

General Instruction-Tuned Multilingual Adapter: Unlocking Cross-Lingual Generalization Through Scaled Parameter-Efficient Adaptation

Sona Elza Simon[†], Sayambhu Sen^{*}, Jonathan Rubin^{*}, Mingxian Wang^{*}

[†] Indian Institute of Technology Bombay, India,

^{*} Amazon Alexa

sona.simon@iitb.ac.in, {sensayam, jonrubin, mingxiw}@amazon.com

Abstract

Pretrained multilingual language models perform well on many NLP tasks, yet their performance remains uneven, with English consistently outperforming most other languages. Multilingual adaptation aims to close this gap while preserving English performance, but faces challenges such as limited task-specific multilingual data, the curse of multilinguality, and catastrophic forgetting. Large language models encode strong task-solving knowledge in English; this knowledge transfers poorly across languages. Existing approaches have explored modular adaptation methods and representation alignment techniques that transfer task knowledge from English or task-agnostic latent spaces, but these methods often require specialized routing mechanisms or can constrain native multilingual generation. In this work, we show that injecting multilingual knowledge through a single scaled parameter-efficient adapter trained on general-purpose multilingual instruction data improves performance across most of the languages and tasks, even without task-specific supervision. Experiments show strong gains on multilingual reasoning and tool-use tasks, with limited improvement on instruction following and arithmetic reasoning. We further show through spectral analysis that scaling the adapter reduces intruder dimensions, enabling effective integration of multilingual knowledge while preserving the model’s pretrained capabilities.

1 Introduction

Pretrained language models form the foundation of modern NLP, yet their capabilities remain uneven across languages. While large language models (LLMs) exhibit strong reasoning and task-solving abilities, these capabilities are concentrated primarily in English. Multilingual LLMs partially reduce this gap, but substantial disparities persist, compounded by the scarcity and cost of task-specific multilingual datasets. Training separate models for

every task–language pair is impractical, motivating extensive work on cross-lingual transfer learning.

Existing approaches broadly fall into three categories. Task-specific methods improve performance across languages for a single task but generalize poorly to new tasks or domains (Chronopoulou et al., 2023, 2024; Bandarkar et al., 2025). Modular approaches train separate task- and language-specific components that must be recombined at inference, requiring explicit routing that scales poorly in dynamic settings (Zhao et al., 2025; Ansell et al., 2022; Simon and Jyothi, 2025). Representation alignment methods map multilingual representations into a shared, often English-centric space, which can restrict native multilingual generation (Yoon et al., 2024; Huang et al., 2024b; Ruan et al., 2025). Although effective in controlled settings, these approaches are ill-suited for real-world multilingual multitasking, where models must adapt across languages and tasks without repeated retraining or manual intervention.

In this work, we explore whether multilingual adaptation can be achieved with a single task-agnostic adapter, without requiring task-specific supervision or separate adaptation for each language. Specifically, we study a scaled parameter-efficient adapter trained on general-purpose multilingual instruction data and ask whether it can transfer across tasks and languages. This leads to three questions:

- **RQ1:** Can a single task-agnostic multilingual adapter improve multilingual performance across diverse tasks and languages without task-specific supervision?
- **RQ2:** Does this generalization extend to languages unseen during the adapter training?
- **RQ3:** Why does evaluation-time scaling preserve task performance while enabling multilingual generalization?

Our experiments show that a single scaled adapter indeed improves multilingual performance across most tasks and languages, with the largest gains on knowledge-intensive tasks. While, math reasoning and generative tasks show little or no improvement. These results suggest that LLMs already encode much of the knowledge required for downstream tasks, while a key bottleneck lies in accessing this knowledge across languages.

2 Related Work

Multi-Task Adaptation. Adapting models to multiple tasks without separate training is an active area of research. LoRA-based approaches such as MultiLoRA (Wang et al., 2023) and LoRAHub (Huang et al., 2024a) compose or combine LoRA modules across tasks. Task Arithmetic (Ilharco et al., 2023) steers model behavior by combining task-specific weight vectors via arithmetic operations. These works demonstrate the potential of modular approaches for multi-task generalization. Scaling of adapter weights has been explored in multi-task adaptation, where task-specific adapters or task-specific weight vectors are combined via weighted arithmetic operations (Ilharco et al., 2023; Huang et al., 2024a). However, they primarily operate in supervised multi-task settings where each adapter corresponds to a specific task.

Cross-Lingual Transfer via Modular Adaptation. Cross-lingual transfer learning aims to transfer task knowledge from high to low-resource languages. Early work focused on encoder-based models using modular adapter frameworks such as MAD-X (Pfeiffer et al., 2020) and MAD-G (Ansell et al., 2021), which learn separate task and language adapters that are recombined at inference for zero-shot transfer. Subsequent work explored arithmetic composition methods. Language - task arithmetic (Chronopoulou et al., 2024), LT-SFT (Ansell et al., 2022), and DeFT-X (Simon and Jyothi, 2025) combine sparse or task-specific parameter updates to improve multilingual transfer.

Recent work extends these ideas to decoder-based LLMs. AdaMergeX (Zhao et al., 2025) disentangles task and language abilities through adaptive adapter merging, while Bandarkar et al. (2025) improve reasoning transfer through layer swapping between language and task experts. Chronopoulou et al. (2023) proposes multilingual task adapters by combining languages from the same families. Although effective, these approaches still rely on

explicit routing or task-specific supervision.

Cross-Lingual Transfer via Representation Alignment. Another line of work bridges multilingual encoders with LLMs through representation alignment. LangBridge (Yoon et al., 2024) maps multilingual encoder representations into the latent space of a task-specialized decoder using only English supervision, but generation remains tied to the decoder’s source language. MindMerger (Huang et al., 2024b) augments queries with translated inputs to combine multilingual representations with the LLM’s reasoning capabilities. LayAlign (Ruan et al., 2025) further introduces layer-wise alignment and adaptive fusion for multilingual transfer. Despite architectural differences, these approaches ultimately project multilingual representations into a shared or English-centric latent space, which can constrain native multilingual generation.

While scaled LoRA adapters have been explored in model merging (Zhao et al., 2025; Huang et al., 2024a), they are generally applied in supervised settings involving multiple task- or language-specific adapters. In contrast, our approach uses a single multilingual adapter trained on general-purpose instruction data and applies evaluation-time scaling in a zero-shot setting. We further provide mechanistic evidence motivating the need for scaling.

3 Methodology

Our goal is to improve the multilingual performance of a pretrained instruction-tuned LLM across diverse tasks and languages without task-specific multilingual supervision. We build on the observation that LLMs already encode strong task-solving capabilities, primarily accessible in English, and that the key bottleneck for multilingual generalization is linguistic access rather than task knowledge (Pires et al., 2019). This motivates a simple question: *Can multilingual knowledge be injected into a pretrained model without disrupting its existing task-solving capabilities?*

3.1 Multilingual Adapter Training

To inject multilingual knowledge efficiently, we use Low-Rank Adaptation (LoRA) (Hu et al., 2022), motivated by the low intrinsic dimensionality of pretrained models (Aghajanyan et al., 2021). We train a single multilingual LoRA adapter BA_{multi} on general-purpose multilingual instruction data from Bactrian-X (Li et al., 2023), covering eight

non-English languages:

$$W = W_0 + BA_{\text{multi}} \quad (1)$$

where $W_0 \in \mathbb{R}^{d \times k}$ is frozen, and $A \in \mathbb{R}^{r \times k}$, $B \in \mathbb{R}^{d \times r}$ are trainable low-rank matrices with $r \ll \min(d, k)$. Standard LoRA initialization is used so that the update is zero at the start of training.

3.2 Evaluation-Time Scaling

A naive approach would directly merge the adapter into the base model. However, prior work shows that LoRA training can introduce *intruder dimensions*, new orthogonal directions in the parameter space that interfere with existing pretrained representations (Shuttleworth et al., 2025). Merging the adapter at full scale can therefore overwrite the different task knowledge encoded in W_0 . To mitigate this, we use an evaluation-time scaling factor λ :

$$W = W_0 + \lambda \cdot BA_{\text{multi}} \quad (2)$$

In practice, we identify the scaling factor λ using a validation set. However, our spectral analysis (Section 5.3) reveals that the optimal λ coincides with a principled structural criterion, suggesting a validation-free method for selecting the scaling factor. This analysis also provides a mechanistic explanation for why scaling enables multilingual transfer without degrading task performance.

