NOT ALL UPDATES HELP: FUNCTIONAL PATH ATTRIBUTION FOR PERSONALITY VECTOR MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Personality vector merging enables training-free personality control by adding parameter displacements obtained from personality fine-tuning. However, existing methods typically treat the entire displacement as a coherent personality vector. We show that this assumption overlooks substantial functional cancellation: along parameter-space merging paths, many parameter displacements contribute negatively to the target personality objective, and including them can weaken the resulting personality behavior. We propose Functional Path Attribution, which integrates parameter-wise contributions along personality merging paths and selectively retains the most functionally constructive updates to construct sparse personality vectors. Across Big Five personality control experiments, these vectors achieve stronger target-trait expression than their source fully fine-tuned models while substantially better preserving general capabilities, outperforming random, magnitude-based, and other parameter-selection baselines. These results show that personality vectors are functionally heterogeneous and that path-aware selective merging provides a more effective approach to parameter-space personality control.

1 INTRODUCTION

As large language models (LLMs) (Brown et al., 2020) increasingly serve as persistent interactive agents (Park et al., 2023), controlling not only what they can do but also how they behave has become important for model personalization (Salemi et al., 2024). Personality provides a structured characterization of such behavioral variation (Serapio-García et al., 2025) and has received growing attention in personalized assistants (Mok et al., 2025), role-playing agents (Wang et al., 2024b), and socially interactive systems (Zhou et al., 2024). Existing approaches induce personality through prompting, model adaptation, or interventions on internal representations (Serapio-García et al., 2025; Li et al., 2025; Deng et al., 2025; Bhandari et al., 2026). Complementary to these approaches, personality vector merging represents a trait as the parameter displacement between a base model and its personality-adapted counterpart, enabling training-free scaling, composition, and transfer (Sun et al., 2025).

However, a fine-tuning displacement is the cumulative result of optimizing a training objective, rather than a purified representation of the target personality. Treating the entire displacement as a coherent semantic direction therefore conflates *parameters that changed during adaptation* with *parameters that functionally promote the desired personality during merging*. Existing sparsification methods select updates using generic criteria such as weight magnitude (Han et al., 2015), update magnitude (Yadav et al., 2023), or random masking (Yu et al., 2024), but do not determine whether individual updates reinforce or suppress the target personality along the merging trajectory.

Our analysis reveals a systematic phenomenon that we term **Functional Cancellation**: parameter updates within the same fine-tuning displacement can contribute with opposite signs to the target-trait score and partially cancel one another during merging. Unlike redundant updates, which are merely dispensable, negatively attributed updates actively counteract the desired personality. Consequently, a sparse subvector containing only functionally constructive updates can produce stronger personality control than the complete displacement. This reframes personality-vector sparsification from a compression problem into a *functional selection* problem.

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.

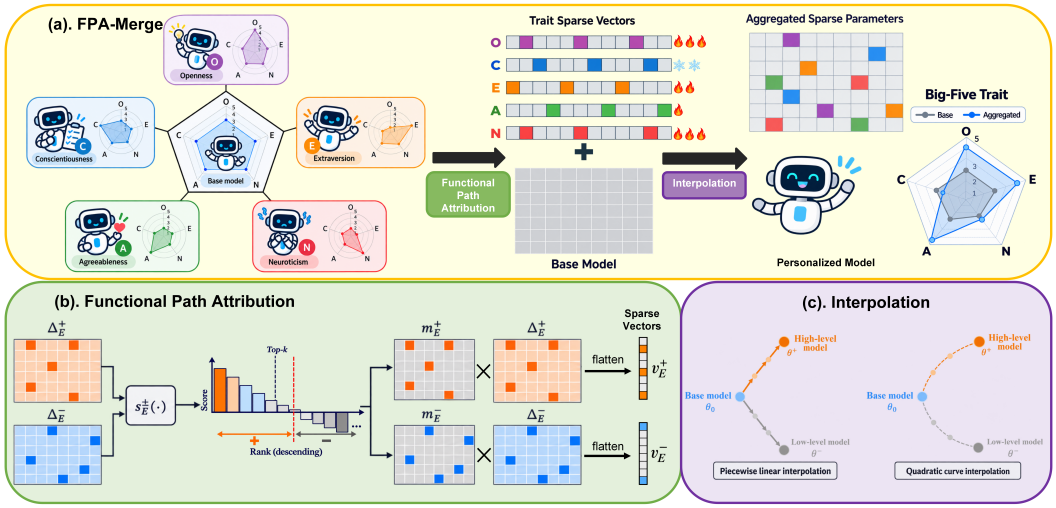


Figure 1: Overview of FPA-Merge. FPA attributes personality changes along a parameter-space merging trajectory and selects functionally constructive updates to form sparse personality vectors.

To identify such updates, we introduce **Functional Path Attribution (FPA)**. FPA defines a contrastive personality score and decomposes its change along a parameter-space merging trajectory into coordinate-wise contributions of individual parameter updates. While path-integrated gradients (Sundararajan et al., 2017) have previously been applied in parameter space for sensitivity estimation (Cui et al., 2026), FPA applies path attribution directly to personality-vector merging. Building on these attributions, **FPA-Merge** retains the most functionally constructive coordinates to form sparse personality vectors, reducing functional cancellation while preserving continuous personality control and multi-trait composition.

Across all Big Five traits and poles, FPA-Merge achieves stronger target-trait control while substantially better preserving general capabilities than dense personality merging and parameter-selection baselines. Its sparse vectors surpass their source Full SFT endpoints in 7/10 trait-pole settings while retaining only 5% of parameter coordinates. They further support continuous personality control, multi-trait composition, and transfer to task-specialized expert models.

Our main contributions are:

- We identify **Functional Cancellation** in personality vectors: fine-tuning-induced parameter updates can make opposing functional contributions during personality merging.
- We propose **Functional Path Attribution (FPA)** and **FPA-Merge**, which attribute personality changes along parameter-space merging trajectories and retain the most functionally constructive updates to build sparse personality vectors.
- Extensive experiments across Big Five personality control show that FPA-Merge achieves stronger target-trait expression with substantially better preservation of general capabilities, while retaining continuous control, multi-trait composition, and expert-model transfer.

2 Related Work

Personality Control in LLMs. Personality control in large language models (LLMs) is commonly studied under the Big Five framework—Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (McCrae & John, 1992). Existing approaches include prompting and psychometric shaping (Serapio-García et al., 2025), training-based adaptation with datasets such as BIG5-CHAT (Li et al., 2025), and interventions on internal representations (Deng et al., 2025; Bhandari et al., 2026; Zheng et al., 2026). These representation-level studies also suggest that opposing personality poles may rely on distinct internal structures, motivating their explicit treatment. Most closely related to our setting, Personality Vector represents a trait as the parameter difference between a

personality-adapted model and its base model, enabling continuous control, composition, and transfer (Sun et al., 2025). We examine the functional structure within this fine-tuning displacement rather than treating it as a uniform personality direction.

Task Vectors and Sparse Model Merging. Task Arithmetic interprets fine-tuning displacements as additive directions in parameter space (Ilharco et al., 2023). Subsequent methods reduce redundancy and interference through update magnitude, sign consistency, random sparsification, or task-specific parameter support, including TIES-Merging, DARE, and TALL-Masks (Yadav et al., 2023; Yu et al., 2024; Wang et al., 2024a). FPA instead characterizes individual updates by their signed contribution to a personality-specific behavioral objective along the merging trajectory, distinguishing reinforcing updates from those that counteract the desired behavior.

Parameter-Space Attribution. Integrated Gradients (IG) attributes changes in a scalar model output by integrating gradients along a path (Sundararajan et al., 2017). The same principle has also been studied in parameter space: FAIG attributes changes in network function to parameter changes (Xie et al., 2021), while IGU-LoRA uses path-integrated parameter sensitivities for adaptive rank allocation (Cui et al., 2026). FPA applies this perspective to personality-vector merging, using a contrastive personality objective along the merging trajectory. The resulting signed coordinate-level attributions are used to identify functionally constructive updates and form sparse personality vectors.

3 Functional Characterization of Personality Vectors

We first formalize personality adaptation as a parameter displacement and define a contrastive personality score for attributing personality changes along parameter-space merging trajectories.

3.1 Personality Vector Formulation

Let $\theta_0 \in \mathbb{R}^d$ denote the parameters of a base language model. For a Big Five trait $t \in T = \{\mathrm{O}, \mathrm{C}, \mathrm{E}, \mathrm{A}, \mathrm{N}\}$ and pole $p \in \{+, -\}$, personality adaptation produces θ_t^p with displacement

$$\Delta_t^p = \theta_t^p - \theta_0.$$

Following personality-vector merging (Sun et al., 2025), the corresponding merged model is

$$\theta_t^p(\alpha) = \theta_0 + \alpha \Delta_t^p, \qquad \alpha \in \mathbb{R}. \tag{1}$$

Because Δ_t^p is an endpoint difference aggregating all changes induced by personality adaptation, it does not distinguish updates that reinforce the target personality from those with negligible or opposing effects during merging. FPA addresses this distinction by attributing personality change to individual parameter coordinates along the merging trajectory.

