REASONING-PRESERVING FINE-TUNING OF POST-RL LLMS WITH NULL-BASIS LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL)-based post-training has become an effective approach for eliciting reasoning capabilities in large language models (LLMs). However, adapting post-RL models to new knowledge domains or behaviors through subsequent supervised fine-tuning (SFT) can severely overwrite these capabilities. Existing approaches mitigate such forgetting through experience replay, specialized initialization, or constrained optimization using gradient projection, but either provide limited preservation or incur substantial training overhead. Our analysis shows that reasoning activations concentrate in low-dimensional subspaces, leaving substantial null-space capacity for adaptation, and that the corresponding approximate null spaces can be reliably estimated from a modest number of examples. Motivated by these observations, we propose Null-Basis Low-Rank Adaptation (NB-LoRA), a parameter-efficient method for adapting post-RL LLMs while preserving their acquired reasoning ability. We formulate reasoning retention as a layer-wise hidden-state preservation constraint and construct a fixed approximate null basis from reasoning activations. LoRA updates are then reparameterized through this basis, enforcing the preservation constraint throughout fine-tuning. Extensive experiments across multiple RL-trained LLMs and diverse downstream tasks show that NB-LoRA matches standard LoRA in adaptation performance, maintains reasoning accuracy near pre-fine-tuning levels, and generalizes this preservation to held-out reasoning benchmarks.

1 INTRODUCTION

Reinforcement learning (RL) based post-training has become an effective approach for eliciting reasoning capabilities in large language models (LLMs) (Shao et al., 2024; Yang et al., 2024). However, most current reasoning-oriented RL methods primarily exploit capabilities already encoded in the base model by reshaping its output distribution, and typically fall short of instilling new knowledge. Moreover, the resulting models often generate lengthy reasoning traces, even for tasks where concise responses would suffice. Subsequent supervised fine-tuning (SFT) therefore remains necessary to inject new knowledge or adapt the model's response style in some target tasks. Yet these RL-acquired capabilities are fragile under such adaptation: RL only mildly perturbs the base model, staying close to it in distribution (Shenfeld et al., 2025), whereas the subsequent SFT is not similarly constrained and can drift far from that distribution, easily overwriting the RL-shaped reasoning behavior (Niu et al., 2026).

Retaining previously acquired capabilities while adapting to new data is a classic problem of catastrophic forgetting, for which continual learning provides several natural solutions (Kirkpatrick et al., 2017). Conventional approaches such as experience replay (Rolnick et al., 2019) and null-space update projection (Wang et al., 2021), however, can be difficult to apply to reasoning-oriented LLM adaptation. Experience replay requires carefully balancing prior reasoning data with new-task data: insufficient replay may fail to preserve reasoning, whereas excessive replay may hinder adaptation to the new task. Null-space update projection methods instead constrain full-model updates, introducing substantial computational and memory overhead at LLM scale. More recent parameter-efficient approaches mitigate forgetting through low-rank adaptation (LoRA) and carefully chosen LoRA initialization (Wang et al., 2025). However, due to unconstrained update, these methods may still overwrite the reasoning behavior acquired during RL post-training, which can be *particularly sensitive* to further adaptation. This raises the central question of this paper:

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.

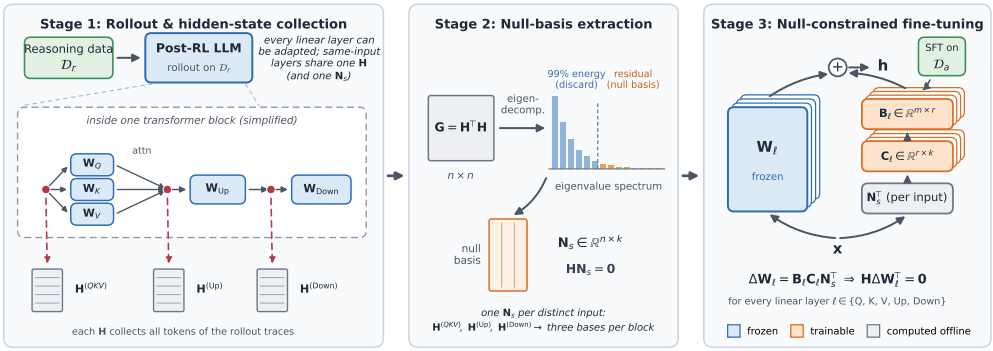


Figure 1: Overview of NB-LoRA. **Stage 1:** We use the post-RL model to generate rollouts on reasoning data $\mathcal{D}_r$ and collect the input hidden states $\mathbf{H}$ of each target linear layer. **Stage 2:** The eigendecomposition of $\mathbf{H}^\top\mathbf{H}$ yields an approximate null basis $\mathbf{N}_s$ from the eigenvectors outside the top ϵ (i.e., 99%) spectral energy, so that $\mathbf{HN}_s = \mathbf{0}$. **Stage 3:** The update is reparameterized as $\Delta\mathbf{W} = \mathbf{BCN}_s^\top$, where $\mathbf{B}$ and $\mathbf{C}$ are trained on the adaptation data $\mathcal{D}_a$, while $\mathbf{N}_s$ remains frozen.

How can we efficiently adapt an RL-trained LLM to new tasks while preserving its acquired reasoning ability?

1.1 Contribution

To address this, we propose Null-Basis LoRA (NB-LoRA), a parameter-efficient fine-tuning method that reparameterizes LoRA updates through a fixed null basis of reasoning hidden states. The idea builds on the fact that if an update does not change a layer's outputs on the hidden states produced by reasoning examples, the reasoning computation carried by those states is left intact. Specifically, we first compute, once and offline, a basis $\mathbf{N}_s$ for the (approximate) null space of the reasoning hidden-state matrix $\mathbf{H}$, and reparameterize the update as $\Delta\mathbf{W} = \mathbf{BCN}_s^\top$ with $\mathbf{N}_s$ frozen and the low-rank factors $\mathbf{B}$ and $\mathbf{C}$ trainable (Figure 1). Any $\mathbf{B}$ and $\mathbf{C}$ satisfy the preservation constraint $\mathbf{H}\Delta\mathbf{W}^\top = \mathbf{0}$ exactly, enabling unconstrained optimization with standard optimizers.

Our main contributions are summarized as follows.

- We formulate post-RL reasoning retention as a layer-wise hidden-state preservation constraint and introduce Null-Basis LoRA (NB-LoRA), a null-basis low-rank reparameterization that satisfies this constraint by construction while retaining the optimizer simplicity of standard LoRA.
- We show that reasoning activations in RL-trained LLMs occupy a low-dimensional subspace despite long reasoning traces, leaving a substantial approximate null space for adaptation that can be estimated from only a small set of reasoning examples.
- We empirically demonstrate that NB-LoRA preserves RL-acquired reasoning ability while maintaining strong downstream adaptation performance, generalizes retention to held-out reasoning benchmarks, and substantially improves training efficiency over existing works.

2 Related Work

Interaction between SFT and RL in LLM post-training. RL has been widely adopted in LLM post-training to improve reasoning capabilities (Yang et al., 2024; Shao et al., 2024). Typical post-training pipelines combine SFT and RL because they provide complementary learning signals: SFT introduces expert behavior and new knowledge, whereas online RL refines the model through exploration. ReFT follows the conventional SFT-then-RL pipeline (Trung et al., 2024), while DeepSeek-R1 employs a multistage SFT-RL-SFT-RL pipeline (Guo et al., 2025). Recently, Ma et al. (2026) proposed ReLIFT, which interleaves RL with online SFT on problems beyond the model's capabilities. Because its SFT data are collected during RL, the two training objectives are tightly coupled. In general, however, subsequent SFT may target a different task and reduce the reward acquired through RL (Niu et al., 2026). We study this RL-then-SFT setting, aiming to adapt the model without overwriting its RL-acquired reasoning.

Mitigating catastrophic forgetting. Existing approaches to catastrophic forgetting include regularization, replay, on-policy distillation, and update constraints. Elastic weight consolidation penalizes changes to parameters important for prior tasks (Kirkpatrick et al., 2017), while replay mixes stored (Rolnick et al., 2019) or model-generated examples (Sun et al., 2020) with new-task data. SDFT uses a demonstration-conditioned copy of the model to supervise on-policy trajectories, reducing distribution mismatch and forgetting, but is limited by the model's in-context learning ability (Shenfeld et al., 2026). Constraint-based methods prevent loss increases on prior tasks (Lopez-Paz & Ranzato, 2017) or project gradients away from directions affecting previous outputs (Farajtabar et al., 2020). More closely related to our work, NSCL projects updates into activation-derived null spaces at each step (Wang et al., 2021). Unlike these soft, replay-based, or per-step constraints, the proposed NB-LoRA encodes a fixed reasoning-preservation constraint directly into the adapter.

Parameter-efficient fine-tuning. Parameter-efficient fine-tuning (PEFT) adapts a frozen backbone by training only a small number of additional parameters (Houlsby et al., 2019; Li & Liang, 2021). A prominent PEFT approach, LoRA, parameterizes weight updates with trainable low-rank factors (Hu et al., 2022). Several extensions apply LoRA to continual learning. O-LoRA assigns sequential tasks to orthogonal low-rank subspaces (Wang et al., 2023), while LB-CL transfers task-relevant low-rank parameters and projects gradients to maintain orthogonality with prior task subspaces (Qiao & Mahdavi, 2024). Merge Before Forget instead orthogonally initializes and continually merges task updates into a single LoRA, maintaining constant memory across tasks (Qiao & Mahdavi, 2026). These methods require prior tasks to be adapted through LoRA, whereas our setting protects reasoning already embedded in an RL-trained backbone. LoRA-Null and MiLoRA initialize adapters using null or minor singular subspaces (Tang et al., 2026; Wang et al., 2025), but do not constrain subsequent updates. Our proposed NB-LoRA instead enforces preservation throughout training by null-basis-based reparameterization.

3 Null-Basis-based Parameter-Efficient Fine-Tuning

3.1 Problem Setup

We consider a reasoning-capable LLM obtained via RL-based post-training. Let $\mathcal{D}_r$ denote the dataset used in the RL stage, whose induced capability we wish to preserve, and let $\mathcal{D}_a$ denote an adaptation dataset for subsequent SFT, such as knowledge injection or concise-response tuning. These two cases are representative of common post-RL adaptation goals: one seeks to augment the model's knowledge, while the other seeks to modify output behavior on target domains without altering its reasoning competence. Our goal is to optimize the model on $\mathcal{D}_a$ while minimizing interference with reasoning performance on $\mathcal{D}_r$.

We focus on adapting linear layers in the model, because transformer-based LLMs are built largely around linear modules, particularly the attention and MLP projections. For a linear layer with weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, standard fine-tuning learns an update $\Delta\mathbf{W}$ and replaces $\mathbf{W}$ with $\mathbf{W} + \Delta\mathbf{W}$. Such unconstrained updates can modify the layer outputs for both adaptation and reasoning examples. The objective of this work is to constrain the update $\Delta\mathbf{W}$ to a restricted subspace in a parameter- and training-efficient manner, such that the update does not alter the model outputs on reasoning data.

