Efficient Compositional Multi-tasking for On-device Large Language Models

Ondrej Bohdal[1], Mete Ozay[1], Jijoong Moon[2], Kyeng-Hun Lee[2], Hyeonmok Ko[2], Umberto Michieli[1]

[1]Samsung R&D Institute UK, United Kingdom, [2]Samsung Research, South Korea

Correspondence: o.bohdal.1@samsung.com

Abstract

Adapter parameters provide a mechanism to modify the behavior of machine learning models and have gained significant popularity in the context of large language models (LLMs) and generative AI. These parameters can be merged to support multiple tasks via a process known as task merging. However, prior work on merging in LLMs, particularly in natural language processing, has been limited to scenarios where each test example addresses only a single task. In this paper, we focus on on-device settings and study the problem of text-based compositional multi-tasking, where each test example involves the simultaneous execution of multiple tasks. For instance, generating a translated summary of a long text requires solving both translation and summarization tasks concurrently. To facilitate research in this setting, we propose a benchmark comprising four practically relevant compositional tasks. We also present an efficient method (Learnable Calibration) tailored for on-device applications, where computational resources are limited, emphasizing the need for solutions that are both resource-efficient and high-performing. Our contributions lay the groundwork for advancing the capabilities of LLMs in real-world multi-tasking scenarios, expanding their applicability to complex, resource-constrained use cases. Project page: https://ondrejbohdal.github.io/CompositionalMultitaskingLLMs.

1 Introduction

Large language models (LLMs) have revolutionized natural language processing (NLP) by demonstrating remarkable capabilities across a wide range of text-based tasks (Zhao et al., 2023; Minaee et al., 2024). As foundational models, LLMs can be fine-tuned to excel in various downstream applications, such as question answering (Sticha et al., 2024), summarization (Liu et al., 2024b), translation (Zhu et al., 2023), and beyond (Shu et al., 2024; Rothe et al., 2021). A common approach to fine-tuning or adapting LLMs is through parameter-efficient fine-tuning (PEFT), where relatively few additional parameters, often referred to as adapters, are inserted into the model (Han et al., 2024; Ding et al., 2023). Among these methods, Low-Rank Adapters (LoRA) (Hu et al., 2022) have gained prominence due to their efficiency and ease of integration. LoRA trains low-rank matrices inserted into the LLM's layers, enabling efficient adaptation to specific tasks. Typically, models are fine-tuned for one task at a time, producing task-specific models or *experts* that can be merged to obtain a final model supporting multiple tasks (Wortsman et al., 2022; Ilharco et al., 2023). A *task* may correspond to, for example, (i) different domains of expertise (*e.g.*, math versus biology Q&A), (ii) different downstream tasks (*e.g.*, summarization versus translation), or (iii) different optimization convergence points (*e.g.*, results from separate training runs). We focus on LLM customization for different downstream tasks.

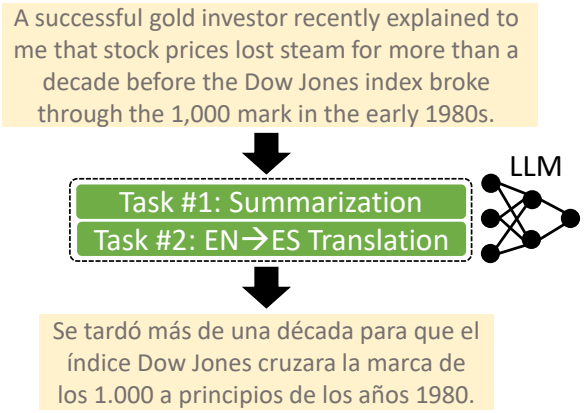


Figure 1: **Compositional multi-tasking** involves performing multiple tasks simultaneously, such as summarization and translation. The challenge lies in executing all tasks jointly within a single inference pass for optimal efficiency, rather than performing them separately through multiple inferences.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 28141–28165
November 4-9, 2025

Model merging for multi-tasking in LLMs has gained significant attention (Yang et al., 2024b) due to its simplicity and ability to support multiple tasks without requiring expensive retraining or fine-tuning. By averaging or combining weights, merged models can perform tasks they were not explicitly trained on, achieving competitive performance across tasks. Techniques such as linear merging (Wortsman et al., 2022; Ilharco et al., 2023), TIES (Yadav et al., 2024), and DARE (Yu et al., 2024) have been proposed to enhance merging efficacy.

After merging, the resulting model can switch between tasks based on user instructions. For instance, a merged model for translation and summarization can either translate or summarize input text. However, existing merging methods are limited to scenarios where each test example involves only one task. This limitation becomes apparent in applications where simultaneous execution of tasks is required—for example, summarizing a text while translating the summary into a different language.

We study compositional multi-tasking, a paradigm where LLMs are asked to perform multiple tasks simultaneously for a single input. As illustrated in Figure 1, compositional multi-tasking requires models to handle tasks such as generating a translated summary in one inference pass. While larger LLMs can directly follow instructions with an acceptable degree of success (Qin et al., 2024), on-device LLMs typically utilize adapters to provide acceptable output quality (Gunter et al., 2024). We focus on the novel research direction of enabling compositional multi-tasking abilities in on-device LLMs. A simple multi-step pipeline—where one expert model performs the first task and other expert models perform the additional tasks—can achieve this (as we verify in Section 5), but it is inefficient, requiring several inference passes and longer processing times. In this paper, we study how we can achieve strong performance with a single inference pass, enabling more practical applications.

Practical real-world use cases of compositional multi-tasking for on-device LLMs abound. For example, joint summarization and translation can be valuable when people move abroad and need to understand the overall message of a long text in the local language. Other examples include proposing suitable replies in cross-lingual scenarios, and combinations of summarization or reply with tone adjustment to suit specific contexts. To study the novel setting of compositional multi-tasking for on-device LLMs, we developed a benchmark comprising four practical compositional tasks. Each task combines a main task (*e.g.*, summarization or reply generation) with an auxiliary task (*e.g.*, translation or tone adjustment). The main task determines the main functionality, while the auxiliary task modifies the output to meet additional requirements. Performance is evaluated based on the main task, with the auxiliary task altering the ground-truth texts.

Mobile devices have limited computational resources (Dhar et al., 2021), but their users would benefit from on-device compositional multi-tasking functionalities. Deploying such systems on personal devices (*e.g.*, smartphones) introduces several *desiderata*: 1) high accuracy: ensure strong task performance; 2) low inference time: achieve compositional multi-tasking in a single inference pass over the LLM; 3) low storage requirements: minimize additional storage needs by reusing task-specialized adapters instead of creating new ones. Therefore, training a separate adapter for each compositional task is impractical due to storage constraints on user devices. Nonetheless, the inefficient solutions that either perform multiple inference passes using different adapters or introduce an additional adapter to solve the compositional task will serve as reference points if we do not consider on-device restrictions.

Further, we show existing merging strategies fail to address compositional multi-tasking in LLMs, opening the question of how we can efficiently solve the challenge. As a solution, we propose Learnable Calibration that achieves strong performance with a single inference pass by combining already-available task-specific adapters and learning a small number of additional parameters. This approach maintains resource efficiency while meeting the performance demands of compositional multi-tasking.

To summarize, our key contributions are: 1) We introduce the novel challenge of enabling compositional multi-tasking in on-device LLMs; 2) We develop a new benchmark with four compositional tasks to evaluate performance of different approaches; 3) We propose a novel method, Learnable Calibration, which achieves high performance with minimal computational and storage overhead.

2 Related Work

Parameter-efficient Fine-tuning (PEFT). PEFT methods enable adapting foundational models by training only a small number of parameters, making them computationally resource-efficient (Han et al., 2024; Ding et al., 2023). Among these, LoRA (Hu et al., 2022) has become the most widely adopted approach. LoRA introduces compact, low-rank matrices into model layers, which are trained while keeping the rest of the model frozen. This approach ensures minimal additional storage and computational requirements, making it particularly suitable for large-scale foundational models (Mao et al., 2025). Various extensions of LoRA have been developed to further improve its performance, for example, DoRA (Liu et al., 2024a), AdaLoRA (Zhang et al., 2023b), Delta-LoRA (Zi et al., 2023). Other approaches include BitFit (Zaken et al., 2022), which trains the bias parameters of the foundational models.

Model Merging. Model merging enables multi-tasking by combining multiple task-specific models into a single model. Early works (Wortsman et al., 2022; Ilharco et al., 2023) demonstrated the feasibility of linear merging, where the weights of fine-tuned expert models are combined as a weighted average. Building on this, advanced merging techniques such as TIES (Yadav et al., 2024), DARE (Yu et al., 2024), Slerp (White, 2016), and adaptive approaches such as LoraHub (Huang et al., 2024), LM-Cocktail (Xiao et al., 2024) and Differentiable Adaptive Merging (DAM) (Gauthier-Caron et al., 2024) have been proposed. TIES resets parameters that changed too little, resolves sign conflicts, and merges only the parameters that align with the selected sign. DARE first drops part of the weight changes and then rescales the remaining ones. Slerp projects the weight changes on the sphere and then performs linear interpolation. LoraHub finds weights for linear merging via gradient-free hyperparameter optimization on a few examples. LM-Cocktail selects weights for linear merging according to loss on several examples. DAM learns column-wise scaling using backpropagation to merge multiple models.

