GenKnowSub: Improving Modularity and Reusability of LLMs through General Knowledge Subtraction

Mohammadtaha Bagherifard[2] **Sahar Rajabi**[1*] **Ali Edalat**[1*]
Yadollah Yaghoobzadeh[1,2]
[1]School of Electrical and Computer Engineering,
College of Engineering, University of Tehran, Tehran, Iran
[2]Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran
taha.bagheri98@gmail.com,
{sahar.rajabi, ali.edalat, y.yaghoobzadeh}@ut.ac.ir

Abstract

Large language models often struggle with zero-shot generalization, and several modular approaches have been proposed to address this challenge. Yet, we hypothesize that a key limitation remains: the entanglement of general knowledge and task-specific adaptations. To overcome this, we propose a modular framework that disentangles these components by constructing a library of task-specific LoRA modules alongside a general-domain LoRA. By subtracting this general knowledge component from each task-specific module, we obtain residual modules that focus more exclusively on task-relevant information—a method we call general knowledge subtraction (GenKnowSub). Leveraging the refined task-specific modules and the Arrow routing algorithm (Ostapenko et al., 2024), we dynamically select and combine modules for new inputs without additional training. Our studies on the Phi-3 model and standard Arrow as baselines reveal that using general knowledge LoRAs derived from diverse languages, including English, French, and German, yields consistent performance gains in both monolingual and cross-lingual settings across a wide set of benchmarks. Further experiments on Phi-2 demonstrate how GenKnowSub generalizes to weaker LLMs. The complete code and data are available at https://github.com/saharsamr/Modular-LLM.

1 Introduction

The rapid advancement of large language models (LLMs) has led to their widespread adoption in various NLP tasks, ranging from text generation to machine translation and question-answering (Brown et al., 2020; Raffel et al., 2020). Despite their remarkable performance, a key challenge remains: ensuring effective generalization to unseen tasks without the need for extensive retraining (Bommasani et al., 2022; Wei et al., 2022).

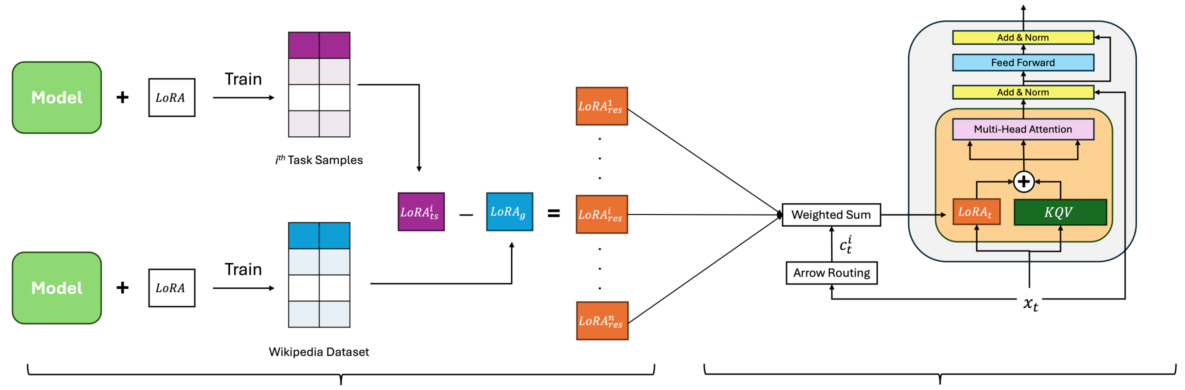
In modular zero-shot transfer approaches (Pfeiffer et al., 2023), a two-stage process is typically followed: (i) task-specific modules are obtained via parameter-efficient fine-tuning (PEFT) methods, such as LoRA (Hu et al., 2021), Adapters (Houlsby et al., 2019), and $(IA)^3$ (Liu et al., 2022), on a multitask dataset (ii) a routing function is used to select and combine task-specific modules to address a new task. While some routing functions require joint training alongside task-specific modules (Fedus et al., 2022; Caccia et al., 2023; Ponti et al., 2023), recent approaches employ post-hoc routing methods that require no further training (Chronopoulou et al., 2023; Ostapenko et al., 2024). Hybrid approaches also exist, where the routing function is trained separately on a downstream dataset after freezing task-specific modules (Muqeeth et al., 2024; Huang et al., 2024).

