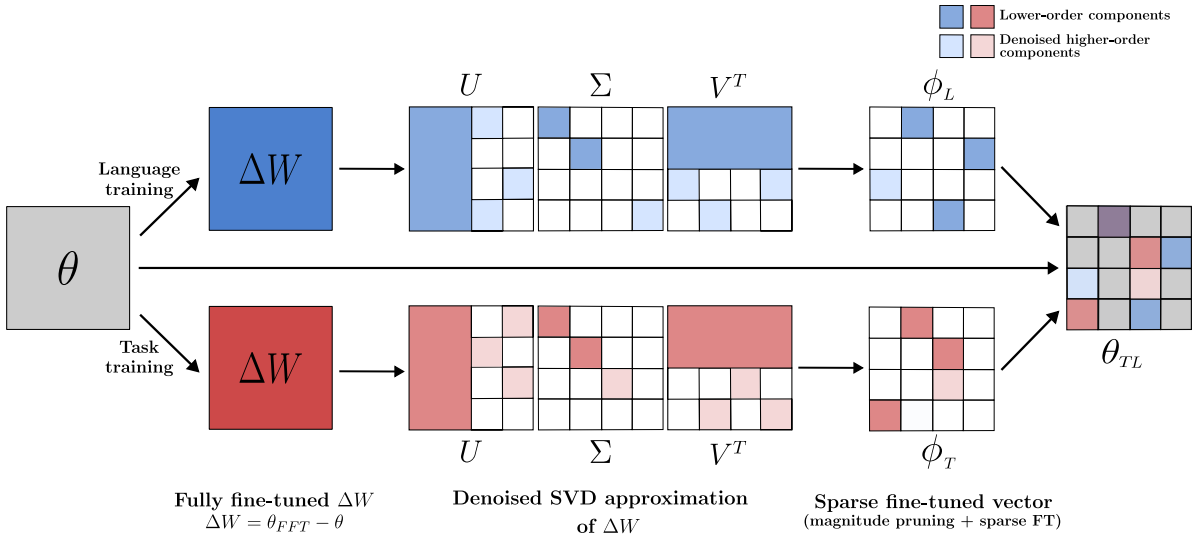

Figure 1: A graphical representation of DEFT-X. The pretrained model θ (gray, left) undergoes full fine-tuning to obtain θ_{FFT}. The difference ΔW (blue and red, left) captures the magnitude difference between θ and θ_{FFT}. Each weight matrix in ΔW is denoised by pruning higher-order components (i.e., lower singular value components) while retaining lower-order components (i.e., high singular value components). The denoised ΔW is then magnitude-pruned and sparsely fine-tuned to produce ϕ. Finally, the language-specific component ϕ_L and task-specific component ϕ_T are combined via addition to form the target language-task model θ_{TL}(left).

In this work, we propose DEFT-X, a novel sparse composable approach for zero-shot cross-lingual transfer. Adopting the LT-SFT template, DEFT-X aims to identify subnetworks for further finetuning using a low-rank approximation and improves the quality of sparse fine-tuned vectors by denoising higher-order components. Specifically, we use Singular Value Decomposition (SVD) (Sharma et al., 2023; Zhao et al., 2025) to decompose each weight matrix into lower and higher-order components (corresponding to high and low singular values, respectively). The higher-order components are denoised and added to the SVD lower-order components, resulting in a signal-amplified weight matrix (refer to §3.1). Next, we apply magnitude pruning to identify a more effective sparse subnetwork by removing noise, followed by fine-tuning to obtain the final sparse fine-tuned vectors (detailed in §3.1). These vectors are finally composed via simple addition for effective cross-lingual transfer (as shown in §3.2). An illustration of DEFT-X is shown in Figure 1.

We evaluate the zero-shot cross-lingual performance of DEFT-X on sentiment analysis (SA) and natural language inference (NLI) tasks for extremely low-resource Indonesian/indigenous languages in the NusaX/AmericasNLI benchmarks, respectively. Our results demonstrate that DEFT-X outperforms state-of-the-art baselines. Our findings highlight the importance of each step in the DEFT-X pipeline – denoising the model before selecting a sparse subnetwork, enforcing sparsity after denoising and fine-tuning the subnetwork, particularly in low-resource settings.

2 Background

2.1 Lottery Ticket Hypothesis (LTH)

The *Lottery Ticket Hypothesis (LTH)* (Frankle and Carbin, 2019; Malach et al., 2020) states that within a randomly initialized, dense neural network, there exists a subnetwork referred to as a *winning ticket* that when trained in isolation with its original initialization, can achieve comparable or even superior performance to the full network. Simple pruning techniques such as magnitude-based pruning can be used to identify such trainable subnetworks in fully connected and convolutional networks (Han et al., 2015, 2016; Yang et al., 2017).

To identify a winning ticket, a neural network $f(x; \theta^{(0)})$ is first initialized with random parameters $\theta^{(0)}$. The network is trained for j iterations, resulting in parameters $\theta^{(j)}$. Next, $p\%$ of the smallest-magnitude weights are pruned, creating a binary mask m. Finally, the remaining parameters are reset to their original values from $\theta^{(0)}$ to yield the winning ticket $f(x; m \odot \theta^{(0)})$. However, if a winning ticket's parameters are randomly re-initialized, its performance deteriorates, highlighting the im-

portance of proper initialization for effective training. This process of pruning and resetting helps uncover subnetworks that retain strong learning abilities when trained in isolation.