Merging has also been explored at the level of individual adapters, allowing finer-grained combinations of model parameters (Tang et al., 2023; Zhang et al., 2023a). While model merging has gained traction in both NLP (Hammoud et al., 2024) and computer vision (Yang et al., 2024b), more advanced use cases have been developed in the vision domain, such as joint subject-style personalization with ZipLoRA (Shah et al., 2024) and LoRA.rar (Shenaj et al., 2025). ZipLoRA is similar to DAM in learning column-wise scaling coefficients via backpropagation, while LoRA.rar predicts these coefficients using a pre-trained hypernetwork. In NLP, model merging has primarily focused on standard multi-tasking, where merged models handle multiple tasks individually, with each test example addressing a single task (Yang et al., 2024b). In contrast, we tackle the challenge of compositional simultaneous multi-tasking, where a single test example requires executing multiple tasks concurrently.

On-device LLMs. LLMs often contain billions of parameters, necessitating significant resources, such as multiple high-end GPUs, for inference (Borzunov et al., 2024). However, many valuable LLM use cases involve sensitive data stored on resource-constrained devices, making it desirable to perform computations locally and avoid transferring data to remote servers (Dhar et al., 2021). For example, users may wish to summarize private conversations or generate personalized replies while maintaining data privacy. To support such use cases, smaller LLMs have been developed for on-device deployment. These models leverage compression techniques and smaller parameter sizes to enable efficient on-device inference. Prominent examples include LLaMA 3.2 1B (Dubey et al., 2024), Qwen2.5 1.5B (Yang et al., 2024a; Qwen Team, 2024), and StableLM2 1.6B (Bellagente et al., 2024). Our work assumes that model size is the key constraint for on-device deployment, as current mobile-compatible LLMs typically range from 1–3B parameters. Single-task LoRAs are typically stored on-device to enable an LLM to support the individual tasks, rather than relying on instruction following (Gunter et al., 2024). On-device LLMs also have a limited context window size, making in-context learning less suitable (Dong et al., 2024).

3 Benchmark

We focus on the novel problem of enabling compositional multi-tasking in on-device LLMs, which requires a suitable benchmark that includes data for both training and evaluation. To facilitate research in this domain, we develop a benchmark targeting practically valuable compositional tasks. Specifically, our benchmark includes four task combina-

tions: summarization and reply suggestion (often referred to as "smart reply") as the main task T_1, combined with translation or tone adjustment (*i.e.*, rewriting) as the auxiliary task T_2. A generic task T maps an input text x to an output text y; *i.e.*, $T(\cdot) : x \mapsto y$. For instance, x may represent a long text or part of a conversation, with the goal of producing a summary or an appropriate reply in a specific language or tone. In general, N tasks can be used, for which the compositional task would be defined as $T^C_{[N]}(x) \stackrel{\text{def}}{=} T_N(\ldots T_2(T_1(x)))$, where

$$T_N(\ldots T_2(T_1(x))) : x \mapsto y_1 \mapsto y_2 \mapsto \ldots \mapsto y_N. \quad (1)$$

Input x is first processed by T_1 to produce y_1, T_2 subsequently transforms y_1 into y_2, *etc.* We primarily use $N = 2$, but also consider $N = 3$ within additional analyses.

Our benchmark features three translation settings (English to Spanish, French, or German) and four tone adjustments (professional, casual, witty, and paraphrase), resulting in fourteen subtasks in total. Existing summarization and dialogue datasets were repurposed for compositional multi-tasking using specialized models. We manually checked outputs to ensure quality and selected the most appropriate models. Table 1 provides details about the tasks and the dataset splits, while Figure 2 illustrates the compositional tasks. Source datasets were selected based on their suitability and licensing terms.

Table 1: **Dataset statistics.** Number of examples in our benchmark across different splits. Note that compositional tasks retain the same number of examples as their respective main tasks. For instance, summarization + translation and summarization + tone adjustment have the same number of examples as the summarization task itself, and similarly, when reply is the main task.

Task	Training	Validation	Test
Summarization	12,460	500	1,500
Reply	225,061	1,000	1,000
Translation	196,026	4,231	5,571
Tone adjustment	2,245	321	642

Summarization + Translation. We use the DialogSum dataset (Chen et al., 2021), which contains English dialogue summaries. To enable compositional multi-tasking, ground-truth summaries are translated to target languages (Spanish, French, German) using the OpusMT model (Tiedemann et al., 2023; Tiedemann and Thottingal, 2020).

Summarization + Tone Adjustment. We change the tone of the ground-truth summaries available in DialogSum via the publicly available RedPajama-INCITE-Base 3B model (Weber et al., 2024) fine-tuned specifically for tone adjustment and paraphrasing (Utsav, 2023). We use the corresponding prompt that was predefined for tone adjustment.

Reply + Translation. For reply tasks, we use the Synthetic Persona Chat dataset (Jandaghi et al., 2024) that contains a large number of dialogues. Each dialogue is converted into pairs of consecutive sentences, where the first sentence serves as the context and the second as the ground-truth reply. Similarly, we use the same machine translation models to translate the ground-truth outputs in target languages. To reduce computational overhead during evaluation, we sample a subset of context-reply pairs for validation and testing.

Reply + Tone Adjustment. Tone-adjusted replies are generated from the Synthetic Persona Chat dataset using the RedPajama-INCITE-Base 3B model, *i.e.*, the model specialized for tone adjustment (adj.) and paraphrasing.

In addition to compositional tasks, we train single task adapters on suitable datasets. Summarization and reply tasks use DialogSum and Synthetic Persona Chat, respectively. Translation tasks utilize the TED Talks dataset (Qi et al., 2018) (English to Spanish, French, and German). Tone adj. tasks are trained on the Sound Natural rephrasing dataset (Einolghozati et al., 2020), where rephrased content is further processed using the tone adj. model.

We adopt metrics commonly used in the literature to evaluate performance. For compositional tasks based on summarization, we report the ROUGE-L (R-L) (Lin, 2004) metric (%, ↑), which measures the longest common subsequence between the generated and ground truth texts. Additional evaluations via ROUGE-1 and ROUGE-2 are shown in Appendix C. For compositional tasks based on reply, we report the Weighted ROUGE (W-R) score (Zhang et al., 2021) metric (%, ↑), which is computed as:

$$\text{W-R} = \frac{\text{ROUGE-1}}{6} + \frac{\text{ROUGE-2}}{3} + \frac{\text{ROUGE-3}}{2}. \quad (2)$$

Weighted ROUGE mitigates sensitivity to small sequence-length changes and has been shown (Zhang et al., 2021) to correlate well with user

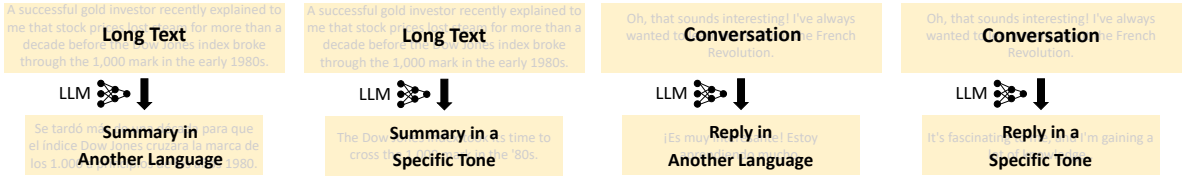


Figure 2: **Overview of the four compositional tasks in our benchmark.** The tasks include three translation settings (English to Spanish, French, and German) and four tone variations (professional, casual, witty, and neutral paraphrase), leading to fourteen sub-tasks overall.

click-through rates, reflecting how often the users utilize the recommended reply.

In addition, we include evaluation via LLM Judge (LLM-J) (Zheng et al., 2023). LLM Judge mimics human evaluation by assessing outputs using a pretrained LLM. We employ Llama 3.1 70B Instruct model (Dubey et al., 2024) with carefully designed prompts (detailed in Appendix A), assigning a binary outcome to each example.

4 Method

Existing approaches that can be used for compositional multi-tasking in on-device LLMs are either inefficient or have low performance, as we show in our analysis. How can we *achieve both efficiency and good performance* so that we can target on-device applications where computational resources and storage are restricted? We propose a new method that addresses this challenge. In particular, we develop an efficient solution that achieves comparable or superior performance to inefficient baselines, such as the multi-step application of adapters or a new adapter trained specifically for each compositional task (referred to as a "joint-expert" adapter).

4.1 Preliminaries

In typical on-device scenarios, we assume access to a small LLM (*e.g.*, 1B–3B parameters) and low-rank adapters (LoRAs) stored on devices to support tasks such as summarization or translation. Indeed, while prompting can be sufficient with large-scale models, LoRAs are commonly used in on-device settings to provide excellent performance in the desired tasks (Gunter et al., 2024). Each LoRA typically requires around 20–100 MB of storage. To handle compositional tasks, we aim to introduce only a negligible number of additional parameters to be stored on devices.

