Towards Modular LLMs by Building and Reusing a Library of LoRAs

Oleksiy Ostapenko[* 1 2 3] Zhan Su[* 2 4]
Edoardo Maria Ponti[5] Laurent Charlin[2 6 7] Nicolas Le Roux[1 2 3 7] Lucas Caccia[* 1] Alessandro Sordoni[* 1 2 3]

Abstract

Given the increasing number of parameter-efficient adapters of large language models (LLMs), how can we reuse them to improve LLM performance on new tasks? We study how to best build a *library* of adapters given multi-task data and devise techniques for both *zero-shot* and *supervised* task generalization through *routing* in such library. We benchmark existing approaches to build this library and introduce model-based clustering, MBC, a method that groups tasks based on the similarity of their adapter parameters, indirectly optimizing for transfer across tasks. In order to reuse the library, we present a novel zero-shot routing mechanism, Arrow, which enables dynamic selection of the most relevant adapters for new inputs without the need for retraining. We experiment with several LLMs, such as Phi-2 and Mistral, on a wide array of held-out tasks, verifying that MBC-based adapters and Arrow routing lead to superior generalization to new tasks. Thus, we make steps towards creating modular, adaptable LLMs that can match or outperform traditional joint training.

1. Introduction

Tailoring large language models (LLMs) towards downstream tasks, domains, or user profiles is of paramount importance given the recent democratization of their usage, catalyzed by the release of open-source LLMs (Zhang et al., 2023b; Microsoft Research, 2023, *inter alia*). This process often relies on an *adapter*, such as LoRA (Hu et al., 2022), a parameter-efficient fine-tuning (PEFT) of a pre-trained LLM (Hu et al., 2022; Liu et al., 2022; Li & Liang, 2021).

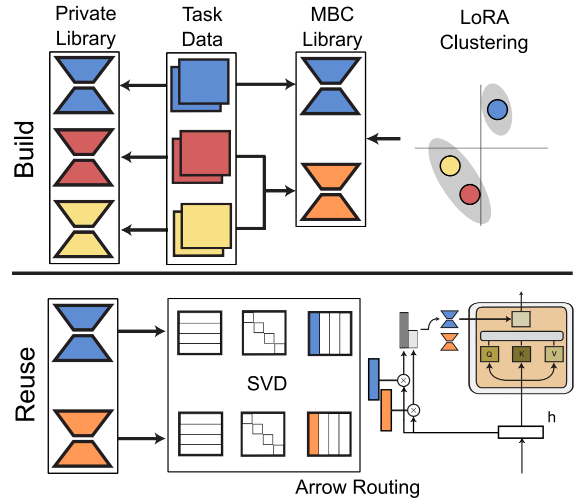

Figure 1. How to coordinate a library of adapters (e.g., LoRAs) for zero-shot generalization to new tasks? To **build** this library (top), we propose MBC, a novel method that clusters tasks based on the similarity of the parameters of corresponding LoRAs. To **reuse** a library (either private or MBC, bottom), we route hidden states to trained LoRAs via Arrow, which leverages the SVD decomposition of each LoRA.

LLM adapters are increasingly available as part of online hubs (Beck et al., 2021; Mangrulkar et al., 2022). These adapters are developed independently and asynchronously by users across the globe. Hence, they implicitly constitute a library built on top of multi-task data (Pfeiffer et al., 2023). Prior works show that mixtures of pretrained adapters can facilitate few-shot adaptation of LLMs to *unseen tasks* (Ponti et al., 2023; Vu et al., 2021; Huang et al., 2024). Reusing pre-existing adapters in a zero-shot fashion remains less explored (Jang et al., 2023; Belofsky, 2023). In contrast to standard mixture-of-experts approaches (Fedus et al., 2022), in this setting, new inputs must be routed to independently trained experts without requiring joint training of the routing mechanism and expert parameters.

This leads to the question: how to create a modular LLM end-to-end by first building and then reusing a library of adapters for supervised adaptation and zero-shot generalization? First, given a base LLM, such as Phi-2 (Microsoft Research, 2023) or Mistral (Jiang et al., 2023), we investigate building a library of adapters by leveraging 256 tasks from

[*]Equal contribution [1]Microsoft Research [2]Mila — Quebec AI Institute [3]Université de Montréal [4]University of Copenhagen [5]University of Edinburgh [6]HEC Montréal [7]Canada CIFAR AI Chair. Correspondence to: A. Sordoni <alsordon@microsoft.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Flan-v2 (Longpre et al., 2023).[1] We focus on LoRA (Hu et al., 2022) and leave the extension to other adapter types for future work. Once the adapter library has been built, we devise routing strategies to evaluate *zero-shot* generalization on 10 downstream tasks comprising common-sense reasoning and coding (ARC (Clark et al., 2018), MBPP (Austin et al., 2021), *inter alia*) and *supervised adaptation* on 12 SuperNatural Instructions (SNI) tasks (Wang et al., 2022b).

How to build the adapter library? One straightforward approach is to operate in a *private* scenario, in which one trains one adapter per task on the multi-task data and mixes those adapters for unseen tasks (Chronopoulou et al., 2023a; Vu et al., 2021; Huang et al., 2024). This is useful when the multi-task data cannot be shared for joint training (Mireshghallah et al., 2020) but trained adapters can. To favour transfer between training tasks, recent approaches compress the multi-task data into a smaller set of reusable, composable adapters (Ponti et al., 2023; Caccia et al., 2023). In this shared data setting, we propose model-based clustering (MBC), a simple two-stage approach to build an adapter library. We find a positive correlation between the similarity of the LoRA weights of a pair of tasks and the transfer between these tasks. Building on this intuition, we first exploit LoRA similarity in weight space between privately trained adapters as a proxy for detecting clusters of similar tasks, then train one adapter per cluster. Our approach empirically improves performance while matching the compute budget.

How to reuse the library for new scenarios? Given a library of trained LoRAs, we examine strategies of how to reuse the library in two settings: *zero-shot generalization* and parameter-efficient *supervised adaptation* to new tasks. Reusing LoRAs in a zero-shot manner is challenging because there is no labelled data to learn a routing mechanism. We propose Arrow (↗), a routing mechanism that automatically selects relevant LoRAs without requiring *i)* joint training and *ii)* access to the data used to train each LoRA, facilitating the vision of a decentralized system where LoRAs can be trained asynchronously and be readily reused with minimal assumptions. Arrow computes a representation for each LoRA as the direction of maximum variance induced by the LoRA parameters. At inference time, Arrow routes *per token* and *per layer*, i.e. each hidden state is routed by computing its alignment with each LoRA representation.

In summary, our contributions are: *i)* we study how to create LoRA-based modular multi-task LLM in a setting where experts are trained independently and the router is created *after* the training of the experts; *ii)* assuming shared multi-task data, we propose a clustering approach (MBC) to train a library of adapters; and, *iii)* we propose Arrow, a zero-shot routing method to select which adapters to use from a library of LoRAs. This allows for routing to independently trained experts without accessing their training data.

2. Preliminaries

We are given a set of tasks $\mathcal{T} = \{t_1, \ldots, t_T\}$, where each task t_i is associated with a dataset containing a set of samples $\mathcal{D}_i = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$. The union of the training sets constitutes our multi-task dataset $\mathcal{D}$; in our case, it is Flan (Longpre et al., 2023). In order to create our library of task adapters, we use LoRA (Hu et al., 2022). LoRA achieves competitive trade-offs between performance and parameter efficiency (Karimi Mahabadi et al., 2021) by modifying the linear transformations in a base LM. For each linear transformation in the base LM, LoRA modifies the base model parameters as follows:

$$h = W\mathbf{x} + s \cdot AB^\top \mathbf{x}, \qquad \text{(LoRA)}$$

where W are the (frozen) weights of the base LM, $A, B \in \mathbb{R}^{d \times r}$ are low-rank learnable parameters and $s \geq 1$ is a tunable scalar hyperparameter. LoRA achieves parameter efficiency because of the reduced rank r ($\ll d$).

3. Building the LoRA Library

We propose different alternatives for building a library $\mathcal{L}$ of adapters that perform well on the tasks they were trained on and are versatile enough to be effective on other unseen downstream tasks. To do so, we seek methods that enhance multi-task transfer while reducing task interference (Wang et al., 2021; Chen et al., 2022).

Private **Adapters** One straightforward solution is to train separate adapters on each training task, i.e. the library will be composed of T adapters (see Fig. 1). Several existing methods operate in this setting, such as LoraHub (Huang et al., 2024), AdapterSoup (Chronopoulou et al., 2023a) and SPoT (Vu et al., 2021). Although this solution does not exploit multi-task training, it is required in settings where the task data is private, e.g., user data, and cannot be shared. Moreover, this setting reflects well the scenario in which adapters are trained by end users in a decentralized fashion and added asynchronously to the library.

Shared **Adapter** To encourage transfer, another solution is to train a single adapter on all the multi-task training data. One possible shortcoming is the reduced capacity to fit the multi-task training data and the possibility of interference between the training tasks (Ponti et al., 2023). Training a single adapter may result in negative transfer because task gradients are misaligned (Wang et al., 2021). An obvious solution to reduce the amount of interference is to increase the number of trainable parameters, e.g. to fine-tune the whole base LM on the multi-task data (Liu et al., 2022).

Poly **/** MHR **Adapters** Polytropon (Poly) and Multi-Head Routing (MHR) (Ponti et al., 2023; Caccia et al., 2023)

[1] We held out SNI tasks to test supervised adaptation.

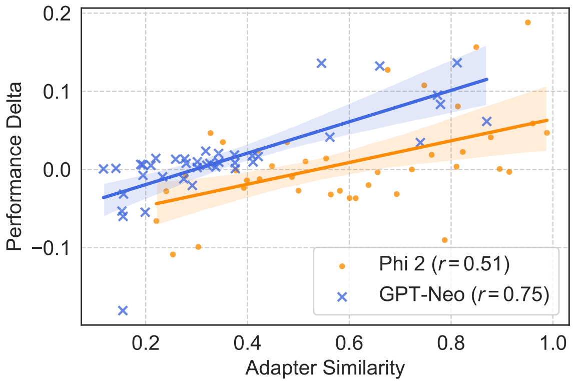

Figure 2. For any pair of tasks, we report the cosine similarity between the corresponding LoRA weights (x-axis) against the delta in performance between LoRAs trained on them individually and jointly (y-axis). The positive correlation indicates that if LoRAs are dissimilar, we should abstain from multi-task training.

explore intermediate approaches between private and shared, where $K < T$ "basis" adapters are trained on the multi-task training data. These K adapters can be considered "latent skills", as each task adapter in the multi-task training set can be expressed as a linear combination of these basis adapters. If private training for all the tasks learns a matrix of parameters $\Phi \in \mathbb{R}^{T\times D}$, where D is the dimensionality of the LoRA adapters, Poly decomposes $\Phi = Z\hat{\Phi}$, where $Z \in \mathbb{R}^{T\times K}$, $\hat{\Phi} \in \mathbb{R}^{K\times D}$, $\hat{\Phi}$ storing the latent skills and Z the linear combination coefficients for each task which specify the task-specific routing w.r.t. the latent skills. Both Z and $\hat{\Phi}$ are trained jointly on the multi-task training set by gradient descent. Note that the skills $\hat{\Phi}$ do not correspond to specific tasks and therefore it is not clear how to reuse them for zero-shot generalization (Caccia et al., 2023).

Model-Based Clustering (MBC) While Polytropon and MHR reduce the inventory size, they require joint training of experts and the router on the combined dataset of all tasks. Here, we propose another approach to compress multi-task data into a set of reusable adapters; we cluster tasks based on their similarity and then train one adapter per task cluster. Ideally, the similarity between two tasks should correlate with the benefit of training a single model on both tasks compared to having two independent models (Fifty et al., 2021; Vu et al., 2020a). Motivated by (Zhou et al., 2022), we rely on the intuition that LoRA parameter vectors similarity can approximate the amount of transfer between a pair of tasks. To confirm this, we devise the following experiment: we sample pairs of tasks $(t_i, t_j), t \in \mathcal{T}$ from the multi-task dataset, and we train both a) a LoRA on each task independently b) a LoRA on the union of the training datasets for the two tasks. We then compute the cosine *similarity* between the flattened LoRA parameters. We quantify *transfer* as the difference in the average log-likelihood induced by the joint and private models when evaluated on the test set of the two tasks. In Fig. 2, we observe that, for two different base models (GPT-Neo and Phi-2), the higher LoRA parameter similarity, the higher the performance delta when we train on the joint dataset.

