# EXPERTGUARD: TRANSCENDING THE STABILITY-PLASTICITY DILEMMA IN LLM RL VIA EXPERT-VECTOR GUARDED POLICY UPDATES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Aligning large language models (LLMs) via reinforcement learning (RL) in multi-task environments is governed by the fundamental stability-plasticity dilemma: preserving historical performance while efficiently acquiring novel capabilities. Although RL training offers greater implicit regularization than supervised fine-tuning, it fails to fundamentally resolve cross-task gradient conflicts. Existing solutions, such as dynamic gradient tracking or static weight merging, either incur prohibitive computational overhead or rely on rigid weight-space heuristics detached from dynamic optimization landscapes. To break this trade-off, we propose **ExpertGuard**, a framework that seamlessly incorporates static expert vectors as non-trainable geometric boundaries directly into the RL optimization loop. Motivated by a first-order geometric analysis, we identify that dynamic stability loss is primarily associated with gradient update components that oppose historical semantic trajectories. ExpertGuard eliminates these conflicting components by executing a subspace residual gradient projection protocol that projects raw policy updates onto the unique intersection of orthogonalized expert null-spaces, thereby neutralizing negative transfer while minimizing perturbation to target learning signals. Formal proofs and extensive experiments on various domains demonstrate that ExpertGuard achieves superior performance with negligible computational and memory overhead, effectively transcending the stability-plasticity dilemma.

## 1 INTRODUCTION

Reinforcement Learning (RL) has emerged as a pivotal paradigm for large language models (LLMs) post-training, yet its optimization landscape in multi-task scenarios remains constrained by the classical stability-plasticity dilemma (Robins, 1995). Specifically, stability requires the parameter space to safeguard historical capabilities by preserving previously acquired semantic knowledge and safety alignment boundaries. Conversely, plasticity governs adaptation and generalization efficiency on new tasks, requiring the models to maintain high sensitivity toward new reward signals.

Prior approaches attempting to balance this trade-off fall into two primary categories: training-free and training-based paradigms. The training-free paradigm, typified by conventional model merging strategies (Yang et al., 2026a; Ilharco et al., 2023), provides an inherently low-overhead solution by interpolating parameter weights without accessing training data. However, its performance is fundamentally bounded by the invalid assumption of linear weight-space composability and the geometric mismatch of sharp minima (Frankle et al., 2020), which often precipitates severe generalization collapse during static arithmetic merging (Yadav et al., 2023). Conversely, training-based methods actively mitigate these structural limitations by manipulating loss scales (Chen et al., 2018), searching for Pareto-optimal frontiers (Sener & Koltun, 2018; Liu et al., 2021), or executing geometric gradient surgery (Saha et al., 2021; Sun et al., 2026). As summarized in Table 1, existing training-based methods suffer from a rigid trade-off: preserving stability strictly degrades target-task plasticity or introduces prohibitive computational overhead (e.g., auxiliary backward propagations or high-dimensional matrix decompositions). This suboptimal trade-off motivates our research question:

***How can we break the stability-plasticity dilemma in an efficient and low-overhead manner in LLM RL scenarios?***



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.

Table 1: Taxonomy and comparison of multi-task optimization strategies. We compare ExpertGuard against existing paradigms across key performance and efficiency dimensions.

| Dimension | Conflict Resolution | Stability | Plasticity | Overhead |
|---|---|---|---|---|
| GradNorm | Dynamic loss weight scaling | Poor | **Excellent** | **Low** |
| Pareto Frontier | Multi-objective gradient synergy | **Excellent** | Poor | High |
| PCGrad | Pairwise gradient surgery | Moderate | **Excellent** | High |
| Global Subspace | Activation subspace mapping | **Excellent** | Poor | High |
| **ExpertGuard (Ours)** | **Subspace residual projection** | **Excellent** | **Excellent** | **Low** |

To address this problem, we leverage a critical insight derived from model merge: the *expert vector*—defined as the parameter displacement between a fine-tuned expert model and its base initialization—serves as an explicit representation of domain-specific knowledge directions. To overcome the capacity bounds of training-free merging without incurring the redundant forward-backward overheads of gradient tracking during on-policy RL, we employ static expert vectors to explicitly anchor historical capabilities. Crucially, through formal theoretical analysis grounded in first-order approximations, we establish a mathematical linkage between static expert vectors and the dynamic online gradients of prior domains, demonstrating that stability is conditionally preserved whenever the target-task gradient forms a non-obtuse angle with historical expert directions under Assumption 1. Leveraging this connection, we treat the expert vector as a non-trainable optimization boundary during new-task fine-tuning and propose **ExpertGuard**, an algorithmic framework centered on a novel subspace residual gradient surgery mechanism. Main contributions of this paper are threefold:

- **Lightweight geometric supervision**: We seamlessly incorporate static expert vectors into the RL optimization loop as non-trainable geometric boundaries, eliminating the need for auxiliary backward propagations while maintaining negligible computational overhead.
- **Conditionally guaranteed stability-plasticity trade-off**: Grounded in a first-order geometric analysis and local directional alignment assumptions, we formulate a joint subspace residual projection algorithm and formally prove its ability of balancing stability and plasticity.
- **Superior empirical scaling**: Extensive evaluations across diverse domains demonstrate that ExpertGuard achieves state-of-the-art stability-plasticity trade-offs with negligible computational overhead, offering a highly practical solution for real-world LLM alignment.

## 2 Background and Preliminaries

**The stability-plasticity dilemma**. The core objective in addressing the stability-plasticity dilemma is to balance the preservation of historical capabilities (stability) against sensitivity to novel reward signals (plasticity) (Robins, 1995). Existing training-based optimizations generally manage this trade-off by manipulating loss scales or gradient directions. However, as synthesized in Table 1, these approaches exhibit an unyielding trade-off: loss-scaling methods like GradNorm (Chen et al., 2018) preserve plasticity with low computational overhead but fail to guarantee stability against gradient conflicts; conversely, Pareto frontier optimization (Sener & Koltun, 2018; Liu et al., 2021) and global subspace projections (Saha et al., 2021) achieve robust stability at the cost of domain plasticity while imposing computational and memory burdens, causing an unsatisfactory trade-off.

**Expert vector as a static lightweight supervision**. Model merging offers a low-overhead approach to balancing stability and plasticity by directly interpolating weights without training data (Ilharco et al., 2023; Yang et al., 2026a). The foundation of this paradigm is the expert vector, which serves as a compact representation of semantic directions (Turner et al., 2023; Yadav et al., 2023; Jin et al., 2023). Beyond parameter editing, expert vectors are widely recognized as encoding distinct semantic trajectories (Ainsworth et al., 2022; Zheng et al., 2025). Inspired by the semantic interpretability of expert vectors, yet seeking to overcome the capacity limits of model merging (Frankle et al., 2020; Yadav et al., 2023), we repurpose these static expert vectors as non-trainable geometric supervision boundaries to anchor the online trajectories of historical domains during RL.

**Policy optimization setup**. Let $\pi_\theta$ denote the policy and $\mathcal{D}$ denote the dataset. Training maximizes an on-policy objective $\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D}_{\text{new}},o\sim\pi_\theta}[\mathcal{R}(q,o)]$, where $\mathcal{R}$ denotes the scalar reward signal.

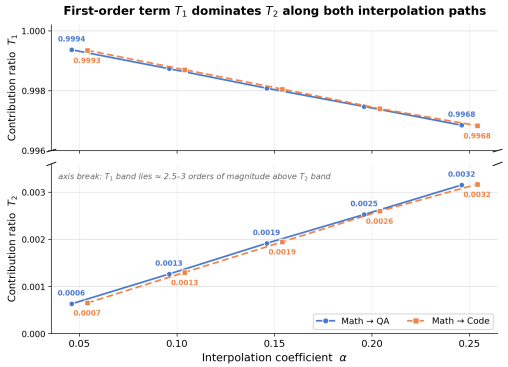



Figure 1: Empirical dominance of the first-order term along interpolation paths. Contribution ratios $R_1 = |T_1|/(|T_1| + |T_2|)$ and $R_2 = 1 - R_1$ evaluated along the linear parameter interpolation path between expert models. While distinct from the non-linear RL trajectory, the first-order gradient conflict term $T_1$ consistently accounts for over 99.6% of total variation, motivating our use of first-order directional alignment.

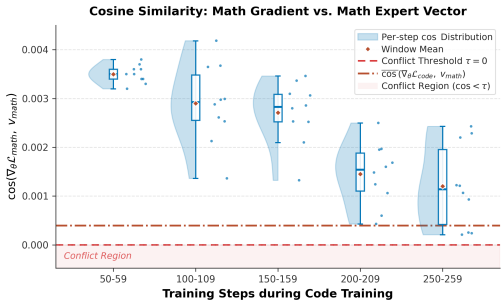



Figure 2: Empirical adherence to Assumption 1. Time-series distribution of the cosine similarity between the online historical gradient $\nabla_\theta J_{\text{math}}$ and the static expert vector $\mathbf{v}_{\text{math}}$ during Code RL. Despite inherent microscopic fluctuations caused by on-policy sampling variance, the macroscopic alignment consistently concentrates in the positive regime ($\cos > 0$), empirically validating that the optimization trajectory operates reliably within the directional validity regime governed by Assumption 1.