LoRAs provide an efficient way to adapt LLMs by adding learnable low-rank factorized matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ (where $r \ll \min(d,k)$) to the model's frozen weights $W_0 \in \mathbb{R}^{d\times k}$. The adjusted forward pass gives:

$$h = W_0 x + \Delta W x = W_0 x + BAx. \quad (3)$$

In line with literature and real-world usage (Gunter et al., 2024), we assume LoRAs $\{B_i, A_i\}_{i=1}^N$ for all N tasks $\{T_i\}_{i=1}^N$ are already available on-device.

4.1.1 Baseline and Merging Strategies

We evaluate several naïve baselines: 1-2) Using only the main (or auxiliary) task LoRA. 3) In-context learning that provides an example input and output but consumes more of the context window. 4) Multi-step application of LoRAs, where an output for a task is passed as input for another task. 5) Joint-expert LoRA trained specifically for the compositional task. In all cases, prompts specifying the tasks are included as part of the input.

We also consider model merging strategies, which have shown success in standard multi-tasking where test examples involve a single task at a time. In particular, we consider the following merging strategies applied to LoRA adapters: 1) Linear, *i.e.*, Model Soup (Wortsman et al., 2022); 2) Concatenation (Mangrulkar et al., 2022); 3) TIES (Yadav et al., 2024); 4) DARE (Yu et al., 2024); 5) Slerp (White, 2016); 6) LoraHub (Huang et al., 2024); 7) LM-Cocktail (Xiao et al., 2024); 8) DAM (Gauthier-Caron et al., 2024) and ZipLoRA (Shah et al., 2024) adapted to our use cases. These methods are typically designed for merging entire models rather than LoRA adapters. We adapt them to merge LoRAs and evaluate their performance on compositional tasks. Note that joint-expert LoRA, LoraHub, LM-Cocktail and DAM/ZipLoRA approaches leverage extra compositional task data.

4.2 Our Approach: Learnable Calibration

Our method aims to achieve strong compositional task performance while being efficient in computation and storage. After evaluating diverse existing merging strategies on compositional tasks, we have observed it is challenging to obtain adequate performance. Consequently, we introduce a learnable approach that utilizes data from compositional tasks during server-side pre-training of the additional parameters.

More specifically, in order to pre-train the additional parameters to enable solving compositional tasks, we assume access to the following: 1) a base LLM, 2) task specific LoRAs $\{B_i, A_i\}_{i=1}^N$, and 3) compositional task data $\mathbb{D}^C$ obtained from the compositional task $T_{[N]}^C$ using (1). The data are in the form of input-output pairs (x, y), where x is an input such as a long text, and y is the ground-truth output, such as a text summary in another language. Hence, our approach is not data-free, unlike the various merging strategies such as TIES and DARE, and the multi-step LoRA usage. However, the pre-training of the additional parameters is done on the server, where we can assume access to ample training data, *i.e.*, we assume data availability is not an issue.

The key idea of our approach is to use merged single-task LoRAs as the initial starting point, which is then calibrated further via a relatively small number of additional parameters P, specific to the compositional task. As LoRA parameters vary across tasks (Figure 3), the best performance is likely to be obtained with additional parameters P that are not shared. Our approach significantly reduces storage requirements compared to a new joint-expert adapter while enabling high performance. The adjusted forward pass is

$$h = W_0 x + \Delta W^c x = W_0 x + f(P, \{B_i, A_i\}_{i=1}^N) x, \tag{4}$$

where f represents the application of P on top of the single-task LoRAs. The parameters P are trained using compositional task data ($\mathbb{D}^C$) and a cross-entropy loss commonly used for LLM training. There are various options for how to utilize the parameters P as described next.

4.3 Variations of Learnable Calibration

We propose two variations of Learnable Calibration, providing a trade-off between model size and performance. Both variations begin with linearly merged single-task LoRAs:

$$B' = \frac{1}{N}\sum_{i=1}^{N} B_i, \quad A' = \frac{1}{N}\sum_{i=1}^{N} A_i, \tag{5}$$

which are then calibrated via f using parameters P. These parameters are shared across layers of models for efficiency, but are specific to each component, such as query or key projection, as these generally differ in size. We provide an overview of the two variations in Figure 4.

Variation #1 (Learnable Calibration). The smaller variation learns a vector of column-wise biases $\boldsymbol{p} \in \mathbb{R}^d$ (a special case of P), applied to the LoRA update matrix $\Delta W' = B'A'$ as

$$\Delta W^c = \boldsymbol{p} \oplus B'A' = \boldsymbol{p} \oplus \Delta W' = \sum_{i=1}^{d} p_i \Delta W'_i, \tag{6}$$

where operation $\oplus$ represents element-wise column addition. Learnable biases in $\boldsymbol{p}$ are initialized to 0.

Variation #2 (Learnable Calibration++). The larger variation introduces a calibration matrix P,

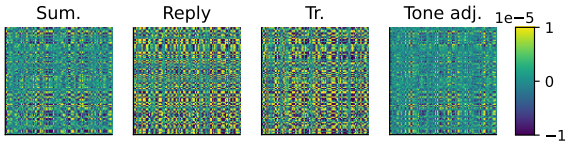


Figure 3: **LoRA weight update matrices differ across tasks,** so sharing additional parameters P is likely to be suboptimal.

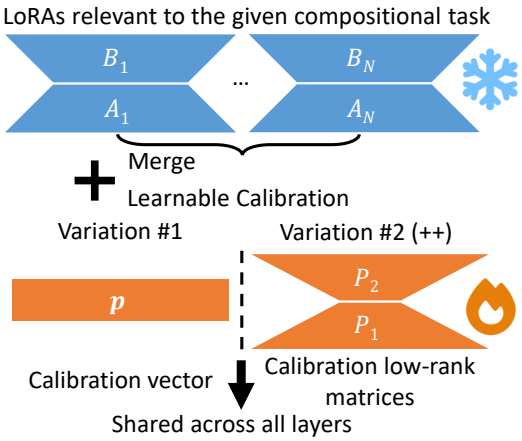


Figure 4: **Our Learnable Calibration.** We add a small number of calibration parameters to correct the initial merged LoRAs. Variation #1 uses a calibration vector of biases, while Variation #2 (++) uses two calibration low-rank matrices.

factorized into two low-rank matrices $P_2 \in \mathbb{R}^{d \times s}$ and $P_1 \in \mathbb{R}^{s \times k}$ (where $s \ll \min(d, k)$), resulting in the following update matrix

$$\Delta W^c = P_2 P_1 + \Delta W'. \quad (7)$$

This effectively adds a new calibration LoRA on top of the merged single-task adapters for improved performance. The calibration parameters are initialized in the same manner as standard LoRAs and are shared across layers.

The on-device pipeline of our system involves two steps: (i) Computing the merged, calibrated LoRA ΔW^c via matrix operations (6) or (7), which is extremely fast and negligible in runtime; (ii) Loading ΔW^c onto the model, which results in *inference latency and throughput identical* to standard LoRAs. Our method is compatible with existing frameworks such as Android AI Core (Burke, 2023) and Apple Intelligence (Gunter et al., 2024), making it readily deployable in real-world systems.

We include an ablation study to show it is indeed beneficial to use the calibration parameters on top of the combination of single-task LoRAs – rather than simply omitting them.

Sharing Calibration Parameters. We also consider a scenario where calibration parameters P are shared across multiple compositional tasks, *i.e.*, the same parameters are used in all considered combinations. In this scenario, we modify the dataset $\mathbb{D}^C$ to include examples from all four tasks in our benchmark. This approach further reduces storage requirements.

5 Experiments

5.1 Implementation Details

We use models that are suitable for on-device settings. More specifically, we evaluate performances on our benchmark using conversational versions of LLaMA 3.2 1B, Qwen2.5 1.5B, and StableLM2 1.6B models. All models are multilingual and support the tested languages.

LoRAs are applied to both the attention components (query, key, value, and output projections) and the MLP components (up, down, gate projections) (Fomenko et al., 2024; Tunstall et al., 2024). Training was conducted for one epoch on the full training set using the Adam optimizer with a learning rate of 5×10^{-5}. For LoRAs, we used a rank of 32, a scaling factor $\alpha = 16$, and a dropout rate of 0.05. Single-task LoRAs were trained on the datasets described in Section 3.

The parameters of Learnable Calibration were trained using the Adam optimizer on a randomly selected subset of 10,000 examples from the respective compositional multi-tasking dataset, with a learning rate of 5×10^{-4}. For the Learnable Calibration++ variation that incorporates factorization into two low-rank matrices, we used a rank of 4.