In this paper, we adopt LoRA as the PEFT module and Arrow (Ostapenko et al., 2024) as the routing function. We choose Arrow for its ability to dynamically route each input token—rather than the entire input—to the most relevant task-specific modules in a post-hoc manner, without requiring additional training. We hypothesize that *redundant general knowledge within task-specific modules hampers generalization.* To mitigate that, we build a general knowledge LoRA using a general corpus, and then subtract it from each task LoRA. We call this process *GenKnowSub*, general knowledge subtraction. The Arrow algorithm then dynamically selects and integrates the most relevant LoRAs for each input token. An overview of the proposed method can be found in Figure 1.

The core intuition behind GenKnowSub is that reducing redundant general knowledge while preserving essential task-specific knowledge improves the model's effectiveness in zero-shot transfer learning. By disentangling task-specific and general-domain knowledge, we prevent redundancy and enable better adaptation. Additionally, remov-

* Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 685–694
July 27 - August 1, 2025



(a) *Training the Modules and General Knowledge Subtraction* (b) *Dynamic Task Adaptation via Arrow Routing*

Figure 1: Overview of our proposed approach. (a) illustrates the process of training task-specific and general modules, followed by performing general knowledge subtraction, or GenKnowSub. (b) represents the dynamic task adaptation stage in a model layer, where the Arrow algorithm selects and combines the most relevant task-specific modules for each input token.

ing redundant knowledge enhances the distinctiveness of residual modules, ensuring that routing mechanisms can more effectively select and compose appropriate modules for solving new tasks.

We evaluate our approach mainly on Phi-3 (Abdin et al., 2024) in a large set of benchmarks across English, German, and French. Experimental results demonstrate noticeable performance gains when compared to the base and Arrow models, underscoring the effectiveness of GenKnowSub in reducing redundancy and enhancing task-specific generalization. We further experiment with Phi-2 (Javaheripi et al., 2023) model and show how our findings extend to this model which is more English-focused with less general capabilities.

Here are our key contributions: (i) We propose GenKnowSub, a novel approach for general knowledge disentanglement by subtracting a general LoRA from task-specific LoRAs. GenKnowSub is simple, scalable, and seamlessly adaptable, making it applicable to broader modular LLM frameworks. (ii) We experimentally show that GenKnowSub improves the standard Arrow method and the Phi-3 baseline performance across multiple benchmarks and languages.

2 Method

In this work, we address zero-shot transfer learning problem, where the goal is to transfer knowledge from a multitask dataset to solve unseen tasks without requiring labeled data for further training. Modular approaches have emerged as promising solutions for this problem. These methods operate by first training task-specific modules and then combining them to solve unseen tasks. Here, we propose to use general knowledge modules to enhance modularity detailed in the following sections.

2.1 Training Modules and General Knowledge Subtraction

LoRA (Hu et al., 2021) is a PEFT method (Han et al., 2024) that updates only a small set of low-rank trainable parameters while keeping the pre-trained model weights frozen. By training LoRA modules on a diverse set of tasks, we enable the acquisition of distinct task-specific skills. To effectively isolate the task-specific knowledge within each LoRA module, we leverage the principle of *forgetting via negation* (Ilharco et al., 2023) in module-level (Zhang et al., 2023). Specifically, we define *Residual LoRA* as follows:

$$LoRA^i_{res} = LoRA^i_{ts} - LoRA_g \tag{1}$$

where $LoRA^i_{ts}$ denotes the module trained on task i and $LoRA_g$ represents the general knowledge module. We name this approach as **GenKnowSub** representing general knowledge subtraction.