2.2 Lottery Ticket Sparse Fine-Tuning (LT-SFT)

Inspired by the LTH, Ansell et al. (2022) proposed a lottery ticket-based algorithm (LT-SFT) for efficient zero-shot cross-lingual transfer learning for low-resource languages. LT-SFT proposes a parameter-efficient fine-tuning approach that is both modular like adapters (Pfeiffer et al., 2021a, 2020) (i.e., it can be easily combined to adapt a model to different knowledge sources) and expressive like sparse fine-tuning (i.e., it changes the behavior of all components). LT-SFT outperformed the state-of-the-art MAD-X adapter-based approach for low-resource cross-lingual transfer; the algorithm is detailed below.

LT-SFT: Generate Sparse Fine-tuned Vectors (SFTs). The LT-SFT algorithm generates sparse fine-tuned vectors (SFTs) in two phases: (1) Pretrained model parameters $\theta^{(0)}$ are fine-tuned on a target task or language to obtain $\theta^{(1)}$. From the top-K winning tickets based on the greatest absolute difference $|\theta_i^{(1)} - \theta_i^{(0)}|$, a binary mask $b \in \{0, 1\}$ is constructed where $b = 1$ for selected parameters and $b = 0$ otherwise. (2) All the parameters are reset to $\theta^{(0)}$ and the model is retrained with keeping all non-winning parameters frozen using b. The resulting fine-tuned parameters $\theta^{(2)}$ yield the sparse fine-tuned vector (SFT) $\phi = \theta^{(2)} - \theta^{(0)}$.

LT-SFT: Zero-shot Transfer using SFTs. For each target language l, a language-specific SFT ϕ_L^l is estimated via the masked language modeling (MLM) objective on text from language l, initialized with the pretrained model weights θ_0. For each task t, a task-specific SFT ϕ_T^t is learned by training LT-SFT on annotated data in the source language s. For learning the task SFT, the LT-SFT algorithm first adapts the pre-trained model to the source language by adding the source language SFT ϕ_L^s to the model initialization i.e, $\theta_0 + \phi_L^s$. The model is then trained on the task to obtain updated parameters $\theta^{'}$. Finally, the task-specific SFT is computed by removing the source language SFT: $\phi_T^t = \theta^{'} - (\theta^{(0)} + \phi_L^s)$. During task training, a classifier head is learned and fully fine-tuned in both phases of LT-SFT, with its random initialization reset at the start of each phase. For zero-shot cross-lingual transfer, the language-specific SFT ϕ_L and task-specific SFT ϕ_T are composed with the pretrained model as $\theta_{TL} = \theta^{(0)} + \phi_T + \phi_L$ to yield the target language-task model. The classifier head trained for the task is stacked on top to obtain the final model.

3 Methodology

3.1 DEFT-X: Motivation

Sparse fine-tuned vectors enable parameter-efficient cross-lingual transfer, but its effectiveness depends heavily on the quality of the identified sparse vectors. In LT-SFT, these vectors might capture noise or irrelevant information. DEFT-X mitigates this by introducing a low-rank denoising step prior to sparse fine-tuning. DEFT-X prunes the higher-order (lower singular value) components from the model weights, which are more likely to capture uninformative or noisy artefacts. The resulting denoised sparse vectors lead to more effective and robust transfer, especially in low-resource language scenarios.

Denoising using Low-Rank Approximation. In DEFT-X, we start with identifying the 'winning tickets' for efficient cross-lingual transfer. We compute the difference between the pretrained model parameters $\theta^{(0)}$ and the fully fine-tuned model parameters $\theta^{(1)}$ to obtain

$$\delta = \theta^{(1)} - \theta^{(0)} \tag{1}$$

To extract the winning tickets in δ, we first obtain a low-rank approximation (Zhao et al., 2025) by decomposing each weight matrix $W \in \mathbb{R}^{m \times n}$ in δ using Singular Value Decomposition (SVD):

$$W = U \Sigma V^T \tag{2}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are the left and right singular vector matrices of W and $\Sigma \in \mathbb{R}^{m \times n}$ is the diagonal matrix of singular values. We construct a low-rank approximation of W by retaining its lower-order (high singular value) components while pruning noise from the higher-order (low singular value) components:

$$L = U_r \Sigma_r V_r^T, \qquad S = m \odot (W - L) \tag{3}$$

where U_r and V_r denote the first r columns of U and V, respectively. The matrix L captures the lower-order components, while S represents the

Task	Target Dataset	Target Languages	Source Language	Source Task Dataset
Sentiment Analysis (SA)	NusaX (Winata et al., 2023)	Acehnese, Balinese, Banjarese, Madurese, Minangkabau	Indonesian	SMSA (Purwarianti and Crisdayanti, 2019; Wilie et al., 2020)
Natural Language Inference (NLI)	AmericasNLI (Ebrahimi et al., 2022)	Aymara, Asháninka, Bribri, Guarani, Náhuatl, Otomí, Quechua, Rarámuri, Shipibo-Konibo, Wixarika	English	MultiNLI (Williams et al., 2018)

Table 1: Details of the tasks, datasets, and languages involved in our zero-shot cross-lingual transfer evaluation. All target languages are low-resource and were unseen during XLM-R pretraining. All the training data was obtained from the authors of Ansell et al. (2022, 2023a). Further details are provided in Appendix B.

denoised higher-order components. The mask m is defined as:

$$m_i = \begin{cases} 1, & \text{if } i \in \text{Top-}n\text{-indices of } |W - L| \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Here, the higher-order components $(W - L)$ are pruned using magnitude-based selection, where the mask m retains the n largest absolute values while discarding the rest. The final low-rank approximation of W is reconstructed as:

$$W \approx L + S \tag{5}$$

Sparse Fine-Tuning. After obtaining the low-rank approximation of δ by altering each matrix as shown in Eqn 5, we apply magnitude pruning to obtain a sparse structure for efficient model composition. Sparsity is crucial to avoid destructive interference during model composition. For magnitude pruning, we construct a binary mask $\mu \in \{0, 1\}$ where $\mu = 1$ for the top-k entries in the denoised δ with the highest absolute values, and $\mu = 0$ otherwise. We reset all the parameters to pretrained weights $\theta^{(0)}$ and perform sparse fine-tuning while keeping the non-winning parameters frozen using μ. The resulting fine-tuned parameters $\theta^{(2)}$ yield the sparse fine-tuned vector $\phi = \theta^{(2)} - \theta^{(0)}$.