5.2 Main Results

The results of our main experiments are summarized in Table 2. From the results, we extract the following observations: 1) Despite including task-specific prompts, zero-shot performance was generally poor (*e.g.*, 0.44% LLM-J score on Sum. + Tr.). Similarly, in-context learning had only limited success. This suggests that on-device LLMs struggle to handle compositional tasks without fine-tuning. 2) Using the main or auxiliary task LoRA generally improved performance compared to the zero-shot baseline (*e.g.*, to 3.49% in LLM-J), but remained limited in effectiveness. 3) Both simple (*e.g.*, TIES) and advanced (*e.g.*, LoraHub) merging strategies performed on par with using a single LoRA, indicating their unsuitability for compositional multi-tasking. 4) Inefficient baselines, such as the multi-step LoRA application and joint-expert LoRA, performed significantly better than the other baselines (*e.g.*, 49.85% and 72.92% in LLM-J), highlighting their ability to address compositional tasks, albeit inefficiently. 5) Our family of approaches achieved strong results (*e.g.*, 65.15% in LLM-J), comparable to or surpassing inefficient baselines, while being highly efficient in storage and inference passes. 6) Overall, the more expressive Learnable Calibration++ consistently achieved the best results across all benchmarks, demonstrating its robustness and adaptability. 7) Our approaches achieve strong performance across all the different compositional tasks studied in our benchmark.

While ROUGE scores are less interpretable in isolation, the analysis of LLM judge scores provided useful insights. With Learnable Calibration, tasks were successfully solved in most cases, demonstrating promising progress. Future research is needed to improve overall performance and consistency across tasks.

5.3 Analyses

We have performed various further analyses, including ones about sharing of parameters across tasks, efficiency, benefit of using existing adapters and analysis of changes in the update matrices.

Table 2: **Benchmark of compositional multi-tasking.** Test results reported as % (↑) and averaged across models and languages or tones. Our Learnable Calibration methods achieve *comparable performance* to inefficient baselines while being significantly *more efficient* in terms of inferences and storage. Similarly, fast baselines, such as various merging strategies, typically fail in compositional multi-tasking. The metrics include ROUGE-L (R-L), Weighted ROUGE (W-R) and LLM judge (LLM-J) scores. The *Efficient?* column captures both runtime and storage efficiency. The bottom block includes results for a version of Learnable Calibration where additional parameters are shared across all four tasks. The best three methods from the main set of results are in bold.

		Sum. + translation		**Sum. + tone adj.**		**Reply + translation**		**Reply + tone adj.**	
	Efficient?	R-L	LLM-J	R-L	LLM-J	W-R	LLM-J	W-R	LLM-J
Zero-shot	✓	7.49	0.44	13.93	6.52	2.03	4.11	3.53	33.66
Main-task LoRA	✓	13.39	3.49	16.43	4.18	1.25	7.17	9.12	36.25
Auxiliary-task LoRA	✓	13.99	0.30	14.33	5.81	4.03	4.73	4.59	36.68
In-context learning	✓	14.46	10.95	16.96	24.12	3.93	8.72	4.66	46.23
Linear merge	✓	14.27	0.33	15.26	2.74	3.94	12.81	7.72	41.93
Concat merge	✓	14.39	0.34	15.27	2.76	3.97	13.10	7.65	41.77
TIES merge	✓	12.25	0.81	14.95	6.06	3.53	8.30	6.47	**47.87**
DARE merge	✓	9.27	0.68	14.51	6.34	2.95	8.57	4.56	41.76
Slerp merge	✓	13.96	0.54	14.87	4.87	3.73	9.04	6.57	46.97
LoraHub merge	✓	13.78	1.59	16.13	3.03	3.26	12.69	8.69	39.07
LM-Cocktail merge	✓	13.62	0.70	14.88	5.70	3.24	7.93	6.67	**48.76**
DAM/ZipLoRA merge	✓	16.19	17.16	15.68	2.78	4.62	31.88	8.00	43.19
Multi-step LoRA usage	✗	21.25	**72.92**	**20.23**	**34.32**	**10.04**	**69.83**	8.09	45.78
Joint-expert LoRA	✗	**21.36**	49.85	19.08	16.14	**14.99**	**65.73**	**14.33**	**47.06**
Learnable Calibration	✓	**25.40**	**59.23**	**24.58**	**28.89**	8.85	57.46	**10.86**	44.99
Learnable Calibration++	✓	**28.64**	**65.15**	**26.96**	**34.34**	**12.14**	**63.81**	**13.54**	45.40
Shared Learnable Calibration	✓	17.04	29.91	20.98	16.86	6.63	49.23	9.23	39.61
Shared Learnable Calibration++	✓	19.23	32.61	22.68	15.54	9.99	56.74	11.06	42.33

Sharing Parameters Across Tasks. To improve efficiency further, we investigate if Learnable Calibration parameters can be shared across tasks. Results from the bottom block of Table 2 show that sharing parameters leads to a slight performance decrease compared to using task-specific parameters. Nevertheless, shared parameters still outperform most baseline approaches, demonstrating their potential as a more storage-efficient solution.

Efficiency Analysis. We compare the efficiency of our solutions against the two well-performing but inefficient baselines in Table 3. Our solutions require only a minimal number of additional parameters, amounting to approximately 0.08–0.56% of the parameters of a joint-expert LoRA. The resulting additional storage on disk is less than 0.5 MB, making our solutions suitable for on-device deployment.

Benefit of Using Existing Adapters. To assess the impact of using existing adapters, we evaluate performance when training only the Learnable Calibration parameters without leveraging pre-existing adapters. Results in Table 4 confirm that starting with existing adapters and calibrating them using learnable parameters (biases or low-rank matrices) improves performance. Without calibration parameters, the approach becomes linear merge that has weak performance, highlighting the critical role of calibration for achieving strong results. The best performance is obtained when the merged adapters are used as a starting point and subsequently refined with learnable calibration parameters.

Table 3: **Efficiency of well-performing approaches.** Our methods require only 0.08–0.56% of additional parameters/storage, depending on the variation (averaged across models). Baselines not reported here, such as *Main-task LoRA* and *Linear Merge*, are efficient (*i.e.*, single inference pass, zero additional parameters/storage) but have significantly lower performance.

Method	**# of Inferences**	**Additional Parameters**	**Additional Storage**
Multi-step LoRAs	2×	0	0.00MB
Joint-expert LoRA	1×	30M	57.10MB
Learnable Calibration	1×	23K	0.05MB
Learnable Calibration++	1×	166K	0.32MB

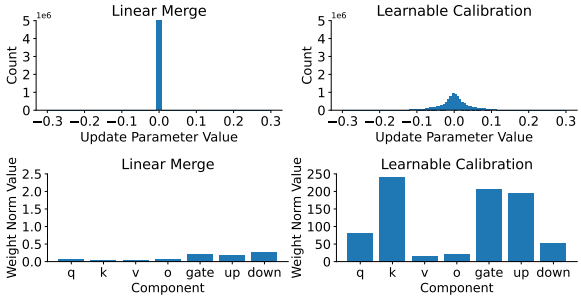
Analysis of Changes in the Update Matrices. To understand how our methods address compo-

Table 4: **Effect of pre-existing adapters.** Leveraging pre-existing adapters enhances performance in compositional multi-tasking for both variations of Learnable Calibration (LC). Results are averaged across different models and languages or tones. The top block shows results with separate learned parameters for each compositional task, while the bottom block shows results with shared parameters across tasks.

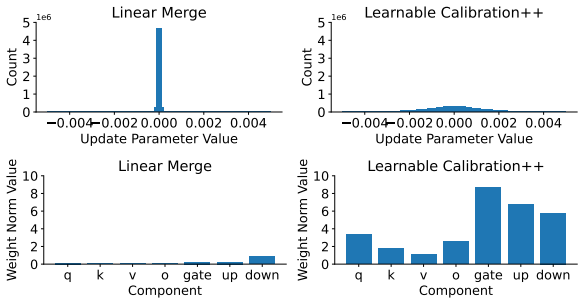
		Sum. + translation		**Sum. + tone adj.**		**Reply + translation**		**Reply + tone adj.**	
		R-L	LLM-J	R-L	LLM-J	W-R	LLM-J	W-R	LLM-J
Separate	LC	25.40	59.23	24.58	28.89	8.85	57.46	10.86	44.99
	LC w/o LoRAs	25.21	57.96	23.62	26.81	7.36	53.18	9.88	40.13
	LC++	28.64	65.15	26.96	34.34	12.14	63.81	13.54	45.40
	LC++ w/o LoRAs	28.45	63.20	26.96	34.73	11.76	61.82	13.13	44.62
Shared	LC	17.04	29.91	20.98	16.86	6.63	49.23	9.23	39.61
	LC w/o LoRAs	15.36	24.31	19.24	11.81	5.83	42.23	8.44	39.12
	LC++	19.23	32.61	22.68	15.54	9.99	56.74	11.06	42.33
	LC++ w/o LoRAs	17.70	35.66	21.53	16.28	9.51	54.79	10.90	42.17

sitional tasks, we analyze the changes in the update matrices compared to simple linear merging. We examine the distribution of values of the parameters via histograms and evaluate weight norms for two scenarios: sum. + tr. and reply + tone adj. (Figure 5). We observe that the addition of calibration parameters significantly increases the diversity of the LoRA update matrix, enabling the handling of auxiliary tasks. Also, the weight norms of the update matrices grow substantially, reflecting the additional complexity required to address compositional tasks. These patterns are consistent across both scenarios, illustrating how Learnable Calibration calibrates the adapters to accommodate multiple tasks effectively.