We hypothesize that fine-tuning the model with LoRA on even a small Wikipedia-like dataset with a causal language modeling objective could act as a bridge or a flashback for the model, bringing forth the general knowledge it acquired during pre-training. This allows the LoRA module to represent broader linguistic and factual knowledge embedded in the base model. This knowledge is redundant since the base model already contains it. Further, we assume that task-specific modules include some of these redundant knowledge alongside their specific functionality. Consequently, GenKnowSub effectively removes unnecessary general knowledge influence, isolating the unique task-specific characteristics essential for solving new unseen tasks.

2.2 Dynamic Task Adaptation

To enhance the adaptation to unseen tasks, we employ the Arrow routing algorithm introduced in Ostapenko et al. (2024), which dynamically selects the k best task-specific modules for each input token in each layer and integrates them to construct an optimal LoRA module for solving unseen tasks, based on the *learning via addition* principle (Ilharco et al., 2023). Arrow computes the SVD of each LoRA, extracts the top right singular vector as a prototype, and projects input tokens onto it. The top k coefficients are selected, softmax-normalized, and others set to zero.

Formally, we define the computed LoRA module in each layer of the model for each input token as:

$$LoRA_t^l = \sum_i^n c_t^{i,l} LoRA_{res}^{i,l} \tag{2}$$

where n is the number of trained task-specific modules, $LoRA_{res}^{i,l}$ represents the residual LoRA trained on task i within layer l of the model, and $c_t^{i,l}$ indicates the importance of $LoRA_{res}^{i,l}$ for the input token t, which is calculated using the Arrow algorithm based on the input in a zero-shot manner.

Given $LoRA_t^l$, the forward path for token t within layer l of the model is formulated as: $y_t^l = W_0^l x_t^l + B_t^l A_t^l x_t^l$ where $W_0^l \in \mathbb{R}^{d \times k}$ denotes the base model weights in layer l, $x_t^l \in \mathbb{R}^k$ is the input representation of token t entering layer l, $A_t^l \in \mathbb{R}^{r \times k}$, $B_t^l \in \mathbb{R}^{d \times r}$ are the corresponding LoRA parameter matrices associated with $LoRA_t^l$, and $r \ll \min(d, k)$ is the rank of low-rank decomposition. Figure 1 shows the overview of our proposed framework, including *Training the Modules*, *General Knowledge Subtraction (GenKnowSub)*, and *Dynamic Task Adaption* stages.

3 Experimental Setup and Results

Here, we first discuss some specifications of our experimental setup including how we build our LoRA modules, and then overview the results.

3.1 Constructing Task-Specific Modules

As stated earlier, the initial step for our proposed method, GenKnowSub, involves training modules, each tailored to a specific task or functionality. To avoid an excessive number of specialized modules, we utilize clustered Flan dataset (Longpre et al., 2023) proposed by (Ostapenko et al., 2024), which contains only English tasks. This dataset was constructed using a model-based clustering approach, where independent LoRAs were initially trained for each task and then clustered using the K-means algorithm. We assume that the clustering within this dataset effectively captures the relationships between tasks, regardless of the base model on which the LoRAs are trained. This assumption allows us to dedicate a single LoRA for each cluster of tasks, thereby reducing the number of experts required without compromising task-specific performance.

We select Phi-3-mini-4k-instruct (Abdin et al., 2024), a 3.8-billion-parameter model, for its strong instruction-following abilities and reasonable multilingual proficiency. To address hardware limitations, we use only 20% of each cluster's data ($\sim$2,000 samples) to improve training efficiency. The details regarding the setup and hyperparameters can be found in Appendix A.

3.2 Creating General Knowledge Modules

To obtain a module that effectively captures the general knowledge of a language, we train LoRAs on small Wikipedia corpora using causal language modeling. We select three higher-resource languages for Phi-3 model: English, French, and German. We assess their impact on GenKnowSub through various combinations across multilingual zero-shot benchmarks. For the Equation (1), we define $LoRA_g$ as follows: $LoRA_g = \{LoRA_{en}, LoRA_{de}, LoRA_{fr}, LoRA_{avg}\}$. Each LoRA is trained on 5,000 Wikipedia segments per language (details in Appendix A), with $LoRA_{avg}$ as their average.