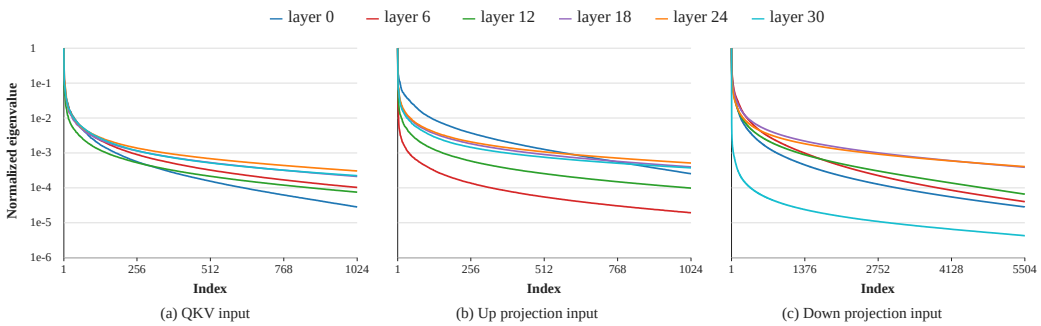
3.2 Hidden-State Preservation Constraint

Let $\mathbf{H} \in \mathbb{R}^{N\times n}$ denote a matrix of hidden states collected from $\mathcal{D}_r$ for the input of a target linear layer, where each row of $\mathbf{H}$ corresponds to one token representation and N is the total number of collected tokens. To preserve the pre-fine-tuning transformation of these reasoning states, we require

$$\mathbf{H}(\mathbf{W} + \Delta\mathbf{W})^\top = \mathbf{H}\mathbf{W}^\top. \tag{1}$$

Equation (1) is equivalent to $\mathbf{H}\Delta\mathbf{W}^\top = \mathbf{0}$. Therefore, a sufficient condition for preserving the layer outputs on reasoning states is that every column of $\Delta\mathbf{W}^\top$ lies in the null space of $\mathbf{H}$.

This constraint has two desirable properties. First, it provides layer-wise and fine-grained preservation signals by constraining each adapted linear layer with hidden states observed on reasoning data, thereby directly guiding the search space of $\Delta\mathbf{W}$ toward directions that preserve reasoning-related



(a) QKV input (b) Up projection input (c) Down projection input

Figure 2: Normalized eigenvalue spectra of $\mathbf{H}^\top\mathbf{H}$ across representative Qwen2.5-3B layers, shown on a log scale.

capabilities. Second, in transformer blocks, multiple projections such as $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ share the same input hidden states, so the corresponding null space can be computed once and reused across these projections, substantially reducing the computational cost of constructing the constraint.

3.3 Feasibility of Null-Space Constraints

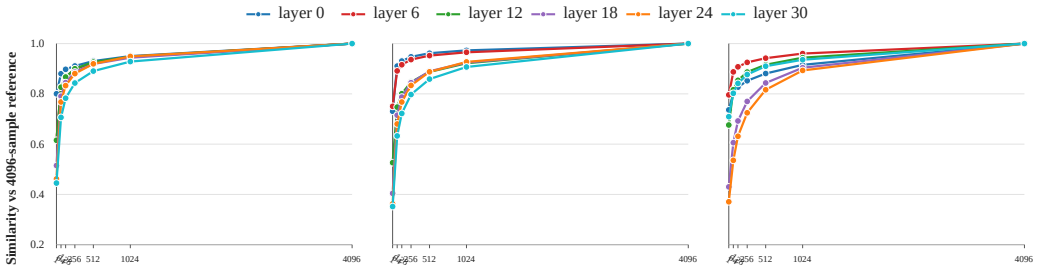
Two key questions arise when applying null-space constraints to reasoning models. *1) Does* $\mathbf{H}$ *admit a sufficiently large approximate null space?* Long reasoning traces produce many token-level observations ($N \gg n$), potentially increasing $\mathrm{rank}(\mathbf{H})$ toward n and reducing its null-space dimension, $\dim(\mathrm{Null}(\mathbf{H})) = n - \mathrm{rank}(\mathbf{H})$. *2) Can this null space be effectively estimated using only a limited number of reasoning examples?* Constructing $\mathbf{H}$ from a large reasoning corpus can be computationally expensive. To answer these questions, we empirically analyze the spectrum and null-space structure of $\mathbf{H}$ using hidden states extracted from Qwen2.5-3B (Qwen et al., 2024) after RL post-training on the MATH-lighteval dataset (Hendrycks et al., 2021). Notably, the null space of $\mathbf{H}$ can be obtained from the eigendecomposition of the much smaller $n \times n$ matrix $\mathbf{H}^\top\mathbf{H}$. Since hidden states are noisy and $\mathbf{H}$ is rarely exactly low rank, we define an approximate null space following the approach of Wang et al. (2021): we retain the principal directions that capture 99% of the spectral energy and treat the remaining directions as the approximate null-space basis.

Table 1: Approximate null-space dimensions (k) of representative Qwen2.5-3B layers under a 99% energy-retention threshold. Percentages are relative to the total dimension.

Module	Total Dimension	Layer 0	Layer 6	Layer 12	Layer 18	Layer 24	Layer 30	Average (%)
QKV	2048	1378	765	755	515	407	393	702 (34.3%)
Down	11008	7028	7343	5474	2272	1701	5940	4960 (45.1%)
Up	2048	1037	1138	574	325	259	251	597 (29.2%)

Spectrum analysis. We collect hidden states (i.e., $\mathbf{H}$) from all the tokens in the reasoning traces of MATH-lighteval training samples. Figure 2 shows the normalized eigenvalue spectrum of $\mathbf{H}^\top\mathbf{H}$ for all the linear modules across layers 0, 6, 12, 18, 24, and 30 of Qwen2.5-3B. In all cases, the spectrum decays rapidly by several orders of magnitude within the first few hundred indices, indicating that the hidden states of reasoning traces span a proper subspace of the full representation space. We report the dimensions of the approximate null spaces (99% energy threshold) of the hidden-state matrices for each module in Table 1, which confirms that substantial null-space capacity remains available for further adaptation.

Null-space Structure Analysis. Since reasoning traces exhibit similar structural patterns (Jiang et al., 2025; Sun et al., 2026), we hypothesize that the null-space structure can be reliably estimated from a moderate number of examples. To verify this, we compare approximate null spaces estimated using different numbers of examples by measuring the principal angles between them. Specifically, we use the null space estimated from 4096 examples as a reference and evaluate its similarity to those estimated using fewer examples. Let $\mathcal{N}_1, \mathcal{N}_2 \subset \mathbb{R}^n$ denote two subspaces with dimensions



(a) QKV input (b) Up projection (c) Down projection

Figure 3: Symmetric projection similarity between null spaces estimated using varying numbers of examples and the 4096-sample reference null space across representative Qwen2.5-3B layers.

$k_1 \le k_2$. The principal angles are defined recursively, for $j = 1, \dots, k_1$, as

$$\theta_j = \min_{\substack{\mathbf{x}_j \in \mathcal{N}_1,\, \mathbf{y}_j \in \mathcal{N}_2 \\ \mathbf{x}_j \perp \mathbf{x}_i^*,\, \mathbf{y}_j \perp \mathbf{y}_i^*,\ \forall i<j}} \arccos\left(\frac{\mathbf{x}_j^\top \mathbf{y}_j}{\|\mathbf{x}_j\|\,\|\mathbf{y}_j\|}\right), \qquad j = 1, \dots, k_1, \tag{2}$$

where $(\mathbf{x}_i^*, \mathbf{y}_i^*)$ denotes an optimizing pair for θ_i. Smaller principal angles $\{\theta_1, \dots, \theta_{k_1}\}$ indicate closer alignment between the two null spaces. This recursive definition gives the *geometric characterization* of principal angles. In practice, they can be computed more conveniently from the orthonormal bases of the two subspaces. Following Zhu & Knyazev (2012), let $\mathbf{N}_1$ and $\mathbf{N}_2$ be orthonormal bases of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively:

$$\mathbf{N}_1^\top \mathbf{N}_2 = \mathbf{U}\,\mathrm{diag}(\sigma_1, \dots, \sigma_{k_1})\,\mathbf{V}^\top, \qquad \theta_i = \arccos(\sigma_i), \quad i = 1, \dots, k_1. \tag{3}$$

Appendix A provides a derivation connecting the definition of principal angles to this SVD-based computation.

In Figure 3, we plot the symmetric projection similarity, a normalized sum of the squared cosines of the principal angles between each estimated null space and the 4096-sample reference null space, defined as $\frac{1}{\sqrt{k_1 k_2}} \sum_{i=1}^{k_1} \cos^2 \theta_i$. The similarity increases rapidly and largely stabilizes beyond 512 samples for most layers and modules, indicating that the estimated null-space structure becomes stable with a moderate number of examples.

3.4 Proposed Null-Basis Reparameterization

Section 3.3 demonstrates the feasibility of using the null-space constraint to preserve reasoning capabilities. Building on this result, we construct a fixed null-space basis and use it to reparameterize the low-rank update $\Delta\mathbf{W}$. This reparameterization structurally constrains every update to the estimated approximate null space, thereby limiting changes to the transformations of reasoning hidden states throughout fine-tuning.

Concretely, let $\mathbf{N}_s \in \mathbb{R}^{n \times k}$ denote the orthonormal basis of the null space of $\mathbf{H}$ (i.e., $\mathbf{H}\mathbf{N}_s = \mathbf{0}$), where k is the null-space dimension. We then parameterize the update as

$$\Delta\mathbf{W} = \mathbf{B}\mathbf{C}\mathbf{N}_s^\top, \tag{4}$$

with trainable matrices $\mathbf{B} \in \mathbb{R}^{m \times r}$ and $\mathbf{C} \in \mathbb{R}^{r \times k}$, where r is a chosen adaptation rank. We thus have

$$\mathbf{H}\Delta\mathbf{W}^\top = \mathbf{H}\mathbf{N}_s\mathbf{C}^\top\mathbf{B}^\top = \mathbf{0}, \quad \forall\, \mathbf{B}, \mathbf{C}, \tag{5}$$

which means the preservation constraint is satisfied by construction. We optimize the trainable factors $\mathbf{B}$ and $\mathbf{C}$ on the adaptation dataset $\mathcal{D}_a$ using the standard SFT objective. The overall procedure is summarized in Algorithm 1.