3.2 FPA: From Integrated Gradients to Functional Path Attribution

Contrastive personality score. Token-level likelihood reflects not only personality but also response-specific factors such as length, lexical choice, formatting, and content. We therefore exploit paired responses to the same prompt. For prompt x, let y_t^p and y_t^{-p} denote responses expressing the target and opposite poles of trait t. We define the token-normalized negative log-likelihood as

$$\mathcal{L}_{\mathrm{NLL}}(\theta; x, y) = -\frac{1}{|y|} \sum_{j=1}^{|y|} \log P_\theta(y_j \mid x, y_{<j}), \tag{2}$$

and the contrastive personality score as

$$S_t^p(\theta) = \mathbb{E}_{(x, y_t^p, y_t^{-p})} \left[\mathcal{L}_{\mathrm{NLL}}(\theta; x, y_t^{-p}) - \mathcal{L}_{\mathrm{NLL}}(\theta; x, y_t^p) \right]. \tag{3}$$

A larger S_t^p indicates greater relative preference for the target pole over its opposite, providing the scalar signal used for parameter attribution.

Functional Path Attribution. Integrated Gradients (IG) (Sundararajan et al., 2017) attributes output changes by integrating gradients along a path. We apply the same principle in parameter space to decompose changes in S_t^p along a personality merging trajectory.

Let $\gamma : [a, b] \to \mathbb{R}^d$ be a differentiable parameter-space path with $\gamma(a) = \theta_a$ and $\gamma(b) = \theta_b$. The Functional Path Attribution of parameter coordinate i is

$$A_{t,i}^p[\gamma] = \int_a^b \left.\frac{\partial S_t^p(\theta)}{\partial \theta_i}\right|_{\theta=\gamma(\alpha)} \frac{d\gamma_i(\alpha)}{d\alpha}\, d\alpha. \tag{4}$$

By the chain rule,

$$\sum_{i=1}^{d} A_{t,i}^p[\gamma] = S_t^p(\theta_b) - S_t^p(\theta_a), \tag{5}$$

so the total score change is exactly decomposed across parameter coordinates. Positive attributions reinforce the target personality along γ, whereas negative attributions oppose it. FPA is intentionally path-dependent because it characterizes parameter effects under the merging intervention being analyzed.

4 FPA-Merge: Selective Personality Vector Merging

We instantiate FPA on bipolar personality trajectories and use the resulting attributions to construct sparse personality vectors.

4.1 Bipolar Personality Paths

Opposing personality poles can exhibit distinct internal representations rather than forming a single inverse direction (Zheng et al., 2026). Accordingly, we do not assume $\Delta_t^+ = -\Delta_t^-$; their parameter-space asymmetry is examined in Appendix C.5. We instead construct a base-preserving path through the independently adapted endpoints, satisfying $\gamma_t(-1) = \theta_t^-$, $\gamma_t(0) = \theta_0$, and $\gamma_t(1) = \theta_t^+$.

Piecewise-linear path.

$$\gamma_t^{\mathrm{PL}}(\alpha) = \begin{cases} \theta_0 + \alpha\Delta_t^+, & \alpha \in [0, 1], \\ \theta_0 - \alpha\Delta_t^-, & \alpha \in [-1, 0]. \end{cases} \tag{6}$$

This path preserves both observed fine-tuning directions but has a tangent discontinuity at the base model.

Quadratic path.

$$\gamma_t^{\mathrm{Q}}(\alpha) = \theta_0 + \frac{\alpha}{2}\left(\Delta_t^+ - \Delta_t^-\right) + \frac{\alpha^2}{2}\left(\Delta_t^+ + \Delta_t^-\right), \qquad \alpha \in [-1, 1]. \tag{7}$$

The quadratic path is smooth through the base model and jointly depends on both personality endpoints throughout the trajectory.

4.2 Attribution-Guided Sparse Support

Because $S_t^p(\theta) = -S_t^{-p}(\theta)$, we orient the bipolar path from θ_t^- to θ_t^+ and use S_t^+ for attribution. Thus, $A_{t,i}^+[\gamma_t] > 0$ indicates that parameter coordinate i promotes the high pole relative to the low pole along the trajectory. Reversing both the path and the contrastive score leaves the attributions unchanged, so this convention is without loss of generality.

Given retention ratio $\rho \in [0, 1]$, FPA-Merge retains the coordinates with the largest attributions:

$$M_{t,i}(\rho) = \mathbb{I}\left\{A_{t,i}^+[\gamma_t] \geq \tau_t(\rho)\right\}, \tag{8}$$

where $\tau_t(\rho)$ retains the top ρ fraction of coordinates. The shared mask is applied to both personality displacements,

$$v_t^p = M_t \odot \Delta_t^p, \qquad p \in \{+, -\}, \tag{9}$$

yielding sparse vectors (v_t^-, v_t^+) supported on the coordinates most constructive for the bipolar personality transition.

4.3 Multi-Trait Composition and Expert-Model Transfer

Let $\tilde{\gamma}_t(\alpha)$ be the sparse counterpart of $\gamma_t(\alpha)$ obtained by replacing (Δ_t^-, Δ_t^+) with (v_t^-, v_t^+), and define $\delta_t(\alpha) = \tilde{\gamma}_t(\alpha) - \theta_0$. For traits $\mathcal{T}$ with coefficients $\{\alpha_t\}_{t\in\mathcal{T}}$, we compose

$$\theta' = \theta + \sum_{t\in\mathcal{T}} \delta_t(\alpha_t), \tag{10}$$

where θ is either the base model θ_0 or an expert model derived from it. Each α_t controls the polarity and strength of trait t, enabling training-free multi-trait control on both base and expert models.

The complete FPA-Merge procedure, including numerical path integration, sparse support construction, and multi-trait composition, is summarized in Algorithm 1 in Appendix H.

5 Experimental Setup

Models. We conduct main experiments on Qwen3-4B-Instruct-2507 (Team, 2025) and additionally evaluate Qwen3.5-9B (Qwen Team, 2026) for robustness across model scales and checkpoints (Appendix G).

Datasets and Benchmarks. For personality adaptation, we use BIG5-CHAT (Li et al., 2025), which provides separate high- and low-pole subsets for all five Big Five traits. We perform full-parameter fine-tuning on each subset to obtain the personality-adapted checkpoints used to construct parameter displacements. For personality evaluation, we use TRAIT (Lee et al., 2025), an LLM-oriented situational multiple-choice benchmark that avoids direct self-report assessment. General capabilities are evaluated on GPQA (Rein et al., 2023), AIME25 (Zhang & Math-AI, 2025), and MATH-500 (Lightman et al., 2023). For expert-model transfer, we fine-tune the same base model on MEDMCQA (Pal et al., 2022) to construct a medical expert and measure its medical QA accuracy before and after personality-vector transfer.

Baselines. We compare FPA-Merge with training-, prompting-, neuron-, and parameter-space baselines. Training-based methods include Full SFT, LoRA (Hu et al., 2022), DPO (Rafailov et al., 2023), and OPD (Yang et al., 2026); we also include trait-specific prompting (Serapio-García et al., 2025) and NPTI (Deng et al., 2025). For parameter-space merging, we compare against: (i) **Weight Magnitude**, selecting coordinates by $|\theta_i|$; (ii) **Update Magnitude**, selecting by $|\Delta\theta_i| = |\theta_i - \theta_0|$ (Yadav et al., 2023); (iii) **MLP Merge**, restricting the displacement to MLP modules; (iv) **Task Vector**, for a target pole p, we reverse the dense displacement of the opposite pole, i.e., $\theta_{\mathrm{TV}}^p = \theta_0 - \Delta_t^{-p}$, following task-vector arithmetic (Ilharco et al., 2023; Sun et al., 2025); and (v) **DARE**, randomly dropping displacement coordinates and rescaling the retained updates (Yu et al., 2024). For sparsification-based baselines, we match the retention ratio to FPA-Merge and rescale retained updates; the rescaled variants perform better under the same sparsity budget (Appendix C.2). Detailed prompt and baseline configurations are provided in Appendix E.2 and A.

Implementation Details. For FPA-Merge and sparsification-based baselines, we use $\rho = 5\%$ as the default retention ratio; Appendix C.1 reports the corresponding sparsity–performance trade-off. Functional attribution is estimated using 200 samples per trait drawn from the held-out BIG5-CHAT validation split and excluded from personality fine-tuning, with the path integral approximated by 8-point Gauss–Legendre quadrature. A sweep over 1–64 quadrature points shows that both top-ρ support stability and downstream personality performance stabilize around eight points (Appendix C.3).

