FLEXMORE: A FLEXIBLE MIXTURE OF RANK-HETEROGENEOUS EXPERTS FOR EFFICIENT MODULAR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in mixture-of-experts (MoE) models can compose independently trained, domain-specialized experts around a shared backbone, but storing every expert at full size limits practical deployment. We introduce *FlexMoRE* (Flexible Mixture of Rank-Heterogeneous Experts), a post-hoc approach that replaces dense experts with low-rank updates relative to a shared public base model. Starting from the FlexOlmo expert collection, we evaluate six experts and combined FlexMoRE models with $a2$, $a4$, and $a7$ active experts over 15 ranks, $r \in \{2^0, \ldots, 2^{14}\}$, on 132 tasks from eight benchmarks. Rank–performance relationships are strongly task-dependent: reasoning-oriented benchmarks, particularly BBH, Math2, and Code4, benefit from increased rank, whereas GEN5 and several knowledge-oriented settings show weak or negative rank sensitivity. This variation motivates heterogeneous capacity allocation. Compared with homogeneous low-rank mixtures, heterogeneous FlexMoRE configurations retain stronger aggregate performance at substantially lower stored capacity. For example, the MC9 selected $a2$ configuration uses 10.75B rather than 33.27B stored parameters (a 67.7% reduction) with an original Avg. of 40.75 versus 40.77 for FlexOlmo; performance is slightly lower when MC9 is excluded. FlexMoRE achieves higher aggregate scores than an AdapterSoup baseline, whose performance deteriorates at higher ranks, supporting separately addressable, conditionally routed experts over static averaging in the evaluated setting. FlexMoRE therefore provides a practical post-hoc route to storage-efficient composition of independently trained experts, while highlighting rank allocation as an open problem.

1 INTRODUCTION

Domain-specific training data cannot always be pooled across organizations. FlexOlmo (Shi et al., 2025) addresses this setting by allowing domain experts to be trained independently using a publicly available, frozen *public base model*. Each expert learns alongside this public expert and shared backbone; its expert modules and router embeddings can subsequently be composed with those of other data owners without joint expert training or sharing their training data.

However, this flexibility comes at severe memory costs: sparse activation reduces the number of expert branches evaluated per token, but not the weights stored for the available expert collection. FlexMoRE addresses this per-expert cost by representing previously dense domain experts as post-hoc low-rank updates relative to the shared public base model. We test this approach through post-hoc LoRA extraction of each expert's weight differences against the public base model. The public base model and released domain-informed router are kept unchanged. Crucially, each expert may have a different best-suited rank, and our results will establish the efficacy of such rank-heterogeneous mixture of experts models.

Specifically, we experiment with FlexOlmo (Shi et al., 2025), which is currently the only available model that satisfies our assumptions of having a shared public base model. After post-hoc LoRA extraction akin to (Vasani et al., 2025), we evaluate homogeneous-rank and heterogeneous-rank mixtures across six domain experts and eight benchmarks. We compare against AdapterSoup (Chronopoulou et al., 2023), implemented here as uniform averaging of the expert updates onto the public base model.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Comparing with an AdapterSoup (Chronopoulou et al., 2023) baseline, we find that FlexMoRE benefits from larger ranks, whereas AdapterSoup degrades – likely due to destructive interference under averaging. FlexMoRE achieves higher aggregate scores, particularly at higher ranks and for the selected heterogeneous configurations.

More broadly, the heterogeneous mixtures outperform homogeneous alternatives at lower memory. With 2 active experts, the MC9-selected rank-heterogeneous mixture matches FlexOlmo on the full average over all downstream tasks, with only small degradation when MC9 is excluded (-0.85 percentage points).

Our rank analysis shows that observed peak ranks vary across experts and benchmarks. Under Avg.-based selection with $r \leq 2^{11}$, Academic, Math, Code, and Reddit select higher ranks than Creative Writing and News; these selections identify observed score maxima, not minimum capacity requirements.

Our contributions are:

- We introduce FlexMoRE (Figure 1), a method that greatly reduces memory requirements of modular mixture of experts architectures like FlexOlmo.
- Testing our method on FlexOlmo, we show empirically that FlexMoRE yields comparable downstream performance at only one third of FlexOlmo's parameters.
- In a systematic rank-sensitivity analysis, we show that rank-heterogenous (i.e., having a different rank per domain) are preferable over rank-homogeneous mixtures.

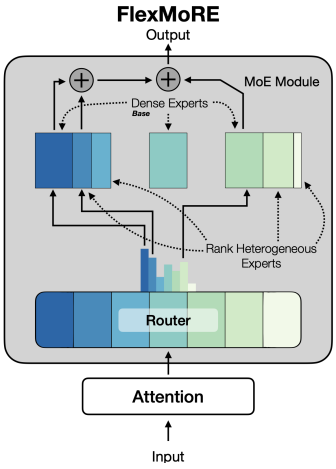

Figure 1: FlexMoRE overview

2 Related Work

Independent expert composition. MoE models use input-dependent routing to activate a subset of expert FFNs (Fedus et al., 2022; Mu & Lin, 2025). FlexOlmo (Shi et al., 2025) extends this setting to independently trained experts anchored to a frozen public base model, with expert opt-in and opt-out at composition time. Federated learning also addresses distributed data access (Abishek et al., 2025; Yi et al., 2024), but is not the training protocol evaluated here: we reuse the released FlexOlmo experts and study their post-hoc compression.

Low-rank adaptation and extraction. LoRA learns low-rank weight updates while freezing the underlying model (Hu et al., 2022). Post-hoc extraction instead approximates the difference between an already-trained model and its reference weights, as in PHLoRA (Vasani et al., 2025). FlexMoRE uses this existing factorization principle; our contribution is its application to independently trained expert composition with expert-specific ranks, together with an empirical study of the resulting trade-offs.

Routed and static expert combination. Adapter mixtures and MoE–LoRA methods combine lightweight modules through routing or learned mixing (Wang et al., 2022; Cao et al., 2025; Li et al., 2024). Our setting retains experts and router components obtained through FlexOlmo's independent training procedure rather than jointly training a new adapter pool. Static parameter merging offers an alternative to conditional expert execution, but interactions among independently learned updates can affect the merged model (Yadav et al., 2023; Gargiulo et al., 2025). We therefore compare FlexMoRE with uniform averaging of its expert updates akin to AdapterSoup (Chronopoulou et al., 2023).

3 Methodology

3.1 Preliminaries

The *public base model* is the complete publicly available pretrained model. Its non-expert components, including attention and embeddings, form the *shared backbone*; its FFN in each routed layer is the *public expert*. A *domain expert* denotes the corresponding domain-specific FFN branch at each layer.

In FlexOlmo's original training procedure, each data owner constructs a two-expert model from the public base model. The public expert, shared backbone, and public router embedding remain frozen, while the domain expert's FFNs and domain router embedding are trained on that owner's data (Shi et al., 2025). The independently trained domain experts and their router embeddings can then be composed without joint expert training. Optional post-composition router tuning is separate from this procedure. FlexMoRE uses the released checkpoints to vary expert ranks and reduce parameter storage post hoc, without requiring expert retraining or additional router tuning.

3.2 Low-rank Expert Extraction

Let $W_{\text{pub}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ be an FFN weight matrix from the public base model and W_i the corresponding matrix from dense domain expert i, with layer and projection indices omitted. Following post-hoc adapter extraction (Vasani et al., 2025), we approximate their difference by truncated SVD:

$$\Delta_i = W_i - W_{\text{pub}} = U_i \Sigma_i V_i^\top, \qquad \widetilde{\Delta}_i^{(r_i)} = U_{i,r_i} \Sigma_{i,r_i} V_{i,r_i}^\top. \tag{1}$$

The truncated factors retain the leading singular components. Defining

$$B_i = U_{i,r_i} \Sigma_{i,r_i}^{1/2}, \qquad A_i = \Sigma_{i,r_i}^{1/2} V_{i,r_i}^\top, \qquad \widetilde{W}_i^{(r_i)} = W_{\text{pub}} + B_i A_i \tag{2}$$

gives $B_i \in \mathbb{R}^{d_{\text{out}} \times r_i}$ and $A_i \in \mathbb{R}^{r_i \times d_{\text{in}}}$, for $r_i \leq \min(d_{\text{out}}, d_{\text{in}})$. We apply this factorization to the domain-expert FFN projections, sharing the public reference weights across experts and retaining expert-specific low-rank factors. This requires no adapter training. At sufficiently low ranks, the factorized representation reduces the memory required for model weights at a given numerical precision; it does not reduce the original expert-training cost.

3.3 Rank-heterogeneous composition and routing

A *homogeneous* mixture assigns the same rank to all low-rank domain experts; a *heterogeneous* mixture allows expert-specific ranks. Our evaluated configuration uses the public base model as the common reference for all domain experts.

For each token and layer, the retained FlexOlmo router scores the public and domain experts and selects the top k. Let x denote the token's hidden representation, $S_k(x)$ the selected set of expert indices, and $g_j(x)$ the routing weight used by the inherited implementation. With index 0 denoting the public expert, the routed output is

$$y(x) = \sum_{j \in S_k(x)} g_j(x)\, \widetilde{M}_j(x), \qquad \widetilde{M}_0 = M_{\text{pub}}, \tag{3}$$

where $\widetilde{M}_i$ for $i > 0$ is the domain-expert FFN with computed from Equation 2.

Each selected low-rank expert requires the public reference weights: a projection with input h computes $W_{\text{pub}} h + B_i(A_i h)$. This is distinct from selecting the public expert's complete FFN as a separate term in the routed output.

3.4 Experimental Setup

Expert collection and number of active experts. We use the six released FlexOlmo domain experts: Code, Creative Writing, Math, News, Academic, and Reddit. The Educational Text expert is not available in the collection used here. Together with the public expert, this gives seven available branches per routed layer. We denote the composed configuration as $7 \times$ 7B and evaluate $a2$, $a4$, and $a7$, meaning $k = 2, 4, 7$ selected branches per token and layer. This notation does not imply seven separately stored complete 7B models: non-expert parameters are shared. The public expert is included in the count of seven, not added on top of k; all seven branches are selected for $a7$.

Individual experts. Each individual low-rank domain expert is evaluated alongside the public expert in a two-expert ($2 \times$ 7B) configuration. Its *dense expert baseline*, labelled **Base**line in the main expert table, uses the corresponding uncompressed domain expert in the same two-expert configuration, not the domain expert in isolation. For composed models, the *FlexOlmo baseline*

contains the same public and six dense domain experts with matching $a2$, $a4$, or $a7$. The standalone *public base model*, labelled **Public** base model in the main expert table, is a separate reference. In figures spanning individual and composed models, *dense baseline* denotes the appropriate matched reference; it describes expert weights, not whether routing is sparse.

In the result tables, *2x7B Expert* denotes the public-plus-domain expert configuration, while *7x7B Flex* denotes the complete composition; *Olmo* and *MoRE* abbreviate FlexOlmo and FlexMoRE. For individual experts, *Base* denotes the matched dense expert baseline (**Base**line), whereas *Public* denotes the standalone **Public** base model. These are distinct references for the relative-change columns.

Rank selection. We evaluate configured ranks $r \in \{2^0, \ldots, 2^{14}\}$ for rank-sensitivity analysis. Selected-model summaries restrict candidates to $r \leq 2^{11}$, the storage-efficient range for the complete expert collection. Homogeneous mixtures use one common rank. For heterogeneous mixtures, we select each expert's rank using its individual two-expert evaluation score on either MC9 (*MC9-selected*) or the unweighted macro average across all eight benchmarks (*All-selected*). The selected tuples are reported in Table 5 and reused across $a2$, $a4$, and $a7$. MC9 selection is a simple single-benchmark heuristic, not a validated predictor of performance across all benchmarks. Neither criterion optimizes ranks jointly in the composed model.

Evaluated heterogeneous compositions. In the expert order (Code, Creative Writing, Math, News, Academic, Reddit), the evaluated rank allocations are

$$\mathbf{r}_{\mathrm{MC9}} = (2^6,\ 2^7,\ 2^{11},\ 2^0,\ 2^3,\ 2^7),$$
$$\mathbf{r}_{\mathrm{All}} = (2^9,\ 2^4,\ 2^{11},\ 2^6,\ 2^{11},\ 2^9).$$

Both compositions retain the same uncompressed public base model and all six domain experts, represented by low-rank factors at the listed ranks. Their total stored parameter counts are 10.75B and 14.84B, respectively, including the public base model. Each allocation is evaluated unchanged at $a2$, $a4$, and $a7$: only the number of selected branches changes, not the available expert collection or stored ranks.

Benchmarks. We follow the FlexOlmo-aligned OLMES evaluation suite, comprising 132 tasks organized into eight benchmarks. **General-purpose evaluation** includes: **MC9**, a collection of nine multiple-choice reasoning benchmarks (ARC-Easy, ARC-Challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), CommonsenseQA (CSQA) (Reddy et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2019), SocialIQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2019)); **GEN5**, consisting of five generative question answering tasks, CoQA (Reddy et al., 2019), SQuAD (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and DROP (Dua et al., 2019)); **AGIEval** with college-level academic reasoning tasks (Zhong et al., 2023); and **BBH** with challenging multi-step reasoning tasks from BIG-Bench (Suzgun et al., 2022).

Domain-specific tasks include **MMLU** (Hendrycks et al., 2021a) and **MMLU-Pro** (Wang et al., 2024), assessing broad academic knowledge across multiple disciplines; **Math2** composed of GSM8K (Cobbe et al., 2021) and MATH, 7 Minerva math tasks (Hendrycks et al., 2021b). **Code4** composed of 4 coding benchmarks composed of MBPP (Austin et al., 2021), MBPPPLUS (Liu et al., 2023), HumanEval (Chen et al., 2021), and HumanEvalPlus (Liu et al., 2023).

Measures. Let s_g be the aggregate score for benchmark g. We report the unweighted macro average $\mathrm{Avg.} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} s_g$, with equal weighting of the eight benchmarks (*Avg. w/ MC9*) or the remaining seven (*Avg. w/o MC9*). Within-benchmark aggregation follows the evaluation protocol; in particular, Math2 averages GSM8K and the mean of the Minerva subjects. Excluding MC9 separates the reported aggregate from the selection benchmark for MC9-selected configurations. This is important because we considered MC9 as a validation set to determine the optimal rank per domain as measured through single expert performance. Therefore, Avg. w/o MC9 should be considered our main test metric.

Relative score change is $100(s_{\mathrm{model}}/s_{\mathrm{baseline}} - 1)$, using the same averaging rule for both scores. Percentage-point differences instead equal $100(s_{\mathrm{model}} - s_{\mathrm{baseline}})$ for scores on the 0–1 scale.

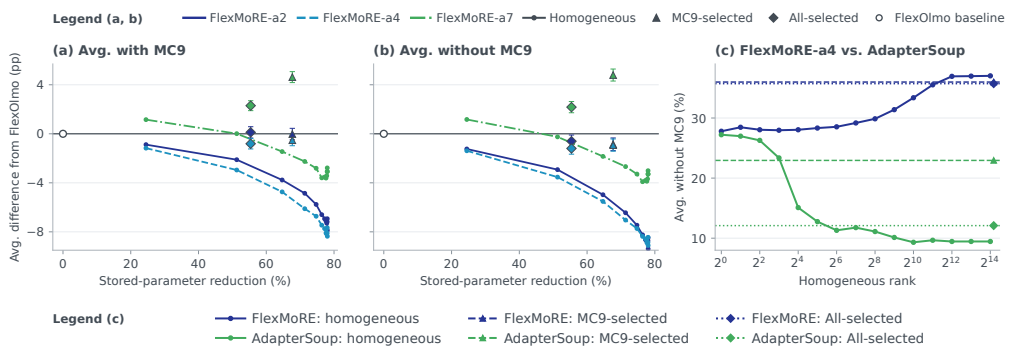

Figure 2: **Performance, parameter storage, and AdapterSoup. (a, b)** Unweighted macro average (Avg.) difference from the matched FlexOlmo baseline, in percentage points, versus stored-parameter reduction, with **(a)** and without **(b)** MC9. Curves show homogeneous ranks; triangles and diamonds denote MC9-selected and All-selected mixtures. Error bars are paired-bootstrap normal approximation 95% confidence intervals. Only configurations with nonnegative parameter savings are shown; savings concern model weights, not total inference memory. **(c)** Avg. without MC9 for FlexMoRE-$a4$ versus uniform averaging of the same expert updates (Static AdapterSoup). Curves show homogeneous rank sweeps; dashed and dotted horizontal lines denote MC9-selected and All-selected configurations, respectively.

Parameter savings refer to total stored parameters relative to the matched baseline, not only activated parameters or total inference memory.

Baselines. Our primary baseline is the FlexOlmo model itself with full-rank experts. Additionally, we compare FlexMoRE with an AdapterSoup Chronopoulou et al. (2023) baseline, formed by uniformly averaging expert updates onto the public reference weights.