3.2 Zero-shot Cross-lingual Transfer Learning

The sparse fine-tuned vector obtained from DEFT-X can be easily composed similar to LT-SFT. For each target language l, a language-specific vector ϕ_L^l is learned via the DEFT-X algorithm using masked language modeling (MLM) with unlabeled text from language l. For each task t, a task-specific vector ϕ_T^t is learned by training DEFT-X on annotated data in the source language s. Similar to LT-SFT, for task vector training we first adapt the model to the source language by applying the language SFT of the source language ϕ_L^s and then train the model on the task. During task training, a classifier head is also learned, which is randomly initialized and fully fine-tuned. For zero-shot cross-lingual transfer, these language and task-specific vectors can be easily composed with the pretrained model to obtain the desired language-task model as follows: $\theta_{TL} = \theta^{(0)} + \phi_T + \phi_L$, where θ_{TL} represents the target language-task model. The classifier head learned for task t is stacked on top to obtain the final model that is used for zero-shot inference.

The algorithm for DEFT-X, including the steps for cross-lingual transfer, is given in Appendix A.

4 Experimental Setup

We evaluate zero-shot cross-lingual transfer learning on two low-resource benchmarks: Sentiment Analysis (NusaX) and Natural Language Inference (AmericasNLI). Table 1 summarizes the experimental setup, including the datasets and target languages considered.

4.1 Baselines

Our primary baseline is LT-SFT (Ansell et al., 2022), the current state-of-the-art framework for zero-shot cross-lingual transfer. We train all task-specific and target-language SFTs using the dataset provided by the LT-SFT authors and report the corresponding results. Additionally, we compare against the MAD-X 2.0 variant (Pfeiffer et al., 2021b), using previously reported MAD-X results from Ansell et al. (2022).

Model	Method	mad	bjn	ban	ace	min	Avg.
XLM-R$_{\text{BASE}}$	MAD-X (Ansell et al., 2022)	68.5	77.6	78.0	74.9	79.9	75.8
	LT-SFT	79.0	82.7	80.4	75.7	83.0	80.2
	DEFT-X (r_l = r_t = 90% var)	**80.5**	**83.5**	**82.7**	74.2	**85.2**	**81.2**
	DEFT-X (r_l = r_t = 100)	**79.8**	**83.8**	**81.4**	**76.8**	**85.1**	**81.4**
	DEFT-X (r_l = 90% var; r_t= 100)	**79.8**	**84.1**	**82.2**	**76.3**	**83.8**	**81.2**
	DEFT-X (r_l = 100; r_t= 90% var)	**79.3**	**82.8**	**81.5**	75.1	**85.0**	**80.7**
XLM-R$_{\text{LARGE}}$	LT-SFT	74.9	86.7	83.4	80.0	87.1	82.4
	DEFT-X (r_l = r_t = 90% var)	74.0	86.0	82.4	78.9	**89.0**	82.1
	DEFT-X (r_l = r_t = 200)	**76.1**	**87.2**	**84.8**	79.2	**88.7**	**83.2**
	DEFT-X (r_l = 90% var; r_t= 200)	**75.8**	**87.0**	82.7	77.9	**87.8**	82.2
	DEFT-X (r_l = 200; r_t= 90% var)	**76.3**	86.0	**84.6**	79.9	**88.2**	**83.0**

Table 2: Zero-shot cross-lingual transfer evaluation (F1-Score) on SA task (NusaX) using XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$. XLM-R$_{\text{LARGE}}$ numbers are without ϕ_L^s initialization for task. For MAD-X baseline, we present the numbers reported in Ansell et al. (2022). Here, r_l and r_t denote the rank used for language and task sparse vectors respectively. **Bold** indicates performance surpassing the baselines, while underline denotes the best performance.

4.2 Training Setup

For our experiments, we use the pretrained XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$ models and conduct all training on a single NVIDIA A100 80GB GPU. To ensure fair comparisons, we adopt the same training setup used in LT-SFT (Ansell et al., 2022, 2023a) for both language and task training. Implementation details including the training steps, optimizer settings, etc. are detailed in Appendix C.

For language SFTs trained using the MLM objective, the number of trainable parameters k is set to $7.6M$ (i.e., 2.8% and 1.4% of the parameters in XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$, respectively). And for task SFTs, k is set to $14.2M$ (i.e., 5.2% and 2.6% of the parameters in XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$ respectively). This choice of k is consistent with the LT-SFT baseline, and the value of k was selected such that it matches the number of parameters in the MAD-X adapters.[1] During task adaptation, we always apply the source language SFT from LT-SFT (Ansell et al., 2022) to the XLM-R$_{\text{BASE}}$ model , while the XLM-R$_{\text{LARGE}}$ model is trained without source language initialization due to the unavailability of source language SFT for the large model.

