Geometric Model Merging for Efficient and Scalable Adaptation of Large Language Models

Lilian Rage
École Centrale d'Électronique (ECE)
Paris, France
lilian.rage@edu.ece.fr

Youri Lalain
École Centrale d'Électronique (ECE)
Paris, France
youri.lalain@edu.ece.fr

Mathis Escriva
École Centrale d'Électronique (ECE)
Paris, France
mathis.escriva@edu.ece.fr

Martial Roberge
École Centrale d'Électronique (ECE)
Paris, France
martial.roberge@edu.ece.fr

Paul Lemaistre
Racine.ai
Paris, France
plemaistre@omnesintervenant.com

André-Louis Rochet
École Centrale d'Électronique (ECE)
TW3 Partners
Paris, France
arochet@tw3partners.com

Gérard Réus
École Centrale d'Électronique (ECE)
Paris, France
greus@ece.fr

Bikram Pratim Bhuyan*
École Centrale d'Électronique (ECE)
Paris, France
LISV Laboratory
Université Paris-Saclay
Velizy, France
bpbhuyan@ece.fr

***Abstract*—Fine-tuning large language models (LLMs) for specialized tasks is resource-intensive and challenging to scale across multiple domains. Model merging, which directly combines parameters from fine-tuned LLMs without additional training, has emerged as an efficient alternative. However, traditional merging methods, such as linear interpolation, often degrade performance due to destructive interference between conflicting parameters. To address these limitations, we propose Layer-Adaptive Spherical Linear Interpolation (Layer-Adaptive SLERP), a novel merging strategy that i) follows a geometry-preserving SLERP path and ii) assigns layer-specific coefficients that respect the distinct roles of attention, feed-forward and embedding blocks. Across 50+ merges spanning six architectures and seven parameter scales (0.5B - 7B), we demonstrate that our method significantly improves merging stability and task-specific performance. Results indicate that the merged 7B variant attains competitive leaderboard performance and supports production-scale deployments, confirming the method's robustness and applicability to real-world adaptation tasks.**

***Index Terms*—Model Merging, Large Language Models, Geometric interpolation, Foundation models, Scalable Deployment**

I. Introduction

In the era of large language models (LLMs), fine-tuning and instruction-tuning have become dominant strategies for adapting pretrained models to specific downstream tasks [1]. However, these methods are increasingly constrained by the computational cost, memory footprint, and time required to individually fine-tune or distill large models for every use case. As LLMs scale into the multi-billion parameter range, practitioners are seeking more efficient methods of model reuse, adaptation, and composition, especially in low-resource and multi-domain environments.

Among emerging alternatives, model merging (combining two or more trained models into a single performant model) [2] has received growing attention [3]. However, most existing approaches, such as linear interpolation, task arithmetic, or weight averaging, are often unstable, yield inconsistent performance, or result in destructive interference when merging models trained on divergent tasks [4]. More critically, there is a lack of empirically grounded guidelines on when, why, and how such merges succeed or fail, especially when applied to instruction-tuned, multilingual, or domain-specific LLMs across different architectures and sizes.

In this work, we explore a geometrically principled model merging strategy using Spherical Linear Interpolation (SLERP) [5] which is a method that interpolates weight vectors on the hypersphere rather than in Euclidean space, preserving angular relationships between models. We extend SLERP with layer-wise parameterization, allowing different interpolation coefficients for attention layers, feedforward modules, and global weight structures. This technique, which we term *Layer-Adaptive SLERP*, provides fine-grained control over merge behavior and shows promising improvements in stability and performance across domains.

To the best of our knowledge, we conducted the largest and most systematic study of SLERP-based model merging to date, covering more than 50 model merges across six architectural families and seven parameter scales (0.5B to 7B), using

*$Corresponding Author$

benchmarks such as IFEval, BBH, MATH, GPQA, MUSR, and MMLU-Pro. Through this analysis, we:

- Identify the architectural and functional conditions under which SLERP merging succeeds or fails.
- Demonstrate the effectiveness of layer-wise SLERP configurations over traditional uniform merges.
- Propose a set of best practices for practical model merging and deployment.
- Offer the community an open framework and reproducible configurations for further exploration.

Our results have immediate applications in enterprise AI deployment, multilingual assistant development, low-resource domain adaptation, and research into modular foundation models. For the IEEE community, our study offers a blueprint for scalable, reusable AI systems through lightweight model composition.

The remainder of this paper is organized as follows. Section II reviews related approaches to parameter merging and geometric interpolation. Section III introduces the theoretical formulation of Layer-Adaptive SLERP and the methodology. Section IV presents results and analysis across multiple model scales, while Section V discusses implications, limitations and highlights directions for future work. Finally, Section VI concludes the paper.

II. Related Work

Recent advancements in LLMs have primarily relied on parameter-efficient fine-tuning strategies such as adapters [6, 7], prompt tuning [8], and instruction tuning [9]. Despite their effectiveness, these approaches remain resource-intensive for adapting models to multiple specialized tasks [10]. Consequently, model merging, which directly combines parameters from multiple pretrained or fine-tuned models without retraining, has emerged as an efficient alternative [10]. Initial methods involved straightforward linear weight averaging or task arithmetic; however, these approaches often degraded model performance due to destructive interference and failure to account for parameter space geometry [11, 12, 13].

To address these limitations, recent research introduced geometric interpolation techniques such as SLERP, which better preserves angular relationships between parameter vectors, thus significantly reducing interference during model merges [14, 15]. However, existing SLERP implementations apply uniform interpolation across all parameters, ignoring the inherent functional differences between transformer layers, such as self-attention versus feed-forward modules [16]. In response, more granular merging methods, including Fisher-weighted merging [12], AdaMerging [17], TIES [18], and Drop-And-Rescale [19], were proposed. These methods adapt merging coefficients at a layer-wise or even parameter-wise level, though typically introducing significant complexity or computational overhead.

Despite these developments, comprehensive empirical exploration of adaptive, layer-wise geometric merging techniques (especially SLERP-based methods) is notably absent, particularly regarding systematic evaluation across diverse architectures and parameter scales [20]. Additionally, practical tools like MergeKit [21] and HuggingFace PEFT [22] have emerged to facilitate merging implementation, yet they lack built-in methods for efficient, adaptive layer-specific geometric merging [16]. Our work fills these critical gaps by introducing Layer-Adaptive SLERP, combining layer-specific parameter interpolation with geometric SLERP to optimize merging efficiency and robustness through extensive experiments and real-world deployment.

III. Methodology

Let us consider M_1 and M_2 be two pretrained language models with identical architectures and parameter sets θ_1 and θ_2, respectively, where $\theta_i \in \mathbb{R}^d$, $\forall i \in \{1, 2\}$ and d be the total number of model parameters. Our goal is to construct a merged model M_{merge} with parameters θ_{merge} that combines the joint capabilities of M_1 and M_2 without retraining. Formally,

$$\theta_{merge} = merge(\theta_1, \theta_2) \tag{1}$$

where the 'merge' function should preserve stability, generalizability, and task-specific performance.

A naïve linear interpolation-based method for the merged parameters can be defined as,

$$\theta_{merge}^{linear} = (1 - c)\theta_1 + c\theta_2, \qquad c \in [0, 1] \tag{2}$$

where 'c' is the interpolation coefficient. However, this approach ignores the curved geometry observed in high-dimensional weight spaces. Hence, we employ spherical linear interpolation (SLERP), which maintains a constant angular velocity between θ_1 and θ_2 in the hypersphere.