4 Results

Performance relative to FlexOlmo. Table 1 and Figure 2 **(a, b)** show that the outcome depends on both activation setting and evaluation summary. The MC9-selected heterogeneous $a2$ model stores 10.75B rather than 33.27B parameters, a 67.7% reduction, with Avg. w/ MC9 of 40.75 versus 40.77. Its paired difference is -0.02 percentage points (95% interval $[-0.49, +0.45]$), consistent with small losses or gains rather than establishing equivalence. Excluding MC9 gives 36.80 versus 37.65 and a difference of -0.85 points ($[-1.37, -0.34]$). The All-selected $a2$ model likewise has a near-equal original Avg. but a lower Avg. without MC9. For $a4$, both selected heterogeneous configurations are below FlexOlmo under both averaging rules; the corresponding paired intervals are negative (Table 14).

For $a7$, the MC9-selected configuration improves over FlexOlmo under both summaries: $+4.63$ points including MC9 ($[+4.18, +5.07]$) and $+4.79$ excluding MC9 ($[+4.30, +5.28]$). The All-selected and homogeneous $a7$ configurations also improve over their matched FlexOlmo baseline. These comparisons support a setting-dependent storage–performance trade-off, not uniform equivalence to or improvement over FlexOlmo.

Heterogeneous versus homogeneous allocation. The MC9-selected mixtures store 10.75B parameters compared with 25.12B for homogeneous $r = 2^{11}$, while attaining higher Avg. with and without MC9 for all three activation settings. On Avg. w/ MC9, their paired gains over this homogeneous reference are $+0.86$, $+0.65$, and $+3.47$ points for $a2$, $a4$, and $a7$, respectively, with positive intervals (Table 14). Without MC9, the $a2$ interval includes zero and the rounded $a4$ lower bound is near zero, while the $a7$ gain remains clearly positive. All-selected mixtures also have higher point estimates than this homogeneous reference at lower storage (14.84B). This supports the utility of the tested heterogeneous allocations, but is not a comparison at equal parameter budgets or proof of optimal rank allocation.

Table 1: **Selected FlexMoRE configurations.** Avg. w/ MC9 is the original unweighted macro average across the eight benchmarks; Avg. w/o MC9 excludes MC9 and averages the remaining seven benchmarks (parameter reduction vs. performance shown in Figure 2). Both $\Delta(\%)$ columns are relative to the corresponding baseline under the same averaging rule. The highest score in each column is shown in **bold**.

7x7B Flex	Params	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg. w/ MC9	Δ(%)	Avg. w/o MC9	Δ(%)
Olmo-a2 baseline	33.27B	0.6257	0.4508	0.3842	**0.4469**	0.5297	0.2415	**0.4632**	0.1193	0.4077	–	**0.3765**	–
MoRE-a2 heterogen. MC9	10.75B	0.6838	0.4868	**0.4184**	0.4269	0.5555	0.2543	0.4193	0.0145	0.4075	-0.05	0.3680	-2.27
MoRE-a2 heterogen. All	14.84B	0.6776	0.4780	0.4169	0.4195	0.5475	0.2537	0.4209	0.0568	**0.4089**	+0.29	0.3705	-1.61
MoRE-a2 homogen. 2^{11}	25.12B	0.6417	0.4664	0.3724	0.4249	0.5358	0.2443	0.4119	0.0933	0.3988	-2.17	0.3641	-3.29
Olmo-a4 baseline	33.27B	0.6709	0.4548	0.3999	0.4014	0.5518	0.2486	0.3788	**0.1482**	0.4068	–	0.3691	–
MoRE-a4 heterogen. MC9	10.75B	**0.6936**	**0.4896**	0.4140	0.4255	0.5516	**0.2564**	0.3565	0.0252	0.4016	-1.29	0.3599	-2.50
MoRE-a4 heterogen. All	14.84B	0.6896	0.4846	0.4121	0.4124	**0.5557**	0.2533	0.3210	0.0609	0.3987	-2.00	0.3571	-3.24
MoRE-a4 homogen. 2^{11}	25.12B	0.6742	0.4526	0.4042	0.4036	0.5496	0.2518	0.3097	0.1152	0.3951	-2.88	0.3552	-3.75
Olmo-a7 baseline	33.27B	0.6550	0.3614	0.3745	0.3560	0.5097	0.2212	0.2526	0.1069	0.3547	–	0.3117	–
MoRE-a7 heterogen. MC9	10.75B	0.6894	**0.4896**	0.4140	0.4255	0.5516	**0.2564**	0.3565	0.0241	0.4009	+13.04	0.3597	+15.38
MoRE-a7 heterogen. All	14.84B	0.6862	0.4834	0.4042	0.3861	0.5398	0.2420	0.2356	0.0435	0.3776	+6.47	0.3335	+6.98
MoRE-a7 homogen. 2^{11}	25.12B	0.6656	0.4678	0.3897	0.3618	0.5209	0.2248	0.2143	0.0847	0.3662	+3.25	0.3234	+3.75

Table 2: **Comparison between the original FlexMoRE-a4 models and the corresponding AdapterSoup outputs.** Deltas are percentage-point differences relative to the original FlexOlmo-a4 baseline. The soup applies a uniform mean merge of the expert updates. Best column score in **bold**.

		MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg. w/ MC9	Δ (pp)	Avg. w/o MC9	Δ (pp)
FlexMoRE	FlexOlmo-a4 (baseline)	0.6709	0.4548	0.3999	0.4014	0.5518	0.2486	**0.3788**	**0.1482**	**0.4068**	+0.00	**0.3691**	+0.00
FlexMoRE	FlexMoRE-a4 homogen. ($r = 2^0$)	0.6397	0.4844	0.3653	0.3226	0.5170	0.2164	0.0382	0.0018	0.3232	-8.36	0.2780	-9.11
FlexMoRE	FlexMoRE-a4 homogen. ($r = 2^2$)	0.6431	**0.4930**	0.3775	0.3153	0.5205	0.2152	0.0409	0.0008	0.3258	-8.10	0.2805	-8.86
FlexMoRE	FlexMoRE-a4 homogen. ($r = 2^7$)	0.6738	0.4862	0.3881	0.3526	0.5248	0.2212	0.0670	0.0025	0.3395	-6.73	0.2918	-7.73
FlexMoRE	FlexMoRE-a4 homogen. ($r = 2^{11}$)	0.6742	0.4526	0.4042	0.4036	0.5496	0.2518	0.3097	0.1152	0.3951	-1.17	0.3552	-1.39
FlexMoRE	FlexMoRE-a4 heterogen. (MC9)	**0.6936**	0.4896	**0.4140**	**0.4255**	0.5516	**0.2564**	0.3565	0.0252	0.4016	-0.52	0.3599	-0.92
FlexMoRE	FlexMoRE-a4 heterogen. (All)	0.6896	0.4846	0.4121	0.4124	**0.5557**	0.2533	0.3210	0.0609	0.3987	-0.81	0.3571	-1.19
AdapterSoup	FlexOlmo-a4 (baseline)	0.3478	0.0000	0.2270	0.0685	0.2476	0.1163	0.0015	0.0000	0.1261	-28.07	0.0944	-27.47
AdapterSoup	FlexMoRE-a4 homogen. ($r = 2^0$)	0.6170	0.4684	0.3537	0.3219	0.4994	0.2073	0.0481	0.0065	0.3153	-9.15	0.2722	-9.69
AdapterSoup	FlexMoRE-a4 homogen. ($r = 2^2$)	0.5999	0.4698	0.3414	0.3152	0.4798	0.1876	0.0422	0.0033	0.3049	-10.19	0.2628	-10.63
AdapterSoup	FlexMoRE-a4 homogen. ($r = 2^7$)	0.3570	0.0774	0.2078	0.1634	0.2481	0.1136	0.0130	0.0000	0.1475	-25.93	0.1176	-25.15
AdapterSoup	FlexMoRE-a4 homogen. ($r = 2^{11}$)	0.3542	0.0020	0.2181	0.0929	0.2437	0.1143	0.0050	0.0000	0.1288	-27.80	0.0966	-27.25
AdapterSoup	FlexMoRE-a4 heterogen. (MC9)	0.5394	0.4012	0.3016	0.2921	0.4071	0.1575	0.0433	0.0032	0.2682	-13.86	0.2294	-13.96
AdapterSoup	FlexMoRE-a4 heterogen. (All)	0.3333	0.0978	0.2214	0.1494	0.2516	0.1143	0.0120	0.0001	0.1475	-25.93	0.1209	-24.81

Averages, benchmark differences, and uncertainty. Evaluation-sample uncertainty is estimated using 5,000 within-task bootstrap replicates, with paired resampling for model differences (Appendix C). These intervals condition on the evaluated checkpoints, prompts, and selected configurations; they do not include training-seed or rank-selection variability. Excluding MC9 reveals that gains on the selection benchmark can offset losses elsewhere; it is therefore important alongside the original Avg. For All selection, both summaries still contain benchmarks used for rank selection. Small aggregate gaps also conceal larger domain-specific differences: for MC9-selected $a2$, MC9 increases from 62.57 to 68.38, whereas Code4 falls from 11.93 to 1.45 and Math2 from 46.32 to 41.93. The loss is therefore not uniformly small across benchmarks. Table 11 reports benchmark-level scores and example-level intervals, while Figure 5 resolves changes within benchmarks. The 5,000-replicate paired intervals quantify evaluation-sample uncertainty for fixed checkpoints, prompts, and allocations; they do not measure training variability or correct for configuration selection (Appendix C).

AdapterSoup Comparison Figure 2 **(c)** compares FlexMoRE-$a4$ with uniform averaging of the same expert updates onto the public reference weights. On Avg. without MC9, the gap is modest at $r = 2^2$ (28.05 for FlexMoRE versus 26.28 for AdapterSoup), but much larger at $r = 2^{11}$ (35.52 versus 9.66). The MC9-selected heterogeneous configuration likewise scores 35.99 under routing versus 22.94 under static averaging (Table 9).

Thus, the tested static averaging procedure is not an adequate substitute for the complete routed system. Because the control averages rather than sums updates, it avoids a simple unscaled-sum construction. Nevertheless, the experiment does not isolate routing, rank heterogeneity, or interference as the causal explanation. Benchmark- and task-level comparisons are provided in Figures 7 and 8, with both averaging rules in Table 2.

Table 3: **Selected low-rank domain experts.** Ranks are selected by Avg. w/ MC9 among $r \leq 2^{11}$. Each domain expert is evaluated alongside the public expert; dense expert baselines use the same two-expert configuration. Avg. w/ MC9 is the unweighted macro average across eight benchmarks; Avg. w/o MC9 averages the remaining seven. Base $\Delta(\%)$ compares against the matched dense expert baseline; Public $\Delta(\%)$ compares against the standalone public base model, using the corresponding averaging rule. The highest score in each evaluation-score column is shown in **bold**.

2x7B Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg. w/ MC9	Base $\Delta(\%)$	Public $\Delta(\%)$	Avg. w/o MC9	Base $\Delta(\%)$	Public $\Delta(\%)$
Public base model	7.30B	–	0.6872	**0.5108**	0.3994	0.3570	0.5587	0.2612	0.0816	0.0091	0.3581	–	–	0.3111	–	–
Code **Base**line	11.63B	4.33B	0.6757	0.4668	0.3909	0.3831	0.5381	0.2443	0.0831	**0.1747**	0.3696	–	+3.20	0.3259	–	+4.74
Code 2^{11}	10.27B	2.97B	0.6779	0.4862	0.3903	0.3776	0.5388	0.2428	0.0909	0.1363	0.3676	-0.54	+2.64	0.3233	-0.79	+3.91
Creative **Base**line	11.63B	4.33B	0.6709	0.4998	0.3965	0.3424	0.5378	0.2478	0.0739	0.0097	0.3474	–	-3.01	0.3011	–	-3.21
Creative 2^{3}	7.31B	11.60M	0.6834	0.5064	0.3941	0.3569	0.5587	0.2582	0.0967	0.0120	0.3583	+3.15	+0.05	0.3119	+3.56	+0.24
Math **Base**line	11.63B	4.33B	0.6836	0.4862	**0.4138**	**0.4589**	**0.5648**	0.2615	**0.4964**	0.1262	**0.4364**	–	+21.86	**0.4011**	–	+28.92
Math 2^{11}	10.27B	2.97B	0.6832	0.4954	0.4072	0.4289	0.5557	0.2598	0.4388	0.1026	0.4215	-3.43	+17.68	0.3841	-4.25	+23.44
News **Base**line	11.63B	4.33B	0.6602	0.5058	0.3691	0.3554	0.5387	0.2446	0.0765	0.0056	0.3445	–	-3.81	0.2994	–	-3.77
News 2^{0}	7.30B	1.45M	**0.6910**	0.4998	0.4032	0.3593	0.5589	**0.2636**	0.0993	0.0116	0.3608	+4.74	+0.76	0.3137	+4.77	+0.82
Academic **Base**line	11.63B	4.33B	0.6732	0.5000	0.3944	0.3622	0.5479	0.2422	0.0857	0.0030	0.3511	–	-1.98	0.3050	–	-1.96
Academic 2^{11}	10.27B	2.97B	0.6710	0.5104	0.3928	0.3653	0.5486	0.2515	0.0797	0.0025	0.3527	+0.48	-1.51	0.3073	+0.73	-1.24
Reddit **Base**line	11.63B	4.33B	0.6171	0.4108	0.3668	0.3668	0.5358	0.2350	0.1000	0.0370	0.3337	–	-6.83	0.2932	–	-5.77
Reddit 2^{10}	8.78B	1.48B	0.6620	0.4946	0.3922	0.3521	0.5474	0.2449	0.0934	0.0322	0.3523	+5.60	-1.62	0.3081	+5.09	-0.97

Measured Memory and Inference Time We measure resident GPU memory, time to first token, and generation throughput in the tested Hugging Face Transformers backend (Appendix A). The storage savings translate into lower resident GPU memory in the tested Hugging Face Transformers eager backend (Table 4). The MC9-selected configuration uses 20.94 rather than 62.74 GiB, a 66.6% reduction; homogeneous $r = 2^7$ uses 16.55 GiB, a 73.6% reduction. However, generation throughput is only $0.88\times$ the FlexOlmo baseline for the MC9-selected configuration and 0.82–$0.83\times$ for homogeneous $r = 2^7$. Time to first token also increases, from approximately 76–79 ms to 99–103 ms. These measurements demonstrate a memory–throughput trade-off, not a general inference-speed improvement. Appendix A specifies hardware, precision, batch sizes, and generation settings; the homogeneous $r = 2^7$ measurement is distinct from the $r = 2^{11}$ performance reference above.

5 Rank Sensitivity Analysis

We seek to understand the relationship between expert rank and task performance. To quantify this, we estimate rank sensitivity using a regression between $\log_2$ of the expert rank and task performance. For each model family and evaluation group, we fit $s(r) = \alpha + \beta \log_2 r$, where $s(r)$ denotes the evaluation score at rank r while β measures sensitivity to rank increases. Positive values of β indicate that increasing rank consistently improves performance, while values near zero or negative indicate diminishing returns. The slope analysis fits score against $\log_2 r$ over all 15 ranks through 2^{14}. The main peak-rank heatmap likewise searches the full sweep, while its score-difference panels use the best score under $r \leq 2^{11}$.

Table 3 compares Avg.-selected low-rank domain experts with their dense expert baselines in the matched two-expert setting. Creative Writing, News, Academic, and Reddit have higher Avg. w/ MC9 at their selected ranks, whereas Code and Math remain below their dense expert baselines. The paired comparisons distinguish these point estimates from uncertainty: Creative Writing, News, and Reddit have positive intervals under both averaging rules; the Code and Academic intervals include zero, while Math remains lower (Table 15). These are conditional comparisons of the selected configurations, not selection-adjusted significance tests.

The low selected ranks for Creative Writing ($r = 2^3$) and News ($r = 2^0$), compared with $r = 2^{11}$ for Math and Code, illustrate that no single rank is best for all experts under this selection criterion, motivating heterogeneous allocation. However, an individual expert's aggregate score does not determine its optimal rank in a composition. Full score curves and task-level differences are reported in Figures 9 and 10; benchmark-level uncertainty is given in Table 13.

Figure 3 **(c)** reports slopes in percentage points per rank doubling. Reasoning-oriented benchmarks BBH, Math2, and Code4 have positive fitted slopes for all three homogeneous FlexMoRE sweeps, with the strongest in Math2 (up to 3.66 points per doubling). GEN5 has negative slopes throughout. The pooled expert column summarizes six distinct sweeps rather than a single expert (Table 23).