Further Analyses. Notably, we show Learnable Calibration 1) works well for models of various sizes such as from 0.5B to 3B (Appendix D), 2) is able to handle domain shift (Appendix E), and 3) obtains strong performance also for three-way compositional tasks (Appendix F). We also perform qualitative analysis in Appendix G. The analysis shows there are essentially two behavioral groups. The first group typically does not perform one of the tasks correctly and includes approaches such as zero-shot and existing merging strategies. The second group succeeds in performing both tasks and includes the two inefficient baselines as well as our Learnable Calibration solutions.



(a) Sum. + translation with Qwen2.5 1.5B model.



(b) Reply + tone adj. with Llama 3.2 1B.

Figure 5: **Changes in the update matrix.** Learnable Calibration makes the overall LoRA update matrix significantly more diverse to handle additional tasks. The weight norms also increase substantially across different components, reflecting the added complexity required for compositional multi-tasking.

6 Conclusion

We introduced the practically valuable problem of compositional multi-tasking for LLMs in on-device settings, where computational and storage resources are constrained. To facilitate research in this area, we developed a comprehensive benchmark comprising diverse practical compositional tasks. Our evaluation demonstrates that existing methods either lack efficiency or fail to achieve adequate performance, highlighting the need for new approaches. As a solution, we proposed Learnable Calibration, a family of methods that leverage pre-existing adapters on a device and calibrate them with a minimal number of additional parameters, achieving strong performance while being efficient in both storage and computation.

Limitations

Our evaluation focused on on-device-sized LLMs as that is the envisioned use case. One could also perform evaluation with significantly larger models, but this would require significant compute resources. We studied combinations of two and three tasks, as these are the most practically relevant, but it could be possible to evaluate with a larger number of tasks too if our benchmark was extended.

Ethical Considerations

Our research lies in the domain of text-generative AI, a field with significant ethical and societal implications. The proposed use cases for compositional multi-tasking, such as summarization and translation, have the potential to benefit users across diverse applications. However, this versatility also introduces risks, as the technology could be misused to achieve undesirable goals.

While LLMs typically incorporate built-in safeguarding mechanisms, these safeguards may be weakened when models are merged (Hammoud et al., 2024) or adapted for compositional multi-tasking. It is crucial to ensure that safeguards remain effective in these scenarios before deploying solutions.

Potential future research could explore the interplay between compositional multi-tasking and safeguarding mechanisms, with a focus on developing robust, high-performing safeguards for the proposed setup. Addressing these challenges will be essential for the responsible advancement and deployment of compositional multi-tasking capabilities.

References

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, and 1 others. 2024. Stable LM 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Alexander Borzunov, Max Ryabinin, Artem Chumachenko, Dmitry Baranchuk, Tim Dettmers, Younes Belkada, Pavel Samygin, and Colin A Raffel. 2024. Distributed inference and fine-tuning of large language models over the internet. In *NeurIPS*.

Dave Burke. 2023. A new foundation for AI on Android. https://android-developers.googleblog.com/2023/12/a-new-foundation-for-ai-on-android.html.

Samuel Carreira, Tomás Marques, José Ribeiro, and Carlos Grilo. 2023. Revolutionizing mobile interaction: Enabling a 3 billion parameter GPT LLM on mobile. *arXiv preprint arXiv:2310.01434*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *ACL Findings*.

Sauptik Dhar, Junyao Guo, Jiayi (Jason) Liu, Samarth Tripathi, Unmesh Kurup, and Mohak Shah. 2021. A survey of on-device machine learning: An algorithms and learning theory perspective. *ACM Trans. Internet Things*, 2(3).

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *EMNLP*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Arash Einolghozati, Anchit Gupta, Keith Diedrick, and Sonal Gupta. 2020. Sound natural: Content rephrasing in dialog systems. In *EMNLP*.

Vlad Fomenko, Han Yu, Jongho Lee, Stanley Hsieh, and Weizhu Chen. 2024. A note on lora. *arXiv preprint arXiv:2404.05086*.

Thomas Gauthier-Caron, Shamane Siriwardhana, Elliot Stein, Malikeh Ehghaghi, Charles Goddard, Mark McQuade, Jacob Solawetz, and Maxime Labonne. 2024. Merging in a bottle: Differentiable adaptive merging (DAM) and the path from averaging to automation. *arXiv preprint arXiv:2410.08371*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *ACL*.

Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, and 1 others. 2024. Apple intelligence foundation language models. *arXiv preprint arXiv:2407.21075*.

Hasan Abed Al Kader Hammoud, Umberto Michieli, Fabio Pizzati, Philip Torr, Adel Bibi, Bernard Ghanem, and Mete Ozay. 2024. Model merging and safety alignment: One bad model spoils the bunch. In *EMNLP Findings*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. In *TMLR*.

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 *ICLR*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024. Lorahub: Efficient cross-task generalization via dynamic lora composition. In *COLM*.

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

Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2024. Faithful persona-based conversational dataset generation with large language models. In *NLP4ConvAI*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, Barcelona, Spain. Association for Computational Linguistics.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. In *ICML*.

Yixin Liu, Kejian Shi, Katherine S He, Longtian Ye, Alexander R Fabbri, Pengfei Liu, Dragomir Radev, and Arman Cohan. 2024b. On learning to summarize with large language models as references. In *NAACL*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2025. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7).

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *NAACL*.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. In *ACL Findings*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *ACL*.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. 2024. Ziplora: Any subject in any style by effectively merging loras. In *ECCV*.

Donald Shenaj, Ondrej Bohdal, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. 2025. Lora.rar: Learning to merge loras via hypernetworks for subject-style conditioned image generation. *ICCV*.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen, and Lei Meng. 2024. Rewritelm: An instruction-tuned large language model for text rewriting. In *AAAI*.

Abigail Sticha, Norbert Braunschweiler, Rama Sanand Doddipatla, and Kate M Knill. 2024. Advancing faithfulness of large language models in goal-oriented dialogue question answering. In *ACM Conference on Conversational User Interfaces*.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. 2023. Parameter efficient multi-task model fusion with partial linearization. In *ICLR*.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Gronroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raul Vazquez, and Sami Virpioja. 2023. Democratizing neural machine translation with OPUS-MT. *Language Resources and Evaluation*, (58).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *EAMT*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alvaro Bartolome, Alexander M. Rush, and Thomas Wolf. 2024. The Alignment Handbook.

Kumar Utsav. 2023. Redpajama-incite-base-3b-v1 model finetuned for paraphrasing and changing the tone. https://huggingface.co/llm-toys.

Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, and 1 others. 2024. Redpajama: an open dataset for training large language models. In *NeurIPS Datasets and Benchmarks*.

Tom White. 2016. Sampling generative networks. *arXiv preprint arXiv:1609.04468*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Xingrun Xing. 2024. LM-Cocktail: Resilient tuning of language models via model merging. In *ACL Findings*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. In *NeurIPS*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024b. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

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 *ICML*.

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

Jinghan Zhang, Junteng Liu, Junxian He, and 1 others. 2023a. Composing parameter-efficient modules with arithmetic operation. In *NeurIPS*.

Mozhi Zhang, Wei Wang, Budhaditya Deb, Guoqing Zheng, Milad Shokouhi, and Ahmed Hassan Awadallah. 2021. A dataset and baselines for multilingual reply suggestion. In *ACL*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023b. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. In *NAACL Findings*.

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. 2023. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *arXiv preprint arXiv:2309.02411*.

A LLM Judge Evaluation

We conducted an evaluation using the LLaMA 3.1 70B instruction-tuned model (Dubey et al., 2024) as an LLM Judge. The evaluation prompts used for the main scenarios are provided in Figures A1, A2. The LLM Judge prompt used for evaluating compositional multi-tasking with three tasks is shown in Figure A3.

Given the large number of experiments conducted, LLM Judge evaluations are relatively expensive to run locally, particularly for researchers operating with limited computational resources. As such, we recommend focusing on standard metrics such as ROUGE-L and Weighted ROUGE when developing and testing new approaches using our benchmark.

Our LLM judge results provide an additional layer of interpretation, offering a practical (albeit somewhat noisy) perspective on the scores derived from standard metrics. These results help bridge the gap between quantitative evaluation and qualitative performance assessment.

B Additional Details

B.1 Task Prompts

For all compositional tasks, we include a prompt explicitly specifying the desired combination of tasks. The prompts used are as follows:

- **Summarization + translation**: *Summarize the following text and translate it from English to* `{language}`.
- **Summarization + tone adjustment**: *Summarize the following text and change its tone to* `{tone}`
- **Reply + translation**: *Suggest a reply for the following text and translate it from English to* `{language}`
- **Reply + tone adjustment**: *Suggest a reply for the following text and change its tone to* `{tone}`

In the case of in-context learning, we also provide one example for the task, showing an example input and output. One example was used because on-device LLMs have a very limited context window, and so in-context learning is generally not a suitable strategy in these settings.

B.2 Hyperparameter Selection

Hyperparameters were selected using validation data. For common merging strategies, such as linear merging and TIES, we tested a wide range of hyperparameters on a representative compositional multi-tasking scenario. However, we observed that performance was consistently similar across different configurations and did not surpass the performance of using either the main or auxiliary-task LoRA paired with prompting.