Denoising using Low-Rank Approximation. We denoise all trainable weight matrices of the model, except for the bias terms; we apply direct magnitude-based pruning for the latter. The SVD operations on each weight matrix are computed in parallel. We explore two methods for selecting the appropriate rank to separate higher- and lower-order components in the matrix. Following Chang et al. (2022), we choose the rank r that captures 90% of the total variance for each matrix i.e, we first calculate the total variance from all singular values in the matrix, then retains the minimum number of singular vectors needed such that their cumulative variance reaches 90% of the total variance. Additionally, we investigate a uniform rank selection across layers by setting a uniform rank r = {100, 200, 300}, based on our empirical observation that most rank values derived from the 90% variance criterion fall within this range. To denoise the higher-order components, we apply magnitude pruning while retaining only 5% of the higher-order components in each matrix.

4.3 Evaluation Datasets

As shown in Table 1, NusaX is an SA benchmark covering five low-resource Indonesian languages.[2] There are 400 test samples in each language to be classified as either positive, negative or neutral. AmericasNLI (Ebrahimi et al., 2022) is an extension of XNLI (Conneau et al., 2018) to 10 low-resource indigenous languages of the Americas with 750 test samples each. We use F1-score

[1] The LT-SFT (Ansell et al., 2022) uses a reduction factor of 2 and 1 for language and task MAD-X adapters, respectively.

[2] There are four other test languages in NusaX. However, no monolingual corpora were available for these languages; hence, these are omitted from our results and LT-SFT.

Method	bzd	oto	hch	tar	cni	shp	aym	gn	nah	quy	Avg.
XLM-R$_{\text{BASE}}$											
MAD-X (Ansell et al., 2022)	44.0	46.8	41.5	43.9	47.6	48.9	58.8	63.5	53.7	58.3	49.5
LT-SFT	43.6	45.6	42.9	44.8	47.5	49.2	60.4	63.3	50.9	62.1	51.0
DEFT-X ($r_l = r_t$ = 90% var)	43.3	**47.6**	**44.0**	41.1	45.6	49.0	58.8	63.0	49.6	61.9	50.4
DEFT-X ($r_l = r_t$ = 200)	42.9	43.8	**45.6**	44.1	**48.0**	**49.3**	58.5	**63.6**	**53.9**	61.7	**51.2**
DEFT-X (r_l = 90% var; r_t= 200)	42.4	45.4	**44.0**	42.8	46.5	48.8	58.1	63.5	51.5	**62.8**	50.6
DEFT-X (r_l = 200; r_t= 90% var)	**44.3**	44.5	**43.6**	44.0	**48.1**	**50.5**	58.1	**64.5**	53.2	**62.3**	**51.3**
XLM-R$_{\text{LARGE}}$											
LT-SFT	43.9	42.1	45.2	42.5	46.9	48.7	58.0	54.7	39.6	50.1	47.2
DEFT-X ($r_l = r_t$ = 90% var)	**44.5**	40.5	44.9	**43.5**	46.8	**50.8**	57.5	52.5	37.8	**50.8**	47.0
DEFT-X ($r_l = r_t$ = 300)	**44.5**	41.4	45.2	**43.1**	46.1	**49.7**	**58.8**	**55.9**	39.3	**51.2**	**47.5**
DEFT-X (r_l = 90% var; r_t= 300)	**44.5**	41.2	43.9	**43.5**	46.7	**51.3**	57.5	**55.2**	**40.4**	**51.6**	**47.6**
DEFT-X (r_l = 300; r_t= 90% var)	**44.8**	40.1	**45.6**	**43.5**	**47.2**	**50.7**	**58.5**	**55.9**	38.9	49.7	**47.5**

Table 3: Zero-shot cross-lingual transfer evaluation (accuracy) on NLI task (AmericasNLI) using XLM-R$_{\text{BASE}}$ and XLM-R$_{\text{LARGE}}$. XLM-R$_{\text{LARGE}}$ numbers are without ϕ_L^s initialization for task. For MAD-X baseline, we present the numbers reported in Ansell et al. (2022). Here, r_l and r_t denote the rank used for language and task sparse vectors respectively. **Bold** indicates performance surpassing the baselines, while underline denotes the best performance.

and accuracy as the evaluation metrics for SA and NLI, respectively.

5 Results and Discussion

We report F1 scores for NusaX using both methods of rank selection: 90% variance and uniform rank r, as shown in Table 2. For XLM-R$_{\text{BASE}}$, we use uniform r = 100 and for XLM-R$_{\text{LARGE}}$ we use a higher rank r = 200; these rank values were selected by observing the overall rank that covers 90% variance. For XLM-R$_{\text{BASE}}$, DEFT-X consistently outperforms the baselines MAD-X and LT-SFT. Our best-performing configuration, with $r_l = r_t$ = 100, surpasses MAD-X and LT-SFT with average gains of 5.6 and 1.2, respectively. For XLM-R$_{\text{LARGE}}$, we report the numbers without source language initialization for the task since the source language sparse vectors are not available for XLM-R$_{\text{LARGE}}$ model. DEFT-X also outperforms the baseline LT-SFT on various settings using XLM-R$_{\text{LARGE}}$.

Similarly, we report the accuracy for AmericasNLI using both methods of rank selection: 90% variance and uniform rank r = 200 and r = 300 for the base and large model respectively, as shown in Table 3. XLM-R$_{\text{BASE}}$, with r_l = 200 and r_t = 90% variance, surpasses MAD-X and LT-SFT by 1.8% and 0.3%, respectively, on average. We observed that the languages in AmericasNLI are more low-resource than those in NusaX and require a higher rank r_l = 200 to capture useful lower-order components. However, both baseline methods and DEFT-X show degraded performance with XLM-R$_{\text{LARGE}}$ compared to the base model. This could be attributed to stronger biases towards high-resource languages during XLM-R$_{\text{LARGE}}$'s pretraining that make it less amenable to adapt to extremely low-resource languages in AmericasNLI. Even in such challenging settings, DEFT-X maintains a consistent albeit modest improvement over LT-SFT.

