DECOUPLING CAPABILITIES VIA SAMPLE-LEVEL GEOMETRIC ROUTING FOR PARAMETER-EFFICIENT LOW-RESOURCE ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting reasoning-capable large language models (LLMs) to low-resource languages presents a critical bottleneck: the model must acquire language-specific knowledge and linguistic capabilities without degrading its foundational reasoning abilities. However, different tasks and individual samples can place different demands on these capabilities, making uniform adaptation particularly challenging when target-language data are scarce. Standard mixed Low-Rank Adaptation (LoRA) applies the same adaptation mechanism across samples, while existing parameter-allocation and LoRA-routing methods primarily focus on allocating parameter capacity or dynamically combining LoRA experts, rather than explicitly modeling sample-specific capability requirements. To address this challenge, we propose **S**ample-Level **A**daptive **G**eometric **R**outing (**SAGR**), a capability-centric framework that dynamically allocates fixed LoRA adaptation signals across reasoning, knowledge, and linguistic directions for each sample. By constructing sparse layer-sensitivity teachers and mapping these capability priors to individual fine-tuning samples, SAGR jointly trains a lightweight residual router with three capability-oriented LoRA branches. The router learns sample-specific capability weights and adaptation strength, enabling tailored combinations of capability-specific updates without increasing adaptation capacity. Extensive experiments on two Qwen3 reasoning models demonstrate the effectiveness over four state-of-the-art baselines. Furthermore, SAGR reveals fine-grained capability contributions of individual samples, offering an interpretable view of sample-level capability allocation.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable generalization across languages and tasks, yet adapting reasoning-capable LLMs to low-resource languages remains a fundamental challenge. Unlike high-resource settings, where abundant supervision allows models to jointly learn language-specific knowledge and reasoning abilities, low-resource adaptation must operate under extremely limited training data and parameter budgets. Consequently, new supervision often serves multiple competing objectives simultaneously: introducing domain knowledge, learning linguistic variations, and acquiring new reasoning capabilities, while sometimes also preserving the reasoning behaviors already embedded in the pretrained model. Effectively balancing these competing objectives is therefore critical for successful adaptation.

Parameter-efficient fine-tuning (PEFT) has become a dominant paradigm for adapting large language models (LLMs) in low-resource settings due to its computational efficiency and low deployment cost (Szep et al., 2024). Early PEFT methods, such as LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023), adapt all training samples within a shared adaptation space, implicitly assuming that heterogeneous supervision can be effectively captured by a common set of adaptation parameters. However, under limited supervision, individual samples often simultaneously encode reasoning, domain knowledge, and linguistic signals. Compressing these heterogeneous objectives into a shared low-rank space can lead to capability interference, where learning new knowledge degrades pretrained reasoning or linguistic representations, thereby limiting adaptation performance. To mitigate such interference, recent methods either combine multiple task-specific LoRA modules

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.

through routing or parameter merging (Zhang et al., 2023; Liu et al., 2024), or jointly optimize multiple LoRA experts within a mixture-of-experts (MoE) architecture (Dou et al., 2024; Huang et al., 2024; Zhuang et al., 2026). While effective for multi-task learning, these approaches still assume that each training sample can be assigned to a single adaptation branch. This assumption is often violated in low-resource adaptation, where a single sample may simultaneously contribute to multiple capabilities. Consequently, heterogeneous supervision remains entangled across adaptation branches, potentially leading to suboptimal capability learning.

Our empirical study illustrates this limitation: adapting a pretrained English-dominant reasoning LLM to Norwegian, a language with approximately 5 million speakers, using a small amount of Norwegian reasoning data through conventional PEFT resulted in degraded downstream reasoning performance. Furthermore, because capability specialization in MoE-based LoRA methods is learned implicitly, the semantic roles of different experts are not explicitly defined, providing limited guidance for capability-specific optimization. We argue that this limitation stems from viewing LLM adaptation primarily as a parameter allocation problem rather than a capability allocation problem. Although capability-aware adaptation is challenging because model capabilities are inherently difficult to define, disentangle, and quantify (Prabhakar et al., 2025), explicitly modeling capability allocation offers a more principled framework for handling heterogeneous supervision in low-resource adaptation.

To address these limitations, we propose **S**ample-Level **A**daptive **G**eometric **R**outing (SAGR), the first capability-centric adaptation framework designed for low-resource LLM adaptation. SAGR introduces a new perspective that formulates adaptation as a ***sample-level capability allocation problem***, where limited supervision is decomposed according to the capabilities it contributes to rather than being optimized within a shared parameter space. Instead of learning routing decisions solely from optimization objectives or task identities, SAGR first constructs sparse cross-lingual layer-sensitivity teachers to estimate how different capabilities are represented across the pretrained model. These capability priors are then transferred to individual training samples through lightweight geometric mapping, providing each sample with an explicit capability profile before optimization. During fine-tuning, a residual routing network dynamically allocates adaptation signals across three capability-specific LoRA branches—knowledge (K), reasoning (R), and linguistic adaptation (C)—allowing each sample to update the capabilities it most strongly supervises while reducing interference among heterogeneous learning objectives. We evaluate SAGR on the cross-lingual adaptation of two Qwen3 reasoning models using mixed English reasoning data and Norwegian low-resource supervision. Experiments across six reasoning, knowledge, and language understanding benchmarks demonstrate that SAGR consistently improves adaptation performance under identical parameter and training budgets. Further analyses show that the learned routing distributions align with task characteristics and reveal interpretable capability contributions of individual samples. These results demonstrate that capability-aware sample decomposition provides a more effective and interpretable paradigm for low-resource LLM adaptation than conventional shared-adapter optimization.

Specifically, our contributions are summarized as follows:

- We introduce a new formulation of low-resource LLM adaptation as a capability allocation problem, decomposing scarce and heterogeneous supervision according to the capabilities each sample contributes rather than optimizing within a shared adaptation space.
- We propose Sample-Level Adaptive Geometric Routing (SAGR), the first capability-based PEFT framework for low-resource LLM adaptation that transfers sparse layer-sensitivity capability priors into sample-level capability representations and dynamically routes adaptation signals across dedicated knowledge, reasoning, and linguistic LoRA branches.
- We develop an interpretable capability decomposition mechanism that reveals how individual samples contribute to different model capabilities, offering fine-grained insights into data composition.
- Extensive experiments demonstrate that SAGR consistently outperforms diverse LoRA-based methods, including conventional, adaptive, and mixture-based variants, under identical parameter and training budgets, while yielding interpretable routing behaviors.