Formally, given that θ_1, θ_2 are normalized in the unit form (i.e. $||\theta_1|| = ||\theta_2|| = 1$), the SLERP-based interpolation can be defined as,

$$\theta_{merge}^{SLERP} = \frac{sin((1 - c)\omega)}{sin(\omega)}\theta_1 + \frac{sin(c\omega)}{sin(\omega)}\theta_2 \tag{3}$$

where, $\omega = arccos(< \theta_1, \theta_2 >)$ is the angle between θ_1, θ_2 and $< \theta_1, \theta_2 >$ represents their dot product. To avoid degeneracy for nearly collinear/antipodal parents we clip the angle: $\tilde{\omega} = \min\{\max\{\omega, \varepsilon\}, \pi - \varepsilon\}$ with $\varepsilon \approx 10^{-6}$ and compute the sine ratio with `hypot`-based normalization.

Traditional model merging methods apply Euclidean interpolation of model parameters, implicitly assuming that the weight space is flat. However, transformer weights evolve on a curved manifold due to normalization constraints and non-linear activations. SLERP instead operates on the hypersphere formed by the normalized weights, ensuring that the merged parameters remain on a consistent manifold and preserve angular relationships between parent models.

A. Layer-Adaptive SLERP

Global SLERP applies a uniform interpolation c across all parameters, while transformer layers are functionally heterogeneous; for example, attention layers capture contextual relationships, while MLP layers focus on feature transformations and embeddings anchor token geometry. Thus, a uniform

interpolation may lead to overwriting or conflicting behaviors between merged models. To address this limitation, we introduce *Layer-Adaptive SLERP*, where each layer group (e.g., attention, MLP) is assigned its own interpolation coefficient. Formally, let $\mathcal{L} \in \mathbb{N}$ be the set of all layers. For each layer $l \in \mathcal{L}$ the parameter subsets $\theta_1^{(l)}$, $\theta_2^{(l)}$ is formulated. Let $c_l \in [0, 1]$ denote the layer-specific interpolation coefficient for layer l. Then, the merged parameters at the layer l is defined by Layer-Adaptive SLERP as,

$$\theta_{merge}^{(l)} = \frac{sin((1-c_l)\omega_l)}{sin(\omega_l)}\theta_1^{(l)} + \frac{sin(c_l\omega_l)}{sin(\omega_l)}\theta_2^{(l)} \tag{4}$$

where, $\omega_l = arccos(\frac{<\theta_1^{(l)}, \theta_2^{(l)}>}{||\theta_1^{(l)}|| \cdot ||\theta_2^{(l)}||})$ is the angular distance between corresponding parameters in layer l. Finally, the complete merged model is represented as,

$$\theta_{merge} = \bigcup_{l=1}^{\mathcal{L}} \theta_{merge}^{(l)} \tag{5}$$

The coefficients c_l are chosen non-uniformly based on a) *Layer Type* (e.g., higher c_l are chosen for self-attention layers (to favour new context modelling), and lower c_l are chosen for MLP layers (to preserve structural embeddings)). b) *Depth Position* (e.g., deeper layers blend more aggressively). c) *Functional Specialization*, i.e. if one source model is more specialized (e.g., math, language, etc.), adaptive schedules can bias layers toward the more competent model.

For theoretical completeness, we note some properties (i) *Geodesic property.* For each l, Eq. 4 lies on the great-circle between parents, preserving per-layer scale and controlling angular displacement smoothly. (ii) *First-order view.* Linearizing $f_{\boldsymbol{\theta}}$ around $\boldsymbol{\theta}_1$ gives

$$f_{\boldsymbol{\theta}}(x) \approx f_{\boldsymbol{\theta}_1}(x) + \sum_l c_l \, (\boldsymbol{\theta}_2^{(l)} - \boldsymbol{\theta}_1^{(l)})^\top \nabla_{\boldsymbol{\theta}^{(l)}} f_{\boldsymbol{\theta}_1}(x).$$

Larger c_l where parents *agree* (e.g., small ω_l or high representation similarity) increases useful transfer and reduces interference; deep MLPs often show lower agreement than early attention.

Thus, we can define an objective function $\mathcal{J}$ where the optimal interpolation coefficients could be found by solving,

$$\{c_l^*\} = \arg \max_{\{c_l\} \in [0,1]^{\mathcal{L}}} \mathcal{J}(\theta_{merge}(\{c_l\})) \tag{6}$$

with $\mathcal{J}$ an application-aligned validation score. In practice (to avoid heavy tuning), we recommend a tiny validation sweep or a budgeted BO/CMA-ES over the six scalars (24–32 trials on a 2–3k-token set). In this work, we heuristically design $\{c_l\}$ based on model architecture and domain-specific knowledge, and validate via benchmarks.

Layer-Adaptive SLERP operates per parameter block and is linear in d. It requires two passes per tensor (to compute ω_l and to merge) and one auxiliary buffer (streamable to cap peak memory). Quantized checkpoints can be handled via dequant–merge–requant in FP16. The procedure leaves optimizer states untouched unless the serving stack requires them.

Merges succeed when parents are *compatible*. We can expose a lightweight score as,

$$S \;=\; \alpha\, O \;+\; \beta\, \overline{A} \;-\; \gamma\, \bar{\omega}, \qquad \alpha, \beta, \gamma > 0, \tag{7}$$

where O is tokenizer overlap (shared tokens/strings), $\overline{A}$ aggregates layer similarity (e.g., cosine/CKA proxies), and $\bar{\omega}$ is the mean layer angle. We can set up some guardrails: (i) if $\bar{\omega} > \tau_\omega$ (large divergence), fall back to global SLERP or add an embedding-alignment step; (ii) if $O < \tau_O$, align embeddings via an orthogonal Procrustes map on shared tokens before Eq. (4); (iii) for 0.5B models, prefer global SLERP due to capacity fragility.

Algorithm 1 concisely represents the process.

Algorithm 1 Layer-Adaptive SLERP (per layer l)

Require: Parents $\{\boldsymbol{\theta}_1^{(l)}\}_{l=1}^{L}$, $\{\boldsymbol{\theta}_2^{(l)}\}_{l=1}^{L}$; schedule $\{c_l\}$; guardrails $\varepsilon, \tau_O, \tau_\omega$.

1: **for** $l = 1$ **to** L **do**

2: $\omega_l \leftarrow \arccos\big(\frac{\langle \theta_1^{(l)}, \theta_2^{(l)} \rangle}{\|\theta_1^{(l)}\| \, \|\theta_2^{(l)}\|}\big)$; $\omega_l \leftarrow \min\{\max\{\omega_l, \varepsilon\}, \pi - \varepsilon\}$.

3: $a \leftarrow \sin((1-c_l)\omega_l)/\sin\omega_l$; $b \leftarrow \sin(c_l\omega_l)/\sin\omega_l$.

4: $\boldsymbol{\theta}_{\text{merge}}^{(l)} \leftarrow a\,\boldsymbol{\theta}_1^{(l)} + b\,\boldsymbol{\theta}_2^{(l)}$.

5: **end for**

6: **return** $\{\boldsymbol{\theta}_{\text{merge}}^{(l)}\}_{l=1}^{L}$ and $\boldsymbol{\theta}_{\text{merge}} = \bigcup_l \boldsymbol{\theta}_{\text{merge}}^{(l)}$.

Both standard averaging and SLERP have linear time complexity in the number of parameters θ, but differ in memory traffic. Standard averaging performs a single pass, requiring $\Theta(\theta)$ operations and roughly $3d$ memory transfers (two reads and one write). SLERP, whether global or layer-adaptive, requires an additional reduction pass to compute per-layer norms and dot products, resulting in about $5d$ memory transfers and an overall complexity of $\Theta(\theta)$. Consequently, the practical wall-clock time of SLERP is approximately 1.6-1.7$\times$ that of simple averaging under memory-bound conditions. Layer-Adaptive SLERP introduces no extra asymptotic cost beyond global SLERP, as the per-layer coefficients $\{c_l\}$ are scalar operations.

[1]https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct
[2]https://huggingface.co/Youlln/ECE.EIFFEIL.ia-0.5B-SLERP
[3]https://huggingface.co/Youlln/ECE-Qwen0.5B-FT-V2
[4]https://huggingface.co/MEscriva/ECE-PRYMMAL-0.5B-FT-V4-MUSR-Mathis
[5]https://huggingface.co/lalainy/ECE-PRYMMAL-YL-0.5B-SLERP-BIS-V1
[6]https://huggingface.co/fblgit/miniclaus-qw1.5B-UNAMGS
[7]https://huggingface.co/Goekdeniz-Guelmez/Josiefied-Qwen2.5-1.5B-Instruct-abliterated-v1
[8]https://huggingface.co/Goekdeniz-Guelmez/Josiefied-Qwen2.5-1.5B-Instruct-abliterated-v2
[9]https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
[10]https://huggingface.co/bond005/meno-tiny-0.1
[11]https://huggingface.co/Qwen/Qwen2.5-1.5B

Name	Method	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
SLERP Merge (t = 0.7)[2]	SLERP	**8.83**	25.61	8.41	**5.97**	2.01	0.94	**10.04**
Fine-tuned on math data[3]	FT	7.57	25.26	7.63	2.04	**2.24**	0.89	7.40
Fine-tuned on QASC[4]	FT	7.26	18.82	8.08	2.72	1.79	**4.13**	8.00
SLERP Merge (layer-adaptive)[5]	SLERP	3.61	14.37	2.93	0.08	0.00	2.94	1.35
Pretrained (no merge)[1]	Baseline	8.14	**30.71**	**8.43**	0.00	1.01	0.94	7.75

TABLE I

COMPARISON OF DIFFERENT MERGING AND FINE-TUNING METHODS ACROSS BENCHMARKS AT 0.5B SCALE.

Symbol	Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
🟢	`fblgit`[6]	17.05	33.48	18.56	10.88	**5.59**	**12.23**	**21.52**
🔵	`Goekdeniz-Guelmez(v1)`[7]	18.44	**47.69**	18.31	20.85	0.00	4.00	19.81
🔴	`Goekdeniz-Guelmez(v2)`[8]	15.57	42.16	16.50	12.69	0.00	4.71	17.35
🟠	`Qwen2.5-1.5B-Instruct`[9]	18.43	44.76	**19.81**	**22.05**	0.78	3.19	19.99
🟤	`meno-tiny-0.1`[10]	**18.85**	45.50	19.64	13.90	4.25	9.97	19.84
⚫	`Qwen2.5-1.5B`[11]	13.85	26.74	16.66	9.14	4.70	5.27	20.61

TABLE II

PERFORMANCE OF 1B-SCALE BASE MODELS ACROSS BENCHMARKS. SYMBOLS ARE FILLED CIRCLES INDICATING THE BASE MODEL.

Combo	Model	SLERP t	Avg	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
🟢 + 🔵	`ECE-PRYMMAL-1B-V1`[12]	t = 0.65	**16.68**	32.51	18.28	10.73	5.48	11.59	**21.51**
🟢 + 🔴	`ECE-PRYMMAL-1B-V2`[13](🟡)	t = 0.65	**16.68**	32.51	18.28	10.73	5.48	11.59	**21.51**
🟡 + 🟠	`ECE-PRYMMAL-1B-V3`[14]	t = 0.65	16.45	32.50	18.23	9.74	5.93	10.83	21.46
🟡 + 🔵	`ECE-PRYMMAL-1B-V4`[15]	t = 0.65	16.44	**33.24**	17.41	10.05	4.81	**12.09**	21.03
🟡 + ⚫	`ECE-PRYMMAL-1B-V5`[16]	t = 0.65	15.84	33.13	18.88	11.10	4.81	5.65	21.45
🟤 + 🟠	`ECE_Poirot`[17]	t = 0.5	15.74	31.07	18.62	9.14	**6.38**	8.33	20.92
⚫ + 🟠	`MiniQwenMathExpert`[18]	t = 0.5	15.08	27.95	**19.02**	**11.40**	4.25	6.51	21.36

TABLE III

PERFORMANCE OF SLERP-MERGED 1B-SCALE MODELS WITH VARIOUS PAIRWISE COMBINATIONS ACROSS BENCHMARKS. EACH COMBINATION IS DENOTED BY TWO FILLED COLORED CIRCLES CORRESPONDING TO ITS BASE MODELS.

IV. RESULTS AND EVALUATION

A. Results at the 0.5B Parameter Scale

We begin with the smallest setting on Qwen2 models[1] to test whether geometric merging is beneficial with 0.5 billion parameters. Table I compares i) two single-task fine-tunes ("MATH" and "QASC"), ii) a *global* SLERP merge using a single coefficient $t = 0.7$, iii) our first *layer-adaptive* SLERP attempt, and iv) the pretrained baseline.

We observed that, the layer-adaptive variant fails dramatically (*Average* 3.61%). The schedule {self_attn :[0, 0.25, 0.5, 0.75, 1]; mlp :[1, 0.75, 0.5, 0.25, 0]} *down-weights* early self-attention blocks (which encode global context) and *over-weights* late MLP layers that were fine-tuned on heterogeneous tasks. This negative outcome is instructive. At 0.5B parameters the model's limited capacity makes it highly sensitive to coefficient imbalance; a uniform SLERP is therefore a safer default.

B. Results at the 1B Parameter Scale

To understand the foundations upon which SLERP merging operates, we first evaluate each individual base model at the 1B parameter scale as shown in Table II. The six base models in Table II show highly complementary skills. `Goekdeniz{v1`(🔵) dominates IFEVAL and MATH; `fblgit`(🟢) excels at MUSR/GPQA; `meno-tiny-0.1`(🟤) is the most balanced. Such diversity highlights the limitation of a single interpolation coefficient and motivates the layer-wise scheme formulated in Eq. (4).

Four heuristic merges (PRYMMAL-V2...V5) employ the heuristic search for the optimal coefficient $\big($self_attn $0\to1$, mlp $1\to0$, t=0.65$\big)$. As Table III shows, each of them equals

[12]https://huggingface.co/Youlln/ECE-PRYMMAL-YL-1B-SLERP-V1
[13]https://huggingface.co/Youlln/ECE-PRYMMAL-YL-1B-SLERP-V2
[14]https://huggingface.co/lalainy/ECE-PRYMMAL-YL-1B-SLERP-V3
[15]https://huggingface.co/lalainy/ECE-PRYMMAL-YL-1B-SLERP-V4
[16]https://huggingface.co/llnYou/ECE-PRYMMAL-YL-1B-SLERP-V5
[17]https://huggingface.co/SpaceYL/ECE_Poirot
[18]https://huggingface.co/Marsouuu/MiniQwenMathExpert-ECE-PRYMMAL-Martial
[19]https://huggingface.co/zake7749/gemma-2-2b-it-chinese-kyara-dpo
[20]https://huggingface.co/google/gemma-2-2b-jpn-it
[21]https://huggingface.co/google/gemma-2-2b-it
[22]https://huggingface.co/cognitivecomputations/dolphin-2.9.4-gemma2-2b
[23]https://huggingface.co/google/gemma-2-2b
[24]https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[25]https://huggingface.co/ValiantLabs/Llama3.2-3B-ShiningValiant2
[26]https://huggingface.co/qingy2019/LLaMa_3.2_3B_Catalysts
[27]https://huggingface.co/Lil-R/2_PRYMMAL-ECE-2B-SLERP-V1
[28]https://huggingface.co/Lil-R/2_PRYMMAL-ECE-2B-SLERP-V2
[29]https://huggingface.co/Marsouuu/MiniMathExpert-2_61B-ECE-PRYMMAL-Martial
[30]https://huggingface.co/llnYou/ECE-PRYMMAL-YL-3B-SLERP-V2
[31]https://huggingface.co/llnYou/ECE-PRYMMAL-YL-3B-SLERP-V1

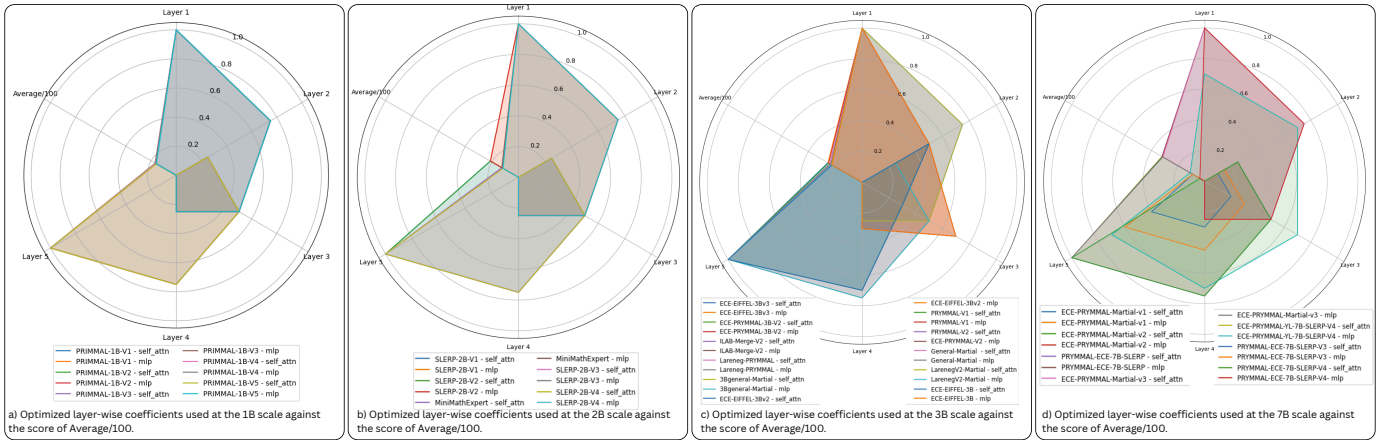

Fig. 1. Optimized layer-wise coefficients used at the 1,2,3 and 6B scale against the score of Average(%)/100.

Symbol	Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
▲	gemma-it-chinese-dpo[19]	19.62	53.82	19.06	8.38	2.24	**16.76**	17.48
▲	gemma-jpn-it[20]	16.68	52.88	17.85	4.76	3.36	4.93	16.30
▲	gemma-it[21]	17.05	56.68	17.98	0.08	3.24	7.08	17.22
▲	dolphin-2.9.4[22]	9.84	8.96	17.37	4.91	4.59	10.91	12.28
▲	gemma-vanilla[23]	10.13	19.93	11.76	2.87	1.68	11.43	13.11
▲	llama-3.2-Instruct[24]	**24.20**	**73.93**	**24.06**	**17.67**	3.80	1.37	**24.39**
▲	Valiant-Shining[25]	14.39	26.25	18.91	8.23	4.03	8.60	20.32
▲	Catalyst[26]	19.93	49.92	21.35	12.92	**5.15**	7.95	22.31

TABLE IV
PERFORMANCE OF 2B-SCALE BASE MODELS ACROSS BENCHMARKS. SYMBOLS ARE FILLED TRIANGLES INDICATING MODEL SOURCE.

Combo	Model	SLERP t	Avg	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
▲ + ▲	ECE-2B-SLERP-V1[27]	0.5	**21.16**	**58.23**	19.53	9.14	**7.49**	13.92	18.64
▲ + ▲	ECE-2B-SLERP-V2[28]	0.5	21.07	55.43	**20.20**	**9.44**	6.38	**15.62**	19.38
▲ + ▲	MiniMathExpert-2.6B[29]	0.5	12.49	25.48	15.30	7.40	3.36	9.27	14.15
▲ + ▲	ECE-2B-SLERP-V3[30]	0.65	11.81	23.09	15.20	1.28	3.58	6.61	**21.11**
▲ + ▲	ECE-2B-SLERP-V4[31]	0.65	11.63	23.46	15.80	0.91	5.82	3.22	20.55

TABLE V
PERFORMANCE OF SLERP-MERGED 2B-SCALE MODELS ACROSS BENCHMARKS. EACH COMBINATION IS DENOTED BY TWO FILLED COLORED TRIANGLES CORRESPONDING TO ITS BASE MODELS.

the stronger parent on three of seven benchmarks, yielding 15.8-16.7% in average accuracy. In sharp contrast, the global-SLERP baselines (Poirot, MiniMathExpert) (without layer adaptive) boosts target metric (GPQA or MATH) but lose up to 5.2 % on IFEVAL, confirming that uniform interpolation (Eq. 3) cannot respect functional heterogeneity.

PRYMMAL-V1 flips the schedule (self_attn $1\to0$, mlp $0\to1$). Although its parents are compatible, performance stalls: IFEVAL drops to 32.5% versus 33.2% for V4, and MUSR lags by 0.5%.

Figure 1a visualises the difference as V1 starves early attention layers (the locus of global context) explaining the shortfall. This ablation validates the design rule (in Section §III) that early layers must prioritise attention from the instruction-rich model, while deeper layers absorb the semantic embeddings of the reasoning-rich model.

C. Results at the 2B Parameter Scale

Table IV shows a heterogeneous landscape; where, gemma-cn-dpo(▲) excels at MUSR, Llama-3.2-Instr(▲) dominates IFEVAL, BBH and MATH; while gemma-it(▲) is strong on instructions but near-zero on MATH.

In the five pairwise merges of Table V, we found that the *symmetric* schedule self_attn $0\to1$, mlp $1\to0$ is optimal, and so we adjust only the global offset $t \in \{0.5, 0.65\}$. The two Gemma-only merges (SLERP-V1 and V2) performs better than thier parent on at least five of seven tasks and lift the average to $21.1 - 21.2\%$ (vs. 19.6% for the best Gemma base), confirming that layer-wise coefficients can fuse complementary instruction-following and multilingual reasoning. MINIMATHEXPERT applies the same schedule to two weak parents and improves only marginally (Avg. 12.5 V3 and V4 inherit isolated strengths but suffer large drops on MATH/MUSR. Thus, the coefficient template of Eq.(4)

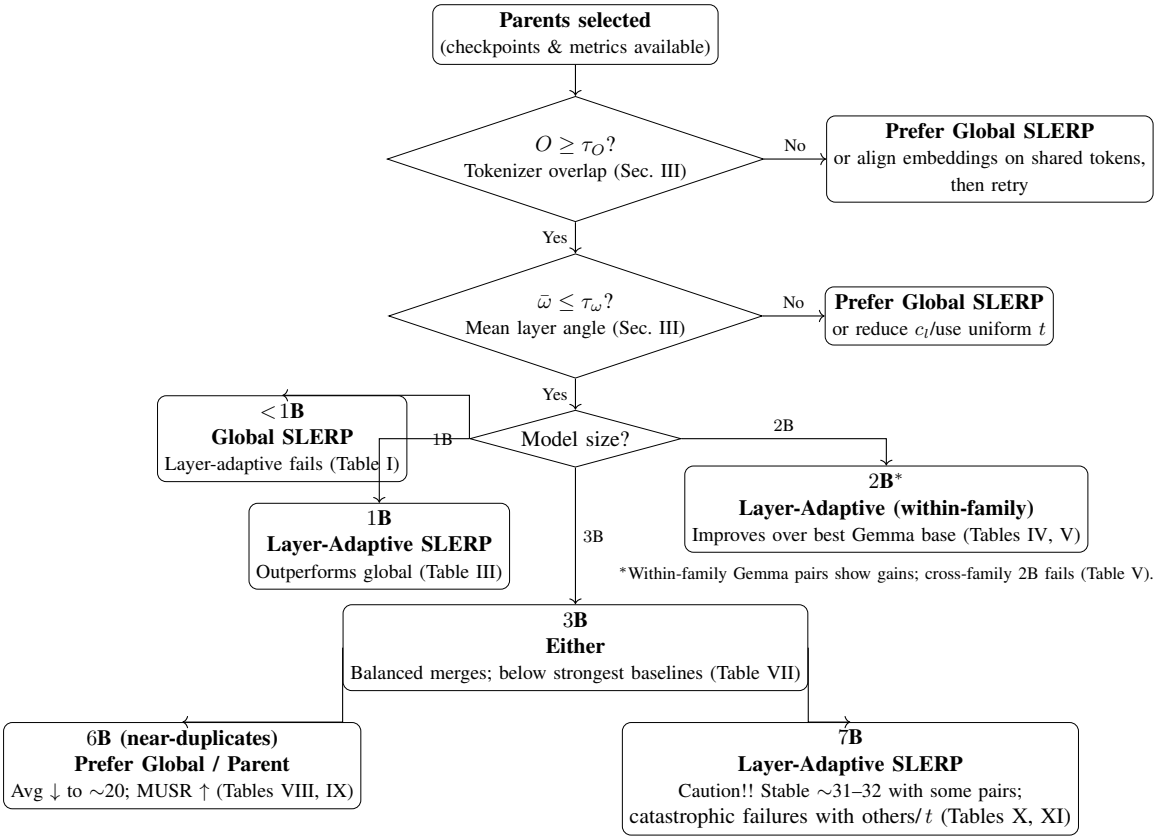

Fig. 2. Decision flow for choosing *Global* vs. *Layer-Adaptive* SLERP using tokenizer overlap O, mean angular distance $\bar{\omega}$, and model size. Thresholds τ_O, τ_ω follow Sec. III. Empirical anchors from the manuscript: $<$1B: Global safer (Table I); 1B: Layer-adaptive $>$ global (Table III); 2B: Layer-adaptive improves within-family Gemma (Tables IV, V); 3B: balanced but below strongest baselines (Table VII); 6B: near-duplicate parents trade Avg ↓ for MUSR ↑ (Tables VIII, IX); 7B: some merges stable ∼31–32, others fail with certain parent choices/t (Tables X, XI). *Note: O* and $\bar{\omega}$ are decision criteria from Sec. III; they are not reported as measured quantities in the results.

is *necessary* but not sufficient: semantic and architectural alignment between sources remains a prerequisite.

Figure 1b confirms that the best models (V1, V2) and the weaker MINIMATHEXPERT use *identical* layer vectors; their divergent outcomes stem from parent incompatibility, not coefficient choice. Empirically, $t \in [0.5, 0.65]$ is a robust global range at this scale.

D. Results at the 3B Parameter Scale

The 3B baselines in Table VI form three clusters - i. `Chocolatine` models are broadly capable but weaker on IFEVAL; ii. `Phi`/`MedIT` models excel at IFEVAL and MATH yet lag on MUSR; iii. Our own `Merge-Test` mixes skills but ignores geometry and show signs of over-fitting.

The 13 merges in Table VII instantiate two schedules- i. **S-template** self_attn $[0, 0.50, 0.30, 0.70, 1]$, mlp $[1, 0.50, 0.70, 0.30, 0]$, $t = 0.50$; **Q-template** self_attn $[0, 0.25, 0.50, 0.75, 1]$, mlp $[1, 0.75, 0.50, 0.25, 0]$, $t \in [0.50, 0.65]$. Both satisfy the design rule of Eq.(4) (early layers favour attention, deep layers favour MLP) but differ in how aggressively they ramp.

Homogeneous `Chocolatine-Merge-Test` merges using **S** (e.g. ECE-EIFFEL-3Bv3) lift GPQA to 10.6% and MUSR to 18.3% while retaining the parents' strong BBH (36.5%), achieving the best merged average (25.5%). Likewise, ILAB-MERGE-V2 employs the milder **Q** schedule on two `Phi`-family models and delivers the top IFEVAL (40.3%) among merges, showing that a gentler ramp suffices when pre-training objectives are already aligned.

Cross-family pairs (`Chocolatine` + `Phi`) using either schedule improve isolated tasks but fall short in overall average ($< 25\%$). Fig.1c plots the attention/MLP weights and reveals a larger angular gap between the parent vectors than at 1B–2B; the fixed schedules cannot fully reconcile this mismatch. Thus layer-adaptive SLERP is *necessary* for skill fusion but *not sufficient* when tokenisation and activation statistics diverge.

Symbol	Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
■	Chocolatine-v1.0[32]	25.43	37.37	36.55	17.82	8.72	**19.47**	32.63
■	Merge-Test[33]	26.08	53.83	33.35	12.08	9.62	15.64	31.93
■	Chocolatine-Revised[34]	28.23	56.23	37.16	18.05	9.62	15.10	33.21
■	Phi-3-mini-4k[35]	25.97	56.13	**39.27**	11.63	9.28	7.64	31.85
■	MedIT-Mesh[36]	**28.32**	58.14	37.55	**20.32**	9.84	10.60	**33.46**
■	Phi-3.5-mini[37]	28.18	57.75	36.75	19.64	**11.97**	10.10	32.91
■	Phi-3-mini-128k[38]	26.34	**59.76**	37.10	14.05	9.06	7.71	30.38

TABLE VI

BASELINE PERFORMANCE OF 3B-SCALE MODEL VARIANTS ACROSS BENCHMARKS. SYMBOLS ARE FILLED SQUARES INDICATING MODEL IDENTITY AT THE 3B SCALE.

Combo	Model	SLERP t	Avg	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
■ + ■	ECE-EIFFEL-3Bv3[39]	0.5	**25.50**	37.86	36.46	16.69	10.63	18.31	33.06
■ + ■	PRYMMAL-3B-V2[40]	0.5	24.99	36.64	35.71	**16.77**	9.51	18.07	33.22
■ + ■	PRYMMAL-3B-V1[41]	0.5	24.96	36.49	35.71	**16.77**	9.51	18.07	33.22
■ + ■	ILAB-Merge-V2[42]	0.5	24.07	**40.29**	36.00	15.18	7.38	13.75	31.78
■ + ■	Lareneg-PRYMMAL[43] (■)	0.5	23.94	33.03	36.35	15.18	9.96	**18.39**	30.74
■ + ■	YL-3B-SLERP-V3[44]	0.65	23.43	35.81	36.63	12.99	7.27	14.05	**33.82**
■ + ■	3Bgeneral-Martial[45]	0.5	23.28	32.89	**36.67**	13.14	9.96	14.43	32.60
■ + ■	ECE-EIFFEL-3Bv2[46]	0.5	23.14	30.13	36.35	11.86	11.41	15.77	33.33
■ + ■	PRYMMAL-V1[47]	0.5	23.14	29.33	35.05	16.62	8.95	16.64	32.23
■ + ■	PRYMMAL-V2[48]	0.5	23.14	29.33	35.05	16.62	8.95	16.64	32.23
■ + ■	General-Martial[49]	0.5	22.98	27.22	35.70	15.48	9.28	18.22	31.96
■ + ■	LarenegV2-Martial[50]	0.5	22.76	28.76	35.45	12.08	**11.30**	15.43	33.51
■ + ■	ECE-EIFFEL-3B[51]	0.5	22.50	34.69	31.29	12.16	10.85	14.70	31.34

TABLE VII

PERFORMANCE OF SLERP-MERGED MODELS (3B SCALE) ACROSS BENCHMARKS. EACH MODEL IS DENOTED BY A COMBINATION OF BASELINE MODELS (SEE TABLE VI) USING SQUARE SYMBOLS.

E. Results at the 6B Parameter Scale

At this scale we have only a single public (Table VIII), already strong on instructions (Avg. 22.8%) but comparatively weak on multi-step reasoning (MUSR 14.0%). A second candidate (ECE-ILAB-Yi-6B-SLERP) has no public scores but is architecturally identical (the angular distance ω_l in Eq.(3) is small layer-wise).

[32]https://huggingface.co/jpacifico/Chocolatine-3B-Instruct-DPO-v1.0
[33]https://huggingface.co/lesubra/merge-test
[34]https://huggingface.co/jpacifico/Chocolatine-3B-Instruct-DPO-Revised
[35]https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
[36]https://huggingface.co/meditsolutions/MedIT-Mesh-3B-Instruct
[37]https://huggingface.co/microsoft/Phi-3.5-mini-instruct
[38]https://huggingface.co/microsoft/Phi-3-mini-128k-instruct
[39]https://huggingface.co/lesubra/ECE-EIFFEL-3Bv3
[40]https://huggingface.co/lesubra/ECE-PRYMMAL-3B-SLERP_2-V2
[41]https://huggingface.co/lesubra/ECE-PRYMMAL-3B-SLERP_2-V1
[42]https://huggingface.co/ECE-ILAB-PRYMMAL/ILAB-Merging-3B-V2
[43]https://huggingface.co/Marsouuu/lareneg3B-ECE-PRYMMAL-Martial
[44]https://huggingface.co/llnYou/ECE-PRYMMAL-YL-3B-SLERP-V3
[45]https://huggingface.co/brgx53/3Bgeneral-ECE-PRYMMAL-Martial
[46]https://huggingface.co/lesubra/ECE-EIFFEL-3Bv2
[47]https://huggingface.co/lesubra/ECE-PRYMMAL-3B-SLERP-V1
[48]https://huggingface.co/lesubra/ECE-PRYMMAL-3B-SLERP-V2
[49]https://huggingface.co/Marsouuu/general3B-ECE-PRYMMAL-Martial
[50]https://huggingface.co/brgx53/3Blareneg-ECE-PRYMMAL-Martial
[51]https://huggingface.co/lesubra/ECE-EIFFEL-3B
[52]https://huggingface.co/01-ai/Yi-1.5-6B-Chat
[53]https://huggingface.co/Youlln/ECE-ILAB-Yi1.5-6B-SLERP
[54]https://huggingface.co/lalainy/ECE-PRYMMAL-YL-6B-SLERP-V1
[55]https://huggingface.co/lalainy/ECE-PRYMMAL-YL-6B-SLERP-V2

Using the symmetric schedule self_attn $0\to1$, mlp $1\to0$, $t = 0.65$ we create two merged models, PRYMMAL-6B-V1/-V2. Compared with the baseline, the merge trades IFEVAL (suggests that early attention weights from the unpublished model overwrite well-aligned instruction layers in Yi-Chat) (–19%) for a sizeable jump in MUSR (+6.6%) and a modest lift in BBH (+0.8% across both merges (Table IX). Because the parents are near-duplicates, layer-wise coefficients simply re-weight similar features rather than fusing complementary ones, so the overall average drops to 20.0%. Future work should pair Yi-6B with a maths or reasoning specialist model to verify that Eq.(4) still scales once legitimate heterogeneity is reintroduced.

F. Results at the 7B Parameter Scale

Table X shows the performance of the base models. The seven merges in Table XI is seen to be optimised for the

[56]https://huggingface.co/fblgit/cybertron-v4-qw7B-MGS
[57]https://huggingface.co/Tsunami-th/Tsunami-0.5x-7B-Instruct
[58]https://huggingface.co/rombodawg/Rombos-LLM-V2.5-Qwen-7b
[59]https://huggingface.co/Goekdeniz-Guelmez/Josiefied-Qwen2.5-7B-Instruct-abliterated-v2
[60]https://huggingface.co/Qwen/Qwen2.5-Math-7B
[61]https://huggingface.co/Marsouuu/general7Bv2-ECE-PRYMMAL-Martial
[62]https://huggingface.co/Marsouuu/general7Bv2-ECE-PRYMMAL-Martial
[63]https://huggingface.co/brgx53/3Blarenegv2-ECE-PRYMMAL-Martial
[64]https://huggingface.co/brgx53/3Blarenegv3-ECE-PRYMMAL-Martial
[65]https://huggingface.co/Youlln/ECE-PRYMMAL-YL-7B-SLERP-V4
[66]https://huggingface.co/Lil-R/2_PRYMMAL-ECE-7B-SLERP-V3
[67]https://huggingface.co/LilRg/PRYMMAL-ECE-7B-SLERP-V4

Symbol	Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
◆	Yi-1.5-6B-Chat[52]	22.78	51.45	23.68	16.24	6.94	14.03	24.37
◆	ECE-ILAB-Yi1.5-6B-SLERP[53]	–	–	–	–	–	–	–

TABLE VIII

BASELINE PERFORMANCE OF 6B-SCALE MODEL VARIANTS ACROSS BENCHMARKS. MODELS ARE DENOTED BY FILLED DIAMONDS TO DIFFERENTIATE FROM OTHER PARAMETER SCALES.

Combo	Model	SLERP t	Avg	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
◆ + ◆	PRYMMAL-6B-V1[54]	t = 0.65	**20.04**	**32.64**	**24.52**	**12.69**	5.15	**20.63**	**24.60**
◆ + ◆	PRYMMAL-6B-V2[55]	t = 0.65	20.01	32.49	**24.52**	**12.69**	5.15	**20.63**	**24.60**

TABLE IX

PERFORMANCE OF SLERP-MERGED MODELS AT THE 6B PARAMETER SCALE ACROSS BENCHMARKS. EACH MODEL IS DENOTED BY A COMBINATION OF BASELINE MODELS (SEE TABLE VIII) USING DIAMOND SYMBOLS

Symbol	Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
⬟	Cybertron-v4-qw7B-MGS[56]	32.40	62.64	37.04	34.89	8.05	13.20	**38.59**
⬟	Tsunami-0.5x-7B-Instruct[57]	**36.00**	70.99	**37.36**	42.07	8.61	**18.57**	38.42
⬟	Rombos-LLM-V2.5-Qwen-7b[58]	32.75	62.37	36.37	38.14	**9.06**	12.00	38.54
⬟	Josiefied-Qwen2.5-7B-v2[59]	35.32	**78.14**	33.33	**45.32**	6.49	13.96	34.66
⬟	Qwen2.5-Math-7B[60]	17.84	24.60	22.01	30.51	5.82	5.00	19.09

TABLE X

PERFORMANCE OF 7B PARAMETER SCALE MODELS PRIOR TO SLERP MERGING. SYMBOLS ARE FILLED PENTAGONS REPRESENTING MODEL SOURCES.

Combo	Model	SLERP t	Avg	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro
⬟ + ⬟	ECE-PRYMMAL-Martial-v1[61]	t = 0.5	**32.11**	**57.53**	37.47	36.56	**9.28**	12.82	**39.01**
⬟ + ⬟	ECE-PRYMMAL-Martial-v2[62]	t = 0.5	31.92	56.93	**37.67**	36.71	8.05	13.28	38.87
⬟ + ⬟	PRYMMAL-ECE-7B-SLERP[63]	t = 0.5	31.52	55.77	36.48	36.33	8.05	**13.48**	38.97
⬟ + ⬟	ECE-PRYMMAL-Martial-v3[64]	t = 0.5	31.46	56.62	37.25	34.97	8.17	12.79	38.95
⬟ + ⬟	ECE-PRYMMAL-YL-7B-SLERP-V4[65]	t = 0.7	10.87	25.10	13.16	5.36	2.01	7.01	12.58
⬟ + ⬟	PRYMMAL-ECE-7B-SLERP-V3[66]	t = 0.2	8.88	22.35	10.61	0.60	0.89	9.74	9.08
⬟ + ⬟	PRYMMAL-ECE-7B-SLERP-V4[67]	t = 0.5	3.54	12.49	2.29	0.98	0.89	3.19	1.41

TABLE XI

PERFORMANCE OF SLERP-MERGED MODELS AT THE 7B PARAMETER SCALE ACROSS BENCHMARKS. SYMBOL PAIRS DENOTE THE SOURCE MODEL COMBINATIONS.

following schedules - i. **H)** self_attn $[0, 0.25, 0.5, 0.75, 1]$, mlp $[1, 0.75, 0.5, 0.25, 0]$, $t = 0.50$ (e.g. GENERAL3BV2); ii. **R)** self_attn $[1, 0.75, 0.5, 0.25, 0]$, mlp $[0, 0.25, 0.5, 0.75, 1]$, $t = 0.50$ (e.g. LARENEG3BV2); iii. **L)** self_attn $[0, 0.10, 0.20, 0.30, 0.40]$, mlp $[0, 0.15, 0.30, 0.45, 0.60]$, $t = 0.20$ (PRYMMAL-V3); iv. **H+)** global $t = 0.70$ without layer vectors (YL-V4).

Pairs of instruction-heavy models (Tsunami + Cybertron or Rombos) using either H or R schedule (MARTIAL-V1/V2/V3 and PRYMMAL) land at Avg. 31.5–32.1%, trading IFEVAL for MUSR and MMLU-PRO versus the Tsunami parent. The near-identical scores of H and R indicate that, when both parents share strong instruction tuning, orientation is less critical; Eq.(4) merely re-weights redundant features.

Blending Tsunami with the maths-only Qwen-Math shows the boundary of our method: the high-mass global run (YL-V4, $t = 0.70$) and the low-mass layered run (PRYMMAL-V3, $t = 0.20$) both collapse to single-digit averages. Fig. 1d reveals that the specialist's attention weights have a large angular distance ω_l from the generalist's, causing destructive interference regardless of coefficient pattern. Thus, At 7 B, layer-adaptive SLERP can still fine-tune the balance between instruction and reasoning when parents are *already* strong generalists, but fails when one parent is highly specialised and architecturally divergent.

A decision tree for the work is shown in Figure 2. Future work must couple Eq.(4) with weight-scope or Fisher alignment to handle such heterogeneous 7B pairs.

G. Geodesic Behavior Across Scales

We analyze how the geodesic property of SLERP (Sec. III) manifests empirically across scales and pair types, using only the results reported in Tables I, II, III, IV, V, VI, and VII.

In Table I, the *layer-adaptive* variant collapses (Average 3.61), whereas a *global* SLERP with a single t achieves 8.83 and the pretrained baseline is 8.14. This indicates that, at very small capacity, per-layer angular adjustments can oversteer modules away from a shared tangent direction, while a uniform geodesic path remains comparatively stable.

Table III compares layer-adaptive merges (PRYMMAL V1–V5) against global SLERP baselines. The adaptive variants attain Average 16.44–16.68 (e.g., V1 and V2: 16.68; V3: 16.45; V4: 16.44), while the global

baselines reach 15.74 (ECE_Poirot, t=0.5) and 15.08 (MiniQwenMathExpert, t=0.5). Beyond the mean, adaptive schedules also moderate trade-offs: for example, V4 preserves IFEVAL (33.24) *and* raises MUSR (12.09), whereas MiniQwenMathExpert pushes MATH (11.40) but lowers IFEVAL (27.95). These patterns are consistent with the hypothesis that respecting per-layer angular structure reduces destructive interference relative to a single global coefficient.

The base landscape in Table IV is heterogeneous: gemma-2-2b-it-chinese-dpo averages 19.62, while Llama-3.2-3B-Instruct (scaled here as a 3B baseline) averages 24.20. Within the *Gemma-only* merges of Table V, layer-adaptive SLERP reaches 21.16 (V1) and 21.07 (V2), improving over the best Gemma base (19.62). In contrast, cross-family pairings (V3/V4) underperform (Averages 11.81 and 11.63), and applying the same schedule to weak parents (MiniMathExpert-2.6B) yields only 12.49. Together, these results show that a geodesic, layer-wise scheme is beneficial when parents are architecturally compatible (here, within-family Gemma), but compatibility remains a prerequisite.

Table VI shows strong baselines (e.g., MedIT-Mesh 28.32, Phi-3.5-mini 28.18). The merged models in Table VII achieve Averages around 23–25.5 (e.g., ECE-EIFFEL-3Bv3 25.50), i.e., below the best single baselines but generally balanced across tasks. For instance, ECE-EIFFEL-3Bv3 maintains competitive MUSR (18.31) and MMLU-PRO (33.06) while avoiding sharp regressions on other metrics. This is consistent with the interpretation that geodesic interpolation can preserve multiple competencies when parent manifolds are partially aligned, even if it does not exceed the strongest individual baseline at this scale.

1) Entreprise deployment: A 7B model[61] merged with our layer-adaptive SLERP recipe has already proven its practical value: it climbed to the #1 spot in the 7 B class on the Open LLM Leaderboard (2024) and now drives several production systems. TW3 Partners employs it in a customer-support agent that handles 45% of daily tickets end-to-end; Racine.AI embeds the same checkpoint in a multilingual data-extraction pipeline with a 35% latency reduction versus an ensemble baseline; and the French legal-tech platform LexiaPro has integrated the model into two new services- Elixir (IA Documentaire) for secure document analysis and Gilbert (IA Conversationnelle) for real-time meeting summarisation and insight generation, showing that our merging guidelines translate directly into state-of-the-art, enterprise-grade deployments.

V. LIMITATIONS AND FUTURE WORK

a) Scope and baselines: This short, practice-focused paper synthesizes empirical lessons but does not report head-to-head numbers against recent adaptive/fisher/sparsity-based approaches (e.g., AdaMerging, Fisher-weighted, TIES), nor multi-parent or cross-architecture (tokenizer-mismatched) settings at scale. Our templates were validated internally and are representative but not exhaustive.

b) Heuristic schedules: The layer coefficients $\{c_l\}$ follow heuristic depth and type-aware ramps (S/Q). While motivated by geometric and functional considerations, they are not learned end-to-end here; stability in extreme specialization (e.g., math-only vs. instruction-only) is not guaranteed.

c) Compatibility sensitivity: Success depends on architectural/vocabulary proximity and parameter similarity. Strong angular divergence or low tokenizer overlap can lead to collapses at larger scales (7B). Embedding alignment and conservative schedules mitigate, but a full theoretical characterization is pending.

d) Statistical reporting: Many observations are aggregated trends across internal runs; comprehensive variance analysis and paired significance testing across seeds and datasets are future work for a journal version.

e) Operational considerations: While merging reduces serving cost relative to ensembling, practical risks remain: quantization drift after merge, optimizer/state mismatches, and potential degradation under distribution shift or adversarial prompts.

Future research work will focus on:

1) Theory and diagnostics: Formalize conditions under which layer-wise SLERP reduces interference (e.g., NTK linearization, representation-similarity/angle bounds) and validate CKA/cosine-based compatibility predictors.
2) Automated schedules: Replace hand-crafted ramps with a budgeted search over a six-scalar family (Bayesian/CMA-ES) and release code, ranges, and seeds; explore meta-learners that predict c_l from parent statistics.
3) Comparative evaluation: Add head-to-head studies with AdaMerging, Fisher-weighted, and TIES; include multi-parent merges (e.g., Riemannian/Karcher barycenters) and star/iterative merges.
4) Cross-architecture/tokenizer alignment: Systematically test embedding-space alignment (orthogonal/CCA/Procrustes), tokenizer remapping, and shared-subword bridges to enable safe merging across families.
5) Robustness and safety: Evaluate robustness to distribution shift, jailbreak prompts, and perturbations; incorporate safety/guardrail objectives into $\mathcal{J}$ during schedule selection.
6) Systems and efficiency: Quantization-aware merging (dequant–merge–requant with error control), streaming/zero-copy implementations, and on-device merges for edge deployments.
7) Neuro-symbolic [23] approach: Include logic as a symbolic component that influences higher ordered learning [24].

VI. Conclusion

This paper introduced Layer-Adaptive SLERP, a geometry-preserving, layer-wise merging strategy that reliably fuses complementary expertise across LLM families from 0.5 B to 7 B parameters. Our empirical study (the largest of its kind) shows that a simple, heuristically designed coefficient ramp can i) outperform single-task fine-tunes, ii) outclass global SLERP on six of seven benchmarks at 1B-3B, and iii) yield a 7B model that tops the Open LLM Leaderboard and is already powering real-world enterprise applications.

References

[1] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T.-S. Chua, "Data-efficient fine-tuning for llm-based recommendation," in *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 2024, pp. 365–374.

[2] D. Tam, M. Li, P. Yadav, R. B. Gabrielsson, J. Zhu, K. Greenewald, M. Yurochkin, M. Bansal, C. Raffel, and L. Choshen, "Llm merging: Building llms efficiently through merging," in *NeurIPS 2024 Competition Track*, 2024.

[3] T. Akiba, M. Shing, Y. Tang, Q. Sun, and D. Ha, "Evolutionary optimization of model merging recipes," *Nature Machine Intelligence*, vol. 7, no. 2, pp. 195–204, 2025.

[4] W. Li, Y. Peng, M. Zhang, L. Ding, H. Hu, and L. Shen, "Deep model fusion: A survey," *arXiv preprint arXiv:2309.15698*, 2023.

[5] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 245–254.

[6] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[7] A. M. O. Camacho, S. Horoi, G. Wolf, and E. Belilovsky, "Non-uniform parameter-wise model merging," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 5946–5954.

[8] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[9] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei *et al.*, "The flan collection: Designing data and methods for effective instruction tuning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 631–22 648.

[10] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International conference on machine learning*. PMLR, 2022, pp. 23 965–23 998.

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

[12] M. S. Matena and C. A. Raffel, "Merging models with fisher-weighted averaging," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 703–17 716, 2022.

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

[14] W. Lu, R. K. Luu, and M. J. Buehler, "Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities," *npj Computational Materials*, vol. 11, no. 1, p. 84, 2025.

[15] Y. K. Jang, D. Kim, B. He, Z. Meng, and S.-N. Lim, "Slerp+: Spherical linear interpolation for unified compositional retrieval."

[16] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao, "Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities," *arXiv preprint arXiv:2408.07666*, 2024.

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

[18] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal, "Resolving interference when merging models," *arXiv preprint arXiv:2306.01708*, vol. 1, no. 3, 2023.

[19] L. Yu, B. Yu, H. Yu, F. Huang, and Y. Li, "Language models are super mario: Absorbing abilities from homologous models as a free lunch," in *Forty-first International Conference on Machine Learning*.

[20] P. Yadav, T. Vu, J. Lai, A. Chronopoulou, M. Faruqui, M. Bansal, and T. Munkhdalai, "What matters for model merging at scale?" *arXiv preprint arXiv:2410.03617*, 2024.

[21] C. Goddard, S. Siriwardhana, M. Ehghaghi, L. Meyers, V. Karpukhin, B. Benedict, M. McQuade, and J. Solawetz, "Arcee's mergekit: A toolkit for merging large language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024, pp. 477–485.

[22] J. Männistö, "A comparative study of peft approaches for language models of code," 2024.

[23] B. P. Bhuyan, A. Ramdane-Cherif, R. Tomar, and T. Singh, "Neuro-symbolic artificial intelligence: a survey," *Neural Computing and Applications*, vol. 36, no. 21, pp. 12 809–12 844, 2024.

[24] Y. Moh Ousellam, B. P. Bhuyan, R. Fissoune, G. Ivanova, and A. Ramdane-Cherif, "Learning directed knowledge using higher-ordered neural networks: Building a predictive framework," *Applied Sciences*, vol. 15, no. 20, p. 11085, 2025.