As a result, we chose hyperparameters that are broadly applicable across diverse tasks: weights of 0.5 (linear, concat, TIES, DARE merges) and density of 0.5 (TIES, DARE merges). These values are intended to represent a reasonable balance, ensuring fair comparisons across different methods.

C Extended Evaluation with Standard Metrics

We provide an extended evaluation using all standard metrics, including ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and Weighted ROUGE (W-R). The detailed results are presented in Table A1 and Table A2. These results are consistent with those reported in the main text, further validating the effectiveness and robustness of our proposed methods.

Prompt for Cross-Lingual Summarization

You are in a setting where you want to summarize a dialogue in another language. The dialogue is in English, but the summary should be in `{language}`. Your task is to judge the quality of the summary and if it is in `{language}`.
The summary should not take the form of a reply to the dialogue.
I give you an example ground-truth summary that is in `{language}` and would be considered a good summary of the dialogue

Dialogue: `{dialogue}`

Ground truth summary: `{ground truth summary}`

Proposed summary: `{proposed summary}`

Does the proposed summary summarize the dialogue and is it in `{language}`? Please respond with 0 or 1. Only output 0 or 1 and nothing else.

Prompt for Cross-Tone Summarization

You are in a setting where you want to summarize a dialogue in a `{tone}` tone. Your task is to judge the quality of the summary and if it uses a `{tone}` tone.
The summary should not take the form of a reply to the dialogue.
I give you an example ground-truth summary that follows a `{tone}` tone and would be considered a good summary of the dialogue.

Dialogue: `{dialogue}`

Ground-truth summary: `{ground-truth summary}`

Proposed summary: `{proposed summary}`

Does the proposed summary summarize the dialogue and is it written in a `{tone}` tone? Please respond with 0 or 1. Only output 0 or 1 and nothing else.

Figure A1: **LLM Judge prompts for summarization-based compositional tasks.** Prompts used by LLaMA 3.1 70B LLM Judge to evaluate compositional tasks with a given target language or tone.

Prompt for Cross-Lingual Reply Suggestion

You are in a setting where you receive a message in English and you reply in {language}. Your task is to judge the usefulness of the proposed reply.
I give you an example ground-truth response in {language} that would be considered a good reply for the message in English.
The proposed reply should not directly translate or paraphrase the message you have received, it should respond to it.

Message in English: {message}

Ground-truth reply in {language}: {ground-truth reply}

Proposed reply: {proposed reply}

Is the proposed reply useful and is it in {language}? Please respond with 0 or 1. Only output 0 or 1 and nothing else.

Prompt for Cross-Tone Reply Suggestion

You are in a setting where you receive a message and you want to reply in a {tone} tone. Your task is to judge the usefulness of the proposed reply.
I give you an example ground-truth response that follows a {tone} tone and would be considered a good reply for the message.

Message: {message}

Ground-truth reply: {ground-truth reply}

Proposed reply: {proposed reply}

Is the proposed reply relevant and is it written in a {tone} tone? Please respond with 0 or 1. Only output 0 or 1 and nothing else.

Figure A2: **LLM Judge prompts for reply-based compositional tasks.** Prompts used by LLaMA 3.1 70B LLM Judge to evaluate compositional tasks with a given target language or tone.

Prompt for Cross-Lingual Tone-Adjusted Summarization

You are in a setting where you want to summarize a dialogue in a {tone} tone in another language. The dialogue is in English, but the summary should be in a {tone} tone in {language}. Your task is to judge the quality of the summary, if it uses a {tone} tone and if it is in {language}.
The summary should not take the form of a reply to the dialogue.
I give you an example ground-truth summary that follows a {tone} tone and is in {language}. The example would be considered a good summary of the dialogue.

Dialogue: {dialogue}

Ground-truth summary: {ground-truth summary}

Proposed summary: {proposed summary}

Does the proposed summary summarize the dialogue and is it written in a {tone} tone in {language}? Please respond with 0 or 1. Only output 0 or 1 and nothing else.

Figure A3: **LLM Judge prompt for three-way compositional tasks.** Prompt used by LLaMA 3.1 70B LLM Judge to evaluate three compositional tasks with a given target language and tone.

Table A1: **Evaluation of compositional multi-tasking across different scenarios.** Test results reported as percentages (%, ↑) and averaged across different models and languages or tones. Our family of methods achieves comparable performance to inefficient baselines while being significantly more efficient in terms of inferences and storage. Similarly, fast baselines, such as various merging strategies, generally fail in compositional multi-tasking. The bottom block includes results for a version where the additional parameters of Learnable Calibration are shared across all four tasks.

	Sum. + translation			**Sum. + tone adj.**			**Reply + translation**	**Reply + tone adj.**
	R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**	**W-R**	**W-R**
Zero-shot	8.68	1.45	7.49	18.84	3.42	13.93	2.03	3.53
Main-task LoRA	16.33	2.73	13.39	21.53	4.43	16.43	1.25	9.12
Auxiliary-task LoRA	16.77	2.98	13.99	18.61	3.71	14.33	4.03	4.59
In-context learning	18.05	3.76	14.46	23.53	5.68	16.96	3.93	4.66
Linear merge	16.90	3.02	14.27	19.84	3.93	15.26	3.94	7.72
Concat merge	17.04	3.08	14.39	19.87	3.94	15.27	3.97	7.65
TIES merge	14.35	2.65	12.25	20.03	3.88	14.95	3.53	6.47
DARE merge	10.78	1.96	9.27	19.50	3.69	14.51	2.95	4.56
Slerp merge	16.40	2.97	13.96	19.68	3.81	14.87	3.73	6.57
LoraHub merge	16.51	2.80	13.78	20.94	4.32	16.13	3.26	8.69
LM-Cocktail merge	15.97	2.88	13.62	19.78	3.81	14.88	3.24	6.67
DAM/ZipLoRA merge	20.38	4.38	16.19	20.36	4.14	15.68	4.62	8.00
Multi-step LoRA usage	27.20	7.92	21.25	27.19	6.85	20.23	10.04	8.09
Joint-expert LoRA	26.77	7.05	21.36	24.98	6.38	19.08	14.99	14.33
Learnable Calibration	31.83	9.53	25.40	33.14	9.48	24.58	8.85	10.86
Learnable Calibration++	35.57	12.11	28.64	36.09	10.95	26.96	12.14	13.54
Shared Learnable Calibration	21.15	5.30	17.04	28.00	7.72	20.98	6.63	9.23
Shared Learnable Calibration++	23.91	6.25	19.23	29.89	8.38	22.68	9.99	11.06

Table A2: **Ablation on usefulness of single-task LoRAs.** Leveraging existing adapters enhances the performance of Learnable Calibration (LC) in compositional multi-tasking. Test scores (%, ↑) are reported, averaged across various models and languages or tones. The top block utilizes separate parameters for each compositional task, while the bottom block employs a single set of shared parameters.

		Sum. + translation			**Sum. + tone adj.**			**Reply + translation**	**Reply + tone adj.**
		R-1	**R-2**	**R-L**	**R-1**	**R-2**	**R-L**	**W-R**	**W-R**
Separate	LC	31.83	9.53	25.40	33.14	9.48	24.58	8.85	10.86
	LC w/o LoRAs	31.53	9.36	25.21	32.02	8.95	23.62	7.36	9.88
	LC++	35.57	12.11	28.64	36.09	10.95	26.96	12.14	13.54
	LC++ w/o LoRAs	35.23	11.94	28.45	36.08	10.95	26.96	11.76	13.13
Shared	LC	21.15	5.30	17.04	28.00	7.72	20.98	6.63	9.23
	LC w/o LoRAs	19.05	4.73	15.36	25.53	6.76	19.24	5.83	8.44
	LC++	23.91	6.25	19.23	29.89	8.38	22.68	9.99	11.06
	LC++ w/o LoRAs	21.91	5.73	17.70	28.31	7.83	21.53	9.51	10.90

D Scaling Analysis

LLMs designed for mobile device deployment are available in various sizes, including newer models with up to 3B parameters (Gunter et al., 2024; Carreira et al., 2023). We analyze the scalability of our solution across different model sizes in Figure A4. For this analysis, we use the Qwen2.5 model, which is available in three sizes suitable for on-device deployment: 0.5B, 1.5B, and 3B parameters. The results demonstrate that our solutions consistently perform well across all model sizes. Additionally, as expected, larger models typically achieve stronger performance.

E Out-of-Domain Generalization

We evaluate the generalization of LoRAs and additional parameters trained on the DialogSum dataset by testing them on the SAMSum dataset (Gliwa et al., 2019). Our aim is to assess whether our method maintains strong performance and improvements when applied to data from a different domain. Specifically, DialogSum features spoken-language conversations from daily life scenarios, while SAMSum includes written-style dialogues from online interactions. For instance, DialogSum contains sentences like "*#Person_1#: Good morning. I wonder whether you have got an answer from your superior.*," whereas SAMSum includes examples such as "*Leo: BTW what are those pics?*"

Table A3 presents results for both in-domain and out-of-domain settings using the cross-lingual English-to-Spanish summarization task. The evaluation spans all three models considered earlier. The results demonstrate that the trained LoRAs and additional parameters deliver strong performance even under mild domain shifts, such as variations in conversation style.