Throughout this work, we instantiate policy optimization using GRPO (Shao et al., 2024) as our base algorithmic backbone, while our method readily generalizes to arbitrary policy update algorithms.

## 3 First-Order Geometric Analysis of Stability Loss

At optimization step $t$, let $\mathbf{g}_{\text{new}}^{(t)} \triangleq \nabla_\theta \mathcal{J}_{\text{new}}(\theta^{(t)})$ denote the accumulated raw gradient. The corrected policy update $\mathbf{g}'^{(t)}$ is directly applied to the parameters: $\theta^{(t+1)} = \theta^{(t)} + \eta_t \mathbf{g}'^{(t)}$, where $\eta_t > 0$ is the learning rate. For any historical domain $k$, we denote its online gradient as $\mathbf{h}_k^{(t)} \triangleq \nabla_\theta J_k(\theta^{(t)})$.

### 3.1 Gradient Conflict and Curvature Sensitivity

Assuming local second-order smoothness of $\mathcal{J}_k$, the objective variation induced by $\mathbf{g}'^{(t)}$ is defined as $\Delta\mathcal{J}_k^{(t)} \triangleq \mathcal{J}_k(\theta^{(t)} + \eta_t \mathbf{g}'^{(t)}) - \mathcal{J}_k(\theta^{(t)})$, which admits the second-order Taylor expansion:

$$\Delta\mathcal{J}_k^{(t)} = \underbrace{\eta_t \left\langle \mathbf{h}_k^{(t)}, \mathbf{g}'^{(t)} \right\rangle}_{T_1} + \underbrace{\frac{\eta_t^2}{2} (\mathbf{g}'^{(t)})^\top \nabla_\theta^2 \mathcal{J}_k(\theta^{(t)}) \mathbf{g}'^{(t)}}_{T_2} + \mathcal{O}\left(\eta_t^3 \|\mathbf{g}'^{(t)}\|^3\right). \tag{1}$$

Prior work attributes stability loss either to obtuse angles between new-task and historical gradients (Yu et al., 2020; Sun et al., 2026) or to curvature effects near stationary points (Yang et al., 2026b). To characterize their relative contribution, we evaluate the first-order ($T_1$) and second-order ($T_2$) variations along linear interpolation paths between expert models (Figure 1). While this linear path fundamentally differs from the true non-linear RL trajectory governed by adaptive optimizers and KL penalties, $T_1$ consistently accounts for over 99.6% of total variation. Rather than a strict global bound, this empirical dominance serves as an operational motivation to adopt first-order directional alignment as a computationally tractable heuristic proxy for stability preservation, grounding the analysis below.

### 3.2 Expert Vectors as Static Directional Proxies

For each historical domain $k$, define the expert vector $\mathbf{v}_k \triangleq \theta_k^* - \theta_{\text{base}}$ as the displacement from the base model to the corresponding domain expert. Its definition does not imply that $\mathbf{v}_k$ is equal or collinear with $\mathbf{h}_k^{(t)}$. We instead use it as a static directional proxy under the following assumption.

**Assumption 1** (Local Historical Directional Alignment)**.** *For historical domain $k$ at step $t$, the historical gradient maintains a non-negative directional alignment with the domain expert vector:*

$$\left\langle \mathbf{h}_k^{(t)}, \mathbf{v}_k \right\rangle \geq 0. \tag{2}$$

**Remark 1.** *We treat Assumption 1 as a local operational condition rather than a global invariant over the entire non-convex parameter manifold. In post-training RL algorithms such as GRPO, the policy update is explicitly regularized by a KL divergence penalty term $\mathcal{L}_{\mathrm{KL}} = \beta \cdot D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$. This penalty restricts the parameter trajectory $\theta^{(t)}$ to a localized trust region around the initialization $\theta_{\mathrm{base}}$. Since the expert vector $\mathbf{v}_k = \theta_k^* - \theta_{\mathrm{base}}$ represents the primary semantic displacement from this initialization, the bounded deviation enforced by the KL constraint naturally prevents the online gradient $\mathbf{h}_k^{(t)}$ from rotating beyond the orthogonal hyperplane of $\mathbf{v}_k$, thereby preserving the directional alignment required by Assumption 1.*

To empirically verify the validity regime of Assumption 1 in practical RL, we track the cosine similarity $\langle \mathbf{h}_k^{(t)}, \mathbf{v}_k \rangle / (\|\mathbf{h}_k^{(t)}\| \|\mathbf{v}_k\|)$ throughout training. As illustrated in Figure 2, while high-dimensional stochasticity introduces minor step-wise fluctuations, the time-averaged alignment robustly concentrates in the positive domain. This confirms that the macroscopic on-policy trajectory reliably respects the geometric conditions of Assumption 1 in practice, justifying its use as a foundational premise for our first-order analysis. Furthermore, as detailed in Appendix C.1.2, even if transient violations occur due to stochastic batch sampling, ExpertGuard's gating mechanism ensures that projecting to the null-space serves as a safe fallback, preventing catastrophic degradation.

### 3.3 Single-Expert First-Order Conflict Removal

We quantify the directional conflict between the target gradient and expert $k$ by the projection coefficient $\alpha_k^{(t)} \triangleq \langle \mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{v}_k \rangle / \|\mathbf{v}_k\|^2$. Expert $k$ incurs an active conflict whenever it exhibits negative transfer ($\alpha_k^{(t)} < 0$). For any conflicting expert, we strip this interfering component via:

$$\mathbf{g}_k'^{(t)} = \mathbf{g}_{\mathrm{new}}^{(t)} - \alpha_k^{(t)} \mathbf{v}_k = \mathbf{g}_{\mathrm{new}}^{(t)} - \frac{\langle \mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{v}_k \rangle}{\|\mathbf{v}_k\|^2} \mathbf{v}_k, \quad \text{satisfying} \quad \langle \mathbf{g}_k'^{(t)}, \mathbf{v}_k \rangle = 0. \tag{3}$$

**Theorem 1** (Single-Expert First-Order Non-Degradation and Improvement Bound)**.** *At step $t$, suppose Assumption 1 holds with a local directional alignment margin $\rho_k^{(t)} \triangleq \cos(\mathbf{h}_k^{(t)}, \mathbf{v}_k) > 0$. If an active conflict is detected along the expert proxy ($\alpha_k^{(t)} < 0$), $\mathbf{g}_k'^{(t)}$ satisfies the algebraic identity:*

$$\left\langle \mathbf{g}_k'^{(t)}, \mathbf{h}_k^{(t)} \right\rangle - \left\langle \mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{h}_k^{(t)} \right\rangle = \rho_k^{(t)} \left| \cos\left( \mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{v}_k \right) \right| \left\| \mathbf{g}_{\mathrm{new}}^{(t)} \right\| \left\| \mathbf{h}_k^{(t)} \right\| \geq 0. \tag{4}$$

The proof is detailed in Appendix C.1.1. Beyond standard non-degradation, the exact algebraic identity established therein reveals that the directional improvement scales jointly with the gradient norms and is explicitly modulated by the proxy conflict severity $|\cos(\mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{v}_k)|$, providing an inherent adaptive mechanism without manual tuning (see Remark 3 for details). Importantly, components of $\mathbf{h}_k^{(t)}$ orthogonal to $\mathbf{v}_k$ remain unconstrained by Assumption 1.

## 4 ExpertGuard

This section introduces ExpertGuard, a subspace residual gradient projection protocol that projects raw policy updates onto the collective orthogonal complement in multi-domain scenarios. Crucially, we demonstrate that this protocol achieves maximal plasticity preservation via minimal distortion.