In summary, we observe that denoising higher-order components before selecting the sparse subnetwork (via magnitude pruning) improves the

Method	SA (F1-Score)	NLI (Accuracy)
LT-SFT	80.2	51.0
LT-SFT w/o ϕ_L^s for task	79.6	50.4
DEFT-X	**81.4**	**51.3**
DEFT-X w/o ϕ_L^s for task	**81.1**	**51.3**

Table 4: Comparison of LT-SFT and DEFT-X without source language initialization for task vectors, using XLM-R$_{\text{BASE}}$ on the SA and NLI tasks. We use $r_l = r_t$ = 100 for SA and r_l = 200, r_t= 90% variance for NLI. **Bold** indicates best performing model.

Method	mad	bjn	ban	ace	min	Avg.
DEFT-X	79.8	83.8	81.4	76.8	85.1	81.4
w/o higher-order components	79.8	82.8	82.1	73.4	84.9	80.6
w/o magnitude pruning + sparse fine-tuning	77.4	81.7	79.7	74.6	82.7	79.2
w/o sparse fine-tuning	72.0	75.4	67.7	57.4	81.4	70.8

Table 5: Analyzing the impact of higher-order components, sparsity(magnitude pruning), and the necessity of re-training (sparse fine-tuning) in DEFT-X using the NusaX dataset on XLM-R_{BASE} with $r_l = r_t = 100$. Underline denotes the best performance.

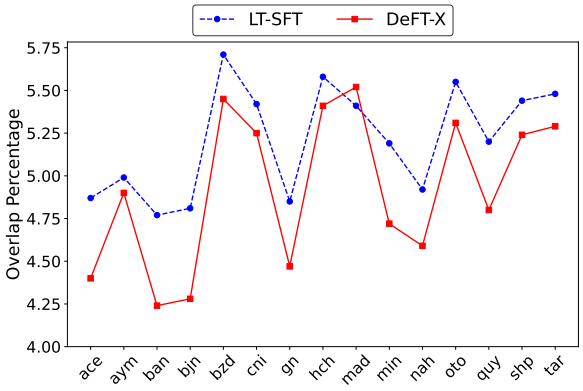

Figure 2: Comparing the overlap between the sparse language vectors and its corresponding task vectors. For DEFT-X, we compare using $r_l = r_t$ =100.

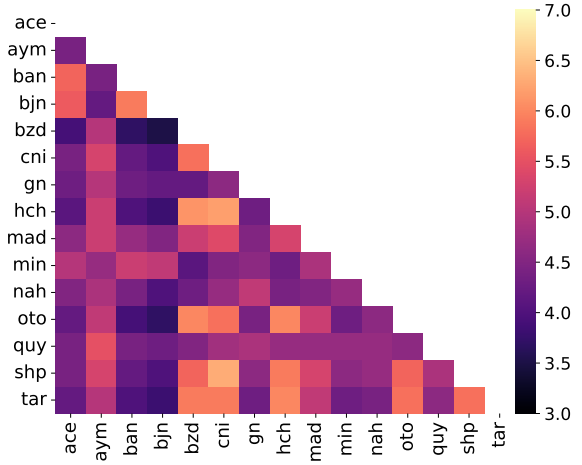

Figure 3: Overlap (in percentage) between the sparse language vectors of DeFT-X at r_l=100.

quality of sparse vectors for composition. Both uniform rank selection and 90% variance-based rank selection perform comparably well. However, very low-resource languages may benefit from a higher rank to capture more meaningful knowledge.

Source Language Initialization for Task Vectors. We analyze the impact of source language initialization on training task sparse vectors in Table 4. Our findings indicate that the baseline LT-SFT is sensitive to source language initialization, as its performance drops in its absence. In contrast, our approach, DEFT-X, maintains comparable performance even without source language initialization. This suggests that denoising higher-order components before selecting the subnetwork (via magnitude pruning) leads to a more robust network compared to LT-SFT, which relies solely on magnitude pruning for subnetwork selection.

Benefits of Denoising, Sparsity and Re-Training. We investigate the need to retain de-noised higher-order parameters by comparing DEFT-X with an alternative that entirely removes these parameters, as shown in Table 5. We observe that higher-order components contain useful information, making it essential to retain them after denoising. We also analyze the impact of magnitude pruning and sparse fine-tuning after denoising in Table 5. We find that magnitude pruning, when combined with sparse fine-tuning, effectively refines the parameter selection beyond denoising, leading to performance gains. In contrast, applying magnitude pruning alone without sparse fine-tuning results in a substantial drop in performance, underscoring the necessity of fine-tuning after pruning. This ablation study highlights the importance of each step in the DEFT-X algorithm.

Parameter Overlap in Language and Task Vectors. One of the key challenges in composing models for cross-lingual transfer is minimizing negative interference. Reducing parameter overlap can mitigate this interference by ensuring that language and task specific vectors learn distinct subnetworks. In Figure 2, we compare the parameter overlap between sparse fine-tuned language vectors and their corresponding task vectors for LT-SFT and DEFT-X. Our findings show that DEFT-X results in lower overlap across languages, indicating that denoising

higher-order components helps remove redundancies while preserving language and task specific information. This leads to efficient cross-lingual transfer with minimized negative interference. We also analyze the overlap between language vectors in Figure 3 and find the overlap to be merely $\sim$5%, suggesting that each language learns a distinct subnetwork within the pretrained model.

