SEQR: Secure and Efficient QR-based LoRA Routing

William Fleshman & Benjamin Van Durme
Johns Hopkins University
will.fleshman@jhu.edu

Abstract

Low-Rank Adaptation (LoRA) has become a standard technique for parameter-efficient fine-tuning of large language models, enabling large libraries of LoRAs, each for a specific task or domain. Efficiently selecting the correct LoRA adapter for a given input remains a challenge, particularly in secure environments where supervised training of routers may raise privacy concerns. Motivated by previous approaches, we formalize the goal of unsupervised LoRA routing in terms of activation norm maximization, providing a theoretical framework for analysis. We demonstrate the discriminative power of activation norms and introduce SEQR, an unsupervised LoRA routing algorithm designed to maximize efficiency while providing strict routing guarantees. SEQR provably identifies the norm-maximizing adapter with significantly greater efficiency, making it a highly scalable and effective solution for dynamic LoRA composition. We validate our results through experiments that demonstrate improved multi-task performance and efficiency.

1 Introduction

Language model users can benefit from fine-tuning existing models on custom data, but may be constrained by security policies surrounding data access control or retention (Fleshman et al., 2024; Shi et al., 2025). Low-Rank Adaptation (LoRA) (Hu et al., 2022) is a popular parameter-efficient technique for fine-tuning these models. Widely-used software packages, such as `peft` (Mangrulkar et al., 2022), and model repositories, such as *huggingface* (Wolf et al., 2020), have contributed to the proliferation of LoRA-based experts fine-tuned for various tasks or data domains (Brüel-Gabrielsson et al., 2024; Huang et al., 2024). The broad deployment of language models has led to techniques for securing and controlling training data (Fleshman et al., 2024; Chowdhury et al., 2025; Shi et al., 2025). For example, AdapterSwap leverages LoRA adapters to segment data into separate parameter groups, enabling user-based access control at the model level (Fleshman et al., 2024). The authorized LoRAs for a particular user can then be applied to the model at inference time, and adapters can be quickly retrained if training data is later removed to meet retention polices (Fleshman et al., 2024).

Naively applying all authorized LoRAs to a model can lead to parameter interference, significantly reducing the model performance (Wortsman et al., 2022; Chronopoulou et al., 2023; Ilharco et al., 2023; Fleshman et al., 2024). Numerous model merging strategies have been developed to address this challenge (Ortiz-Jimenez et al., 2023; Yadav et al., 2023; Tang et al., 2024; Yu et al., 2024; Stoica et al., 2025). Alternatively, LoRAs for the same model can be treated as a *mixture-of-experts* (Jacobs et al., 1991; Fedus et al., 2022) by learning to route inputs to a smaller set of appropriate adapters (Pfeiffer et al., 2021; Wang et al., 2022; Caccia et al., 2023; Ponti et al., 2023; Fleshman et al., 2024; Huang et al., 2024; Zadouri et al., 2024). Multi-LoRA frameworks have also been used for federated learning, where LoRA training dynamics suggest that the LoRA A matrices learn global features which can be shared among the different adapters (Sun et al., 2024b; Guo et al., 2025).

Supervised training of a router using data across protected silos is not an option in strict data security scenarios, as adversarial techniques exist for leaking information related to the data (Shokri et al., 2017; Carlini et al., 2022; Yao, 2024; Zhou et al., 2025). Recent approaches perform LoRA routing in an unsupervised manner by selecting adapters for a given input without any router training or cross-silo data requirements (Ostapenko et al., 2024; Fleshman & Van Durme, 2025a;b). In this work, we formalize the goal of these techniques and analyze their routing procedures. We introduce a new

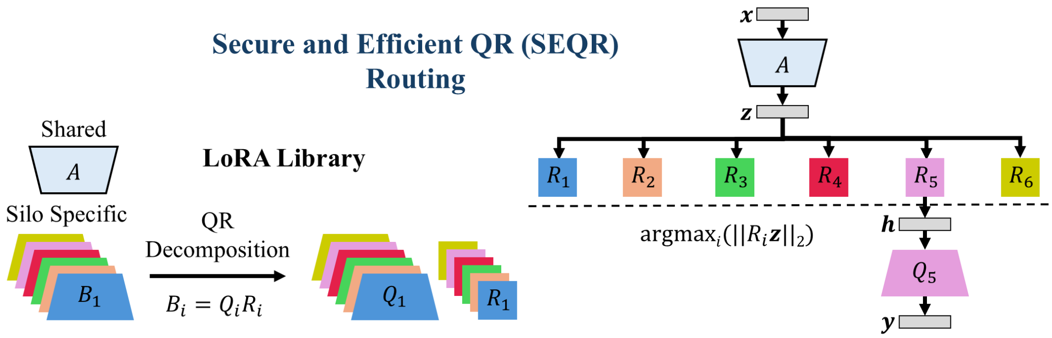

Figure 1: Secure and Efficient QR (SEQR) routing: Rank-r LoRAs are trained on multiple datasets using a shared A matrix frozen at initialization. Each B_i is stored in terms of its QR decomposition. During inference, input vectors are routed efficiently using the smaller $r \times r$ matrices.

method, SEQR (Figure 1), which is more efficient than previous approaches while providing strict routing guarantees. Specifically we:

- Formalize unsupervised LoRA routing as activation norm-maximization;
- Provide theoretical results for current approaches under this framework;
- Introduce a more efficient routing scheme, SEQR, which provably selects the norm-maximizing adapter; and
- Perform empirical experiments demonstrating the benefits of our approach.