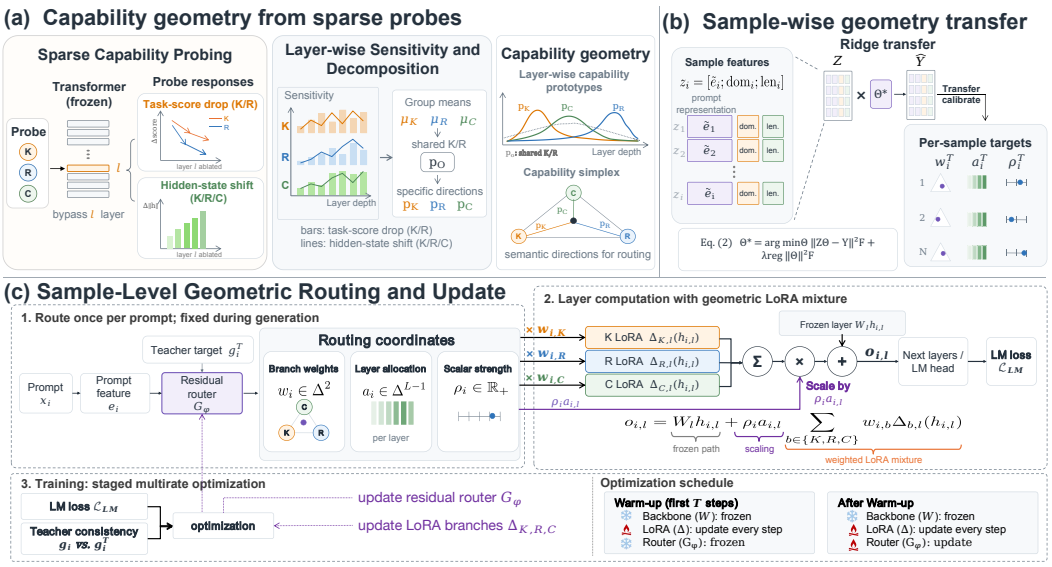


Figure 1: Overview of SAGR. (a) Layer bypassing on knowledge (K), reasoning (R), and linguistic (C) probes yields depth-wise sensitivity profiles, which are aggregated into capability prototypes p_K, p_R, and p_C and the corresponding capability simplex. (b) A ridge map transfers this geometry to adaptation samples, producing calibrated teacher targets $g_i^{\mathrm{T}} = (w_i^{\mathrm{T}}, a_i^{\mathrm{T}}, \rho_i^{\mathrm{T}})$. (c) A prompt-conditioned residual router refines these targets into $g_i = (w_i, a_i, \rho_i)$, mixes the K/R/C LoRA residuals, and scales layer-l updates by $\rho_i a_{i,l}$. Routing is computed once per prompt. LoRA branches are updated throughout training, while router updates begin after warm-up under a multirate schedule.

2 Method

2.1 Problem Setup and Overview

We consider parameter-efficient adaptation of a frozen reasoning-capable LLM f_θ to a heterogeneous, low-resource training mixture $\mathcal{D}_{\mathrm{tr}} = \{(x_i, y_i)\}$, where x_i is a prompt and y_i is its target response. The mixture combines English reasoning with Norwegian knowledge and linguistic supervision. Because individual samples can contribute differently to these capabilities, SAGR formulates adaptation as a sample-level capability allocation problem, representing each sample by a soft capability coordinate rather than assigning the entire mixture to one residual path.

SAGR attaches three parallel LoRA branches to a quantized frozen backbone (Hu et al., 2022; Dettmers et al., 2023): knowledge (K), reasoning (R), and shared linguistic adaptation (C). Together, these branches define a capability simplex, with each branch corresponding to a distinct capability direction. Each sample is represented by a nonnegative branch vector w_i with unit sum, which specifies its soft coordinate on the simplex and determines how its adaptation signal is distributed across the three capability directions. Layer weights a_i determine where the adaptation is applied, and a scalar ρ_i controls its overall strength. Thus, SAGR separates three aspects of sample-level adaptation: capability allocation, layer allocation, and overall adaptation strength.

SAGR consists of two complementary components, as summarized in Figure 1: a ***geometric teacher***, corresponding to panels (a)–(b), and a ***sample-level geometric router and routed update***, shown in panel (c). First, a ***geometric teacher*** applies sparse layer interventions to knowledge (K), reasoning (R), and linguistic (C) probes to estimate capability-specific sensitivity profiles. This sparse geometry is then transferred via regularized ridge regression to produce a geometric target for each adaptation sample, specifying its branch weights, layer allocation, and update strength. Second, a ***sample-level geometric router*** refines this geometric target from the prompt representation and determines the routing coordinates for the three LoRA branches, thereby controlling their layer-wise contributions and overall update strength. The resulting framework enables sample-specific capabil-

ity allocation within a fixed-capacity bank of LoRA branches shared across samples, while keeping the base LLM frozen.

2.2 GEOMETRIC TEACHER

Sparse capability geometry. The geometric teacher characterizes how distinct supervision types interact with the depth of the frozen model. For each probe, a layer intervention measures both behavioral degradation and displacement in downstream representations. Their normalized combination yields a sensitivity profile s_i over layers. Knowledge and reasoning probes use both signals, whereas linguistic probes use only representational displacement, which captures changes that are not naturally expressed by a discrete reasoning score.

Mean profiles from the three probe groups define capability prototypes. We subtract the common component of the knowledge and reasoning profiles to obtain their capability-specific directions, p_K and p_R. The shared linguistic prototype p_C combines the idiom profile with this common component. The affinity of probe i to branch b is

$$q_{i,b} = \max\{0, \cos(s_i, p_b)\}, \qquad b \in \mathcal{B} = \{K, R, C\}. \tag{1}$$

Here, cos denotes cosine similarity, and $q_{i,b}$ measures the alignment between probe i and capability direction b. Multiple positive affinities allow a single probe to supervise several capabilities, while the magnitude of the intervention response provides an independent target for update strength. Appendix A.1 provides full details on sensitivity-profile construction, capability prototypes, and intervention-derived update strength.

Transferring the teacher to adaptation samples. The sparse probes define the geometry, but the router requires sample-level supervision across the full adaptation set. We transfer the probe affinities and strength targets through regularized regression (Hoerl & Kennard, 1970):

$$z_i = [\tilde{e}_i; u_i; \ell_i], \qquad \Theta^* = \arg\min_{\Theta} \|Z\Theta - Y\|_F^2 + \lambda_{\text{reg}}\|\Theta\|_F^2. \tag{2}$$

The vector $\tilde{e}_i$ is a compact frozen-model prompt representation, u_i encodes the source domain, and ℓ_i contains training-example length features. Semicolons denote concatenation. Rows of Z contain centered probe features, while rows of Y contain centered affinities and strength targets. The matrix Θ contains the regression coefficients, λ_{reg} is the regularization weight, and $\|\cdot\|_F$ denotes the Frobenius norm. An intercept restores the target mean when applying the map to new examples.