Algorithm 1 Model-Based Clustering (MBC)

```
Input: Multi-task data $\mathcal{D}_1, \ldots, \mathcal{D}_T$, base model $\text{LLM}_\theta$, number of library adapters $K$
Output: Library $\mathcal{L}$
$\mathcal{L} = \{\}, \mathcal{A} = \{\}$                                ▷ LoRA params
for $t = 1$ to $T$ do
    $A_t, B_t = \text{train}(\mathcal{D}_t, \text{LLM}_\theta)$         ▷ Train LoRA on task t
    $\mathcal{A} = \mathcal{A} \cup \{\text{cat}(\text{flatten}(A_t),\ \text{flatten}(B_t))\}$
end for
$U = \text{SVD}(\mathcal{A})$                                         ▷ Reduce LoRA dim
$S = \text{cosine-similarity}(U, U)$                                  ▷ $T \times T$ similarities
$c_1, \ldots, c_K = \text{k-means}(S, K)$                             ▷ Cluster similarities
for $k = 1$ to $K$ do
    $\mathcal{D}_k = \bigcup \mathcal{D}_t, \forall t \in c_k$                 ▷ Join datasets in cluster
    $A_k, B_k = \text{train}(\mathcal{D}_k, \text{LLM}_\theta)$
    $\mathcal{L} = \mathcal{L} \cup \{(A_k, B_k)\}$
end for
Returns $\mathcal{L}$
```

The previous observation warrants our simple two-stage training procedure illustrated in Fig. 1 (top). Given a fixed computation training budget of N training steps per task, we use the first n steps to train private LoRAs. We then use these LoRA parameters to group tasks into K clusters by running a standard clustering algorithm (K-Means). In the second stage of training, we train one adapter per cluster for an additional $N - n$ training steps, which keeps the total amount of computation similar to other approaches. We refer to this method as Model-Based Clustering (MBC) as it uses the model-based information encoded in the weights to determine a similarity metric between tasks (see Alg. 1).

4. Reusing the LoRA Library

Next, we study the reuse of a trained library $\mathcal{L}$ in two scenarios: for new inputs $\mathbf{x}^*$, i.e. *zero-shot*, and in a *supervised adaptation* setting, where new tasks t^* come equipped with their training data $\mathcal{D}_{t^*}$. While the latter has been addressed in recent works (Huang et al., 2024; Caccia et al., 2023; Vu et al., 2021), the former scenario remains less explored (Jang et al., 2023; Belofsky, 2023). We first devise routing strategies in the zero-shot and supervised settings and then describe how to aggregate the contributions of adapters selected by the routing strategies.

4.1. Routing

We denote the hidden state for any token at a given transformer layer produced by the input token $\mathbf{x}^*$ as $\mathbf{h}^*$. Similar to MoE approaches, we seek to parameterize a layer-specific routing distribution that prescribes which adapters to use. We denote this categorical distribution over $|\mathcal{L}|$ outcomes as

$p(\cdot \mid \mathbf{h}^*, \mathbf{x}^*)$, where we drop the dependence on the layer for simplicity. For example, in standard MoE approaches (Fedus et al., 2022), $p(\cdot \mid \mathbf{h}^*, \mathbf{x}^*) = \text{softmax}(W\mathbf{h}^*)$. Given that we relax the assumption that the routing and the library should be trained together, we must devise ways to learn such routing distribution a posteriori.

4.1.1. Zero-Shot Routing

μ **Routing** One straightforward way to route to existing experts is to set the routing distribution to uniform for all layers, $p(\cdot \mid \mathbf{h}^*, \mathbf{x}^*) = [1/|\mathcal{L}|, \ldots, 1/|\mathcal{L}|]$. Despite its simplicity, μ routing was shown to be quite effective in recent work (Caccia et al., 2023; Chronopoulou et al., 2023a) and, due to the linearity of the LoRA adapters, effectively boils down to averaging the weights of the adapters uniformly.

`TP` **Routing** treats routing as an $|\mathcal{L}|$-way classification problem. Specifically, given an input $\mathbf{x}$ belonging to task t in our multi-task training set, we train a task predictor f by minimizing the categorical cross-entropy loss $-\log f(\mathbf{x})[t]$, where $f(\mathbf{x})$ is a probability distribution obtained by learning a classifier on top of a T5 encoder (Raffel et al., 2020). We then set $p(\cdot \mid \mathbf{h}^*, \mathbf{x}^*) = f(\mathbf{x}^*)$ at inference time. Note that the routing decisions are not dependent on the hidden state $\mathbf{h}^*$, so this is a router dependent on the whole input but independent of the particular token or layer in the Transformer. We call this predictor `TP` (Task Predictor).

`CM` **Routing** computes expert prototypes (for each layer) by averaging the hidden representations obtained by a forward pass of the LLM on each expert dataset (Centroid Matching). These prototypes can be stored in the columns of the routing matrix W. Once the prototypes for each expert have been obtained, the routing distribution is calculated by taking the cosine similarity between $\mathbf{h}^*$ and each expert prototype and finally applying softmax. This routing is similar in spirit to Jang et al. (2023) and Belofsky (2023).

Arrow Routing ↗ The rows of every routing matrix W of standard MoE routing can be interpreted as expert "prototypes". `Arrow` prescribes a way to estimate such routing matrix in a 0-shot fashion without requiring data access. Let's denote by $\{A_i, B_i\}$ the parameters for expert i at layer ℓ, where we drop the dependency on ℓ. The i-th LoRA expert transforms each token's hidden state $\mathbf{h}^*$ as $\mathbf{h}_i^* = A_i B_i^T \mathbf{h}^*$. `Arrow` finds a prototype for the expert i by decomposing the outer product $A_i B_i^T$ with SVD and taking the right first singular vector of this transformation (see Alg. 2). The prototype determines the direction of most variance induced by expert i in the space of hidden states $\mathbf{h}$. If the LoRA adapters are of rank 1, i.e. $A_i, B_i \in \mathbb{D}^{D\times 1}$ the prototype for the expert i will be equal to the normalized B_i vector, i.e. $\text{argmax}_{\mathbf{v}, \|\mathbf{v}\|_2=1} \|A_i B_i^T \mathbf{v}\|_2 = B_i / \|B_i\|_2$. In Section 8.1, we provide empirical evidence that indeed, $\|A_i B_i^T \mathbf{v}\|_2$ is larger when $\mathbf{v}$ belongs to task i, thus motivating this routing approach. Given that both $\mathbf{v}$ and $-\mathbf{v}$ are valid singular vectors, we compute expert logits as the absolute value of the dot product between prototypes and inputs. Alg. 2 details the prototype initialization and the routing step of `Arrow`.

Algorithm 2 `Arrow` Routing ↗

```
Weight Initialization
  Input: LoRA library $\mathcal{L}$, layer $\ell$
  Output: Routing parameters for layer $\ell$: $W_\ell$
  for $i = 1$ to $L$ do
    $A_i, B_i = \mathcal{L}[i, \ell]$                ▷ Get weights for expert i
    $U, D, V = \text{SVD}(A_i B_i^T)$
    $W_\ell[i] = V[:, 0]$                  ▷ First right singular vector
  end for
  Returns $W_\ell$

Routing
  Input: Routing parameters for layer $\ell$: $W_\ell \in \mathbb{R}^{|\mathcal{L}|\times d}$, token in
         layer $\ell$: $\mathbf{h}_\ell \in \mathbb{R}^d$, top-k routing: $k$
  Output: Routing probabilities for layer $\ell$: $\mathbf{p}_\ell$
  logits = abs($W_\ell \mathbf{h}_\ell$)
  $p_\ell[i] = \begin{cases} \text{logits}[i] & \text{if } i \in \text{arg top-}k(\text{logits}) \\ -\infty & \text{else} \end{cases}$
  Returns softmax($\mathbf{p}_\ell$)
```

`Arrow` offers several advantages: a) it doesn't require access to training data; b) it routes differently in every layer and token, increasing the overall model expressivity, and c) it is compute efficient since it requires no further training and SVD decomposition can be computed efficiently for low-rank matrices (Elhage et al., 2021; Nakatsukasa, 2019).

4.1.2. Supervised Task Routing

When generalizing to a new task, we can learn the optimal routing given the task data $\mathcal{D}^*$. This setting is similar to previous task generalization works (Ponti et al., 2023; Caccia et al., 2023; Huang et al., 2024). We compare results in this supervised setting to both Poly (Ponti et al., 2023) and LoraHub (Huang et al., 2024).

`Poly` **Routing** treats the distribution over experts at each layer as an $|\mathcal{L}|$-dimensional parameter that is learned by minimizing the cross-entropy on the new task data $\mathcal{D}^*$. It optimizes the merging coefficients of LoRAs for the new task, i.e. $A^* = \sum_{i=1}^{|\mathcal{L}|} w^i A_i$ and $B^* = \sum_{i=1}^{|\mathcal{L}|} w^i B_i$. Here $p(\cdot|\mathbf{h}^*, \mathbf{x}) = (w^1, \ldots, w^n)$ is the (input-independent) learnable routing distribution for a given layer.

`LoraHub` **Routing** (Huang et al., 2024) is similar to `Poly` with the exception that a) it resorts to gradient-free optimization to learn routing coefficients and b) it doesn't fine-tune the experts' parameters, making it less expressive than `Poly`.

`π-tuning` **Routing** uses the Fisher Information Matrix(FIM) to create an embedding for each task-specific expert. In the fine-tuning process, π-tuning first trains an expert for the target task, and then it retrieves a subset of experts most similar to the target task's expert using FIM embeddings. Finally, both the interpolation coefficients and experts' parameters are tuned on the target task's data (Wu et al., 2023).

4.2. LoRA Composition

Given a routing distribution $\mathbf{w} = p(\cdot \mid \mathbf{h}^*, \mathbf{x})$ obtained either using the previously presented zero-shot or supervised routing, we linearly combine adapters in the library, i.e. $A^* = \sum_{i=1}^{|\mathcal{L}|} w_i A_i$, $B^* = \sum_{i=1}^{|\mathcal{L}|} w_i B_i$ and use the resulting adapter to perform inference at every layer of the base LLM (Ponti et al., 2023; Huang et al., 2024). For 0-shot task generalization, we employ top-k routing, composing the k experts with the highest routing logits.

5. Experiments

Our experimental evaluation aims to answer the following questions: 1) How does building a LoRA library compare to non-modular methods (e.g. full fine-tuning)? 2) How large is the gap between privately trained libraries (similar to online hubs) and libraries which assume access to multi-task data? 3) To what extent does routing facilitate reusing a library of LoRA adapters?

Multi-Task Dataset We train expert modules on 256 tasks from the original Flan v2 dataset (Longpre et al., 2023). We exclude the SNI tasks ($>$ 1000 tasks) (Wang et al., 2022b) from training for computational reasons, and reserve 12 SNI tasks for downstream out-of-domain evaluation.

Evaluation For our supervised adaptation study, we use 12 held-out SNI tasks, each corresponding to a different SNI category. We threshold the number of training examples to 10,000 examples per task and reserve 1,000 for validation. We evaluate performance with Rouge-L scores (Lin & Hovy, 2003). For zero-shot evaluation, we mainly use ten tasks widespread in the literature, including 1) commonsense reasoning: WinoGrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020); 2) question answering: boolQ (Clark et al., 2019), OpenbookQA (Mihaylov et al., 2018), ARC-easy and hard (Clark et al., 2018); 3) coding: (HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021); 4) general-purpose reasoning: BBH. (Suzgun et al., 2022)[2] We remove overlaps between the evaluation tasks and the Flan multi-task training set (boolQ, ARC, WinoGrande, HellaSwag, OpenbookQA and PIQA). We also include zero-shot results on the 12 held-out SNI tasks in the appendix.