### 4.1 Subspace Residual Gradient Projection

Given $K$ expert vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$, we identify conflicting domains by computing projection coefficients $\alpha_k^{(t)} \triangleq \langle \mathbf{g}_{\mathrm{new}}^{(t)}, \mathbf{v}_k \rangle / \|\mathbf{v}_k\|^2$ and defining the active conflict set $\mathcal{I}_t \triangleq \{k : \alpha_k^{(t)} < \tau\}$. By default, we set the conflict threshold $\tau = 0$ to precisely activate intervention if and only if strict negative transfer occurs, which serves as the theoretical anchor for our formal guarantees;

however, in practical engineering deployments, $\tau$ can be customized as a flexible hyperparameter to proactively modulate the stability-plasticity trade-off. If $\mathcal{I}_t = \varnothing$, we retain $\mathbf{g}'^{(t)} = \mathbf{g}_{\text{new}}^{(t)}$. Otherwise, let $\mathcal{I}_t = \{\pi_1, \ldots, \pi_M\}$ denote the indices of the active conflicting experts, spanning the joint active expert subspace $\mathcal{S}_t \triangleq \text{span}\{\mathbf{v}_{\pi_1}, \ldots, \mathbf{v}_{\pi_M}\}$. To decouple mutual non-orthogonal dependencies among expert proxies, we construct an orthogonal basis for $\mathcal{S}_t$ via the Gram–Schmidt process (Golub & Van Loan, 2013), setting $\widetilde{\mathbf{v}}_{\pi_1} = \mathbf{v}_{\pi_1}$ and computing residuals recursively:

$$\widetilde{\mathbf{v}}_{\pi_m} = \mathbf{v}_{\pi_m} - \sum_{j=1}^{m-1} \frac{\langle \mathbf{v}_{\pi_m}, \widetilde{\mathbf{v}}_{\pi_j} \rangle}{\|\widetilde{\mathbf{v}}_{\pi_j}\|^2} \widetilde{\mathbf{v}}_{\pi_j}, \qquad m = 2, \ldots, M. \tag{5}$$

Numerically degenerate residuals are discarded, yielding an orthogonal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_r\}$ with $r = \dim(\mathcal{S}_t) \leq M$ . Initializing at $\mathbf{g}^{(0)} = \mathbf{g}_{\text{new}}^{(t)}$, ExpertGuard successively strips the conflicting projection components across the orthogonal basis:

$$\mathbf{g}^{(j)} = \mathbf{g}^{(j-1)} - \frac{\langle \mathbf{g}^{(j-1)}, \mathbf{q}_j \rangle}{\|\mathbf{q}_j\|^2} \mathbf{q}_j, \qquad j = 1, \ldots, r, \tag{6}$$

yielding the final corrected update $\mathbf{g}'^{(t)} = \mathbf{g}^{(r)}$, followed by the parameter step $\theta^{(t+1)} = \theta^{(t)} + \eta_t \mathbf{g}'^{(t)}$ . The complete procedure is outlined in Algorithm 1.

**Equivalence to Joint Subspace Projection.** Given that $\mathbf{q}_j$ are mutually orthogonal, the sequential operation in Eq. 6 admits a closed-form characterization as a global projection operator:

$$\mathbf{P}_{\mathcal{S}_t^\perp} \triangleq \mathbf{I} - \sum_{j=1}^{r} \frac{\mathbf{q}_j \mathbf{q}_j^\top}{\|\mathbf{q}_j\|^2}, \quad \mathbf{g}'^{(t)} = \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\text{new}}^{(t)}, \quad \text{satisfying} \quad \langle \mathbf{g}'^{(t)}, \mathbf{v}_k \rangle = 0, \quad \forall k \in \mathcal{I}_t. \tag{7}$$

The formal equivalence proof is provided in Appendix C.1.4. Eq. 7 constitutes the exact geometric signature of ExpertGuard: it purges all gradient components lying within the collective historical capability subspace $\mathcal{S}_t$, providing a definitive algebraic decoupling guarantee on the expert manifold.

### 4.2 Theoretical Guarantees: Resolving the Stability-Plasticity Dilemma

**Multi-Expert Stability via Joint Proxy Decoupling.** In single-expert environments, the non-degradation guarantee (Theorem 1) hinges on neutralizing the directional conflict along the isolated proxy. In the multi-task regime, however, applying uncoordinated pairwise projections is known to induce severe order sensitivity (Yu et al., 2020), where repairing one domain inadvertently re-introduces negative projections against previously sanitized capabilities. ExpertGuard circumvents this coupling by performing a collective projection onto $\mathcal{S}_t^\perp$, guaranteeing simultaneous first-order recovery without supplementary structural assumptions:

**Proposition 1** (Simultaneous Proxy Recovery)**.** *Let $\mathcal{I}_t$ denote the set of active conflicting experts, and let $\mathbf{g}'^{(t)} = \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\text{new}}^{(t)}$ be the update generated via the Gram–Schmidt residual projection in Eq. 7. For every conflicting domain $k \in \mathcal{I}_t$, the directional variation along its proxy vector satisfies:*

$$\left\langle \mathbf{g}'^{(t)}, \mathbf{v}_k \right\rangle - \left\langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k \right\rangle = \left| \left\langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k \right\rangle \right| > 0. \tag{8}$$

The proof is provided in Appendix C.1.6. Proposition 1 confirms that the directional improvements analogous to Theorem 1 are achieved across all conflicting historical domains simultaneously.

**Remark 2.** *Unlike pairwise sequential projection methods (e.g., PCGrad), which suffer from order sensitivity because repairing one domain may inadvertently reintroduce negative projections against previously sanitized non-orthogonal proxies, ExpertGuard circumvents this coupling fundamentally. By constructing an orthogonal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_r\}$ via Gram–Schmidt, the joint projector $P_{S_t^\perp}$ becomes uniquely defined and idempotent ($P_{S_t^\perp}^2 = P_{S_t^\perp}$). This guarantees that the nullification of the gradient component along any proxy $\mathbf{v}_k$ is mathematically decoupled from the others, establishing strict order-invariant stability on the surrogate expert manifold.*

**Maximal Plasticity via Minimal Distortion.** To maximize plasticity under standard Euclidean geometry, the update should minimize distortion relative to the unconstrained gradient $g_{\text{new}}^{(t)}$. By the Hilbert projection theorem, the unique optimal solution in the Euclidean sense is precisely the orthogonal projection:

$$g'^{(t)} = \arg\min_{u \in S_t^{\perp}} \frac{1}{2}\|u - g_{\text{new}}^{(t)}\|^2 = P_{S_t^{\perp}} g_{\text{new}}^{(t)}. \tag{9}$$

Consequently, Pythagorean orthogonality guarantees the exact energy retention identity:

$$\|g'^{(t)}\|^2 = \|g_{\text{new}}^{(t)}\|^2 - \|P_{S_t} g_{\text{new}}^{(t)}\|^2. \tag{10}$$

Eq. 9- 10 confirm that ExpertGuard introduces the minimal possible Euclidean perturbation to the target gradient: all learning signals linearly independent of $S_t$ are preserved without distortion in the parameter space. We note that while practical implementations employ adaptive optimizers (e.g., AdamW) which implicitly alter the optimization geometry, this Euclidean projection serves as a principled and computationally tractable first-order approximation for maximal plasticity preservation (Appendix C.1.7).

# 5 Experiments

## 5.1 Experimental Setup

**Implementation details**. We conduct all training procedures on 8×NVIDIA H20 GPUs using the ROLL framework (Wang et al., 2025). We employ GRPO with a group size of $G = 16$, a rollout batch size of 64, a training step of 300. Sampling parameters are configured with Top-$p = 0.95$, Top-$k = 50$, and a temperature of 0.8 during training (0.6 during validation). Qwen3-8B-Base (Yang et al., 2025) is the unified initialization. Detailed training settings are provided in Appendix D.

**Datasets**. We evaluate ExpertGuard across three domains: Math, Code and QA. We employ Numina Li et al. (2024), KodCode Xu et al. (2025) and CrossThinkQA Akter et al. (2026) as the datasets, respectively, each comprising 9K training instances and 1K test samples. Pass@1 is employed as the core metric and each test sample is evaluated 16 times.

**Baselines**. We evaluate ExpertGuard against diverse baselines: (1) **Single-Task Experts**, fine-tuned solely on target objectives; (2) **Agent-Dice** (Wu et al., 2026), a state-of-the-art post-training model merging baseline; and (3) **Refresh** (Yang et al., 2026b), a re-training strategy that incorporates 10% domain data from the guarded domain based on the expert model. While classic multi-task gradient surgery strategies (e.g., PCGrad, Pareto MTL) provide valuable theoretical taxonomy (Table 1), their direct application in on-policy LLM RL requires auxiliary backward propagations or pairwise gradient evaluations for every active domain at each step. For an 8B-parameter model trained with GRPO across $K$ domains, this incurs an $\mathcal{O}(K)$ multiplicative overhead in backward passes per optimization step, rendering them practically inapplicable under standard RL compute budgets. Therefore, to ensure a fair comparison under realistic deployment constraints, we benchmark ExpertGuard against state-of-the-art lightweight and baselines in the main text and provide additional results compared with PCGrad in Appendix D.2

## 5.2 Main Results

**Cross-domain effectiveness.** *ExpertGuard consistently outperforms baselines across all scenarios (Table 2), achieving the lowest degradation while maintaining optimal target-task plasticity.*

*(i) Stability on historical domains.* Single-task experts suffer catastrophic forgetting (e.g., Math Expert's Code accuracy drops below the base initialization). Baseline methods fail to reconcile this: Refresh incurs noticeable collapse despite mixing 10% historical data, while Agent-Dice provides suboptimal retention. In contrast, ExpertGuard systematically secures stability across all benchmarks, maintaining guarded-domain performance within a minimal gap to expert upper bounds. *(ii) Plasticity on target domains.* Conventional baselines suppress learning plasticity to enforce stability. ExpertGuard attains the highest mean accuracies and minimal average gaps across the overall trade-off. Notably, in certain regimes (e.g., Scenario A), ExpertGuard yields target-task performance slightly exceeding the unconstrained single-task expert baseline. Rather than being a primary design