The number of trainable parameters is $mr + rk$ for this reparameterization, which is smaller than the $mr + rn$ of standard LoRA as the null-space dimension $k < n$. As the preservation constraint in (1) is enforced structurally through the parameterization, $\mathbf{B}$ and $\mathbf{C}$ are unconstrained parameters and

Algorithm 1 Null-Basis Low-Rank Adaptation (NB-LoRA)

Require: Post-RL model with target weight matrices $\{\mathbf{W}_\ell\}$; reasoning data $\mathcal{D}_r$; adaptation data $\mathcal{D}_a$; energy threshold ϵ (e.g., 0.99); rank r

1: **Stage 1: Rollout**
2: Roll out the model on a subset of reasoning data $\mathcal{D}_r$ and collect input hidden states $\mathbf{H}_\ell$ for each target layer ℓ across all tokens
3: **Stage 2: Null-basis computation (offline, once)**
4: **for** each target layer ℓ **do**
5: Form $\mathbf{G}_\ell = \mathbf{H}_\ell^\top \mathbf{H}_\ell$ and compute its eigendecomposition
6: Discard the leading eigenvectors covering ϵ of the spectral energy; stack the remaining eigenvectors as $\mathbf{N}_{s,\ell} \in \mathbb{R}^{n \times k_\ell}$
7: **end for**
8: **Stage 3: Constrained adaptation**
9: Freeze $\mathbf{W}_\ell$ and $\mathbf{N}_{s,\ell}$; introduce trainable matrices $\mathbf{B}_\ell \in \mathbb{R}^{m \times r}$ and $\mathbf{C}_\ell \in \mathbb{R}^{r \times k_\ell}$, parameterizing $\Delta\mathbf{W}_\ell = \mathbf{B}_\ell \mathbf{C}_\ell \mathbf{N}_{s,\ell}^\top$
10: Optimize $\{\mathbf{B}_\ell, \mathbf{C}_\ell\}$ on $\mathcal{D}_a$ using the standard SFT objective

Output: Adapted model with $\mathbf{W}_\ell + \mathbf{B}_\ell \mathbf{C}_\ell \mathbf{N}_{s,\ell}^\top$ for each target layer ℓ

can be updated by any standard optimizer (e.g., SGD (Bottou, 2010), Adam (Kingma & Ba, 2015)) without requiring any projection or constraint enforcement at each iteration. In addition, since $\mathbf{N}_s$ has orthonormal columns, composing through it preserves the gradient Lipschitz constant of the loss, and our method enjoys the same $O(1/\log(T))$ convergence rate to a stationary point as standard LoRA under the same smoothness assumptions, where T denotes the number of optimization steps (Mu & Klabjan, 2026). We provide a formal statement in Appendix B.

Remark 3.1. Mathematically, one may simplify (4) into $\Delta\mathbf{W} = \widetilde{\mathbf{B}}\mathbf{N}_s^\top$ with $\widetilde{\mathbf{B}} \in \mathbb{R}^{m \times k}$. We keep the factorization $\mathbf{B}\mathbf{C}\mathbf{N}_s^\top$ because it provides a tunable bottleneck through rank r, which can substantially reduce trainable parameters when k is large. This reparameterization can be interpreted as a constrained low-rank update. Standard LoRA sets $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$ with no restriction on $\mathbf{A}$. In contrast, our method sets $\mathbf{A} = \mathbf{C}\mathbf{N}_s^\top$, so that the row space of $\mathbf{A}$ is restricted to the null space of the reasoning-state matrix. The method therefore retains the parameter-efficiency benefits of low-rank adaptation while structurally limiting changes to layer outputs on reasoning-related hidden states.

4 Experiments

We consider two reasoning models, Qwen2.5-3B and Qwen2.5-1.5B, which are trained via GRPO on MATH-lighteval (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021), respectively. In SFT stage, we adapt these models to a tool-use task (Tang et al., 2023) and eight target tasks from the Commonsense Reasoning benchmark (Hu et al., 2023). After fine-tuning on the target tasks, we evaluate reasoning retention on both the anchor benchmark and several held-out benchmarks whose data are not used to construct the preservation constraint, including MATH-500, AGIEval-EN-MATH (Zhong et al., 2024), and MMLU-STEM (Hendrycks et al., 2020). More details about the hyperparameters and datasets are provided in Appendix D.

Baselines. We compare our method against vanilla LoRA and the following baselines.

- LoRA-Null (Tang et al., 2026): A null-space-based initialization baseline. For each target layer, it uses the null basis of the hidden state matrix to decompose the backbone weight into a frozen residual component and a null-space-projected component, then initializes the LoRA factors from the projected component.
- MiLoRA (Wang et al., 2025): Another SVD-based LoRA baseline that decomposes each pre-trained weight into principal and minor components, freezes the principal component in the base model, and initializes the LoRA from the minor singular components.
- NSCL (Wang et al., 2021): A null-space-based projection method that trains the full model, projecting each layer's full gradient onto the approximate null space of the previous task's hidden states before each update.

Table 2: **Adaptation on new target tasks and retention on the reasoning benchmark.** Reasoning is evaluated on the benchmark used for RL post-training: MATH-lighteval for Qwen2.5-3B and GSM8K for Qwen2.5-1.5B. Our proposed method matches Vanilla LoRA on target task accuracy while retaining the reasoning performance.

Method	ARC-C	ARC-E	BoolQ	HellaSwag	OpenBookQA	PIQA	SocialIQA	WinoGrande	ToolUse	Avg.
Target					**Qwen2.5-3B**					
Vanilla LoRA	0.824	**0.948**	0.715	**0.947**	0.868	0.861	**0.814**	**0.857**	**0.790**	**0.847**
LoRA-Null	0.823	0.924	0.696	0.944	0.854	0.875	0.810	0.838	0.758	0.836
MiLoRA	0.773	0.926	0.720	0.946	**0.884**	**0.878**	0.754	0.839	0.710	0.826
NSCL	0.794	0.937	0.706	0.945	0.880	0.870	0.812	0.830	0.774	0.839
Exp. Replay	0.815	0.935	**0.722**	0.944	**0.884**	0.868	0.810	0.849	0.774	0.845
SDFT	0.643	0.856	0.601	0.702	0.728	0.730	0.681	0.698	0.645	0.698
Ours	**0.826**	0.941	0.713	0.945	0.882	0.873	0.811	0.847	0.774	0.846
Reasoning					**Qwen2.5-3B** (Pre-SFT: 63.8%; MATH-lighteval)					
Vanilla LoRA	0.089	0.060	0.013	0.245	0.064	0.000	0.162	0.033	0.582	0.139±0.183
LoRA-Null	0.387	0.477	0.622	0.577	0.493	0.556	0.624	0.612	0.604	0.550±0.082
MiLoRA	0.160	0.103	0.357	0.000	0.102	0.204	0.133	0.095	0.272	0.158±0.106
NSCL	**0.637**	0.635	**0.637**	0.635	0.635	**0.638**	0.638	**0.631**	0.614	**0.633**±0.008
Exp. Replay	0.618	0.602	0.625	0.592	0.613	0.556	0.632	0.622	**0.620**	0.609±0.023
SDFT	0.635	0.634	0.633	0.633	**0.637**	0.634	**0.639**	**0.631**	0.612	0.632±0.008
Ours	0.635	**0.638**	0.634	**0.637**	0.629	0.636	0.633	**0.631**	0.617	0.632±0.006
Target					**Qwen2.5-1.5B**					
Vanilla LoRA	0.753	0.906	0.686	**0.918**	0.842	0.852	0.777	0.785	**0.661**	**0.798**
LoRA-Null	0.734	0.900	**0.692**	0.913	0.854	**0.854**	**0.783**	0.774	0.177	0.742
MiLoRA	0.753	0.908	0.690	0.913	0.836	0.845	0.780	**0.789**	0.000	0.724
NSCL	0.695	0.893	0.689	0.908	**0.856**	0.843	0.774	0.765	0.065	0.721
Exp. Replay	0.758	0.902	0.689	0.914	0.844	0.836	0.753	0.783	0.000	0.720
SDFT	0.066	0.103	0.065	0.021	0.058	0.403	0.014	0.183	0.161	0.119
Ours	**0.771**	**0.912**	0.683	0.914	0.846	0.843	0.767	0.763	0.629	0.792
Reasoning					**Qwen2.5-1.5B** (Pre-SFT: 80.1%; GSM8K)					
Vanilla LoRA	0.162	0.020	0.304	0.017	0.092	0.021	0.054	0.081	0.750	0.167±0.237
LoRA-Null	0.435	0.657	0.666	0.718	0.606	0.661	0.544	0.608	0.508	0.600±0.090
MiLoRA	0.406	0.575	0.497	0.561	0.663	0.286	0.547	0.514	0.531	0.509±0.108
NSCL	**0.801**	0.794	0.769	0.773	0.801	0.782	0.793	0.794	0.788	0.788±0.011
Exp. Replay	0.775	0.775	0.648	0.654	0.643	0.607	0.388	0.756	0.262	0.612±0.177
SDFT	0.799	0.792	0.801	0.798	**0.811**	**0.801**	**0.795**	0.792	0.799	0.799±0.006
Ours	0.799	**0.800**	**0.804**	**0.801**	0.797	0.795	0.792	**0.809**	**0.803**	**0.800**±0.005

- Experience Replay (Rolnick et al., 2019): It employs standard LoRA and trains the LoRA modules on each task's SFT data mixed with a small replay buffer of reasoning traces.
- SDFT (Shenfeld et al., 2026): Conditions the model on downstream demonstrations to generate on-policy rollouts, which are subsequently used to fine-tune the model to reduce forgetting.

4.1 Main Results

Adaptation and reasoning retention. Table 2 reports target-task accuracy alongside reasoning accuracy on the benchmark used for RL post-training. Vanilla LoRA achieves strong target-task accuracy, but retained reasoning accuracy, averaged across all nine adaptation tasks, drops from 63.8% to 13.9% on Qwen2.5-3B and from 80.1% to 16.7% on Qwen2.5-1.5B. LoRA-Null and MiLoRA mitigate this collapse, yet neither closes the gap. SDFT largely preserves reasoning performance but struggles to learn the target tasks, especially on Qwen2.5-1.5B. NB-LoRA, by contrast, holds reasoning accuracy at essentially its pre-fine-tuning level on both models while matching Vanilla LoRA's target accuracy, and it does so uniformly: its per-task variation is among the smallest of these methods, whereas Experience Replay loses more than forty points on individual tasks. NSCL is the only baseline that achieves comparable reasoning retention without severely compromising target-task performance. However, NSCL updates the full model and projects the gradients at every optimization step, whereas NB-LoRA encodes the constraint directly in the adapter parameterization, requiring no gradient projection while keeping the backbone frozen.

Generalization of reasoning retention. Table 3 examines whether reasoning preservation generalizes beyond the anchor corpus used to construct the null basis. Although the constraint is estimated

Table 3: **Generalization of reasoning retention.** Accuracy on five reasoning benchmarks before and after downstream fine-tuning. **Post-RL** is the reference checkpoint before SFT. For each model, the preservation constraint is constructed from a single anchor corpus: MATH-lighteval for Qwen2.5-3B and GSM8K for Qwen2.5-1.5B. Performance on the remaining four benchmarks measures whether reasoning preservation generalizes beyond the anchor corpus.