6 Related Work

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning (PEFT) adapts large pretrained models to downstream tasks with minimal trainable parameters. DiffPruning (Guo et al., 2021) learns sparse task-specific deltas using a differentiable L_0 penalty, while BitFit (Zaken et al., 2022) restricts the updates to bias terms. Adapters (Pfeiffer et al., 2021a) insert lightweight task-specific bottlenecks into Transformer layers, keeping the rest frozen. LoRA (Hu et al., 2022) introduces trainable low-rank decomposition matrices into each Transformer layer. Wanda (Sun et al., 2023) prunes weights based on the elementwise product of its magnitude and the corresponding input activation norm. These approaches significantly reduce the number of trainable parameters while preserving performance. Similarly, our approach DEFT-X uses magnitude-based pruning, while further mitigating noise through low-rank denoising.

Task Arithmetic and Multi-Task Transfer Learning. Ilharco et al. (2023) introduces task vectors obtained by subtracting the weights of a pretrained model from those of a fine-tuned model. These task vectors can be manipulated through arithmetic operations to steer model behavior. Approaches like Ansell et al. (2024) and DARE (Yu et al., 2024) propose identifying better task vectors by dynamically dropping and learning parameters. Various approaches were proposed to mitigate the destructive interference during task arithmetic in multi-task setting like resolving sign conflicts while merging (Yadav et al., 2023) and learning mutually sparse vectors (Panda et al., 2024). DEFT-X can also be used in multi-task setting to obtain better task vectors by denoising the redundant task information and mitigating destructive interference.

Cross-Lingual Transfer Learning. Cross-lingual transfer learning improves task performance in low-resource languages by leveraging knowledge from high-resource languages. Approaches like MAD-X (Pfeiffer et al., 2020) use modular language and task specific adapters for composable transfer. MAD-G (Ansell et al., 2021) introduces a contextual parameter generator trained on typological features from URIEL to build efficient adapters for low-resource languages. LT-SFT (Ansell et al., 2022) proposes sparse task and language vectors that can be arithmetically composed for zero-shot transfer. Subsequent works combine few-shot fine-tuning with LT-SFT (Ansell et al., 2023a) and explore scaling in task arithmetic (Parović et al., 2024). Ansell et al. (2023b) proposes a bilingual distillation approach to extract language-specific models from massively multilingual transformers for cross-lingual transfer. Layer swapping for transferring linguistic knowledge to reasoning tasks has also been explored (Bandarkar et al., 2024). Our approach DEFT-X, builds on LT-SFT by improving sparse vector quality via SVD-based denoising.

Low-Rank Approximation using SVD. Singular Value Decomposition (SVD) is widely used for low-rank approximation, aiding both efficiency and interpretability in neural networks. LASER (Sharma et al., 2023) shows that pruning small singular components from Transformer weight matrices can improve reasoning by denoising internal representations without retraining. LoRS-Merging (Zhao et al., 2025) merges multilingual speech models using coarse-grained singular value pruning to retain essential structures and fine-grained magnitude pruning to remove redundancy. Our approach DEFT-X, applies similar low-rank denoising to enhance the quality of sparse fine-tuned vectors.

7 Conclusion and Future Work

We introduced DEFT-X, a composable, denoised, sparse fine-tuning approach for efficient zero-shot cross-lingual transfer. By leveraging SVD to denoise model weights, DEFT-X identifies better subnetworks for sparse fine-tuning. We also explored different strategies for selecting the matrix rank during denoising. Compared to the state-of-the-art LT-SFT approach, DEFT-X demonstrates improvements in SA and NLI tasks for low-resource languages. In future work, we plan to explore the broader applicability of DEFT-X beyond cross-lingual transfer, including its potential in multimodal learning and domain adaptation.

Limitations

While DEFT-X shows promising results, our study has a few limitations. First, we evaluate the method only on encoder-only architectures, specifically transformer-based language models, leaving its effectiveness on decoder-only or encoder-decoder models unexplored. Second, our experiments are restricted to classification tasks (SA, NLI), and the applicability of DEFT-X to other tasks remains to be investigated. Finally, the choice of the optimal rank for denoising via SVD is currently model and task-specific and requires manual tuning.

References

Alan Ansell, Marinela Parović, Ivan Vulić, Anna Korhonen, and Edoardo Ponti. 2023a. Unifying cross-lingual transfer across scenarios of resource scarcity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3995, Singapore. Association for Computational Linguistics.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. 2023b. Distilling efficient language-specific models for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8147–8165, Toronto, Canada. Association for Computational Linguistics.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alan Ansell, Ivan Vulić, Hannah Sterz, Anna Korhonen, and Edoardo M. Ponti. 2024. Scaling sparse fine-tuning to large language models. *Preprint*, arXiv:2401.16405.

Lucas Bandarkar, Benjamin Muller, Pritish Yuvraj, Rui Hou, Nayan Singhal, Hongjiang Lv, and Bing Liu. 2024. Layer swapping for zero-shot cross-lingual transfer in large language models. *Preprint*, arXiv:2410.01335.

Tyler Chang, Zhuowen Tu, and Benjamin Bergen. 2022. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1135–1143, Cambridge, MA, USA. MIT Press.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. *Preprint*, arXiv:2212.04089.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR.

Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. 2024. Lottery ticket adaptation: Mitigating destructive interference in llms. *Preprint*, arXiv:2406.16797.