Table 2: Cross-domain effectiveness. Under **Training Task**, $\Delta_{\text{Exp}}$ and $\Delta_{\text{Base}}$ denote performance gaps relative to the training expert ($A - A^{\text{train}}_{\text{train_expert}}$) and the base model ($A - A^{\text{train}}_{\text{base}}$). Under **ExpertGuard Task**, $\Delta_{\text{TrainExp}}$ evaluates the anti-forgetting gain over the unprotected training expert on the guarded domain ($A - A^{\text{guard}}_{\text{train_expert}}$), while $\Delta_{\text{GdExp}}$ reflects the preservation gap relative to the guarded-domain expert upper bound ($A - A^{\text{guard}}_{\text{guard_expert}}$). Under **Overall Trade-off**, Mean Acc. is the unweighted average accuracy, and Avg. Gap is the mean gap relative to the respective domain experts ($\frac{\Delta_{\text{Exp}}+\Delta_{\text{GdExp}}}{2}$). Bold values denote ours, while underlined values indicate the best results.

| Method | Training Task (Plasticity) | | | ExpertGuard Task (Stability) | | | Overall Trade-off | |
|---|---|---|---|---|---|---|---|---|
| | Acc | $\Delta_{\text{Exp}}$ | $\Delta_{\text{Base}}$ | Acc | $\Delta_{\text{TrainExp}}$ | $\Delta_{\text{GdExp}}$ | Mean Acc. | Avg. Gap |
| *Scenario A: Training Task = Code, ExpertGuard Task = Math* | | | | | | | | |
| Base Model | 36.19 | -21.50 | – | 38.97 | -5.00 | -13.24 | 37.58 | -17.37 |
| Code Expert | 57.69 | – | +21.50 | 43.97 | – | -8.24 | 50.83 | -4.12 |
| Math Expert | 35.22 | -22.47 | -0.97 | 52.21 | +8.24 | – | 43.72 | -11.24 |
| Agent-Dice | 46.06 | -11.63 | +9.87 | 49.25 | +5.28 | -2.96 | 47.66 | -7.30 |
| Refresh | 50.44 | -7.25 | +14.25 | 58.00 | +14.03 | +5.79 | 54.22 | -0.73 |
| **ExpertGuard** | **61.28** | **+3.59** | **+25.09** | **48.28** | **+4.31** | **-3.93** | **54.78** | **-0.17** |
| *Scenario B: Training Task = Math, ExpertGuard Task = QA* | | | | | | | | |
| Base Model | 38.97 | -13.24 | – | 35.75 | -0.81 | -11.31 | 37.36 | -12.27 |
| Math Expert | 52.21 | – | +13.24 | 36.56 | – | -9.50 | 44.39 | -4.25 |
| QA Expert | 39.34 | -12.87 | +0.37 | 46.06 | +9.50 | – | 42.70 | -6.44 |
| Agent-Dice | 49.25 | -2.96 | +10.28 | 37.75 | +1.19 | -8.31 | 43.50 | -5.64 |
| Refresh | 50.28 | -1.93 | +11.31 | 38.00 | +1.44 | -8.06 | 44.14 | -5.00 |
| **ExpertGuard** | **49.94** | **-2.27** | **+10.97** | **40.13** | **+3.57** | **-5.93** | **45.04** | **-4.10** |
| *Scenario C: Training Task = QA, ExpertGuard Task = Code* | | | | | | | | |
| Base Model | 35.75 | -10.31 | – | 36.19 | -5.22 | -21.50 | 35.97 | -15.91 |
| QA Expert | 46.06 | – | +10.31 | 41.41 | – | -16.28 | 43.74 | -8.14 |
| Code Expert | 36.75 | -9.31 | +1.00 | 57.69 | +16.28 | – | 47.22 | -4.66 |
| Agent-Dice | 37.75 | -8.31 | +2.00 | 46.06 | +4.65 | -11.63 | 41.91 | -9.97 |
| Refresh | 44.88 | -1.18 | +9.13 | 53.13 | +11.72 | -4.56 | 49.01 | -2.87 |
| **ExpertGuard** | **46.50** | **+0.44** | **+10.75** | **52.78** | **+11.37** | **-4.91** | **49.64** | **-2.24** |

objective, we hypothesize this phenomenon emerges as a beneficial byproduct of implicit regularization: projecting updates onto $\mathcal{S}_t^{\perp}$ restricts the effective dimensionality of the parameter update, mitigating overfitting to spurious reward signals commonly observed in unconstrained LLM RL.

**Multi-expert scaling results.** *ExpertGuard achieves superior multi-domain protection via joint subspace projection (Table 3).*

While the Math Expert collapses severely on guarded domains and Agent-Dice yields capped accuracies, ExpertGuard (Residual) drastically reduces the average multi-domain performance gap to $-2.47\%$, representing a substantial improvement over baselines. Tracking multiple historical boundaries does not bottleneck target learning; instead, it further stabilizes the optimization trajectory, confirming that global subspace residual projection maximizes both stability and plasticity simultaneously compared to sequential stripping.

### 5.3 Ablation Study and Training Dynamics

**Projection protocols within ExpertGuard.** *We investigate two projection protocols (Table 3), validating that global subspace projection maximizes stability and plasticity simultaneously.* The Sequential baseline serially strips gradient components sorted by severity, remaining susceptible to order sensitivity where repairing one domain inadvertently reintroduces negative projections against

Table 3: Multi-expert scalability analysis ($K = 2$). We evaluate the capability of ExpertGuard to simultaneously preserve two distinct domains of prior knowledge while optimizing a third target domain. Training Task is Math, Guarded Task 1 is Code, and Guarded Task 2 is QA. Under ExpertGuard, we report two projection protocols: *Sequential*, an ablation variant that serially projects updates based on conflict order, and *Residual (Ours)*, our proposed method utilizing joint subspace orthogonalized projection. $\Delta_{\text{Exp}}$ denotes the performance gap relative to the corresponding single-task expert upper bound. Avg. Gap represents the mean difference against all experts across the three tasks. Underlined values denote the best results, and bold values highlight ExpertGuard variants.

| **Method** | **Training Task (Math)** | | **Guarded Task 1 (Code)** | | **Guarded Task 2 (QA)** | | **Avg. Gap** |
|---|---|---|---|---|---|---|---|
| | **Acc** | $\Delta_{\text{Exp}}$ | **Acc** | $\Delta_{\text{Exp}}$ | **Acc** | $\Delta_{\text{Exp}}$ | |
| Base Model | 38.97 | -13.24 | 36.19 | -21.50 | 35.75 | -10.31 | -15.02 |
| Math Expert | <u>52.21</u> | – | 35.22 | -22.47 | 36.56 | -9.50 | -10.66 |
| Code Expert | 43.97 | -8.24 | <u>57.69</u> | – | 36.75 | -9.31 | -5.85 |
| QA Expert | 39.34 | -12.87 | 41.41 | -16.28 | <u>46.06</u> | – | -9.72 |
| Agent-Dice | 49.25 | -2.96 | 46.06 | -11.63 | 37.75 | -8.31 | -7.63 |
| **ExpertGuard** | | | | | | | |
| *Sequential* | 52.06 | -0.15 | 46.88 | -10.81 | 39.88 | -6.18 | <u>-5.71</u> |
| *Residual (Ours)* | **53.03** | **+0.82** | **58.34** | **+0.65** | **37.19** | **-8.87** | **-2.47** |

previously sanitized capabilities. In contrast, our Residual protocol projects directly onto the collective orthogonal complement, eliminating mutual non-orthogonal dependencies. Empirically, Residual drastically reduces the average gap from $-5.71\%$ to $-2.47\%$ and boosts both target and guarded task accuracies.

**Sensitivity of conflict threshold** $\tau$. *The zero-boundary threshold ($\tau = 0$) optimally balances plasticity and stability, while tuning $\tau$ flexibly modulates this trade-off.*

Table 4: Sensitivity analysis of the conflict threshold $\tau$ under Scenario A. $\Delta_{\text{Exp}}$ indicates the performance gap relative to the single-task expert upper bound ($57.69\%$ on Code and $52.21\%$ on Math). Avg. Gap represents the mean difference.

| **Threshold $\tau$** | **Training: Code** | | **Guarded: Math** | | **Avg. Gap** |
|---|---|---|---|---|---|
| | **Acc** | $\Delta_{\text{Exp}}$ | **Acc** | $\Delta_{\text{Exp}}$ | |
| $-0.001$ | 61.69 | +4.00 | 47.34 | -4.87 | -0.44 |
| $0.000$ | **61.28** | **+3.59** | **48.28** | **-3.93** | **<u>-0.17</u>** |
| $+0.001$ | 58.91 | +1.22 | 50.13 | -2.08 | -0.43 |

As shown in Table 4, increasing $\tau$ from $-0.001$ to $+0.001$ triggers more frequent projections, which aggressively modifies the learning signal; this enhances historical stability (Math improves from $47.34\%$ to $50.13\%$) but restricts target-task plasticity (Code drops from $61.69\%$ to $58.91\%$), and vice versa for decreasing $\tau$. Operating at the theoretical zero boundary activates intervention precisely when negative transfer occurs, achieving the tightest overall gap ($\text{Avg. Gap} = -0.17\%$) with $61.28\%$ on Code and $48.28\%$ on Math. These trajectories confirm that while $\tau = 0$ serves as the optimal anchor balancing both domains, adjusting $\tau$ provides a practical engineering lever to navigate stability-plasticity preferences.