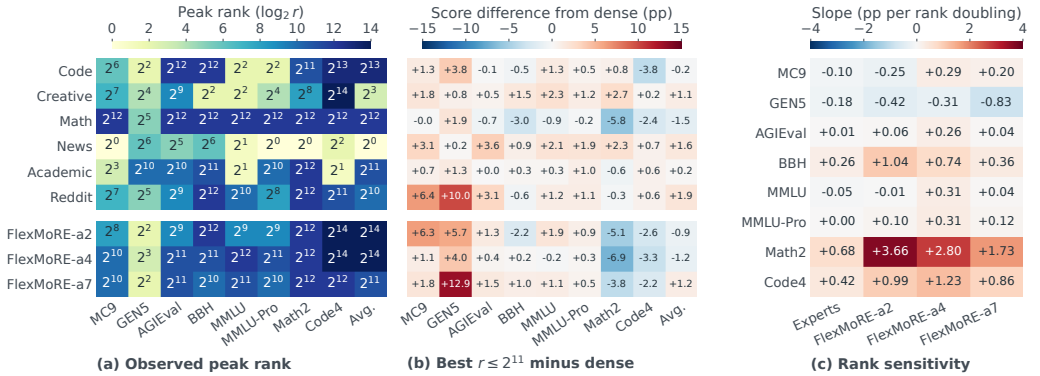

Figure 3: **Peak ranks, low-rank recovery, and rank sensitivity.** In **(a, b)**, top rows show individual experts; bottom rows show FlexMoRE compositions. **(a)** Observed best rank over $r = 2^0, \ldots, 2^{14}$, with ties resolved toward the lower rank. **(b)** Best observed score under $r \leq 2^{11}$ minus the matched dense baseline, in percentage points; positive values indicate improvement. For individual experts, the baseline is the dense domain expert evaluated alongside the public expert; for compositions, it is the corresponding FlexOlmo-$a2$, -$a4$, or -$a7$. Here, *dense* describes expert weights, not activation of all experts. Avg. is the unweighted macro average including MC9. Panels (a, b) report post-hoc maxima, not held-out selection results. **(c)** Linear-regression slopes of score against $\log_2 r$ over the full sweep, in percentage points per rank doubling. Signs indicate fitted trends, not necessarily monotonic changes. Experts pools the six individual expert sweeps; FlexMoRE columns represent homogeneous-rank sweeps for each activation setting.

To identify the rank at which performance peaks, we further define the **typical peak rank** for expert e and evaluation group g as $r^*_{e,g} = \arg\max_{r \in \{2^0, \ldots, 2^{14}\}} s_{e,g}(r)$, computed from the observed scores without regression, smoothing, or normalization (ties broken by lowest rank). The median expert peak is $\log_2 r = 10.5$ for Avg. w/ MC9, compared with 2 for MMLU and 11.5 for BBH, Math2, and Code4. Figure 3 **(a, b)** shows the expert-specific peaks and, separately, the best score under $r \leq 2^{11}$ relative to the matched dense baseline. The appendix compares median peaks under both rank caps (Figure 11; Table 26).

The later peaks and positive slopes in Math2 and Code4 are consistent with stronger sensitivity to retained rank in these evaluations. They help contextualize the residual losses under compression, but do not establish reduced capacity as their sole cause: routing is inherited rather than re-tuned, and effects differ across experts and tasks. Conversely, favorable scores at much lower ranks in other settings motivate heterogeneous rather than uniform allocation. These descriptive post-hoc analyses do not themselves provide a held-out selection rule; per-expert regressions and task-level differences are reported in Appendix E and Figures 10 and 5.

6 Discussion

Our **key findings** are that (i) our proposed FlexMoRE approach achieves downstream performance that competitive to FlexOlmo at only third of effective parameters. (ii) rank-heterogeneous mixtures outperform homogeneous oens, and that (iii) the best-performing rank differs per domain with math and code domains benefiting from higher ranks, whereas News and Creative writing achieve peak downstream performance at lower ranks.

Is rank selection circular? While we have experimented with rank selection based on single-expert performance across all tasks, the final proposed mixture relies on the ranks determined from the MC9 subset. To fairly evaluate this selection strategy, we then report the Avg. across all benchmark tasks *without* MC9, as this was used for rank selection. We still report numbers for "All-selected", as we deem it noteworthy that the MC9 selection outperforms the All selection at $a = 4$ and $a = 7$ active experts. This can be explained as both rank selection strategies are conducted on the expert + base model setting, and not on the full mixtures.

Budget matching in rank-heterogeneous mixtures It is difficult to achieve fully comparable effective parameter budgets across all comparisons between heterogeneous vs. homogeneous. However, at an exemplary budget-matched comparison of a homogeneous mixture of rank 512 (11.75B effective parameters) against a heterogeneous mixture (both with 2 active experts), shows that the heterogeneous mixture attains an Avg. of 40.75 against 37.00.

The $a2$ performance measured by Avg. without MC is lower than the global Avg. with substantial Code4 losses. The $a4$ also loses aggregate performance, whereas $a7$ improves under both summaries. Paired intervals clarify these direct differences, but do not turn a small aggregate gap into evidence of uniform benchmark-level retention. Similarly, the HF measurements establish lower resident memory alongside lower throughput. The demonstrated benefit is reduced stored capacity and, in the tested backend, resident GPU memory, rather than a universal quality or speed improvement.

Our AdapterSoup-style expert averaging highlights a further practical distinction: preserving separately addressable expert branches achieves higher average scores, with substantially larger gaps at higher ranks. Together with the rank-dependent results, this motivates studying compression and heterogeneous expert composition jointly.

Practically, FlexMoRE enables the post-hoc adaptation of expert compositions relative to accelerator-memory budgets, providing maximum flexibility and cooperation between entities, particularly when the optimal composition depends on the specific set of used experts. By retaining FlexOlmo's independently trained experts as separately addressable components and allowing their retained ranks to differ without requiring retraining or additional router tuning, FlexMoRE makes independently contributed experts more accessible on memory-constrained hardware. These rank-dependent results motivate the development of future selection methods that account jointly for the expert composition, target tasks, and memory budget, rather than selecting each expert's rank in isolation. Our empirical selection rules provide a foundational starting point, rather than an optimal allocation.

Rank selection and capacity-sensitive benchmarks. MC9 and All are empirical criteria, not optimal joint rank-allocation or expert-combination methods. Avg. without MC9 excludes the MC9 selection benchmark but still overlaps with All's selection benchmarks. Preliminary effective-rank heuristics (Roy & Vetterli, 2007), with varying thresholds and implementations, yielded no reliable performance-preserving rule. Math2 and Code4 are particularly rank-sensitive in our sweeps, and aggregate retention can mask substantial losses on both. Whether different allocations or router adaptation recover these losses at the same storage budget remains untested.

7 Conclusion

FlexMoRE represents independently trained FlexOlmo domain experts as post-hoc low-rank updates relative to the public base model, retaining the released router without additional training. The evaluated heterogeneous mixtures provide stronger aggregate performance than the selected homogeneous reference at lower storage, while comparisons with FlexOlmo depend on activation setting and benchmark aggregation. Rank sensitivity, paired uncertainty, and the AdapterSoup control make these trade-offs explicit: capacity needs differ, small averages can conceal larger domain losses, and static averaging is not an equivalent substitute for routed composition.

Limitations This work comes with several limitations: First, experiments are limited to a single starting point model, FlexOlmo. This is because FlexOlmo is the only architecture that satisfies the requirement of having a public base model against which other experts are routed. Second, all our experiments rely on post-hoc LoRA extraction from existing experts. Training new experts from scratch with LoRA fine-tuning is beyond the scope of this work. Third, the method relies on empirical search over candidate ranks using a set of downstream tasks. We have therefore reported average scores excluding this set of downstream tasks. Lastly, despite memory efficiency gains, FlexMoRE's has less inference throughput (0.82-0.88x) due to the implementation not being tuned for inference performance.

Future work Future work should investigate held-out, budget-aware rank allocation, directly trained low-rank experts, and transfer across public base models and expert collections. Our findings regarding the effectiveness of rank-heterogeneity opens up new avenues of future work of how to best allocate capacity across domains.

AI USE STATEMENT

We used generative AI tools to provide feedback on research methodology, writing and experiments, assist with code, prepare figures and tables, and discuss the presentation of experimental results. The authors take responsibility for this work, including AI-assisted code and other artifacts.

ETHICS STATEMENT

FlexMoRE inherits FlexOlmo's support for expert composition without pooling the domain training data, a setting relevant to privacy, confidentiality, and data-governance constraints. Our experiments reuse released checkpoints and do not demonstrate a new privacy guarantee: avoiding raw-data sharing does not by itself prevent information leakage through model weights or outputs. Reduced parameter storage and measured GPU memory may lower the memory requirements of deployment, but we do not establish reduced total energy use or faster inference. Biases and other limitations of the public base model and domain experts may persist after compression.

REPRODUCIBILITY STATEMENT

Our experiments use publicly released FlexOlmo checkpoints. Section 3 describes the post-hoc low-rank extraction and expert composition, while Section 3.4 specifies the evaluated experts, rank allocations, benchmarks, and scoring conventions. Appendices B, D, and E provide detailed results across ranks and benchmarks. Appendix C describes the bootstrap procedure, and Appendix A reports the hardware and inference settings used for memory and timing measurements. Code is available at `https://anonymous.4open.science/r/FlexMoRE-5FB3/README.md`.

REFERENCES

Alok Abishek, Lisa Erickson, and Tushar Bandopadhyay. Data and ai governance: Promoting equity, ethics, and fairness in large language models. *MIT Science Policy Review*, 6:139–146, August 2025. doi: 10.38105/spr.1sn574k4lp. URL `http://dx.doi.org/10.38105/spr.1sn574k4lp`.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL `https://arxiv.org/abs/2108.07732`.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL `https://arxiv.org/abs/1911.11641`.

Jie Cao, Tianwei Lin, Hongyang He, Rolan Yan, Wenqiao Zhang, Juncheng Li, Dongping Zhang, Siliang Tang, and Yueting Zhuang. Moa: Heterogeneous mixture of adapters for parameter-efficient fine-tuning of large language models, 2025. URL `https://arxiv.org/abs/2506.05928`.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL `https://arxiv.org/abs/2107.03374`.

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

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019. URL https://arxiv.org/abs/1905.10044.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs, 2019. URL https://arxiv.org/abs/1903.00161.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging, 2025. URL https://arxiv.org/abs/2412.00081.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021b. URL https://arxiv.org/abs/2103.03874.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts, 2024. URL https://arxiv.org/abs/2404.15159.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018. URL https://arxiv.org/abs/1809.02789.

Siyuan Mu and Sen Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications, 2025. URL https://arxiv.org/abs/2503.07137.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge, 2019. URL https://arxiv.org/abs/1808.07042.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019. URL https://arxiv.org/abs/1904.09728.

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.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Bhoomit Vasani, Jack FitzGerald, Anjie Fang, and Sushmit Vaish. Phlora: data-free post-hoc low-rank adapter extraction from full-rank checkpoint, 2025. URL https://arxiv.org/abs/2509.10971.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5744–5760, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max W.F. Ku, Kai Wang, Alex Zhuang, Rongqi "Richard" Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *ArXiv*, abs/2406.01574, 2024. URL https://api.semanticscholar.org/CorpusID:270210486.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023. URL https://arxiv.org/abs/2306.01708.

Liping Yi, Han Yu, Chao Ren, Heng Zhang, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedmoe: Data-level personalization with mixture of experts for model-heterogeneous personalized federated learning, 2024. URL https://arxiv.org/abs/2402.01350.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023. URL https://arxiv.org/abs/2304.06364.

Table 4: Hugging Face Transformers inference measurements for 7×7B FlexMoRE configurations and their FlexOlmo baseline. Measurements use one NVIDIA B200 GPU, bfloat16, eager inference, KV caching, greedy decoding, a 128-token prompt, and 1,024 generated tokens per sequence; each entry is the median of three repetitions. FlexMoRE substantially reduces resident GPU memory (66.6% for the MC9-selected configuration and 73.6% for homogeneous $r = 128$), but is slower under this unfused eager backend.

Configuration	Batch	Params (B) FlexMoRE / FlexOlmo	Resident GPU (GiB) FlexMoRE / FlexOlmo	TTFT (ms) FlexMoRE / FlexOlmo	Throughput (tok/s) FlexMoRE / FlexOlmo	Relative throughput FlexMoRE / FlexOlmo
MC9-selected heterogeneous	1	10.75 / 33.27	20.94 / 62.74	99.41 / 75.61	21.28 / 24.17	0.88×
MC9-selected heterogeneous	4	10.75 / 33.27	20.94 / 62.74	100.19 / 78.92	84.69 / 95.98	0.88×
Homogeneous ($r = 128$)	1	8.41 / 33.27	16.55 / 62.74	102.90 / 76.60	21.40 / 25.74	0.83×
Homogeneous ($r = 128$)	4	8.41 / 33.27	16.55 / 62.74	102.86 / 75.98	84.96 / 103.71	0.82×

A Resident Memory and Inference Time

To assess the deployment trade-off beyond parameter counts, we measure end-to-end inference with the Hugging Face Transformers eager backend. Table 4 reports time to first token (TTFT), end-to-end generation throughput, and resident GPU memory for selected 7×7B FlexMoRE configurations and their FlexOlmo baseline. FlexMoRE reduces resident GPU memory by 66.6 -73.6%, but reaches only 0.82–0.88× the FlexOlmo baseline's throughput in this unfused backend. These results therefore demonstrate a memory–throughput trade-off rather than a general latency advantage; absolute throughput is expected to depend on the serving backend, hardware, batching, and availability of fused adapter/MoE kernels.

B FlexMoRE and AdapterSoup

Tables 6, 7, and 8 report the full rank sweeps and selected heterogeneous configurations for $a2$, $a4$, and $a7$, respectively. Figure 4 summarizes benchmark scores, while Figure 5 shows task-level differences from the matched FlexOlmo baselines.

B.1 FlexMoRE Performance Across Ranks

Tables 6, 7, and 8 report benchmark scores, Avg. with MC9, stored parameter counts, and relative changes in Avg. from the matched FlexOlmo baselines for $a2$, $a4$, and $a7$, respectively. They cover homogeneous ranks $r = 2^0, \ldots, 2^{14}$ and the MC9-selected and All-selected heterogeneous configurations, whose rank allocations are listed in Table 5. Figure 4 summarizes the benchmark-level curves and Avg. with and without MC9; Figure 5 shows task-level score differences from the matched FlexOlmo baselines.

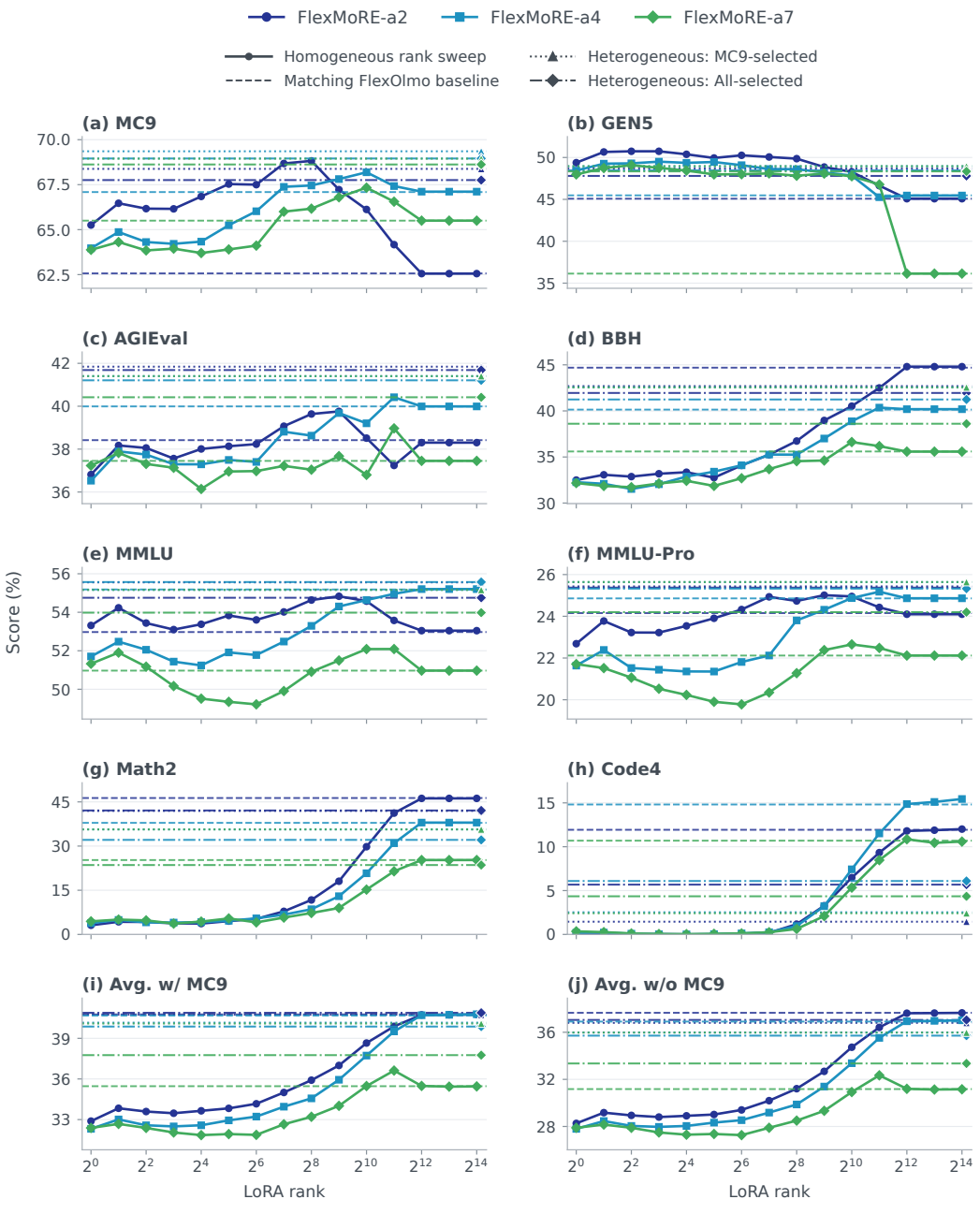

Figure 4: **FlexMoRE performance across ranks.** Panels show the eight benchmarks and Avg. with and without MC9. Curves represent homogeneous mixtures; horizontal references show the matched FlexOlmo baselines and selected heterogeneous allocations for each activation setting.