The transferred affinities are calibrated against the probe geometry and normalized into branch weights w_i^{T}. They are also used to combine the layer prototypes into the layer allocation a_i^{T}, while the transferred intervention magnitude determines the update strength ρ_i^{T}. Together, these quantities form the geometric teacher target $g_i^{\mathrm{T}} = [w_i^{\mathrm{T}}; a_i^{\mathrm{T}}; \rho_i^{\mathrm{T}}]$, where the superscript T denotes teacher-derived quantities. This transfer extends the capability geometry learned from sparse interventions to the full adaptation set while preserving sample-level variation. Appendix A.1 details the affinity calibration and the construction of the branch, layer-allocation, and update-strength targets.

2.3 SAMPLE-LEVEL GEOMETRIC ROUTING

A lightweight multilayer perceptron takes the standardized prompt-last hidden state e_i as input and predicts residual corrections to the teacher target:

$$g_i = G_\phi(e_i; g_i^{\mathrm{T}}), \qquad g_i = [w_i; a_i; \rho_i]. \tag{3}$$

Here, G_ϕ denotes the residual router with trainable parameters ϕ. It predicts corrections to the branch, layer, and strength coordinates relative to the transferred teacher reference. Branch and layer corrections are normalized in log-probability space; for example, $w_i = \mathrm{softmax}(\log w_i^{\mathrm{T}} + \bar{\delta}_i^w)$, where $\bar{\delta}_i^w$ is the scaled branch-logit correction. A bounded multiplicative correction controls the update strength. Both w_i and a_i remain nonnegative and sum to one. This residual parameterization preserves the semantic organization supplied by the teacher while allowing each input to refine its own allocation. Details of the residual router parameterization and confidence-weighted teacher regularization objective can be found in Appendix A.2.

The resulting gates combine the parallel low-rank updates. For an adapted projection W_l at layer l,

$$o_{i,l} = W_l h_{i,l} + \rho_i a_{i,l} \sum_{b \in \mathcal{B}} w_{i,b} \Delta_{b,l}(h_{i,l}). \tag{4}$$

Here, $h_{i,l}$ and $o_{i,l}$ denote the projection input and output, and $\Delta_{b,l}$ denotes the LoRA residual of branch b at layer l. The product $\rho_i a_{i,l} w_{i,b}$ determines the contribution of branch b for sample i at layer l. The low-rank factors are trained while W_l remains frozen. Thus, each sample selects both a direction in the capability simplex and the depth and magnitude at which that direction modifies the frozen model. Together, (w_i, a_i, ρ_i) provide a structured characterization of each sample-level update. The capability coordinate w_i describes *what* type of capability is being adapted, the layer coordinate a_i identifies *where* in the backbone the update is applied, and the strength coordinate ρ_i determines *how strongly* the LoRA residual modifies the layer output. The K/R/C simplex gives the capability coordinate an explicit semantic interpretation, while the layer and strength coordinates expose the depth and magnitude of the adaptation.

2.4 Training Objective and Inference

We jointly optimize the low-rank branches and residual router using a teacher-regularized language-model objective:

$$\mathcal{L} = \mathbb{E}_i \left[\mathcal{L}_{\mathrm{LM}}(x_i, y_i; \theta, \mathcal{A}, g_i) + \omega_i \| g_i - g_i^{\mathrm{T}} \|_{\Lambda}^2 \right]. \tag{5}$$

Here, $\mathcal{A}$ contains the trainable LoRA parameters, while the backbone parameters θ remain frozen. The term $\mathcal{L}_{\mathrm{LM}}$ is the causal language-model loss, ω_i is the confidence of the geometric teacher, and $\|v\|_{\Lambda}^2 = v^{\top} \Lambda v$ uses a fixed diagonal matrix Λ to balance deviations in branch allocation, layer allocation, and update strength. The language-model objective couples the routed updates to the adaptation task, while the geometric regularizer anchors their capability interpretation.

Training follows a staged multirate schedule. During an initial warm-up, the router remains fixed while the LoRA branches establish their residual directions under the transferred geometry. The adapters are subsequently updated at every optimizer step, while the router is activated according to its update interval and optimized with a separate learning rate. The main configuration uses an interval of one, so the adapters and router are updated jointly after warm-up while retaining distinct optimization scales. This schedule separates the initial formation of the update space from its sample-conditioned refinement. Appendix A.4 reports the detailed training configuration, including the warm-up length, learning rates, and objective weights.

At inference, the mean transferred geometry of the corresponding training partition provides the initial reference for the router. The frozen backbone supplies the prompt representation e_i, from which the router predicts the sample-specific correction to the branch, layer, and strength coordinates. The resulting coefficients remain fixed throughout the generated response. Appendix A.3 specifies the reference construction and partition mapping; Appendix C.1 evaluates an alternative support-aware reference derived directly from the input and neighboring training examples.

3 Experiments

We evaluate SAGR from three complementary perspectives. The main comparison tests whether geometric routing improves adaptation under a matched data and optimization budget. Controlled variants isolate the contribution of the geometric teacher and residual router. Finally, routing analyses and selective continuation examine whether the learned coordinates provide an interpretable account of how individual samples shape the adapted model.

3.1 Datasets and Evaluation

To assess models performance across diverse capabilities, the training and evaluation suite includes a reasoning-intensive task set including MixChain-Z-GSM8K (Ma et al., 2025), StrategyQA, and knowledge-free reasoning KFRD arithmetic dataset (Hu et al., 2024), as well as knowledge-intensive datasets including self-collected Norwegian Instruction samples, NorTruthfulQA, NorOpenBookQA, NRK Quiz, Norwegian idiom completion and NorQuAD from NorEval (Mikhailov et al., 2025). Specifically, given the limited data availability, we reserved 200 samples from each of the