2 Background and Related Work

2.1 LoRA

LoRA updates the pretrained layer weights $W_0 \in \mathbb{R}^{m \times n}$ by freezing the existing weights and injecting two low-rank matrices of learnable parameters $A \in \mathbb{R}^{r \times n}$ and $B \in \mathbb{R}^{m \times r}$ such that the new weights are $W = W_0 + BA$, with a small rank $r \ll \min(m, n)$ that considerably reduces the number of trainable parameters (Hu et al., 2022). For an input vector $\mathbf{x} \in \mathbb{R}^n$, the output $\mathbf{y} \in \mathbb{R}^m$ can be computed directly with the new weights as $\mathbf{y} = W\mathbf{x}$ or separately as $\mathbf{y} = W_0\mathbf{x} + BA\mathbf{x}$. LoRA routing is necessary in the case where many LoRAs are trained on different groups of data, resulting in a set of available LoRAs $\mathcal{A} = \{B_1A_1, B_2A_2, \ldots, B_NA_N\}$ for each adapted layer of the model. The goal of unsupervised routing is to choose the LoRA(s) best suited for each vector in a sequence, without explicitly training the router (Ostapenko et al., 2024; Fleshman & Van Durme, 2025b;a).

2.2 Privacy & Security

Organizations may have various security or privacy concerns depending on the data used for training individual LoRAs. Training with differential privacy (DP) provides probabilistic guarantees that an adversary can not infer if particular examples were in the training data (Dwork & Roth, 2014; Abadi et al., 2016). DP can be used to protect user privacy in cases where adversaries may have access to the LoRA weights (Shi et al., 2025). Stricter security requirements incorporate data access control, completely preventing user access to LoRA weights trained on data the user is unauthorized to view (Fleshman et al., 2024). In these cases, training a router to distinguish between LoRAs would introduce security concerns, as adversaries with access to the router could potentially leak information from the LoRAs themselves (Shokri et al., 2017; Carlini et al., 2022; Yao, 2024; Zhou et al., 2025). We focus on the strict security case, where unsupervised routing approaches are needed.

2.3 Activation Norms

Unsupervised LoRA routing can be framed as an *in-distribution* (ID) detection problem, where inputs are routed to the adapters trained on data similar to the queries. Prior work has shown that the norm of the activation vector produced by model layers can effectively distinguish between in- and out-of distribution (OOD) data (Park et al., 2023; Liu et al., 2024; Shin & Chung, 2024; Sun et al., 2024a; Wan et al., 2024). ID data tends to produce large activation spikes in neural networks, including in large language models (Sun et al., 2024a). Park et al. (2023) analyze this phenomenon and find that the activation norm distinguishes OOD and ID similar to a classifier confidence score. These findings justify trying to route to LoRAs which maximize the norm of adapter activations $||BA\mathbf{x}||$.

2.4 ARROW ROUTING

Ostapenko et al. (2024) use the singular value decomposition (SVD) to convert each LoRA adapter $B_iA_i \in \mathcal{A}$ into a product of three matrices with an equivalent product:

$$B_iA_i = U_iS_iV_i^T, \tag{1}$$

where $U_i \in \mathbb{R}^{m\times r}$ is the orthonormal matrix of left singular vectors, $S_i \in \mathbb{R}^{r\times r}$ is the diagonal matrix of singular values, and $V_i \in \mathbb{R}^{n\times r}$ is the orthonormal matrix of right singular vectors. ARROW routing leverages the fact that the right singular vector $\mathbf{v}_i$ associated with the largest singular value corresponds to the direction capturing the most variation in the space of input vectors $\mathbf{x}$ (Ostapenko et al., 2024). This *arrow vector* $\mathbf{v}_i$ satisfies $\mathbf{v}_i = \max_{\mathbf{x},||\mathbf{x}||_2=1}||B_iA_i\mathbf{x}||_2$, meaning it maximizes the norm of the corresponding adapter activations among unit-length input vectors. We use norm-maximization as the explicit goal in this work, allowing for analysis of these approaches. Ostapenko et al. (2024) use the set of arrows as prototypes for each of the adapters in $\mathcal{A}$, assigning the most weight to the adapter corresponding to the arrow satisfying $\mathrm{argmax_i}|\mathbf{v}_i^T\mathbf{x}|$. The use of vector prototypes makes ARROW routing especially efficient, requiring a simple dot product per adapter: $\mathcal{O}(Nn)$ for N adapters with input dimension n. ARROW routing performs reasonably well, and the authors empirically show that the ID adapter tends to produce higher ARROW scores (Ostapenko et al., 2024).

2.5 SPECTRAL ROUTING AND LAG

SPECTR builds on ARROW by using all right singular vectors to make routing decisions (Fleshman & Van Durme, 2025b). Equation 1 is used by SPECTR to convert each adapter into two new matrices:

$$\hat{B}_i = U_i \tag{2}$$

$$\hat{A}_i = S_iV_i^T, \tag{3}$$

such that $\hat{B}_i\hat{A}_i = B_iA_i$ with $\hat{A}_i$ now containing the orthogonal directions of maximum variation scaled by the singular values. SPECTR generalizes the ARROW scoring method by assigning the most weight to the adapter satisfying $\mathrm{argmax}_i||\hat{A}_i\mathbf{x}||_2$. Computing the SPECTR routing scores is less efficient than ARROW: $\mathcal{O}(Nrn)$, but SPECTR outperforms ARROW in routing accuracy and downstream task performance (Fleshman & Van Durme, 2025b).

LoRA-Augmented Generation (LAG) combines the efficiency of ARROW routing with the improved performance of SPECTR by using a two-stage approach (Fleshman & Van Durme, 2025a). First, LAG performs top-k filtering using ARROW to reduce the final routing decision to $k \ll N$ adapters. LAG then uses SPECTR to route to the top adapter in the filtered set. Routing complexity is reduced to $\mathcal{O}(Nn + krn)$ while still outperforming ARROW (Fleshman & Van Durme, 2025a).