**Training overhead**. *ExpertGuard introduces virtually zero GPU memory overhead and acceptable latency increments, demonstrating superior scalability for distributed LLM RL.*

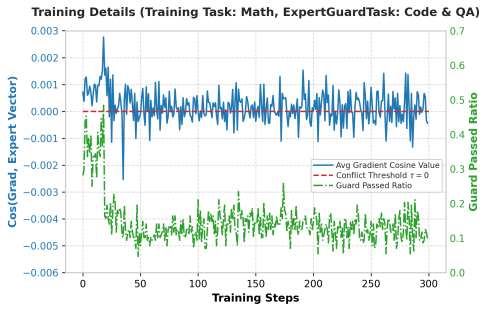
Table 5: Training overhead comparison. Thr. denotes throughput (tokens/s); Mem. represents peak GPU memory usage (GB); Step Time indicates the wall-clock duration per step (s).

| **Method** | **Thr.** | **Mem.** | **Step Time** |
|---|---|---|---|
| Expert Training | 1141.73 | 92.24 | 643.14 |
| **Ours** ($K = 1$) | 927.22 | 94.71 | 747.67 |
| **Ours** ($K = 2$) | 945.61 | 92.10 | 771.59 |

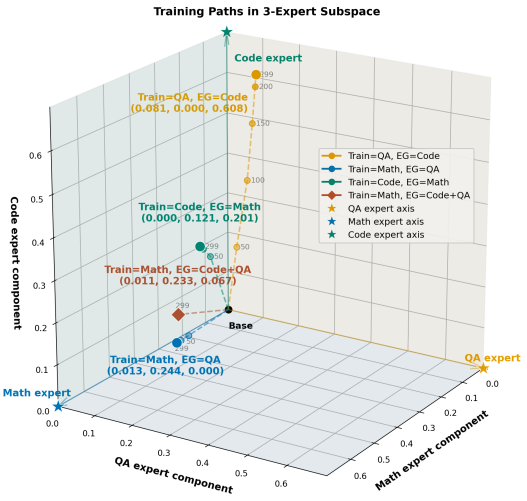
As shown in Table 5, ExpertGuard maintains near-constant peak GPU memory. This stability stems from keeping static expert vectors resident in CPU host RAM, streaming them to the GPU only when the inner-product is computed; thus, the marginal fluctuations are merely experimental noise. Meanwhile, throughput and step time inherently depend on specific prompt inference dynamics during rollouts. These negligible footprints confirm ExpertGuard as a lightweight, plug-and-play solution for large-scale RL.




(a) Single-expert training details under Scenario A.




(b) Dual-expert training details.




(c) Optimization trajectories in the three-expert subspace. Parameter displacements $\Delta\theta_t = \theta_t - \theta_{\text{base}}$ are projected onto the non-orthogonal basis formed by expert vectors $\mathbf{v}_d = \theta_d^* - \theta_{\text{base}}$ ($d \in \{\text{QA}, \text{Math}, \text{Code}\}$). Single-guard runs remain confined to 2D coordinate planes, while the dual-guard run navigates within the positive octant interior ($\mathbf{a} \geq 0$).

Figure 3: **Step-wise dynamics and trajectory geometry of ExpertGuard.** (a)-(b) Cosine values and guard-passed ratios confirm that ExpertGuard selectively intervenes only upon detecting negative transfer, preserving unconflicted learning signals. (c) Projected parameter displacements reveal distinct geometric behaviors across single-guard and dual-guard regimes.

**Step-wise dynamics and trajectory geometry**. *Empirical tracking demonstrates that ExpertGuard executes adaptive, selective interventions rather than rigid projections, steering parameter trajectories strictly within the positive orthant of the historical expert subspace.*

Figure 3a-3b illustrates gradient-expert cosine alignments and non-conflict ratios across training steps. Cosine values fluctuate around zero, stabilizing within a tight $\pm 0.001$ band after warmup due to the intrinsic directional stochasticity of high-dimensional RL. Crucially, the guard-passed ratio is notably lower in the dual-expert scenario, indicating that enforcing two boundaries triggers more frequent interventions. This confirms ExpertGuard selectively purges interference only upon detecting negative transfer ($\alpha_k^{(t)} < 0$), preserving unconflicted exploration signals.

Furthermore, projecting parameter displacements $\Delta\theta_t$ (Figure 3c) reveals distinct geometric behaviors: single-guard results lie within their respective 2D coordinate planes formed by the two relevant expert vectors (rather than computationally yielding an exact zero on the third axis), while the dual-guard trajectory navigates stably within the fully positive octant. Across all regimes, projected coordinates remain strictly non-negative ($a \geq 0$), empirically confirming that ExpertGuard correctly steers updates along—never opposing—historical semantic directions.

## 6 Conclusion

We propose **ExpertGuard**, a lightweight framework that breaks the stability-plasticity trade-off in multi-task LLM RL. By treating static expert vectors as geometric boundaries, ExpertGuard employs a subspace residual gradient projection to project policy updates onto orthogonalized expert null-spaces, eliminating first-order gradient conflicts. Empirical evaluations demonstrate the superior performance of ExpertGuard with minimal computational overhead.

## AI USE STATEMENT

In this work, we used generative AI tools (specifically, Large Language Models) for language polishing, grammatical error correction, and structural refinement of the manuscript text. We have not used generative AI tools for conceptualizing the core research ideas, deriving the mathematical proofs, designing the ExpertGuard algorithm, or generating the experimental data and code.

All AI-assisted text modifications were strictly limited to improving clarity and readability. The authors have manually reviewed, verified, and taken full responsibility for all AI-suggested edits to ensure they accurately reflect our original scientific claims. Specifically, all mathematical formulations (including the exact algebraic identities in Theorem 1 and Proposition 1), theoretical assumptions (Assumption 1), and empirical conclusions (such as the implicit regularization effect discussed in Section 5.2) were derived, validated, and interpreted entirely by the human authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

## ETHICS STATEMENT

This work focuses on improving the stability and plasticity of large language models during reinforcement learning alignment. While our proposed ExpertGuard framework mitigates catastrophic forgetting across multiple domains, it does not inherently address the safety or ethical quality of the underlying reward signals. If the target or historical tasks involve harmful objectives, preserving them via our method could inadvertently retain undesirable behaviors. Therefore, we strongly recommend that practitioners apply ExpertGuard in conjunction with robust safety alignment protocols and human oversight. All datasets used in this study (NuminaMath, KodCode, CrossThinkQA) are publicly available and do not contain personally identifiable information or sensitive human subject data.

## REPRODUCIBILITY STATEMENT

To ensure the statistical reliability and reproducibility of our empirical findings, all experiments and evaluations reported in this work—including both single-domain and multi-expert scaling analyses—are independently conducted across three distinct random seeds. We report the mean performance metrics across these runs to mitigate the variance inherent in on-policy reinforcement learning optimization for large language models. Furthermore, comprehensive implementation details, encompassing hyperparameter configurations, GRPO training arguments, sampling strategies, and hardware specifications, are rigorously documented and provided in Appendix D. These measures guarantee that our proposed ExpertGuard framework can be faithfully reproduced by the research community.

## REFERENCES


Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, et al. Nemotron-crossthink: Scaling self-learning beyond math reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 984–1002, 2026.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pp. 3259–3269. PMLR, 2020.

Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

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.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Percy Liang, Tengyu Ma, Robbie Jones, Aditi Raghunathan, and Ananya Kumar. Fine-tuning can distort pretrained features and underperform out-of-distribution. *International Conference on Learning Representations (ICLR)*, 2022.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *International conference on learning representations*, 2021.

Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14551–14558, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guanglong Sun, Siyuan Zhang, Liyuan Wang, Jun Zhu, Hang Su, and Yi Zhong. Safety alignment as continual learning: Mitigating the alignment tax via orthogonal gradient projection. *arXiv preprint arXiv:2602.07892*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025.

Zheng Wu, Xingyu Lou, Xinbei Ma, Yansi Li, Weiwen Liu, Weinan Zhang, Jun Wang, and Zhuosheng Zhang. Agent-dice: Disentangling knowledge updates via geometric consensus for agent continual learning. *arXiv preprint arXiv:2601.03641*, 2026.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6980–7008, 2025.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026a.

Lei Yang, Siyu Ding, and Deyi Xiong. A local perturbation theory for cross-domain interference and recovery in multi-domain rl. *arXiv preprint arXiv:2606.02398*, 2026b.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Model extrapolation expedites alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1025–1041, 2025.