Model	Method	Math	OpenSci	OBQA	NRK	Idioms	NoCoLA	Macro6	Acc5
Qwen3-8B	Frozen base	65.0	44.5	85.0	59.0	7.0	16.37	46.15	52.10
	LoRA ($r = 8$)	71.50	53.00	76.50	57.00	5.50	17.53	46.84	52.70
	LoRA ($r = 24$)	71.00	54.50	78.00	54.00	2.50	27.10	47.85	52.00
	AdaLoRA (target $r = 24$)	76.00	50.50	**82.00**	56.50	7.00	15.47	47.91	54.40
	MoE-LoRA (Top-2, 3E)	72.00	51.50	80.00	58.50	7.00	17.38	47.73	53.80
	MoE-LoRA (Top-3, 3E)	74.50	50.50	80.00	54.00	7.00	16.50	47.08	53.20
	LoRAMoE (Dense, 4E)	33.00	51.50	79.00	52.00	7.50	37.01	43.34	44.60
	LoRAMoE (Top-2, 4E)	33.50	49.50	81.00	53.00	6.50	**37.39**	43.48	44.70
	LoRAMoE (Top-3, 4E)	33.50	50.50	74.50	43.00	6.50	36.06	40.68	41.60
	SAGR ($3 \times$ rank 8)	**78.50**	**55.00**	78.00	**60.00**	**8.00**	16.32	**49.30**	**55.90**
Qwen3-4B	Frozen base	68.00	49.50	79.50	47.00	4.00	16.38	44.06	49.60
	LoRA ($r = 8$)	67.00	49.00	79.50	55.00	4.00	18.97	45.58	50.90
	LoRA ($r = 24$)	57.00	**54.50**	76.50	52.50	5.00	16.73	43.71	49.10
	AdaLoRA (target $r = 24$)	67.50	51.00	77.50	46.00	3.50	20.10	44.27	49.10
	MoE-LoRA (Top-2, 3E)	66.00	48.50	78.50	52.50	3.00	15.67	44.03	49.70
	MoE-LoRA (Top-3, 3E)	62.50	47.00	77.50	49.00	3.50	15.15	42.44	47.90
	LoRAMoE (Dense, 4E)	26.50	48.50	71.00	46.00	8.00	26.32	37.72	40.00
	LoRAMoE (Top-2, 4E)	29.00	49.50	70.50	47.50	**9.00**	**28.27**	38.96	41.10
	LoRAMoE (Top-3, 4E)	29.00	48.00	71.00	46.00	7.00	22.78	37.30	40.20
	SAGR ($3 \times$ rank 8)	**69.50**	49.50	**81.00**	**55.50**	3.00	24.45	**47.16**	**51.70**

Table 1: Qwen3-8B/4B adaptation with 200 examples per task and the same one-epoch training mixture. MoE-LoRA uses three rank-8 prompt-routed Q/V experts; LoRAMoE uses four token-routed FFN experts with Dense, Top-2, or Top-3 routing. NoCoLA reports ERRANT F0.5; Macro6 averages all six tasks and Acc5 the five accuracy tasks. Bold/underline denote the best/second-best adapted scores.

MixChain, NRK, Idiom, and OpenBookQA datasets for evaluation. From the training data, we randomly sampled 50 examples from each of the Idiom, Instruction, NorOpenBookQA, NorQuAD, MixChain, and KFRD datasets to train the teacher model. To further evaluate the model's generalization beyond the training data, we additionally include a mixed-capability dataset, OpenScience[1] and a semantically intensive dataset NoCola from NorEval. All tasks are evaluated using accuracy, except for NoCoLA, for which corpus-level ERRANT $F_{0.5}$ (Bryant et al., 2017) is reported. Macro-averaged performance is also reported to assess the model's overall performance across all six evaluation tasks. Appendix B reports the complete data composition.

3.2 Training protocol and competitive baselines

We compare SAGR against four representative parameter-efficient fine-tuning baselines: LoRA (Hu et al., 2021), AdaLoRA (Zhang et al., 2023), MoE-LoRA (Zhuang et al., 2026), and LoRAMoE (Dou et al., 2024). AdaLoRA employs adaptive rank allocation, while MoE-LoRA and LoRAMoE are router-based MoE approaches that use prompt-level and token-level routing strategies, respectively.

The base models are Qwen3-8B and Qwen3-4B. We set the LoRA rank to 8 and 24, and the AdaLoRA rank to 24, consistent with our experimental configurations. Similarly, the MoE-LoRA configuration follows the SAGR setup, using three rank-8 experts. For each prompt, we evaluate both top-2 and top-3 expert selection. For LoRAMoE, we evaluate the four-expert configuration adopted in the original paper, evaluating both all four experts and the first three experts to facilitate a fair comparison with SAGR. Full optimization and decoding settings appear in Appendix A.4.

3.3 Main Results

Experimental results are reported in Table 1. As can be observed, SAGR consistently outperform all baselines across most benchmarks. The results demonstrate that, for a mature pretrained model, parameter-efficient fine-tuning with limited training data does not necessarily improve the targeted

[1] https://huggingface.co/datasets/nvidia/OpenScienceReasoning-2

capabilities and may, in some cases, degrade existing capabilities. This highlights an important trade-off in low-resource adaptation: given a fixed model and limited supervision, it is necessary to balance the preservation of existing capabilities against the acquisition of new ones. In our experiments, LoRAMoE improves certain knowledge-intensive capabilities, but this improvement is accompanied by a substantial degradation in mathematical performance. In contrast, SAGR achieves a more balanced performance profile across the evaluated capabilities, better preserving existing capabilities while enhancing the targeted ones. We also observe that increasing the number of experts does not necessarily improve performance. Across the six LoRAMoE configurations evaluated on the two base models, the results suggest that experts may compete with one another, potentially interfering with and weakening individual capabilities.

3.4 Ablation and Analysis

Teacher and router contributions. We conduct an ablation study by removing geometric calibration and prompt-conditioned refinement. As shown in Table 2, removing calibration and the residual router reduces Macro6 from 49.30 to 47.34 and 47.42, respectively. A neutral three-branch router achieves 48.59, while adding the geometry teacher improves it to 49.30, with reasoning increasing from 63.50 to 66.75. These results highlight the importance of geometric guidance and flexible multi-branch updates for capability-aware adaptation.

Configuration	Reason	Local	Lang	Macro6
No calibration	64.50	65.75	11.77	47.34
Teacher only	61.75	68.00	12.50	47.42
Neutral teacher	63.50	69.00	13.26	48.59
Full SAGR	66.75	69.00	12.16	49.30

Table 2: Routing controls on Qwen3-8B. No calibration uses the uncalibrated teacher; teacher-only inference disables the residual router; neutral teacher retrains the three SAGR branches without geometric targets or teacher alignment.

Branch Decomposition To further investigate whether the number of branches affects performance, we conducted experiments with 1–5 branches. A single branch corresponds to standard LoRA. The results are presented in Table 3. Among the configurations with 1–5 branches and rank 8, the three-branch K/R/C decomposition achieves the highest Macro6 score of 49.30. Using fewer branches may merge distinct types of supervision, whereas finer-grained partitioning—for example, separating local from general knowledge or mathematical from general reasoning—may disperse the limited supervision across individual branches. These findings further support the results reported in Table 1, suggesting that an appropriate degree of capability decomposition is important for effective low-resource adaptation.

Branches	Math	OpenSci	OBQA	NRK	Idioms	NoCoLA	Macro6
1: Universal	77.5	51.5	78.0	56.0	7.0	16.55	47.76
2: K/R	77.0	54.5	78.0	59.5	5.5	17.94	48.74
3: K/R/C	78.5	55.0	78.0	60.0	8.0	16.32	49.30
4: KL/KG/R/C	76.5	49.5	82.5	60.5	7.0	17.26	48.88
5: KL/KG/RM/RG/C	77.0	49.5	82.5	60.5	5.5	15.27	48.38

Table 3: Branch-count comparison for Qwen3-8B. Each branch has rank 8. KL/KG split local and general knowledge; RM/RG split mathematical and general reasoning. All rows use the same training mixture and epoch count.