4 Experimental Setup

We evaluate our approach on Qwen3-4B (Qwen/Qwen3-4B) and Mistral-7B-Instruct (mistralai/Mistral-7B-Instruct-v0.3). We compare against two baselines: the zero-shot performance of the base instruction-tuned model and the use of weighted or scaled monolingual adapters trained separately for each language. While monolingual adapters offer an ideal baseline by directly applying language-specific adaptation, maintaining separate adapters for every language is impractical. In contrast, a single multilingual adapter that matches or surpasses monolingual performance is far more suitable for real-world deployment.

4.1 Training Dataset and Setup

Adapter Training. We train multilingual LoRA adapters using the Bactrian-X dataset (Li et al., 2023). All the experiments were performed on NVIDIA A10G (24GB) GPUs. For LoRA adaptation, we set $r = 16$ and $\alpha = 32$. We apply LoRA to all linear projection layers in the Transformer blocks. More details in Appendix A.1.

Monolingual and Multilingual Adapters. We train the language adapters using general instruction fine-tuning data from the Bactrian-X (Li et al., 2023) dataset. In this work, we focus on eight languages: Spanish (es), French (fr), German (de), Italian (it), Portuguese (pt), Japanese (ja), Czech (cs), and Russian (ru). The dataset contains 67k samples per language. For training monolingual adapters, we use all 67k samples. For the multilingual adapter, we maintain the same total training budget by sampling 67k/8 samples per language and mix them randomly for training. We use a single scaling factor across all languages and tasks, selected using the available validation sets: 0.25 for Qwen and 0.3 for Mistral.

4.2 Evaluation Datasets

We evaluate on a diverse set of multilingual benchmarks. Refer to Appendix A.2 for more details.

Multilingual Reasoning. We evaluate on Massive Multitask Language Understanding (MMMLU) (Hendrycks et al., 2021), which spans 57 subjects across 14 languages, and MMLU-ProX (Xuan et al., 2025), which spans 29 languages with 10-choice multiple-choice questions.

Instruction Following. We evaluate on M-IFEval (Dussolle et al., 2025), a multilingual extension of IFEval (Zhou et al., 2023) that evaluates adherence to constraints on format, length, and content across four languages.

Tool Use. We evaluate multilingual tool invocation using the BenchMAX Tool-Use benchmark (Huang et al., 2025), which requires generating function calls with correct names and arguments from natural language queries in 17 languages.

Arithmetic Reasoning. We evaluate on MGSM (Shi et al., 2022), the Multilingual Grade School Math benchmark, which was constructed by manually translating the GSM8K dataset (Cobbe et al., 2021) into ten typologically diverse languages.

5 Results and Discussion

5.1 RQ1: Can a single task-agnostic multilingual adapter improve multilingual performance across diverse tasks and languages without task-specific supervision?

Multilingual Reasoning. Tables 1 and 2 report zero-shot results on MMMLU and MMLU-ProX.

Method	en	es	fr	de	it	pt	ja	Avg. Non-En	Avg.
Qwen3-4B	71.7	68.5	67.3	65.6	67.8	68.3	64.1	66.9	67.6
+ Monolingual Adapter	80.0	70.4	69.4	67.4	69.5	70.1	69.0	69.3	70.8
+ Multilingual Adapter	74.4	70.9	69.5	68.4	70.1	70.2	70.0	69.9	70.5
Mistral-7B-Instruct	25.5	28.3	27.6	27.9	30.4	26.0	26.9	27.8	27.5
+ Monolingual Adapter	55.0	46.7	38.5	42.8	47.4	38.8	37.0	41.9	43.7
+ Multilingual Adapter	50.9	43.7	39.5	44.0	45.0	37.0	35.3	40.8	42.2

Table 1: Results of multilingual adaptation on the MMMLU benchmark on **seen** languages. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Method	en	es	fr	de	it	pt	ru	cs	ja	Avg. Non-En	Avg.
Qwen3-4B	46.0	46.7	46.5	44.1	46.2	46.1	46.5	44.5	44.5	45.6	45.7
+ Monolingual Adapter	56.4	50.9	50.7	49.4	50.2	50.7	50.6	49.2	49.5	50.1	50.8
+ Multilingual Adapter	52.6	51.9	51.4	50.0	51.9	52.4	49.5	48.6	49.0	50.6	50.8
Mistral-7B-Instruct	24.0	20.2	20.1	18.4	20.5	19.5	17.7	17.1	16.4	18.7	19.3
+ Monolingual Adapter	28.0	26.7	23.0	26.1	27.5	23.9	21.2	22.8	21.1	24.0	24.5
+ Multilingual Adapter	31.4	26.3	25.9	27.1	24.2	24.1	23.0	24.0	20.3	24.4	25.1

Table 2: Results of multilingual adaptation on the MMLU-ProX benchmark on **seen** languages. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

The scaled multilingual adapter consistently improves over the base models. For Qwen, it achieves the best average non-English performance on both MMMLU (+3.0) and MMLU-ProX (+5.0), matching or exceeding monolingual adapters. For Mistral, gains are also substantial (+13.0 on MMMLU, +5.7 on MMLU-ProX). English performance improves for both models despite being excluded from training.

Instruction Following. Table 3 presents zero-shot results on M-IFEval. Improvements are limited: for Qwen, the adapter improves Spanish but underperforms on French and Japanese. For Mistral, both adapter types underperform the base model. Multilingual adaptation appears less effective for generation-heavy tasks with strict formatting constraints.

Tool Use. Table 4 shows zero-shot results on Tool Use. For Qwen, the multilingual adapter improves non-English performance (+1.9), outperforming monolingual adapters overall. For Mistral, it improves substantially over the base (+6.4), though monolingual adapters remain stronger.

Arithmetic Reasoning. Table 5 reports the zero-shot MGSM results. Gains are modest for both models compared to knowledge-intensive benchmarks.

Overall, these results provide an answer to RQ1. A single multilingual adapter trained without task-specific supervision can improve multilingual per-

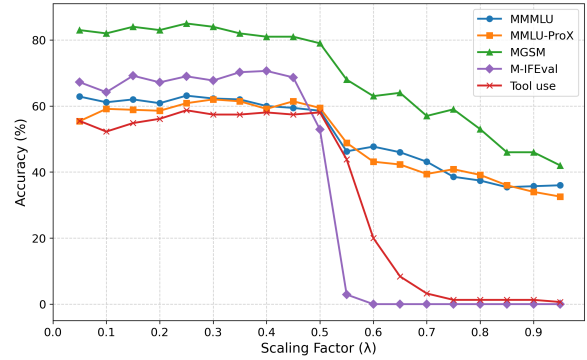


Figure 1: Average accuracy across eight languages vs. scaling factor (λ) for Qwen3-4B. Performance peaks at small λ and drops sharply beyond $\lambda \geq 0.5$.

formance across diverse tasks and languages, often matching monolingual adapters on knowledge-intensive benchmarks. However, the gains are task-dependent. The largest improvements occur on multilingual knowledge-intensive tasks such as MMMLU, MMLU-ProX, and Tool Use. Generation and reasoning-intensive benchmarks such as M-IFEval and MGSM show inconsistent gains.

Influence of the Scaling Factor (λ). Figures 1 and 2 show the effect of evaluation-time scaling. For both models, small scaling factors produce the strongest performance, while values $\lambda \geq 0.5$ cause sharp degradation across tasks. We select a single λ per model based on the validation splits of MMLU, MMLU-ProX, and MGSM, the only benchmarks for which validation data are avail-

Method	en		es		fr		ja		Avg. Non-En		Avg.	
	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose
Qwen3-4B	81.1	84.0	81.7	84.6	79.7	82.1	51.0	54.8	70.8	73.8	73.4	76.4
+ Monolingual Adapter	78.8	81.9	83.7	86.5	77.4	81.1	46.5	47.1	69.2	71.6	71.6	74.1
+ Multilingual Adapter	78.6	81.9	87.5	89.4	78.3	81.1	47.7	50.3	71.2	73.6	73.0	75.7
Mistral-7B-Instruct	48.5	51.6	61.5	66.4	48.1	49.1	24.5	30.3	44.7	48.6	45.7	49.3
+ Monolingual Adapter	45.4	51.4	61.9	66.0	43.4	47.6	24.1	30.0	43.1	47.9	43.7	48.8
+ Multilingual Adapter	45.2	51.0	62.5	66.4	42.5	46.7	23.1	25.3	42.7	46.1	43.3	47.3

Table 3: Results of multilingual adaptation on the M-IFEval benchmark on **seen** languages. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. We report both strict and loose accuracy. **Bold** indicates the best performance.