3.3 Results

We compare GenKnowSub against a range of baselines to contextualize its performance. These include the base *Phi-3* model, *Arrow* (Ostapenko et al., 2024), and two additional ablations introduced in this work: (i) *Shared*, a single LoRA trained on the full multitask subset (20% of each cluster); and (ii) *Mean Normalization*, where the average of all task-specific LoRA modules is subtracted from

Method	Setting	PIQA	BOOLQ	SWAG	HSWAG	ARC-E	ARC-C	WG	OQA	BBH	Avg
Phi-3		78.24	81.47	68.99	73.59	71.75	44.48	**65.98**	42.80	42.83	63.35
Shared		80.00	63.39	71.00	72.16	78.77	49.83	54.46	45.40	41.21	61.80
Mean Norm		78.02	80.89	71.90	72.53	71.58	44.48	56.83	44.00	40.07	62.25
Arrow		80.20	80.00	68.95	71.89	80.53	53.85	**65.98**	47.40	41.23	65.56
GenKnowSub	*En*	80.20	81.96	70.00	73.36	82.11	53.85	64.72	48.40	43.30	66.43
	De	**80.30**	82.01	**73.30**	72.79	81.75	54.85	63.30	**49.80**	42.04	66.68
	Fr	78.78	82.11	71.64	**74.02**	81.75	**57.19**	64.40	49.00	**44.40**	67.03
	Avg	80.03	**82.45**	72.70	73.45	**82.28**	55.85	64.64	49.60	43.51	**67.17**

Table 1: Comparison of accuracy across different methods using Phi-3 as the base model on some **English** reasoning datasets in a zero-shot setting. The "Setting" column refers to the general knowledge LoRA used for subtraction in GenKnowSub— e.g., 'En' indicates subtraction of the English general LoRA.

each individual module as an alternative means of removing redundant information. While other recent approaches—such as Poly (Ponti et al., 2023) and MHR (Caccia et al., 2023)—offer additional insight into modular routing, they rely on joint training of experts and routing mechanisms, and are therefore not directly comparable in our setup.

Table 1 presents the performance on nine **English reasoning benchmark datasets**, including PIQA (Bisk et al., 2019), BoolQ (Clark et al., 2019), SWAG (Zellers et al., 2018), HellaSwag (Zellers et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), BIG-Bench Hard (Suzgun et al., 2023), and OpenBookQA (Mihaylov et al., 2018). We evaluate the impact of the different configurations of GenKnowSub on dynamic task adaptation.

As shown in Table 1, modular approaches significantly outperform the non-modular baseline. Specifically, both *Arrow* and *GenKnowSub* improve substantially over the *Shared* baseline, confirming the effectiveness of modularity in parameter-efficient transfer. Beyond this, *GenKnowSub* further enhances performance over *Arrow* by removing redundant general knowledge from task-specific modules. When using the average general LoRA for subtraction, GenKnowSub achieves a consistent gain of 1.6% over Arrow. Notably, this performance is not matched by the *Mean Normalization* baseline, which naively subtracts the average of task-specific modules and yields inconsistent improvements. This highlights that our targeted subtraction of language-informed general knowledge is key to the gains observed. Finally, the effectiveness of GenKnowSub is also evident when subtracting individual language-specific LoRAs, suggesting that even language-tied general modules encode broadly shared knowledge.

To evaluate the effectiveness of GenKnowSub beyond multiple-choice settings, we conduct experiments on the Super-Natural Instructions (SNI) dataset (Wang et al., 2022), a large and diverse benchmark for open-ended generation tasks. We specifically select SNI to maintain a strict zero-shot setting, consistent with our earlier evaluations on multiple-choice tasks. Additionally, SNI has been widely used to assess generalization ability in modular approaches, including in the original Arrow (Ostapenko et al., 2024) work. Evaluating on 10,000 randomly sampled test examples covering 119 open-ended tasks, *GenKnowSub* achieves a Rouge-L score of 46.91, outperforming the base *Phi-3* model (42.85), *Arrow* (45.44), the *Shared* baseline (34.48), and the *Mean Normalization* baseline (43.07). These results further demonstrate the generality and effectiveness of our approach in open-ended generation tasks under a zero-shot setting.