Table 5: **Rank allocations and stored capacity of the six-expert collection.** MC9-selected and All-selected configurations use the per-expert rank tuples listed here; the same tuples are used for $a2$, $a4$, and $a7$. The number of activated experts changes inference sparsity, not the stored expert collection or its parameter count. *All* selects each expert rank using its unweighted macro average (Avg.) with MC9 across all eight benchmarks. The homogeneous range refers to $r \in \{2^0, \ldots, 2^{11}\}$ used in selected-model summaries. We additionally evaluate $r = 2^{12}, \ldots, 2^{14}$ for rank-sensitivity analysis (up to 149.84B stored parameters), but do not treat these settings as practically low rank.

Selection	Stored params.	Code	Creative	Math	News	Academic	Reddit
MC9-selected	10.75B	2^6	2^7	2^{11}	2^0	2^3	2^7
All-selected	14.84B	2^9	2^4	2^{11}	2^6	2^{11}	2^9
Homogeneous	7.31–25.12B	r	r	r	r	r	r

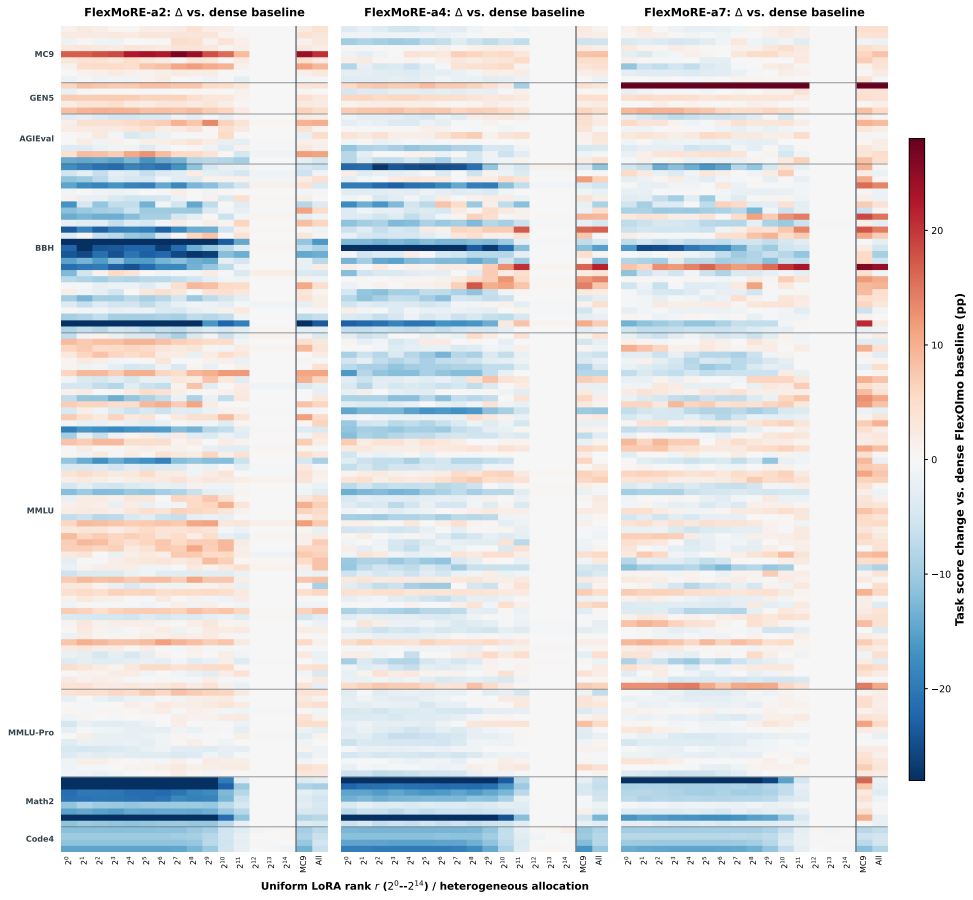

Figure 5: **Task-level rank sensitivity of homogeneous and heterogeneous FlexMoRE mixtures.** Panels show FlexMoRE with $a2$, $a4$, and $a7$ active experts. Rows are the 132 evaluation tasks grouped by benchmark family. The first 15 columns are homogeneous mixtures with uniform LoRA ranks $r = 2^0, \ldots, 2^{14}$; the final two columns are the heterogeneous MC9-selected and all-benchmark-selected allocations. Each cell is the task-score change in percentage points relative to the corresponding FlexOlmo baseline. Blue denotes degradation, red improvement, and near-white indicates near-baseline performance.

Table 6: **Bold** indicates best or better than baseline, where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

7x7B-1T	Params	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)
FlexOlmo-a2 (baseline)	33.27B	0.6257	0.4508	0.3842	0.4469	0.5297	0.2415	**0.4632**	0.1193	0.4077	–
FlexMoRE-a2 heterogen. (MC9)	10.75B	0.6838	0.4868	**0.4184**	0.4269	**0.5555**	**0.2543**	0.4193	0.0145	0.4075	-0.05
FlexMoRE-a2 heterogen. (All)	14.84B	0.6776	0.4780	0.4169	0.4195	0.5475	0.2537	0.4209	0.0568	**0.4089**	**+0.29**
FlexMoRE-a2 homogen. ($r = 1$)	7.31B	0.6526	0.4938	0.3681	0.3249	0.5332	0.2269	0.0311	0.0003	0.3289	-19.33
FlexMoRE-a2 homogen. ($r = 2$)	7.32B	0.6647	0.5066	0.3817	0.3307	0.5423	0.2377	0.0428	0.0003	0.3384	-17.00
FlexMoRE-a2 homogen. ($r = 4$)	7.33B	0.6617	**0.5074**	0.3805	0.3287	0.5344	0.2322	0.0424	0.0002	0.3359	-17.59
FlexMoRE-a2 homogen. ($r = 8$)	7.37B	0.6616	**0.5074**	0.3756	0.3318	0.5310	0.2322	0.0377	0.0003	0.3347	-17.90
FlexMoRE-a2 homogen. ($r = 16$)	7.44B	0.6684	0.5038	0.3801	0.3335	0.5338	0.2354	0.0368	0.0000	0.3365	-17.46
FlexMoRE-a2 homogen. ($r = 32$)	7.58B	0.6753	0.4994	0.3813	0.3275	0.5383	0.2390	0.0452	0.0002	0.3383	-17.02
FlexMoRE-a2 homogen. ($r = 64$)	7.86B	0.6750	0.5026	0.3823	0.3409	0.5360	0.2432	0.0526	0.0006	0.3417	-16.19
FlexMoRE-a2 homogen. ($r = 128$)	8.41B	0.6868	0.5006	0.3907	0.3524	0.5402	0.2493	0.0783	0.0018	0.3500	-14.14
FlexMoRE-a2 homogen. ($r = 256$)	9.53B	**0.6882**	0.4986	0.3964	0.3673	0.5463	0.2473	0.1170	0.0116	0.3591	-11.91
FlexMoRE-a2 homogen. ($r = 512$)	11.75B	0.6722	0.4886	0.3976	0.3897	0.5483	0.2501	0.1806	0.0327	0.3700	-9.25
FlexMoRE-a2 homogen. ($r = 1024$)	16.21B	0.6612	0.4828	0.3851	0.4052	0.5458	0.2495	0.2978	0.0648	0.3865	-5.19
FlexMoRE-a2 homogen. ($r = 2048$)	25.12B	0.6417	0.4664	0.3724	0.4249	0.5358	0.2443	0.4119	0.0933	0.3988	-2.17
FlexMoRE-a2 homogen. ($r = 4096$)	42.93B	0.6256	0.4508	0.3830	**0.4480**	0.5304	0.2410	0.4619	0.1180	0.4073	-0.08
FlexMoRE-a2 homogen. ($r = 8192$)	78.57B	0.6256	0.4508	0.3830	**0.4480**	0.5304	0.2410	0.4619	0.1189	0.4074	-0.05
FlexMoRE-a2 homogen. ($r = 16384$)	149.84B	0.6256	0.4508	0.3830	**0.4480**	0.5304	0.2410	0.4619	**0.1201**	0.4076	-0.02

Table 7: **Bold** indicates best or better than baseline, where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

7x7B-1T	Params	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)
FlexOlmo-a4 (baseline)	33.27B	0.6709	0.4548	0.3999	0.4014	0.5518	0.2486	0.3788	0.1482	0.4068	–
FlexMoRE-a4 heterogen. (MC9)	10.75B	**0.6936**	0.4896	**0.4140**	**0.4255**	0.5516	**0.2564**	0.3565	0.0252	0.4016	-1.29
FlexMoRE-a4 heterogen. (All)	14.84B	0.6896	0.4846	0.4121	0.4124	**0.5557**	0.2533	0.3210	0.0609	0.3987	-2.00
FlexMoRE-a4 homogen. ($r = 1$)	7.31B	0.6397	0.4844	0.3653	0.3226	0.5170	0.2164	0.0382	0.0018	0.3232	-20.56
FlexMoRE-a4 homogen. ($r = 2$)	7.32B	0.6487	0.4926	0.3790	0.3209	0.5247	0.2239	0.0491	0.0021	0.3301	-18.85
FlexMoRE-a4 homogen. ($r = 4$)	7.33B	0.6431	0.4930	0.3775	0.3153	0.5205	0.2152	0.0409	0.0008	0.3258	-19.92
FlexMoRE-a4 homogen. ($r = 8$)	7.37B	0.6421	**0.4950**	0.3730	0.3205	0.5144	0.2144	0.0402	0.0002	0.3250	-20.12
FlexMoRE-a4 homogen. ($r = 16$)	7.44B	0.6433	0.4936	0.3729	0.3288	0.5124	0.2136	0.0418	0.0001	0.3258	-19.91
FlexMoRE-a4 homogen. ($r = 32$)	7.58B	0.6524	0.4946	0.3750	0.3341	0.5191	0.2135	0.0464	0.0004	0.3294	-19.02
FlexMoRE-a4 homogen. ($r = 64$)	7.86B	0.6602	0.4908	0.3740	0.3410	0.5178	0.2181	0.0546	0.0012	0.3322	-18.34
FlexMoRE-a4 homogen. ($r = 128$)	8.41B	0.6738	0.4862	0.3881	0.3526	0.5248	0.2212	0.0670	0.0025	0.3395	-16.54
FlexMoRE-a4 homogen. ($r = 256$)	9.53B	0.6746	0.4860	0.3863	0.3525	0.5329	0.2380	0.0855	0.0095	0.3457	-15.03
FlexMoRE-a4 homogen. ($r = 512$)	11.75B	0.6781	0.4822	0.3968	0.3700	0.5430	0.2431	0.1298	0.0322	0.3594	-11.65
FlexMoRE-a4 homogen. ($r = 1024$)	16.21B	0.6819	0.4784	0.3920	0.3887	0.5463	0.2487	0.2076	0.0743	0.3772	-7.27
FlexMoRE-a4 homogen. ($r = 2048$)	25.12B	0.6742	0.4526	0.4042	0.4036	0.5496	0.2518	0.3097	0.1152	0.3951	-2.88
FlexMoRE-a4 homogen. ($r = 4096$)	42.93B	0.6711	0.4546	0.3999	0.4019	0.5520	0.2486	**0.3797**	0.1487	0.4071	**+0.06**
FlexMoRE-a4 homogen. ($r = 8192$)	78.57B	0.6711	0.4546	0.3999	0.4019	0.5520	0.2486	**0.3797**	0.1511	0.4074	**+0.14**
FlexMoRE-a4 homogen. ($r = 16384$)	149.84B	0.6711	0.4546	0.3999	0.4019	0.5520	0.2486	**0.3797**	**0.1544**	**0.4078**	**+0.24**

Table 8: **Bold** indicates best or better than baseline, where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

7x7B-1T	Params	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)
FlexOlmo-a7 (baseline)	33.27B	0.6550	0.3614	0.3745	0.3560	0.5097	0.2212	0.2526	0.1069	0.3547	–
FlexMoRE-a7 heterogen. (MC9)	10.75B	**0.6894**	0.4896	**0.4140**	**0.4255**	**0.5516**	**0.2564**	**0.3565**	0.0241	**0.4009**	**+13.04**
FlexMoRE-a7 heterogen. (All)	14.84B	0.6862	0.4834	0.4042	0.3861	0.5398	0.2420	0.2356	0.0435	0.3776	**+6.47**
FlexMoRE-a7 homogen. ($r = 1$)	7.31B	0.6388	0.4798	0.3723	0.3217	0.5132	0.2171	0.0443	0.0035	0.3238	-8.69
FlexMoRE-a7 homogen. ($r = 2$)	7.32B	0.6431	0.4874	0.3782	0.3185	0.5190	0.2152	0.0504	0.0026	0.3268	-7.85
FlexMoRE-a7 homogen. ($r = 4$)	7.33B	0.6384	**0.4908**	0.3731	0.3172	0.5118	0.2106	0.0476	0.0011	0.3238	-8.69
FlexMoRE-a7 homogen. ($r = 8$)	7.37B	0.6394	0.4876	0.3713	0.3213	0.5017	0.2053	0.0368	0.0005	0.3205	-9.64
FlexMoRE-a7 homogen. ($r = 16$)	7.44B	0.6370	0.4844	0.3614	0.3241	0.4951	0.2023	0.0435	0.0003	0.3185	-10.19
FlexMoRE-a7 homogen. ($r = 32$)	7.58B	0.6390	0.4798	0.3696	0.3186	0.4935	0.1990	0.0542	0.0007	0.3193	-9.97
FlexMoRE-a7 homogen. ($r = 64$)	7.86B	0.6411	0.4800	0.3698	0.3270	0.4922	0.1978	0.0408	0.0011	0.3187	-10.13
FlexMoRE-a7 homogen. ($r = 128$)	8.41B	0.6600	0.4810	0.3722	0.3368	0.4991	0.2035	0.0572	0.0025	0.3265	-7.93
FlexMoRE-a7 homogen. ($r = 256$)	9.53B	0.6617	0.4782	0.3704	0.3454	0.5091	0.2127	0.0728	0.0062	0.3321	-6.37
FlexMoRE-a7 homogen. ($r = 512$)	11.75B	0.6680	0.4808	0.3767	0.3462	0.5150	0.2238	0.0898	0.0209	0.3401	-4.09
FlexMoRE-a7 homogen. ($r = 1024$)	16.21B	0.6733	0.4776	0.3680	0.3662	0.5209	0.2265	0.1522	0.0534	0.3548	**+0.03**
FlexMoRE-a7 homogen. ($r = 2048$)	25.12B	0.6656	0.4678	0.3897	0.3618	0.5209	0.2248	0.2143	0.0847	0.3662	**+3.25**
FlexMoRE-a7 homogen. ($r = 4096$)	42.93B	0.6550	0.3614	0.3745	0.3558	0.5097	0.2212	0.2526	**0.1086**	0.3548	**+0.05**
FlexMoRE-a7 homogen. ($r = 8192$)	78.57B	0.6550	0.3614	0.3745	0.3558	0.5097	0.2212	0.2526	0.1044	0.3543	-0.09
FlexMoRE-a7 homogen. ($r = 16384$)	149.84B	0.6550	0.3614	0.3745	0.3558	0.5097	0.2212	0.2526	0.1059	0.3545	-0.04

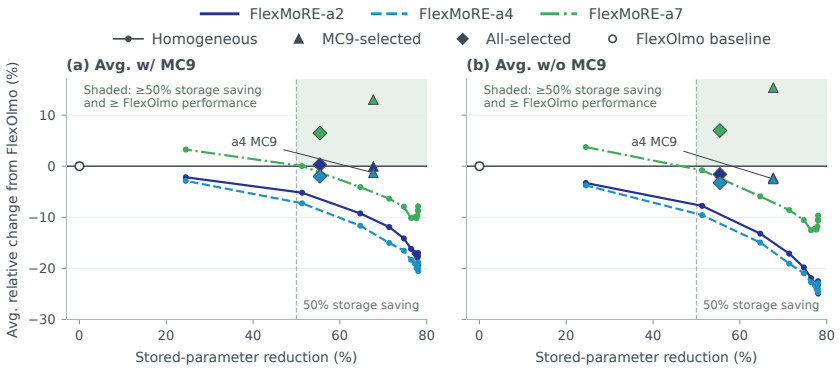

Figure 6: **Relative performance versus parameter savings.** The two panels report Avg. with and without MC9. The horizontal axis shows the percentage reduction in total stored parameters relative to FlexOlmo; the vertical axis shows relative score change, computed as $100(\text{Avg.}_{\text{FlexMoRE}}/\text{Avg.}_{\text{FlexOlmo}} - 1)$ using the corresponding averaging rule. Zero indicates equal performance. Curves connect evaluated homogeneous-rank configurations, while triangles and diamonds denote heterogeneous configurations selected using MC9 and all benchmarks, respectively. Only configurations with no more stored parameters than FlexOlmo are included. The shaded region indicates at least 50% parameter savings with performance at or above FlexOlmo; it is not a confidence band. Parameter savings describe stored model weights, not total inference memory usage.