Benchmarks	Before SFT	After SFT						
	Post-RL	Vanilla LoRA	LoRA-Null	MiLoRA	NSCL	Exp. Replay	SDFT	Ours
Qwen2.5-3B								
MATH-lighteval	0.638	0.089	0.387	0.160	**0.637**	0.618	0.632	0.635
MATH-500	0.624	0.052	0.428	0.150	0.622	**0.626**	0.608	0.622
GSM8K	0.842	0.000	0.034	0.007	**0.844**	0.562	0.784	0.843
AGIEval-EN-MATH	0.620	0.051	0.371	0.156	0.611	0.603	0.607	**0.621**
MMLU-STEM	0.659	0.420	0.487	0.528	**0.636**	0.476	0.611	**0.636**
Average	0.677	0.122	0.341	0.200	0.670	0.577	0.648	**0.671**
Qwen2.5-1.5B								
MATH-lighteval	0.462	0.082	0.131	0.083	0.455	0.388	0.448	**0.467**
MATH-500	0.468	0.068	0.120	0.086	0.462	0.396	0.460	**0.468**
GSM8K	0.801	0.162	0.435	0.406	0.798	0.775	0.793	**0.801**
AGIEval-EN-MATH	0.445	0.071	0.127	0.087	0.451	0.374	0.432	**0.453**
MMLU-STEM	0.547	0.451	0.461	0.456	**0.547**	0.522	0.542	**0.547**
Average	0.545	0.167	0.255	0.224	0.543	0.491	0.535	**0.547**

solely from MATH-lighteval for Qwen2.5-3B and GSM8K for Qwen2.5-1.5B, NB-LoRA preserves reasoning performance across all five benchmarks, including the four unseen during constraint construction. Its average accuracy remains close to the Post-RL reference: 67.1% versus 67.7% on Qwen2.5-3B and 54.7% versus 54.5% on Qwen2.5-1.5B. These results suggest that the null space estimated from one reasoning corpus protects representations shared across reasoning benchmarks. Among the baselines, only NSCL provides comparable cross-benchmark retention, while NB-LoRA offers substantially greater efficiency, as discussed next.

4.2 Efficiency Comparison

Among the compared methods, NSCL and NB-LoRA achieve similar adaptation performance and reasoning retention, but differ substantially in training efficiency. As shown in Figure 4, NB-LoRA consistently reduces runtime, memory usage, and the number of trainable parameters across both model scales. On Qwen2.5-3B, it is $6.3\times$ faster and uses $2.1\times$ less peak CUDA memory than NSCL. The gap is structural: NSCL must keep full-model optimizer states together with a per-layer projector of size $O(d^2)$, while applying the projection incurs $O(d^3)$ cost per layer per optimizer step for hidden dimension d. In contrast, NB-LoRA enforces the null-space constraint directly through low-rank reparameterization, avoiding repeated projection during training. As model size increases, this structural difference leads to a widening efficiency gap, indicating better scalability of NB-LoRA to larger LLMs.

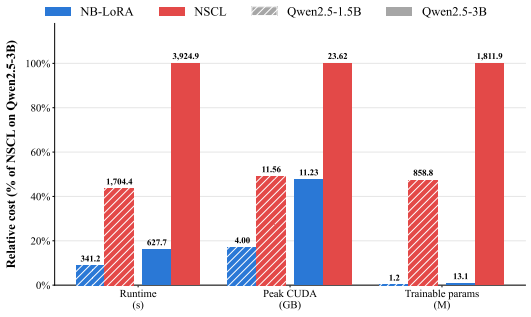
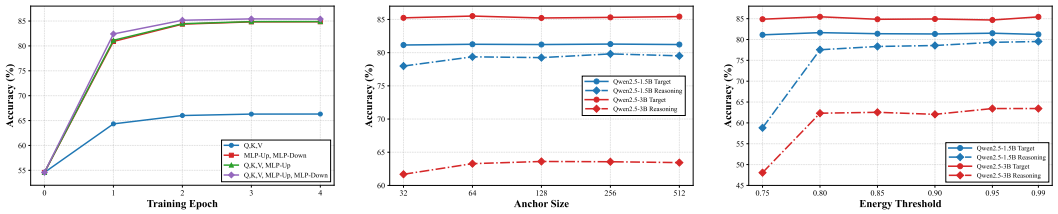


Figure 4: Training efficiency comparison between NB-LoRA and NSCL (3 epochs, batch size 1). Bars show runtime, peak CUDA memory, and trainable parameters relative to NSCL.

4.3 Ablation Study

Module selection. Figure 5a compares four choices of tunable modules: only the attention Q/K/V projections, only the MLP up/down projections (i.e., the feed-forward module), Q/K/V plus the



(a) Target-task convergence for different tunable modules on Qwen2.5-3B.

(b) Effect of the number of reasoning examples used to estimate the null basis.

(c) Effect of the spectral-energy threshold used to define the approximate null space.

Figure 5: Ablations of the design choices of the proposed NB-LoRA. Target denotes average downstream accuracy over the eight commonsense reasoning adaptation tasks. Reason denotes average retained reasoning accuracy across the eight adapted checkpoints, evaluated on GSM8K for Qwen2.5-1.5B and MATH-lighteval for Qwen2.5-3B.

MLP up projection, and all five projections. Restricting NB-LoRA to Q/K/V substantially underfits the target tasks, whereas including MLP projections closes most of the gap. Tuning MLP up/down or Q/K/V+MLP-up achieves nearly the same target accuracy as tuning all five projections, which performs best overall. These results also reveal lower-cost alternatives to the full module set: Q/K/V+MLP-up avoids the higher-dimensional null basis required for MLP-down, while MLP-only tuning adapts fewer modules.

Anchor data size. Figure 5b varies the number of reasoning examples used to estimate the null basis. Target-task accuracy remains nearly unchanged from 32 to 512 examples for both models, while reasoning retention is stable from 64 to 512 examples. Performance degrades mainly at 32 examples, particularly for Qwen2.5-3B, likely because the resulting null-space estimate is noisier. Together with the spectral analysis in Section 3.3, these results indicate that a moderate number of reasoning examples is sufficient to capture a stable null space, an important property for the practical use of NB-LoRA.

Energy threshold. Figure 5c varies the spectral-energy threshold used to define the approximate null space. Across those thresholds, target-task accuracy remains stable for both models, which can be attributed to the null space being large enough to accommodate adaptation even under stricter preservation as we showed in Section 3.3. Reasoning retention is robust for thresholds from 0.80 to 0.99, but drops sharply at 0.75. This suggests that NB-LoRA is insensitive to the threshold over a broad range, with degradation occurring only when the constraint becomes too permissive and more reasoning directions enter the trainable subspace. Such robustness provides flexibility in threshold selection, with higher thresholds favoring stronger preservation and moderately lower thresholds potentially allowing a larger trainable subspace when needed.

Detailed per-task results for the module-selection, anchor-size, and energy-threshold ablations are provided in Appendices C.1.1 to C.1.3, respectively.

5 Conclusion

We study post-RL adaptation: how to efficiently learn new tasks without erasing reasoning abilities acquired through RL. We find that standard LoRA can substantially degrade these abilities. Our analysis shows that reasoning activations concentrate in low-dimensional subspaces, leaving substantial null-space capacity for adaptation. This observation motivates Null-Basis LoRA (NB-LoRA), which constrains low-rank updates to an approximate null space of reasoning activations. The constraint is maintained throughout fine-tuning without replay or per-step gradient projection. We further show that NB-LoRA preserves the relevant smoothness constants and inherits standard LoRA's $O(1/\log(T))$ convergence guarantee under the same assumptions. Across two RL-trained models and diverse downstream tasks, NB-LoRA matches standard LoRA's adaptation performance while retaining reasoning accuracy near its pre-fine-tuning level. Its preservation generalizes to unseen reasoning benchmarks, and its training is more computationally and memory efficient than full-model null-space projection. Overall, our results demonstrate that NB-LoRA provides an efficient approach for adapting post-RL models to new tasks while preserving their reasoning abilities.

AI USE STATEMENT

We used generative AI tools to assist in drafting and revising manuscript text, improve wording and readability, check for typographical errors and notation inconsistencies, refine table presentation, and review code implementations. The research ideas and proposed methodology originated with the authors. Generative AI was not used to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical proofs; propose or refine hypotheses; design or provide feedback on the research methodology or experiments; implement methods; assist with translation; clean or reformat datasets; support qualitative or thematic data analysis; or interpret results. The authors reviewed all AI-assisted work and take responsibility for the final content, including its text, claims, code, and other artifacts.

REPRODUCIBILITY STATEMENT

We provide the procedure for constructing the approximate null bases and training NB-LoRA in Algorithm 1. Appendix D describes the experimental hardware and datasets, with training and null-space construction hyperparameters in Table 8 and prompt templates in Tables 9 and 10. Appendix C.1 reports detailed per-task ablation results. Code is provided in the supplementary materials.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Wenzhi Fang, Dong-Jun Han, Liangqi Yuan, Seyyedali Hosseinalipour, and Christopher G Brinton. Federated sketching lora: A flexible framework for heterogeneous collaborative fine-tuning of llms. *arXiv preprint arXiv:2501.19389*, 2025.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773. PMLR, 2020. URL https://proceedings.mlr.press/v108/farajtabar20a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

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 *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 5254–5276, 2023.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Wei Bi, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations (ICLR)*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://doi.org/10.1073/pnas.1611835114.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Bin Cui, and Wentao Zhang. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *The Fourteenth International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2506.07527. URL https://arxiv.org/abs/2506.07527.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

Siqiao Mu and Diego Klabjan. On the convergence rate of LoRA gradient descent. In *International Conference on Machine Learning*, 2026.

Xueyan Niu, Bo Bai, Wei Han, and Weixi Zhang. On the non-decoupling of supervised fine-tuning and reinforcement learning in post-training. *arXiv preprint arXiv:2601.07389*, 2026. doi: 10.48550/arXiv.2601.07389. URL https://arxiv.org/abs/2601.07389.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: Parameter efficient continual learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3092. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b0bc711f48724237b38823c4d9cee10b-Abstract-Conference.html.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single LoRA continual learning via continual merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=i1Rj7yU6eF.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. doi: 10.48550/arXiv.2601.19897. URL https://arxiv.org/abs/2601.19897.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: Language modeling for lifelong language learning. In *The Eighth International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Skgxcn4YDS.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Llm reasoning as trajectories: Step-specific representation geometry and correctness signals. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26872–26887, 2026.

Pengwei Tang, Xiaolin Hu, Yong Liu, Lizhong Ding, Dongjie Zhang, Xing Wu, and Debing Zhang. Put the space of lora initialization to the extreme to preserve pre-trained knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33232–33240, 2026.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. ReFT: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7601–7614. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.410. URL https://aclanthology.org/2024.acl-long.410/.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4823–4836, 2025.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 184–193, 2021.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the association for computational linguistics: NAACL 2024*, pp. 2299–2314, 2024.

Peizhen Zhu and Andrew V Knyazev. Principal angles between subspaces and their tangents. *arXiv preprint arXiv:1209.0523v1*, 2012. URL https://arxiv.org/abs/1209.0523v1.