# A PSEUDOCODE

**Algorithm 1** Subspace Residual Gradient Projection

**Input:** Accumulated gradient $\mathbf{g}_{\text{new}}^{(t)}$, expert vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$, and conflict threshold $\tau$ (default $\tau = 0$ for theoretical guarantees).
**Output:** Corrected gradient $\mathbf{g}'^{(t)}$.

**for** $k = 1$ **to** $K$ **do**
  $\alpha_k^{(t)} \leftarrow \langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k \rangle / \|\mathbf{v}_k\|^2$
**end for**
$\mathcal{I}_t \leftarrow \{k : \alpha_k^{(t)} < \tau\}$  // Trigger intervention if conflict exceeds threshold
**if** $I_t = \varnothing$ **then**
  **return** $\mathbf{g}'^{(t)} \leftarrow \mathbf{g}_{\text{new}}^{(t)}$
**end if**
// Optional: Sort $\mathcal{I}_t = \{\pi_1, \ldots, \pi_M\}$ by increasing $\alpha_{\pi_m}^{(t)}$ for numerical stability in classical Gram–Schmidt.
// The final subspace $S_t^{\perp}$ is uniquely defined and order-invariant (see Remark 2), making sorting theoretically unnecessary.
Apply Gram–Schmidt orthogonalization to the active expert vectors $\{\mathbf{v}_k\}_{k \in \mathcal{I}_t}$
Discard zero or numerically degenerate residuals; denote the remaining orthogonal basis by $\{\mathbf{q}_1, \ldots, \mathbf{q}_r\}$
$\mathbf{g}^{(0)} \leftarrow \mathbf{g}_{\text{new}}^{(t)}$
**for** $j = 1$ **to** $r$ **do**
  $\mathbf{g}^{(j)} \leftarrow \mathbf{g}^{(j-1)} - \frac{\langle \mathbf{g}^{(j-1)}, \mathbf{q}_j \rangle}{\|\mathbf{q}_j\|^2} \mathbf{q}_j$
**end for**
**return** $\mathbf{g}'^{(t)} \leftarrow \mathbf{g}^{(r)}$

# B PRACTICAL CONSIDERATIONS

**Expert vector storage.** At initialization, expert vectors are constructed as $\mathbf{v}_k = \theta_k^* - \theta_{\text{base}}$. The original checkpoint state dictionaries can then be released while retaining the expert vectors. During training, the relevant vectors are applied to the accumulated gradient immediately before the direct parameter update.

**Conflict threshold.** All theoretical results and main experiments use $\tau = 0$. A nonzero $\tau$ may be retained as an engineering-level intervention trigger, but the surgery remains a zero-boundary projection; therefore, the theoretical results above do not directly cover arbitrary $\tau \neq 0$.

**Coupled with distributed framework.** Existing distributed frameworks, such as Megatron-LM, rely on micro-batching gradient updates. To avoid the prohibitive overhead incurred by frequent expert vector switching at every micro-step, ExpertGuard executes gradient conflict detection and surgery once per gradient accumulation boundary—immediately after `all-reduce` but prior to the optimizer step.

**Layer-targeted conflict detection (Optional Strategy).** Empirical observations and recent literature demonstrate that parameter updates in deep layers govern task-specific behavioral alignment, whereas shallow layers predominantly encode universal representations (Liang et al., 2022; Liu et al., 2024; Meng et al., 2022). Balancing computational efficiency with intervention precision, ExpertGuard incorporates a customizable layer selection strategy as an optional module: conflict detection can be evaluated exclusively on a subset of tail-end parameters, and full-parameter gradient surgery is triggered if and only if a conflict is detected. This decoupled paradigm eliminates redundant full-rank operations during non-conflicting steps while guaranteeing comprehensive parameter rectification when interference arises, allowing practitioners to flexibly navigate the trade-off between computational overhead and intervention granularity. We acknowledge that this layer-targeted triggering mechanism is an engineering heuristic designed for computational efficiency. The strict

first-order non-degradation guarantees established in Theorem 1 and Proposition 1 formally apply to the full parameter space $\mathbb{R}^d$. Applying the global projection based solely on tail-layer conflict detection relaxes these formal guarantees in practice, though empirical results demonstrate that tail-layer conflicts serve as highly reliable indicators for global semantic interference in LLMs.

# C Proofs and Supporting Results

This appendix proves the main results in Sections 3 and 4. The analysis assumes that all projected expert vectors are nonzero and that numerically degenerate Gram–Schmidt residuals are discarded.

## C.1 Proofs of the Main Results

### C.1.1 Proof of Theorem 1

By the definition of the single-expert surgery in Eq. 3, the directional derivative difference along the historical objective satisfies:

$$\langle \mathbf{g}_k'^{(t)}, \mathbf{h}_k^{(t)} \rangle - \langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{h}_k^{(t)} \rangle = -\alpha_k^{(t)} \langle \mathbf{v}_k, \mathbf{h}_k^{(t)} \rangle. \tag{11}$$

Since expert $k$ is classified as conflicting, we have $\alpha_k^{(t)} < 0$, which implies $-\alpha_k^{(t)} > 0$. Under Assumption 1, the local directional alignment margin satisfies $\rho_k^{(t)} \triangleq \cos(\mathbf{h}_k^{(t)}, \mathbf{v}_k) > 0$. Expressing the inner product via its cosine similarity yields:

$$\langle \mathbf{v}_k, \mathbf{h}_k^{(t)} \rangle = \rho_k^{(t)} \|\mathbf{v}_k\| \|\mathbf{h}_k^{(t)}\| \geq 0. \tag{12}$$

Multiplying a strictly positive term ($-\alpha_k^{(t)}$) with a non-negative term (Eq. 12) directly establishes the non-degradation guarantee:

$$-\alpha_k^{(t)} \langle \mathbf{v}_k, \mathbf{h}_k^{(t)} \rangle \geq 0. \tag{13}$$

Furthermore, to characterize the exact magnitude of this recovery, we substitute the definition of the projection coefficient $\alpha_k^{(t)} = \frac{\|\mathbf{g}_{\text{new}}^{(t)}\| \|\mathbf{v}_k\| \cos(\mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k)}{\|\mathbf{v}_k\|^2}$ into Eq. 11. Since $\alpha_k^{(t)} < 0$ is strictly equivalent to $\cos(\mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k) < 0$, we obtain the exact algebraic identity:

$$-\alpha_k^{(t)} \langle \mathbf{v}_k, \mathbf{h}_k^{(t)} \rangle = \rho_k^{(t)} \left| \cos(\mathbf{g}_{\text{new}}^{(t)}, \mathbf{v}_k) \right| \|\mathbf{g}_{\text{new}}^{(t)}\| \|\mathbf{h}_k^{(t)}\|. \tag{14}$$

Since $\rho_k^{(t)} > 0$ by assumption and the absolute value ensures non-negativity, the right-hand side is strictly non-negative. This exact algebraic identity directly establishes Eq. 4 in the main text, completing the proof. ■

**Remark 3.** *The exact identity derived in Eq. 14 formalizes two critical properties of ExpertGuard beyond basic stability preservation:*

1. *Non-infinitesimal Recovery: The directional improvement scales jointly with the gradient norms $\|g_{\text{new}}^{(t)}\| \|h_k^{(t)}\|$. This proves that the recovery margin is a primary first-order quantity governed by the active optimization dynamics, rather than an infinitesimal artifact bounded by higher-order curvature terms.*

2. *Natural Adaptive Scaling: The recovery is explicitly scaled by the proxy conflict severity $|\cos(g_{\text{new}}^{(t)}, v_k)|$. This indicates that ExpertGuard possesses an inherent adaptive mechanism: more severe destructive interference mathematically guarantees a proportionally larger stability recovery, eliminating the need for manual tuning of intervention strengths during on-policy RL.*

### C.1.2 Remark C.2: Robustness to Transient Assumption Violations

While Figure 2 demonstrates strong empirical adherence to Assumption 1, the inherent variance of on-policy RL gradients implies that transient, microscopic violations (i.e., isolated steps where $\rho_k^{(t)} < 0$) cannot be theoretically excluded. We clarify why ExpertGuard remains robust under such edge cases:

1. *Expectation-Level Guarantee:* Theorem 1 establishes the recovery margin proportional to $\rho_k^{(t)}$. Even if $\rho_k^{(t)}$ experiences transient negative spikes due to stochastic batch sampling, the KL-regularized trust region ensures that the expected alignment over the sampling distribution at any step $t$ remains strictly positive, i.e., $\mathbb{E}_{\text{batch}}[\rho_k^{(t)}] > 0$. Thus, the cumulative stability guarantee holds in expectation over the optimization trajectory.
2. *Algorithmic Fallback Mechanism:* Crucially, ExpertGuard's intervention is gated by the active conflict detection condition ($\alpha_k^{(t)} < 0$). If a transient violation occurs where both $\alpha_k^{(t)} < 0$ and $\rho_k^{(t)} < 0$, the projection operation forces the updated gradient to satisfy $\langle \mathbf{g}_k'^{(t)}, \mathbf{v}_k \rangle = 0$. Geometrically, this means the algorithm defaults to halting any further movement along the proxy direction $\mathbf{v}_k$. While it may not yield the optimal positive recovery margin in that specific anomalous step, projecting to the null-space definitively prevents catastrophic degradation along the expert vector, serving as a safe theoretical fallback.