B.2 FlexMoRE Relative Parameter Savings

Figure 6 reports the relative parameter savings for FlexMoRE compositions.

B.3 FlexMoRE versus AdapterSoup

The results show (Table 9; Figure 7–8) that the static AdapterSoup is relatively competitive only at very low ranks, but its performance deteriorates sharply as rank increases. In contrast, FlexMoRE retains separately addressable expert updates and combines expert computations using the released input-dependent FlexOlmo router. Thus, the comparison shows that naïve static averaging is weaker than the FlexMoRE system as a whole; it does not isolate the contribution of routing, low-rank factorization, or heterogeneous rank allocation individually.

The AdapterSoup is constructed from the mean expert delta rather than an unscaled sum. This avoids a simple unscaled-sum construction, but does not isolate the effects of update magnitude. A plausible explanation is that uniformly merging increasingly expressive domain-specific updates introduces conflicting directions that are beneficial for different inputs, whereas conditional routing need not apply all such updates equally. We present this as an interpretation, not a demonstrated causal mechanism. This behavior is consistent with model-merging work that identifies conflicts among independently fine-tuned task updates (e.g., TIES-Merging Yadav et al. (2023); TSV-Merge Gargiulo et al. (2025)).

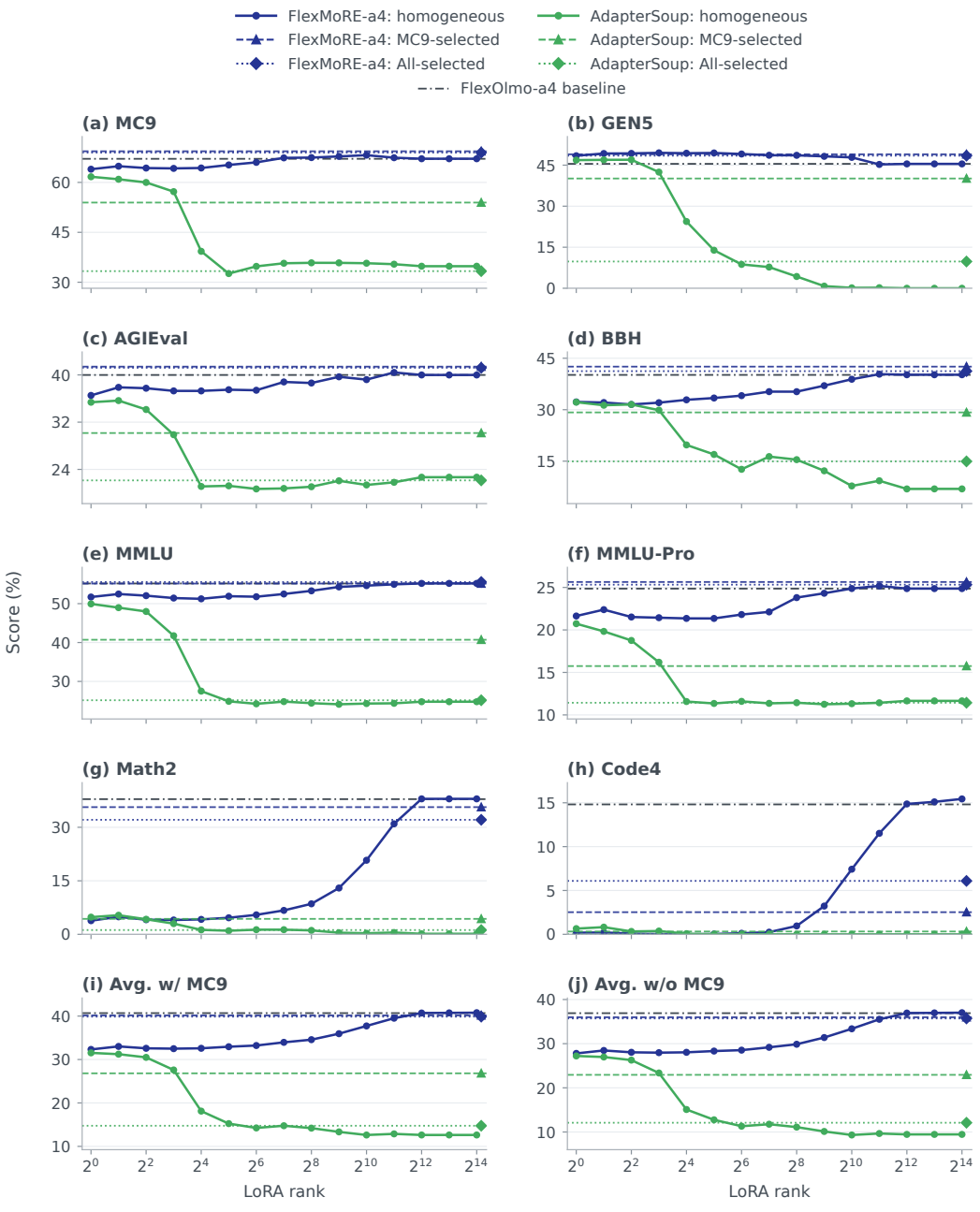

Figure 7: **FlexMoRE-$a4$ vs. AdapterSoup across ranks.** Panels show the eight benchmarks and Avg. with and without MC9. Homogeneous rank sweeps and selected heterogeneous configurations are compared using the same expert updates. The dense reference is the FlexOlmo-$a4$ baseline.

Table 9: FlexMoRE-a4 and AdapterSoup comparison using the paper's unweighted macro average over the seven non-MC9 benchmarks. The static soup uses a uniform mean merge of the expert updates.

Configuration	**Avg. without MC9**
FlexMoRE, homogeneous ($r = 1$)	27.80
AdapterSoup, homogeneous ($r = 1$)	27.22
FlexMoRE, homogeneous ($r = 4$)	28.05
AdapterSoup, homogeneous ($r = 4$)	26.28
FlexMoRE, homogeneous ($r = 128$)	29.18
AdapterSoup, homogeneous ($r = 128$)	11.76
FlexMoRE, homogeneous ($r = 2048$)	35.52
AdapterSoup, homogeneous ($r = 2048$)	9.66
FlexMoRE, MC9-selected heterogeneous	35.99
AdapterSoup, MC9-selected heterogeneous	22.94

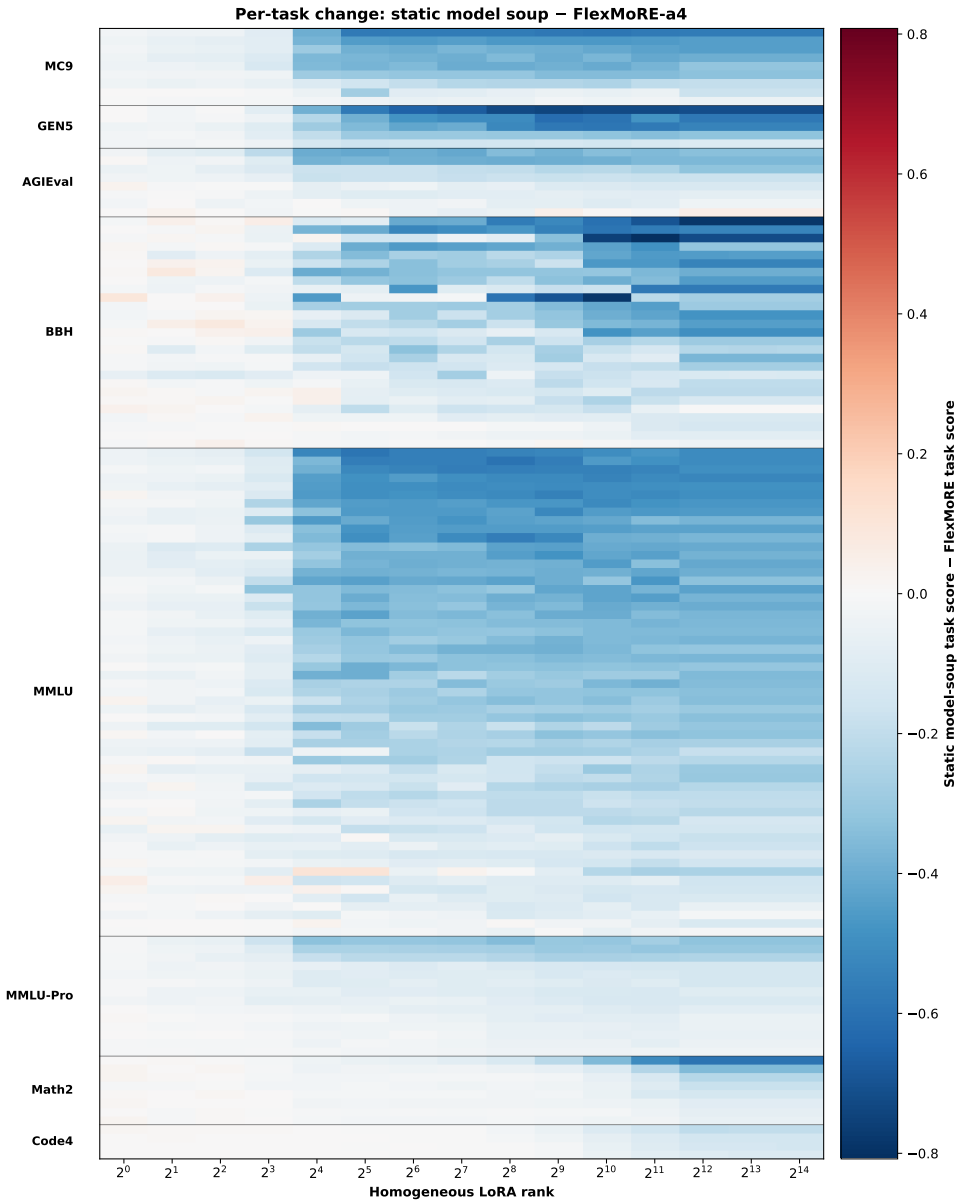

Figure 8: **Per-task comparison of static model merging and FlexMoRE-a4.** Rows are the 132 evaluation tasks, grouped by benchmark family, and columns are homogeneous LoRA ranks $r = 2^0, \ldots, 2^{14}$. Each cell is the AdapterSoup task score minus the corresponding FlexMoRE-a4 task score; blue values therefore favour FlexMoRE, whereas red values favor AdapterSoup. The AdapterSoup applies one uniform mean merge of the expert updates, while FlexMoRE retains separately addressable experts and uses the released input-dependent router.

C EXAMPLE-LEVEL EVALUATION UNCERTAINTY

We quantify finite-evaluation-set uncertainty with 5,000 nonparametric bootstrap replicates. In each replicate, we resample the full number of evaluation examples with replacement within every task, recompute the task score, and apply the same aggregation hierarchy as in the main results: tasks are weighted equally within each benchmark, benchmarks are weighted equally, and Math2 remains the mean of GSM8K and the mean of the Minerva subjects. Score intervals are reported as the point estimate ± 1.96 times the bootstrap-estimated standard error, i.e., normal-approximation 95% confidence intervals. They quantify uncertainty from the finite evaluation sets conditional on the evaluated checkpoints and fixed prompts; they are not estimates of training-seed, optimization, or model-randomness variation.

Tables 10 and 11 report macro- and benchmark-level intervals for FlexMoRE, respectively; Tables 12 and 13 provide the corresponding results for the individual experts. The macro-average intervals have half-widths of approximately 0.4–0.5 percentage points.

For completeness, Tables 14 and 15 provide secondary paired comparisons for selected model pairs. In each paired replicate, the same resampled document indices are used for both models. The bracketed intervals are normal-approximation 95% intervals for the paired score difference, computed as the difference ± 1.96 times its paired bootstrap-estimated standard error.

Table 10: Example-level uncertainty for FlexMoRE macro averages. Scores are in percentage points and are reported as mean ± 1.96 times the bootstrap-estimated standard error from 5,000 within-task bootstrap replicates.

	Macro Avg. incl. MC9	**Macro Avg. excl. MC9**
FlexOlmo-a2 (baseline)	40.77 ± 0.46	37.65 ± 0.51
FlexMoRE-a2 heterogen. (MC9)	40.75 ± 0.43	36.80 ± 0.47
FlexMoRE-a2 heterogen. (All)	40.89 ± 0.45	37.05 ± 0.49
FlexMoRE-a2 homogen. ($r = 2048$)	39.88 ± 0.45	36.41 ± 0.49
FlexOlmo-a4 (baseline)	40.68 ± 0.48	36.91 ± 0.53
FlexMoRE-a4 heterogen. (MC9)	40.16 ± 0.44	35.99 ± 0.48
FlexMoRE-a4 heterogen. (All)	39.87 ± 0.45	35.71 ± 0.49
FlexMoRE-a4 homogen. ($r = 2048$)	39.51 ± 0.47	35.52 ± 0.51
FlexOlmo-a7 (baseline)	35.47 ± 0.46	31.17 ± 0.50
FlexMoRE-a7 heterogen. (MC9)	40.09 ± 0.44	35.97 ± 0.49
FlexMoRE-a7 heterogen. (All)	37.76 ± 0.44	33.35 ± 0.47
FlexMoRE-a7 homogen. ($r = 2048$)	36.62 ± 0.45	32.34 ± 0.49

Table 11: Benchmark-level uncertainty for FlexMoRE mixtures. Entries are mean benchmark scores ± 1.96 times the bootstrap-estimated standard error (normal-approximation 95% intervals), in percentage points, from 5,000 within-task bootstrap replicates.

	MC9	**GEN5**	**AGIEval**	**BBH**	**MMLU**	**MMLU-Pro**	**Math2**	**Code4**
FlexOlmo-a2 (baseline)	62.57 ± 1.03	45.08 ± 1.25	38.42 ± 1.93	44.69 ± 1.09	52.97 ± 0.97	24.15 ± 0.82	46.32 ± 1.57	11.93 ± 1.47
FlexMoRE-a2 heterogen. (MC9)	68.38 ± 0.99	48.68 ± 1.26	41.84 ± 1.94	42.69 ± 1.08	55.55 ± 0.96	25.43 ± 0.84	41.93 ± 1.55	1.45 ± 0.53
FlexMoRE-a2 heterogen. (All)	67.76 ± 0.99	47.80 ± 1.24	41.69 ± 1.94	41.95 ± 1.08	54.75 ± 0.94	25.37 ± 0.84	42.09 ± 1.57	5.68 ± 0.96
FlexMoRE-a2 homogen. ($r = 2048$)	64.17 ± 1.03	46.64 ± 1.24	37.24 ± 1.91	42.49 ± 1.09	53.58 ± 0.97	24.43 ± 0.83	41.19 ± 1.54	9.33 ± 1.27
FlexOlmo-a4 (baseline)	67.09 ± 1.01	45.48 ± 1.25	39.99 ± 1.92	40.14 ± 1.07	55.18 ± 0.97	24.86 ± 0.83	37.88 ± 1.62	14.82 ± 1.70
FlexMoRE-a4 heterogen. (MC9)	69.36 ± 0.99	48.96 ± 1.26	41.40 ± 1.94	42.55 ± 1.07	55.16 ± 0.96	25.64 ± 0.84	35.65 ± 1.60	2.52 ± 0.58
FlexMoRE-a4 heterogen. (All)	68.96 ± 0.99	48.46 ± 1.25	41.21 ± 1.94	41.24 ± 1.07	55.57 ± 0.95	25.33 ± 0.82	32.10 ± 1.60	6.09 ± 0.99
FlexMoRE-a4 homogen. ($r = 2048$)	67.42 ± 1.02	45.26 ± 1.28	40.42 ± 1.91	40.36 ± 1.07	54.96 ± 0.97	25.18 ± 0.83	30.97 ± 1.60	11.52 ± 1.44
FlexOlmo-a7 (baseline)	65.50 ± 1.01	36.14 ± 1.16	37.45 ± 1.90	35.60 ± 1.05	50.97 ± 0.98	22.12 ± 0.79	25.26 ± 1.59	10.69 ± 1.38
FlexMoRE-a7 heterogen. (MC9)	68.94 ± 1.00	48.96 ± 1.25	41.40 ± 1.94	42.55 ± 1.07	55.16 ± 0.97	25.64 ± 0.84	35.65 ± 1.63	2.41 ± 0.55
FlexMoRE-a7 heterogen. (All)	68.62 ± 0.98	48.34 ± 1.26	40.42 ± 1.88	38.61 ± 1.07	53.98 ± 0.98	24.20 ± 0.81	23.56 ± 1.54	4.35 ± 0.80
FlexMoRE-a7 homogen. ($r = 2048$)	66.56 ± 1.02	46.78 ± 1.24	38.97 ± 1.93	36.18 ± 1.07	52.09 ± 0.96	22.48 ± 0.80	21.43 ± 1.54	8.47 ± 1.21

Table 12: Example-level uncertainty for individual-expert macro averages. Scores are in percentage points and are reported as mean ± 1.96 times the bootstrap-estimated standard error from 5,000 within-task bootstrap replicates.