Appendix

A Derivation of the SVD-Based Principal Angle Computation **15**

B Convergence Analysis of the Proposed Algorithm **15**

C Further Experiments **17**

C.1 Detailed Ablation Results 17

C.1.1 Per-Task Results of the Module-Selection Ablation 17

C.1.2 Per-Task Results of the Anchor-Size Ablation 17

C.1.3 Per-Task Results of the Energy-Threshold Ablation 18

C.2 Null-Space Capacity for Sequential Multi-Task Adaptation 19

D Experiment Details **21**

D.1 Hyperparameter Details 21

D.2 Dataset Details 21

D.2.1 Commonsense Reasoning Benchmarks 23

D.2.2 Tool-Use Tasks 23

D.2.3 MMLU-STEM Benchmark 24

E Qualitative Case Study: Reasoning Retention and Downstream Adaptation **24**

E.1 MATH Reasoning Examples 24

E.2 HellaSwag Examples 26

E.3 Summary 27

A Derivation of the SVD-Based Principal Angle Computation

For completeness, we derive the relationship between principal angles and the singular values of the cross-basis matrix (Horn & Johnson, 2012; Zhu & Knyazev, 2012).

Let $\mathcal{N}_1, \mathcal{N}_2 \subset \mathbb{R}^n$ be two subspaces with dimensions k_1 and k_2, where $k_1 \leq k_2$, and let $\mathbf{N}_1 \in \mathbb{R}^{n\times k_1}$ and $\mathbf{N}_2 \in \mathbb{R}^{n\times k_2}$ be orthonormal bases of these subspaces. We show that the recursive definition of principal angles yields

$$\cos\theta_i = \sigma_i(\mathbf{N}_1^\top \mathbf{N}_2), \qquad i = 1, \ldots, k_1. \tag{6}$$

For the first principal angle, any vectors $\mathbf{x} \in \mathcal{N}_1$ and $\mathbf{y} \in \mathcal{N}_2$ can be written as

$$\mathbf{x} = \mathbf{N}_1\mathbf{a}, \qquad \mathbf{y} = \mathbf{N}_2\mathbf{b}, \tag{7}$$

for coefficient vectors $\mathbf{a} \in \mathbb{R}^{k_1}$ and $\mathbf{b} \in \mathbb{R}^{k_2}$. Since $\mathbf{N}_1$ and $\mathbf{N}_2$ have orthonormal columns, we have

$$\|\mathbf{x}\|_2 = \|\mathbf{a}\|_2, \qquad \|\mathbf{y}\|_2 = \|\mathbf{b}\|_2. \tag{8}$$

Therefore, the first principal angle satisfies

$$\cos\theta_1 = \max_{\substack{\mathbf{x}\in\mathcal{N}_1,\ \mathbf{y}\in\mathcal{N}_2 \\ \|\mathbf{x}\|_2=\|\mathbf{y}\|_2=1}} \mathbf{x}^\top\mathbf{y} = \max_{\substack{\mathbf{a}\in\mathbb{R}^{k_1},\ \mathbf{b}\in\mathbb{R}^{k_2} \\ \|\mathbf{a}\|_2=\|\mathbf{b}\|_2=1}} \mathbf{a}^\top\mathbf{N}_1^\top\mathbf{N}_2\mathbf{b}. \tag{9}$$

By the variational characterization of the singular values, this maximum is exactly the largest singular value of $\mathbf{N}_1^\top\mathbf{N}_2$. Hence,

$$\cos\theta_1 = \sigma_1(\mathbf{N}_1^\top\mathbf{N}_2). \tag{10}$$

Now consider the singular value decomposition

$$\mathbf{N}_1^\top\mathbf{N}_2 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{11}$$

where $\mathbf{U} \in \mathbb{R}^{k_1\times k_1}$ and $\mathbf{V} \in \mathbb{R}^{k_2\times k_2}$ are orthogonal, and

$$\mathbf{\Sigma} = [\mathrm{diag}(\sigma_1, \ldots, \sigma_{k_1}) \quad \mathbf{0}]. \tag{12}$$

Let $\mathbf{u}_i$ and $\mathbf{v}_i$ denote the left and right singular vectors associated with σ_i, and define

$$\mathbf{x}_i = \mathbf{N}_1\mathbf{u}_i, \qquad \mathbf{y}_i = \mathbf{N}_2\mathbf{v}_i. \tag{13}$$

Because $\mathbf{N}_1$ and $\mathbf{N}_2$ have orthonormal columns and the singular vectors are orthonormal, the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{k_1}$ are orthonormal in $\mathcal{N}_1$, and $\mathbf{y}_1, \ldots, \mathbf{y}_{k_1}$ are orthonormal in $\mathcal{N}_2$. Moreover,

$$\mathbf{x}_i^\top\mathbf{y}_j = \mathbf{u}_i^\top\mathbf{N}_1^\top\mathbf{N}_2\mathbf{v}_j = \mathbf{u}_i^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{v}_j = \sigma_j\mathbf{u}_i^\top\mathbf{u}_j. \tag{14}$$

Thus, $\mathbf{x}_i^\top\mathbf{y}_i = \sigma_i$, and for $i \neq j$ these pairs satisfy the orthogonality constraints required by the recursive definition of principal angles. Applying the same maximization argument on the orthogonal complements after removing the first $i-1$ principal vector pairs yields

$$\cos\theta_i = \sigma_i, \qquad i = 1, \ldots, k_1. \tag{15}$$

Therefore, the principal angles between $\mathcal{N}_1$ and $\mathcal{N}_2$ are given by

$$\theta_i = \arccos\sigma_i(\mathbf{N}_1^\top\mathbf{N}_2), \qquad i = 1, \ldots, k_1. \tag{16}$$

B Convergence Analysis of the Proposed Algorithm

This appendix shows that the null-space reparameterization does not weaken the convergence guarantee for standard LoRA gradient descent established by Mu & Klabjan (2026). The key observation is that right-multiplication by $\mathbf{N}_s^\top$, where $\mathbf{N}_s$ has orthonormal columns, is an isometry under the Frobenius norm. Consequently, composing the training loss with this map preserves the smoothness and lower-bound assumptions required by their analysis. We formalize this observation in a bridging lemma and then apply their result to NB-LoRA.

Let $\mathbf{W} \in \mathbb{R}^{m\times n}$ be a frozen backbone weight and let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ denote the SFT loss on $\mathcal{D}_a$ as a function of its update $\Delta\mathbf{W}$, with the remaining model parameters fixed. Standard LoRA parameterizes this update as $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$, whereas NB-LoRA minimizes

$$g(\mathbf{B}, \mathbf{C}) := f\big(\mathbf{B}\mathbf{C}\mathbf{N}_s^\top\big) \quad \text{over} \quad \mathbf{B} \in \mathbb{R}^{m\times r},\ \mathbf{C} \in \mathbb{R}^{r\times k}, \tag{17}$$

where $\mathbf{N}_s \in \mathbb{R}^{n\times k}$ is the fixed null basis of (4). Since $\mathbf{N}_s$ is an orthonormal basis of the null space of $\mathbf{H}$, we have $\mathbf{N}_s^\top \mathbf{N}_s = \mathbf{I}_k$. We analyze the gradient descent updates

$$\mathbf{B}_{t+1} = \mathbf{B}_t - \eta_t \nabla_{\mathbf{B}}\, g(\mathbf{B}_t, \mathbf{C}_t), \qquad \mathbf{C}_{t+1} = \mathbf{C}_t - \eta_t \nabla_{\mathbf{C}}\, g(\mathbf{B}_t, \mathbf{C}_t), \tag{18}$$

where η_t is the learning rate at step t.

Following standard convergence analyses of gradient descent methods (Bottou et al., 2018; Bubeck, 2015; Mu & Klabjan, 2026), we assume that the loss f is differentiable and satisfies the standard smoothness and lower-boundedness conditions. Note that both are stated for the loss as a function of the *full* weight update: no smoothness of the reparameterized objective g and no boundedness of the trainable factors is assumed.

Assumption B.1 (L-smoothness)**.** The loss f is differentiable, and there exists a constant $L \geq 1$ such that for all $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{m\times n}$,

$$\big\|\nabla f(\mathbf{X}) - \nabla f(\mathbf{Y})\big\|_F \leq L\big\|\mathbf{X} - \mathbf{Y}\big\|_F. \tag{19}$$

Assumption B.2 (Lower boundedness)**.** There exists f^* such that $f(\mathbf{X}) \geq f^*$ for all $\mathbf{X} \in \mathbb{R}^{m\times n}$.

The following lemma bridges the two settings by showing that composition with the fixed basis $\mathbf{N}_s$ preserves both assumptions with the same constants.

Lemma B.3 (Null-basis composition preserves smoothness)**.** Define $\tilde{f} : \mathbb{R}^{m\times k} \to \mathbb{R}$ by $\tilde{f}(\mathbf{M}) := f\big(\mathbf{M}\mathbf{N}_s^\top\big)$. Under Assumptions B.1 and B.2, $\tilde{f}$ is L-smooth with the same constant L and is lower bounded by the same constant f^*.

Proof. By the chain rule, $\nabla \tilde{f}(\mathbf{M}) = \nabla f\big(\mathbf{M}\mathbf{N}_s^\top\big)\mathbf{N}_s$. Since $\mathbf{N}_s^\top \mathbf{N}_s = \mathbf{I}_k$, right-multiplication by $\mathbf{N}_s^\top$ is a Frobenius isometry, $\|\mathbf{X}\mathbf{N}_s^\top\|_F^2 = \mathrm{Tr}(\mathbf{X}\mathbf{N}_s^\top \mathbf{N}_s \mathbf{X}^\top) = \|\mathbf{X}\|_F^2$ for all $\mathbf{X} \in \mathbb{R}^{m\times k}$, and the spectral norm of $\mathbf{N}_s$ equals one. Hence, for any $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{m\times k}$,

$$\begin{aligned}
\big\|\nabla \tilde{f}(\mathbf{M}_1) - \nabla \tilde{f}(\mathbf{M}_2)\big\|_F &= \Big\|\big[\nabla f\big(\mathbf{M}_1\mathbf{N}_s^\top\big) - \nabla f\big(\mathbf{M}_2\mathbf{N}_s^\top\big)\big]\mathbf{N}_s\Big\|_F \\
&\leq \big\|\nabla f\big(\mathbf{M}_1\mathbf{N}_s^\top\big) - \nabla f\big(\mathbf{M}_2\mathbf{N}_s^\top\big)\big\|_F \\
&\leq L\big\|(\mathbf{M}_1 - \mathbf{M}_2)\mathbf{N}_s^\top\big\|_F = L\big\|\mathbf{M}_1 - \mathbf{M}_2\big\|_F,
\end{aligned} \tag{20}$$

where the first inequality uses $\|\mathbf{Y}\mathbf{N}_s\|_F \leq \|\mathbf{Y}\|_F \|\mathbf{N}_s\|_2 = \|\mathbf{Y}\|_F$, the second uses Assumption B.1, and the final equality uses the isometry property. Lower boundedness is immediate: $\tilde{f}(\mathbf{M}) = f(\mathbf{M}\mathbf{N}_s^\top) \geq f^*$. □

By (17), $g(\mathbf{B}, \mathbf{C}) = \tilde{f}(\mathbf{B}\mathbf{C})$, so the updates (18) are precisely the original LoRA gradient descent algorithm applied to the loss $\tilde{f}$, with input dimension k in place of n. By Lemma B.3, $\tilde{f}$ satisfies the assumptions of Mu & Klabjan (2026) with unchanged constants, their main convergence result thus applies directly to NB-LoRA.