2.6 SHARED A

While ARROW, SPECTR, and LAG use traditional LoRA fine-tuning, recent work explores a special case of LoRA where the A matrix is frozen at initialization or shared among several LoRAs in a federated setting, resulting in similar or improved performance with reduced storage costs (Zhang et al., 2023; Sun et al., 2024b; Zhu et al., 2024; Guo et al., 2025). Zhu et al. (2024) provides a theoretical analysis showing that the LoRA updates are dominated by the B matrix during fine-tuning, and that a LoRA with a frozen random A matrix should perform similarly to one that is fully trained. The asymmetry in training dynamics lends itself to using a global A matrix and unique B matrices in multi-LoRA scenarios (Sun et al., 2024b; Guo et al., 2025). We explore this direction in our work, and show that a shared A matrix allows for more efficient unsupervised LoRA routing techniques.

3 Theoretical Results and SEQR

Problem Statement We formalize the goal of unsupervised LoRA routing to provide a framework for theoretical analysis. Given the success of using activations for ID/OOD detection and the similar motivation of current unsupervised routing approaches, we propose the following problem:

Problem

LoRA Activation Norm-Maximization. Given a library of LoRA adapters, $\mathcal{A} = \{B_1A_1, B_2A_2, \ldots, B_NA_N\}$ and an input vector $\mathbf{x}$, efficiently find $\text{argmax}_i ||B_iA_i\mathbf{x}||_2$.

We add "efficiently" to the problem statement as an algorithm that simply computes all activation norms directly would be $\mathcal{O}(Nr(m+n))$, far worse than current routing approaches. We will demonstrate the discriminative power of LoRA activation norms in Section 4.3.

3.1 Arrow is not Norm-Maximizing

Our first result shows that Arrow is not guaranteed to find the norm-maximizing adapter.

Theorem 3.1. *There exists a set of LoRA adapters* $\{B_1A_1, B_2A_2, \ldots, B_NA_N\}$ *with corresponding arrow vectors* $\{v_1, v_2, \ldots, v_N\}$ *and* $\mathbf{x} \in \mathbb{R}^n$ *where* $\text{argmax}_i |\mathbf{v}_i^T\mathbf{x}| \neq \text{argmax}_i ||B_iA_i\mathbf{x}||_2$.

We provide the proof by construction in Appendix A and confirm with experiments in Section 4.4. The main observation from the proof is that alignment with the top singular vector is not enough to guarantee the adapter will have the largest norm, as misalignment can be overcome with larger singular values. Routing with Lag inherits the lack of guarantee from Arrow, but the top-k selection improves the chances of including the norm-maximizing adapter in the set used for SpectR selection.

3.2 SpectR is Norm-Maximizing

Our next results show that SpectR scores are equivalent to the activation norms, and therefore SpectR is norm-maximizing. The proof for Theorem 3.2 is provided in Appendix B.

Theorem 3.2. *Let* $B \in \mathbb{R}^{m\times r}$ *and* $A \in \mathbb{R}^{r\times n}$ *be LoRA matrices with* $\hat{A}$ *derived from* BA *using Equations 1 and 3, then* $\forall \mathbf{x} \in \mathbb{R}^n$, $||\hat{A}\mathbf{x}||_2 = ||BA\mathbf{x}||_2$.

Corollary 3.2.1. *Let* $\{B_1A_1, B_2A_2, \ldots, B_NA_N\}$ *be a set of LoRA adapters converted with Equations 1-3 to the set* $\{\hat{B}_1\hat{A}_1, \hat{B}_2\hat{A}_2, \ldots, \hat{B}_N\hat{A}_N\}$, *then* $\text{argmax}_i ||\hat{A}_i\mathbf{x}||_2 = \text{argmax}_i ||B_iA_i\mathbf{x}||_2$.

These results show that SpectR provides optimal routing under the stated goal. We are interested in new approaches providing the same guarantee but with improved efficiency.

3.3 Secure and Efficient QR (SEQR) Routing

Now we explore the special case of our problem statement where all adapters in $\mathcal{A}$ share the same matrix A. This matrix is randomly initialized and kept frozen to ensure the same data security provided by other unsupervised routing approaches. For an input $\mathbf{x}$, we compute $\mathbf{z} = A\mathbf{x}$ as an intermediate step. Routing is then required for the set of B matrices. Directly computing the norm for all would require $\mathcal{O}(Nmr)$, which is already equivalent to SpectR for $m = n$. We can improve further by doing a one-time preprocessing step similar to the SVD in Arrow and SpectR. We precompute the reduced QR decomposition of each B_i:

$$B_i = Q_iR_i, \tag{4}$$

where $Q_i \in \mathbb{R}^{m\times r}$ is an orthogonal matrix and $R_i \in \mathbb{R}^{r\times r}$ is upper triangular. Similar to SpectR, we can throw away the original B_i and store the adapter in this new form. The vector $\mathbf{z}$ is then routed to the adapter satisfying $\text{argmax}_i ||R_i\mathbf{z}||_2$.[1] The routing complexity is only $\mathcal{O}(Nr^2)$, which is far better than SpectR and is even more efficient than Arrow routing in the typical LoRA scenario where $r \ll n$. We present the complete Seqr routing process in Algorithm 1. Like SpectR, we show Seqr scores are equivalent to the activation norm for each adapter. Therefore, Seqr always selects the norm-maximizing adapter. The proof for Theorem 3.3 is provided in Appendix C.

[1] We z-score these raw scores based on our findings in Section 4.3.