Models and Training This work focuses on augmenting LLMs with a library of adapters to transform them into modular architectures. Our primary focus is on Phi-2 (Microsoft Research, 2023), a state-of-the-art model (as of March 2024) with 2.8 billion parameters, leading its class of models with parameter counts below 3 billion, according to the open leaderboard (Beeching et al., 2023). Additionally, we conducted experiments using the larger Mistral 7B (Jiang et al., 2023) model, given its widespread use in the community. For all models, we only patch attention layers with LoRA adapters. Unless stated otherwise, for all our multi-task training and single-task adaptation scenarios, we use LoRA rank of 4, dropout of 0.05 and learning rate of 1e-4. Unless specified, we set the number of clusters for `MBC` to 10, resulting in the best upstream validation loss and downstream performance for Phi-2, as demonstrated in Fig. 4.

Methods We consider the following methods in both zero-shot and supervised scenarios (except for `FullFT`):

- `Base`: the base model tested without any adaptation;
- `Shared`: a single expert LoRA finetuned on the joint training set of all tasks (256 tasks unless stated otherwise) on top of the base model with multi-task learning;
- `FullFT`: like `Shared` but the full model is finetuned.

We adopt the following naming convention for the models using a library of experts: <`library`>–<`routing`>. For the library type, we consider `Poly`, `MHR`, `Private` and `MBC` libraries described in Sec. 3. For `MBC`, we match the total amount of compute, meaning that we use 40% of the training steps to compute the LoRA clustering and the other 60% to compute the final cluster adapters. For routing, we use μ, `TP`, `CM` and `Arrow` in the zero-shot scenario and `Poly` and `LoraHub`[3] for the supervised scenario, described in Sec. 4.

5.1. Zero-Shot Results

In the zero-shot scenario, downstream tasks are evaluated without further fine-tuning. Tab. 1 presents the mean downstream accuracy for 10 held-out tasks. First, we analyze Phi-2 results. We observe that `MHR`-μ achieves strong zero-shot performance, competitive with `Shared` and `FullFT`, in line with the results of Caccia et al. (2023). Interestingly, training one adapter per task and then taking the average, `Private`-μ, still achieves gains w.r.t. `Base`, albeit falling short of multi-task training (`FullFT` and `Shared`), highlighting the competitiveness of uniform (μ) adapter

[2] We test on a subset of 1000 randomly sampled examples to reduce evaluation costs.

[3] For `LoraHub`, we match the amount of compute used by SGD. Assuming the backward pass is twice the compute of a forward pass, and since nevergrad (NG; Rapin & Teytaud, 2018) only does forward passes, to match the compute of 5 SGD training epochs, we perform 30 epochs of NG with 1/2 of the training data used by SGD methods.

	Library	Route	$\lvert\mathcal{L}\rvert$	PIQA	BOOLQ	WG	HSWAG	ARCE	ARCC	HE	OQA	BBH	MBPP	Acc.
Phi-2 (2.8B)	Base	-	-	79.2	82.7	75.7	72.5	77.5	52.9	45.1	49.8	48.0	56.0	63.8
	FullFT	-	-	80.3	80.8	77.0	73.2	83.5	57.9	50.0	48.0	47.7	57.2	65.6
	Shared	-	1	80.4	82.4	76.6	73.4	83.2	55.8	46.3	50.4	48.4	58.4	65.5
	Poly	μ	8	80.6	82.3	76.7	71.7	82.7	55.3	48.2	50.4	49.8	59.1	65.7
	MHR	μ	8	80.1	83.0	77.1	70.4	83.2	55.5	46.3	53.4	52.0	58.0	65.9
	Private	μ	256	79.5	83.2	76.0	73.1	81.4	54.5	43.9	47.8	48.5	59.9	64.8
	Private	↗	256	80.2	84.3	77.6	72.6	84.2	56.4	50.6	52.2	47.7	59.9	66.6
	MBC	μ	10	80.3	85.1	77.3	73.1	84.3	57.7	48.8	50.2	51.6	62.3	67.1
	MBC	↗	10	79.9	84.7	77.7	72.9	84.8	57.9	51.8	50.2	52.2	62.3	67.4
Mistral (7B)	Base	-	-	81.1	82.2	66.5	78.8	68.9	49.6	28.0	44.6	47.9	47.5	59.5
	Shared	-	1	50.4	84.6	68.6	79.5	84.8	60.0	24.4	50.4	49.2	47.5	63.1
	Private	μ	256	82.1	82.7	67.2	79.6	78.7	54.8	29.9	45.2	49.0	49.4	61.9
	Private	↗	256	82.8	86.6	66.6	81.1	85.7	60.8	30.5	50.6	49.5	49.4	64.4
	MBC	μ	10	83.0	87.6	68.5	80.8	86.2	60.9	28.7	48.6	51.5	50.2	64.6
	MBC	↗	10	82.8	87.3	70.6	80.9	84.5	59.6	28.0	52.8	45.5	47.1	63.9

Table 1. Downstream zero-shot results for Phi-2 and Mistral backbones. $\lvert\mathcal{L}\rvert$ denotes the library size. For comparison with other routing baselines, see Fig. 3.

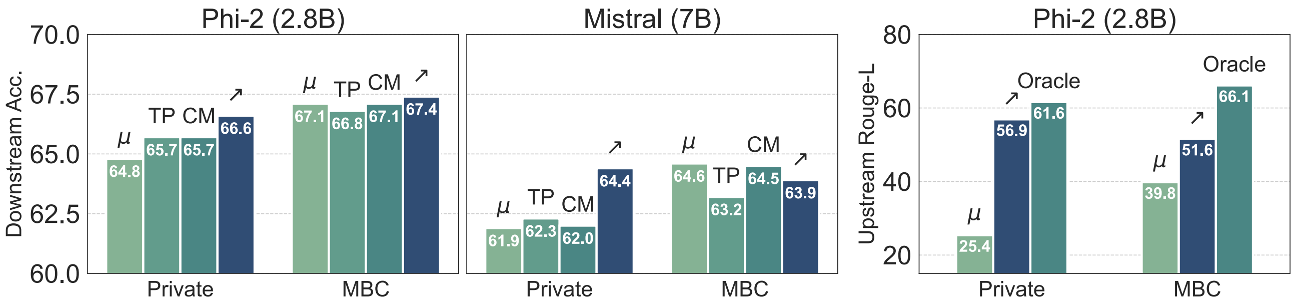

Figure 3. Comparison of routing approaches with both `Private` and `MBC` libraries. *Left & Middle.* Downstream zero-shot performance on two backbones; `Arrow` outperforms other routing approaches in the case of private libraries, while in the case of `MBC` libraries, routing is less important. *Right.* Upstream performance on the held-out sets of each of the 256 training tasks. `Arrow` nearly matches Oracle routing (which uses information about the task identity) in the case of `Private` library and noticeably improves for `MBC`.

routing (Chronopoulou et al., 2023a). Comparing the performance of our proposed `MBC` approach for library construction (`MBC`-μ) to previous approaches, we notice a sizable bump in performance of 1.2% absolute accuracy points over the strongest baseline (`MHR`). Similarly, when studying the zero-shot performance of Phi-2 on 12 SNI tasks in Tab. 5 we observe that `MBC`-μ strongly outperforms other baselines. Importantly, both `Shared` and `FullFT` methods, as well as `Poly` and `MHR` libraries assume simultaneous access to the full dataset of all tasks. In contrast, `Private` and `MBC` libraries can be trained in an embarrassingly parallel manner and therefore do not require any distributed training infrastructure (Li et al., 2022).

Next, we analyze whether more informed routing can improve performance beyond the $\mu-$routing. The full results are reported in Figure 3 (*Left & Middle*). We see that `TP`, `CM` and `Arrow` routing improve the performance over μ for the `Private` Phi-2 library, gaining 0.9%, 0.9% and 1.8% points respectively. This highlights the importance of routing for larger libraries. Notably, `Arrow` (66.6%) can surpass the performance of `FullFT` (65.6%) on the `Private` library.

On the `MBC` library, `TP` routing decreases performance when compared to uniform routing, while `MBC`-↗ improves over `MBC`-μ by 0.3% points and proves itself as a more robust routing method for both `Private` and `MBC` libraries. Overall, `MBC`-↗ improves 3.6 points over the base model and 1.8% absolute over `FullFT`.

For Mistral, we find a similar trend with `MBC` libraries achieving the best performance. `Arrow` routing results in a 2.5% increase in average performance over μ routing when used with the `Private` library (`Private`-↗ vs. `Private`-μ). `Arrow` is able to narrow the performance gap with `MBC`, without requiring simultaneous data access across tasks. We do not see any gains from using other routing methods for

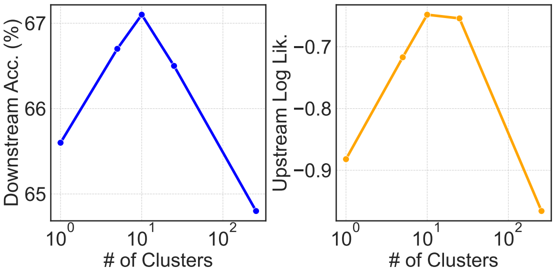

Figure 4. Phi-2 zero-shot accuracy on the 10 held-out tasks (*left*) and validation log-likelihood on the training tasks (*right*) as a function of the number of MBC clusters.

10 experts in the MBC library in this case. We make similar observations analyzing 0-shot SNI-12 results presented in Table 5, where Private-↗ attains notable gains of 10 Rouge-L points over Private-μ while MBC-μ strongly outperforms all other baselines.

MBC **Analysis** Overall, MBC enhances the performance of the library across all our results. To investigate this further, we compare different clustering techniques. First, we compare to clusters obtained by randomly selecting examples (RandomExamples). This is equivalent to randomly partitioning the joint multi-task dataset. Then, we compare to clusters obtained by randomly choosing tasks from the entire set of training tasks (RandomTask). Finally, we cluster task embeddings, which are obtained by forwarding task-specific examples through the model and averaging their representation at the model's penultimate layer (Embeddings). For all these methods, we set the number of clusters to 10.

The results are shown in Table 2. RandomTask surpasses RandomExamples by 1.6%, which indicates that grouping tasks rather than task examples is crucial for positive transfer. Embeddings underperforms MBC and supports our observation that the cosine similarity between the weights of privately-trained LoRA correlates better than using representation similarity for 0-shot generalization. Additionally, we also report average pairwise cluster "similarity" (as measured by the cosine similarity of the LoRA weights for each cluster) and observe a tendency that expert clusters with lower similarity, i.e. higher diversity, tend to result in higher performance. We conjecture that this stems from different clusters contributing distinct features to the joint model; however, we leave further investigation in this direction to future work (Jolicoeur-Martineau et al., 2023).

5.2. Upstream Performance

We further assess the efficacy of Arrow routing by looking at the *upstream* in-distribution performance, measured as the average of the Rouge-L on the validation sets of the 256 training tasks. Within this setting, we can compute the performance of the Oracle routing, which selects for each task the corresponding expert. In Fig. 3 (*Right*) we report the results for Arrow and μ routing with both MBC and Private libraries. For both libraries, ↗ increases performance w.r.t. μ and almost matches Oracle performance in the Private setting. This demonstrates Arrow's ability to correctly select the most relevant modules from a large library of experts.

Clustering	Mean Acc.	Similarity
RandExamples*-μ	64.8	0.82
RandTask*-μ	66.4	0.58
RandTask-μ	66.4	0.58
Embeddings*-μ	66.1	0.37
MBC*-μ	66.7	0.37
MBC-μ	67.1	0.27

Table 2. Ablation of task clustering: RandTask clusters *tasks* randomly, RandExamples clusters *examples* randomly, Embeddings clusters examples based on their embedding similarity. '*' denotes one epoch of training to save computation. We also report average cosine similarity between cluster adapters.