	Method	Setting	HSWAG	ARC-C	XNLI	MMLU	Avg
German	**Phi-3**		**52.48**	36.24	36.02	33.82	39.64
	Shared		49.75	38.93	43.00	36.00	41.92
	Mean Norm		51.00	36.91	33.67	33.50	38.77
	Arrow		48.58	40.94	43.45	35.40	42.09
	GenKnowSub	*En*	51.16	40.60	50.14	36.85	44.69
		De	50.58	**42.95**	50.42	37.00	45.24
		Fr	50.58	42.62	49.17	37.17	44.88
		Avg	51.08	42.62	**52.33**	**37.92**	**45.99**
French	**Phi-3**		57.67	34.56	50.75	33.33	44.08
	Shared		57.08	40.27	50.17	35.33	45.71
	Mean Norm		57.42	35.91	52.42	33.25	44.75
	Arrow		55.33	41.61	44.38	34.79	44.02
	GenKnowSub	*En*	56.08	41.95	50.66	**36.69**	46.34
		Fr	**57.83**	**42.95**	**53.65**	36.13	**47.64**
		De	56.42	42.28	46.33	35.58	45.15
		Avg	57.08	42.62	52.92	35.58	47.05

Table 2: Performance comparison of different methods with Phi-3 as the base model in a zero-shot setting for **German** and **French** languages. Various configurations of GenKnowSub are evaluated, with accuracy as the reported metric.

To evaluate GenKnowSub across **non-English languages**, we use XNLI (Conneau et al., 2018), the translated versions of the HellaSwag, MMLU (Hendrycks et al., 2021), and ARC-Challenge datasets provided by (Lai et al., 2023). Table 2 shows that *GenKnowSub* consistently achieves the best performance across both German and French benchmarks. In contrast, *Arrow* exhibits inconsistent results—outperforming *Shared* and *Mean Norm* in German, but falling behind them in French—highlighting its variability across languages. GenKnowSub surpasses all baselines in both settings, with its strongest configuration (average subtraction) improving over Arrow by 3.9% in German and 3.6% in French. These results confirm the generality and robustness of our approach, and reinforce the findings from the previous experiments on the English benchmark datasets: removing shared general knowledge from task-specific modules before task adaptation leads to more effective zero-shot transfer across languages.

A key factor in cross-lingual transfer learning is the base model's ability to encode at least a minimal level of multilinguality. To assess its impact more precisely, we run an additional experiment using **Phi-2**, which is weaker than Phi-3 in both multilingual and instruction-following capability. Following (Ostapenko et al., 2024), we use unquantized Phi-2 here, with task modules trained on the full task cluster data. As shown in Table 3 in Appendix, GenKnowSub, after subtracting the English general knowledge module, improves performance on English benchmark datasets in a zero-shot setting, increasing the average score by 1.1%. However, in German and French experiments (Table 4 in Appendix), both the base Phi-2 model and its combination with Arrow perform poorly—around 13% lower than Phi-3 in a similar setting—due to Phi-2's weak multilingual capabilities. Consequently, GenKnowSub achieves only comparable results, underperforming Arrow by 0.3%. These findings further confirm that our approach can enhance performance, provided the base model has at least a minimal level of multilinguality. Additional details and results are provided in Appendix B.

4 Conclusion

In this work, we propose a modular approach to zero-shot transfer learning, leveraging task-specific and general knowledge modules to enhance adaptability to unseen tasks. Our method first isolates task-relevant representations through GenKnowSub, then dynamically adapts these modules using the Arrow routing algorithm. By minimizing redundancy in task representations, our approach improves both efficiency and transferability. We demonstrate that applying GenKnowSub prior to task adaptation improves generalization in zero-shot settings for both Phi-3 and Phi-2 models across both multiple-choice and open-ended generation tasks. Our results show that this method not only enhances performance in monolingual tasks but also facilitates effective cross-lingual transfer when the language is highly present in the base model. Future work includes exploring alternative task adaptation methods, extending our approach to additional languages, especially low-resource ones, and testing it on other models.