Theorem B.4 (Convergence of NB-LoRA; Theorem 3.5 of Mu & Klabjan, 2026)**.** Suppose Assumptions B.1 and B.2 hold. Run T steps of the updates (18) with the learning rate

$$\eta_t = \min\left\{\frac{1}{4\sqrt{2}\,L\big(\|\mathbf{V}_t\|_F^2 + \|\nabla \tilde{f}(\mathbf{B}_t\mathbf{C}_t)\|_F\big)},\, 1\right\}, \qquad \mathbf{V}_t = \begin{bmatrix}\mathbf{B}_t \\ \mathbf{C}_t^\top\end{bmatrix} \in \mathbb{R}^{(m+k)\times r}. \tag{21}$$

Then

$$\min_{t=0,\ldots,T-1} \big\|\nabla g(\mathbf{B}_t, \mathbf{C}_t)\big\|_F^2 = O\left(\frac{1}{\log T}\right), \tag{22}$$

where the $O(\cdot)$ notation hides dependence on $\mathbf{V}_0$, $g(\mathbf{B}_0, \mathbf{C}_0)$, f^*, and L. Moreover, if there exists a constant $C_V > 0$ such that $\|\mathbf{V}_t\|_F \leq C_V$ for all t, the rate improves to $O(1/T)$.

Remark B.5 (Same guarantee as standard LoRA)**.** Lemma B.3 shows that the smoothness constant is preserved *exactly*: right-composition with $\mathbf{N}_s^\top$ is an isometric embedding of $\mathbb{R}^{m\times k}$ into $\mathbb{R}^{m\times n}$. Consequently, NB-LoRA satisfies the same assumptions and inherits the same convergence rate as standard LoRA. Overall, the null-space reparameterization reduces the input dimension from n to k without weakening the convergence guarantee.

Table 4: **Per-task results of the module-selection ablation in Figure 5a.** Each column group corresponds to one choice of tunable modules: Q, K, and V are the query, key, and value projections of the attention layers, and MLP-Up/MLP-Down are the up- and down-projections of the feed-forward block. Target and Reason denote target-task and reasoning accuracy (%) at the post-SFT checkpoint.

Task	Q,K,V		Q,K,V, MLP-Up		MLP-Up, MLP-Down		Q,K,V, MLP-Up, MLP-Down	
	Target	Reason	Target	Reason	Target	Reason	Target	Reason
Qwen2.5-1.5B (Reasoning task: GSM8K)								
ARC-Challenge	45.90	79.91	72.35	79.98	74.74	80.14	**77.05**	79.91
ARC-Easy	79.76	79.30	89.02	79.91	90.45	78.70	**91.20**	79.98
BoolQ	65.60	79.61	69.60	79.98	**69.88**	79.98	68.31	79.53
HellaSwag	90.26	79.61	**91.58**	78.62	91.23	79.45	91.37	79.83
OpenBookQA	68.60	79.68	84.20	79.23	**85.00**	79.53	84.60	78.62
PIQA	82.59	78.77	83.41	79.76	83.46	79.23	**84.30**	79.53
SocialIQA	**77.43**	79.61	77.28	78.77	76.87	78.70	76.73	78.77
WinoGrande	73.40	79.53	**78.22**	78.70	75.37	79.38	76.19	79.98
Average	72.94	79.50	80.71	79.37	80.88	79.39	**81.22**	79.52
Qwen2.5-3B (Reasoning task: MATH-lighteval)								
ARC-Challenge	54.35	63.28	79.22	63.32	81.83	63.64	**82.56**	63.50
ARC-Easy	44.87	63.28	92.34	63.74	93.39	63.60	**93.69**	63.16
BoolQ	66.06	63.80	70.18	63.68	71.04	63.34	**71.28**	63.40
HellaSwag	93.24	63.00	**94.78**	63.56	94.60	63.10	94.24	63.54
OpenBookQA	46.80	63.46	**89.00**	63.38	87.40	62.90	88.80	63.88
PIQA	76.28	63.18	87.21	63.04	**87.65**	63.36	87.16	63.60
SocialIQA	67.96	63.70	**81.78**	64.36	79.99	63.02	81.11	63.32
WinoGrande	80.82	63.44	84.53	63.26	82.72	63.34	**84.61**	63.10
Average	66.30	63.39	84.88	63.54	84.83	63.29	**85.43**	63.44

C Further Experiments

C.1 Detailed Ablation Results

This appendix provides detailed per-task results for the ablation studies in Figure 5. Target denotes downstream-task accuracy at the post-SFT checkpoint, while Reason denotes reasoning accuracy on GSM8K for Qwen2.5-1.5B and MATH-lighteval for Qwen2.5-3B. All values are percentages.

C.1.1 Per-Task Results of the Module-Selection Ablation

Table 4 reports the per-task results underlying the module-selection ablation in Figure 5a. The detailed breakdown reinforces the main-text observation that target-task adaptation relies strongly on the MLP projections. Tuning only Q/K/V underfits severely on several tasks (e.g., on Qwen2.5-3B, ARC-Easy drops from 93.69% under the full module set to 44.87%, and OpenBookQA from 88.80% to 46.80%), whereas configurations including MLP projections recover most of the performance of the full module set. In particular, MLP-only and Q/K/V+MLP-Up achieve average target accuracy within one point of the full configuration for both models, supporting their use as lower-cost alternatives when reducing the number or dimensionality of adapted modules is desirable. Meanwhile, reasoning accuracy remains stable across the four module choices, indicating that the preservation behavior of NB-LoRA is largely insensitive to which modules are adapted.

C.1.2 Per-Task Results of the Anchor-Size Ablation

Table 5 reports the per-task results underlying the anchor-size ablation in Figure 5b, where the number of reasoning examples used to estimate the hidden-state null basis varies from 32 to 512. Consistent with Figure 5b, target-task accuracy is largely insensitive to anchor size: the average target accuracy varies by at most 0.3 points for both models, and no individual task exhibits a systematic

Table 5: **Per-task results of the anchor-size ablation in Figure 5b.** Each column group corresponds to the number of reasoning examples used to estimate the null basis. Target and Reason denote target-task and reasoning accuracy (%) at the post-SFT checkpoint. Target accuracy remains nearly unchanged across anchor sizes, while reasoning retention is stable for 64 or more examples and degrades mainly at the smallest anchor set.

Task	32		64		128		256		512	
	Target	**Reason**	**Target**	**Reason**	**Target**	**Reason**	**Target**	**Reason**	**Target**	**Reason**
Qwen2.5-1.5B (Reasoning task: GSM8K)										
ARC-Challenge	75.60	77.68	75.43	79.61	76.45	79.83	76.93	80.97	77.05	79.91
ARC-Easy	90.49	78.21	90.45	79.68	90.82	79.76	89.98	79.76	91.20	79.98
BoolQ	68.62	77.00	69.24	79.08	69.69	79.45	68.29	79.30	68.31	79.53
HellaSwag	91.26	78.32	91.47	79.08	91.10	78.39	91.22	80.67	91.37	79.83
OpenBookQA	85.20	78.29	84.60	79.98	84.00	79.76	86.80	79.61	84.60	78.62
PIQA	83.03	77.61	84.77	78.92	84.06	78.77	84.11	79.68	84.30	79.53
SocialIQA	76.71	78.85	76.87	79.98	77.33	78.77	76.92	79.98	76.73	78.77
WinoGrande	78.30	77.71	77.27	78.62	76.32	79.23	76.09	78.47	76.19	79.98
Average	81.15	77.96	81.26	79.37	81.22	79.25	81.29	79.81	81.22	79.52
Qwen2.5-3B (Reasoning task: MATH-lighteval)										
ARC-Challenge	82.85	63.84	82.60	63.82	82.60	63.52	82.53	63.56	82.56	63.50
ARC-Easy	93.73	63.72	93.52	63.38	94.28	63.18	93.50	64.26	93.69	63.16
BoolQ	71.07	63.72	71.40	63.78	71.39	63.96	71.41	63.70	71.28	63.40
HellaSwag	94.04	63.96	94.65	63.12	94.46	63.58	94.29	63.76	94.24	63.54
OpenBookQA	88.80	63.20	88.20	63.52	87.80	63.96	88.40	63.34	88.80	63.88
PIQA	86.94	62.82	87.38	63.30	87.76	63.34	86.89	63.64	87.16	63.60
SocialIQA	81.32	62.94	81.38	63.16	80.40	63.64	81.32	63.16	81.11	63.32
WinoGrande	83.19	49.20	84.93	62.14	83.03	63.68	84.11	63.14	84.61	63.10
Average	85.24	61.68	85.51	63.28	85.22	63.61	85.31	63.57	85.43	63.44

trend. Reasoning retention is similarly stable once the anchor set contains at least 64 examples, with reasoning accuracy remaining close to the pre-fine-tuning level across adapted checkpoints. The clearest degradation occurs at the smallest anchor set of 32 examples: Qwen2.5-1.5B loses roughly one to two points of reasoning accuracy on most tasks, while Qwen2.5-3B drops sharply on WinoGrande to 49.20%, likely reflecting a noisier estimate of the reasoning subspace from too few examples. These per-task results further support the main-text conclusion that a moderate anchor set, around 64 examples in our experiments, is sufficient to obtain a stable null-space estimate, making NB-LoRA practical even with limited anchor data.

C.1.3 Per-Task Results of the Energy-Threshold Ablation

Table 6 reports the per-task results underlying the energy-threshold ablation in Figure 5c, where the spectral-energy threshold defining the approximate null space varies from 0.75 to 0.99. The breakdown mirrors the findings in Figure 5c. Target-task accuracy is stable across the entire range: raising the threshold, and thus shrinking the trainable subspace, does not degrade adaptation on any task, in line with the spectral analysis in Section 3.3, which shows that the null space remains large even at a 0.99 energy threshold. Reasoning retention is robust for thresholds from 0.80 to 0.99 but collapses at 0.75, and the per-task view reveals that this collapse is task-dependent rather than uniform: on Qwen2.5-1.5B, adaptation on HellaSwag, PIQA, and SocialIQA reduces reasoning accuracy to 31.54%, 40.64%, and 28.89%, respectively, whereas ARC and OpenBookQA leave it largely intact. The same tasks produce the largest degradations on Qwen2.5-3B, including 19.12% after HellaSwag and 10.92% after SocialIQA. This pattern is consistent with a permissive threshold admitting more reasoning-relevant directions into the trainable subspace, with the resulting degradation depending on the target task being adapted. Finally, within the robust range, retention on Qwen2.5-1.5B improves mildly as the threshold increases (from 77.54% at 0.80 to 79.52% at 0.99 on average). This wide robust range suggests some flexibility in threshold selection: if a target task may benefit from

Table 6: **Per-task results of the energy-threshold ablation in Figure 5c.** Each column group corresponds to one spectral-energy threshold used to define the approximate null space. Target and Reason denote target-task and reasoning accuracy (%) at the post-SFT checkpoint. Target accuracy remains stable across thresholds, while reasoning retention is robust from 0.80 to 0.99 and degrades sharply at 0.75 for checkpoints adapted to several target tasks.