All methods use the same system prompt, chat template, and decoding configuration within each benchmark. Full training, baseline, prompt, and evaluation protocols are provided in Appendix A, E, and F.

6 Results and Analysis

We first evaluate the overall personality-capability trade-off of FPA-Merge against a broad set of baselines and across all Big Five traits and poles. We then examine whether the resulting sparse personality vectors remain compositional, transferable to task-specialized models, and continuously controllable, followed by further analyses and ablations.

Table 1: **Overall performance on Extraversion-High.** Personality control is measured by TRAIT, while GPQA, AIME25, and MATH-500 measure general capability. Higher is better for all columns.

Method	TRAIT	GPQA	AIME25	MATH-500
Base	37.6	60.56	47.86	92.4
Prompt	71.2	58.84	46.67	92.4
Full SFT	68.1	38.64	39.74	83.0
LoRA	47.2	59.85	47.40	93.0
DPO	47.3	55.24	43.89	90.4
OPD	55.5	48.38	41.82	89.4
NPTI	58.3	41.91	39.08	85.6
Task Vector	75.0	49.75	41.25	88.4
DaRE	64.9	43.18	35.31	83.0
Weight Magnitude (θ)	60.8	43.69	40.84	86.4
Update Magnitude (Δ)	58.9	42.55	40.10	83.8
MLP Merge	63.4	46.97	40.00	82.2
FPA-Merge (PL)	**90.1**	57.17	47.56	93.4
FPA-Merge (Q)	89.2	59.06	45.78	91.8

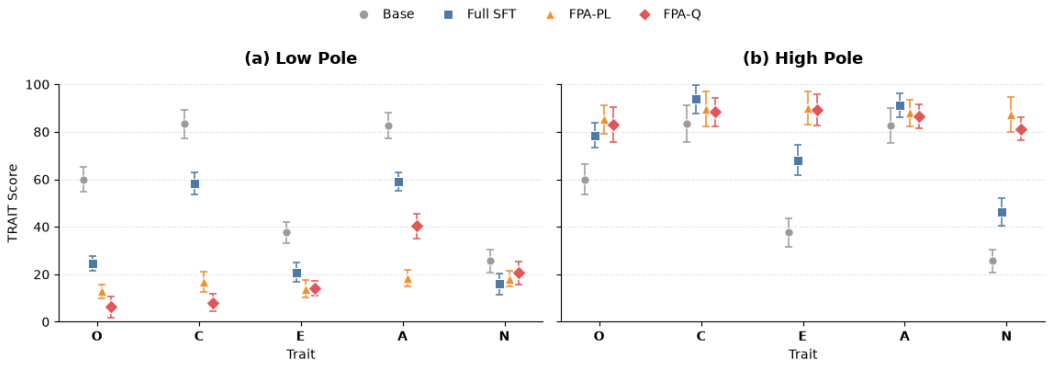


Figure 2: **Trait-wise personality control across all Big Five dimensions.** Raw TRAIT scores for the base model, Full SFT, and the two FPA-Merge path variants across high and low personality poles. Higher scores correspond to stronger high-pole expression, while lower scores correspond to stronger low-pole expression.

6.1 Overall Performance

We first compare FPA-Merge with all baselines on Extraversion-High, and then evaluate its consistency across all five Big Five traits and both poles. Piecewise-linear and quadratic paths are reported separately.

Table 1 shows that FPA-Merge achieves the strongest personality control while largely preserving general capability. The strongest non-FPA baseline, Task Vector, reaches a TRAIT score of 75.0, whereas FPA-Merge obtains 90.1 (PL) and 89.2 (Q). Meanwhile, Full SFT substantially degrades GPQA/AIME25/MATH-500 from 60.56/47.86/92.4 to 38.64/39.74/83.0, while FPA-Merge retains 57.17/47.56/93.4 (PL) and 59.06/45.78/91.8 (Q).

Figure 2 further shows that this behavior extends across personality dimensions. Both FPA-Merge variants move all ten trait-pole settings in the intended direction and outperform the corresponding Full SFT endpoint in 7/10 settings, despite retaining only 5% of parameter coordinates. The exceptions are Conscientiousness-High, Agreeableness-High, and Neuroticism-Low.

The same qualitative behavior is also observed on Qwen3.5-9B (Appendix G), suggesting that the effect is not specific to the 4B base model.

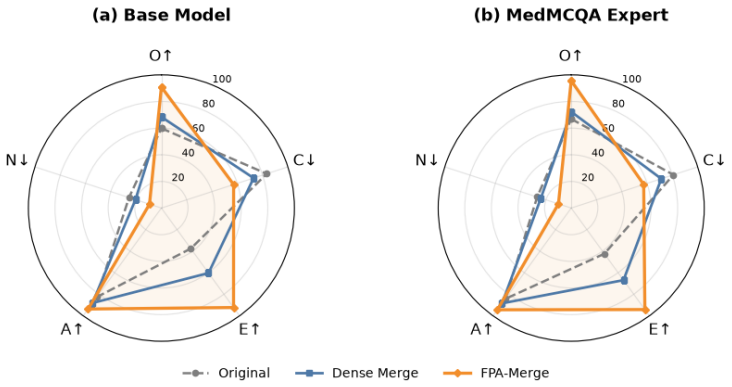


Figure 3: **Multi-trait composition and expert-model transfer.** Big Five profiles before and after dense and FPA-based personality merging on the base model and the MedMCQA expert.

Table 2: **MedMCQA capability after personality transfer.**

Model	Original	+ Dense Merge	+ FPA-Merge
Base Model	46.04	28.97	44.57
MedMCQA Expert	63.33	39.56	60.24

6.2 Multi-Trait Composition and Expert-Model Transfer

We next examine whether multiple sparse personality vectors can be jointly composed and transferred to a task-specialized model. This setting is non-trivial because personality dimensions are not fully disentangled in parameter space: the selected supports of different traits overlap by approximately $4\times$ the random-selection expectation (Appendix D). This substantial shared support is consistent with the coupled trait changes observed under composition, rather than a fully independent parameterization of personality dimensions. We construct a mixed-polarity target profile, $[O, C, E, A, N] = [+, -, +, +, -]$, and apply the same personality intervention to both the base model and a MedMCQA expert initialized from the same checkpoint. Dense personality merging is evaluated under the same coefficients for comparison.

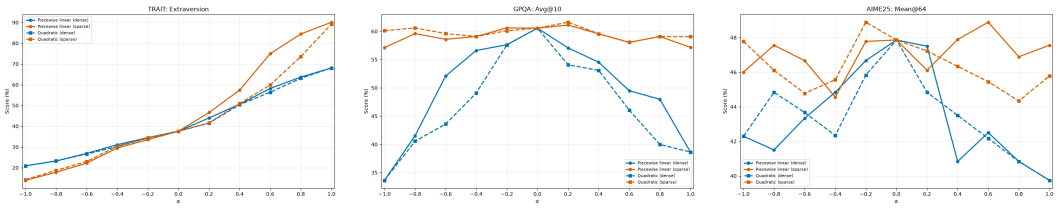
As shown in Figure 3, both dense and FPA-based merging move all five personality dimensions toward the intended profile. FPA-Merge produces substantially stronger shifts, and yields highly similar post-merge personality profiles on the base and expert models despite their different initial profiles. This indicates that the sparse personality vectors remain compositional and can be directly reused after task-specific adaptation.

We further evaluate whether personality transfer preserves the capabilities acquired by the expert model. Table 2 reports MedMCQA accuracy before and after personality merging.

Task-specific training improves MedMCQA accuracy from 46.04 to 63.33. Dense personality merging substantially degrades this capability to 39.56, whereas FPA-Merge retains 60.24, only 3.09 points below the original expert. Together, these results show that FPA-Merge supports multi-trait personality composition while transferring effectively to a task-specialized model with limited loss of its acquired capability.

6.3 Continuous Personality Control

We further examine whether FPA-Merge preserves the continuous control property of dense personality vectors. Using Extraversion as a representative setting, we vary the merging coefficient α along the bipolar trajectory and compare dense merging with the top-5% FPA-Merge vectors under both piecewise-linear and quadratic paths.



(a) Extraversion (b) GPQA (c) AIME25

Figure 4: **Continuous personality control and capability preservation.** Performance across merging coefficients α for dense and $\rho = 0.05$ FPA-Merge vectors under piecewise-linear and quadratic paths.