Algorithm 1 Secure and Efficient QR (SEQR) Routing

Require: Pretrained weight matrix $W \in \mathbb{R}^{m \times n}$
  Shared adapter matrix $A \in \mathbb{R}^{r \times n}$ ▷ Randomly initialized and frozen during training
  LoRA matrices $\{B_i \in \mathbb{R}^{m \times r}\}_{i=1}^{N}$
  Norm statistics $\{\mu_i, \sigma_i\}_{i=1}^{N}$ ▷ Estimated using training data

  Preprocessing
  for each adapter B_i **do**
    Compute reduced QR decomposition: $B_i = Q_i R_i$ ▷ B_i can be discarded
  end for

  Inference (given input $\mathbf{x} \in \mathbb{R}^n$)
  Compute shared intermediate representation: $\mathbf{z} \leftarrow A\mathbf{x}$
  for each adapter $i = 1, \ldots, N$ **do**
    Projected activation: $\mathbf{h}_i \leftarrow R_i \mathbf{z}$
    Score: $s_i \leftarrow (\|\mathbf{h}_i\|_2 - \mu_i)/\sigma_i$ ▷ Z-scored activation norm
  end for
  Select top adapter: $i^* \leftarrow \arg\max_i s_i$ ▷ Adapter with max activation norm
  Compute final output: $\mathbf{y} \leftarrow W\mathbf{x} + Q_{i^*}\mathbf{h}_{i^*}$ ▷ $Q_{i^*}\mathbf{h}_{i^*} = B_{i^*}A\mathbf{x}$
  return y

Theorem 3.3. *Let $B \in \mathbb{R}^{m \times r}$ and $A \in \mathbb{R}^{r \times n}$ be LoRA matrices such that $B = QR$ from Equation 4, then $\forall \mathbf{x} \in \mathbb{R}^n$, $||RA\mathbf{x}||_2 = ||BA\mathbf{x}||_2$.*

Corollary 3.3.1. *Let $\{B_1, B_2, \ldots, B_N\}$ be a set of LoRA adapters with a shared A matrix and $\{Q_1R_1, Q_2R_2, \ldots, Q_NR_N\}$ from Equation 4, then $\text{argmax}_i||R_iA\mathbf{x}||_2 = \text{argmax}_i||B_iA_i\mathbf{x}||_2$.*

3.4 Routing Complexity

We revisit the routing complexities of ARROW routing, SPECTR, LAG, and SEQR using dimensions reported in the LAG experiments for added context (Fleshman & Van Durme, 2025a). Table 1 includes the FLOPs used for routing by each method in this example, including the naive approach of computing the norm directly for each adapter. SEQR is two orders of magnitude more efficient than any other approach. SEQR also decreases storage costs by offsetting the storage of each R_i by sharing A across the library. ARROW can also take advantage of improved storage when using a shared A matrix, but arrow vectors require more space than the R_i matrices when $n > r^2$.

Table 1: Routing complexity and example FLOPs for each method assuming $N = 1000$ adapters, $n = m = 4096$ hidden dimension, $k = 20$ LAG filtering, and $r = 8$ rank adapters.

	Naive	SPECTR	LAG	ARROW	SEQR
FLOPs	66M	33M	5M	4M	**64**K
Complexity	$\mathcal{O}(Nr(m+n))$	$\mathcal{O}(Nrn)$	$\mathcal{O}(Nn + krn)$	$\mathcal{O}(Nn)$	$\mathcal{O}(Nr^2)$

4 Experiments

We conduct experiments to validate our theoretical results and to test whether SEQR provides similar or better performance over less efficient alternatives. First, we confirm prior work showing that using a fixed A matrix in LoRA works as well as learning A individually. We analyze the differences in activation norms between these two settings and introduce a calibration step to ensure norms between adapters are on the same scale. We measure the ability of each approach to select the norm-maximizing adapter and the resulting multi-task performance and efficiency.

Table 2: Accuracy for LoRAs using a unique or fixed A matrix shared across datasets.

	agnews	cola	dbped	hswag	mnli	mrpc	qnli	qqp	rte	sst2	AVG
Unique	90.4	78.8	98.7	83.6	86.1	84.7	84.9	86.5	88.2	92.4	87.4
Shared	90.0	78.9	99.0	81.5	85.7	85.0	85.5	86.3	87.9	92.8	87.3

4.1 Models and Data

We replicate the experiments of Fleshman & Van Durme (2025b) using the Llama-3.2-3B-Instruct model (Grattafiori et al., 2024). We train LoRAs for a variety of tasks: agnews[2], cola (Warstadt et al., 2019), dbpedia (Auer et al., 2007), hellaswag (Zellers et al., 2019), mnli (Williams et al., 2018), mrpc (Dolan & Brockett, 2005), qnli (Rajpurkar et al., 2016), qqp,[3] rte (Wang et al., 2018), and sst2 (Socher et al., 2013). Similar to Ostapenko et al. (2024), we subsample the datasets for computational feasibility. Using different random seeds, we produce three sets of LoRAs per dataset and category (shared vs. unique A matrix), each trained on 1000 samples from the corresponding dataset. Learning rates were optimized per dataset and category but shared across random seeds. All evaluations are performed using a held-out set of 1000 examples from each dataset. LoRA A matrices are initialized from $\mathcal{N}(0, 1/r^2)$ and frozen in the shared setting. The B matrices are initialized with 0s and trained in both cases (Hu et al., 2022). The complete adapter training details are included in Appendix D.

4.2 Unique vs. Shared

Before measuring routing performance, we ensure that using frozen A matrices results in similar LoRA performance. Table 2 shows the accuracy of each adapter on the corresponding test set, averaged across the three different initializations. Accuracy is within 1% between the two categories in most cases, with the largest deviation being a 2% difference on hellaswag when using the frozen A matrices. Overall, the average performance is nearly identical, a finding consistent with prior work showing similar performance with a frozen A (Zhang et al., 2023; Sun et al., 2024b; Zhu et al., 2024).

4.3 Activation Norms

Activation norms of a given adapter can be informative for distinguishing ID from OOD data. However, to ensure bias-free routing, these norms must be comparable across adapters. For instance, the agnews adapter may produce lower norms than the cola adapter regardless of the dataset, even if it generates higher norms on agnews data specifically. In such cases, the routing procedure would be biased toward selecting the cola adapter. We explore and mitigate this potential bias in norms.

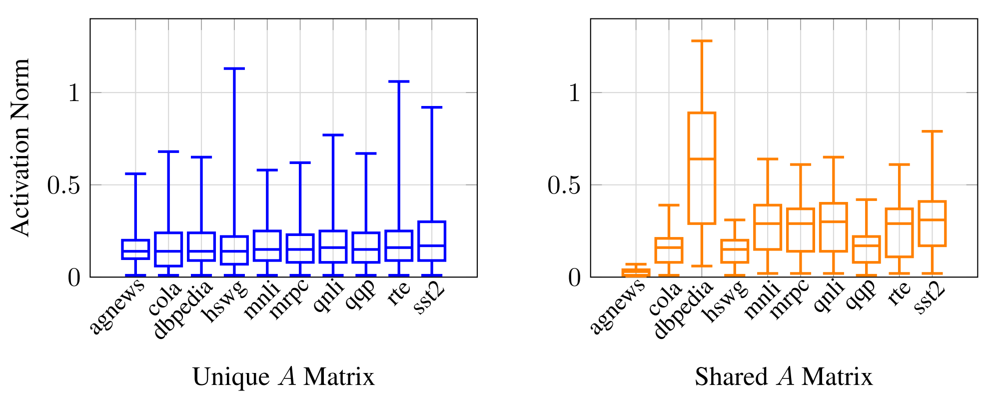

Figure 2: Distribution of average activation norms for each dataset when using LoRA adapters with unique A matrices or a fixed A matrix shared across adapters.

We gather the average activation norms across model layers for each adapter in Figure 2. We find that the activation norms are very consistent across adapters when using LoRAs trained with unique

[2] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[3] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

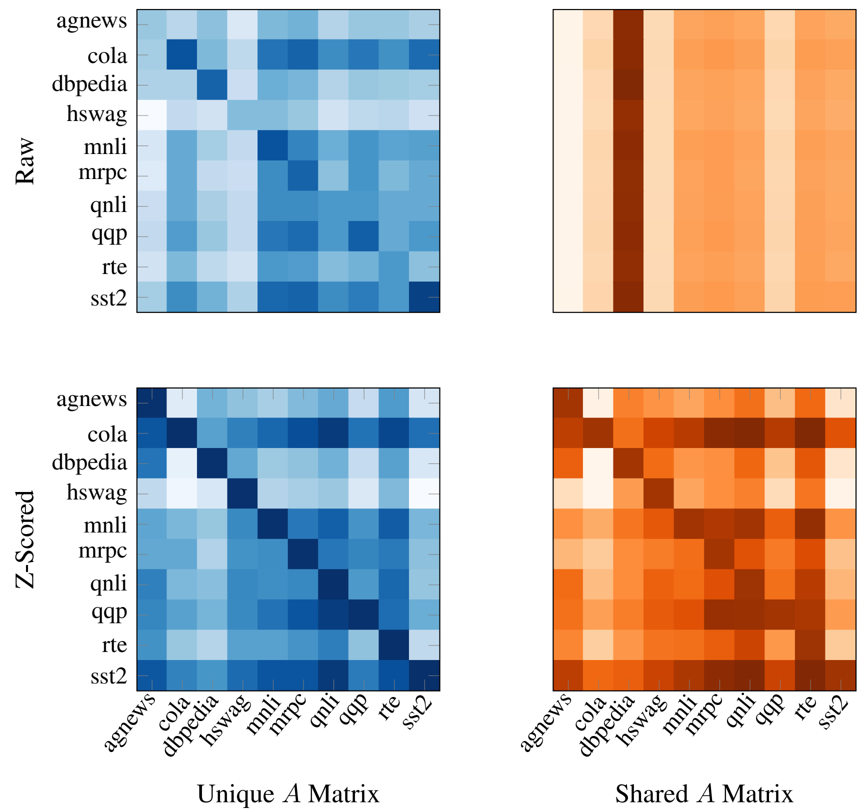

Figure 3: Raw (top) versus z-scored (bottom) activation norms for the adapters using unique (left) or shared (right) A matrices. Rows are datasets and columns are the applied adapter.

A matrices. However, when the adapters share a frozen A matrix across datasets, the variance in activation norms is considerable. To address the issue, we introduce an offline calibration step for the norm-based approaches. We compute the mean, μ_i, and standard deviation, σ_i, of the activation norm for each adapter using its associated training data. The scores for SPECTR become $s_i = (||\hat{A}_i\mathbf{x}||_2 - \mu_i)/\sigma_i$ and similarly for SEQR $s_i = (||R_iA\mathbf{x}||_2 - \mu_i)/\sigma_i$, which are the z-scores of the original raw scores to ensure all adapters are on the same scale. These normalized scores are already included in Algorithm 1. ARROW scores remain the same, as $\mathbf{v}_i$ is unit-length by construction.

We visualize the impact of z-scoring by measuring the average activation norm for each adapter, on each dataset, before and after normalizing the scores (Figure 3). We see that the norms for adapters with unique A matrices are already discriminative, but normalizing does sharpen the distribution. The biased norms in the shared case completely prevent accurate discrimination, with the dbpedia adapter producing the largest average norm regardless of the dataset. Z-scoring significantly improves the results, leading to similar relative averages when compared to the traditionally trained LoRAs with unique A matrices.

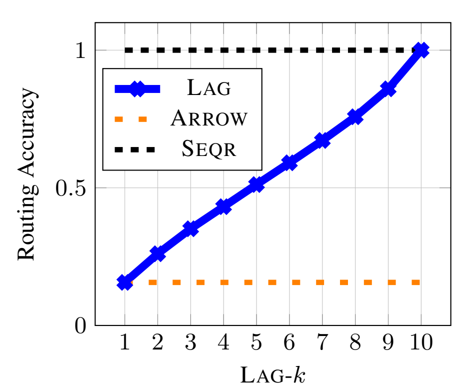

Figure 4: Routing accuracy as k increases for LAG. LAG is equivalent to ARROW at $k = 1$ and to SPECTR at $k = 10$. ARROW chooses the norm-maximizing adapter for 16% of tokens.

4.4 ROUTING ACCURACY

We validate our theoretical results by measuring the percentage of tokens routed to the norm-maximizing adapter (Figure 4). ARROW chooses

Table 3: Mean and standard deviation of performance achieved across datasets and routing methods. SpectR achieves identical performance as SEQR but at a higher computational cost.

	Mu	Arrow	LAG	SEQR
agnews	16 ± 9.7	89 ± 0.9	89 ± 1.0	**91** ± 0.5
cola	86 ± 3.6	92 ± 1.6	94 ± 0.8	**96** ± 0.9
dbpedia	89 ± 2.6	**100** ± 0.1	**100** ± 0.2	**100** ± 0.2
hswag	53 ± 0.0	78 ± 12.2	86 ± 2.1	**87** ± 2.3
mnli	48 ± 2.5	78 ± 2.1	**81** ± 6.2	**81** ± 7.5
mrpc	76 ± 4.5	92 ± 1.2	**93** ± 1.5	**93** ± 2.2
qnli	79 ± 0.6	92 ± 2.8	95 ± 0.9	**96** ± 1.4
qqp	61 ± 4.4	**97** ± 0.9	**97** ± 0.2	**97** ± 0.1
rte	73 ± 1.1	94 ± 3.1	**96** ± 1.5	**96** ± 2.1
sst2	94 ± 0.3	97 ± 1.0	**98** ± 0.9	**98** ± 2.3
AVG	67.5	90.9	92.9	**93.5**

the adapter with the top singular vector most aligned to the input. This adapter is the norm-maximizing adapter for only 16% of tokens. The routing accuracy of LAG scales almost linearly with k, as LAG is equivalent to Arrow at $k = 1$ and equivalent to SpectR at $k = 10$. SpectR and SEQR both choose the norm-maximizing adapter in all cases. These results empirically confirm our theoretical findings and are consistent with prior work showing Arrow routing accuracies just above random change and improved routing with SpectR (Fleshman & Van Durme, 2025b).

4.5 Task Performance

We measure multi-task performance by evaluating the routing methods on the withheld data from each dataset. Keeping with previous work, we include Mu-routing as an additional baseline (Ostapenko et al., 2024; Fleshman & Van Durme, 2025b). Mu forgoes routing to individual LoRAs and instead computes the mean update using all adapters: $\mathbf{y} = W\mathbf{x} + \frac{1}{N}\sum_{i=1}^{N} B_i A\mathbf{x}$. While simple, averaging adapters can lead to poor performance due to interference in parameter space, especially with a large number of adapters (Ortiz-Jimenez et al., 2023; Tang et al., 2024; Stoica et al., 2025).

Fleshman & Van Durme (2025a) use $k = 20$ with LAG, filtering their adapter library to 2% of the total before using SpectR to make the final selection. With only 10 adapters, we use a 30% reduction with $k = 3$ for demonstration purposes, but note the LAG task performance is equivalent to Arrow for $k = 1$ and to SpectR and SEQR at $k = 10$. We control for variation in task difficulty by dividing each score by the performance of the correct adapter from Table 2 (Ostapenko et al., 2024; Fleshman & Van Durme, 2025b). We report the mean performance and standard deviation over three random seeds in Table 3. SEQR and SpectR route equivalently, so we only include SEQR in the table. All other approaches significantly outperform Mu routing. SEQR achieves the highest average score in all cases, outperforming Arrow and LAG.[4] The similar task-performance with LAG and identical performance with SpectR make differences in efficiency a primary consideration for choosing among the various approaches. Next, we explore these differences in more detail.

4.6 Routing Efficiency

SEQR yields the same improved multi-task performance as SpectR, but with far greater efficiency. We measure the realized FLOPs and peak GPU memory used by each approach under various conditions (Figure 5). Total memory usage is dominated by the storage of the adapter library, so SEQR and Arrow are around twice as efficient when using shared A matrices. SpectR and LAG require storing unique $\hat{A}_i$ matrices per adapter, even when the original A matrix is shared. SEQR stores an extra Nr^2 parameters for the R_i matrices while Arrow stores an extra Nn for the arrow vectors. This gives SEQR an additional advantage in storage costs when $r^2 < n$. For computation, SEQR provides a significant reduction in FLOPs over other methods, especially for large adapter libraries using a smaller LoRA rank per adapter. Arrow requires fewer FLOPs than SEQR when

[4] A paired t-test produces $p = 0.013$ when comparing with Arrow and $p = 0.096$ with LAG.

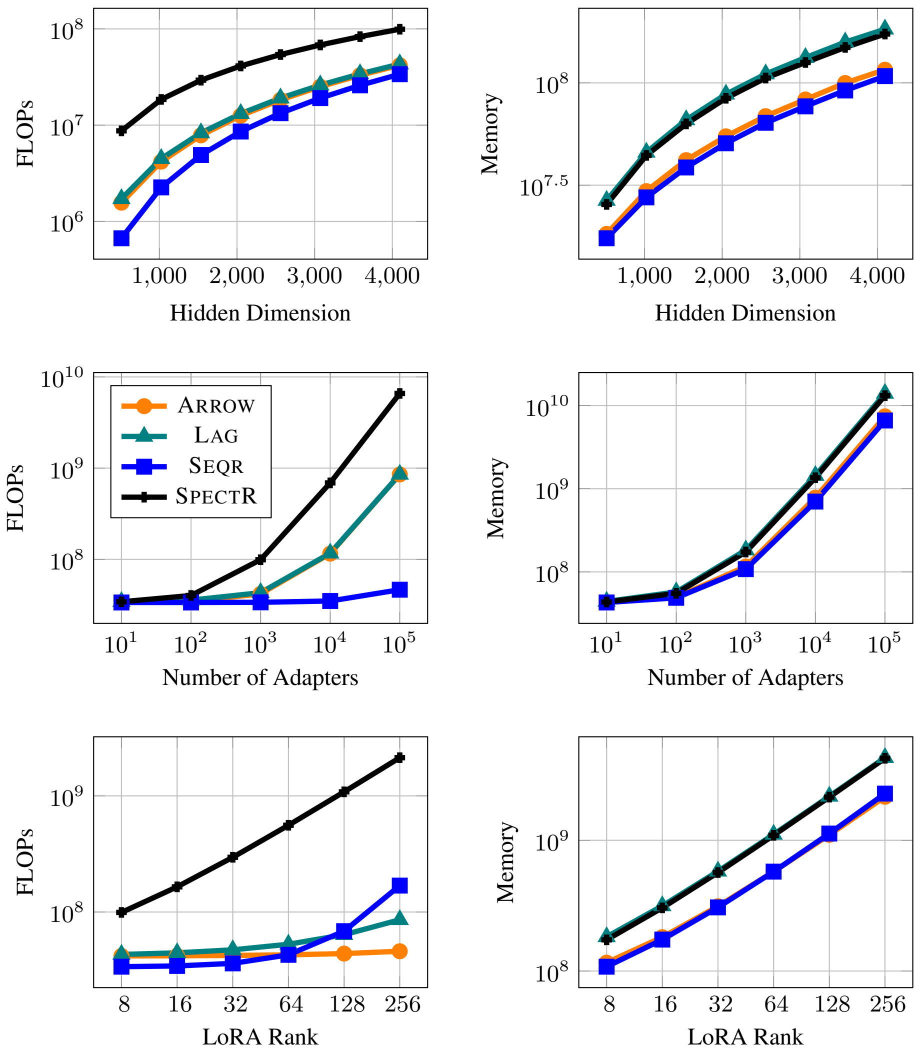

Figure 5: FLOPs (left) and GPU bytes used (right) for each method while varying hidden dimension (top), number of adapters in library (middle), and LoRA rank (bottom). Settings are fixed to $n = 4096$, $N = 1,000$, and $r = 8$ when not under evaluation. LAG uses $k = 20$ for ARROW filtering.

$r > \sqrt{n}$, but the relative task-performance of ARROW degrades at higher rank, where routing decisions are still limited by the rank-1 prototypes (Fleshman & Van Durme, 2025b).

5 Conclusion

In conclusion, we introduced SEQR, a state-of-the-art unsupervised LoRA routing algorithm. We formalized the goal of unsupervised LoRA routing in terms of activation norm-maximization and provided theoretical results for previous routing methods under this framework. The approaches that guarantee selecting the norm-maximizing adapter had better multi-task performance in our experiments. We showed that SEQR has this guarantee while being orders of magnitude more efficient than existing alternatives. SEQR leverages prior work showing that similar performance can be achieved when using LoRAs with frozen A matrices shared across adapters, a finding we empirically validate. Sharing the A matrices resulted in a higher variance in activation norms, which we corrected via an offline calibration step. Calibration improved performance for SPECTR and SEQR, with SEQR being significantly faster in execution due to the increased efficiency. SEQR maintains the security benefits of other unsupervised methods, preventing data leakage without access to the LoRA weights.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux (eds.), *The Semantic Web*, pp. 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-76298-0.

Rickard Brüel-Gabrielsson, Jiacheng Zhu, Onkar Bhardwaj, Leshem Choshen, Kristjan Greenewald, Mikhail Yurochkin, and Justin Solomon. Compress then serve: Serving thousands of lora adapters with little overhead, 2024. URL https://arxiv.org/abs/2407.00066.

Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. Multi-head adapter routing for cross-task generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qcQhBli5Ho.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Somnath Basu Roy Chowdhury, Krzysztof Marcin Choromanski, Arijit Sehanobish, Kumar Avinava Dubey, and Snigdha Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oe51Q5Uo37.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. AdapterSoup: Weight averaging to improve generalization of pretrained language models. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2054–2063, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.153. URL https://aclanthology.org/2023.findings-eacl.153/.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002/.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. 23(1), January 2022. ISSN 1532-4435.

William Fleshman and Benjamin Van Durme. Lora-augmented generation (lag) for knowledge-intensive language tasks, 2025a. URL https://arxiv.org/abs/2507.05346.

William Fleshman and Benjamin Van Durme. Spectr: Dynamically composing lm experts with spectral routing, 2025b. URL https://arxiv.org/abs/2504.03454.

William Fleshman, Aleem Khan, Marc Marone, and Benjamin Van Durme. Adapterswap: Continuous training of llms with data removal and access-control guarantees. In *Proceedings of Conference on Applied Machine Learning in Information Security (CAMLIS) 2024*, 2024. URL https://arxiv.org/abs/2404.08417.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iX3uESGdsO.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TrloAXEJ2B.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Robert Jacobs, Michael Jordan, Steven Nowlan, and Geoffrey Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 03 1991. doi: 10.1162/neco.1991.3.1.79.

Yibing Liu, Chris XING TIAN, Haoliang Li, Lei Ma, and Shiqi Wang. Neuron activation coverage: Rethinking out-of-distribution detection and generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SNGXbZtK6Q.

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

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0A9f2jZDGW.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0ZFWfeVsaD.

Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1557–1567, 2023. doi: 10.1109/ICCV51070.2023.00150.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39/.

Edoardo Maria Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. Combining parameter-efficient modules for task-level generalisation. In Andreas Vlachos and Isabelle Augenstein (eds.), *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. doi: 10.18653/v1/2023.eacl-main.49. URL https://aclanthology.org/2023.eacl-main.49/.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Weijia Shi, Akshita Bhagia, Kevin Farhat, Niklas Muennighoff, Pete Walsh, Jacob Morrison, Dustin Schwenk, Shayne Longpre, Jake Poznanski, Allyson Ettinger, Daogao Liu, Margaret Li, Dirk Groeneveld, Mike Lewis, Wen tau Yih, Luca Soldaini, Kyle Lo, Noah A. Smith, Luke Zettlemoyer, Pang Wei Koh, Hannaneh Hajishirzi, Ali Farhadi, and Sewon Min. Flexolmo: Open language models for flexible data use, 2025. URL https://arxiv.org/abs/2507.07024.