Method	en	es	fr	de	ru	cs	ja	Avg. Non-En	Avg.
Qwen3-4B	59.6	54.7	58.9	54.4	58.2	57.1	54.0	56.2	56.7
+ Mono. Adapter	61.0	60.3	57.1	53.3	59.6	56.8	52.3	56.6	57.2
+ Multi. Adapter	62.7	60.6	60.3	56.5	59.9	54.4	57.1	58.1	58.8
Mistral-7B-Instruct	17.1	11.2	12.9	9.8	12.2	12.2	12.2	11.7	12.5
+ Mono. Adapter	26.4	18.0	23.7	21.3	18.0	14.6	22.0	19.6	20.6
+ Multi. Adapter	20.6	17.8	20.2	19.2	18.1	14.6	18.8	18.1	18.5

Table 4: Results of multilingual adaptation on the Tool Use benchmark on **seen** languages. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Method	en	es	fr	de	ru	ja	Avg. Non-En	Avg.
Qwen3-4B	83.6	80.0	77.8	76.9	76.9	69.8	76.3	77.5
+ Mono. Adapter	86.7	79.6	79.6	78.7	78.7	72.4	77.8	79.2
+ Multi. Adapter	85.8	80.1	78.7	79.1	77.8	71.6	77.4	78.8
Mistral-7B-Instruct	50.7	37.3	42.7	36.9	40.0	28.9	37.2	39.4
+ Mono. Adapter	51.0	43.6	37.3	38.2	43.1	25.3	37.5	39.7
+ Multi. Adapter	52.4	41.3	40.4	42.7	39.0	25.0	37.7	40.1

Table 5: Results of multilingual adaptation on the MGSM benchmark on **seen** languages. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

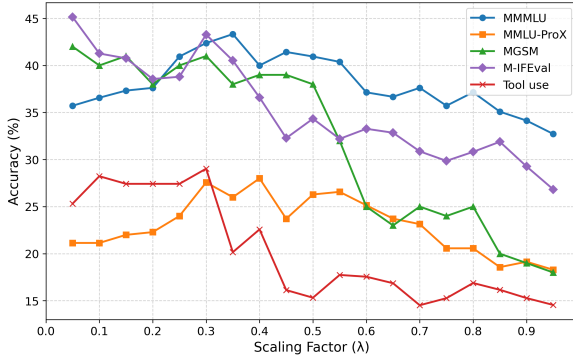


Figure 2: Average accuracy across eight languages vs. scaling factor (λ) for Mistral-7B-Instruct. Performance peaks at small λ and drops sharply beyond $\lambda \geq 0.3$.

able: $\lambda = 0.25$ for Qwen and $\lambda = 0.3$ for Mistral. As shown, this single λ generalizes well across all benchmarks. The strong sensitivity to scaling motivates the mechanistic analysis in RQ3.

5.2 RQ2: Does this generalization extend to languages absent from the adapter training?

Tables 6–9 evaluate transfer to unseen languages. For MIMLU, MIMLU-ProX, and Tool Use, the multilingual adapter improves average performance for both Qwen and Mistral. In contrast, MGSM shows mixed results: Qwen improves

slightly (+0.8 average), while Mistral remains unchanged. We do not report unseen-language results on M-IFEval because all three non-English benchmark languages are already present in the adapter training set. Overall, the pattern on unseen languages closely mirrors that observed for seen languages: multilingual reasoning and tool-use tasks benefit the most, while arithmetic reasoning shows weaker improvements.

These results provide a positive but nuanced answer to RQ2. The multilingual adapter generalizes beyond the training languages, although the magnitude of transfer varies across benchmarks and languages. The strongest unseen-language gains tend to appear in languages that share script or typological features with the training languages (e.g., Afrikaans +5.9, Ukrainian +5.4 on MIMLU-ProX), while typologically distant languages show smaller or mixed results.

5.3 RQ3: Why does evaluation-time scaling preserve task performance while enabling multilingual generalization?

To understand why small evaluation-time scaling factors are necessary, we analyze multilingual adaptation from three different perspectives: the layer-wise distribution of the update, the spectral effect of scaling, and the contrast with full fine-tuning.

Method	ar	bn	af	hi	hu	id	ko	mr	ne	sr	sw	te	th	uk	ur	vi	wo	yo	zh	zu	Avg.
Qwen3-4B	43.2	34.4	43.2	36.3	41.9	45.8	42.0	35.2	34.5	41.2	21.4	27.0	41.4	42.6	37.9	44.4	19.0	15.6	42.1	16.8	35.3
+ Multi. Adapter	45.7	30.6	49.1	34.9	39.7	49.8	44.2	34.6	36.0	47.8	23.9	29.3	47.2	48.0	40.0	50.7	22.8	17.2	46.6	19.8	37.9
Mistral-7B-Instruct	12.6	9.2	14.4	13.4	16.8	17.1	16.0	12.1	11.6	17.5	12.3	8.4	11.4	18.5	11.2	16.1	11.6	10.1	18.9	10.2	13.5
+ Multi. Adapter	14.1	9.7	22.6	13.3	19.0	23.2	22.1	12.6	12.7	22.1	18.5	11.1	12.4	23.0	10.5	19.6	13.2	13.1	22.8	15.1	16.5

Table 6: Results of multilingual adaptation on the **unseen** languages in MMLU-ProX benchmark. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Method	ar	bn	hu	ko	sr	sw	te	th	vi	zh	Avg.
Qwen3-4B	56.5	42.5	52.3	57.1	55.1	32.8	37.6	56.5	55.1	59.9	50.5
+ Multilingual Adapter	55.1	44.3	56.1	59.2	58.5	34.2	39.4	56.8	58.5	57.5	52.0
Mistral-7B-Instruct	11.5	12.5	10.0	13.6	13.2	7.0	9.4	13.6	13.6	12.5	11.7
+ Multilingual Adapter	17.1	13.2	15.3	18.8	21.6	12.0	13.4	10.0	15.5	13.2	15.0

Table 7: Results of multilingual adaptation on the **unseen** languages in Tool Use benchmark. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Method	ar	bn	hi	id	ko	sw	yo	zh	Avg.
Qwen3-4B	61.8	52.6	54.6	64.1	62.5	31.5	22.1	62.2	51.4
+ Multi. Adapter	64.1	51.7	57.4	66.4	63.3	31.5	21.1	64.0	52.4
Mistral-7B-Instruct	27.2	23.4	26.4	28.4	29.1	21.8	18.6	27.5	25.3
+ Multi. Adapter	30.0	22.1	24.5	34.9	34.6	27.4	25.6	41.1	30.0

Table 8: Results of multilingual adaptation on the **unseen** languages in MMMLU benchmark. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Method	zh	sw	th	bn	te	Avg.
Qwen3-4B	73.3	21.8	75.1	66.2	50.2	57.3
+ Multilingual Adapter	74.7	22.7	77.3	66.2	49.8	58.1
Mistral-7B-Instruct	34.2	8.4	24.0	8.9	4.4	16.0
+ Multilingual Adapter	38.2	8.0	20.4	9.8	4.0	16.1

Table 9: Results of multilingual adaptation on the **unseen** languages in MGSM benchmark. $\lambda = 0.25$ for Qwen-4B and $\lambda = 0.3$ for Mistral-7B-Instruct. **Bold** indicates the best performance.

Layer-wise Analysis. Figure 3 shows the layer-wise average max norm of the weight differences between the multilingual-adapted and base Qwen models. The largest changes occur in the early and late transformer layers, while the middle layers remain stable. This trend is consistent across different numbers of training languages. A finer-grained analysis by projection type, including comparisons with full fine-tuning, is provided in Appendix A.3.