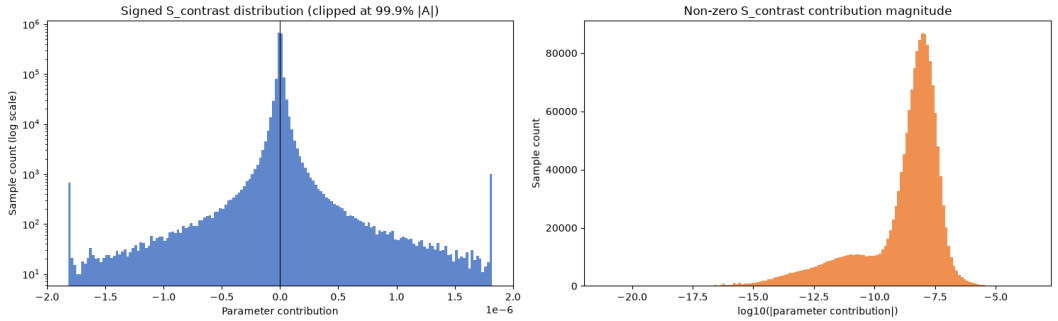


Figure 5: **Distribution of parameter-wise FPA contributions.** Signed contributions span both positive and negative values, while their magnitudes vary substantially across parameters, revealing strong functional heterogeneity within the dense personality vector.

Figure 4 shows that both dense and sparse vectors provide smooth control over Extraversion as α varies from the low to the high pole. FPA-Merge further expands the controllable range, reaching approximately 90% TRAIT score at $\alpha = 1$, compared with about 68% for dense merging, under both path constructions. Meanwhile, sparse merging substantially stabilizes general capability across the trajectory: GPQA remains close to its base-model performance, whereas dense merging degrades sharply toward both poles; AIME25 exhibits the same overall trend despite larger fluctuations. These results show that functional sparsification preserves continuous personality control while enabling stronger steering with substantially less capability degradation.

6.4 Functional Cancellation

We examine *Functional Cancellation*, where parameter updates within the same personality displacement contribute in opposing directions and partially offset each other. As shown in Figure 5, FPA contributions span both positive and negative values, with magnitudes varying by several orders. This confirms that dense personality vectors are functionally heterogeneous and contain updates that counteract, rather than uniformly reinforce, the target personality.

We further accumulate parameter coordinates in descending order of their FPA scores, with the total contribution of the complete dense vector normalized to one. Figure 6 shows that the cumulative contribution rises to roughly $15\times$ the full-vector level in the sparse regime and then declines as lower-ranked updates are restored, revealing strong functional cancellation within the dense personality vector.

The same pattern is reflected in realized personality behavior: for Extraversion-high, retaining the top 20% coordinates increases the TRAIT score from 68.1 for the full displacement to 97.3, while restoring lower-ranked updates weakens the effect. We term this phenomenon *Functional Cancellation*: unlike redundant updates that contribute little, opposing updates actively offset constructive ones. This explains why a sparse FPA-selected vector can outperform its dense counterpart.

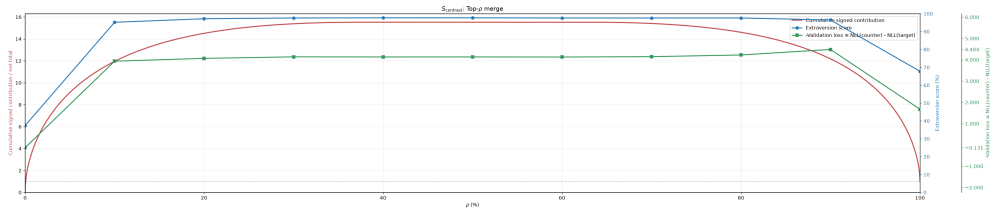


Figure 6: **Functional cancellation under progressive parameter restoration.** Parameters are added in descending FPA order, with the full-vector contribution normalized to 1. Cumulative contribution and personality strength peak in the sparse regime and decline as lower-ranked updates are restored.

The same qualitative cancellation pattern also holds under a target-only attribution objective (Appendix C.4).

7 Discussion and Limitations

Our results show that personality fine-tuning displacements are not functionally homogeneous: parameter updates produced by the same adaptation process can contribute differently, and sometimes oppositely, to personality behavior during model merging. This Functional Cancellation differs from conventional parameter redundancy, since removing an update may improve personality control not merely because it is unnecessary, but because it counteracts the desired behavioral change. FPA therefore provides a behavior-aware criterion for selective merging by evaluating parameter displacements with respect to a specified personality functional along the merging trajectory. Its attributions should nevertheless be interpreted relative to the chosen path, model parameterization, and behavioral functional, rather than as intrinsic, path-independent measures of parameter importance. FPA also requires an offline attribution stage with gradient evaluations, and our current transfer experiments are restricted to expert models derived from the same base checkpoint.

In this work, we introduced Functional Path Attribution (FPA) and FPA-Merge to characterize and selectively retain the parameter updates that functionally support personality control. Across Big Five experiments, FPA-Merge achieves stronger target-trait expression while better preserving model capabilities than dense personality merging and parameter-selection baselines. The resulting sparse personality vectors remain continuously controllable, compositional across multiple traits, and transferable to task-specialized models. Overall, our findings suggest that effective parameter-space behavior control depends not only on identifying what changes during adaptation, but on determining which of those changes actually help realize the desired behavior.

AI use statement

In this work, we did not use generative AI tools to formulate research questions, develop the methodology, design experiments, generate experimental results, interpret results, or formulate scientific claims. Generative AI tools were used to assist with editing the manuscript for grammar, wording, clarity, and readability, and with writing selected data-processing functions based on data-processing procedures specified by the authors. Certain illustrative elements in Figure 1, such as the robot graphics, were generated with the assistance of generative AI. The overall figure design, conceptual structure, layout, annotations, and all other graphical elements were created and assembled by the authors. All AI-assisted text was reviewed and revised by the authors. All AI-assisted code was inspected, tested, and verified for correctness before being used in the experiments. We take responsibility for the final content of this work, including all text, code, figures, results, claims, and other artifacts produced with the aid of generative AI.

Reproducibility Statement

We provide detailed information to support the reproducibility of this work throughout the main paper, appendix, and supplementary materials. Section 5 describes the models, datasets, baselines, and

main implementation settings. The appendix provides additional details on training configurations, baseline implementations, functional attribution, numerical integration, prompt templates, and evaluation protocols. The implementation, configuration files, and evaluation scripts are available in an anonymous code repository.[1]

REFERENCES

Pranav Bhandari, Nicolas Fay, Sanjeevan Selvaganapathy, Amitava Datta, Usman Naseem, and Mehwish Nasim. Activation-space personality steering: Hybrid layer selection for stable trait control in LLMs. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6388–6403, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.300. URL https://aclanthology.org/2026.eacl-long.300/.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Xuan Cui, HuiYue Li, Run Zeng, Yunfei Zhao, Jinrui Qian, Wei Duan, Bo Liu, and Zhanpeng Zhou. IGU-loRA: Adaptive rank allocation via integrated gradients and uncertainty-aware scoring. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MnToYQx9My.

Jia Deng, Tianyi Tang, Yanbin Yin, Wenhao yang, Xin Zhao, and Ji-Rong Wen. Neuron based personality trait induction in large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 85059–85083, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d399b67fa017f0f7670102c88507720c-Paper-Conference.pdf.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

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.

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

Seungbeen Lee, Seungwon Lim, Seungju Han, Giyeong Oh, Hyungjoo Chae, Jiwan Chung, Minju Kim, Beong-woo Kwak, Yeonsoo Lee, Dongha Lee, Jinyoung Yeo, and Youngjae Yu. Do LLMs have distinct and consistent personality? TRAIT: Personality testset designed for LLMs with psychometrics. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8412–8452, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7.

[1] https://anonymous.4open.science/r/FPA-Merge-4029/

doi: 10.18653/v1/2025.findings-naacl.469. URL https://aclanthology.org/2025.findings-naacl.469/.

Wenkai Li, Jiarui Liu, Andy Liu, Xuhui Zhou, Mona T. Diab, and Maarten Sap. BIG5-CHAT: Shaping LLM personalities through training on human-grounded data. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20434–20471, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.999. URL https://aclanthology.org/2025.acl-long.999/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Robert R. McCrae and Oliver P. John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, 1992. doi: https://doi.org/10.1111/j.1467-6494.1992.tb00970.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6494.1992.tb00970.x.

Jisoo Mok, Ik-hwan Kim, Sangkwon Park, and Sungroh Yoon. Exploring the potential of LLMs as personalized assistants: Dataset, evaluation, and analysis. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10212–10239, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.504. URL https://aclanthology.org/2025.acl-long.504/.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Gregory Serapio-García, Mustafa Safdari, Clément Crepy, Luning Sun, Stephen Fitz, Peter Romero, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. A psychometric framework for evaluating and shaping personality traits in large language models. *Nature Machine Intelligence*, 7(12):1954–1968, 2025.

Seungjong Sun, Seo Yeon Baek, and Jang Hyun Kim. Personality vector: Modulating personality of large language models by model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24667–24688, 2025.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50268–50287. PMLR, 2024a.

Noah Wang, Z.Y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.878. URL https://aclanthology.org/2024.findings-acl.878/.