Table A3: **Comparison of in-domain and out-of-domain settings.** Test scores reported as % (↑) and averaged across models for the cross-lingual English-to-Spanish summarization task. The approaches work well even in the presence of mild domain shift.

	In-domain		Out-of-domain	
	R-L	LLM-J	R-L	LLM-J
Zero-shot	9.59	0.20	10.58	3.54
Main-task LoRA	14.25	3.09	15.97	9.89
Auxiliary-task LoRA	14.42	0.16	19.67	1.14
Linear merge	15.60	0.18	18.30	1.26
Multi-step LoRA usage	21.71	75.51	26.81	76.35
Joint-expert LoRA	23.47	52.51	25.38	56.33
Learnable Calibration	27.68	61.58	27.78	57.26
Learnable Calibration++	30.98	67.98	28.97	59.91

F Compositional Multi-tasking with Three Tasks

We demonstrate that combinations involving three tasks can also be effectively addressed using our Learnable Calibration framework. Specifically, we evaluate the joint tasks of summarization, professional tone adjustment, and translation from English to Spanish, German, or French. We created data for this task by converting ground-truth summaries into a professional tone and then translating them into multiple languages. The same models for tone adjustment and translation were applied in this setup. Given the large number of possible combinations, we focus on one tone and main task for simplicity. Additionally, we evaluate only a subset of the most relevant approaches for this scenario.

Table A4: **Compositional multi-tasking with three-way tasks.** Combination of summarization, professional tone adjustment, and translation to various languages. Test scores reported as % (↑) and averaged across languages and models. Our solution also works well for three-way compositional tasks.

	R-L	LLM-J
Zero-shot	7.35	0.03
Summarization LoRA	9.41	0.00
Translation LoRA	10.95	0.00
Tone adj. LoRA	8.39	0.01
Linear merge	9.89	0.01
Multi-step LoRA usage	16.50	43.82
Joint-expert LoRA	14.35	12.75
Learnable Calibration	19.19	28.66
Learnable Calibration++	21.42	39.16

We report the results of our compositional multi-tasking analysis with three-way tasks in Table A4. The findings show: 1) inefficient baselines outperform simple baselines in this more challenging scenario, 2) our Learnable Calibration approach outperforms the inefficient baselines both in terms of accuracy and footprints, and 3) Learnable Calibration++ performs better than its simpler variant. While the overall performance is lower than that for the simpler combination of summarization and translation, these results confirm our framework can successfully handle compositions involving more than two tasks, further proving its flexibility.

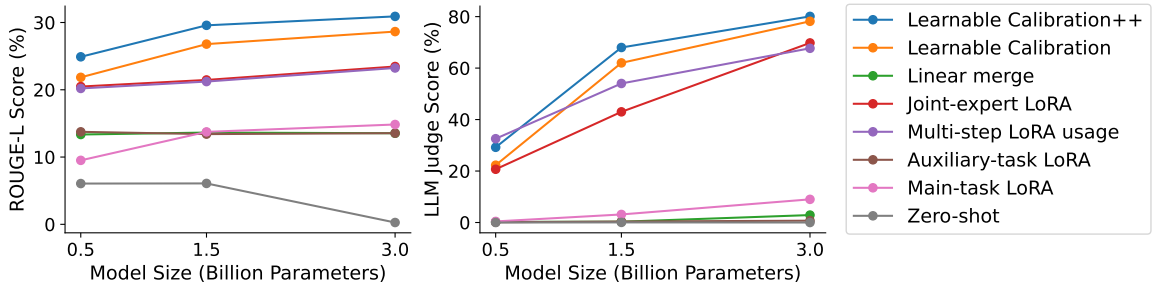


Figure A4: **Evaluation across model sizes.** Evaluation of compositional multi-tasking on the cross-lingual summarization task for the Qwen2.5 model across different sizes: 0.5B, 1.5B, and 3B. Test results for ROUGE-L and LLM Judge score (%, ↑) are averaged across languages. Our solutions consistently demonstrate strong performance across all model sizes, with performance improving as model size increases.

G Qualitative Analysis

ROUGE scores measure the overlap between generated and reference texts but can be challenging to interpret. To better understand these differences, we conducted a qualitative analysis.

After inspecting model outputs and ground-truth references, we have observed there are essentially two behavioral groups. The first group typically does not perform one of the tasks, *e.g.*, sometimes generates a summary but fails to translate it correctly or provides a translated text but does not summarize. This group includes approaches such as zero-shot and existing merging strategies. The second group succeeds in performing both tasks. For instance, they correctly execute both translation and summarization. Representatives of this category are the two inefficient baselines as well as our Learnable Calibration solutions. The qualitative differences in ROUGE scores reflect these behaviors. Within the group itself, higher scores indicate better execution quality and consistency across more examples. The frequency with which approaches successfully complete both tasks aligns with our LLM judge analysis, providing a quantitative measure of task success.

More specifically, we perform the qualitative analysis using test examples with ROUGE scores close to the average for each method. We compare the performance of three approaches—zero-shot, main-task LoRA, and Learnable Calibration++—to illustrate varying levels of success. For this analysis, we use the Qwen2.5 1.5B model with either Spanish as the target language or professional tone as the target tone.

Figures A5, A6 and A7 highlight common issues encountered by zero-shot and main-task LoRA methods:

1. Failing to summarize the input text when summarization is required.

2. Not adhering to the target language or tone specified in the task.

In contrast, Learnable Calibration++ demonstrates significantly better performance, successfully executing both tasks in the compositional task.

We conducted further qualitative analysis by examining outputs from different methods for the same input. This analysis focused on the cross-lingual summarization task, with results shown in Figures A8, A9, A10, and A11.

The results reveal that multi-step and joint-expert approaches, as well as our Learnable Calibration solutions, consistently succeed in performing the compositional task. Conversely, other approaches often fail to execute one or both tasks, reinforcing the advantage of our methods in handling compositional multi-tasking scenarios.

H Learning Curves Analysis of Our Approach

We analyze the training process of Learnable Calibration in both variations to better understand its behavior. For this analysis, we use the Qwen2.5 1.5B model with Spanish as the target language and a professional target tone. The results in Figure A12 shows the loss consistently decreases as training progresses and more samples are processed.

Cross-Lingual (English to Spanish) Summarization

Zero-shot:
Reference: #Person1# está rogando a Steven que persuada a su esposa para que no se divorcie de él. Steven está de acuerdo.
Prediction: #Persona1#: Steven, necesito tu ayuda muy necesitado.
Translated reference: #Person1# is begging Steven to persuade his wife not to divorce him. Steven agrees.
Translated prediction: #Person1#: Steven, I need your much needed help.
ROUGE-L score: 7.14

Main-task LoRA:
Reference: #Person1# sospecha que #Person2# tiene varicela y quiere mantenerse alejado de #Person2#. #Person2# piensa que #Person1# está exagerando.
Prediction: #Person1#: ¿Qué te pasa? ¿Por qué estás picando tanto?
Translated reference: #Person1# suspects that #Person2# has chickenpox and wants to stay away from #Person2#. #Person2# thinks that #Person1# is exaggerating.
Translated prediction: #Person1#: What's wrong with you? Why are you stinging so much?
ROUGE-L score: 14.29

Learnable Calibration++:
Reference: Frank describe el calendario de su nuevo trabajo y el seguro en la Oficina de Correos a Judy.
Prediction: Frank le dice a Judy que obtuvo un nuevo trabajo y le dice a Judy que el gobierno ofrece beneficios de salud de calidad.
Translated reference: Frank describes the schedule of his new job and insurance at Judy's Post Office.
Translated prediction: Frank tells Judy that he got a new job and tells Judy that the government offers quality health benefits.
ROUGE-L score: 28.57

Figure A5: **Analysis of representative examples for cross-lingual summarization.** Zero-shot and main-task LoRA approaches often fail to perform summarization, instead continuing the conversation in Spanish. In contrast, our approach successfully executes both summarization and translation or tone adjustment tasks.

Cross-Tone (Professional) Summarization

Zero-shot:
Reference: #Person1# is not satisfied with the steak and is considering alternatives.
Prediction: #Person1#: Excuse me.
ROUGE-L score: 14.29

Main-task LoRA:
Reference: In order to prevent employees from engaging in unnecessary activities, Mr. #1# decides to discontinue the use of Instant Messaging programs and requests Ms. Dawson to issue a memo to all employees by the morning.
Prediction: #Person1#: Ms. Dawson, I need you to take a dictation for me.
ROUGE-L score: 17.02

Learnable Calibration++:
Reference: As a result of #Person1's desire to visit the science museum, they became lost. To resolve the issue, #Person2# assisted them in purchasing a ticket and providing them with directions.
Prediction: #Person2# assists #Person1# in navigating the science museum and provides guidance on the ticket machines.
ROUGE-L score: 26.09

Figure A6: **Analysis of representative examples for cross-tone summarization.** Zero-shot and main-task LoRA approaches often fail to perform summarization, instead continuing the conversation in Spanish. In contrast, our approach successfully executes both summarization and translation or tone adjustment tasks.