Input-conditioned reference. We further derive the geometric reference directly from prompt representations and neighboring training samples. This input-only route reaches 55.20 Acc5 and 48.80 Macro6, close to 55.90 and 49.30 for the calibrated reference. The result shows that the learned geometry can be recovered from the input itself while retaining sample-specific routing behavior. Appendix C.1 gives the complete construction.

Reference	Acc5	NoCoLA $F_{0.5}$	Macro6
Calibrated	55.90	16.32	49.30
Input-only	55.20	16.78	48.80

Table 4: Alternative inference references for the same Qwen3-8B SAGR model. The input-only variant derives its reference from prompt similarity and training-set support.

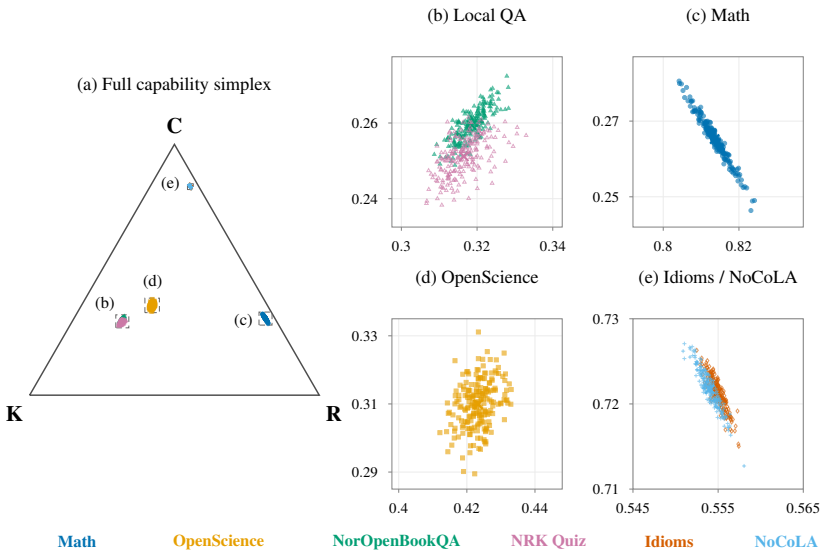


Figure 2: Sample-level routing geometry. Panel (a) shows all prompts on the K/R/C simplex; panels (b)–(e) enlarge selected task regions, with crosses marking task means. The zoomed views reveal within-task variation across semantically distinct routing regions.

3.5 Task-conditioned routing.

Sample-level routing geometry. Figure 2 reveals clear task-dependent routing structure: Math is R-dominant; NorOpenBookQA and NRK are K-dominant; Idioms and NoCoLA are strongly C-dominant; and OpenScience occupies a mixed K/R/C region. The simplex also reveals substantial within-task variation, showing that routing is sample-specific rather than determined solely by task identity. Appendix C reports the corresponding task-level means.

Task-Conditioned Branch Allocation Table 5 averages the branch weights over 200 inputs per task. The weights combine the calibrated geometric reference and the prompt-conditioned correction defined in Appendix A. Math is R-dominant, local question answering is K-dominant, and idiom completion and grammatical correction are C-dominant. These averages describe the learned allocation; the continuation experiment provides the intervention on branch-specific adaptation. OpenScience distributes its mass more evenly, with 39.8% on K, 24.4% on R, and 35.8% on C.

Task	w_K	w_R	w_C
Math	3.3	66.0	30.6
OpenScience	39.8	24.4	35.8
NorOpenBookQA	53.2	16.9	30.0
NRK Quiz	53.8	17.2	29.0
Idioms	2.9	13.8	83.3
NoCoLA	2.9	13.7	83.3

Table 5: Mean branch weights (%) for Qwen3-8B over 200 inference routes per task; rows may not sum to 100 due to rounding.

4 Related Work

4.1 Multilingual Reasoning and Low-Resource Language Adaptation

Recent studies show that language and reasoning can be partially disentangled in multilingual LLMs. Zhao et al. (2026) identify distinct language-specific and reasoning representations across layers, while Hu et al. (2025) find that knowledge-free reasoning exhibits substantial cross-lingual representational overlap, whereas knowledge-related representations are more language-dependent. Tran et al. (2026) further demonstrate that explicitly separating language understanding from reasoning through English-pivoted chain-of-thought can facilitate reasoning transfer to an extremely low-

resource language. These findings motivate separating linguistic adaptation from reasoning preservation when adapting reasoning-capable models to low-resource languages.

Empirical evidence also suggests that language adaptation can improve target-language generation while affecting pretrained reasoning abilities (Khade et al., 2025). Existing approaches address this problem through language-specific and task-specific adapters, such as MAD-X (Pfeiffer et al., 2020) and AdapterFusion (Pfeiffer et al., 2021), as well as approaches targeting unseen languages (Chen & Chen, 2024) and capability preservation (Gurgurov et al., 2025; Lee et al., 2025; Sarasua et al., 2025). These approaches primarily organize adaptation at the language, task, or model level. However, adaptation involves heterogeneous capability requirements: different tasks can place different relative demands on linguistic adaptation, local knowledge acquisition, and reasoning preservation, with these demands varying across individual samples within and across tasks. SAGR explicitly models this heterogeneity at the sample level, learning *sample-level capability allocation* among linguistic, knowledge, and reasoning directions during parameter-efficient adaptation.

4.2 Parameter-Efficient Adaptation and LoRA Routing

Parameter-efficient methods such as LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023) adapt LLMs by training a small number of additional parameters. AdaLoRA (Zhang et al., 2023) and ALoRA (Liu et al., 2024) further adapt the allocation of parameter or rank capacity across model components. These approaches focus on parameter capacity, determining where and how much adaptation capacity to allocate, whereas SAGR keeps the *parameter capacity fixed* and learns how to distribute adaptation signals across capability directions for each sample.

Another line of work composes or routes among multiple LoRA modules. LoRAHub (Huang et al., 2024) combines pretrained LoRA modules through learned module-level weights, while LoRAMoE (Dou et al., 2024) and Mixture of LoRA Experts (MoLE) (Huang & Wei, 2024) introduce routing mechanisms to dynamically combine multiple LoRA experts. More recent work further explores input-aware routing among LoRA experts (Zhuang et al., 2026). These methods primarily focus on dynamically selecting or combining expert updates.