Prior work associates early and late transformer layers with language-specific processing, while middle layers are more strongly linked to task reasoning (Bandarkar et al., 2025). The observed up-

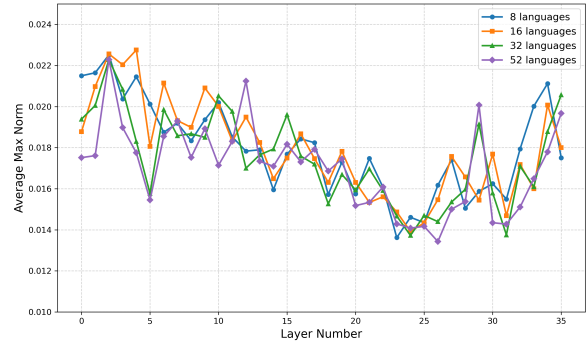


Figure 3: Layer-wise average max norm of weight difference between the base and the multilingual LoRA adapted Qwen3-4B model.

date pattern, therefore, suggests that multilingual adaptation primarily modifies language encoding and decoding behavior while largely preserving the pretrained task-solving representations. This pattern may also explain transfer to unseen languages. LoRA modifies the early and late layers more substantially, while the middle-layer updates remain relatively small. Scaling the LoRA update further attenuates these changes, reducing the already small middle-layer updates to negligible values while retaining small perturbations in the early and late layers. We hypothesize that this effectively modifies how multilingual inputs are mapped into the model’s shared representation space while preserving the pretrained task-solving representations, which may facilitate transfer beyond the training languages.

Spectral Analysis. LoRA updates can introduce new orthogonal directions to the base model weight

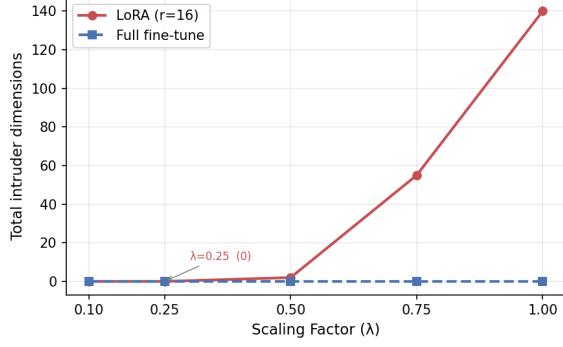


Figure 4: Effect of λ on the total intruder dimensions for $\text{LoRA}_{r=16}$ and full fine-tuning on Qwen3-4B. All intruder dimensions are suppressed at $\lambda = 0.25$.

Model	MMMLU	MMLU ProX	M-IFEval (Strict)	Tool Use	MGSM
Qwen3-4B	66.9	45.6	70.8	<u>56.2</u>	76.3
+ FFT ($\lambda = 1.00$)	14.1	8.5	21.3	0.9	5.9
+ FFT ($\lambda = 0.25$)	74.7	57.1	70.2	51.6	78.8
+ LoRA ($\lambda = 1.00$)	60.5	38.1	42.8	42.1	25.2
+ LoRA ($\lambda = 0.25$)	<u>69.9</u>	<u>50.6</u>	71.2	58.1	<u>77.8</u>

Table 10: Comparison of LoRA and FFT under evaluation-time scaling (λ). Results averaged over **seen** non-English languages. **Bold** denotes the best performance and underline denotes the second best.

matrices W_0 in parameter space, termed *intruder dimensions* (Shuttleworth et al., 2025), that interfere with pretrained task representations (formal definition in Appendix A.4). Figure 4 shows that at full scale ($\lambda = 1.0$), the multilingual LoRA adapter introduces a substantial number of intruder dimensions. As λ decreases, these are progressively suppressed, reaching zero at the optimal $\lambda = 0.25$. Full fine-tuning maintains zero intruder dimensions at all scaling values.

This suggests a practical, validation-free heuristic for selecting λ : compute intruder dimensions using only the base model and adapter weights, and scale down until the count reaches zero. At this point, the adapted model exhibits no intruder dimensions, indicating that the update introduces no new orthogonal directions. For Qwen3-4B, this criterion yields $\lambda = 0.25$, which coincides with the validation-selected optimum.

We formally prove that such a critical threshold exists: for any intruder dimension threshold ϵ , there is a λ^* below which no intruder dimensions can appear, determined by the spectral gaps of W_0 and the operator norm of the adapter (Theorem 1, Appendix A.6).

Model	MMMLU	MMLU ProX	M-IFEval (Strict)	Tool Use	MGSM
Qwen3-4B	66.9	45.6	70.8	56.2	76.3
+ LoRA ($\lambda = 0.25$)	69.9	50.6	71.2	58.1	77.8
+ DeFT-X ($\lambda = 0.25$)	75.8	59.4	72.5	56.8	79.5

Table 11: Comparison of LoRA- and DeFT-X-based multilingual adaptation on multiple multilingual benchmarks using Qwen3-4B ($\lambda = 0.25$). Results are averaged over **seen** non-English languages. **Bold** denotes the best performance.

Comparison with Full Fine-Tuning (FFT). Table 10 compares LoRA and FFT under evaluation-time scaling. Without scaling ($\lambda = 1.0$), both methods severely degrade the base model, with FFT collapsing more dramatically (e.g., Tool Use drops to 0.9) because it modifies all parameters with much larger total magnitude, effectively overwriting the pretrained model. With scaling ($\lambda = 0.25$), FFT recovers strongly on knowledge-intensive benchmarks (MMMLU, MMLU-ProX, MGSM), as its full-rank updates provide higher capacity to encode the multilingual signal. However, since FFT introduces no intruder dimensions, scaling suppresses all learned updates indiscriminately, explaining the persistent degradation on M-IFEval and Tool Use. LoRA’s low-rank constraint limits its peak knowledge transfer but concentrates the multilingual signal in a compact subspace, allowing scaling to selectively suppress interfering directions (intruder dimensions) while preserving task-relevant ones, yielding improvements across all five benchmarks.

Together, these analyses provide a consistent answer to RQ3. Evaluation-time scaling preserves task performance as the multilingual adapter concentrates its modifications in language-processing layers and introduces spectrally distinct intruder dimensions. Scaling selectively suppresses these interfering directions while retaining the distributed cross-lingual signals.

5.4 Generalization to Sparse Fine-Tuning

While our main experiments use LoRA, we investigate whether the benefits of scaled parameter addition extend to other multilingual adaptation methods¹ like DeFT-X (Simon and Jyothi, 2025), a sparse fine-tuning-based adaptation method. We train a sparse DeFT-X adapter spanning all layers

¹We exclude methods that require task-specific supervision as part of their algorithm, since our setting is task-free. AdaMergeX (Zhao et al., 2025) requires task-specific data to decouple task and language abilities, while LayAlign (Ruan et al., 2025) requires multilingual task-specific data in its two-stage training. Unlike DeFT-X, whose language module can be isolated for our setting.

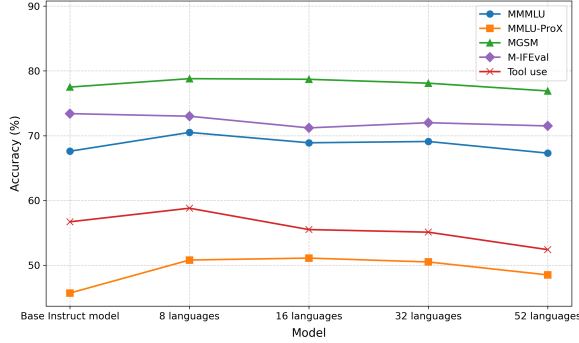


Figure 5: Average performance of Qwen3-4B ($\lambda = 0.25$) multilingual adapters trained on 8, 16, 32, and 52 languages.

using the same multilingual data as LoRA. Since DeFT-X has no scaling coefficient, we apply our evaluation-time scaling ($\lambda = 0.25$). For a fair comparison, we adjust its sparsity to match the trainable-parameter count of LoRA.

As shown in Table 11, DeFT-X adaptation of the base Qwen3-4B model yields substantial improvements across a diverse set of multilingual benchmarks. DeFT-X outperforms the LoRA adapter on four out of five benchmarks, indicating that parameter-efficient multilingual gains are not unique to low-rank updates. However, these gains come at a substantially higher training cost, requiring approximately twice the training time. DeFT-X adapters require a two-stage training approach: a first full fine-tuning pass over all parameters to identify the sparse mask (along with the SVD-based denoising step), followed by a second sparse fine-tuning pass. In contrast, LoRA requires only a single low-rank training stage. We thus treat DeFT-X as a strong but more expensive baseline and retain LoRA as our object of study, both because of its lower training cost and because its low-rank structure makes the spectral analysis in Section 5.3 directly applicable. Detailed results for each benchmark are provided in Appendix A.7.