Marinela Parović, Ivan Vulić, and Anna Korhonen. 2024. Investigating the potential of task arithmetic for cross-lingual transfer. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 124–137, St. Julian's, Malta. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. 2024. Modular deep learning. *Preprint*, arXiv:2302.11529.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ayu Purwarianti and Ida Ayu Putu Ari Crisdayanti. 2019. Improving bi-lstm performance for indonesian sentiment analysis using paragraph vector. In *2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, page 1–5. IEEE.

Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. 2023. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *Preprint*, arXiv:2312.13558.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *Preprint*, arXiv:2402.05672.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. 2017. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6071–6079.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *Preprint*, arXiv:2106.10199.

Qiuming Zhao, Guangzhi Sun, Chao Zhang, Mingxing Xu, and Thomas Fang Zheng. 2025. Low-rank and

sparse model merging for multi-lingual speech recognition and translation. *Preprint*, arXiv:2502.17380.

A DEFT-X Algorithm

Algorithm 1 Cross-Lingual Transfer with DEFT-X

function DEFT-X($D, L, \theta^{(0)}, \eta, k, n$)
  $\theta^{(1)} \leftarrow \theta^{(0)}$ ▷ Full fine-tuning
  while not converged **do**
    $\theta^{(1)} \leftarrow \theta^{(1)} - \eta \nabla L(\theta^{(1)}, D)$
  end while
  $\delta \leftarrow \theta^{(1)} - \theta^{(0)}$ ▷ SVD-based magnitude pruning
  for each weight matrix $W \in \delta$ **do**
    $U \Sigma V^T \leftarrow SVD(W)$
    $L \leftarrow U_r \Sigma_r V_r^T$
    $m_i \leftarrow \begin{cases} 1 & \text{if } i \in \text{Top-}n\text{-indices of } |W - L| \\ 0 & \text{otherwise} \end{cases}$
    $S \leftarrow m \odot (W - L)$
    $W \leftarrow L + S$
  end for
  $\mu_i \leftarrow \begin{cases} 1 & \text{if } i \in \text{Top-}k\text{-indices of } |\delta| \\ 0 & \text{otherwise} \end{cases}$
  $\theta^{(2)} \leftarrow \theta^{(0)}$ ▷ Sparse fine-tuning
  while not converged **do**
    $\theta^{(2)} \leftarrow \theta^{(2)} - \mu \odot \eta \nabla L(\theta^{(2)}, D)$
  end while
  $\phi \leftarrow \theta^{(2)} - \theta^{(0)}$
  return ϕ
end function

function CROSSLINGUAL($D_{src}, D_{tar}, D_{task}, L_{task}, \theta^{(0)}, \eta, k, n$)
  $\phi_{src} \leftarrow$ DEFT-X($D_{src}, L_{MLM}, \theta^{(0)}, \eta, k, n$)
  $\phi_{task} \leftarrow$ DEFT-X($D_{task}, L_{task}, \theta^{(0)} + \phi_{src}, \eta, k, n$)
  $\phi_{tar} \leftarrow$ DEFT-X($D_{tar}, L_{MLM}, \theta^{(0)}, \eta, k, n$)
  return $\theta^{(0)} + \phi_{task} + \phi_{tar}$
end function

B Dataset Details

We use publicly available data for training and evaluation with CC-BY-SA license. Table 6 provides a comprehensive overview of languages, their codes, linguistic families, and monolingual data sizes. For training the task vectors, we utilize the SMSA dataset (Purwarianti and Crisdayanti, 2019; Wilie et al., 2020) and the MultiNLI dataset (Williams et al., 2018).

Note: Since the NusaX dataset (Winata et al., 2023) was created through human translation of a subset of the SMSA dataset and we use the NusaX test set used by Ansell et al. (2023a) where they carefully removed every example from SMSA which appears in its original or modified form in the NusaX test set to avoid a data leak.

C Detailed Training Setup

Language Adaptation. The language vector is trained on the Masked Language Modeling (MLM) objective for the lesser of 100 epochs or 100,000 steps, using a batch size of 8 and a maximum sequence length of 256. However, training is subject to an absolute minimum of 30,000 steps, as 100 epochs appeared insufficient for some languages with very small corpora. Model checkpoints are evaluated every 1,000 steps on a held-out set comprising 5% of the corpus, and the checkpoint with the lowest validation loss is selected at the end of training. We use the AdamW optimizer with an initial learning rate of $5\text{e}{-5}$, which is linearly decayed to 0 over the course of training. For language SFTs, the number of trainable parameters, k is set to $7.6M$ (i.e., 2.8% of the parameters in XLM-R_{BASE}). For adapters, the reduction factor (i.e., the ratio between model hidden size and adapter size) is set to 2 to ensure the number of trainable parameters matches that of SFT. Additionally, layer normalization parameters are kept fixed, while all other parameters remain trainable. For language adaptation, we apply L1 regularization with $\lambda = 0.1$. The specified training regime is applied consistently across both phases of LT-SFT.

Task Adaptation. The task vector for SA is trained for 10 epochs with a batch size of 16, with checkpoint evaluation on the validation set every 250 steps, and the best checkpoint is taken at the end of training based on F1-score. The task vector for NLI is trained for 5 epochs with a batch size of 32, with checkpoint evaluation on the validation set every 625 steps, and the best checkpoint is taken at the end of training based on accuracy. Similarly to language adaptation, the task SFT training uses the AdamW optimizer with an initial learning rate of $5\text{e}{-5}$, which is linearly decayed to 0 over the course of training. A two-layer multi-class classification head is applied atop the XLM-R model output corresponding to the [CLS] token. For task SFTs, the number of trainable parameters k is set to $14.2M$ (i.e., 5.1% of the parameters in XLM-R_{BASE}). For adapters, the reduction factor is set to 1 to ensure the number of trainable parameters matches that of SFT. During task adaptation, we always apply the source language SFT from LT-SFT (Ansell et al., 2022).