SAGR instead introduces an *interpretable capability decomposition* in which its LoRA branches are jointly trained to represent three capability directions—linguistic adaptation, local knowledge acquisition, and reasoning—rather than independently trained task or domain experts. A geometric teacher provides capability priors from layer sensitivity, which are refined by a sample-aware router to determine the contribution of each capability for each sample. Thus, SAGR uses routing not only to dynamically combine adaptation signals, but also to reveal how individual samples contribute to different capabilities, providing an interpretable view of the adaptation data composition.

5 Conclusion

In this work, we proposed Sample-Level Adaptive Geometric Routing (SAGR), a capability-centric PEFT framework that reframes low- and extremely-low-resource LLM adaptation as a capability allocation problem. By transferring cross-lingual layer-sensitivity priors into sample-level capability representations, SAGR dynamically routes adaptation signals across knowledge, reasoning, and linguistic LoRA branches, reducing interference among heterogeneous supervision. Experiments on cross-lingual adaptation demonstrate that SAGR consistently improves performance over conventional LoRA-based approaches under identical training budgets while providing interpretable routing behaviors. These results highlight the potential of capability-aware adaptation for efficient and controllable low-resource LLM customization. Future work will explore extending SAGR to broader model architectures, multilingual settings, and more diverse capability dimensions.

AI use statement

In the preparation of this manuscript, generative AI tools were used exclusively for language polishing and grammar checking to improve the readability of the text. All technical ideas, methodological designs, mathematical formulations, and experimental results were independently conceived, developed, and verified by the authors.

Reproducibility statement

We provide detailed information to support reproducibility of our experiments. The main text describes the SAGR architecture, training objective, routing mechanism, and evaluation protocol, while the appendix provides the complete geometric teacher construction, sample-level transfer procedure, implementation details, hyperparameters, data composition, and additional ablations. The source code is included in the Supplementary Material to facilitate reproduction of the reported results.

References

Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, pp. 793–805, 2017.

Po-Heng Chen and Yun-Nung Chen. Efficient unseen language adaptation for multilingual pretrained language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18983–18994, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen Zhao, Yuhao Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. LoRAMoE: Alleviate world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5729–5752, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.309. URL https://aclanthology.org/2024.acl-long.309/.

Daniil Gurgurov, Ivan Vykopal, Josef Van Genabith, and Simon Ostermann. Small models, big impact: Efficient corpus and graph-based adaptation of small multilingual language models for low-resource languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 355–395, 2025.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

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

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

Peng Hu, Sizhe Liu, Changjiang Gao, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. Large language models are cross-lingual knowledge-free reasoners, 2024.

Peng Hu, Sizhe Liu, Changjiang Gao, Xin Huang, Xue Han, Junlan Feng, Chao Deng, and Shujian Huang. Large language models are cross-lingual knowledge-free reasoners. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1525–1542, 2025.

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

Shaohan Huang and Furu Wei. Mixture of lora experts. In *ICLR 2024*, April 2024. URL https://www.microsoft.com/en-us/research/publication/mixture-of-lora-experts/.

Omkar Khade, Shruti Jagdale, Abhishek Phaltankar, Gauri Takalikar, and Raviraj Joshi. Challenges in adapting multilingual llms to low-resource languages using lora peft tuning. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pp. 217–222, 2025.

Jungseob Lee, Seongtae Hong, Hyeonseok Moon, and Heui-Seok Lim. Cross-lingual optimization for language transfer in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15100–15119, 2025.

Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. Alora: Allocating low-rank adaptation for fine-tuning large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 622–641, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.35. URL https://aclanthology.org/2024.naacl-long.35/.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035, 2025.

Vladislav Mikhailov, Tita Enstad, David Samuel, Hans Christian Farsethås, Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. NorEval: A Norwegian language understanding and generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3495–3541, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.181. URL https://aclanthology.org/2025.findings-acl.181/.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 7654–7673, 2020.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume*, pp. 487–503, 2021.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. Lora soups: Merging loras for practical skill composition tasks. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pp. 644–655, 2025.

Ixak Sarasua, Ander Corral, and Xabier Saralegi. DIPLomA: Efficient adaptation of instructed LLMs to low-resource languages via post-training delta merging. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24898–24912, Suzhou, China, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.1355/.

Marton Szep, Daniel Rueckert, Rüdiger von Eisenhart-Rothe, and Florian Hinterwimmer. Fine-tuning large language models with limited data: a survey and practical guide. *arXiv preprint arXiv:2411.09539*, 2024.

Khanh-Tung Tran, Barry O'Sullivan, and Hoang D Nguyen. Reasoning transfer for an extremely low-resource and endangered language: Bridging languages through sample-efficient language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33277–33286, 2026.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Weixiang Zhao, Jiahe Guo, Yang Deng, Tongtong Wu, Wenxuan Zhang, Yulin Hu, Xingyu Sui, Yanyan Zhao, Wanxiang Che, Bing Qin, et al. When less language is more: Language-reasoning disentanglement makes llms better multilingual reasoners. *Advances in Neural Information Processing Systems*, 38:38608–38642, 2026.

Yuan Zhuang, Yi Shen, Yuexin Bian, Qing Su, Shihao Ji, Yuanyuan Shi, and Fei Miao. Ld-mole: Learnable dynamic routing for mixture of lora experts. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZTjM1gH0ug.

APPENDIX

A IMPLEMENTATION DETAILS OF SAGR

This appendix specifies the implementation used in the main Qwen3-8B experiments. SAGR comprises three capability-specific LoRA branches and a residual router supervised by fixed geometric targets constructed before adaptation. We define the teacher targets and residual routing below; Table 7 collects the numerical settings.

A.1 GEOMETRIC TEACHER CONSTRUCTION

The teacher assigns each adaptation example a target $g_i^{\mathrm{T}} = [w_i^{\mathrm{T}}; a_i^{\mathrm{T}}; \rho_i^{\mathrm{T}}]$ over capability weights, layer allocation, and update strength. Sparse probes first define the capability geometry, which is then transferred to the complete adaptation set.

Sparse capability profiles. The layer profile is

$$s_i = \begin{cases} \mathcal{N}\big(\alpha\bar{\Delta}_i^{\mathrm{task}} + (1-\alpha)\bar{\Delta}_i^{\mathrm{hid}}\big), & i \in \mathcal{P}_K \cup \mathcal{P}_R, \\ \mathcal{N}(\bar{\Delta}_i^{\mathrm{hid}}), & i \in \mathcal{P}_C. \end{cases} \tag{6}$$

The sets $\mathcal{P}_K$, $\mathcal{P}_R$, and $\mathcal{P}_C$ contain the knowledge, reasoning, and linguistic probes. The vectors $\bar{\Delta}_i^{\mathrm{task}}$ and $\bar{\Delta}_i^{\mathrm{hid}}$ contain their nonnegative, maximum-normalized task degradation and hidden-state change across layers, and α combines the two signals. The operator $\mathcal{N}(v) = v/\sum_l v_l$ normalizes a nonnegative vector to unit mass.