5.5 Ablation Study

Curse of Multilinguality. We study the effect of increasing the number of training languages from 8 to 16, 32, and 52 while keeping the total amount of training data fixed². Figure 5 shows average performance across nine representative languages

²The 8-language setting follows the languages defined earlier. We cumulatively expand this set by adding zh, ar, hi, bn, tr, pl, id, ko to obtain 16 languages, followed by vi, th, fa, uk, ro, hu, nl, sv, el, he, bg, sr, ms, ta, te, ur to obtain 32 languages, and the remaining languages to obtain the full 52-language training set. The language sets provide a general mix of language families and scripts.

Model	MMMLU	MMLU ProX	M-IFEval (Strict)	Tool Use	MGSM
Qwen3-4B	66.9	45.6	70.8	56.2	76.3
+ Mono. Adapter	69.3	50.1	69.2	56.6	79.2
+ Multi. Adapter	69.9	50.6	71.2	58.1	78.8
Qwen3-14B	71.2	46.8	75.7	64.3	74.3
+ Mono. Adapter	71.8	50.2	75.9	65.7	76.9
+ Multi. Adapter	72.2	51.8	75.3	66.6	77.2

Table 12: Effect of the scaled LoRA adapter ($\lambda = 0.25$) across Qwen model sizes. Results averaged over **seen** non-English languages. **Bold** denotes the best performance.

Model	MMMLU	MMLU ProX	M-IFEval (Strict)	Tool Use	MGSM
Qwen3-4B	66.9	45.6	70.8	56.2	76.3
+ Bactrian-X Adapter	69.9	50.6	71.2	58.1	78.8
+ Aya Adapter	74.3	56.5	72.0	51.9	78.1

Table 13: Multilingual LoRA adapter with Bactrian-X vs. Aya training data on Qwen-4B ($\lambda = 0.25$). Results averaged over **seen** non-English languages. **Bold** denotes the best performance.

for the Qwen model. The 8-language setting performs best, while performance remains stable with 16 and 32 languages, indicating a saturation region where the model effectively balances multilingual generalization without degradation. However, a noticeable drop is observed when all 52 languages are included, suggesting that excessive multilinguality leads to interference.

Scalability Across Model Sizes. We also evaluate the scalability of our approach across different model sizes. Specifically, we conduct experiments using Qwen3 models at two scales: 4B and 14B parameters, as shown in Table 12. We observe that the improvements achieved on the 4B model largely transfer to the larger 14B model, although gains are smaller on some benchmarks, likely due to the stronger inherent multilingual capabilities of larger pretrained models.

Effect of Training Data. Table 13 compares adapters trained on Bactrian-X and Aya. Both improve over the base model on most benchmarks, confirming that multilingual adaptation generalizes across training corpora. The Aya adapter yields larger gains on knowledge-intensive tasks (MMMLU +7.4, MMLU-ProX +10.9), likely reflecting its richer instructional content. However, Aya degrades Tool Use below the base model (51.9 vs. 56.2), while Bactrian-X improves it (58.1), sug-

gesting that training data composition affects which task types benefit from adaptation.

6 Conclusions and Future Work

We showed that a single scaled parameter-efficient adapter trained on general-purpose multilingual instruction data, without task-specific supervision, can improve multilingual capabilities across a range of tasks. The gains are strongest on multilingual knowledge and tool-use benchmarks, while arithmetic reasoning and instruction-following show smaller or mixed improvements. The multilingual adapter also generalizes to unseen languages, although the magnitude of transfer varies across tasks and languages. Our analysis shows that adaptation is concentrated in language-processing layers while task-reasoning pathways remain relatively stable, and that evaluation-time scaling suppresses LoRA-induced intruder dimensions while preserving the multilingual signal required for transfer. Future work will explore language data mixing strategies and extend evaluation to additional model families.

Limitations

While the scaled multilingual adapter-based approach shows promising results across languages, its performance still depends on the availability and quality of general multilingual instruction-tuning data as well as a strong model pre-trained on good-quality multilingual data. Furthermore, our experiments focus primarily on knowledge and instruction-following benchmarks; extending the analysis to open-ended generation and dialog tasks would provide deeper insights into the generalization capabilities of the multilingual adaptation.

Acknowledgments

This work was done during the first author’s internship at Amazon. We thank Amazon for supporting this work and the anonymous reviewers for their thoughtful and constructive feedback.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. [Intrinsic dimensionality explains the effectiveness of language model fine-tuning](#). 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 7319–7328, Online. 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, 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.

Lucas Bandarkar, Benjamin Muller, Pritish Yuvraj, Rui Hou, Nayan Singhal, Hongjiang Lv, and Bing Liu. 2025. [Layer swapping for zero-shot cross-lingual transfer in large language models](#). *Preprint*, arXiv:2410.01335.

Alexandra Chronopoulou, Jonas Pfeiffer, Joshua Maynez, Xinyi Wang, Sebastian Ruder, and Priyanka Agrawal. 2024. [Language and task arithmetic with parameter-efficient layers for zero-shot summarization](#). In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 114–126, Miami, Florida, USA. Association for Computational Linguistics.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2023. [Language-family adapters for low-resource multilingual neural machine translation](#). In *Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pages 59–72, Dubrovnik, Croatia. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.

Antoine Dussolle, A. Cardeña, Shota Sato, and Peter Devine. 2025. [M-IFEval: Multilingual instruction-following evaluation](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6161–6176, Albuquerque, New Mexico. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). *Preprint*, arXiv:2009.03300.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.

- Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024a. [Lorahub: Efficient cross-task generalization via dynamic lora composition](#). *Preprint*, arXiv:2307.13269.
- Xu Huang, Wenhao Zhu, Hanxu Hu, Conghui He, Lei Li, Shujian Huang, and Fei Yuan. 2025. [Benchmax: A comprehensive multilingual evaluation suite for large language models](#). *Preprint*, arXiv:2502.07346.
- Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024b. [Mindmerger: Efficiently boosting llm reasoning in non-english languages](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 34161–34187. Curran Associates, Inc.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2023. [Editing models with task arithmetic](#). *Preprint*, arXiv:2212.04089.
- Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. [Bactrian-x: Multilingual replicable instruction-following models with low-rank adaptation](#). *Preprint*, arXiv:2305.15011.
- 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.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Zhiwen Ruan, Yixia Li, He Zhu, Longyue Wang, Weihua Luo, Kaifu Zhang, Yun Chen, and Guanhua Chen. 2025. [LayAlign: Enhancing multilingual reasoning in large language models via layer-wise adaptive fusion and alignment strategy](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1481–1495, Albuquerque, New Mexico. Association for Computational Linguistics.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. [Language models are multilingual chain-of-thought reasoners](#). *Preprint*, arXiv:2210.03057.
- Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. 2025. [Lora vs full fine-tuning: An illusion of equivalence](#). *Preprint*, arXiv:2410.21228.
- Sona Elza Simon and Preethi Jyothi. 2025. [DeFT-X: De-noised sparse fine-tuning for zero-shot cross-lingual transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1895–1909, Suzhou, China. Association for Computational Linguistics.
- G.W. Stewart and Ji-Guang Sun. 1990. *Matrix Perturbation Theory*. Academic Press.
- Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023. [Multilora: Democratizing lora for better multi-task learning](#). *Preprint*, arXiv:2311.11501.
- Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjie Wang, Fan Gao, Jinghui Lu, Yuang Jiang, Huitao Li, Xin Li, Kunyu Yu, Ruihai Dong, Shangding Gu, Yuekang Li, Xiaofei Xie, and 13 others. 2025. [Mmlu-prox: A multilingual benchmark for advanced large language model evaluation](#). *Preprint*, arXiv:2503.10497.
- Dongkeun Yoon, Joel Jang, Sungdong Kim, Seungone Kim, Sheikh Shafayat, and Minjoon Seo. 2024. [Lang-Bridge: Multilingual reasoning without multilingual supervision](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7522, Bangkok, Thailand. Association for Computational Linguistics.
- Yiran Zhao, Wenxuan Zhang, Huiming Wang, Kenji Kawaguchi, and Lidong Bing. 2025. [AdaMergeX: Cross-lingual transfer with large language models via adaptive adapter merging](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9785–9800, Albuquerque, New Mexico. Association for Computational Linguistics.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Appendix

A.1 Training Setup

Adapter Training. We train multilingual adapters using the Bactrian-X dataset (Li et al., 2023). We perform supervised fine-tuning with the TRL SFT Trainer for one epoch. Training was conducted on four NVIDIA A10G (24GB) GPUs with a per-device batch size of 4 and gradient accumulation set to 1, resulting in an effective batch size of 16. The total training time is approximately 7 hours. We optimize using AdamW with a learning rate of 2×10^{-4} , cosine learning rate scheduling, a warmup ratio of 0.01, and weight decay of 0.1. In the LoRA adaptation framework, updates are scaled by α/r during training and this scaling is also applied when merging the adapter into the base model, where α is the LoRA scaling factor and r is the rank. We set $r = 16$ and $\alpha = 32$. We apply LoRA to all linear projection layers in the Transformer blocks, namely *q_proj*, *k_proj*, *v_proj*, *o_proj*, *gate_proj*, *up_proj*, and *down_proj*. The prompt used for training Bactrian-X is given below.