Cross-Lingual (English to Spanish) Reply Suggestion

Zero-shot:
Reference: Enseño biología en una universidad.
Prediction: What do you do for a living?
Translated reference: I teach biology at a university.
Translated prediction: What do you do for a living?
Weighted ROUGE score: 2.38

Main-task LoRA:
Reference: Sólo he hecho modelaje de moda.
Prediction: Oh, that's cool! I've always wanted to be a model.
Translated reference: I've only done fashion modeling.
Translated prediction: Oh, that's cool! I've always wanted to be a model.
Weighted ROUGE score: 1.75

Learnable Calibration++:
Reference: Me gusta ir al cine, jugar videojuegos y pasar el rato con mis amigos.
Prediction: Me gusta pasar tiempo con mi familia y amigos.
Translated reference: I like to go to the movies, play video games and hang out with my friends.
Translated prediction: I like spending time with my family and friends.
Weighted ROUGE score: 11.87

Cross-Tone (Professional) Reply Suggestion

Zero-shot:
Reference: Yes, it is indeed!
Prediction: Absolutely, it sounds like a fantastic opportunity!
Weighted ROUGE score: 3.03

Main-task LoRA:
Reference: It's an exciting opportunity to unleash your creativity and unleash the power of apps and websites.
Prediction: It's a lot of fun! You get to create your own programs and apps.
Weighted ROUGE score: 9.51

Learnable Calibration++:
Reference: I enjoy attending the movies and engaging in leisure activities such as reading.
Prediction: I enjoy reading, watching movies, and spending time with my family.
Weighted ROUGE score: 13.01

Figure A7: **Analysis of representative examples for cross-lingual and cross-tone reply suggestion.** Zero-shot and main-task LoRA approaches often fail to use the correct language or apply the desired tone. In contrast, our approach successfully executes both reply suggestion and translation or tone adjustment tasks.

Cross-Lingual (English to Spanish) Summarization

Reference: Eric y Rob van a ver un stand-up en YouTube.
Translated reference: Eric and Rob are going to watch a stand-up on YouTube.

Zero-shot prediction: Eric: Máquina!
Zero-shot translated prediction: Eric: Machine!

Main-task LoRA prediction: Eric: Máquina!
Main-task LoRA translated prediction: Eric: Machine!

Auxiliary-task LoRA prediction: Eric: Máquina! Rob: ¡Esto es genial! Eric: Lo sé. Y muestra cómo los estadounidenses ven a Rusia. Rob: Y es muy divertido. Eric: Lo sé. Y me encanta la parte del tren. Rob: ¡Hahahaha! No se habla a la máquina así. Eric: ¿Y es su única actuación? Rob: No sé. Lo veré. Eric: Sí. Rob: ¡Sí!
Auxiliary-task LoRA translated prediction: Eric: Machine! Rob: This is great! Eric: I know. And it shows how Americans see Russia. Rob: And it's really fun. Eric: I know. And I love the part of the train. Rob: Hahaha! You don't talk to the machine like that. Eric: And it's your only performance? Rob: I don't know. I'll see it. Eric: Yes. Rob: Yes!

Linear merge prediction: Eric: Máquina!
Linear merge translated prediction: Eric: Machine!

Multi-step LoRA usage prediction: Eric y Rob hablan de un video divertido que muestra una máquina hablando como estadounidenses. Lo encuentran divertido y planean ver más de los videos de stand-up de Eric.
Multi-step LoRA usage translated prediction: Eric and Rob talk about a funny video that shows a machine talking like Americans. They find it fun and plan to see more of Eric's stand-up videos.

Joint-expert LoRA prediction: Eric y Rob están viendo un stand-up de Eric, un hombre americano que hace stand-up. Eric está muy divertido y se ríe mucho. Rob también está viendo y se ríe.
Joint-expert LoRA translated prediction: Eric and Rob are watching a stand-up by Eric, an American man who makes stand-up. Eric is very funny and laughs a lot. Rob is also watching and laughing.

Learnable Calibration prediction: Rob y Eric están interesados en ver los vídeos de stand-up de Eric.
Learnable Calibration translated prediction: Rob and Eric are interested in watching Eric's stand-up videos.

Learnable Calibration++ prediction: Eric y Rob están viendo los stand-up de Eric.
Learnable Calibration++ translated prediction: Eric and Rob are watching Eric's stand-up.

Figure A8: **Analysis of representative examples for cross-lingual summarization using a fixed input.**

Cross-Tone (Professional) Summarization

Reference: #Person2# was late due to traffic jam. #Person1# encouraged #Person2# to utilize public transportation to maintain a healthy lifestyle and safeguard the environment.

Zero-shot prediction: #Person1#: You're finally here! What took so long?

Main-task LoRA prediction: #Person2# is going to quit driving to work.

Auxiliary-task LoRA prediction: Summary: Person 2 is stuck in traffic again, and Person 1 suggests finding a different route or using public transport. Person 2 acknowledges the benefits of public transport but is concerned about missing the freedom of driving. Person 1 suggests biking to work when the weather is nicer, which Person 2 agrees with. Person 2 decides to quit driving to work and switch to public transport or biking.

Linear merge prediction: #Person1#: You're finally here! What took so long?

Multi-step LoRA usage prediction: The individual designated as #Person2# will be ceasing their practice of commuting via automobile for employment purposes.

Joint-expert LoRA prediction: #Person2# is considering switching to public transport to work.

Learnable Calibration prediction: #Person2# arrived late due to traffic congestion, and #Person1# suggested that #Person2# consider taking public transport to work.

Learnable Calibration++ prediction: #Person2# is stuck in traffic and is considering taking public transport to work. #Person1# suggests that #Person2# consider biking to work.

Figure A9: **Analysis of representative examples for cross-tone summarization using a fixed input.**

Cross-Lingual (English to Spanish) Reply Suggestion

Reference: ¡Lo es!
Translated reference: It is!

Zero-shot prediction: ¡Eso suena como un trabajo excelente!
Zero-shot translated prediction: That sounds like an excellent job!

Main-task LoRA prediction: Thanks! I'm really enjoying it.
Main-task LoRA translated prediction: Thank you! I'm really enjoying it.

Auxiliary-task LoRA prediction: Eso suena como un trabajo excelente!
Auxiliary-task LoRA translated prediction: That sounds like an excellent job!

Linear merge prediction: ¡Eso suena genial!
Linear merge translated prediction: That sounds great!

Multi-step LoRA usage prediction: Sí, me encanta.
Multi-step LoRA usage translated prediction: Yeah, I love it.

Joint-expert LoRA prediction: Me encanta trabajar con niños.
Joint-expert LoRA translated prediction: I love working with kids.

Learnable Calibration prediction: Sí, es muy divertido.
Learnable Calibration translated prediction: Yeah, it's a lot of fun.

Learnable Calibration++ prediction: Gracias, lo es.
Learnable Calibration++ translated prediction: Thank you, it is.

Figure A10: **Analysis of representative examples for cross-lingual reply suggestion using a fixed input.**

Cross-Tone (Professional) Reply Suggestion

Reference: I will certainly consider your recommendation. Thank you for the valuable input.

Zero-shot prediction: Absolutely, it is a stunning location.

Main-task LoRA prediction: I will! Thanks for the recommendation.

Auxiliary-task LoRA prediction: Absolutely, it's a wonderful location.

Linear merge prediction: I agree. It's a great place to visit.

Multi-step LoRA usage prediction: I will! Thank you for your recommendation.

Joint-expert LoRA prediction: I will certainly consider it. Thank you for the recommendation.

Learnable Calibration prediction: I will definitely consider it.

Learnable Calibration++ prediction: I will consider it.

Figure A11: **Analysis of representative examples for cross-tone reply suggestion using a fixed input.**

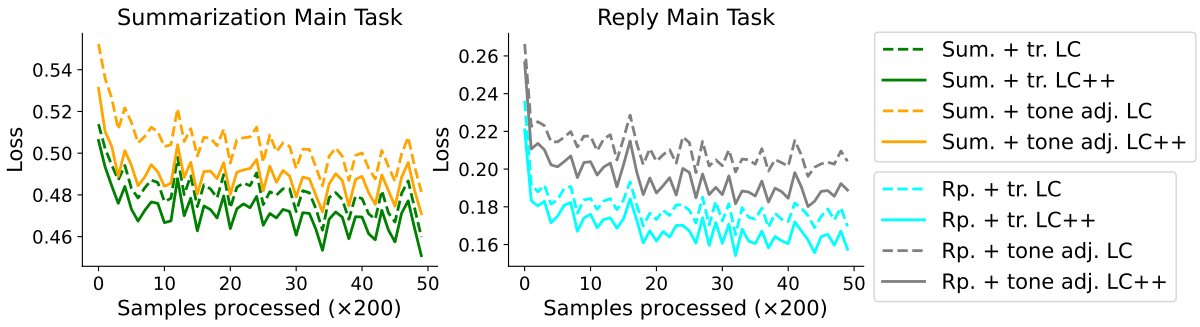


Figure A12: **Learning curves analysis of our approach.** The training loss for Learnable Calibration parameters decreases consistently as training progresses.