### C.1.3 Gram–Schmidt Subspace Preservation

Every retained Gram–Schmidt residual is a linear combination of the original active expert vectors, so

$$\operatorname{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_r\} \subseteq \mathcal{S}_t.$$

Conversely, each nondegenerate original active vector can be reconstructed recursively from the retained residuals and preceding basis vectors. Therefore,

$$\operatorname{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_r\} = \mathcal{S}_t.$$

The recursive construction also gives $\langle \mathbf{q}_i, \mathbf{q}_j \rangle = 0$ for $i \neq j$.

### C.1.4 Proof of the Cascaded Projection Equivalence

We prove by induction that

$$\mathbf{g}^{(m)} = \mathbf{g}_{\text{new}}^{(t)} - \sum_{j=1}^{m} \frac{\langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{q}_j \rangle}{\|\mathbf{q}_j\|^2} \mathbf{q}_j. \tag{15}$$

The statement is immediate for $m = 0$. Suppose it holds for $m - 1$. Pairwise orthogonality gives

$$\left\langle \mathbf{g}^{(m-1)}, \mathbf{q}_m \right\rangle = \left\langle \mathbf{g}_{\text{new}}^{(t)}, \mathbf{q}_m \right\rangle.$$

Substitution into Eq. 6 yields Eq. 15 for $m$. Taking $m = r$ gives

$$\mathbf{g}'^{(t)} = \left( \mathbf{I} - \sum_{j=1}^{r} \frac{\mathbf{q}_j \mathbf{q}_j^\top}{\|\mathbf{q}_j\|^2} \right) \mathbf{g}_{\text{new}}^{(t)} = \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\text{new}}^{(t)},$$

which proves Eq. 7.

### C.1.5 Proof of Active Expert Orthogonality

By Eq. 7, $\mathbf{g}'^{(t)} \in \mathcal{S}_t^\perp$. Since $\mathbf{v}_k \in \mathcal{S}_t$ for every $k \in \mathcal{I}_t$, it follows that

$$\left\langle \mathbf{g}'^{(t)}, \mathbf{v}_k \right\rangle = 0, \qquad k \in \mathcal{I}_t.$$

This proves Eq. 1.

### C.1.6 Proof of Proposition 1 (Multi-Expert Directional Decoupling)

**Part (i): Simultaneous Proxy Recovery.** By the Gram–Schmidt construction in Eq. 5, the retained orthogonal basis $\{q_1, \ldots, q_r\}$ spans the identical linear subspace as the active expert proxies:

$$S_t \triangleq \mathrm{span}\{v_{\pi_1}, \ldots, v_{\pi_M}\} = \mathrm{span}\{q_1, \ldots, q_r\}. \tag{16}$$

The joint orthogonal complement projector is defined as $P_{S_t^\perp} = I - \sum_{j=1}^{r} \frac{q_j q_j^\top}{\|q_j\|^2}$. Since every active expert vector $v_k$ ($k \in I_t$) belongs to $S_t$ by definition, the fundamental property of orthogonal projection dictates that $P_{S_t^\perp} v_k = \mathbf{0}$. Utilizing the self-adjointness of the projector ($P_{S_t^\perp}^\top = P_{S_t^\perp}$), we have:

$$\langle g'^{(t)}, v_k \rangle = \langle P_{S_t^\perp} g_{\mathrm{new}}^{(t)}, v_k \rangle = \langle g_{\mathrm{new}}^{(t)}, P_{S_t^\perp} v_k \rangle = 0. \tag{17}$$

Since expert $k \in I_t$ is conflicting if and only if $\langle g_{\mathrm{new}}^{(t)}, v_k \rangle < 0$, it immediately follows that:

$$\langle g'^{(t)}, v_k \rangle - \langle g_{\mathrm{new}}^{(t)}, v_k \rangle = -\langle g_{\mathrm{new}}^{(t)}, v_k \rangle = \left| \langle g_{\mathrm{new}}^{(t)}, v_k \rangle \right| > 0, \tag{18}$$

which establishes Eq. 8 for all $k \in I_t$ simultaneously.

**Part (ii): Algebraic Decoupling Guarantee (Absence of Order Sensitivity).** To formally establish why joint projection avoids the error accumulation inherent in cascaded pairwise projections, consider the operation applied to an arbitrary intermediate gradient $u$. Because $P_{S_t^\perp}$ is constructed upon a mutually orthogonal basis, it is strictly idempotent:

$$P_{S_t^\perp}^2 = P_{S_t^\perp}. \tag{19}$$

This idempotence implies that once the gradient is projected onto $S_t^\perp$, subsequent projections along any direction within $S_t$ yield zero displacement. In contrast, sequential projection operators $P_{v_i^\perp} P_{v_j^\perp}$ for non-orthogonal proxies $v_i, v_j$ do not commute and are not idempotent, meaning projecting out $v_j$ can reintroduce a negative component along $v_i$. ExpertGuard's formulation maps the multi-expert conflict resolution into a single, unique, and order-invariant geometric operator, thereby definitively neutralizing negative transfer without supplementary structural assumptions. ■

### C.1.7 Proof of the Minimum-Distortion Property

For any feasible $\mathbf{u} \in \mathcal{S}_t^\perp$, write

$$\mathbf{g}_{\mathrm{new}}^{(t)} = \mathbf{P}_{\mathcal{S}_t} \mathbf{g}_{\mathrm{new}}^{(t)} + \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\mathrm{new}}^{(t)}.$$

Then

$$\mathbf{u} - \mathbf{g}_{\mathrm{new}}^{(t)} = \left( \mathbf{u} - \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\mathrm{new}}^{(t)} \right) - \mathbf{P}_{\mathcal{S}_t} \mathbf{g}_{\mathrm{new}}^{(t)}.$$

The two terms on the right belong to orthogonal subspaces. Hence,

$$\left\| \mathbf{u} - \mathbf{g}_{\mathrm{new}}^{(t)} \right\|^2 = \left\| \mathbf{u} - \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\mathrm{new}}^{(t)} \right\|^2 \tag{20}$$

$$+ \left\| \mathbf{P}_{\mathcal{S}_t} \mathbf{g}_{\mathrm{new}}^{(t)} \right\|^2 \tag{21}$$

$$\geq \left\| \mathbf{P}_{\mathcal{S}_t} \mathbf{g}_{\mathrm{new}}^{(t)} \right\|^2. \tag{22}$$

Equality holds if and only if $\mathbf{u} = \mathbf{P}_{\mathcal{S}_t^\perp} \mathbf{g}_{\mathrm{new}}^{(t)} = \mathbf{g}'^{(t)}$. Thus, Eq. 9 holds.

### C.1.8 Lemma C.2: Subspace-Confined Perturbation on True Gradients

While Proposition 1 guarantees recovery along the static proxies $\mathbf{v}_k$, we further characterize the net modification introduced by the surgery along the true online historical gradient $\mathbf{h}_k^{(t)} \in \mathbb{R}^d$. According to the orthogonal decomposition theorem, $\mathbb{R}^d = S_t \oplus S_t^\perp$. We uniquely decompose $\mathbf{h}_k^{(t)}$ as:

$$\mathbf{h}_k^{(t)} = \mathbf{h}_{k,\parallel}^{(t)} + \mathbf{h}_{k,\perp}^{(t)}, \quad \text{where } \mathbf{h}_{k,\parallel}^{(t)} \triangleq P_{S_t} \mathbf{h}_k^{(t)} \in S_t,\ \mathbf{h}_{k,\perp}^{(t)} \triangleq P_{S_t^\perp} \mathbf{h}_k^{(t)} \in S_t^\perp. \tag{23}$$

The net modification vector satisfies $\mathbf{g}'^{(t)} - \mathbf{g}_{\text{new}}^{(t)} = (P_{S_t^\perp} - I)\mathbf{g}_{\text{new}}^{(t)} = -P_{S_t}\mathbf{g}_{\text{new}}^{(t)}$. Taking the inner product with $\mathbf{h}_k^{(t)}$:

$$\langle \mathbf{g}'^{(t)} - \mathbf{g}_{\text{new}}^{(t)}, \mathbf{h}_k^{(t)} \rangle = \langle -P_{S_t}\mathbf{g}_{\text{new}}^{(t)}, \mathbf{h}_{k,\|}^{(t)} + \mathbf{h}_{k,\perp}^{(t)} \rangle \tag{24}$$

$$= -\langle P_{S_t}\mathbf{g}_{\text{new}}^{(t)}, \mathbf{h}_{k,\|}^{(t)} \rangle - \underbrace{\langle P_{S_t}\mathbf{g}_{\text{new}}^{(t)}, \mathbf{h}_{k,\perp}^{(t)} \rangle}_{=0 \text{ (orthogonality)}} \tag{25}$$

$$= -\langle P_{S_t}\mathbf{g}_{\text{new}}^{(t)}, P_{S_t}\mathbf{h}_k^{(t)} \rangle. \tag{26}$$

This identity proves that, under the first-order proxy framework, the surgical perturbation is strictly confined to the active expert subspace $S_t$. Components of the true historical gradient residing in $S_t^\perp$ are theoretically undisturbed by the projection operator itself, providing a principled algebraic basis for plasticity preservation beyond the proxy level.