	Macro Avg. incl. MC9	**Macro Avg. excl. MC9**
Public base model	35.81 ± 0.39	31.11 ± 0.43
Code (baseline)	36.96 ± 0.45	32.59 ± 0.50
Code ($r = 2048$)	36.76 ± 0.44	32.33 ± 0.49
Creative (baseline)	34.74 ± 0.40	30.11 ± 0.43
Creative ($r = 8$)	35.83 ± 0.40	31.19 ± 0.44
Math (baseline)	43.64 ± 0.47	40.11 ± 0.52
Math ($r = 2048$)	42.15 ± 0.45	38.41 ± 0.50
News (baseline)	34.45 ± 0.40	29.94 ± 0.43
News ($r = 1$)	36.08 ± 0.40	31.37 ± 0.44
Academic (baseline)	35.11 ± 0.40	30.50 ± 0.44
Academic ($r = 2048$)	35.27 ± 0.40	30.73 ± 0.43
Reddit (baseline)	33.37 ± 0.42	29.32 ± 0.45
Reddit ($r = 1024$)	35.23 ± 0.42	30.81 ± 0.45

Table 13: Benchmark-level uncertainty for individual experts. Entries are mean benchmark scores ± 1.96 times the bootstrap-estimated standard error (normal-approximation 95% intervals), in percentage points, from 5,000 within-task bootstrap replicates.

	MC9	**GEN5**	**AGIEval**	**BBH**	**MMLU**	**MMLU-Pro**	**Math2**	**Code4**
Public base model	68.72 ± 1.02	51.08 ± 1.27	39.94 ± 1.91	35.70 ± 1.06	55.87 ± 0.95	26.12 ± 0.84	8.16 ± 1.07	0.91 ± 0.21
Code (baseline)	67.57 ± 1.00	46.68 ± 1.23	39.09 ± 1.93	38.31 ± 1.08	53.81 ± 0.96	24.43 ± 0.84	8.31 ± 1.08	17.47 ± 1.69
Code ($r = 2048$)	67.79 ± 1.02	48.62 ± 1.22	39.03 ± 1.91	37.76 ± 1.08	53.88 ± 0.97	24.28 ± 0.84	9.09 ± 1.14	13.63 ± 1.54
Creative (baseline)	67.09 ± 1.01	49.98 ± 1.23	39.65 ± 1.94	34.24 ± 1.07	53.78 ± 0.96	24.78 ± 0.83	7.39 ± 1.06	0.97 ± 0.25
Creative ($r = 8$)	68.34 ± 1.02	50.64 ± 1.26	39.41 ± 1.91	35.69 ± 1.05	55.87 ± 0.95	25.82 ± 0.85	9.67 ± 1.17	1.20 ± 0.26
Math (baseline)	68.36 ± 0.99	48.62 ± 1.25	41.38 ± 1.93	45.89 ± 1.07	56.48 ± 0.96	26.15 ± 0.85	49.64 ± 1.60	12.62 ± 1.50
Math ($r = 2048$)	68.32 ± 1.00	49.54 ± 1.25	40.72 ± 1.95	42.89 ± 1.08	55.57 ± 0.96	25.98 ± 0.85	43.88 ± 1.58	10.26 ± 1.31
News (baseline)	66.02 ± 1.04	50.58 ± 1.24	36.91 ± 1.90	35.54 ± 1.04	53.87 ± 0.95	24.46 ± 0.83	7.65 ± 1.07	0.56 ± 0.18
News ($r = 1$)	69.10 ± 1.00	49.98 ± 1.26	40.32 ± 1.87	35.93 ± 1.04	55.89 ± 0.95	26.36 ± 0.85	9.93 ± 1.17	1.16 ± 0.26
Academic (baseline)	67.32 ± 1.01	50.00 ± 1.27	39.44 ± 1.90	36.22 ± 1.06	54.79 ± 0.97	24.22 ± 0.83	8.57 ± 1.12	0.30 ± 0.12
Academic ($r = 2048$)	67.10 ± 1.01	51.04 ± 1.25	39.28 ± 1.88	36.53 ± 1.08	54.86 ± 0.95	25.15 ± 0.83	7.97 ± 1.08	0.25 ± 0.10
Reddit (baseline)	61.71 ± 1.03	41.08 ± 1.22	36.68 ± 1.91	36.68 ± 1.07	53.58 ± 0.96	23.50 ± 0.81	10.00 ± 1.18	3.70 ± 0.76
Reddit ($r = 1024$)	66.20 ± 1.02	49.46 ± 1.23	39.22 ± 1.92	35.21 ± 1.06	54.74 ± 0.94	24.49 ± 0.83	9.34 ± 1.16	3.22 ± 0.73

Table 14: Secondary paired comparison of selected FlexMoRE mixtures. Differences are candidate minus reference, in percentage points. Bracketed intervals are normal-approximation 95% paired-bootstrap intervals from 5,000 replicates over the same evaluation examples.

Candidate − reference	**Δ incl. MC9 (95% CI)**	**Δ excl. MC9 (95% CI)**
a2 Hetero MC9 − baseline	-0.02 [-0.49, +0.45]	-0.85 [-1.37, -0.34]
a2 Hetero All − baseline	+0.12 [-0.34, +0.58]	-0.61 [-1.11, -0.10]
a2 Homogeneous − baseline	-0.88 [-1.31, -0.46]	-1.24 [-1.71, -0.77]
a2 Hetero MC9 − homogeneous	+0.86 [+0.43, +1.30]	+0.38 [-0.09, +0.86]
a4 Hetero MC9 − baseline	-0.52 [-0.97, -0.07]	-0.92 [-1.42, -0.42]
a4 Hetero All − baseline	-0.81 [-1.24, -0.39]	-1.19 [-1.67, -0.72]
a4 Homogeneous − baseline	-1.17 [-1.54, -0.80]	-1.39 [-1.80, -0.97]
a4 Hetero MC9 − homogeneous	+0.65 [+0.24, +1.06]	+0.46 [+0.00, +0.92]
a7 Hetero MC9 − baseline	+4.63 [+4.18, +5.07]	+4.79 [+4.30, +5.28]
a7 Hetero All − baseline	+2.29 [+1.89, +2.70]	+2.18 [+1.72, +2.63]
a7 Homogeneous − baseline	+1.15 [+0.81, +1.50]	+1.17 [+0.78, +1.55]
a7 Hetero MC9 − homogeneous	+3.47 [+3.06, +3.89]	+3.63 [+3.17, +4.09]

Table 15: Secondary paired comparison of selected low-rank experts. Differences are candidate minus reference, in percentage points. Bracketed intervals are normal-approximation 95% paired-bootstrap intervals from 5,000 replicates over the same evaluation examples.

Candidate − reference	**Δ incl. MC9 (95% CI)**	**Δ excl. MC9 (95% CI)**
Code r2048 − baseline	-0.20 [-0.48, +0.08]	-0.26 [-0.57, +0.05]
Creative r8 − baseline	+1.09 [+0.75, +1.44]	+1.07 [+0.69, +1.45]
Math r2048 − baseline	-1.50 [-1.88, -1.12]	-1.71 [-2.12, -1.29]
News r1 − baseline	+1.63 [+1.26, +2.01]	+1.43 [+1.03, +1.83]
Academic r2048 − baseline	+0.17 [-0.12, +0.45]	+0.22 [-0.09, +0.54]
Reddit r1024 − baseline	+1.87 [+1.46, +2.27]	+1.49 [+1.05, +1.94]

D 2X7B EXPERTS

In Section D and Tables 16-21; Figure 10, we report our results on 2x7B experts.

2x7B expert summary results for individual models across ranks in Table 16- 21.

Table 16: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T Code Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	0.6872	**0.5108**	**0.3994**	0.3570	**0.5587**	**0.2612**	0.0816	0.0085	0.3581	–	–
Flex-code (baseline)	11.63B	4.33B	0.6757	0.4668	0.3909	**0.3831**	0.5381	0.2443	0.0831	0.1747	0.3696	–	**+3.22**
Flex-code ($r = 1$)	7.30B	1.45M	0.6747	0.5014	0.3800	0.3438	0.5499	0.2416	0.0397	0.0005	0.3415	-7.61	-4.64
Flex-code ($r = 2$)	7.30B	2.90M	0.6716	0.5004	0.3834	0.3507	0.5494	0.2448	0.0696	0.0005	0.3463	-6.30	-3.28
Flex-code ($r = 4$)	7.30B	5.80M	0.6784	0.5048	0.3862	0.3517	0.5516	0.2498	0.0660	0.0003	0.3486	-5.68	-2.64
Flex-code ($r = 8$)	7.31B	11.60M	0.6797	0.5048	0.3782	0.3509	0.5507	0.2479	0.0681	0.0005	0.3476	-5.95	-2.92
Flex-code ($r = 16$)	7.32B	23.20M	0.6843	0.5002	0.3815	0.3510	0.5493	0.2445	0.0697	0.0001	0.3476	-5.95	-2.93
Flex-code ($r = 32$)	7.35B	46.40M	0.6873	0.5038	0.3802	0.3578	0.5499	0.2435	0.0672	0.0002	0.3487	-5.64	-2.60
Flex-code ($r = 64$)	7.39B	92.80M	**0.6889**	0.5024	0.3752	0.3626	0.5457	0.2438	0.0695	0.0010	0.3487	-5.66	-2.63
Flex-code ($r = 128$)	7.48B	185.60M	0.6860	0.4988	0.3751	0.3681	0.5464	0.2488	0.0751	0.0044	0.3503	-5.21	-2.16
Flex-code ($r = 256$)	7.67B	371.20M	0.6843	0.4974	0.3793	0.3730	0.5481	0.2460	0.0789	0.0110	0.3523	-4.69	-1.62
Flex-code ($r = 512$)	8.04B	742.39M	0.6866	0.4946	0.3824	0.3704	0.5464	0.2492	0.0818	0.0346	0.3558	-3.74	-0.64
Flex-code ($r = 1024$)	8.78B	1.48B	0.6851	0.4952	0.3801	0.3728	0.5403	0.2445	0.0811	0.0808	0.3600	-2.60	**+0.53**
Flex-code ($r = 2048$)	10.27B	2.97B	0.6779	0.4862	0.3903	0.3776	0.5388	0.2428	**0.0909**	0.1363	0.3676	-0.54	**+2.66**
Flex-code ($r = 4096$)	13.24B	5.94B	0.6758	0.4670	0.3909	**0.3831**	0.5381	0.2443	0.0831	0.1742	0.3696	-0.00	**+3.22**
Flex-code ($r = 8192$)	19.18B	11.83B	0.6758	0.4670	0.3909	**0.3831**	0.5381	0.2443	0.0831	**0.1769**	**0.3699**	**+0.09**	**+3.31**
Flex-code ($r = 16384$)	31.06B	23.76B	0.6758	0.4670	0.3909	**0.3831**	0.5381	0.2443	0.0831	0.1701	0.3691	-0.14	**+3.07**

Table 17: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T Creative Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	0.6872	**0.5108**	0.3994	0.3570	0.5587	**0.2612**	0.0816	0.0085	0.3581	–	–
Flex-creative (baseline)	11.63B	4.33B	0.6709	0.4998	0.3965	0.3424	0.5378	0.2478	0.0739	0.0097	0.3474	–	-2.99
Flex-creative ($r = 1$)	7.30B	1.45M	0.6864	0.5028	0.3969	0.3483	0.5590	0.2584	0.0922	0.0094	0.3567	**+2.68**	-0.39
Flex-creative ($r = 2$)	7.30B	2.90M	0.6813	0.5046	0.3940	0.3569	0.5601	0.2568	0.0934	0.0100	0.3571	**+2.81**	-0.26
Flex-creative ($r = 4$)	7.30B	5.80M	0.6844	0.5070	0.3897	**0.3578**	**0.5605**	0.2574	0.0954	0.0107	0.3579	**+3.03**	-0.05
Flex-creative ($r = 8$)	7.31B	11.60M	0.6834	0.5064	0.3941	0.3569	0.5587	0.2582	0.0967	0.0120	**0.3583**	**+3.15**	**+0.07**
Flex-creative ($r = 16$)	7.32B	23.20M	0.6854	0.5074	0.3959	0.3545	0.5601	0.2595	0.0901	0.0087	0.3577	**+2.98**	-0.10
Flex-creative ($r = 32$)	7.35B	46.40M	0.6853	0.5070	0.3942	0.3538	0.5589	0.2587	0.0904	0.0094	0.3572	**+2.84**	-0.23
Flex-creative ($r = 64$)	7.39B	92.80M	0.6876	0.5028	0.3992	0.3508	0.5569	0.2580	0.0963	0.0069	0.3573	**+2.87**	-0.21
Flex-creative ($r = 128$)	7.48B	185.60M	**0.6886**	0.5024	0.3964	0.3554	0.5565	0.2566	0.0995	0.0087	0.3580	**+3.07**	-0.01
Flex-creative ($r = 256$)	7.67B	371.20M	0.6839	0.5038	0.4001	0.3514	0.5533	0.2535	**0.1006**	0.0083	0.3569	**+2.74**	-0.33
Flex-creative ($r = 512$)	8.04B	742.39M	0.6824	0.5046	**0.4020**	0.3461	0.5565	0.2531	0.0998	0.0085	0.3566	**+2.67**	-0.40
Flex-creative ($r = 1024$)	8.78B	1.48B	0.6812	0.5006	0.4012	0.3456	0.5481	0.2511	0.0893	0.0055	0.3528	**+1.57**	-1.46
Flex-creative ($r = 2048$)	10.27B	2.97B	0.6713	0.5026	0.3915	0.3438	0.5416	0.2456	0.0793	0.0098	0.3482	**+0.24**	-2.76
Flex-creative ($r = 4096$)	13.24B	5.94B	0.6709	0.4998	0.3965	0.3424	0.5378	0.2478	0.0739	0.0107	0.3475	**+0.04**	-2.95
Flex-creative ($r = 8192$)	19.18B	11.83B	0.6709	0.4998	0.3965	0.3424	0.5378	0.2478	0.0739	0.0109	0.3475	**+0.05**	-2.94
Flex-creative ($r = 16384$)	31.06B	23.76B	0.6709	0.4998	0.3965	0.3424	0.5378	0.2478	0.0739	**0.0123**	0.3477	**+0.10**	-2.90

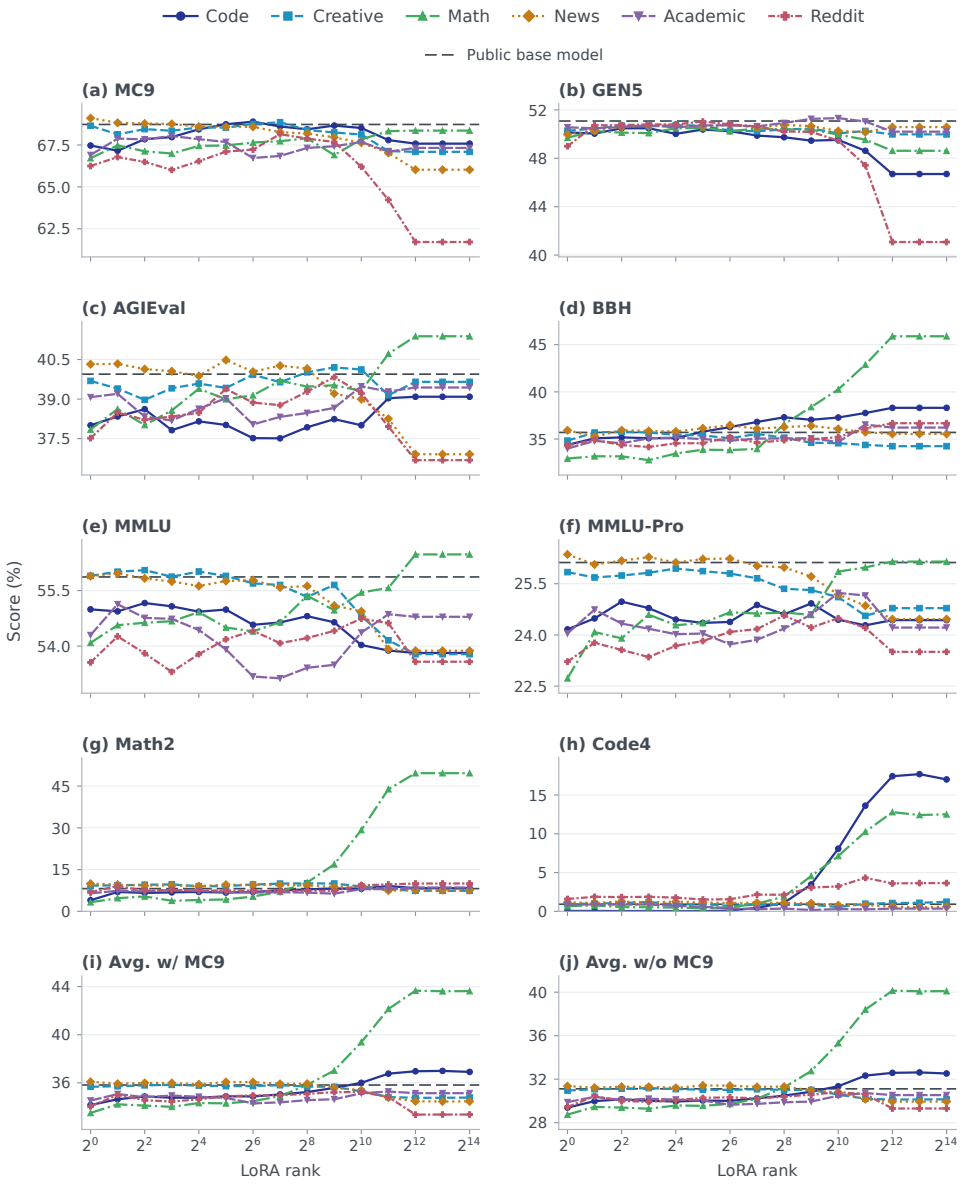

Figure 9: **Individual domain-expert performance across ranks.** Panels show the eight benchmarks and Avg. with and without MC9. Each expert curve is evaluated alongside the public expert in a two-expert configuration. The public base model is evaluated separately as a standalone reference.