Liangbin Xie, Xintao Wang, Chao Dong, Zhongang Qi, and Ying Shan. Finding discriminative filters for specific degradations in blind super-resolution. In *Advances in Neural Information Processing Systems*, volume 34, pp. 51–61. Curran Associates, Inc., 2021.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Lifan Zheng, Xue Yang, Jiawei Chen, Chenyan WU, Jingyuan Zhang, Fanheng Kong, Xinyi Zeng, Xiang Chen, and Yu Tian. DPN-LE: Dual personality neuron localization and editing for large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 30567–30591, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1528. URL https://aclanthology.org/2026.findings-acl.1528/.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. Sotopia: Interactive evaluation for social intelligence in language agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 40975–41019, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/b3075b88e583a0e98d8b24338a613060-Paper-Conference.pdf.

A EXPERIMENTAL DETAILS AND REPRODUCIBILITY

This section specifies the data splits, supervised fine-tuning setup, and training-based baselines. Unless stated otherwise, all personality models are initialized from Qwen3-4B-Instruct-2507, use the tokenizer and chat template released with that checkpoint, and are trained or evaluated with two NVIDIA RTX 4090 GPUs.

A.1 DATA CONSTRUCTION AND SPLITS

For personality adaptation, we use BIG5-CHAT (Li et al., 2025). The dataset contains 100,000 dialogues covering the five Big Five dimensions and their high and low poles. We split examples by the dataset field `original_index`, rather than by individual rows: with seed 42, 90% of unique dialogue indices are assigned to training and 10% to validation. All trait and polarity variants derived from one dialogue therefore remain in the same split. We then filter by the requested trait and pole and remove examples whose input or output begins with a known refusal template. This procedure yields ten personality-adapted models, one for each trait–pole pair.

A.2 PERSONALITY SUPERVISED FINE-TUNING

We perform full-parameter supervised fine-tuning (SFT). The system and user portions are formatted with the native chat template and masked with label -100, so the loss is evaluated only on the assistant response and the end-of-sequence token. Its objective is the response-only token-level cross-entropy, equivalently the negative log-likelihood

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{|y|}\sum_{j=1}^{|y|}\log P_\theta(y_j \mid x, y_{<j}). \tag{11}$$

Table 3 lists the effective training configuration.

Table 3: Personality SFT configuration. The effective batch size is $8 \times 4 \times 2 = 64$ sequences.

Setting	Value
Fine-tuning scope	Full parameter
Maximum sequence length	1,024
Per-device train / evaluation batch size	8 / 8
Gradient accumulation steps	4
Maximum steps	1,000
Learning rate	1.0×10^{-5}
Adam β_1, β_2	0.9, 0.999
Weight decay / warmup steps	0 / 0
Evaluation / checkpoint interval	20 / 200 steps
Random seed / data seed	42 / 42
Numerical precision	BF16; TF32 enabled
Distributed optimization	DeepSpeed ZeRO-2
Gradient checkpointing	Enabled

A.3 EXPERT-MODEL SFT

For the specialist-model experiment, we fine-tune the same base checkpoint on the official MedMCQA training split and evaluate during training on its validation split; the unlabeled test split is not used. Explanations are included in the target, and examples exceeding 1,024 tokens are removed. The official training split contains 182,822 examples before length filtering. The run uses full-parameter training, a per-device batch size of 4, eight gradient-accumulation steps, two GPUs (effective batch size 64), a learning rate of 1.0×10^{-5}, zero weight decay, a linear scheduler with a 0.03 warmup ratio, gradient clipping at 1.0, BF16, TF32, and DeepSpeed ZeRO-2. Training runs for 3,000 steps.

A.4 TRAINING-BASED BASELINES

LoRA. The LoRA baseline uses rank 16, scaling factor 32, and dropout 0.05. Adapters are applied to `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. It uses the same data split, sequence length, optimizer, learning rate, batch size, and step limit as the SFT configuration in Table 3.

DPO. For direct preference optimization, responses from the high and low poles of the same `original_index` and trait are paired. The target-pole response is the chosen response and the opposite-pole response is rejected; both share the same neutral dialogue prompt. Let y^+ and y^- denote the chosen and rejected responses, and let $r_\theta(x, y) = \log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)$. The preference component is the standard sigmoid DPO loss

$$\mathcal{L}_{\text{DPO}}(\theta) = -\log \sigma\big(\beta \left[r_\theta(x, y^+) - r_\theta(x, y^-)\right]\big) . \tag{12}$$

We use $\beta = 0.1$ and optimize $\mathcal{L}_{\text{DPO}} + 0.1\mathcal{L}_{\text{SFT}}$. The training configuration uses full-parameter optimization, a maximum sequence length of 1,024, a learning rate of 1.0×10^{-6}, zero weight decay, a linear scheduler without warmup, gradient clipping at 1.0, a per-device batch size of 8, eight accumulation steps, and two GPUs (effective batch size 128 preference pairs). Dropout is disabled and training runs for 2,000 steps in BF16 with TF32 and DeepSpeed ZeRO-2.

OPD. We use a prompt-conditioned self-distillation variant of the reverse-KL on-policy distillation (OPD) objective analyzed by Yang et al. (2026). Responses are sampled from the current policy under the neutral dialogue prompt. On the same sampled tokens, the personality-conditioned prompt and the neutral prompt define two distributions using the same model parameters. The personality-conditioned distribution supplies the target log-probabilities, while the neutral distribution is the trainable policy. Thus, this baseline does not load a separately parameterized teacher model. Let x denote the neutral dialogue input and c_t^p the extra target-trait prompt. The two conditionings induce $\pi_\theta(\cdot \mid x)$ and $\pi_\theta(\cdot \mid c_t^p, x)$, and the objective is the reverse KL

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_x \left[D_{\text{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \text{sg}[\pi_\theta(\cdot \mid c_t^p, x)])\right] . \tag{13}$$

Here, sg denotes stop-gradient on the personality-conditioned teacher branch. We estimate this reverse KL on responses sampled from $\pi_\theta(\cdot \mid x)$ and use token-level credit assignment with advantage scale 1.0 and clipping at 5.0. The additional neutral-base KL coefficient is zero, so no extra KL anchor is applied. Table 4 lists the remaining settings.

Table 4: Context-OPD configuration for the Extraversion-low baseline.

Setting	Value
Fine-tuning scope	Full parameter
Prompt / response length	256 / 128
Thinking	Disabled
Rollout temperature / top-p	0.8 / 0.95
Per-device train batch size	16
Gradient accumulation steps	4
Training steps	2,000
Learning rate	1.0×10^{-5}
Weight decay	0
Scheduler / warmup steps	Cosine / 25
Maximum gradient norm	1.0
Random seed / data seed	42 / 42
Precision	BF16; TF32 enabled
Distributed optimization	DeepSpeed ZeRO-2
Gradient checkpointing	Enabled

B FUNCTIONAL ATTRIBUTION IMPLEMENTATION

B.1 GAUSS–LEGENDRE QUADRATURE ALONG THE MERGING PATH

For an n-point rule on a normalized path coordinate $a \in [0, 1]$, we obtain the standard Gauss–Legendre nodes x_j and weights $\widetilde{w}_j$ on $[-1, 1]$ and map them as $a_j = (x_j + 1)/2$ and $w_j = \widetilde{w}_j/2$.

For a linear path $\theta(a) = \theta_0 + a\Delta$, the coordinate-wise attribution is approximated by

$$A_i^{(n)} \approx \Delta_i \sum_{j=1}^{n} w_j \left. \frac{\partial S(\theta)}{\partial \theta_i} \right|_{\theta=\theta_0+a_j\Delta}. \tag{14}$$

For non-uniform path parameterizations, the implementation additionally multiplies each integrand by the true tangent $d\gamma(a_j)/da$. We use $n = 8$ by default. The resulting nodes and weights are listed in Table 5.

Table 5: Gauss–Legendre nodes and weights on $[0, 1]$ for the default setting $n = 8$.

j	a_j	w_j
1	0.019855071751232	0.050614268145188
2	0.101666761293187	0.111190517226687
3	0.237233795041836	0.156853322938944
4	0.408282678752175	0.181341891689181
5	0.591717321247825	0.181341891689181
6	0.762766204958164	0.156853322938944
7	0.898333238706813	0.111190517226687
8	0.980144928248768	0.050614268145188

The implementation also records a numerical completeness check comparing $\sum_i A_i$ with $S(\gamma(1)) - S(\gamma(0))$; the residual reflects finite-order quadrature and finite-precision model and gradient computation.

C Selection and Ablation Details

C.1 Retention-ratio Selection

FPA-Merge ranks scalar parameter displacements by positive contrastive attribution and retains a fixed fraction ρ of the candidate scope. Thus, ρ denotes a percentage of scalar parameters rather than a percentage of tensors. Table 6 reports the ablation used to choose the default ratio. All rows use the Extraversion-high endpoint and the same linear path, attribution channel, and merge coefficient.

Table 6: Retention-ratio ablation for positive contrastive attribution. TRAIT reports the Extraversion-high score.