#### C.1.9 Optional Finite-Step Consequence

If $\nabla_\theta \mathcal{J}_k$ is locally $L_{k,t}$-Lipschitz along the update segment, then

$$\mathcal{J}_k(\theta^{(t+1)}) - \mathcal{J}_k(\theta^{(t)}) \tag{27}$$

$$\geq \eta_t \left\langle \mathbf{h}_k^{(t)}, \mathbf{g}'^{(t)} \right\rangle - \frac{L_{k,t}\eta_t^2}{2} \|\mathbf{g}'^{(t)}\|^2. \tag{28}$$

Therefore, a first-order directional guarantee does not imply finite-step monotonicity of the historical objective; even a zero first-order term may be followed by a negative second-order contribution.

## D Experiments

### D.1 Detailed Training Settings

The complete ROLL training configuration (YAML format) used in our **ExpertGuard** experiments under the Group Relative Policy Optimization (GRPO) setting is presented in Listing 1.

Listing 1: ExpertGuard Key Configuration

```yaml
# Base configurations
max_steps: 300
rollout_batch_size: 64  # prompt
prompt_length: 1024
response_length: 4096

# GRPO-related
num_return_sequences_in_group: 16
ppo_epochs: 1
adv_estimator: "grpo"
loss_agg_mode: "seq-mean-token-mean"

# KL
use_kl_loss: true
kl_loss_coef: 0.15

# clip
value_clip: 0.5
reward_clip: 10
advantage_clip: 1.0
dual_clip_loss: true
pg_clip: 0.2
max_grad_norm: 0.8
train_infer_ratio_mask: false

# normalize
norm_mean_type: ~
norm_std_type: ~

# data mask
max_len_mask: true
difficulty_mask: true
difficulty_low_threshold: 0.05
difficulty_high_threshold: 0.95
```

```yaml
error_max_len_clip: true
error_max_len_threshold: 4096

# data weight
difficulty_loss_weight: false
length_loss_weight: false

# reward
add_token_level_kl: false

# advantage
whiten_advantages: false

validation:
  generating_args:
    max_new_tokens: ${response_length}
    top_p: 0.95
    top_k: 50
    num_beams: 1
    temperature: 0.6
    num_return_sequences: 1

actor_train:
  training_args:
    learning_rate: 5.0e-7
    lr_scheduler_type: cosine
    lr_scheduler_kwargs:
      min_lr: 1.0e-7
    weight_decay: 0.01
    gradient_accumulation_steps: 8
    warmup_steps: 30
    num_train_epochs: 1
  infer_batch_size: 2

actor_infer:
  generating_args:
    max_new_tokens: ${response_length}
    top_p: 0.95
    top_k: 50
    num_beams: 1
    temperature: 0.8
    num_return_sequences: ${num_return_sequences_in_group}
  data_args:
    template: qwen3
  strategy_args:
    strategy_name: vllm
    strategy_config:
      gpu_memory_utilization: 0.6
      block_size: 16
      max_model_len: 6000
  infer_batch_size: 2

reference:
  infer_batch_size: 2

rewards:
  math_rule:
    infer_batch_size: 2
    query_filter_config:
      type: mean_filter
      filter_args:
        threshold_up: 0.95
        threshold_down: 0.05
```

### D.2 Additional Experimental Results

In this subsection, we extend our evaluation to the Qwen3-4B-Base model to assess the generalizability of ExpertGuard. We benchmark our method against the corresponding single-domain experts and PCGrad (Yu et al., 2020), a strong multi-task learning baseline that leverages historical online gradients for conflict resolution. Specifically, we adopt Math as the training task and QA as the ExpertGuard task, which corresponds to Scenario B in Table 2. For PCGrad, historical gradients are computed with a batch size of 8 at intervals of every 5 training steps. The total training budget is set to 100 steps, while all other hyperparameters remain consistent with those detailed in Appendix D.

Table 6: Cross-domain generalization on Qwen3-4B-Base. We evaluate ExpertGuard against single-domain experts and PCGrad using Math as the training task and QA as the guarded domain (corresponding to Scenario B in Table 2). For PCGrad, historical gradients are computed every 5 steps with a batch size of 8. Training runs for 100 steps; all other hyperparameters follow Appendix D. Under **Training Task**, $\Delta_{\text{Exp}}$ and $\Delta_{\text{Base}}$ denote performance gaps relative to the training expert ($A - A^{\text{train}}_{\text{train_expert}}$) and the base model ($A - A^{\text{train}}_{\text{base}}$), respectively. Under **ExpertGuard Task**, $\Delta_{\text{TrainExp}}$ measures the anti-forgetting gain over the unprotected training expert ($A - A^{\text{guard}}_{\text{train_expert}}$), while $\Delta_{\text{GdExp}}$ indicates the gap to the guarded-domain expert upper bound ($A - A^{\text{guard}}_{\text{guard_expert}}$). Under **Overall Trade-off**, Mean Acc. is the unweighted average accuracy across both tasks, and Avg. Gap is the mean deviation from respective domain experts ($\frac{\Delta_{\text{Exp}}+\Delta_{\text{GdExp}}}{2}$). Bold denotes our method; underlined values indicate the best results.

| **Method** | **Training Task (Plasticity)** | | | **ExpertGuard Task (Stability)** | | | **Overall Trade-off** | |
|---|---|---|---|---|---|---|---|---|
| | **Acc** | $\Delta_{\text{Exp}}$ | $\Delta_{\text{Base}}$ | **Acc** | $\Delta_{\text{TrainExp}}$ | $\Delta_{\text{GdExp}}$ | **Mean Acc.** | **Avg. Gap** |
| Base Model | 38.31 | -6.88 | – | 22.50 | -9.19 | -19.59 | 30.41 | -13.24 |
| Math Expert | 45.19 | – | +6.88 | 31.69 | – | -10.40 | 38.44 | -5.20 |
| QA Expert | 40.99 | -4.20 | +2.68 | 42.09 | +10.40 | – | 41.54 | -2.10 |
| PCGrad | 40.99 | -4.20 | +2.68 | 32.00 | +0.31 | -10.09 | 36.50 | -7.15 |
| **ExpertGuard** | **48.00** | **+2.81** | **+9.69** | **37.01** | **+5.32** | **-5.08** | **42.51** | **-1.14** |

**Generalization on Qwen3-4B-Base.** *ExpertGuard generalizes effectively to larger architectures (Table 6), achieving a superior stability-plasticity trade-off compared to single-domain experts and gradient-based baselines under constrained training budgets.*

*(i) Stability on the guarded domain.* While the QA Expert naturally preserves optimal performance on its native task, it incurs non-trivial degradation on the training task ($\Delta_{\text{Exp}} = -4.20$). Conversely, the Math Expert suffers catastrophic forgetting on the QA task, dropping substantially below the base model initialization ($\Delta_{\text{TrainExp}} = -9.19$ vs. Base Acc of 22.50). PCGrad fails to reconcile this conflict: despite reducing the historical gradient computation interval to every 5 steps, its batch size of 8 samples remains insufficient to reliably approximate the global gradient direction. Unlike a static reference vector that consistently defines the protected subspace, these highly stochastic online mini-batch gradients yield noisy projections, resulting in severe instability ($\Delta_{\text{GdExp}} = -10.09$). In contrast, ExpertGuard systematically secures stability, maintaining guarded-domain performance within a minimal gap to the expert upper bound ($\Delta_{\text{GdExp}} = -5.08$), significantly outperforming both the unprotected Math Expert and the PCGrad baseline.

*(ii) Plasticity on the target domain.* Conventional multi-task optimization often suppresses learning plasticity to enforce stability. However, ExpertGuard attains the highest training-task accuracy (48.00) and the lowest average deviation from respective domain experts (Avg. Gap $= -1.14$) across the overall trade-off. Notably, ExpertGuard yields target-task performance slightly exceeding the unconstrained Math Expert baseline ($+2.81$). Rather than being a primary design objective, we hypothesize this phenomenon emerges as a beneficial byproduct of implicit regularization: projecting parameter updates onto $\mathcal{S}_t^{\perp}$ restricts the effective dimensionality of the optimization trajectory. This structural constraint mitigates overfitting to spurious reward signals commonly observed in unconstrained LLM reinforcement learning, thereby enhancing generalization on the target task even within a limited training budget of 100 steps.