Limitations

One limitation of this study is the restricted scope of model evaluation. Due to hardware constraints—such as limited GPU VRAM and slower processing speeds—we limited our experiments to two models: Phi-3 and Phi-2. These constraints precluded testing on larger or more diverse models, thereby limiting the breadth of our analysis. Additionally, although we included multilingual evaluations, our focus remained on high-resource languages (e.g., English, French, German), and we did not investigate performance in low-resource settings. Future work should aim to expand the evaluation to a broader range of models, tasks, and languages to further assess the generality of the proposed approach.

Ethics

Our research utilizes publicly available datasets and pre-trained models, ensuring compliance with ethical data usage practices and avoiding the use of private, proprietary, or personally identifiable information. All models and associated code will be made publicly available under permissive licenses, promoting accessibility, reproducibility, and unrestricted use for research and application development. However, we acknowledge that pre-trained language models (PLMs) and large language models (LLMs) have been shown to exhibit biases, as highlighted in prior work (Liang et al., 2021; He et al., 2023). Users should be mindful of these limitations when applying such models in practice. Our work does not introduce additional fairness or privacy concerns.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. 2023. Multi-head adapter routing for cross-task generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

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

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Preprint*, arXiv:2101.03961.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*.

Junheng He, Nankai Lin, Menglan Shen, Dong Zhou, and Aimin Yang. 2023. Exploring bias evaluation techniques for quantifying large language model biases. In *2023 International Conference on Asian Language Processing (IALP)*, pages 265–270.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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 *First Conference on Language Modeling*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *ArXiv*, abs/2307.16039.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2024. Loftq: LoRA-fine-tuning-aware quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. 2024. Learning to route among specialized experts for zero-shot generalization. *ArXiv*, abs/2402.05859.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. 2024. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward

Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. 2023. Combining parameter-efficient modules for task-level generalisation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 687–702, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

WikimediaFoundation. Wikimedia downloads.

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

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Implementation Details

A.1 Base Model

We utilize Phi-3-mini-4k-instruct (Abdin et al., 2024) with 4-bit quantization to reduce memory usage while maintaining strong performance. We selected this model due to its exceptional instruction-following capabilities and its acceptable multilingual proficiency. Additionally, with only 3.8 billion parameters, Phi-3-mini strikes an effective balance between model size and performance, allowing for efficient fine-tuning and deployment in resource-constrained environments while still demonstrating competitive reasoning and generalization abilities.

A.2 PEFT Structure

As the PEFT structure, we employ LoftQ (Li et al., 2024) with a rank of $r = 4$. LoftQ extends LoRA by integrating low-rank adaptation directly into the quantization process, thereby optimizing both

Method	Setting	PIQA	BOOLQ	SWAG	HSWAG	ARC-E	ARC-C	WG	OQA	BBH	Avg
Phi-2		78.99	81.16	63.50	66.75	82.11	53.51	56.51	44.00	48.00	63.84
Arrow		79.65	81.13	65.75	66.41	83.38	54.84	60.85	48.60	54.75	65.15
GenKnowSub	*En*	79.97	80.12	65.58	66.75	84.38	54.51	61.24	49.80	54.00	**66.26**
	Avg	80.47	78.47	66.10	67.96	84.03	56.19	60.69	47.80	54.00	66.19

Table 3: Comparison of accuracy across different methods using Phi-2 as the base model on English datasets in a zero-shot setting, with Accuracy as the reported metric.

	Method	Setting	HSWAG	ARC-C	XNLI	MMLU	Avg
German	**Phi-2**		28.78	23.84	34.50	24.19	27.83
	Arrow		28.75	24.83	32.50	26.91	**28.25**
	GenKnowSub	*En*	28.33	24.55	33.33	24.91	27.78
		Avg	28.42	23.49	34.33	25.50	27.93
French	**Phi-2**		33.33	26.84	34.16	24.77	29.77
	Arrow		32.91	27.51	34.16	25.68	**30.06**
	GenKnowSub	*En*	33.50	25.50	31.83	26.11	29.23
		Avg	32.25	24.16	37.50	25.00	29.73

Table 4: Performance comparison of different methods using Phi-2 as the base model on multilingual datasets in a zero-shot setting for German and French, with Accuracy as the reported metric.

fine-tuning and inference through rank-wise quantization, which minimizes precision loss while updating quantized model weights. We applied our PEFT modules to both the QKV components (concatenation of Query, Key, and Value matrices in the self-attention block) and the output projection layer of the multi-head attention.