Retained parameters	TRAIT ↑	GPQA Avg@10 ↑	AIME25 Mean@64 ↑	MATH-500 Avg@1 ↑
Full SFT (100%)	68.1	38.64	39.74	83.0
1%	69.5	59.09	46.83	91.40
5%	90.1	57.17	47.56	93.4
10%	95.2	56.36	45.42	92.60
20%	97.3	56.28	44.58	90.60

The 5% retention ratio provides a favorable balance between target-trait expression and general-capability preservation. Relative to 1%, it increases the target TRAIT score by 20.6 points, while GPQA decreases from 59.09 to 57.17. Increasing the retained fraction to 10% doubles the number of merged parameters, but improves the TRAIT score only modestly, from 90.1 to 95.2, while GPQA decreases slightly from 57.17 to 56.36. We therefore use 5% as the default sparsity–performance trade-off and apply the same ratio to sparsification-based baselines whenever applicable.

C.2 MLP-restricted Module Merge

The Module Merge baseline restricts the dense SFT displacement to all MLP blocks. For Qwen3-4B, this scope consists of the `gate_proj`, `up_proj`, and `down_proj` submodules in every transformer layer and accounts for approximately 67% of model parameters. Because this baseline merges a complete module family rather than selecting individual parameters, we use aggregate attribution mass, rather than mean attribution per parameter, to characterize the functional contribution of each family. For Table 7, net contribution is the sum of signed S_{contrast} attributions, where

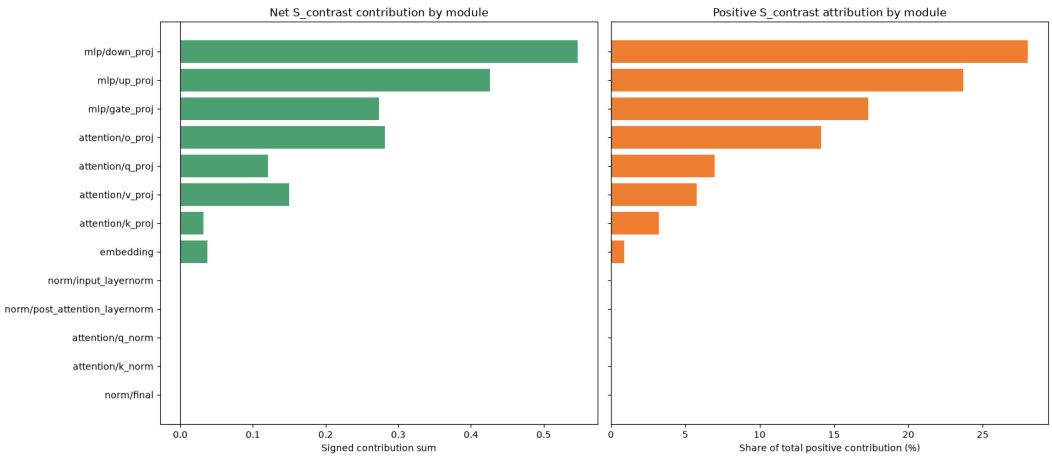


Figure 7: Aggregate S_{contrast} attribution by module type for the Extraversion-high SFT checkpoint at step 1,000. The left panel reports the sum of signed attributions; the right panel reports each module type's share of the total positive attribution. These are aggregate, not per-parameter, statistics.

$S_{\text{contrast}} \equiv S_t^+$ for this Extraversion-high run, and positive share is the fraction of the total positive attribution assigned to that family. A positive S_{contrast} attribution means that the corresponding update favors the high-pole target over the low-pole response; in general, this directional interpretation follows the designated target pole. Positive S_{contrast} denotes the positive part of the contrastive attribution and is not the target-only score alone.

Table 7: Aggregate S_{contrast} attribution by module family for the Extraversion-high SFT checkpoint at step 1,000.

Module family	Parameter share (%)	Net contribution	Positive share (%)
MLP	66.864	1.2459	68.993
Attention	23.461	0.5840	30.101
Embedding	9.670	0.0370	0.900
Normalization	0.005	−0.0002	0.006

Figure 7 further decomposes the aggregate statistics into projection and normalization module types. The three MLP projections jointly account for the largest net contribution and approximately 69% of all positive attribution mass.

The module-restricted comparison in Table 8 shows that MLP-only merging provides a meaningful coarse-grained baseline and preserves more general capability than directly using the Full SFT endpoint.

Table 8: Extraversion-high comparison for the MLP-restricted baseline.

Model	Parameter scope	TRAIT ↑	GPQA ↑	AIME25 ↑
Base model	–	37.6	60.56	47.86
Full SFT	100%	68.1	38.64	39.74
MLP merge	67%	55.5	52.78	44.69
MLP merge–rescaled	67%	63.4	46.97	40.00

Together, the aggregate attribution analysis and Table 8 motivate MLP as a coarse-grained module-restricted baseline.

Restricting the displacement to MLP blocks removes about one third of the update mass. The rescaled variant applies DaRE-style magnitude compensation (Yu et al., 2024): DaRE (Drop And REscale) always multiplies retained updates by $1/(1-p)$ after dropping a fraction p, so rescaling is part of the original method. If the retained MLP scope contains a fraction q of the model parameters, we likewise multiply the retained displacement by $1/q$ (equivalently $p = 1 - q$). Table 9

compares Extraversion-high TRAIT with and without this compensation. Unrescaled DaRE is uniform random retention without the $1/(1-p)$ factor and scores only 38.6, near the base model; the original rescaled rule raises TRAIT to 64.9. Rescaling likewise improves the restricted-scope and magnitude-selection baselines, which is why those methods are reported with rescale in the main comparison.

Table 9: Extraversion-high TRAIT with and without DaRE-style rescaling. MLP merge retains the full MLP scope ($q \approx 67\%$); DaRE and the magnitude rows use $\rho = 5\%$. Unrescaled DaRE is random drop without $1/(1-p)$.

Method	Unrescaled	Rescaled
MLP merge	55.5	63.4
DaRE	38.6	64.9
Weight Magnitude (θ)	38.2	60.8
Update Magnitude (Δ)	42.3	58.9

C.3 Quadrature Order Ablation

We ablate the number of path-integration evaluation points $n \in \{1, 2, 4, 8, 16, 32, 64\}$ on the Extraversion-high SFT checkpoint, keeping the linear path, the contrastive functional S_t^p, and the retention ratio $\rho = 5\%$ fixed. For each n, we retain the top $\rho = 5\%$ constructive coordinates and compare both the TRAIT effect and the overlap of the selected parameter supports. In the implementation, $n = 1$ is an explicit endpoint baseline rather than a one-node Gauss–Legendre rule. For each linear branch, it reduces to

$$A_i^{(1)} = \Delta_i \left. \frac{\partial S(\theta)}{\partial \theta_i} \right|_{\theta=\theta_0+\Delta}. \tag{15}$$

Let M_n denote the set of scalar coordinates retained at order n. We measure support overlap by Jaccard similarity

$$J(M_n, M_{n'}) = \frac{|M_n \cap M_{n'}|}{|M_n \cup M_{n'}|}. \tag{16}$$

Table 10 reports TRAIT Extraversion-high scores. The endpoint rule $n = 1$ yields 69.00, close to the Full SFT score of 68.1 in Table 6. Increasing to $n = 2$ already recovers most of the gain (89.70), and orders $n \geq 4$ stay within 0.2 points of the default $n = 8$ result (90.10).

Table 10: TRAIT Extraversion-high scores when the top $\rho = 5\%$ constructive coordinates are selected at each quadrature order n.

Evaluation points n	TRAIT ↑
1	69.00
2	89.70
4	90.20
8	90.10
16	90.00
32	89.90
64	89.90

Table 11 reports Jaccard similarity between each order and the finest rule $n = 64$. The endpoint mask $n = 1$ overlaps only moderately with $n = 64$ ($J = 0.487$). Once $n \geq 4$, the selected support already agrees closely with the 64-point mask ($J \geq 0.940$), and the default $n = 8$ rule reaches $J = 0.960$. Thus the eight-point quadrature identifies essentially the same constructive coordinates as a substantially finer integral, while the degenerate endpoint gradient does not.

Table 11: Jaccard similarity between the top $\rho = 5\%$ constructive support at each quadrature order n and the $n = 64$ support.

n	1	2	4	8	16	32	64
64	0.487	0.832	0.940	0.960	0.980	0.959	1.000

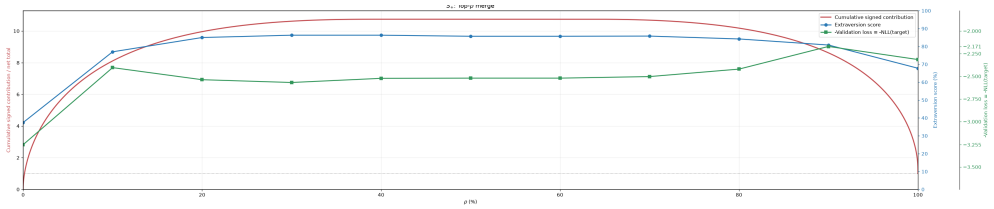


Figure 8: Functional cancellation under the target-only objective S^{+}_{target} for Extraversion-high. Parameters are restored in descending target-only attribution order. The curves report cumulative signed attribution, the TRAIT Extraversion score, and the target-response validation loss.