Training Prompt

```
{instruction} {input}
Response: {output}
```

A.2 Evaluation Datasets

Table 14 provides a detailed overview of the evaluation benchmarks. The MMMLU³, MMLU-ProX⁴, Tool Use⁵, and MGSM⁶ are released under the MIT license, while M-IFEval⁷ uses the Apache-2.0 license. The evaluation prompts are given below.

MMMLU

Answer the following multiple choice question. The last line of your response should be of the following format: 'Answer: \$LETTER' (without quotes) where LETTER is one of ABCD. Think step by step before answering.

Question:
{question}

³<https://huggingface.co/datasets/openai/MMMLU>

⁴<https://huggingface.co/datasets/li-lab/MMLU-ProX>

⁵https://huggingface.co/datasets/LLaMAX/BenchMAX_Multiple_Functions

⁶<https://huggingface.co/datasets/juletxara/mgsm>

⁷<https://github.com/lightblue-tech/M-IFEval>

- A. {A}
- B. {B}
- C. {C}
- D. {D}

MMLU-ProX

Answer the following multiple choice question. The last line of your response should be of the following format: 'Answer: \$LETTER' (without quotes) where LETTER is one of ABCDEFGHIJ. Think step by step before answering.

Question:
{question}

- A. {A}
- B. {B}
- C. {C}
- D. {D}
- E. {E}
- F. {F}
- G. {G}
- H. {H}
- I. {I}
- J. {J}

M-IFEval

{prompt}

Tool Use

Respond to the human as helpfully and accurately as possible. You have access to the following tools:

```
[
  {
    "type": {type},
    "function":
      {
        "name": {name},
        "description": {description}
      }
  },
  {
    "type": {type},
    "function":
      {
```

Dataset	Languages	# Samples per language	Evaluation Metric
MMLU (Hendrycks et al., 2021)	ar, bn, de, es, fr, hi, id, it, ja, ko, pt, sw, yo, zh	14,079	Accuracy
MMLU-ProX (Xuan et al., 2025)	en, zh, ja, ko, fr, de, es, pt, ar, th, hi, bn, sw, af, cs, hu, id, it, mr, ne, ru, sr, te, uk, ur, vi, wo, yo, zu	11,829	Accuracy
M-IFEval (Dussolle et al., 2025)	en, es, fr, ja	en: 541, es: 115, fr: 235, ja: 172	Function name & arguments accuracy
Tool Use (Huang et al., 2025)	en, ar, bn, cs, de, es, fr, hu, ja, ko, ru, sr, sw, te, th, vi, zh	318	Strict accuracy, loose accuracy
MGSM (Shi et al., 2022)	en, es, fr, de, ru, zh, ja, th, sw, bn, te	250	Accuracy

Table 14: Detailed description of the evaluation benchmarks.

```

    "name": {name},
    "description": {description}
  }
]

```

Reminder to ALWAYS respond with a valid json enclosed within <tool_call> tags. For example:

```

<tool_call>
{"name": function name, "parameters":
dictionary of argument name and its value}
</tool_call>

```

Do not use variables or output comments.

{prompt}

MGSM

Answer the following math question. The last line of your response should be of the following format: 'Answer: numerical_answer'. Think step by step before answering.

Question:
{question}

A.3 Weight Analysis Based on Projection-Type

To further analyze the observations from Section 5.3, we decompose the weight changes by projection type. Figure 7 shows the max norm of the weight difference for each of the seven lin-

ear projections: q_proj , k_proj , v_proj , o_proj , $gate_proj$, up_proj , $down_proj$ across all transformer layers, comparing the LoRA-adapted model ($\lambda = 0.25$) with the full fine-tuned model.

For the LoRA adapter, $gate_proj$ and o_proj exhibit the largest variability in weight differences, concentrated in the initial and final layers, while the remaining attention projections (q_proj , k_proj , v_proj) show comparatively smaller changes. Among the MLP projections, $gate_proj$ consistently shows larger modifications than up_proj and $down_proj$. Full fine-tuning produces uniformly smaller weight differences (by magnitude) across all projection types and layers, with less concentration at the extremes.

A.4 Intruder Dimensions

Intruder dimensions are formally defined as follows: An intruder dimension (Shuttleworth et al., 2025) is defined as a singular vector y_j from the fine-tuned weight matrix W_{tuned} that exhibits low cosine similarity with all singular vectors x_i of the pre-trained weight matrix W_0 , calculated as $\max_i(\cos(y_j, x_i)) < \epsilon$ (where ϵ is a predefined similarity threshold). These dimensions represent entirely new, orthogonal directions of parameter updates introduced during fine-tuning, rather than minor adjustments to existing pre-trained features.

Figure 6 reveals that intruder dimensions are most concentrated in the MLP projections ($gate_proj$, up_proj , o_proj), consistent with the layer-wise norm analysis in Section 5.3. This pattern holds across a range of cosine similarity thresholds ϵ , confirming the robustness of the finding.

This per-component view reinforces the finding from Section 5.3 that the LoRA adapter makes tar-

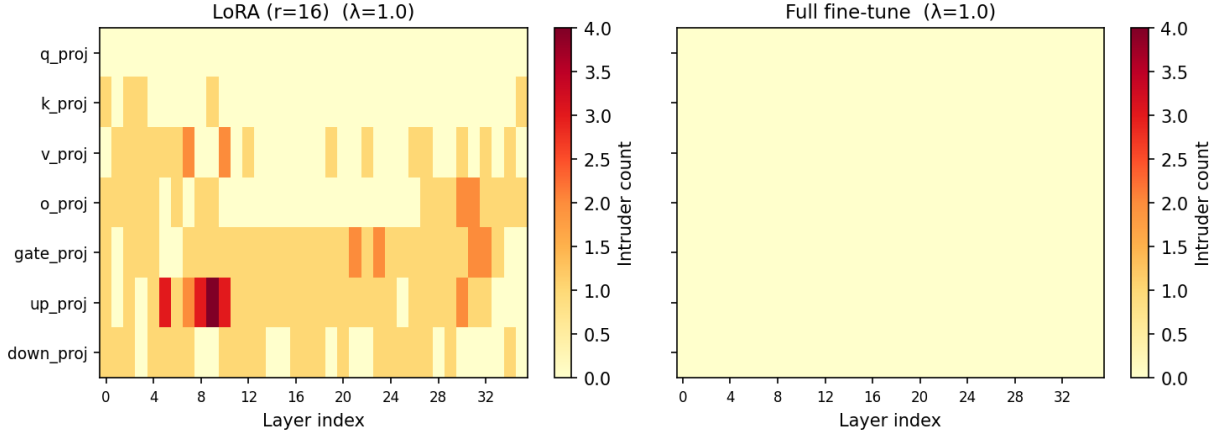


Figure 6: Layer-wise Intruder dimensions for different weight matrices in Qwen3-4B for rank=16 and alpha=32 vs Full Fine-Tuning

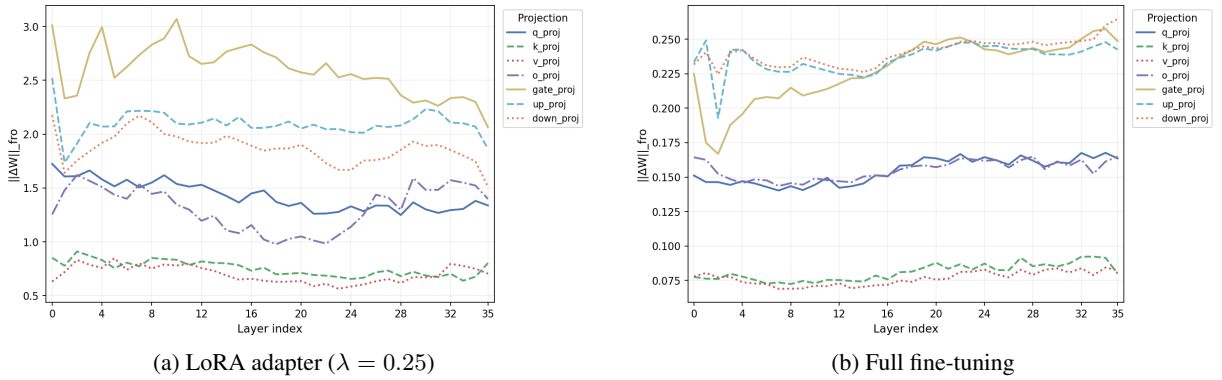


Figure 7: Layer-wise max norm of weight difference between the multilingual-adapted and the Qwen-4B model, broken down by projection type.

geted modifications to specific components within the language-processing layers rather than distributing changes broadly. The prominence of gate_proj modifications is notable, as the gating mechanism controls information flow through the MLP block, suggesting the LoRA adaptation primarily adjusts which features are amplified or suppressed at the token level in the language-processing layers.

A.5 Effect of Cosine Similarity Threshold on Intruder Dimension Analysis

We analyze the sensitivity of the intruder dimension count to the cosine similarity threshold ϵ . Figure 8a shows that the total number of intruder dimensions increases monotonically with ϵ for LoRA ($\lambda = 1.0$), as a higher threshold admits more singular vectors as intruders. This trend is consistent across all λ values; decreasing λ reduces the intruder count, since the scaling factor suppresses their contribution to the adapted weights. As shown in Figure 8b, this monotonic growth is uniform across all projection modules, suggesting that the

sensitivity to ϵ is a global property of LoRA adaptation rather than an artifact of any particular module.

A.6 Intruder Dimension Suppression Theorem

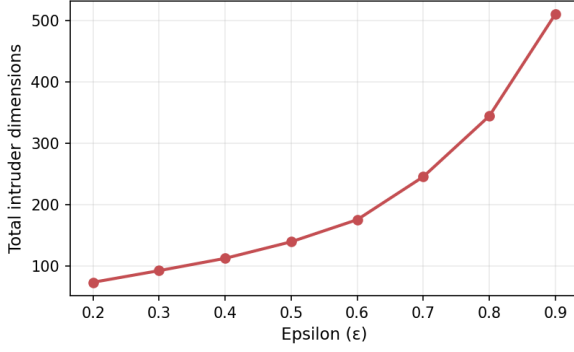
Theorem 1. Let $W_0 = U\Sigma V^\top$ with singular values $\sigma_1 \geq \dots \geq \sigma_d$, and let BA be a rank- r LoRA update with operator norm $\rho = \|BA\|_2$. For any cosine similarity threshold $\epsilon \in (0, 1)$ and integer $k \geq 1$, let $\Delta_{\min}^{(k)} = \min_{j \leq k} \min_{i \neq j} |\sigma_j - \sigma_i|$. If

$$\lambda < \frac{\sqrt{1 - \epsilon^2} \cdot \Delta_{\min}^{(k)}}{(1 + 2\sqrt{1 - \epsilon^2}) \cdot \rho},$$

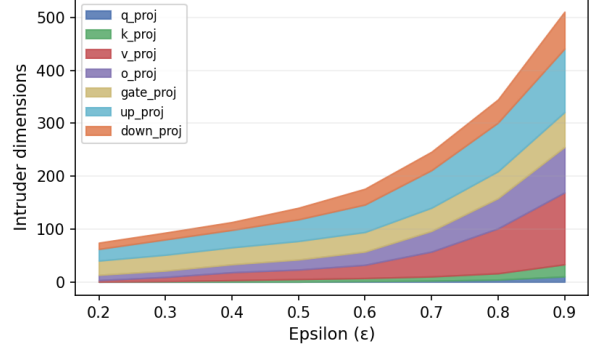
then $W(\lambda) = W_0 + \lambda \cdot BA$ contains no intruder dimensions among its top- k singular vectors.

Proof intuition. The proof shows that when λ is small enough, the singular vectors of $W(\lambda) = W_0 + \lambda \cdot BA$ cannot deviate far enough from those of W_0 to become intruder dimensions.

Step 1: Singular values stay separated. By Weyl’s inequality (Stewart and Sun, 1990), adding



(a) Total intruder dimensions vs. ϵ .



(b) Intruder dimensions broken down by projection module.

Figure 8: Intruder dimension count increases monotonically with ϵ , consistently across all projection modules for LoRA ($\lambda = 1.0$).

$\lambda \cdot BA$ shifts each singular value by at most $\lambda\rho$:

$$|\sigma_j(W(\lambda)) - \sigma_j(W_0)| \leq \lambda\rho.$$

If $\lambda\rho < \Delta_{\min}^{(k)}/2$, no two top- k singular values can merge, so each singular vector of $W(\lambda)$ remains identifiable with a unique pretrained one.

Step 2: Singular vectors rotate boundedly. Wedin’s theorem (Stewart and Sun, 1990) bounds the angular displacement when singular values are separated by a gap δ_j :

$$\sin \angle(\tilde{u}_j, u_j) \leq \frac{\lambda\rho}{\delta_j}.$$

From Step 1, $\delta_j \geq \Delta_{\min}^{(k)} - 2\lambda\rho$, giving:

$$\sin \angle(\tilde{u}_j, u_j) \leq \frac{\lambda\rho}{\Delta_{\min}^{(k)} - 2\lambda\rho}.$$

A larger spectral gap means more stability; a larger adapter norm means more rotation.

Step 3: Bounded rotation implies no intruders. Requiring $\cos \angle(\tilde{u}_j, u_j) \geq \epsilon$ is equivalent to:

$$\sin \angle(\tilde{u}_j, u_j) \leq \sqrt{1 - \epsilon^2}.$$

Solving the inequality from Step 2 against this threshold:

$$\frac{\lambda\rho}{\Delta_{\min}^{(k)} - 2\lambda\rho} \leq \sqrt{1 - \epsilon^2}$$

yields the bound in the theorem. Every top- k singular vector of $W(\lambda)$ retains cosine similarity $\geq \epsilon$ with its pretrained counterpart, so none qualifies as an intruder dimension.

Remark. The bound provides a sufficient condition and is conservative by construction: it applies uniformly to all weight matrices and uses worst-case spectral gaps. In practice, the empirically optimal λ may be larger than the theoretical λ^* because (i) not all weight matrices are equally sensitive to perturbation, (ii) the LoRA update BA may be partially aligned with the pretrained singular subspace rather than fully orthogonal, and (iii) the intruder dimension threshold ϵ used in practice may be more permissive than the worst-case assumption.

A.7 Generalization to Sparse Fine-Tuning

While our main experiments use LoRA, we investigate whether the benefits of scaled parameter addition extend to other multilingual adaptation methods. Specifically, we apply the same approach to DeFT-X (Simon and Jyothi, 2025), a sparse fine-tuning-based adaptation method. The detailed per-language results for each benchmark are provided in Tables 18 - 22.

Benchmark	Method	en	es	fr	de	it	pt	ru	cs	ja	Avg. Non-En	Avg.
MMMLU	Qwen3-14B	73.6	71.7	70.4	70.3	71.2	73.6	–	–	70.1	71.2	71.5
	+ Mono. Adapter	76.2	72.3	71.0	70.9	72.1	73.9	–	–	70.3	71.8	72.4
	+ Multi. Adapter	76.4	73.1	71.9	71.0	72.3	73.1	–	–	71.8	72.2	72.8
MMLU-Pro-X	Qwen3-14B	47.9	48.0	47.3	45.2	47.7	48.0	47.2	45.4	45.9	46.8	47.0
	+ Mono. Adapter	52.2	51.2	50.8	49.3	50.8	49.9	50.7	49.7	49.2	50.2	50.4
	+ Multi. Adapter	53.7	52.7	53.6	52.4	52.7	53.0	50.2	50.0	50.1	51.8	52.0
MGSM	Qwen3-14B	78.7	80.4	73.8	72.4	–	–	77.8	–	67.1	74.3	75.0
	+ Mono. Adapter	79.6	80.7	76.5	76.3	–	–	79.2	–	71.6	76.9	77.3
	+ Multi. Adapter	80.9	81.3	76.0	77.3	–	–	79.1	–	72.0	77.2	77.8
M-IFEval (Strict)	Qwen3-14B	85.2	86.0	82.4	–	–	–	–	–	58.8	75.7	78.1
	+ Mono. Adapter	84.9	89.0	81.8	–	–	–	–	–	57.1	75.9	78.2
	+ Multi. Adapter	83.8	89.4	79.7	–	–	–	–	–	56.8	75.3	77.4
Tool Use	Qwen3-14B	65.6	63.0	65.3	63.1	–	–	67.5	64.8	61.9	64.3	64.5
	+ Mono. Adapter	71.0	64.8	67.1	65.6	–	–	68.9	63.1	64.6	65.7	66.4
	+ Multi. Adapter	72.1	66.6	66.2	66.9	–	–	68.3	65.9	65.9	66.6	67.4

Table 15: Results of multilingual adaptation on Qwen3-14B across benchmarks. **Bold** indicates the best performance per benchmark and language.