A.3 Arrow Routing

To incorporate the Arrow routing algorithm, we implemented it from scratch using PyTorch (Paszke et al., 2019) and the PEFT library from HuggingFace (Wolf et al., 2020). We trained 10 task-specific modules, and selected the 3 best modules for each input token in each layer of the model to be combined.

A.4 Module Training

To train the LoRA modules representing general knowledge, we fine-tuned the model on a Wikipedia dataset using a causal language modeling objective on a single Quadro RTX 6000 GPU. We ensured consistency across languages by sampling exactly 5,000 segments per language from the Hugging Face wikimedia/wikisource (WikimediaFoundation) dataset. Each segment contained 512 words, which were split into a 507-word prompt and a 5-word completion. LoRA modules were fine-tuned using the same supervised setup for all languages. This uniform approach ensured that the amount, structure, and formatting of training data were identical for each language, mitigating any length- or volume-based bias.

For task-specific LoRA modules, we employed a supervised fine-tuning approach. Given the presence of relatively long examples in our dataset, we set the maximum sequence length to 4000 tokens to accommodate the full input structure.

Both the task-specific and general knowledge LoRA modules were trained for one epoch with a learning rate of $1e{-}4$, using cosine scheduling with a warmup start. To stabilize training, we applied gradient clipping. Additionally, to optimize memory efficiency, we utilized the Paged AdamW 8-bit optimizer (Dettmers et al., 2022), a quantized variant of AdamW (Loshchilov and Hutter, 2019), designed to reduce GPU memory consumption.

The batch size was set to 16 for training general knowledge modules and 1 for task-specific modules, due to the long input lengths. To further improve memory efficiency, we applied gradient checkpointing and gradient accumulation, enabling support for larger batch sizes when training task-specific modules.

B Experiments on Phi-2

B.1 Implementation Details

For conducting experiments using Phi-2, the baseline in Ostapenko et al. (2024), we used the LoRA modules they trained to evaluate their proposed routing algorithm, Arrow (the implementation details of Arrow are provided in Appendix A.3). Specifically, we used task LoRA modules trained by Ostapenko et al. (2024), available on Hugging Face.[1] These modules are obtained by fine-tuning Phi-2 on clustered Flan datasets (explained in Section 3.1) and are provided in PyTorch Lightning format.[2] Since our models are trained using the PEFT

[1]Library of LoRAs: `https://huggingface.co/zhan1993/mbc_library_phi2_icml`

[2]PyTorch Lightning: `https://github.com/Lightning-AI/pytorch-lightning`

library,[3] we converted the existing LoRA weights into the PEFT format to ensure compatibility. Our implementation loads expert weights following the PEFT framework, and, consistent with their setup, the Phi-2 experiment weights remain unquantized.

Additionally, we obtained general knowledge LoRA modules by fine-tuning Phi-2 in a setup aligned with the task-specific LoRA modules. The training process was similar to that of the Phi-3 version, as detailed in Appendix A.4.

B.2 Resutls

We demonstrated that with a sufficiently strong multilingual base model, we can effectively leverage its multilingual capabilities to generalize better to unseen tasks across different languages. The Phi-2 experiments further highlight the importance of base model strength and knowledge. As shown in Table 3, GenKnowSub, with subtracting the English general knowledge module, outperforms other settings, whereas averaging modules across different languages is less effective than using English alone. Additionally, Table 4 shows that all settings, including Phi-2 and Arrow, perform poorly in German and French. The improvement of our method on the English zero-shot dataset, along with its performance in the multilingual setting, demonstrates that our method can significantly enhance results—provided the base model exhibits at least a minimal level of cross-lingual capability.

[3]PEFT: https://github.com/huggingface/peft