D Ablation Experiments using Uniform Rank

We conducted ablation experiments on the uniform rank variant by setting rank $r = \{100, 200\}$ for the SA and NLI tasks. The results are presented in Table 7 and Table 8.

E mE5_{BASE}

In Table 9, we show the results for SA task on mE5_{BASE}. Multilingual E5 text embedding models (mE5) (Wang et al., 2024) are initilizalized from XLM-R models and trained using weakly-supervised contrastive pre-training on billions of text pairs, followed by supervised fine-tuning on a small quantity of high-quality labeled data. Due to the lack of availability of source lang initialization ϕ_L^s for mE5_{BASE}, we compare the performance without source lang initialization ϕ_L^s for task vectors. We observe an improvement of 2% using DEFT-X in mE5_{BASE} model.

Task	Language	ISO Code	Family	Corpus size (MB)
SA	Madurese	mad	Austronesian, Malayo-Sumbawan	0.84
	Banjarese	bjn	Austronesian, Malayo-Sumbawan	28.40
	Balinese	ban	Austronesian, Malayo-Sumbawan	42.50
	Acehnese	ace	Austronesian, Malayo-Sumbawan	89.70
	Minangkabau	min	Austronesian, Malayo-Sumbawan	92.80
NLI	Bribri	bzd	Chibchan, Talamanca	0.32
	Otomí	oto	Oto-Manguean, Otomian	0.40
	Wixarika	hch	Uto-Aztecan, Corachol	0.45
	Rarámuri	tar	Uto-Aztecan, Tarahumaran	0.61
	Asháninka	cni	Arawakan	1.40
	Shipibo-Konibo	shp	Panoan	2.00
	Aymara	aym	Aymaran	2.20
	Guarani	gn	Tupian, Tupi-Guarani	6.60
	Náhuatl	nah	Uto-Aztecan, Aztecan	7.70
	Quechua	quy	Quechuan	16.00

Table 6: Details of languages, their codes, linguistic families, and monolingual data sizes used for MLM training of language vectors.

Method	mad	bjn	ban	ace	min	Avg.
MAD-X (Ansell et al., 2022)	68.5	77.6	78.0	74.9	79.9	75.8
LT-SFT	79.0	82.7	80.4	75.7	83.0	80.2
DEFT-X ($r_l = r_t = 100$)	**79.8**	**83.8**	**81.4**	**76.8**	**85.1**	**81.4**
DEFT-X ($r_l = r_t = 200$)	78.4	**83.0**	**81.0**	73.9	**83.4**	80.0
DEFT-X ($r_l = 100$; $r_t = 200$)	**79.1**	**83.4**	**80.7**	74.1	**83.9**	80.2
DEFT-X ($r_l = 200$; $r_t = 100$)	**78.5**	82.5	**83.0**	**76.5**	**84.6**	**81.3**

Table 7: Zero-shot cross-lingual transfer evaluation (F1-Score) on SA task (NusaX) for different uniform rank values using XLM-R_{BASE}. Here, r_l and r_t denote the rank used for language and task sparse vectors respectively. **Bold** indicates performance surpassing the baseline, while underline denotes the best performance.

Method	bzd	oto	hch	tar	cni	shp	aym	gn	nah	quy	Avg.
MAD-X (Ansell et al., 2022)	44.0	46.8	41.5	43.9	47.6	48.9	58.8	63.5	53.7	58.3	49.5
LT-SFT	43.6	45.6	42.9	44.8	47.5	49.2	60.4	63.3	50.9	62.1	51.0
DEFT-X ($r_l = r_t = 100$)	44.0	45.4	42.3	40.0	46.1	**50.7**	60.3	63.0	52.0	61.1	50.5
DEFT-X ($r_l = r_t = 200$)	42.9	43.8	**45.6**	44.1	**48.0**	**49.3**	58.5	**63.6**	**53.9**	61.7	**51.2**
DEFT-X ($r_l = 100$; $r_t = 200$)	43.2	45.3	**44.4**	41.5	47.3	**50.1**	60.4	62.0	**54.6**	61.5	51.0
DEFT-X ($r_l = 200$; $r_t = 100$)	**44.7**	43.3	42.7	43.7	46.7	**49.3**	57.6	**63.9**	**53.9**	61.6	50.7

Table 8: Zero-shot cross-lingual transfer evaluation (accuracy) on NLI task (AmericasNLI) for different uniform rank values using XLM-R_{BASE}. Here, r_l and r_t denote the rank used for language and task sparse vectors respectively. **Bold** indicates performance surpassing the baseline, while underline denotes the best performance.

Model	Method	mad	bjn	ban	ace	min	Avg.
$mE5_{BASE}$	LT-SFT	59.0	82.1	73.6	68.9	77.0	72.1
	DEFT-X (r_l = r_t = 100)	**67.0**	**83.1**	70.9	64.7	**77.2**	**72.6**
	DEFT-X (r_l = 90% var; r_t= 90% var)	**68.8**	**83.5**	73.0	67.7	76.9	**74.0**
	DEFT-X (r_l = 100; r_t= 90% var)	**66.2**	**83.5**	70.3	63.8	77.0	72.1
	DEFT-X (r_l = 90% var; r_t= 100)	**69.3**	**83.5**	72.5	68.2	76.7	**74.1**

Table 9: Zero-shot cross-lingual transfer evaluation (F1-Score) on SA task (NusaX) using $mE5_{BASE}$ without ϕ_L^s initialization for task. For each model, **bold** indicates performance surpassing the baseline LT-SFT, while underline denotes the best performance.