Dong Geun Shin and Hye Won Chung. Representation norm amplification for out-of-distribution detection in long-tail learning, 2024. URL https://arxiv.org/abs/2408.10676.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, Los Alamitos, CA, USA, May 2017. IEEE Computer Society. doi: 10.1109/SP.2017.41. URL https://doi.ieeecomputersociety.org/10.1109/SP.2017.41.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=67X93aZHII.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024a. URL https://openreview.net/forum?id=F7aAhfitX6.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=NLPzL6HWNl.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iynRvVVAmH.

Weilin Wan, Weizhong Zhang, Quan Zhou, Fan Yi, and Cheng Jin. Out-of-distribution detection using neural activation prior, 2024. URL https://arxiv.org/abs/2402.18162.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5744–5760, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.388. URL https://aclanthology.org/2022.emnlp-main.388/.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Dixi Yao. Risks when sharing lora fine-tuned diffusion model weights, 2024. URL https://arxiv.org/abs/2409.08482.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*, 2024. URL https://openreview.net/forum?id=fq0NaiU8Ex.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=EvDeiLv7qc.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning, 2023. URL https://arxiv.org/abs/2308.03303.

Xin Zhou, Martin Weyssow, Ratnadira Widyasari, Ting Zhang, Junda He, Yunbo Lyu, Jianming Chang, Beiqi Zhang, Dan Huang, and David Lo. Lessleak-bench: A first investigation of data leakage in llms across 83 software engineering benchmarks, 2025. URL https://arxiv.org/abs/2502.06215.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=txRZBD8tBV.

A ARROW PROOF

Proof. We will construct 2x2 LoRA adapters C and D and an input $\mathbf{x}$ that satisfy the condition of the theorem.

1. Define $C = B_1 A_1$:

   Let matrix $C = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$.

   The singular values are $\sigma_1(C) = 2$ and $\sigma_2(C) = 1$. The right singular vector corresponding to σ_1 is $\mathbf{v}_C = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

2. Define $D = B_2 A_2$:

   We construct D from the singular value decomposition $D = USV^T$ and choose the components to satisfy the theorem.

   Let $U = I$ the identity.

   Let the singular values be $\sigma_1(D) = 3$ and $\sigma_2(D) = 1$.

   Let the right singular vectors be $\mathbf{v}_D = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$.

   $D = USV^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}^T = \begin{pmatrix} \frac{3}{\sqrt{2}} & \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$.

3. Choose vector $\mathbf{x}$:

   Let $\mathbf{x} = \mathbf{v}_C = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

4. Verify inequality:

$$\begin{aligned} LHS &= \text{argmax}_i |\mathbf{v}_i^T \mathbf{x}| \\ &= \text{argmax}_i \{|\mathbf{v}_C^T \mathbf{x}|, |\mathbf{v}_D^T \mathbf{x}|\} \\ &= \text{argmax}_i \{1, \frac{1}{\sqrt{2}}\} = \text{(adapter 1)}. \end{aligned}$$

$$\begin{aligned} RHS &= \text{argmax}_i ||B_i A_i \mathbf{x}||_2 \\ &= \text{argmax}_i \{||C\mathbf{x}||_2, ||D\mathbf{x}||_2\} \\ &= \text{argmax}_i \{ \left\| \begin{matrix} 2 \\ 0 \end{matrix} \right\|_2, \left\| \begin{matrix} \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{matrix} \right\|_2 \} \\ &= \text{argmax}_i \{2, \sqrt{5}\} = \text{(adapter 2)}. \end{aligned}$$

$LHS \neq RHS.$

$\square$

B SPECTR PROOF

Proof. Let $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$, and $\mathbf{x} \in \mathbb{R}^n$. So,

$$
\begin{aligned}
||BA\mathbf{x}||_2 &= ||USV^T\mathbf{x}||_2 && \text{(Equation 1)} \\
&= ||U\hat{A}\mathbf{x}||_2 && \text{(Equation 3)} \\
&= \sqrt{||U\hat{A}\mathbf{x}||_2^2} \\
&= \sqrt{\mathbf{x}^T\hat{A}^TU^TU\hat{A}\mathbf{x}} && \text{(Definition of squared 2-norm)} \\
&= \sqrt{\mathbf{x}^T\hat{A}^T\hat{A}\mathbf{x}} && \text{(Orthonormal columns } \implies U^TU = I) \\
&= \sqrt{||\hat{A}\mathbf{x}||_2^2} \\
&= ||\hat{A}\mathbf{x}||_2
\end{aligned}
$$

□

C SEQR PROOF

Proof. Let $B \in \mathbb{R}^{m \times r}$, $A \in \mathbb{R}^{r \times n}$, $\mathbf{x} \in \mathbb{R}^n$, and $B = QR$ from Equation 4. So,

$$
\begin{aligned}
||BA\mathbf{x}||_2 &= ||QRA\mathbf{x}||_2 && \text{(substitution)} \\
&= \sqrt{||QRA\mathbf{x}||_2^2} \\
&= \sqrt{\mathbf{x}^TA^TR^TQ^TQRA\mathbf{x}} \\
&= \sqrt{\mathbf{x}^TA^TR^TRA\mathbf{x}} && \text{(Orthonormal columns } \implies Q^TQ = I) \\
&= \sqrt{||RA\mathbf{x}||_2^2} \\
&= ||RA\mathbf{x}||_2
\end{aligned}
$$

□

D ADAPTER DETAILS

We fit LoRA adapters targeting all attention layers in the network (query, key, value, and output projection layers). We choose initial settings for the LoRAs using the `unsloth` hyperparameter guide.[5] We use rank-32 adapters with a LoRA $\alpha = 64$ and dropout of 0.05. We train for two epochs using a cosine schedule with warm-up ratio of 5% and a batch size of 8. We sweep learning rates in the set {5e-6, 1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3} for each dataset, but share learning rates across random seeds.

[5] https://docs.unsloth.ai/get-started/fine-tuning-llms-guide/lora-hyperparameters-guide