5.3. Supervised Adaptation

In Table 3, we present the supervised adaptation results for Phi-2 on the full (100% of training data) and limited (10% of training data) data regimes. The detailed per-task performance as well as the adaptation results for the Mistral model are presented in Table 8 and 7. First, for all models (Phi-2, Mistral) we observe a notable performance boost coming from using Private and MBC libraries compared to No Library, which optimizes a LoRA for each downstream task by starting from a random initialization, and Shared, which starts from the multi-task trained LoRA solution. Secondly, similarly to zero-shot results, we observe that MBC can boost the performance with both Poly and μ routing: for Phi-2 the performance of MBC-μ tops Private-μ. Additionally, we see that randomly grouping tasks RandomTask-Poly outperforms the non-library baselines but does not quite match MBC-based clustering for all the models. The low performance of LoraHub can be attributed to the fact that LoraHub does not fine-tune the LoRA experts' weights but only their routing coefficients (due to gradient-free optimization). Refer to App. 8.2 for more insights onto this point. Finally, MBC-μ performs similarly to MBC-Poly, echoing results in (Caccia et al., 2023).

5.4. Summary of Results

Mirroring the questions at the start of this section, we list our main takeaway messages below:

(i) When appropriately routed, independently trained experts (Private-↗) can match and surpass the zero-shot performance of full fine-tuning (for Phi-2) and shared tuning (for Mistral 7B). This is a rather surprising result given that experts are independently trained and routing is learned *post-hoc*. These results

Method	100% Data	10% Data
Base	22.2	22.2
No Library	75.5	53.9
Shared	75.8	56.4
Poly	73.4	61.7
MHR	74.8	64.5
π-tuning	76.7	64.6
Private-μ	76.9	62.5
RandTask-Poly	76.7	67.6
MBC-μ	78.8	67.0
MBC-Poly	78.8	68.2

Table 3. Supervised adaptation results on 12 SNI held-out tasks for Phi-2 obtained in the full (100% of training data) and limited (10% of the training data) data settings.

show promise for building collaboratively and asynchronously trained LMs.

(ii) If data sharing is possible, then clustering tasks by their similarity with MBC constitutes a very effective strategy. In this case, simply averaging the LoRA adapters obtained through MBC (MBC-μ) is sufficient compared to more sophisticated routing. Our zero-shot and supervised adaptation results underscore the importance of task-based clustering.

(iii) Arrow appears to be a very performant zero-shot routing strategy while requiring minimal information about the trained LoRAs and none about the training data. For supervised adaptation, training both adapters and the routing coefficients appears to be crucial. Overall, if routing seems beneficial for large libraries of adapters, the gains for smaller libraries are diminishing. This appears to stand in contrast with sparse MoE models, where (non-uniform) routing is crucial (Jiang et al., 2024). This may be due to the linearity of LoRA experts, which stands in contrast with MLP experts in sparse MoEs (Fedus et al., 2022); we leave this investigation for future work.

Our main finding is that adapter parameters are suitable both to inform task clustering, and thus guide library building, and to route new inputs, thus facilitating library reuse.

6. Related Work

Multi-task learning involves training on a joint set of all tasks (Caruana, 1997), potentially leading to performance degradation due to task interference (Zhao et al., 2018). An extensive literature studies how to partition learnable parameters into shared and task-specific ones (Ding et al., 2023; Strezoski et al., 2019; Bragman et al., 2019; Zaremoodi et al., 2018; Wallingford et al., 2022; Fifty et al., 2021). We operate in the parameter-efficient multi-task learning setting (Ponti et al., 2023; Vu et al., 2021; Chronopoulou et al., 2023a; Pfeiffer et al., 2021). Vu et al. (2021) train one prefix adapter (Li & Eisner, 2019) per task and learn to re-use them for other tasks based on the adapter similarities. MBC can be seen as an extension of this approach where we cluster tasks based on their weight similarity to ensure more transfer during multi-task pre-training.

Mixture of experts (MoEs), when coupled with sparse routing, are notable for augmenting model capacity with minimal computational overhead (Fedus et al., 2022) . Among the most important differences in this work: i) adapter experts are not trained during base model pre-training, ii) they are parameter-efficient and iii) they are tailored to specific tasks instead of being opaque computation units at the token level whose specialization is not easily interpretable (Jiang et al., 2024). Regarding ii), Wang et al. (2022a); Zadouri et al. (2023); Muqeeth et al. (2023) employs routing each example to a set of experts, showcasing enhanced performance on unseen tasks. Gupta et al. (2022) trains a separate router for each task and picks a router from a similar task based on domain knowledge. Ye et al. (2022) proposes task-level MoEs that treat a collection of transformer layers as experts and a router chooses from these experts dynamically. Recent work by Caccia et al. (2023); Ponti et al. (2023); Ostapenko et al. (2023) investigate the effectiveness of densely routed adapter experts trained end-to-end with an expert library for MTL fine-tuning. For expert aggregation, we employ parameter-space weighted averaging (Wortsman et al., 2022; Zhang et al., 2023a; Ramé et al., 2023) with weights induced by a learned router, a technique akin to those in previous works (Ostapenko et al., 2023; Zadouri et al., 2023). Several recent works have also proposed techniques for learning how to route queries to specialized pretrained open-source LLMs (Lu et al., 2023; Shnitzer et al., 2023).

Model ensembling techniques aim to enhance model robustness and generalization by integrating multiple distinct models (Frankle et al., 2020; Wortsman et al., 2022; Ramé et al., 2023; Jin et al., 2022; Matena & Raffel, 2022; Chronopoulou et al., 2023b; Yang et al., 2023). Parameter space averaging of independent models serves as an efficient ensembling method for full models (Ilharco et al., 2022; Ainsworth et al., 2022; Jin et al., 2022) and adapters (Zhang et al., 2023a; Yadav et al., 2024), requiring only a single forward pass through the model, unlike output space ensembling (Dietterich, 2000; Breiman, 1996), that requires many forward passes. Efficient output ensembling techniques that can be applied in conjunction with our work are in (Wen et al., 2020). Similarly, Pfeiffer et al. (2021) proposes ensembling bottleneck style adapters with the subsequent finetuning step. Tam et al. (2023) presents a merging framework called MaTs using the conjugate gradient method. Yadav et al. (2024) proposes Ties-Merging to mitigate interference due to redundant parameter values. Daheim et al. (2024) merge models by reducing their individual gradient mismatch with an ideal joint model, weighting their parameters with normalized Fisher Information.

Data Clustering for LMs have been proposed to improve performance and decrease task interference (Fifty et al., 2021; Gururangan et al., 2023; Gou et al., 2023). These methods include clustering using similarities computed by tf-idf and neural embeddings, K-means clustering with balanced linear assignment, and soft clustering with GMMs (Gross et al., 2017; Chronopoulou et al., 2023a; 2021; Gururangan et al., 2023; Duan et al., 2021; Caron et al., 2018). Recent work by Zhou et al. (2022) observes the potential of adapter parameters as effective task embeddings for clustering purposes, a concept we leverage in this work. A similar observation, but regarding task gradients, has been made by Vu et al. (2020b).

Building libraries of composable experts has been envisioned in several previous works (Pfeiffer et al., 2021; Wu et al., 2023; Huang et al., 2023; Shah et al., 2023; Xun Wu, 2024). Beck et al. (2021); Poth et al. (2023) orchestrated a framework for assembling diverse adapters, offering flexibility in both training and inference. Most related to this work, Huang et al. (2023) build LoRAHub, a library of task-specific LoRAs that can be combined for few-shot generalization. Pfeiffer et al. (2021) introduce a two-stage learning algorithm that leverages knowledge from multiple tasks. They first learn task-specific experts and then combine the experts in a separate knowledge composition step. Xun Wu (2024) introduces a learnable gating function to combine multiple LoRAs, called Mixture of LoRA Experts (MoLE). Wu et al. (2023) presents π-tuning for vision, language, and vision-language few-shot tasks. π-tuning trains task-specific experts and then uses task embedding based on the diagonal of the Fisher information matrix to retrieve the top-k most similar tasks to a target task. We extend and complement these works by *i)* proposing novel methods to build a library, and *ii)* proposing techniques for zero-shot post-hoc routing independently trained adapters. Related to *ii)*, in a concurrent work, Muqeeth et al. (2024) learns a sigmoid gate for each expert, which is later used as the expert prototype for zero-shot transfer. Notably, this method is applicable to the same setting as `Arrow`, and generalizes beyond linear adapters. However, in contrast to `Arrow`, obtaining expert prototypes requires additional training after the experts are learned.

Routing through expert LLMs has become an increasingly popular research topic with the emergence of a plethora of LLMs with their own expertise domains such as mathematics (Shao et al., 2024), code (Roziere et al., 2023), medical (Tu et al., 2023) etc.. Integrating LLMs with different architectures requires merging in the output space of the models. To this end similarly to the `TP` routing featured in this work, (Shnitzer et al., 2023; Chai et al., 2024; Lu et al., 2023) learn a router on a separate dataset that can predict how relevant are different expert LLMs to the current input sample. On a similar note, several recent works explored collaborative decoding ideas for LLMs, where routing is performed at a token level in the model output space (Shen et al., 2024; Leviathan et al., 2023).

7. Conclusions and Future Work

We investigate how to build and reuse a library of adapters “end-to-end”. We show the potential of reusing independently (or partially independently) trained adapters with a zero-shot routing strategy. Overall, we strategically investigate the modular augmentation of LLMs, offering a promising direction for research that prioritizes efficiency, flexibility, and performance.

The current investigation focuses on LoRA adapters. For future work, we are excited by the exploration of a heterogeneous “universe” of adapters—including soft and hard prompts (Lester et al., 2021; Wen et al., 2023), MLPs (Houlsby et al., 2019), etc.—and combinations thereof. Whether our approach can result in encouraging results at a greater scale (both in terms of data and model size) remains open to further investigation. Using the proposed routing strategy for modular continual learning (Ostapenko et al., 2021; Ermis et al., 2022; Wang et al., 2022c) is another promising direction for future work, especially given the fact that the `Arrow` router is local to each expert. In principle, it may be less susceptible to catastrophic forgetting as no gradient-based training is required to incorporate new experts into the library.

Impact Statement

This work sheds light on different ways of extending the capabilities of language models by surrounding them with a universe of lightweight adapters that can be trained on conventional hardware. Allowing the reuse of adapters might enable systems that are trained in a collaborative and distributed fashion and that use less total energy, with positive ramifications for the environment, but still attain the performance of vanilla systems. Further, this might allow users with smaller computational resources to more easily use and customize LLMs. There are also many potential societal consequences of improving LLMs, some being less desirable and even undesirable, but none of which we feel must be specifically highlighted here.

Acknowledgement

We acknowledge the support of Matheus Pereira for maintaining and optimizing the code, as well as for preparing the code release.

References

Ainsworth, S. K., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Beck, T., Bohlender, B., Viehmann, C., Hane, V., Adamson, Y., Khuri, J., Brossmann, J., Pfeiffer, J., and Gurevych, I. Adapterhub playground: Simple and flexible few-shot learning with adapters. *arXiv preprint arXiv:2108.08103*, 2021.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

Belofsky, J. Token-level adaptation of lora adapters for downstream task generalization, 2023.

Bisk, Y., Zellers, R., Gao, J., Choi, Y., et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 7432–7439, 2020.

Bragman, F. J., Tanno, R., Ourselin, S., Alexander, D. C., and Cardoso, J. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1385–1394, 2019.

Breiman, L. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

Caccia, L., Ponti, E., Su, Z., Pereira, M., Roux, N. L., and Sordoni, A. Multi-head adapter routing for cross-task generalization, 2023.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Caruana, R. Multitask learning. *Machine learning*, 28: 41–75, 1997.

Chai, Z., Wang, G., Su, J., Zhang, T., Huang, X., Wang, X., Xu, J., Yuan, J., Yang, H., Wu, F., et al. An expert is worth one token: Synergizing multiple expert llms as generalist via expert token routing. *arXiv preprint arXiv:2403.16854*, 2024.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, Z., Shen, Y., Ding, M., Chen, Z., Zhao, H., Learned-Miller, E., and Gan, C. Mod-squad: Designing mixture of experts as modular multi-task learners, 2022.

Chronopoulou, A., Peters, M. E., and Dodge, J. Efficient hierarchical domain adaptation for pretrained language models. *arXiv preprint arXiv:2112.08786*, 2021.