Task	0.75		0.80		0.85		0.90		0.95		0.99	
	Target	Reason	Target	Reason	Target	Reason	Target	Reason	Target	Reason	Target	Reason
Qwen2.5-1.5B (Reasoning task: GSM8K)												
ARC-Challenge	77.13	78.54	78.07	80.36	77.99	80.44	75.60	80.14	76.54	79.98	77.05	79.91
ARC-Easy	90.24	75.28	91.04	79.76	91.04	79.53	91.04	80.29	90.99	79.53	91.20	79.98
BoolQ	68.10	61.03	69.33	70.89	68.72	75.51	68.69	72.63	70.09	78.47	68.31	79.53
HellaSwag	90.85	31.54	90.98	74.68	90.68	75.51	91.04	79.45	90.97	79.45	91.37	79.83
OpenBookQA	84.00	77.33	86.20	79.68	85.20	79.38	85.60	80.36	84.60	79.91	84.60	78.62
PIQA	85.15	40.64	84.06	78.70	83.62	78.85	83.68	78.70	84.71	78.54	84.30	79.53
SocialIQA	76.92	28.89	76.31	79.15	76.61	78.32	76.87	78.47	75.90	79.38	76.73	78.77
WinoGrande	76.56	77.33	77.19	77.10	77.27	79.08	78.14	78.47	78.30	79.30	76.19	79.98
Average	81.12	58.82	81.65	77.54	81.39	78.33	81.33	78.56	81.51	79.32	81.22	79.52
Qwen2.5-3B (Reasoning task: MATH-lighteval)												
ARC-Challenge	83.02	63.06	82.51	63.14	83.62	63.32	82.42	63.20	80.38	63.58	82.56	63.50
ARC-Easy	93.94	58.80	94.15	62.98	92.85	63.76	93.90	63.86	93.01	63.62	93.69	63.16
BoolQ	71.01	58.68	72.14	61.74	71.68	62.08	71.41	62.54	71.19	64.22	71.28	63.40
HellaSwag	92.10	19.12	94.03	62.26	93.29	57.28	93.42	53.68	93.88	61.84	94.24	63.54
OpenBookQA	88.00	55.56	89.00	61.28	87.20	63.16	87.60	63.66	88.00	63.62	88.80	63.88
PIQA	87.00	54.90	86.06	61.76	86.34	63.38	86.07	63.02	86.34	63.58	87.16	63.60
SocialIQA	81.27	10.92	81.17	62.04	81.10	63.58	81.05	63.16	81.07	63.82	81.11	63.32
WinoGrande	82.64	63.42	84.53	63.38	82.72	63.76	83.50	63.28	83.58	63.24	84.61	63.10
Average	84.87	48.06	85.45	62.32	84.85	62.54	84.92	62.05	84.68	63.44	85.43	63.44

a larger update subspace, moderately lowering the threshold (e.g., to 0.95 or 0.90) could provide additional adaptation capacity with only a marginal reduction in reasoning retention.

C.2 Null-Space Capacity for Sequential Multi-Task Adaptation

The experiments in Section 4.1 demonstrate that NB-LoRA preserves reasoning during adaptation to an individual downstream task. We next investigate whether the null space provides sufficient capacity for multiple downstream tasks. Starting from each post-RL checkpoint, we first jointly adapt the model to eight commonsense tasks using NB-LoRA. We then merge the learned update, recompute the null basis using examples representing both the original reasoning capability and the newly acquired commonsense capabilities, and sequentially adapt the resulting model to tool use. This setting tests whether the null space can support cumulative capability acquisition rather than only a single adaptation.

Table 7: **Accuracy across sequential adaptation stages.** Reasoning, commonsense, and tool-use accuracy at the original reasoning-tuned checkpoint, after eight-task commonsense fine-tuning, and after sequential tool-use fine-tuning. Both adaptation stages use NB-LoRA. FT denotes fine-tuning.

Model	Stage	Reasoning acc.	Commonsense acc.	Tool-use acc.
Qwen2.5-1.5B GSM8K	Prior to commonsense FT	80.10%	33.30%	14.52%
	After commonsense FT	79.61%	79.52%	6.45%
	After tool-use FT	79.83%	79.72%	72.58%
Qwen2.5-3B MATH	Prior to commonsense FT	63.80%	54.58%	11.29%
	After commonsense FT	63.78%	84.62%	35.48%
	After tool-use FT	63.86%	84.68%	74.19%

As shown in Table 7, the models successfully accumulate new capabilities across the two adaptation stages. Commonsense accuracy increases from 33.30% to 79.52% for Qwen2.5-1.5B and from 54.58% to 84.62% for Qwen2.5-3B. Subsequent tool-use adaptation raises tool-use accuracy to 72.58% and 74.19%, respectively, while preserving the acquired commonsense performance. Reasoning accuracy also remains near its post-RL level throughout the sequence. These results demonstrate that the approximate null space can accommodate multiple heterogeneous downstream tasks while preserving both the original reasoning ability and capabilities acquired during earlier adaptation stages.

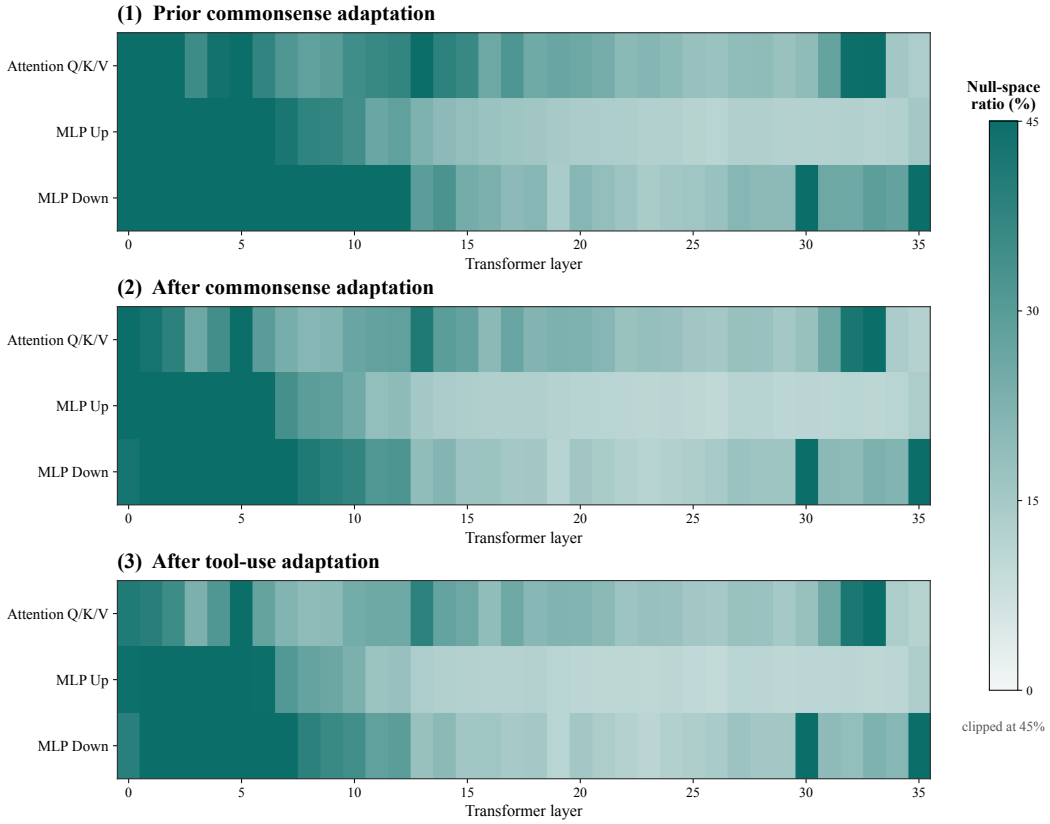


Figure 6: **Layer-wise null-space capacity across adaptation stages.** Heatmaps show the null-space ratio for attention projections, MLP up-projections, and MLP down-projections across transformer layers at three stages of sequential adaptation. All panels share the same color scale, clipped at 45% for visibility. Although the retained null space contracts across the adaptation stages, it remains broadly distributed across layers.

As the protected capability set expands, its activation subspace occupies more directions and the corresponding approximate null space gradually contracts (see Figure 7). Nevertheless, after adaptation to the eight commonsense tasks and tool use, the Qwen2.5-3B model retains approximately 25.0% null-space capacity in attention projections, 23.0% in MLP up-projections, and 33.4% in MLP down-projections. The layer-wise heatmaps in Figure 6 further show that the remaining capacity is broadly distributed across the network. Together, these results indicate that sequential adaptation with NB-LoRA does not exhaust the available null space, leaving a substantial subspace for further adaptation.

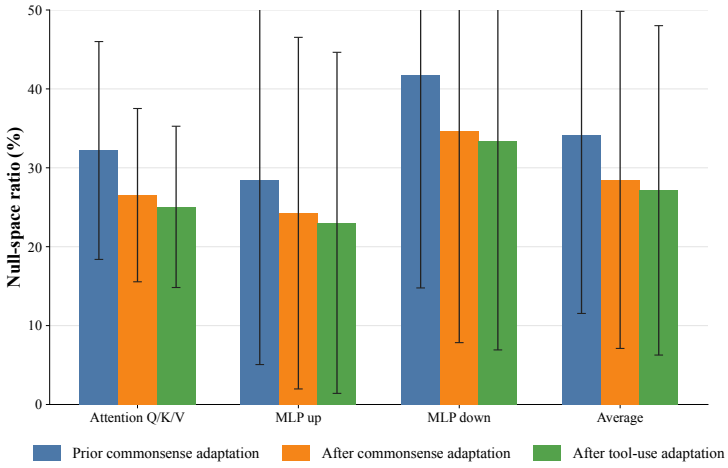


Figure 7: **Null-space capacity remains after sequential adaptation.** Mean null-space ratio across target module groups before commonsense fine-tuning, before tool-use adaptation, and after tool-use adaptation. The null-space ratio is defined as the retained basis dimension divided by the activation-space dimension. Error bars denote one standard deviation across layers. Although the null-space ratio decreases across the adaptation stages, the post-tool-use model retains substantial approximate null-space capacity across attention and MLP projections.