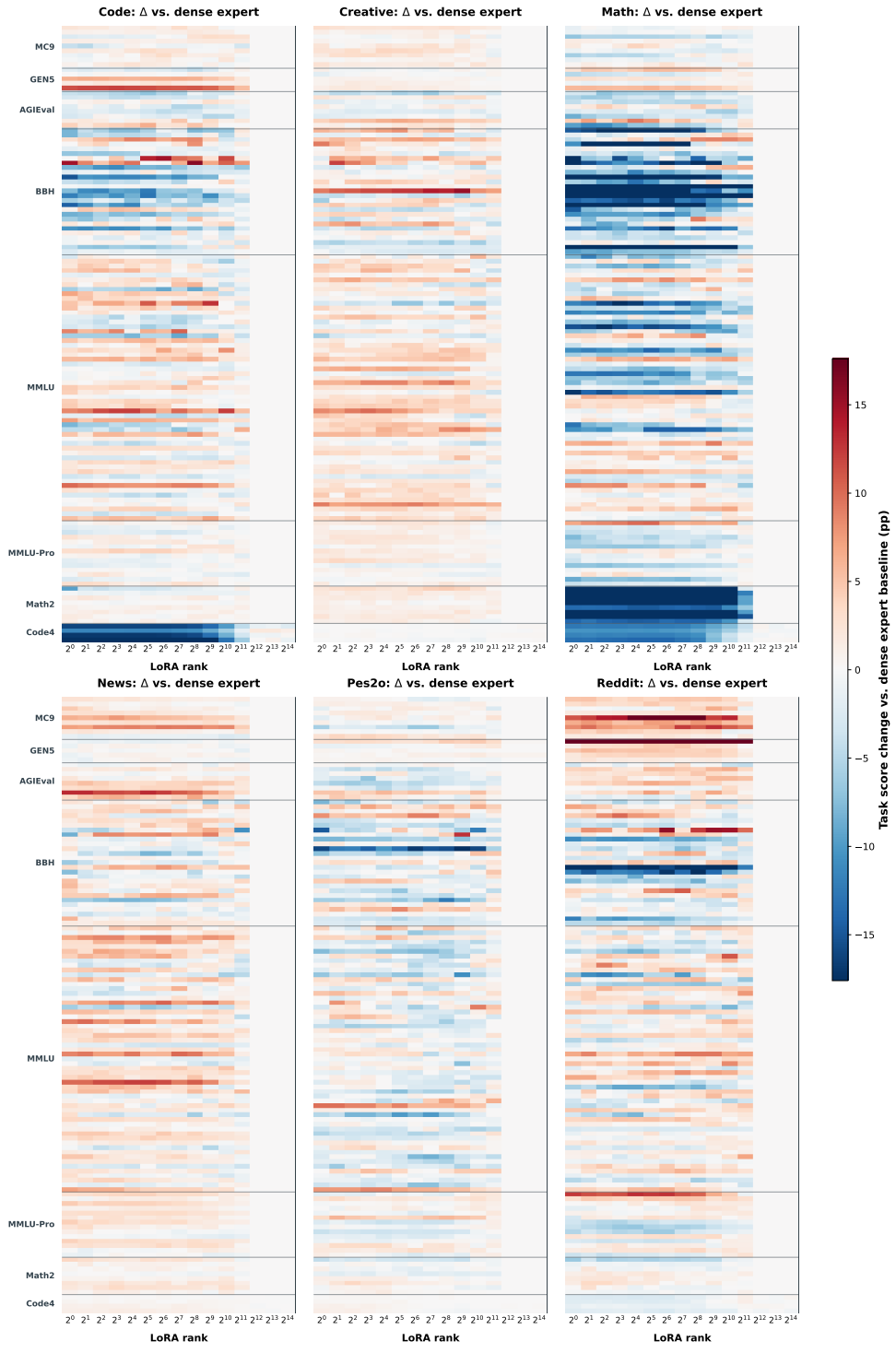

Figure 10: **Task-level rank sensitivity of individual experts.** Each panel corresponds to one independently trained domain expert. Rows are the 132 evaluation tasks, grouped by benchmark family; columns are LoRA ranks $r = 2^0, \ldots, 2^{14}$. Each cell shows the task-score change in percentage points relative to its dense expert baseline; both the low-rank and dense domain expert are evaluated alongside the public expert in the same two-expert configuration. Blue denotes degradation and red denotes improvement; near-white cells indicate performance close to the dense baseline, rather than missing evaluations.

Table 18: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T Math Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	**0.6872**	**0.5108**	0.3994	0.3570	0.5587	0.2612	0.0816	0.0085	0.3581	–	–
Flex-math (baseline)	11.63B	4.33B	0.6836	0.4862	**0.4138**	**0.4589**	**0.5648**	**0.2615**	**0.4964**	0.1262	0.4364	–	**+21.88**
Flex-math ($r = 1$)	7.30B	1.45M	0.6671	0.4970	0.3785	0.3293	0.5409	0.2273	0.0326	0.0066	0.3349	-23.26	-6.46
Flex-math ($r = 2$)	7.30B	2.90M	0.6748	0.5026	0.3862	0.3317	0.5456	0.2408	0.0471	0.0076	0.3421	-21.62	-4.47
Flex-math ($r = 4$)	7.30B	5.80M	0.6712	0.5012	0.3802	0.3317	0.5463	0.2390	0.0535	0.0054	0.3411	-21.85	-4.75
Flex-math ($r = 8$)	7.31B	11.60M	0.6699	0.5008	0.3857	0.3276	0.5467	0.2459	0.0379	0.0053	0.3400	-22.10	-5.05
Flex-math ($r = 16$)	7.32B	23.20M	0.6744	0.5048	0.3939	0.3347	0.5492	0.2429	0.0406	0.0048	0.3432	-21.37	-4.16
Flex-math ($r = 32$)	7.35B	46.40M	0.6747	0.5050	0.3899	0.3387	0.5450	0.2435	0.0426	0.0038	0.3429	-21.43	-4.23
Flex-math ($r = 64$)	7.39B	92.80M	0.6764	0.5024	0.3914	0.3384	0.5440	0.2466	0.0531	0.0051	0.3447	-21.02	-3.74
Flex-math ($r = 128$)	7.48B	185.60M	0.6772	0.5046	0.3970	0.3395	0.5465	0.2463	0.0703	0.0098	0.3489	-20.05	-2.56
Flex-math ($r = 256$)	7.67B	371.20M	0.6790	0.5040	0.3947	0.3657	0.5536	0.2466	0.1035	0.0191	0.3583	-17.91	**+0.06**
Flex-math ($r = 512$)	8.04B	742.39M	0.6690	0.5036	0.3952	0.3839	0.5497	0.2459	0.1688	0.0455	0.3702	-15.17	**+3.39**
Flex-math ($r = 1024$)	8.78B	1.48B	0.6783	0.4996	0.3931	0.4026	0.5545	0.2586	0.2922	0.0714	0.3938	-9.77	**+9.97**
Flex-math ($r = 2048$)	10.27B	2.97B	0.6832	0.4954	0.4072	0.4289	0.5557	0.2598	0.4388	0.1026	0.4215	-3.43	**+17.70**
Flex-math ($r = 4096$)	13.24B	5.94B	0.6836	0.4862	**0.4138**	**0.4589**	**0.5648**	**0.2615**	**0.4964**	**0.1279**	**0.4366**	**+0.05**	**+21.94**
Flex-math ($r = 8192$)	19.18B	11.83B	0.6836	0.4862	**0.4138**	**0.4589**	**0.5648**	**0.2615**	**0.4964**	0.1242	0.4362	-0.06	**+21.81**
Flex-math ($r = 16384$)	31.06B	23.76B	0.6836	0.4862	**0.4138**	**0.4589**	**0.5648**	**0.2615**	**0.4964**	0.1251	0.4363	-0.03	**+21.84**

Table 19: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T News Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	0.6872	**0.5108**	0.3994	0.3570	0.5587	0.2612	0.0816	0.0085	0.3581	–	–
Flex-news (baseline)	11.63B	4.33B	0.6602	0.5058	0.3691	0.3554	0.5387	0.2446	0.0765	0.0056	0.3445	–	-3.79
Flex-news ($r = 1$)	7.30B	1.45M	**0.6910**	0.4998	0.4032	0.3593	0.5589	**0.2636**	**0.0993**	0.0116	**0.3608**	**+4.74**	**+0.78**
Flex-news ($r = 2$)	7.30B	2.90M	0.6882	0.5026	0.4034	0.3524	**0.5596**	0.2606	0.0961	0.0109	0.3592	**+4.27**	**+0.33**
Flex-news ($r = 4$)	7.30B	5.80M	0.6878	0.5058	0.4014	0.3593	0.5583	0.2618	0.0923	**0.0126**	0.3599	**+4.47**	**+0.51**
Flex-news ($r = 8$)	7.31B	11.60M	0.6876	0.5070	0.4006	0.3586	0.5574	0.2628	0.0932	0.0115	0.3598	**+4.45**	**+0.50**
Flex-news ($r = 16$)	7.32B	32.20M	0.6859	0.5072	0.3988	0.3580	0.5562	0.2612	0.0903	0.0125	0.3588	**+4.14**	**+0.20**
Flex-news ($r = 32$)	7.35B	46.40M	0.6859	0.5064	**0.4048**	0.3614	0.5576	0.2623	0.0958	0.0115	0.3607	**+4.70**	**+0.74**
Flex-news ($r = 64$)	7.39B	92.80M	0.6858	0.5078	0.4004	**0.3646**	0.5578	0.2624	0.0947	0.0107	0.3605	**+4.65**	**+0.69**
Flex-news ($r = 128$)	7.48B	185.60M	0.6827	0.5054	0.4027	0.3609	0.5558	0.2602	0.0943	0.0111	0.3591	**+4.25**	**+0.30**
Flex-news ($r = 256$)	7.67B	371.20M	0.6818	0.5076	0.4016	0.3627	0.5562	0.2598	0.0936	0.0108	0.3593	**+4.28**	**+0.33**
Flex-news ($r = 512$)	8.04B	742.39M	0.6796	0.5066	0.3921	0.3640	0.5509	0.2571	0.0871	0.0103	0.3560	**+3.33**	-0.59
Flex-news ($r = 1024$)	8.78B	1.48B	0.6763	0.5026	0.3899	0.3607	0.5494	0.2519	0.0920	0.0080	0.3538	**+2.71**	-1.18
Flex-news ($r = 2048$)	10.27B	2.97B	0.6702	0.5016	0.3825	0.3573	0.5392	0.2486	0.0744	0.0081	0.3477	**+0.94**	-2.89
Flex-news ($r = 4096$)	13.24B	5.94B	0.6602	0.5058	0.3691	0.3554	0.5387	0.2446	0.0765	0.0060	0.3445	**+0.01**	-3.77
Flex-news ($r = 8192$)	19.18B	11.83B	0.6602	0.5058	0.3691	0.3554	0.5387	0.2446	0.0765	0.0052	0.3445	-0.01	-3.80
Flex-news ($r = 16384$)	31.06B	23.76B	0.6602	0.5058	0.3691	0.3554	0.5387	0.2446	0.0765	0.0058	0.3445	**+0.01**	-3.78

Table 20: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T Academic Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	**0.6872**	0.5108	**0.3994**	0.3570	**0.5587**	**0.2612**	0.0816	0.0085	**0.3581**	–	–
Flex-pes2o (baseline)	11.63B	4.33B	0.6732	0.5000	0.3944	0.3622	0.5479	0.2422	**0.0857**	0.0030	0.3511	–	-1.96
Flex-pes2o ($r = 1$)	7.30B	1.45M	0.6690	0.5056	0.3907	0.3399	0.5430	0.2404	0.0644	0.0086	0.3452	-1.67	-3.59
Flex-pes2o ($r = 2$)	7.30B	2.90M	0.6788	0.5048	0.3920	0.3481	0.5513	0.2474	0.0738	**0.0089**	0.3506	-0.12	-2.07
Flex-pes2o ($r = 4$)	7.30B	5.80M	0.6783	0.5058	0.3834	0.3455	0.5476	0.2433	0.0715	0.0087	0.3480	-0.87	-2.81
Flex-pes2o ($r = 8$)	7.31B	11.60M	0.6803	0.5072	0.3819	0.3508	0.5474	0.2418	0.0767	0.0088	0.3494	-0.48	-2.43
Flex-pes2o ($r = 16$)	7.32B	23.20M	0.6784	0.5056	0.3863	0.3515	0.5443	0.2402	0.0736	0.0071	0.3484	-0.76	-2.70
Flex-pes2o ($r = 32$)	7.35B	46.40M	0.6768	0.5072	0.3903	0.3499	0.5391	0.2404	0.0697	0.0059	0.3474	-1.04	-2.97
Flex-pes2o ($r = 64$)	7.39B	92.80M	0.6672	0.5072	0.3804	0.3482	0.5318	0.2373	0.0669	0.0039	0.3429	-2.33	-4.24
Flex-pes2o ($r = 128$)	7.48B	185.60M	0.6684	0.5064	0.3833	0.3504	0.5313	0.2386	0.0686	0.0028	0.3437	-2.09	-4.00
Flex-pes2o ($r = 256$)	7.67B	371.20M	0.6732	0.5092	0.3847	0.3507	0.5341	0.2419	0.0672	0.0035	0.3456	-1.56	-3.49
Flex-pes2o ($r = 512$)	8.04B	742.39M	0.6742	0.5126	0.3867	0.3508	0.5349	0.2460	0.0634	0.0017	0.3463	-1.36	-3.29
Flex-pes2o ($r = 1024$)	8.78B	1.48B	0.6768	**0.5132**	0.3949	0.3469	0.5437	0.2523	0.0796	0.0029	0.3513	**+0.06**	-1.90
Flex-pes2o ($r = 2048$)	10.27B	2.97B	0.6710	0.5104	0.3928	**0.3653**	0.5486	0.2515	0.0797	0.0025	0.3527	**+0.48**	-1.49
Flex-pes2o ($r = 4096$)	13.24B	5.94B	0.6732	0.5020	0.3944	0.3622	0.5479	0.2422	**0.0857**	0.0035	0.3514	**+0.09**	-1.87
Flex-pes2o ($r = 8192$)	19.18B	11.83B	0.6732	0.5020	0.3944	0.3622	0.5479	0.2422	**0.0857**	0.0035	0.3514	**+0.09**	-1.87
Flex-pes2o ($r = 16384$)	31.06B	23.76B	0.6732	0.5020	0.3944	0.3622	0.5479	0.2422	**0.0857**	0.0035	0.3514	**+0.09**	-1.87

Table 21: **Bold** indicates best where high is better. single underline indicates best LoRA rank r $\leq$ 2048 and double underline indicates the second best score.