Chronopoulou, A., Peters, M. E., Fraser, A., and Dodge, J. Adaptersoup: Weight averaging to improve generalization of pretrained language models. *arXiv preprint arXiv:2302.07027*, 2023a.

Chronopoulou, A., Pfeiffer, J., Maynez, J., Wang, X., Ruder, S., and Agrawal, P. Language and task arithmetic with parameter-efficient layers for zero-shot summarization. *arXiv preprint arXiv:2311.09344*, 2023b.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Daheim, N., Möllenhoff, T., Ponti, E., Gurevych, I., and Khan, M. E. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=D7KJmfEDQP.

Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Ding, C., Lu, Z., Wang, S., Cheng, R., and Boddeti, V. N. Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7756–7765, 2023.

Duan, Z., Zhang, H., Wang, C., Wang, Z., Chen, B., and Zhou, M. Enslm: Ensemble language model for data diversity by semantic clustering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2954–2967, 2021.

EleutherAI. Multiple-choice normalization. https://blog.eleuther.ai/

multiple-choice-normalization/, 2021. Accessed: 2024-05-12.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

Ermis, B., Zappella, G., Wistuba, M., Rawal, A., and Archambeau, C. Memory efficient continual learning with transformers. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=U07d1Y-x2E.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., and Finn, C. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Gou, Y., Liu, Z., Chen, K., Hong, L., Xu, H., Li, A., Yeung, D.-Y., Kwok, J. T., and Zhang, Y. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379*, 2023.

Gross, S., Ranzato, M., and Szlam, A. Hard mixtures of experts for large scale weakly supervised vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6865–6873, 2017.

Gupta, S., Mukherjee, S., Subudhi, K., Gonzalez, E., Jose, D., Awadallah, A. H., and Gao, J. Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*, 2022.

Gururangan, S., Li, M., Lewis, M., Shi, W., Althoff, T., Smith, N. A., and Zettlemoyer, L. Scaling expert language models with unsupervised domain discovery. *arXiv preprint arXiv:2303.14177*, 2023.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799, 2019. URL http://proceedings.mlr.press/v97/houlsby19a/houlsby19a.pdf.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. Lorahub: Efficient cross-task generalization via dynamic lora composition, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Jang, J., Kim, S., Ye, S., Kim, D., Logeswaran, L., Lee, M., Lee, K., and Seo, M. Exploring the benefits of training expert language models over instruction tuning, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024.

Jin, X., Ren, X., Preotiuc-Pietro, D., and Cheng, P. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Jolicoeur-Martineau, A., Gervais, E., Fatras, K., Zhang, Y., and Lacoste-Julien, S. Population parameter averaging (papa), 2023.

Karimi Mahabadi, R., Ruder, S., Dehghani, M., and Henderson, J. Parameter-efficient multi-task fine-tuning for Transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 565–576, August 2021. URL https://aclanthology.org/2021.acl-long.47.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/pdf/2104.08691.pdf.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Li, M., Gururangan, S., Dettmers, T., Lewis, M., Althoff, T., Smith, N. A., and Zettlemoyer, L. Branch-train-merge: Embarrassingly parallel training of expert language models. *arXiv preprint arXiv:2208.03306*, 2022.

Li, X. L. and Eisner, J. Specializing word embeddings (for parsing) by information bottleneck. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2744–2754, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1276. URL https://www.aclweb.org/anthology/D19-1276.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Lin, C.-Y. and Hovy, E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, pp. 150–157, 2003.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022. URL https://arxiv.org/abs/2205.05638.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

Lu, K., Yuan, H., Lin, R., Lin, J., Yuan, Z., Zhou, C., and Zhou, J. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*, 2023.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Matena, M. S. and Raffel, C. A. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Microsoft Research. Phi-2: The Surprising Power of Small Language Models, 2023.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Mireshghallah, F., Taram, M., Vepakomma, P., Singh, A., Raskar, R., and Esmaeilzadeh, H. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254*, 2020.

Muqeeth, M., Liu, H., and Raffel, C. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745*, 2023.

Muqeeth, M., Liu, H., Liu, Y., and Raffel, C. Learning to route among specialized experts for zero-shot generalization. *arXiv preprint arXiv: 2402.05859*, 2024.

Nakatsukasa, Y. The low-rank eigenvalue problem. *arXiv preprint arXiv:1905.11490*, 2019.

Ostapenko, O., Rodriguez, P., Caccia, M., and Charlin, L. Continual learning via local module composition. *Advances in Neural Information Processing Systems*, 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/fe5e7cb609bdbe6d62449d61849c38b0-Paper.pdf.

Ostapenko, O., Caccia, L., Su, Z., Le Roux, N., Charlin, L., and Sordoni, A. A case study of instruction tuning with mixture of parameter-efficient experts. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., and Gurevych, I. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 487–503, April 2021. URL https://aclanthology.org/2021.eacl-main.39.

Pfeiffer, J., Ruder, S., Vulić, I., and Ponti, E. M. Modular deep learning. *arXiv preprint arXiv:2302.11529*, 2023. URL https://arxiv.org/pdf/2302.11529.pdf.

Ponti, E. M., Sordoni, A., Bengio, Y., and Reddy, S. Combining parameter-efficient modules for task-level generalisation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 687–702, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.eacl-main.49.

Poth, C., Sterz, H., Paul, I., Purkayastha, S., Engländer, L., Imhof, T., Vulić, I., Ruder, S., Gurevych, I., and Pfeiffer, J. Adapters: A unified library for parameter-efficient and modular transfer learning. *arXiv preprint arXiv:2311.11077*, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21: 1–67, 2020. URL https://www.jmlr.org/papers/volume21/20-074/20-074.pdf.

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 28656–28679. PMLR, 2023.

Rapin, J. and Teytaud, O. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad, 2018.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., and Jampani, V. Ziplora: Any subject in any style by effectively merging loras. *arXiv preprint arXiv:2311.13600*, 2023.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shen, S. Z., Lang, H., Wang, B., Kim, Y., and Sontag, D. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*, 2024.

Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., and Yurochkin, M. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Strezoski, G., Noord, N. v., and Worring, M. Many task learning with task routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1375–1384, 2019.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Tam, D., Bansal, M., and Raffel, C. Merging by matching models in task subspaces. *arXiv preprint arXiv:2312.04339*, 2023.

Tu, T., Azizi, S., Driess, D., Schaekermann, M., Amin, M., Chang, P., Carroll, A., Lau, C., Tanno, R., Ktena, I., et al. Towards generalist biomedical ai. arxiv. *Preprint posted online*, 26, 2023.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7882–7926, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.635. URL https://aclanthology.org/2020.emnlp-main.635.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. Exploring and predicting transferability across nlp tasks. *arXiv preprint arXiv:2005.00770*, 2020b.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., and Cer, D. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*, 2021.

Wallingford, M., Li, H., Achille, A., Ravichandran, A., Fowlkes, C., Bhotika, R., and Soatto, S. Task adaptive parameter sharing for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7561–7570, 2022.

Wang, Y., Agarwal, S., Mukherjee, S., Liu, X., Gao, J., Awadallah, A. H., and Gao, J. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *arXiv preprint arXiv:2205.12410*, 2022a.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Naik, A., Stap, D., et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 2022b.

Wang, Z., Tsvetkov, Y., Firat, O., and Cao, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=F1vEjWK-lH_.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022c.

Wen, Y., Tran, D., and Ba, J. Batchensemble: An alternative approach to efficient ensemble and lifelong learning, 2020.

Wen, Y., Jain, N., Kirchenbauer, J., Goldblum, M., Geiping, J., and Goldstein, T. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*, 2023.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Wu, C., Wang, T., Ge, Y., Lu, Z., Zhou, R., Shan, Y., and Luo, P. π-tuning: Transferring multimodal foundation models with optimal multi-task interpolation. In *International Conference on Machine Learning*, pp. 37713–37727. PMLR, 2023.

Xun Wu, Shaohan Huang, F. W. Mole: Mixture of lora experts. In *International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024.

Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., and Tao, D. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

Ye, Q., Zha, J., and Ren, X. Eliciting and understanding cross-task skills with task-level mixture-of-experts. *arXiv preprint arXiv:2205.12701*, 2022.

Zadouri, T., Üstün, A., Ahmadian, A., Ermiş, B., Locatelli, A., and Hooker, S. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.

Zaremoodi, P., Buntine, W., and Haffari, G. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 656–661, 2018.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Zhang, J., Chen, S., Liu, J., and He, J. Composing parameter-efficient modules with arithmetic operations. *arXiv preprint arXiv:2306.14870*, 2023a.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023b.

Zhao, X., Li, H., Shen, X., Liang, X., and Wu, Y. A modulation module for multi-task learning with applications in image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 401–416, 2018.

Zhou, W., Xu, C., and McAuley, J. Efficiently tuned parameters are task embeddings. *arXiv preprint arXiv:2210.11705*, 2022.

	Library		L	piqa	boolq	wgrande	hswag	arcE	arcC	HE	oqa	bbh	mbpp	Acc.
StableLM (3B)	Base	-	-	78.2	73.1	66.6	73.7	59.6	41.5	18.3	37.6	34.7	32.3	51.6
	Shared	-	1	79.4	80.3	68.0	71.3	74.7	42.1	11.6	38.0	38.3	21.0	52.5
	Private	μ	100	79.4	76.8	67.3	74.4	72.4	44.0	16.5	42.6	37.0	34.6	54.5
	Private	$\nearrow$	100	80.1	72.1	70.8	74.8	73.4	45.3	16.5	43.6	36.1	33.5	54.6
	MBC	μ	10	80.4	80.4	68.2	74.7	76.7	47.4	14.6	43.0	35.4	36.2	55.7
	MBC	$\nearrow$	10	80.5	79.0	68.2	73.6	75.2	46.4	13.4	43.0	32.0	27.6	53.9

Table 4. **Out-of-distribution zero-shot results**: Accuracy on held-out tasks for StableLM. The best results are underlined.

8. Appendix

8.1. Analyzing $\|AB^T v\|_2$ for in-distribution and out-of-distribution samples

In this section, we analyze whether the motivation behind `Arrow` routing holds in practice. Recall that at each layer, `Arrow` routing initializes prototypes in the linear router for expert i with the unit vector v_i maximizing $\|AB^T v\|_2$. Concretely, we hypothesize that for a hidden activation h computed from $x \in \mathcal{D}_i$, we have $\|A_i B_i^T v\|_2 > \|A_j B_j^T v\|_2$, for experts i, j. In other words, the norm of the linearly transformed prototype will be higher under the expert belonging to the same task as the input h.

To test this hypothesis, we run the following experiment. Let h_l denote the input to the expert at layer l, and $(AB^T)_l^i$ denote the linear transformation of expert i at layer l. We first sample 5000 examples from the multitask dataset. Then, for a given input $x \in \mathcal{D}_i$ at each layer l, we compute both $\|(AB^T)_l^i \cdot h_l\|_2$ and $\|(AB^T)_l^j \cdot h_l\|_2$ where j is another randomly sampled expert such that $i \neq j$. We then compute the average norm ratio r across all layers, i.e.

$$r = \sum_l^L \frac{1}{L} \frac{\|(AB^T)_l^i \cdot h_l^i\|_2}{\|(AB^T)_l^j \cdot h_l^i\|_2}.$$

Note that the random expert j is sampled at every layer, and the output of the in-distribution expert is propagated to the next layer. As such, $r > 1$ indicates that on average, the in-distribution expert produces a higher norm output, which would validate the use of the norm-maximizing initialization that `Arrow` routing uses. In figure 5, we see that for all the points considered, this ratio is positive, indicating that in-distribution experts tend to maximize the norm of the linearly transformed input.

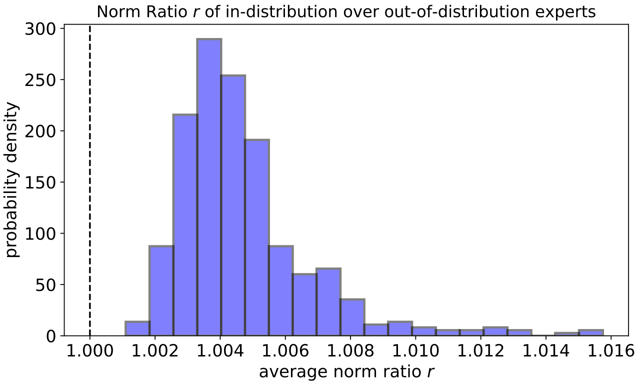

Figure 5. Histogram of the ratios r computed over 5000 samples.

8.2. Few-shot adaptation

We apply some of the proposed methods to a data scarce setting with up to only 0.5% of the original training data per task (approx. 40 examples per task). We show the results in Table 6. Even in this setting gradient-based method `MBC-Poly` considerably outperforms `LoraHub`, where the `LoraHub` is given compute equivalent to training gradient-based methods on

	Method	L	SNI Tasks												Rouge-L
			202	304	614	613	362	242	1728	1557	035	1356	039	1153	
Phi-2 (2.8B)	Base	-	4.0	3.3	26.4	3.5	16.2	32.5	35.2	62.5	54.2	12.8	8.2	7.6	22.2
	Shared	1	38.3	17.9	36.4	11.5	77.2	39.4	45.8	84.5	40.7	21.5	34.3	24.1	39.3
	Private-μ	256	10.6	16.1	35.6	9.6	64.8	58.2	42.6	72.2	61.7	17.5	25.1	24.0	36.6
	Private-↗	256	20.4	18.8	31.8	10.5	76.3	36.4	46.8	84.2	41.8	19.2	33.4	28.7	37.4
	MBC-μ	10	31.8	26.9	33.9	12.7	77.6	77.9	47.2	86.0	49.2	22.4	37.0	29.8	44.4
	MBC-↗	10	32.6	15.9	31.3	7.6	79.6	36.6	41.7	80.2	33.1	21.5	32.0	28.5	36.7
Mistral (7B)	Base	-	13.7	10.7	31.8	5.6	37.4	22.0	35.6	49.1	58.6	13.7	22.2	14.2	26.4
	Shared	1	50.8	18.0	37.5	8.8	67.4	80.0	54.1	81.9	59.6	30.0	32.0	27.0	45.6
	Private-μ	256	30.1	17.2	10.2	7.7	70.4	37.7	38.6	63.0	63.0	20.7	25.4	23.2	36.4
	Private-↗	256	38.5	23.7	43.7	12.7	78.0	76.7	54.6	83.3	57.9	25.2	35.4	33.4	46.9
	MBC-μ	10	54.0	26.1	46.4	15.0	80.8	80.1	46.0	82.7	66.5	28.1	46.1	36.0	50.6
	MBC-↗	10	38.1	24.3	35.9	12.8	85.5	77.1	43.3	82.1	57.7	29.0	33.9	31.2	45.9

Table 5. **Out-of-distribution zero-shot results on 12 held-out SNI tasks** for library built for the Phi-2 and Mistral base models. Applying ↗ routing to the Private libraries results in performance improvements over the μ routing for both models, with a notable improvement of over 10 Rouge-L points in case of Mistral. It is worth noticing that μ routing performed better than ↗ in case of MBC library for both models. We note that ↗ only selects top-4 experts for routing, whereas μ averages full libraries. The best results are underlined.

the full dataset. Additionally, we observe that MBC-PolyZ, a method similar to MBC-Poly that only updates the routings and not the expert's weights, performs similarly to LoraHub. Interestingly, when data amount is lowered, the performance of MBC-PolyZ is reduced by a relatively smaller margin than MBC-Poly which can be explained by a smaller amount of updated parameters.

Method	L	SNI Tasks												Rouge-L
		202	304	614	613	362	242	1728	1557	035	1356	039	1153	
Full data														
MBC-LoraHub	10	41.5	21.9	37.4	17.5	78.1	68.3	48.0	82.0	62.6	21.2	33.5	31.1	45.3
MBC-Poly	10	96.9	84.4	67.2	53.9	96.4	97.8	60.2	87.9	91.3	29.4	81.7	99.7	78.9
MBC-PolyZ	10	37.7	27.9	36.2	12.6	75.9	74.4	48.7	81.3	58.9	22.5	36.1	31.1	45.3
10%														
MBC-Poly	10	89.6	53.3	64.5	44.5	93.5	98.5	58.5	75.7	87.3	27.2	65.6	66.8	68.8
MBC-PolyZ	10	34.3	27.5	36.2	12.4	76.5	74.3	47.5	86.2	57.9	22.7	35.3	31.4	45.2
5%														
MBC-Poly	10	87.0	43.0	61.3	41.7	92.0	95.2	55.3	77.3	89.0	25.4	59.1	47.8	64.5
MBC-PolyZ	10	32.6	28.6	36.0	13.0	76.4	73.9	47.3	86.3	57.7	22.7	36.6	31.0	45.2
0.5%														
MBC-Poly	10	49.7	30.4	43.7	20.3	77.3	78.4	48.2	86.5	72.2	23.2	43.0	29.1	50.2
MBC-PolyZ	10	32.0	27.4	34.3	12.6	77.6	78.1	47.2	86.5	53.2	22.2	37.0	29.1	44.8

Table 6. Rouge-L score for Phi-2 model after adaptation with different portions of data per task ranging from full dataset down to 10% and 5% of data per task. Note, MBC-PolyZ only tunes the routing weights, whereas MBC-Poly trains both the routing weights and the expert parameters.

9. Implementation details and hyperparameters

We provide some technical details about the experiments conducted in this paper.

Training hyperparameters. For all LoRA experts trained in this paper we employ LoRA rank of 4, LoRA dropout probability of 0.05, LoRA α of 16, and a learning rate of 1e-4 with a learning rate warm-up and annealing phases. We experimented with only patching fully connected layers (FC), only attention layers + attention output projection (ATT+O) or both (BOTH). For the preliminary experiments in Figure 2 we modify only the MLP (FC) layers of the transformer (.*fc[12].*). We found that patching FC layers severely underperform ATT+O layers. Patching BOTH gives marginal gains over ATT+O while significantly increasing computation cost and memory usage due to the wide projection (4 * hidden size) of the first FC layer in the transformer residual block. Therefore, for the rest of the experiments, we modified attention

	Method	*L*	SNI Tasks												Rouge-L
			202	304	614	613	362	242	1728	1557	035	1356	039	1153	
Phi-2 (2.8B)	No Library	-	93.2	74.2	64.9	51.4	95.9	96.2	59.3	81.4	90.5	26.9	73	99.1	75.5
	Shared	1	93.1	73.4	65.0	48.9	96.0	95.9	58.4	86.8	91.2	29.0	73.4	98.4	75.8
	MHR	1	94.5	66.9	63.0	47.8	94.9	95.6	59.5	86.6	91.0	27.9	70.7	98.2	74.8
	Poly	10	92.1	66.4	63.0	45.9	94.9	96.4	56.0	85.3	90.2	27.6	70.0	93.2	73.4
	Private-μ	256	93.6	78.1	65.0	50.7	94.8	97.8	59.7	87.9	90.7	28.1	76.4	99.7	76.9
	MBC-μ	10	96.4	83.2	67.6	53.5	96.2	98.0	60.5	88.2	90.7	29.8	82.3	99.5	78.8
	MBC-LoraHub	10	41.5	21.9	37.4	17.5	78.1	68.3	48.0	82.0	62.6	21.2	33.5	31.1	45.3
	RandTask-Poly	10	96.4	77.1	66.5	48.6	96.7	98.9	59.9	85.1	90.7	28.6	73.9	97.5	76.7
	MBC-Poly	10	96.9	84.4	67.2	53.9	96.4	97.8	60.2	87.9	91.3	29.4	81.7	99.7	78.9
Mistral (7B)	No Library	-	97.6	88.3	68.9	59.9	98.8	98.8	62.9	87.3	91.8	37.5	80.5	100	81.0
	Shared	1	95.8	87.4	69.9	52.7	98.7	99.2	63.5	87.6	91.6	37.2	78.1	100	80.1
	Private-μ	256	98.5	87.2	70.6	54.1	98.3	99.1	64.0	89.1	92.0	37.3	81.0	100	80.9
	MBC-μ	10	98.1	84.8	70.1	54.2	98.7	95.7	62.8	82.9	92.0	38.2	82.1	99.5	79.9
	MBC-LoRAHub	10	47.8	23.1	45.9	14.0	81.4	79.6	49.7	84.9	69.6	28.6	42.2	34.5	50.1
	RandTask-Poly	10	98.4	86.9	69.3	53.8	96.1	93.0	64.7	84.5	92.1	39.3	80.1	99.5	79.8
	MBC-Poly	10	98.7	88.7	69.2	56.1	97.4	99.5	64.1	82.2	92.2	38.7	81.1	99.0	80.6

Table 7. **Supervised adaptation results (100% training data per task)**: Rouge-L on 12 held-out SNI for Phi-2 and Mistral 7B models for different libraries. LoraHub follows the original implementation and optimizes the weighting coefficients for the adapters in the library with a non-gradient-based optimizer. The best results are underlined.

	Method	*L*	SNI Tasks (10%)												Rouge-L
			202	304	614	613	362	242	1728	1557	035	1356	039	1153	
Phi-2 (2.8B)	No Library	-	71.5	36.1	53.6	36.9	80.0	85.5	46.7	62.3	84.0	21.7	41.3	27.2	53.9
	Shared	1	76.8	35.5	55.5	39.4	82.6	89.3	47.6	61.8	86.0	23.3	47.9	31.2	56.4
	MHR	1	83.6	45.1	58.2	40.0	91.3	94.3	54.0	84.1	85.7	25.7	58.0	54.2	64.5
	Poly	1	74.4	38.3	57.8	39.5	82.5	92.2	50.8	85.1	85.1	25.5	54.8	54.1	61.7
	Private-μ	256	81.7	41.2	60.6	40.4	89.8	96.0	49.6	75.1	87.1	23.8	57.2	47.9	62.5
	MBC-μ	10	86.2	52.3	64.3	43.8	93.8	97.3	53.3	75.0	87.5	26.3	61.1	63.8	67.0
	MBC-LoraHub	10	43.6	22.2	36.5	13.5	77.0	68.8	45.5	82.2	63.2	21.2	34.6	27.6	44.7
	RandTask-Poly	10	87.9	51.0	63.5	41.4	94.1	95.8	55.6	79.6	89.0	27.1	61.1	65.3	67.6
	MBC-Poly	10	88.9	52.0	64.4	45.6	94.3	96.9	56.7	75.2	87.5	27.1	64.4	66.0	68.2
Mistral (7B)	No Library	-	91.7	66.8	66.2	47.8	95.2	98.3	59.5	69.9	90.7	33.9	65.8	68.1	71.2
	Shared	1	94.6	64.9	65.1	45.3	90.7	91.0	60.3	82.7	89.6	33.4	66.3	91.3	72.9
	Private-μ	256	89.0	64.3	66.0	47.2	94.7	98.8	59.4	84.1	90.6	33.5	67.6	92.9	74.0
	MBC-μ	10	94.2	62.6	66.3	47.9	95.9	96.6	59.9	83.6	90.7	33.7	69.3	92.8	74.5
	MBC-LoRAHub	10	47.7	23.2	47.0	15.4	85.8	72.4	49.0	81.9	71.7	27.3	27.7	30.6	48.3
	RandTask-Poly	10	95.0	64.8	66.0	48.8	94.3	92.1	59.6	87.1	90.6	34.4	66.4	92.4	74.3
	MBC-Poly	10	95.1	66.3	66.0	48.3	96.4	98.4	60.2	84.9	90.5	33.7	67.8	92.7	75.0

Table 8. **Supervised few-shot adaptation results (10% training data per task)**: Rouge-L on 12 held-out SNI for Phi-2 and Mistral 7B models for different libraries. LoraHub follows the original implementation and optimizes the weighting coefficients for the adapters in the library with a non-gradient-based optimizer. The best results are underlined.

layers + attention output projection (e.g. .*Wqkv.* |.*out_proj.* for Phi-2).

Downstream zero-shot results. All library-bases downstream zero-shot results are reported using top-4 routing with temperature 1. Unless stated otherwise, for all MBC libraries we use 10 experts. Additionally, in our implementation of the downstream evaluation, we append an EOS token to the target options to mark the end of a sentence. We use token-length normalized scores for selecting continuations for multiple-choice tasks evaluation (EleutherAI, 2021).

Adaptation experiments. For the adaptation experiments we also use the learning rate of 1e-4, with the same learning rate schedule as stated above. For both MHR and Poly adaptation, we tune both the experts and the routing weights.

c0	”ropes_background_new_situation_answer”, ”ropes_prompt_bottom_no_hint”, ”ropes_plain_background_situation”, ”ropes_new_situation_background_answer”, ”ropes_given_background_situation”, ”ropes_prompt_bottom_hint_beginning”, ”ropes_prompt_beginning”, ”ropes_read_background_situation”, ”ropes_plain_bottom_hint”, ”ropes_plain_no_background”, ”ropes_prompt_mix”, ”ropes_background_situation_middle”
c1	”glue_sst2_2_0_0”, ”adversarial_qa_droberta_generate_question”, ”true_case”, ”stream_qed”, ”huggingface_xsum”, ”cot_esnli”, ”cot_gsm8k”, ”trec_1_0_0”, ”yelp_polarity_reviews_0_2_0”, ”lambada_1_0_0”, ”glue_cola_2_0_0”, ”ag_news_subset_1_0_0”, ”gem_dart_1_1_0”, ”math_dataset_algebra__linear_1d_1_0_0”, ”cnn_dailymail_3_4_0”, ”wiki_hop_original_explain_relation”, ”dbpedia_14_given_list_what_category_does_the_paragraph_belong_to”, ”gem_wiki_lingua_english_en_1_1_0”, ”fix_punct”, ”imdb_reviews_plain_text_1_0_0”, ”race_middle_Write_a_multi_choice_question_for_the_following_article”, ”gigaword_1_2_0”, ”dbpedia_14_given_a_list_of_category_what_does_the_title_belong_to”, ”gem_web_nlg_en_1_1_0”, ”word_segment”, ”race_high_Write_a_multi_choice_question_for_the_following_article”, ”wmt16_translate_de_en_1_0_0”, ”cot_ecqa”, ”aeslc_1_0_0”, ”dream_generate_first_utterance”, ”wmt16_translate_fi_en_1_0_0”, ”dream_answer_to_dialogue”, ”para_crawl_enes”, ”adversarial_qa_dbert_generate_question”, ”race_middle_Write_a_multi_choice_question_options_given_”, ”wmt14_translate_fr_en_1_0_0”
c2	”adversarial_qa_dbidaf_question_context_answer”, ”super_glue_record_1_0_2”, ”wiki_hop_original_generate_object”, ”adversarial_qa_droberta_tell_what_it_is”, ”dbpedia_14_given_a_choice_of_categories_”, ”wiki_hop_original_choose_best_object_affirmative_3”, ”quac_1_0_0”, ”wiki_hop_original_choose_best_object_interrogative_1”, ”wiki_hop_original_choose_best_object_affirmative_1”, ”adversarial_qa_dbert_answer_the_following_q”, ”wiki_hop_original_choose_best_object_interrogative_2”, ”adversarial_qa_droberta_question_context_answer”, ”squad_v2_0_3_0_0”, ”wiki_hop_original_generate_subject”, ”wiki_bio_guess_person”, ”adversarial_qa_dbidaf_answer_the_following_q”, ”adversarial_qa_droberta_answer_the_following_q”, ”adversarial_qa_dbert_tell_what_it_is”, ”race_high_Write_a_multi_choice_question_options_given_”, ”wiki_hop_original_choose_best_object_affirmative_2”, ”wiki_hop_original_generate_subject_and_object”, ”drop_2_0_0”, ”adversarial_qa_dbert_question_context_answer”, ”adversarial_qa_dbidaf_tell_what_it_is”
c3	”wiqa_what_might_be_the_first_step_of_the_process”, ”wiqa_what_is_the_final_step_of_the_following_process”, ”wmt16_translate_ro_en_1_0_0”, ”wiqa_what_might_be_the_last_step_of_the_process”, ”wiki_bio_key_content”, ”gem_common_gen_1_1_0”, ”duorc_SelfRC_build_story_around_qa”, ”app_reviews_generate_review”, ”wiki_bio_what_content”, ”wiki_bio_who”, ”gem_e2e_nlg_1_1_0”, ”cot_esnli_ii”, ”wmt16_translate_tr_en_1_0_0”, ”wiqa_what_is_the_missing_first_step”, ”wiki_bio_comprehension”, ”coqa_1_0_0”, ”duorc_ParaphraseRC_build_story_around_qa”, ”multi_news_1_0_0”
c4	”wiki_qa_found_on_google”, ”app_reviews_categorize_rating_using_review”, ”race_middle_Is_this_the_right_answer”, ”super_glue_cb_1_0_2”, ”wiki_qa_Topic_Prediction_Answer_Only”, ”wiki_qa_Direct_Answer_to_Question”, ”super_glue_wsc_fixed_1_0_2”, ”cot_gsm8k_ii”, ”unified_qa_science_inst”, ”race_high_Is_this_the_right_answer”, ”cot_strategyqa”, ”cot_ecqa_ii”, ”quarel_do_not_use”, ”wiki_qa_exercise”, ”wiki_qa_automatic_system”, ”cot_creak_ii”, ”quarel_heres_a_story”, ”quarel_choose_between”, ”stream_qed_ii”, ”wiki_qa_Topic_Prediction_Question_Only”, ”glue_qnli_2_0_0”, ”cot_sensemaking_ii”, ”super_glue_copa_1_0_2”, ”social_i_qa_Generate_the_question_from_the_answer”, ”social_i_qa_Show_choices_and_generate_index”, ”quarel_testing_students”, ”wiki_qa_Topic_Prediction_Question_and_Answer_Pair”, ”wiki_qa_Decide_good_answer”, ”wiki_qa_Jeopardy_style”, ”wiki_qa_Generate_Question_from_Topic”, ”definite_pronoun_resolution_1_1_0”, ”wiqa_effect_with_label_answer”, ”glue_wnli_2_0_0”, ”cot_qasc”, ”cot_strategyqa_ii”, ”quarel_logic_test”, ”stream_aqua_ii”
c5	”quoref_Context_Contains_Answer”, ”duorc_SelfRC_generate_question_by_answer”, ”quoref_Find_Answer”, ”duorc_ParaphraseRC_movie_director”, ”duorc_ParaphraseRC_answer_question”, ”quoref_Found_Context_Online”, ”quoref_Read_And_Extract_”, ”duorc_ParaphraseRC_title_generation”, ”duorc_ParaphraseRC_decide_worth_it”, ”quoref_What_Is_The_Answer”, ”duorc_ParaphraseRC_generate_question”, ”quoref_Guess_Title_For_Context”, ”quoref_Answer_Test”, ”duorc_SelfRC_question_answering”, ”duorc_SelfRC_title_generation”, ”duorc_ParaphraseRC_generate_question_by_answer”, ”duorc_ParaphraseRC_extract_answer”, ”duorc_SelfRC_answer_question”, ”duorc_SelfRC_decide_worth_it”, ”duorc_ParaphraseRC_question_answering”, ”quoref_Answer_Question_Given_Context”, ”duorc_SelfRC_extract_answer”, ”quoref_Guess_Answer”, ”quoref_Answer_Friend_Question”, ”duorc_SelfRC_movie_director”, ”duorc_SelfRC_generate_question”, ”quoref_Given_Context_Answer_Question”
c6	”super_glue_rte_1_0_2”, ”cot_sensemaking”, ”super_glue_wic_1_0_2”, ”cos_e_v1_11_rationale”, ”anli_r3_0_1_0”, ”dream_generate_last_utterance”, ”paws_wiki_1_1_0”, ”cos_e_v1_11_generate_explanation_given_text”, ”cot_creak”, ”stream_aqua”, ”snli_1_1_0”, ”cos_e_v1_11_i_think”, ”glue_qqp_2_0_0”, ”cos_e_v1_11_explain_why_human”, ”anli_r2_0_1_0”, ”anli_r1_0_1_0”, ”glue_stsb_2_0_0”, ”cos_e_v1_11_aligned_with_common_sense”, ”glue_mnli_2_0_0”, ”social_i_qa_I_was_wondering”, ”cosmos_qa_1_0_0”, ”glue_mrpc_2_0_0”, ”social_i_qa_Generate_answer”
c7	”dream_read_the_following_conversation_and_answer_the_question”, ”app_reviews_convert_to_star_rating”, ”cos_e_v1_11_question_option_description_text”, ”social_i_qa_Show_choices_and_generate_answer”, ”quartz_answer_question_based_on”, ”sciq_Direct_Question_Closed_Book_”, ”qasc_qa_with_separated_facts_3”, ”quartz_given_the_fact_answer_the_q”, ”quartz_answer_question_below”, ”kilt_tasks_hotpotqa_final_exam”, ”sciq_Multiple_Choice”, ”wiqa_does_the_supposed_perturbation_have_an_effect”, ”cos_e_v1_11_question_description_option_text”, ”wiki_qa_Is_This_True_”, ”quartz_use_info_from_question_paragraph”, ”sciq_Direct_Question”, ”qasc_qa_with_separated_facts_2”, ”wiqa_which_of_the_following_is_the_supposed_perturbation”, ”app_reviews_convert_to_rating”, ”cos_e_v1_11_question_option_description_id”, ”wiqa_effect_with_string_answer”, ”qasc_qa_with_separated_facts_5”, ”dream_baseline”, ”quartz_having_read_above_passage”, ”cos_e_v1_11_question_description_option_id”, ”qasc_qa_with_separated_facts_1”, ”cos_e_v1_11_description_question_option_text”, ”qasc_qa_with_combined_facts_1”, ”qasc_is_correct_1”, ”cos_e_v1_11_description_question_option_id”, ”social_i_qa_Check_if_a_random_answer_is_valid_or_not”, ”sciq_Multiple_Choice_Closed_Book_”, ”quartz_use_info_from_paragraph_question”, ”qasc_is_correct_2”, ”qasc_qa_with_separated_facts_4”, ”quartz_read_passage_below_choose”, ”quartz_paragraph_question_plain_concat”, ”sciq_Multiple_Choice_Question_First”
c8	”race_middle_Read_the_article_and_answer_the_question_no_option_”, ”race_high_Select_the_best_answer”, ”quail_description_context_question_answer_id”, ”quail_context_question_description_text”, ”race_high_Read_the_article_and_answer_the_question_no_option_”, ”race_high_Select_the_best_answer_no_instructions_”, ”quail_context_description_question_answer_id”, ”race_high_Taking_a_test”, ”super_glue_multirc_1_0_2”, ”race_middle_Select_the_best_answer”, ”quail_context_question_description_answer_id”, ”quail_description_context_question_answer_text”, ”quail_context_question_answer_description_text”, ”race_high_Select_the_best_answer_generate_span_”, ”race_middle_Select_the_best_answer_generate_span_”, ”quail_context_question_answer_description_id”, ”quail_context_description_question_answer_text”, ”quail_context_description_question_text”, ”quail_context_question_description_answer_text”, ”quail_description_context_question_text”, ”race_middle_Taking_a_test”, ”quail_no_prompt_id”, ”quail_no_prompt_text”, ”race_middle_Select_the_best_answer_no_instructions_”
c9	”natural_questions_open_1_0_0”, ”web_questions_whats_the_answer”, ”web_questions_question_answer”, ”dbpedia_14_pick_one_category_for_the_following_text”, ”kilt_tasks_hotpotqa_combining_facts”, ”web_questions_short_general_knowledge_q”, ”kilt_tasks_hotpotqa_straighforward_qa”, ”adversarial_qa_dbidaf_generate_question”, ”adversarial_qa_droberta_based_on”, ”web_questions_get_the_answer”, ”kilt_tasks_hotpotqa_complex_question”, ”web_questions_potential_correct_answer”, ”trivia_qa_rc_1_1_0”, ”kilt_tasks_hotpotqa_formulate”, ”adversarial_qa_dbert_based_on”, ”adversarial_qa_dbidaf_based_on”, ”squad_v1_1_3_0_0”

Table 9. Task names for each of the 10 clusters obtained by applying `MBC` clustering to Phi-2 private library with 256 experts, with each expert trained for 2 epochs.

c0	”adversarial_qa_dbert_generate_question”, ”adversarial_qa_dbidaf_generate_question”, ”adversarial_qa_droberta_generate_question”, ”app_reviews_generate_review”, ”cot_creak”, ”cot_esnli”, ”cot_esnli_ii”, ”dream_generate_first_utterance”, ”dream_generate_last_utterance”, ”duorc_ParaphraseRC_title_generation”, ”duorc_SelfRC_title_generation”, ”fix_punct”, ”gem_common_gen_1_1_0”, ”gem_dart_1_1_0”, ”gigaword_1_2_0”, ”huggingface_xsum”, ”lambada_1_0_0”, ”race_high_Write_a_multi_choice_question_for_the_following_article”, ”race_high_Write_a_multi_choice_question_options_given_”, ”race_middle_Write_a_multi_choice_question_for_the_following_article”, ”race_middle_Write_a_multi_choice_question_options_given_”, ”stream_aqua”, ”stream_qed”, ”wiqa_what_is_the_missing_first_step”, ”wmt16_translate_fi_en_1_0_0”, ”wmt16_translate_ro_en_1_0_0”, ”yelp_polarity_reviews_0_2_0”
c1	”ag_news_subset_1_0_0”, ”app_reviews_convert_to_rating”, ”app_reviews_convert_to_star_rating”, ”cot_creak_ii”, ”cot_ecqa_ii”, ”cot_gsm8k_ii”, ”cot_sensemaking_ii”, ”cot_strategyqa”, ”dbpedia_14_given_a_choice_of_categories_”, ”dbpedia_14_given_a_list_of_category_what_does_the_title_belong_to”, ”dbpedia_14_given_list_what_category_does_the_paragraph_belong_to”, ”glue_mnli_2_0_0”, ”glue_qnli_2_0_0”, ”glue_qqp_2_0_0”, ”glue_stsb_2_0_0”, ”glue_wnli_2_0_0”, ”kilt_tasks_hotpotqa_complex_question”, ”paws_wiki_1_1_0”, ”qasc_is_correct_1”, ”qasc_is_correct_2”, ”snli_1_1_0”, ”social_i_qa_Check_if_a_random_answer_is_valid_or_not”, ”social_i_qa_Generate_answer”, ”social_i_qa_Generate_the_question_from_the_answer”, ”social_i_qa_I_was_wondering”, ”squad_v1_1_3_0_0”, ”squad_v2_0_3_0_0”, ”stream_qed_ii”, ”super_glue_multirc_1_0_2”, ”super_glue_rte_1_0_2”, ”super_glue_wic_1_0_2”, ”super_glue_wsc_fixed_1_0_2”, ”trec_1_0_0”, ”wiki_bio_guess_person”, ”wiki_qa_Is_This_True_”
c2	”app_reviews_categorize_rating_using_review”, ”cos_e_v1_11_question_option_description_text”, ”cot_qasc”, ”cot_strategyqa_ii”, ”dbpedia_14_pick_one_category_for_the_following_text”, ”definite_pronoun_resolution_1_1_0”, ”kilt_tasks_hotpotqa_final_exam”, ”math_dataset_algebra__linear_1d_1_0_0”, ”qasc_qa_with_separated_facts_4”, ”quarel_do_not_use”, ”quoref_Context_Contains_Answer”, ”race_high_Is_this_the_right_answer”, ”race_middle_Is_this_the_right_answer”, ”sciq_Direct_Question”, ”sciq_Multiple_Choice”, ”sciq_Multiple_Choice_Closed_Book_”, ”sciq_Multiple_Choice_Question_First”, ”social_i_qa_Show_choices_and_generate_index”, ”stream_aqua_ii”, ”super_glue_cb_1_0_2”, ”super_glue_copa_1_0_2”, ”unified_qa_science_inst”, ”wiki_qa_Decide_good_answer”, ”wiki_qa_Direct_Answer_to_Question”, ”wiki_qa_Generate_Question_from_Topic”, ”wiki_qa_Jeopardy_style”, ”wiki_qa_Topic_Prediction_Answer_Only”, ”wiki_qa_Topic_Prediction_Question_Only”, ”wiki_qa_Topic_Prediction_Question_and_Answer_Pair”, ”wiki_qa_automatic_system”, ”wiki_qa_exercise”, ”wiki_qa_found_on_google”
c3	”adversarial_qa_dbert_answer_the_following_q”, ”adversarial_qa_dbert_based_on”, ”adversarial_qa_dbert_question_context_answer”, ”adversarial_qa_dbert_tell_what_it_is”, ”adversarial_qa_dbidaf_answer_the_following_q”, ”adversarial_qa_dbidaf_based_on”, ”adversarial_qa_dbidaf_question_context_answer”, ”adversarial_qa_dbidaf_tell_what_it_is”, ”adversarial_qa_droberta_answer_the_following_q”, ”adversarial_qa_droberta_based_on”, ”adversarial_qa_droberta_question_context_answer”, ”adversarial_qa_droberta_tell_what_it_is”, ”cos_e_v1_11_aligned_with_common_sense”, ”cos_e_v1_11_explain_why_human”, ”cos_e_v1_11_generate_explanation_given_text”, ”cos_e_v1_11_i_think”, ”cos_e_v1_11_rationale”, ”drop_2_0_0”, ”duorc_ParaphraseRC_generate_question_by_answer”, ”duorc_SelfRC_generate_question_by_answer”, ”kilt_tasks_hotpotqa_combining_facts”, ”kilt_tasks_hotpotqa_formulate”, ”kilt_tasks_hotpotqa_straighforward_qa”, ”natural_questions_open_1_0_0”, ”trivia_qa_rc_1_1_0”, ”web_questions_get_the_answer”, ”web_questions_potential_correct_answer”, ”web_questions_question_answer”, ”web_questions_short_general_knowledge_q”, ”web_questions_whats_the_answer”
c4	”duorc_ParaphraseRC_answer_question”, ”duorc_ParaphraseRC_decide_worth_it”, ”duorc_ParaphraseRC_extract_answer”, ”duorc_ParaphraseRC_generate_question”, ”duorc_ParaphraseRC_movie_director”, ”duorc_ParaphraseRC_question_answering”, ”duorc_SelfRC_answer_question”, ”duorc_SelfRC_decide_worth_it”, ”duorc_SelfRC_extract_answer”, ”duorc_SelfRC_generate_question”, ”duorc_SelfRC_movie_director”, ”duorc_SelfRC_question_answering”, ”quac_1_0_0”, ”quoref_Answer_Friend_Question”, ”quoref_Answer_Test”, ”quoref_Find_Answer”, ”quoref_Found_Context_Online”, ”quoref_Given_Context_Answer_Question”, ”quoref_Guess_Answer”, ”quoref_Guess_Title_For_Context”, ”quoref_Read_And_Extract_”, ”quoref_What_Is_The_Answer”
c5	”cos_e_v1_11_description_question_option_id”, ”cos_e_v1_11_question_description_option_id”, ”dream_baseline”, ”dream_read_the_following_conversation_and_answer_the_question”, ”quail_context_description_question_answer_id”, ”quail_context_description_question_answer_text”, ”quail_context_description_question_text”, ”quail_context_question_answer_description_id”, ”quail_context_question_answer_description_text”, ”quail_context_question_description_answer_id”, ”quail_context_question_description_answer_text”, ”quail_context_question_description_text”, ”quail_description_context_question_answer_id”, ”quail_description_context_question_answer_text”, ”quail_description_context_question_text”, ”quail_no_prompt_id”, ”quail_no_prompt_text”, ”race_high_Read_the_article_and_answer_the_question_no_option_”, ”race_high_Select_the_best_answer”, ”race_high_Select_the_best_answer_generate_span_”, ”race_high_Select_the_best_answer_no_instructions_”, ”race_high_Taking_a_test”, ”race_middle_Read_the_article_and_answer_the_question_no_option_”, ”race_middle_Select_the_best_answer”, ”race_middle_Select_the_best_answer_generate_span_”, ”race_middle_Select_the_best_answer_no_instructions_”, ”race_middle_Taking_a_test”
c6	”cos_e_v1_11_description_question_option_text”, ”cos_e_v1_11_question_description_option_text”, ”cos_e_v1_11_question_option_description_id”, ”qasc_qa_with_combined_facts_1”, ”qasc_qa_with_separated_facts_1”, ”qasc_qa_with_separated_facts_2”, ”qasc_qa_with_separated_facts_3”, ”qasc_qa_with_separated_facts_5”, ”quarel_choose_between”, ”quarel_heres_a_story”, ”quarel_logic_test”, ”quarel_testing_students”, ”quartz_answer_question_based_on”, ”quartz_answer_question_below”, ”quartz_given_the_fact_answer_the_q”, ”quartz_having_read_above_passage”, ”quartz_paragraph_question_plain_concat”, ”quartz_read_passage_below_choose”, ”quartz_use_info_from_paragraph_question”, ”quartz_use_info_from_question_paragraph”, ”quoref_Answer_Question_Given_Context”, ”ropes_background_new_situation_answer”, ”ropes_background_situation_middle”, ”ropes_given_background_situation”, ”ropes_new_situation_background_answer”, ”ropes_plain_background_situation”, ”ropes_plain_bottom_hint”, ”ropes_plain_no_background”, ”ropes_prompt_beginning”, ”ropes_prompt_bottom_hint_beginning”, ”ropes_prompt_bottom_no_hint”, ”ropes_prompt_mix”, ”ropes_read_background_situation”, ”sciq_Direct_Question_Closed_Book_”, ”social_i_qa_Show_choices_and_generate_answer”, ”wiqa_does_the_supposed_perturbation_have_an_effect”, ”wiqa_effect_with_label_answer”, ”wiqa_effect_with_string_answer”, ”wiqa_which_of_the_following_is_the_supposed_perturbation”
c7	”aeslc_1_0_0”, ”cnn_dailymail_3_4_0”, ”coqa_1_0_0”, ”cot_gsm8k”, ”dream_answer_to_dialogue”, ”duorc_ParaphraseRC_build_story_around_qa”, ”duorc_SelfRC_build_story_around_qa”, ”gem_e2e_nlg_1_1_0”, ”gem_web_nlg_en_1_1_0”, ”gem_wiki_lingua_english_en_1_1_0”, ”multi_news_1_0_0”, ”wiki_bio_comprehension”, ”wiki_bio_key_content”, ”wiki_bio_what_content”, ”wiki_bio_who”, ”wiqa_what_is_the_final_step_of_the_following_process”, ”wiqa_what_might_be_the_first_step_of_the_process”, ”wiqa_what_might_be_the_last_step_of_the_process”, ”wmt16_translate_tr_en_1_0_0”
c8	”anli_r1_0_1_0”, ”anli_r2_0_1_0”, ”anli_r3_0_1_0”, ”cosmos_qa_1_0_0”, ”cot_ecqa”, ”cot_sensemaking”, ”glue_cola_2_0_0”, ”glue_mrpc_2_0_0”, ”glue_sst2_2_0_0”, ”imdb_reviews_plain_text_1_0_0”, ”para_crawl_enes”, ”super_glue_record_1_0_2”, ”true_case”, ”wmt14_translate_fr_en_1_0_0”, ”wmt16_translate_de_en_1_0_0”, ”word_segment”
c9	”wiki_hop_original_choose_best_object_affirmative_1”, ”wiki_hop_original_choose_best_object_affirmative_2”, ”wiki_hop_original_choose_best_object_affirmative_3”, ”wiki_hop_original_choose_best_object_interrogative_1”, ”wiki_hop_original_choose_best_object_interrogative_2”, ”wiki_hop_original_explain_relation”, ”wiki_hop_original_generate_object”, ”wiki_hop_original_generate_subject”, ”wiki_hop_original_generate_subject_and_object”

Table 10. Task names for each of the 10 clusters obtained by applying MBC clustering to Mistral 7B private library with 256 experts.