We separate shared layer sensitivity from the K- and R-specific profiles. Let μ_K, μ_R, and μ_C be the mean profiles in the three probe groups. Their prototypes are

$$\begin{aligned} p_O &= \mathcal{N}\big(\sqrt{\mu_K \odot \mu_R}\big), \\ p_K &= \mathcal{N}([\mu_K - p_O]_+), \qquad p_R = \mathcal{N}([\mu_R - p_O]_+), \\ p_C &= \mathcal{N}\big(\eta\mu_C + (1-\eta)p_O\big). \end{aligned} \tag{7}$$

Here, p_O captures shared K/R sensitivity, $\odot$ denotes elementwise multiplication, $[\cdot]_+$ clips negative entries to zero, and η weights the linguistic profile in the shared prototype. The resulting p_K, p_R, and p_C define the capability directions used by Equation 1.

Each intervention bypasses one decoder layer. For probe i and intervention layer l, task degradation is the original task score minus the score after the intervention. Downstream hidden change is

$$\Delta_{i,l}^{\mathrm{hid}} = \frac{1}{L-l} \sum_{k=l}^{L-1} \max\{0, 1 - \cos(\bar{h}_{i,k}, \bar{h}_{i,k}^{(-l)})\}. \tag{8}$$

Here, layers are indexed from zero, L is their number, $\bar{h}_{i,k}$ is the collector's mean hidden representation at layer k, and $\bar{h}_{i,k}^{(-l)}$ is its counterpart with layer l bypassed. Positive task and representation changes are normalized before applying Equation 6. The resulting probe set contains 155 knowledge, 95 reasoning, and 50 linguistic examples, balanced across the source groups.

The mean of the three largest positive intervention responses supplies a normalized magnitude $m_i \in [0, 1]$. A linear map converts m_i into the teacher strength ρ_i^{probe} and confidence $\omega_i^{\mathrm{probe}}$, separating update direction from its magnitude.

Sample-level transfer. Equation 2 fits centered probe features to centered capability affinities and strength targets. The intercept restores their mean when predicting teacher quantities for adaptation examples. The transfer features combine a compact frozen prompt representation with source indicators and length descriptors. A fixed calibration operator blends each prediction with its partition-level probe mean and constructs the branch and layer targets:

$$\begin{aligned} \widetilde{q}_i &= (1-\gamma)\widehat{q}_i + \gamma\bar{q}_{\pi(i)}, \\ w_i^{\mathrm{T}} &= \mathrm{softmax}\big(\log(c \odot \widetilde{q}_i + \epsilon) + \beta_{\pi(i)}\big), \\ a_i^{\mathrm{T}} &= \mathcal{N}\left(\sum_b \widetilde{q}_{i,b} p_b\right). \end{aligned} \tag{9}$$

Here, $\widehat{q}_i$ is the regression prediction, $\pi(i)$ identifies the training partition, $\bar{q}_{\pi(i)}$ is its mean probe affinity, γ is the blend weight, c is a branch scale, and $\beta_{\pi(i)}$ is a partition-level branch bias. The constant $\epsilon > 0$ protects the logarithm. Affinities are clipped to $[0, 1]$, and the transferred magnitude supplies ρ_i^{T}. This yields one fixed geometric target for every adaptation example.

A.2 Residual Routing and Optimization

A two-layer perceptron reads the standardized prompt-last hidden state and predicts residuals $(\delta_i^w, \delta_i^a, \delta_i^\rho)$ around the geometric teacher:

$$\begin{aligned} w_i &= \mathrm{softmax}(\log w_i^{\mathrm{T}} + \tau_w \delta_i^w), \\ a_i &= \mathrm{softmax}(\log a_i^{\mathrm{T}} + \tau_a \delta_i^a), \\ \rho_i &= \mathrm{clip}\big(\rho_i^{\mathrm{T}} \exp(\tau_\rho \tanh \delta_i^\rho), \rho_{\min}, \rho_{\max}\big)\,. \end{aligned} \tag{10}$$

The constants τ_w, τ_a, and τ_ρ control the residual scales; $\rho_{\min}$ and $\rho_{\max}$ bound the strength. Teacher probabilities are floored and renormalized before taking logarithms. This parameterization preserves the teacher's initial allocation and lets each prompt adjust it.

Expanding the weighted norm in Equation 5 gives the confidence-weighted geometric regularizer:

$$\mathcal{L}_{\mathrm{T}} = \frac{1}{N} \sum_{i=1}^{N} \omega_i \left[\frac{\lambda_w}{3} \|w_i - w_i^{\mathrm{T}}\|_2^2 + \frac{\lambda_a}{L} \|a_i - a_i^{\mathrm{T}}\|_2^2 + \lambda_\rho (\rho_i - \rho_i^{\mathrm{T}})^2 \right]. \tag{11}$$

Here, N is the batch size, L is the number of layers, ω_i is teacher confidence, and λ_w, λ_a, and λ_ρ weight the three components. The optimized objective is $\mathcal{L} = \mathcal{L}_{\mathrm{LM}} + \mathcal{L}_{\mathrm{T}}$, which is equivalent to Equation 5. The configuration and optimization schedule are summarized in Table 7.

A.3 Inference

The calibrated reference is the mean of the projected teachers in the corresponding training partition. The mapping is shown in Table 6. The frozen model then supplies the prompt representation, and the residual router predicts one set of branch, layer, and strength coefficients that remains fixed throughout the response.

Evaluation task	Training reference partition
Math	Math
OpenScience	Instruction knowledge
NorOpenBookQA, NRK Quiz	Knowledge QA (NorOpenBookQA domain)
Idioms, NoCoLA	Idioms

Table 6: Evaluation-to-training-partition mapping for the calibrated geometric reference. The knowledge-QA partition includes NRK training examples.

A.4 Experimental Configuration

Component	Setting	Value
Teacher	Probe counts (K/R/C)	155/95/50
Teacher	Sensitivity mixture $\alpha/(1-\alpha)$	0.9/0.1
Teacher	Shared linguistic weight η	0.7
Teacher	Strength / confidence ranges	$[0.95, 1.05]$ / $[0.55, 1]$
Transfer	Representation / ridge coefficient	SVD-128 / 10
Transfer	Blend γ / branch scale c	0.7 / $(1, 1, 1.2)$
Router	Hidden width / residual scales	256 / $(0.2, 0.2, 0.1)$
Adapters	Branches / rank / projections	3 / 8 / query, value
Optimization	LoRA / router learning rate	10^{-4} / 10^{-5}
Optimization	Alignment weights $(\lambda_w, \lambda_a, \lambda_\rho)$	$(0.05, 0.10, 0.02)$
Optimization	Strength range / warm-up	$[0.9, 1.1]$ / 50 steps
Optimization	Epoch / optimizer steps / seed	1 / 1,420 / 42

Table 7: Configuration used for the main Qwen3-8B experiments. Numerical settings are fixed across the primary SAGR comparisons.