D Experiment Details

This appendix provides additional implementation details for reproducibility, including the training hyperparameters, datasets, prompt templates, and evaluation settings used in our experiments. All experiments were conducted on NVIDIA H100 GPUs with 80 GB of memory.

D.1 Hyperparameter Details

The hyperparameters used for NB-LoRA are summarized in Table 8.

Table 8: Training and null-space construction hyperparameters for NB-LoRA.

Hyperparameter	Commonsense reasoning	Tool use
Optimizer	Adam	
Learning rate	1×10^{-4}	2×10^{-4}
Epochs	3	
Batch size	16	
Maximum sequence length	1024	4096
LoRA rank r	16	
LoRA scaling α	32	
LoRA dropout	0.05	
Adapted modules	Query, key, value, up-, and down-projections	
Anchor corpus	MATH-lighteval (3B); GSM8K (1.5B)	
Number of anchor examples	512	
Protected spectral energy	99%	
Reasoning tokens	All tokens in the generated reasoning traces	

D.2 Dataset Details

We provide additional details on the commonsense reasoning and tool-use datasets used for downstream adaptation, as well as MMLU-STEM, which is used only for held-out evaluation.

Table 9: The prompt template of the commonsense reasoning datasets.

Dataset	Input Template
ARC-C/E	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer4: [ANSWER_4] Answer format: answer1/answer2/answer3/answer4 the correct answer is [ANSWER]
BoolQ	Please answer the following question with true or false, question: [QUESTION] Answer format: true/false the correct answer is [ANSWER]
HellaSwag	Please choose the correct ending to complete the given sentence: [ACTIVITY_LABEL]: [CONTEXT] Ending1: [ENDING_1] Ending2: [ENDING_2] Ending3: [ENDING_3] Ending4: [ENDING_4] Answer format: ending1/ending2/ending3/ending4 the correct answer is [ANSWER]
OpenBookQA	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer4: [ANSWER_4] Answer format: answer1/answer2/answer3/answer4 the correct answer is [ANSWER]
PIQA	Please choose the correct solution to the question: [QUESTION] Solution1: [SOLUTION_1] Solution2: [SOLUTION_2] Answer format: solution1/solution2 the correct answer is [ANSWER]
SocialIQA	Please choose the correct answer to the question: [QUESTION] Answer1: [ANSWER_1] Answer2: [ANSWER_2] Answer3: [ANSWER_3] Answer format: answer1/answer2/answer3 the correct answer is [ANSWER]
WinoGrande	Please choose the correct answer to fill in the blank to complete the given sentence: [SENTENCE] Option1: [OPTION_1] Option2: [OPTION_2] the correct answer is [ANSWER]

D.2.1 COMMONSENSE REASONING BENCHMARKS

The commonsense reasoning benchmarks contain approximately 170K training examples from eight tasks: ARC-C and ARC-E (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SocialIQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2021). We briefly describe these eight tasks below.

- **ARC-C/E** comprises the Challenge and Easy partitions of ARC, a collection of multiple-choice science questions drawn from grade-school examinations.
- **BoolQ** contains naturally occurring yes-or-no questions paired with passages that provide the information needed to answer them.
- **HellaSwag** asks the model to select the most plausible continuation of a given context from four candidate endings.
- **OpenBookQA** evaluates elementary science question answering that requires combining core scientific facts with broader commonsense knowledge.
- **PIQA** measures physical commonsense by asking the model to choose the more plausible of two solutions to an everyday problem.
- **SocialIQA** uses three-choice questions to assess reasoning about people's intentions, actions, reactions, and social consequences.
- **WinoGrande** evaluates commonsense and coreference reasoning through binary-choice fill-in-the-blank questions derived from ambiguous sentences.

The input template, i.e., prompt format for these datasets is detailed in Table 9 (Hu et al., 2023; Fang et al., 2025).

D.2.2 TOOL-USE TASKS

For the tool-use task, we use ToolAlpaca (Tang et al., 2023), in which the model receives a user request and natural-language documentation for a tool collection, and must select an API and produce its arguments as a JSON object. We retain only single-call examples because they provide clean, self-contained supervision, whereas later calls in multi-call trajectories may depend on unobserved intermediate tool outputs. The resulting subset contains 2,513 training examples and 62 evaluation examples. The generic prompt format is shown in Table 10, and representative examples are shown in Table 11.

Table 10: The prompt template of the tool-use adaptation dataset.

Dataset	Input Template
Tool use	Your task is to answer the user's question using available tools. You have access to the following tools: Name: [TOOL NAME] Description: [TOOL DESCRIPTION] Documentation: [ACTION NAME]: [ACTION DESCRIPTION] Parameters: [JSON PARAMETER SCHEMA] Output: [OUTPUT DESCRIPTION] Use the following format: Thought: you should always think about what to do Action: the action to take, which should be one of the available action names Action Input: the input to the action in JSON format Begin! Question: [USER QUESTION]

Table 11: Representative tool-use examples from the Axolotl tool collection. Each gold response identifies the correct API and supplies its arguments as a JSON object.

Example	User Request and Gold Tool Call
Random image	**Question:** Hey, can you show me a random picture of an axolotl? **Action:** `getRandomAxolotlImage` **Action Input:** {}
Filtered search	**Question:** I'm looking for an axolotl that is wild in color and medium in size. Can you help me find some pictures? **Action:** `searchAxolotlImages` **Action Input:** `{"color": "wild", "gender": "", "size": "medium", "page": 1}`
Fact lookup	**Question:** Can you tell me three interesting facts about axolotls' habitats? **Action:** `getAxolotlFacts` **Action Input:** `{"category": "habitat", "limit": 3}`

D.2.3 MMLU-STEM Benchmark

MMLU (Hendrycks et al., 2020) is a multiple-choice benchmark spanning 57 academic and professional subjects. Its STEM subset contains 3,018 evaluation questions across 18 subjects in mathematics, physics, chemistry, biology, computer science, and engineering, ranging from elementary to college level. MMLU-STEM is used only for held-out evaluation; its examples are not used for downstream adaptation or null-space construction.

E Qualitative Case Study: Reasoning Retention and Downstream Adaptation

We examine two complementary aspects of post-RL adaptation. First, using MATH-lighteval queries (Hendrycks et al., 2021), we compare the Qwen2.5-3B post-RL model before and after adaptation with Vanilla LoRA and NB-LoRA, thereby assessing retention of its original mathematical reasoning ability. Second, using HellaSwag queries (Zellers et al., 2019), we illustrate adaptation to a downstream commonsense task.

E.1 MATH Reasoning Examples

Query. How many vertical asymptotes does the graph of $y = 2/(x^2 + x - 6)$ have? Think step by step and place the final answer in $\boxed{\text{answer}}$.

Reference answer. 2.

Post-RL reasoning model (correct)

To determine the number of vertical asymptotes, we need to find the values of x that make the denominator zero. First, factor the denominator:

$$x^2 + x - 6 = (x + 3)(x - 2).$$

Setting it equal to zero gives $x = -3$ and $x = 2$. These are potential vertical asymptotes, and neither is canceled because the numerator is the nonzero constant 2. Therefore, the function has two vertical asymptotes, and the final answer is $\boxed{2}$.

After Vanilla LoRA (incorrect)

```
the correct answer is \boxed{6}
```

After NB-LoRA (correct)

The given function is a rational function. To find its vertical asymptotes, we find the values of x that make the denominator zero:

$$x^2 + x - 6 = (x + 3)(x - 2) = 0.$$

Thus, $x = -3$ and $x = 2$, so the graph has two vertical asymptotes. The final answer is $\boxed{2}$.

Query. What is the smallest value of x such that $|5x - 1| = |3x + 2|$? Express the answer as a common fraction, think step by step, and place the final answer in $\boxed{\text{answer}}$.

Reference answer. $-1/8$.

Post-RL reasoning model (correct)

The expressions inside the absolute values change sign at $x = 1/5$ and $x = -2/3$, which divide the number line into three intervals.
For $x < -2/3$, both expressions are negative, so

$$-(5x - 1) = -(3x + 2) \implies x = \frac{3}{2},$$

which is outside this interval. For $-2/3 < x < 1/5$, the first expression is negative and the second is positive, so

$$-(5x - 1) = 3x + 2 \implies x = -\frac{1}{8},$$

which lies in this interval. For $x > 1/5$, both expressions are positive, so

$$5x - 1 = 3x + 2 \implies x = \frac{3}{2},$$

which is valid in this interval. The smaller of the two valid solutions is $-1/8$, so the final answer is $\boxed{-\frac{1}{8}}$.

After Vanilla LoRA (incorrect)

```
the correct answer is \boxed{\frac{3}{4}}
```

After NB-LoRA (correct)

The critical points are $x = 1/5$ and $x = -2/3$, yielding the intervals $(-\infty, -2/3)$, $(-2/3, 1/5)$, and $(1/5, \infty)$.
For $x < -2/3$, both expressions are negative:

$$-(5x - 1) = -(3x + 2) \implies x = \frac{3}{2},$$

which is not in the interval. For $-2/3 < x < 1/5$, the first expression is negative and the second is positive:

$$-(5x - 1) = 3x + 2 \implies x = -\frac{1}{8},$$

which is valid. For $x > 1/5$, both expressions are positive:

$$5x - 1 = 3x + 2 \implies x = \frac{3}{2},$$

which is also valid. Hence, the smaller solution is $-1/8$, and the final answer is $\boxed{-\frac{1}{8}}$.

E.2 HellaSwag Examples

Query. A man is holding a pocket knife while sitting on some rocks in the wilderness. Then he:

1. opens a can of oil, puts oil on the knife, presses it through the can, then cuts pieces from sandwiches;
2. takes a small stone from the flowing river and smashes it on another stone;
3. uses the knife to shave his leg; or
4. sands the rocks and tops them using strong pressure.

Reference answer. `ending2`.

Post-RL reasoning model (incorrect)

```
The correct ending to complete the given sentence is: Ending3:
uses the knife to shave his leg.
```

After Vanilla LoRA (correct)

```
The correct answer is ending2.
```

After NB-LoRA (correct)

```
The correct answer is ending2, because it logically follows.
```

Query. A boy is running down a track. The boy:

1. runs into a car;
2. gets in a mat;
3. lifts his body above the height of a pole; or
4. stands on his hands and springs.

Reference answer. `ending3`.

Post-RL reasoning model (incorrect label)

```
The correct ending to complete the given sentence is: Ending1:
lifts his body above the height of a pole.
```

After Vanilla LoRA (correct)

```
The correct answer is ending3.
```

After NB-LoRA (correct)

```
The correct answer is ending3, because it logically follows.
```

E.3 SUMMARY

Across the MATH examples, NB-LoRA retains the post-RL model's multi-step reasoning ability and required answer format after downstream adaptation, whereas Vanilla LoRA produces incorrect answers without an explicit reasoning process. The HellaSwag examples complement this analysis by showing that both adaptation methods acquire the downstream commonsense task and select the correct continuation in both cases. Taken together, these examples illustrate that NB-LoRA achieves the intended balance between preserving the original reasoning ability and acquiring a new downstream capability.