Benchmark	Method	en	es	fr	de	it	pt	ru	cs	ja	Avg. Non-En	Avg.
MMMLU	Qwen3-4B	71.7	68.5	67.3	65.6	67.8	68.3	–	–	64.1	66.9	67.6
	+ FFT ($\lambda = 1.00$)	24.9	6.0	21.3	4.6	8.0	18.0	–	–	16.2	12.3	14.1
	+ FFT ($\lambda = 0.25$)	80.0	74.9	75.2	75.0	75.7	74.6	–	–	72.7	74.7	75.4
MMLU-Pro-X	Qwen3-4B	46.0	46.7	46.5	44.1	46.2	46.1	46.5	44.5	44.5	45.6	45.7
	+ FFT ($\lambda = 1.00$)	11.5	8.6	10.8	7.3	8.9	11.5	1.9	8.2	11.3	8.5	8.9
	+ FFT ($\lambda = 0.25$)	61.9	58.5	58.1	57.9	57.5	57.5	57.3	55.3	54.5	57.1	57.6
MGSM	Qwen3-4B	83.6	80.0	77.8	76.9	–	–	76.9	–	69.8	76.3	77.5
	+ FFT ($\lambda = 1.00$)	7.6	7.6	5.6	6.8	–	–	7.2	–	2.4	5.9	6.2
	+ FFT ($\lambda = 0.25$)	85.9	80.8	79.6	78.2	–	–	81.8	–	73.8	78.8	80.0
M-IFEval (Strict)	Qwen3-4B	81.1	81.7	79.7	–	–	–	–	–	51.1	70.8	73.4
	+ FFT ($\lambda = 1.00$)	20.2	33.0	17.5	–	–	–	–	–	13.4	21.3	21.0
	+ FFT ($\lambda = 0.25$)	81.3	89.4	75.5	–	–	–	–	–	45.8	70.2	73.0
Tool Use	Qwen3-4B	59.6	54.7	58.9	54.4	–	–	58.2	57.1	54.0	56.2	56.7
	+ FFT ($\lambda = 1.00$)	2.5	1.2	1.1	1.2	–	–	1.1	0.5	0.3	0.9	1.1
	+ FFT ($\lambda = 0.25$)	52.6	51.6	46.5	47.7	–	–	55.4	56.3	52.3	51.6	51.8

Table 16: Results of full finetuning on Qwen3-4B across benchmarks. **Bold** indicates the best performance per benchmark and language.

Benchmark	Method	en	es	fr	de	it	pt	ru	cs	ja	Avg. Non-En	Avg.
MMMLU	Qwen3-4B	71.7	68.5	67.3	65.6	67.8	68.3	–	–	64.1	66.9	67.6
	+ Multi. Adapter	70.2	62.3	61.9	61.7	61.8	61.6	–	–	53.9	60.5	61.9
MMLU-Pro-X	Qwen3-4B	46.0	46.7	46.5	44.1	46.2	46.1	46.5	44.5	44.5	45.6	45.7
	+ Multi. Adapter	46.1	39.7	38.3	37.2	41.2	39.5	41.0	36.1	31.8	38.1	39.0
MGSM	Qwen3-4B	83.6	80.0	77.8	76.9	–	–	76.9	–	69.8	76.3	77.5
	+ Multi. Adapter	60.9	24.9	26.2	22.6	–	–	39.1	–	13.2	25.2	31.2
M-IFEval (Strict)	Qwen3-4B	81.1	81.7	79.7	–	–	–	–	–	51.1	70.8	73.4
	+ Multi. Adapter	44.2	59.1	43.0	–	–	–	–	–	26.2	42.8	43.1
Tool Use	Qwen3-4B	59.6	54.7	58.9	54.4	–	–	58.2	57.1	54.0	56.2	56.7
	+ Multi. Adapter	31.7	36.2	35.5	43.1	–	–	45.3	47.5	45.3	42.1	40.7

Table 17: Results of Qwen3-4B with LoRA Multilingual Adapter with scaling factor $\lambda = 1$. **Bold** indicates the best performance per benchmark and language.

Method	en	es	fr	de	it	pt	ja	Avg. Non-En	Avg.
Qwen3-4B	71.7	68.5	67.3	65.6	67.8	68.3	64.1	66.9	67.6
+ LoRA Multi. Adapter	74.4	70.9	69.5	68.4	70.1	70.2	70.0	69.9	70.5
+ DeFT-X Multi. Adapter	80.4	77.2	75.9	75.8	76.6	74.6	74.7	75.8	76.4

Table 18: Comparison of multilingual adaptation methods on the MMMLU benchmark for seen languages using Qwen3-4B. We compare our LoRA-based multilingual adapter with DeFT-X. $\lambda = 0.25$ was used. **Bold** indicates the best performance.

Method	en	es	fr	de	it	pt	ru	cs	ja	Avg. Non-En	Avg.
Qwen3-4B	46.0	46.7	46.5	44.1	46.2	46.1	46.5	44.5	44.5	45.6	45.7
+ LoRA Multi. Adapter	52.6	51.9	51.4	50.0	51.9	52.4	49.5	48.6	49.0	50.6	50.8
+ DeFT-X Multi. Adapter	65.4	61.5	60.4	59.3	60.3	59.6	59.9	57.1	57.5	59.4	60.1

Table 19: Comparison of multilingual adaptation methods on the MMLU-ProX benchmark for seen languages using Qwen3-4B. We compare our LoRA-based multilingual adapter with DeFT-X. $\lambda = 0.25$ was used. **Bold** indicates the best performance.

Method	en		es		fr		ja		Avg. Non-En		Avg.	
	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose	Strict	Loose
Qwen3-4B	81.1	84.0	81.7	84.6	79.7	82.1	51.0	54.8	70.8	73.8	73.4	76.4
+ LoRA Multi. Adapter	78.6	81.9	87.5	89.4	78.3	81.1	47.7	50.3	71.2	73.6	73.0	75.7
+ DeFT-X Multi. Adapter	83.9	87.1	88.7	90.4	76.6	79.1	52.3	57.6	72.5	75.7	75.4	78.6

Table 20: Comparison of multilingual adaptation methods on the M-IFEval benchmark for seen languages using Qwen3-4B. We compare our LoRA-based multilingual adapter with DeFT-X. $\lambda = 0.25$ was used. We report both strict and loose accuracy. **Bold** indicates the best performance.

Method	en	es	fr	de	ru	cs	ja	Avg. Non-En	Avg.
Qwen3-4B	59.6	54.7	58.9	54.4	58.2	57.1	54.0	56.2	56.7
+ LoRA Multi. Adapter	62.7	60.6	60.3	56.5	59.9	54.4	57.1	58.1	58.8
+ DeFT-X Multi. Adapter	61.3	59.4	56.3	52.2	60.1	56.3	56.3	56.8	57.4

Table 21: Comparison of multilingual adaptation methods on the Tool Use benchmark for seen languages using Qwen3-4B. We compare our LoRA-based multilingual adapter with DeFT-X. $\lambda = 0.25$ was used. **Bold** indicates the best performance.

Method	en	es	fr	de	ru	ja	Avg. Non-En	Avg.
Qwen3-4B	83.6	80.0	77.8	76.9	76.9	69.8	76.3	77.5
+ LoRA Multi. Adapter	85.8	80.1	78.7	79.1	77.8	71.6	77.4	78.8
+ DeFT-X Multi. Adapter	87.2	81.2	78.8	81.2	74.0	82.4	79.5	80.8

Table 22: Comparison of multilingual adaptation methods on the MGSM benchmark for seen languages using Qwen3-4B. We compare our LoRA-based multilingual adapter with DeFT-X. $\lambda = 0.25$ was used. **Bold** indicates the best performance.