Comparison consistency and capacity. All Qwen3-8B comparisons use the same Hugging Face evaluation pipeline, tokenizer-native chat format, thinking behavior, deterministic decoding, generation limit, examples, and parsers. A rank-8 adapter contains 3.83M trainable parameters; the three SAGR branches contain 11.50M, matching a rank-24 LoRA, and the router adds 1.05M parameters. All methods are trained on the same 22,711 examples for one epoch (1,420 optimizer steps), so the central comparison holds data exposure and optimization budget fixed.

B Data Composition

Source	Train	Validation
Norwegian idioms	3,180	353
Article-based instruction QA	3,229	359
Other Norwegian instruction QA	2,359	262
NorTruthfulQA	394	44
NRK Quiz	2,643	294
English math with solutions	5,706	634
Other English math	4,600	511
English StrategyQA	445	49
English knowledge-free reasoning	155	17
Total	22,711	2,523

Table 8: Main adaptation mixture, counted from the processed source manifest.

The 22,711-example mixture retains Norwegian knowledge examples and reconstructs the reasoning portion in English while preserving the processed train/validation assignments. Of its reasoning examples, 5,706 include an English solution or rationale plus the final answer. The remaining 5,200 use an English question and a final-answer target. Norwegian evaluation prompts for Math and OpenScience use the translated input fields. Evaluation language and reasoning-training language are thus distinct parts of the design.

The probe pool is drawn from separate annotation sources. Its knowledge portion contains 58 instruction-knowledge, 42 NorOpenBookQA, and 55 NorQuAD probes; its reasoning portion contains 46 math and 49 knowledge-free reasoning probes; its linguistic portion contains 50 idiom probes. Selection is stratified by source according to the predefined split metadata.

C Seed Variation and Routing Distributions

C.1 Input-Conditioned Reference

The input-only analysis replaces the calibrated task reference with a prior retrieved from the adaptation set. For prompt x, the eight nearest training teachers are weighted by a temperature-softmax over frozen prompt representations. Writing the retrieved prior as $P_{\mathrm{ret}}(x)$ and the global training prior as P_{global}, the support-aware reference is

$$P_{\mathrm{auto}}(x) = \kappa(x)P_{\mathrm{ret}}(x) + (1 - \kappa(x))P_{\mathrm{global}}. \tag{12}$$

Here, $\kappa(x) \in [0, 1]$ is predicted from retrieval-support features fitted on validation teacher reconstruction. Branch and layer distributions are renormalized after averaging, and strength is averaged in log space. The index, global prior, and reliability model use training and validation data only.

C.2 Training-Seed Variation

We compare SAGR with total-rank-matched LoRA at seeds 7, 42, and 314. Each pair uses the same training mixture, one-epoch schedule, and fixed 200-example evaluation slice per task. Table 9 reports the complete task results, and Table 10 summarizes variation across seeds.

Method	Seed	Math	OpenSci	OBQA	NRK	Idioms	NoCoLA	Acc5	Macro6
SAGR	7	68.0	48.0	80.0	62.5	9.5	17.18	53.60	47.53
SAGR	42	78.5	55.0	78.0	60.0	8.0	16.32	55.90	49.30
SAGR	314	75.5	53.5	78.5	62.5	7.0	15.27	55.40	48.71
LoRA ($r = 24$)	7	68.5	52.0	79.0	54.0	6.0	32.21	51.90	48.62
LoRA ($r = 24$)	42	71.0	54.5	78.0	54.0	2.5	27.10	52.00	47.85
LoRA ($r = 24$)	314	70.5	49.0	68.5	50.0	6.0	24.91	48.80	44.82

Table 9: Complete SAGR and rank-24 LoRA results across three training seeds. All scores are on a 0–100 scale. NoCoLA uses ERRANT $F_{0.5}$; the remaining tasks use accuracy. Acc5 and Macro6 are computed before rounding.

Method	Acc5	NoCoLA $F_{0.5}$	Macro6
SAGR	54.97 ± 1.21	16.26 ± 0.96	48.52 ± 0.90
LoRA ($r = 24$)	50.90 ± 1.82	28.07 ± 3.75	47.10 ± 2.01

Table 10: Mean and sample standard deviation across seeds 7, 42, and 314 on the fixed evaluation slice.

SAGR improves Acc5 by 1.70, 3.90, and 6.60 points in seed order, with a mean gain of 4.07. Its Macro6 differences are -1.09, $+1.45$, and $+3.89$, averaging $+1.42$. Averaged over seeds, the task-level gains are 4.00 on Math, 0.33 on OpenScience, 3.67 on NorOpenBookQA, 9.00 on NRK, and 3.33 on Idioms. Rank-24 LoRA is stronger on NoCoLA, averaging 28.07 versus 16.26.

Supplementary rank-8 comparison. Table 11 retains the comparison with the fully mixed rank-8 LoRA baseline. SAGR improves Acc5 and Macro6 in all three matched pairs.

Metric	Seed	LoRA ($r = 8$)	SAGR	Paired gain
Acc5	7	50.80	53.60	+2.80
	42	52.70	55.90	+3.20
	314	47.90	55.40	+7.50
	Mean	50.47	54.97	+4.50
NoCoLA $F_{0.5}$	7	4.69	17.18	+12.49
	42	17.53	16.32	−1.21
	314	21.48	15.27	−6.21
	Mean	14.57	16.26	+1.69
Macro6	7	43.12	47.53	+4.42
	42	46.84	49.30	+2.47
	314	43.50	48.71	+5.22
	Mean	44.48	48.52	+4.03

Table 11: Supplementary paired comparison with ordinary rank-8 LoRA. Acc5 is the primary accuracy aggregate; NoCoLA and Macro6 are reported separately.

Task	Repaired	Introduced	Net
Idioms	10	5	+5
Math	28	14	+14
NorOpenBookQA	17	14	+3
NRK Quiz	22	16	+6
OpenScience	18	14	+4
Total	95	63	+32

Table 12: Prediction flips from ordinary rank-8 LoRA to SAGR-3 on the five accuracy tasks at seed 42.

Paired Outcome Shifts The paired outputs further reveal a qualitative shift in response behavior. Among the ordinary-LoRA errors corrected by SAGR, 27 omit an explicit final answer, 21 are unusually long, and 11 contain repetitive loops. Ninety of the 95 corrected outputs instead present an explicit final answer. The largest net improvement occurs on mathematics, while every accuracy task exhibits a positive correction balance.