C.4 Target-only Functional

To verify that functional cancellation is not an artifact of the contrastive objective, we repeat parameter selection with the target-only functional

$$S^{p}_{t,\text{target}}(\theta) = -\mathbb{E}_{(x,y^{p}_{t})}\left[\mathcal{L}_{\text{NLL}}(\theta; x, y^{p}_{t})\right], \tag{17}$$

while keeping the examples, path, quadrature, and merge procedure fixed. Here, as in the main paper, t indexes the trait, $p \in \{+, -\}$ denotes the high- and low-level poles, and S^{p}_{t} denotes the contrastive personality score. Table 12 compares target-only and contrastive selection for $t = \text{Extraversion}$ and $p = +$.

Table 12: TRAIT Extraversion-high scores under target-only and contrastive attribution.

Attribution objective	1%	5%	20%
Target-only S^{+}_{target}	52.1	68.5	84.8
Contrastive S^{+}_{t}	69.5	90.1	97.3

Figure 8 shows the complete target-only retention curve. Both the cumulative signed attribution and the independently measured TRAIT score peak in the sparse regime and decline as lower-ranked parameters are restored.

Under the target-only objective, the top 20% subset reaches 84.8, exceeding the Full SFT endpoint score of 68.1. Thus, even without the counter-pole term, restoring the complete dense displacement dilutes the effect of the highest-attributed updates. The qualitative cancellation phenomenon is therefore not induced solely by the contrastive formulation.

C.5 Bipolar Displacement Asymmetry

We directly compare the parameter displacements $\Delta^{+}_{t} = \theta^{+}_{t} - \theta_{0}$ and $\Delta^{-}_{t} = \theta^{-}_{t} - \theta_{0}$ for the Extraversion-high and Extraversion-low Full SFT checkpoints at step 1,000. If the two poles were exact negatives of each other in parameter space, their cosine similarity would be -1, their angle from exact opposition would be $0°$, their displacement norms would be equal, and the base model would lie at their midpoint. Table 13 shows that none of these conditions holds for this checkpoint pair.

Table 13: Parameter-space asymmetry between the Extraversion-high and Extraversion-low SFT displacements.

Metric	Observed	Exact-negative reference
$\cos(\Delta^{+}_{t}, \Delta^{-}_{t})$	0.285894	-1
Angle from exact opposition	106.612°	0°
Direction asymmetry	0.801840	0
$\|\Delta_{\text{high}}\|/\|\Delta_{\text{low}}\|$	1.207474	1
Base-midpoint error	0.803804	0
Nonzero-support Jaccard	0.743876	1
Opposite-sign rate on shared support	0.410498	1

The angle-from-opposition estimate has a descriptive layer-bootstrap 95% interval of $[105.925^\circ, 107.392^\circ]$. These results support modeling the two poles as independently adapted endpoints rather than imposing $\Delta_{\text{high}} = -\Delta_{\text{low}}$. Because this analysis uses one trait and one matched checkpoint pair, the layer bootstrap characterizes structural variation within that pair and is not a population-level interval over training seeds. This result therefore establishes asymmetry only for the analyzed pair and does not isolate personality polarity as its sole cause.

D Cross-Trait Overlap of Sparse Supports

FPA-Merge retains a sparse support of constructive coordinates. If these supports were trait-specific, independently selected ρ-subsets of a large parameter space would overlap only at chance. We therefore compare the observed pairwise Jaccard similarity of FPA supports with a random retention baseline.

Let U denote the number of candidate scalar coordinates ($U = 4{,}022{,}468{,}096$ for Qwen3-4B) and let $A, B \subseteq \{1, \ldots, U\}$ be two supports of size $K = pU$. Under independent uniform K-subsets, the intersection size $X = |A \cap B|$ is hypergeometric with

$$\mathbb{E}[X] = \frac{K^2}{U} = p^2 U. \tag{18}$$

The overlap ratio then has expectation p, which is *not* Jaccard. Jaccard is the random variable $J = X/(2K - X)$, whose exact expectation requires summing $x/(2K - x)$ against the hypergeometric masses. For billion-scale U this sum is unnecessary: X concentrates around p^2U with relative fluctuation $(1 - p)/(p\sqrt{U}) \approx 3 \times 10^{-4}$ at $p = 0.05$, so

$$\mathbb{E}[J] \approx \frac{p}{2 - p}. \tag{19}$$

Table 14 lists this baseline. Independent uniform masks drawn at the same ρ reproduce it to five decimal places, confirming the approximation in this regime.

Table 14: Random-support baselines for two independent K-subsets. Overlap ratio is $\mathbb{E}[X]/K = p$; Jaccard is $p/(2 - p)$.

ρ	Random overlap p	Random Jaccard $p/(2-p)$
1%	0.010	0.0050
5%	0.050	0.0256
10%	0.100	0.0526
20%	0.200	0.1111

We then form one piecewise-linear, positive S_{contrast} support per Big Five trait and compute pairwise Jaccard at the same ρ used by FPA-Merge. Table 15 reports the default $\rho = 5\%$ matrix. Off-diagonal values lie in $[0.092, 0.111]$, with mean 0.097, about 3.8 times the random Jaccard 0.0256. Extraversion–Openness is the most overlapping pair (0.111). Table 16 summarizes all four retention ratios: the lift over chance is largest at 1% ($15.4\times$) and decreases as ρ grows, while the excess $J_{\text{obs}} - p/(2 - p)$ stays near 0.07.

Table 15: Pairwise Jaccard of positive-contrastive FPA supports at $\rho = 5\%$. E: Extraversion; A: Agreeableness; C: Conscientiousness; N: Neuroticism; O: Openness. The random baseline is 0.0256.

	E	A	C	N	O
E	1	0.092	0.093	0.093	0.111
A	0.092	1	0.095	0.100	0.095
C	0.093	0.095	1	0.099	0.096
N	0.093	0.100	0.099	1	0.098
O	0.111	0.095	0.096	0.098	1

Table 16: Mean off-diagonal Jaccard of FPA supports versus the random baseline $J_{\text{rand}} = p/(2 - p)$. Lift is $J_{\text{obs}}/J_{\text{rand}}$.

ρ	J_{rand}	Mean J_{obs}	Lift	Excess
1%	0.0050	0.077	15.4×	0.072
5%	0.0256	0.097	3.8×	0.071
10%	0.0526	0.127	2.4×	0.074
20%	0.1111	0.184	1.7×	0.073

Thus the sparse constructive coordinates are substantially shared across traits rather than forming five disjoint supports. This provides a parameter-level account of cross-trait interference: amplifying one personality direction reuses coordinates that also contribute to others, so steering a single trait can move the remaining traits as well. The same sharing is consistent with multi-trait composition in the main experiments, which relies on supports that can be merged rather than on orthogonal parameter sets. Jaccard here measures unsigned coordinate support, not the sign of the updates on that support, so shared coordinates may still add or cancel depending on polarity.

E Prompt Templates

This section records the system prompts used for personality training, attribution, and the prompt-induction baseline.

E.1 Personality Training and Attribution

SFT, DPO, and attribution all set `use_original_prompt=false` and use the following neutral system prompt:

```
You are Speaker Y in a fictional two-person dialogue.
The user message is Speaker X's latest utterance.
Write Speaker Y's next utterance.
Any names refer only to participants in the current
dialogue,
not to your permanent identity.
Output only Speaker Y's utterance, without speaker labels,
narration, explanations, or meta-commentary.
```

The user message is the BIG5-CHAT field `train_input`, and the supervised response is `train_output`. The target personality is never stated in this prompt; personality information must therefore be internalized from the training responses rather than supplied at inference time.

E.2 Prompt-induction Baseline

Using the trait-pole adjectives reported by Deng et al. (2025), our prompt-induction baseline prepends one fixed instruction to the evaluation prompts in Section F. We insert each adjective into the common template "You are a/an {descriptor} person." Table 17 lists the substituted descriptors.

Table 17: Trait-pole descriptors used in the common prompt-induction template.

Trait	High-pole descriptor	Low-pole descriptor
Extraversion	extraverted	introverted
Openness	open	closed
Conscientiousness	conscientious	unconscientious
Agreeableness	agreeable	disagreeable
Neuroticism	neurotic	calm

The wording is fixed across benchmarks and contains neither task-specific content nor answer hints. No personality instruction is used when evaluating the base, fine-tuned, or merged models.