2x7B-1T Reddit Expert	Total	Expert	MC9	GEN5	AGIEval	BBH	MMLU	MMLU-Pro	Math2	Code4	Avg	Baseline Δ(%)	Public Δ(%)
Public base model	7.30B	–	**0.6872**	**0.5108**	**0.3994**	0.3570	**0.5587**	**0.2612**	0.0816	0.0085	**0.3581**	–	–
Flex-reddit (baseline)	11.63B	4.33B	0.6171	0.4108	0.3668	**0.3668**	0.5358	0.2350	**0.1000**	0.0370	0.3337	–	-6.81
Flex-reddit ($r = 1$)	7.30B	1.45M	0.6624	0.4900	0.3752	0.3432	0.5356	0.2322	0.0686	0.0159	0.3404	**+2.02**	-4.93
Flex-reddit ($r = 2$)	7.30B	2.90M	0.6678	0.5072	0.3849	0.3491	0.5426	0.2377	0.0909	0.0187	0.3499	**+4.85**	-2.29
Flex-reddit ($r = 4$)	7.30B	5.80M	0.6649	0.5074	0.3821	0.3437	0.5380	0.2356	0.0741	0.0183	0.3455	**+3.55**	-3.50
Flex-reddit ($r = 8$)	7.31B	11.60M	0.6601	0.5080	0.3834	0.3416	0.5330	0.2335	0.0765	0.0187	0.3444	**+3.21**	-3.82
Flex-reddit ($r = 16$)	7.32B	23.20M	0.6653	0.5080	0.3848	0.3455	0.5378	0.2368	0.0745	0.0177	0.3463	**+3.79**	-3.28
Flex-reddit ($r = 32$)	7.35B	46.40M	0.6710	0.5104	0.3938	0.3456	0.5418	0.2382	0.0747	0.0152	0.3488	**+4.55**	-2.57
Flex-reddit ($r = 64$)	7.39B	92.80M	0.6722	0.5080	0.3887	0.3519	0.5441	0.2409	0.0742	0.0156	0.3494	**+4.73**	-2.41
Flex-reddit ($r = 128$)	7.48B	185.60M	0.6814	0.5064	0.3877	0.3461	0.5408	0.2417	0.0719	0.0215	0.3497	**+4.80**	-2.34
Flex-reddit ($r = 256$)	7.67B	371.20M	0.6788	0.5022	0.3928	0.3488	0.5422	0.2457	0.0747	0.0214	0.3508	**+5.14**	-2.02
Flex-reddit ($r = 512$)	8.04B	742.39M	0.6768	0.5018	0.3983	0.3499	0.5441	0.2422	0.0737	0.0305	0.3522	**+5.54**	-1.65
Flex-reddit ($r = 1024$)	8.78B	1.48B	0.6620	0.4946	0.3922	0.3521	0.5474	0.2449	0.0934	0.0322	0.3523	**+5.60**	-1.60
Flex-reddit ($r = 2048$)	10.27B	2.97B	0.6422	0.4742	0.3796	0.3613	0.5462	0.2421	0.0967	**0.0432**	0.3482	**+4.35**	-2.76
Flex-reddit ($r = 4096$)	13.24B	5.94B	0.6171	0.4108	0.3668	**0.3668**	0.5358	0.2350	**0.1000**	0.0360	0.3335	-0.04	-6.85
Flex-reddit ($r = 8192$)	19.18B	11.83B	0.6171	0.4108	0.3668	**0.3668**	0.5358	0.2350	**0.1000**	0.0364	0.3336	-0.02	-6.84
Flex-reddit ($r = 16384$)	31.06B	23.76B	0.6171	0.4108	0.3668	**0.3668**	0.5358	0.2350	**0.1000**	0.0364	0.3336	-0.02	-6.83

Table 22: Median rank sensitivity (slope per $\log_2$ rank) across eight benchmarks, computed via linear regression using at least four rank points per task (rank 2^0 - 2^{14}). Positive values indicate an increasing fitted trend, not necessarily monotonic improvement. Slopes in this table use scores on the 0–1 scale; multiply by 100 for the percentage-point units used in the heatmap. Each row summarizes the distribution of rank sensitivity across benchmarks, where the median reflects the typical effect and the min/max capture task-dependent variability.

Model Group	median	min	max	n benchmarks
FlexMoRE 7x7B-1T				
FlexMoRE-a4	0.0031	-0.0031	0.0280	8
FlexMoRE-a7	0.0016	-0.0083	0.0173	8
FlexMoRE-a2	0.0008	-0.0042	0.0366	8
Experts				
Experts	0.0001	-0.0018	0.0068	8

E Rank-Dependent Contributions

E.1 Rank Sensitivity Analysis

To characterize how performance varies with LoRA rank, we analyze rank sensitivity using simple linear regression between evaluation score and $\log_2$ LoRA rank. For each model family and benchmark, we fit $s(r) = \alpha + \beta \log_2 r$ where $s(r)$ denotes the observed score at rank r. The coefficient β provides a coarse summary of how performance changes with increasing rank: positive values indicate an increasing fitted trend, while near-zero or negative values indicate a weak or decreasing fitted trend. This analysis uses all 15 configured ranks $r = 2^0, \ldots, 2^{14}$ for both individual experts and homogeneous FlexMoRE compositions. Slopes in the following tables use scores on the 0–1 scale; multiplying them by 100 gives the percentage-point slopes shown in Figure 3**(c)**. Detailed results are provided in Tables 22, 23, 24 and 25.

Table 23: Per-benchmark rank sensitivity, measured as the slope of a linear regression of score on $\log_2 r$, together with Pearson correlation and the number of observations. Homogeneous FlexMoRE rows use 15 rank settings ($r = 2^0, \ldots, 2^{14}$). Expert rows pool six independently trained experts across the same 15 ranks (90 expert–rank observations). Dense baselines and heterogeneous rank allocations are excluded because they do not form a comparable rank sweep. Rank sensitivity varies substantially across evaluation benchmarks and model families.

Model Group	Benchmark	Slope per $\log_2$ rank	Pearson r	n points
AGIEval				
Experts	AGIEval	0.0001	0.0385	90
FlexMoRE-a2	AGIEval	0.0006	0.3527	15
FlexMoRE-a4	AGIEval	0.0026	0.9105	15
FlexMoRE-a7	AGIEval	0.0004	0.2844	15
BBH				
Experts	BBH	0.0026	0.4765	90
FlexMoRE-a2	BBH	0.0104	0.9433	15
FlexMoRE-a4	BBH	0.0074	0.9608	15
FlexMoRE-a7	BBH	0.0036	0.9081	15
Code4				
Experts	Code4	0.0042	0.4544	90
FlexMoRE-a2	Code4	0.0099	0.8828	15
FlexMoRE-a4	Code4	0.0123	0.8684	15
FlexMoRE-a7	Code4	0.0086	0.8594	15
GEN5				
Experts	GEN5	-0.0018	-0.4110	90
FlexMoRE-a2	GEN5	-0.0042	-0.8594	15
FlexMoRE-a4	GEN5	-0.0031	-0.8410	15
FlexMoRE-a7	GEN5	-0.0083	-0.7460	15
MC9				
Experts	MC9	-0.0010	-0.3110	90
FlexMoRE-a2	MC9	-0.0025	-0.5318	15
FlexMoRE-a4	MC9	0.0029	0.8473	15
FlexMoRE-a7	MC9	0.0020	0.7162	15
MMLU				
Experts	MMLU	-0.0005	-0.2409	90
FlexMoRE-a2	MMLU	-0.0001	-0.0642	15
FlexMoRE-a4	MMLU	0.0031	0.8983	15
FlexMoRE-a7	MMLU	0.0004	0.2032	15
MMLU-Pro				
Experts	MMLU-Pro	0.0000	0.0074	90
FlexMoRE-a2	MMLU-Pro	0.0010	0.6395	15
FlexMoRE-a4	MMLU-Pro	0.0031	0.8800	15
FlexMoRE-a7	MMLU-Pro	0.0012	0.5407	15
Math2				
Experts	Math2	0.0068	0.3372	90
FlexMoRE-a2	Math2	0.0366	0.9072	15
FlexMoRE-a4	Math2	0.0280	0.8908	15
FlexMoRE-a7	Math2	0.0173	0.8806	15

Table 24: Per-expert rank sensitivity (slope per $\log_2$ rank) computed by linear regression between LoRA rank (2^0–2^{14}) and the aggregated average evaluation score (Avg_mean).

Expert	**Slope per $\log_2$ rank**	**Pearson r**	**n points**
Flex-code-2x7B-1T	0.0017	0.8920	15
Flex-creative-2x7B-1T	-0.0005	-0.6390	15
Flex-math-2x7B-1T	0.0061	0.8340	15
Flex-news-2x7B-1T	-0.0008	-0.7330	15
Flex-pes2o-2x7B-1T	0.0001	0.1610	15
Flex-reddit-2x7B-1T	0.0007	0.7160	15

Table 25: Per-expert, per-evaluation-Benchmark rank sensitivity (slope per $\log_2$ rank) computed via linear regression between LoRA rank (2^0–2^{14}) and each evaluation Benchmark's mean score.

Expert	Benchmark	Slope per $\log_2$ rank	Pearson r	n points
Code				
Flex-code-2x7B-1T	MC9	0.0000	0.0260	15
Flex-code-2x7B-1T	GEN5	-0.0027	-0.8400	15
Flex-code-2x7B-1T	AGIEval	0.0007	0.5620	15
Flex-code-2x7B-1T	BBH	0.0030	0.9800	15
Flex-code-2x7B-1T	MMLU	-0.0011	-0.9140	15
Flex-code-2x7B-1T	MMLU-Pro	-0.0001	-0.1160	15
Flex-code-2x7B-1T	Math2	0.0023	0.8380	15
Flex-code-2x7B-1T	Code4	0.0142	0.8670	15
Creative Writing				
Flex-creative-2x7B-1T	MC9	-0.0011	-0.7580	15
Flex-creative-2x7B-1T	GEN5	-0.0004	-0.7370	15
Flex-creative-2x7B-1T	AGIEval	0.0002	0.3230	15
Flex-creative-2x7B-1T	BBH	-0.0011	-0.8230	15
Flex-creative-2x7B-1T	MMLU	-0.0018	-0.9000	15
Flex-creative-2x7B-1T	MMLU-Pro	-0.0010	-0.8840	15
Flex-creative-2x7B-1T	Math2	-0.0014	-0.6610	15
Flex-creative-2x7B-1T	Code4	0.0000	0.0620	15
Math				
Flex-math-2x7B-1T	MC9	0.0010	0.8220	15
Flex-math-2x7B-1T	GEN5	-0.0010	-0.6550	15
Flex-math-2x7B-1T	AGIEval	0.0024	0.9280	15
Flex-math-2x7B-1T	BBH	0.0108	0.9230	15
Flex-math-2x7B-1T	MMLU	0.0016	0.8920	15
Flex-math-2x7B-1T	MMLU-Pro	0.0021	0.9260	15
Flex-math-2x7B-1T	Math2	0.0386	0.8910	15
Flex-math-2x7B-1T	Code4	0.0101	0.8860	15
News				
Flex-news-2x7B-1T	MC9	-0.0023	-0.9260	15
Flex-news-2x7B-1T	GEN5	0.0001	0.1730	15
Flex-news-2x7B-1T	AGIEval	-0.0026	-0.8600	15
Flex-news-2x7B-1T	BBH	-0.0001	-0.1040	15
Flex-news-2x7B-1T	MMLU	-0.0017	-0.8990	15
Flex-news-2x7B-1T	MMLU-Pro	-0.0015	-0.8960	15
Flex-news-2x7B-1T	Math2	-0.0016	-0.8370	15
Flex-news-2x7B-1T	Code4	-0.0005	-0.8790	15
Academic				
Flex-pes2o-2x7B-1T	MC9	-0.0003	-0.2790	15
Flex-pes2o-2x7B-1T	GEN5	-0.0000	-0.0580	15
Flex-pes2o-2x7B-1T	AGIEval	0.0006	0.5030	15
Flex-pes2o-2x7B-1T	BBH	0.0013	0.8160	15
Flex-pes2o-2x7B-1T	MMLU	0.0000	0.0300	15
Flex-pes2o-2x7B-1T	MMLU-Pro	0.0002	0.2400	15
Flex-pes2o-2x7B-1T	Math2	0.0011	0.6390	15
Flex-pes2o-2x7B-1T	Code4	-0.0005	-0.8490	15
Reddit				
Flex-reddit-2x7B-1T	MC9	-0.0033	-0.6510	15
Flex-reddit-2x7B-1T	GEN5	-0.0065	-0.7490	15
Flex-reddit-2x7B-1T	AGIEval	-0.0008	-0.3500	15
Flex-reddit-2x7B-1T	BBH	0.0017	0.8690	15
Flex-reddit-2x7B-1T	MMLU	0.0001	0.1190	15
Flex-reddit-2x7B-1T	MMLU-Pro	0.0003	0.2750	15
Flex-reddit-2x7B-1T	Math2	0.0019	0.7120	15
Flex-reddit-2x7B-1T	Code4	0.0018	0.8670	15

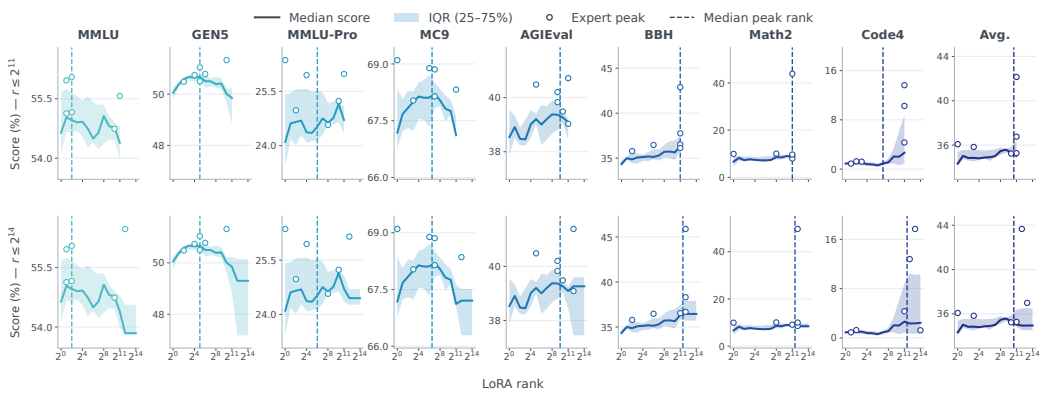

Figure 11: **Expert score curves and observed peak ranks.** Top: configured ranks $r \leq 2^{11}$; bottom: $r \leq 2^{14}$. Each panel shows the median score across the six domain-expert two-expert evaluations, with shading for the interquartile range of scores, not a confidence interval. Hollow markers denote each expert's observed peak; ties are resolved toward the lowest rank. The dashed line marks the median of the experts' peak $\log_2$ ranks, not the peak of the median score curve. Avg. includes MC9.

E.2 Typical Peak Rank by Benchmark

Because linear trends do not capture where performance saturates, we additionally report the rank at which peak performance is observed. For each expert e and benchmark g, the peak rank is defined as $r^*_{e,g} = \arg\max_{r \in \{2^0,\ldots,2^n\}} s_{e,g}(r)$, computed directly from observed scores without regression or smoothing. The main analysis in Section 5 uses $n = 14$. Figure 11 compares $n = 11$ in the top row with $n = 14$ in the bottom row. In the case of ties, the lowest rank achieving the maximum score is selected. We summarize peak-rank behavior using the median and interquartile range across experts. Results provided in Table 26.

Table 26: Typical $\log_2$ LoRA rank at which experts achieve peak performance. For each expert e and benchmark g, the peak rank is defined as $r^*_{e,g} = \arg\max_{r \in \{2^0,\ldots,2^n\}} s_{e,g}(r)$, computed directly from observed scores after sorting by score (descending) and rank (ascending), so ties are resolved by selecting the lowest rank. The table reports the median and interquartile range (25th–75th percentiles) of $\log_2 r^*_{e,g}$ across experts, together with the number of experts.

	$\mathbf{r} \leq log_2\ 2^{11}$			$\mathbf{r} \leq log_2\ 2^{14}$			
Benchmark	**median**	**q25**	**q75**	**median**	**q25**	**q75**	**n**
MMLU	2.00	1.25	8.00	2.00	1.25	8.00	6
GEN5	5.00	4.25	5.75	5.00	4.25	5.75	6
MMLU-Pro	6.00	2.50	9.50	6.00	2.50	9.50	6
MC9	6.50	3.75	7.00	6.50	3.75	7.00	6
Code4	7.00	2.25	11.00	11.50	4.25	12.75	6
AGIEval	9.50	9.00	10.75	9.50	9.00	11.50	6
BBH	11.00	7.25	11.00	11.50	7.25	12.00	6
Math2	11.00	8.75	11.00	11.50	8.75	12.00	6
Avg.	10.50	4.75	11.00	10.50	4.75	11.75	6