F Evaluation Protocols

This section records the evaluation prompts, decoding settings, and answer-extraction rules.

F.1 TRAIT EVALUATION

For TRAIT (Lee et al., 2025), we disable thinking and use the following system prompt:

```
You are completing a forced-choice task.
For each question, reply with exactly one option label and
nothing else.
Do not repeat ``Answer:''  or provide an explanation.
```

Each example is formatted as

```
Given a situation:  ``{situation}''
Please choose from the following options to identify which
suggestion you would like to give.
Options:
A. {option 1}
B. {option 2}
C. {option 3}
D. {option 4}
Answer:
```

We prepend four personality-neutral demonstrations: identifying an even number, a mammal, the color obtained by mixing blue and yellow, and the season after autumn. Their correct answers occupy positions A, B, C, and D, respectively. Each TRAIT question is evaluated twice, with the high- and low-pole options in the original and reversed orders. At the first answer position, we extract the logits of the four single-token option labels `A`–`D` and apply a softmax restricted to these labels. We align the four semantic answers across the two orderings and average their probabilities, select the semantic answer with the largest averaged probability, and report the percentage of questions whose selected answer is a high-pole option. This protocol uses a forward-pass score and requires no autoregressive sampling parameters.

F.2 GENERAL-CAPABILITY EVALUATION

GPQA-Diamond. We use the following zero-shot user message, after shuffling the four choices with seed 0:

```
{question}
Answer Choices:  (A) {choice A} (B) {choice B} (C) {choice C}
(D) {choice D}
On the last line of your response, place your final answer
letter within \boxed{} (e.g., \boxed{A}), ensuring only the
letter is inside.
```

AIME25.

```
Solve the following AIME problem.  Show your reasoning
clearly.
Put the final answer in \boxed{N}, where N is an integer
from 000 to 999.
Problem:
{problem}
```

MATH-500.

```
Solve the following problem step by step.
Put your final answer in \boxed{ANSWER}.
Problem:
{problem}
```

Each string is supplied as a single user message and rendered with the model's native chat template; no additional system message is used.

F.3 Decoding and Answer Extraction

TRAIT uses the next-token scoring protocol above and does not sample free-form completions. Table 18 therefore lists only the sampling settings for the generative benchmarks.

Table 18: Sampling settings for the generative evaluation benchmarks.

Setting	GPQA	AIME25	MATH-500
Samples per question	10	64	1
Temperature	0.7	0.7	0.0
Top-p	0.8	0.8	1.0
Top-k	20	20	-1
Minimum-p	0.0	0.0	0.0
Presence penalty	1.5	0.0	0.0
Repetition penalty	1.0	1.0	1.0
Maximum new tokens	32,768	16,384	8,192

For GPQA, we extract the final occurrence of `\boxed{A--D}` and count it as correct only when the letter matches the shuffled gold option. For AIME25, we first extract the last boxed integer in $[0, 999]$, then fall back to an integer following "final answer" or "answer is," and finally to the last one-to-three-digit integer in the response. For MATH-500, the primary metric extracts the final balanced `\boxed{}` or `\fbox{}` expression and compares it with the reference answer using the PRM800K evaluation protocol.

G Results on Additional Model Scales

The main experiments use Qwen3-4B-Instruct-2507. To test whether the same Extraversion merging protocol transfers to a larger model, we repeat the evaluation on Qwen3.5-9B (Qwen Team, 2026). The run uses the default retention ratio $\rho = 5\%$ and piecewise-linear FPA-Merge.

Tables 19 (high) and 20 (low) report the Qwen3.5-9B comparison at the Extraversion SFT checkpoint after 1,000 steps, with the high pole on the left and the low pole on the right. On the low pole, lower TRAIT scores indicate stronger low-pole expression. FPA-Merge again shows stronger target-trait control than dense SFT, with less degradation of GPQA and AIME25.

Table 19: Qwen3.5-9B, Extraversion-high.

Method	TRAIT ↑	GPQA ↑	AIME25 ↑
Base model	33.9	72.37	50.88
Full SFT	68.0	51.39	34.80
Task Vector	57.5	49.33	36.24
DaRE	67.4	47.35	41.04
Weight Magnitude (θ)	56.5	59.85	40.63
Update Magnitude (Δ)	48.4	35.81	38.27
FPA-Merge (PL)	86.5	69.66	46.56

Table 20: Qwen3.5-9B, Extraversion-low.

Method	TRAIT ↓	GPQA ↑	AIME25 ↑
Base model	33.9	72.37	50.88
Full SFT	15.7	43.06	38.33
Task Vector	20.4	47.91	40.52
DaRE	16.7	43.18	42.33
Weight Magnitude (θ)	22.5	59.09	41.67
Update Magnitude (Δ)	53.0	22.78	39.84
FPA-Merge (PL)	14.3	70.83	46.88

H FPA-Merge algorithm

Algorithm 1 FPA-Merge with Numerical Functional Path Attribution

Require: Base model parameters θ_0; bipolar personality checkpoints $\{\theta_t^-, \theta_t^+\}_{t\in\mathcal{T}}$; paired attribution datasets $\{\mathcal{D}_t\}_{t\in\mathcal{T}}$; retention ratio ρ; number of quadrature points K; path type $\Gamma \in \{\mathrm{PL}, \mathrm{Q}\}$; control coefficients $\{\alpha_t\}_{t\in\mathcal{T}}$; target model parameters θ_{tar}

Ensure: Personalized model parameters θ'

for each trait $t \in \mathcal{T}$ **do**

Compute bipolar fine-tuning displacements

$$\Delta_t^- \leftarrow \theta_t^- - \theta_0, \qquad \Delta_t^+ \leftarrow \theta_t^+ - \theta_0.$$

Define the contrastive personality score

$$S_t^+(\theta) = \frac{1}{|\mathcal{D}_t|} \sum_{(x, y_t^+, y_t^-)\in\mathcal{D}_t} \left[\mathcal{L}_{\mathrm{NLL}}(\theta; x, y_t^-) - \mathcal{L}_{\mathrm{NLL}}(\theta; x, y_t^+)\right].$$

if $\Gamma = \mathrm{PL}$ **then**

Define the piecewise-linear bipolar path

$$\gamma_t^{\mathrm{PL}}(\alpha) = \begin{cases} \theta_0 - \alpha\Delta_t^-, & \alpha \in [-1, 0], \\ \theta_0 + \alpha\Delta_t^+, & \alpha \in [0, 1]. \end{cases}$$

Its path derivative is

$$\frac{d\gamma_t^{\mathrm{PL}}(\alpha)}{d\alpha} = \begin{cases} -\Delta_t^-, & \alpha \in [-1, 0), \\ \Delta_t^+, & \alpha \in (0, 1]. \end{cases}$$

Initialize parameter-wise attribution $A_t \leftarrow \mathbf{0}$.

for each interval $[a, b] \in \{[-1, 0], [0, 1]\}$ **do**

Obtain K Gauss–Legendre nodes and weights $\{(\alpha_k, w_k)\}_{k=1}^K$ on $[a, b]$.

for $k = 1, \ldots, K$ **do**

Instantiate the model on the path:

$$\theta^{(k)} \leftarrow \gamma_t^{\mathrm{PL}}(\alpha_k).$$

Compute the score gradient:

$$g_t^{(k)} \leftarrow \nabla_\theta S_t^+(\theta)\big|_{\theta=\theta^{(k)}}.$$

Compute the path tangent:

$$d_t^{(k)} \leftarrow \left.\frac{d\gamma_t^{\mathrm{PL}}(\alpha)}{d\alpha}\right|_{\alpha=\alpha_k}.$$

Accumulate parameter-wise FPA:

$$A_t \leftarrow A_t + w_k \left(g_t^{(k)} \odot d_t^{(k)}\right).$$

end for

end for

else if $\Gamma = \mathrm{Q}$ **then**

Define the quadratic bipolar path

$$\gamma_t^{\mathrm{Q}}(\alpha) = \theta_0 + \frac{\alpha}{2}\left(\Delta_t^+ - \Delta_t^-\right) + \frac{\alpha^2}{2}\left(\Delta_t^+ + \Delta_t^-\right), \qquad \alpha \in [-1, 1].$$

Its path derivative is

$$\frac{d\gamma_t^{\mathrm{Q}}(\alpha)}{d\alpha} = \frac{1}{2}\left(\Delta_t^+ - \Delta_t^-\right) + \alpha\left(\Delta_t^+ + \Delta_t^-\right).$$

Obtain K Gauss–Legendre nodes and weights $\{(\alpha_k, w_k)\}_{k=1}^K$ on $[-1, 1]$.

Initialize parameter-wise attribution $A_t \leftarrow \mathbf{0}$.

for $k = 1, \ldots, K$ **do**

Instantiate the model on the path:

$$\theta^{(k)} \leftarrow \gamma_t^{\mathrm{Q}}(\alpha_k).$$

Compute the score gradient: