GRABLORA: A GRASSMANNIAN FRAMEWORK FOR TRACKING GRADIENT SUBSPACE ROTATION IN LORA

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation (LoRA) is a widely used approach in parameter-efficient fine-tuning, and a recent line of work augments it by initializing LoRA with gradient-informed subspaces. Such one-time placement, however, overlooks the evolution of the gradient space during training. In this work, we investigate two questions: Does the latent gradient subspace *rotate* as LoRA training proceeds? If yes, then how to *keep track of* the rotating subspace? To study these problems in a geometry-aware setting, we introduce GrabLoRA, a Bayesian Grassmannian LoRA framework that treats the low-rank subspace as an evolving state rather than a fixed initialization. To realize this, GrabLoRA inherits the information-rich early gradient subspace as a prior, realigns the left and right subspaces on the Grassmann manifold using incoming gradient evidence, and applies an uncertainty-aware preconditioner that balances historical geometric uncertainty with newly observed gradients. Across natural language understanding and generation tasks, GrabLoRA achieves consistent improvements over standard LoRA and gradient-initialized baselines[1]. We also provide theoretical insights for gradient-space coverage and convergence of GrabLoRA.

1 INTRODUCTION

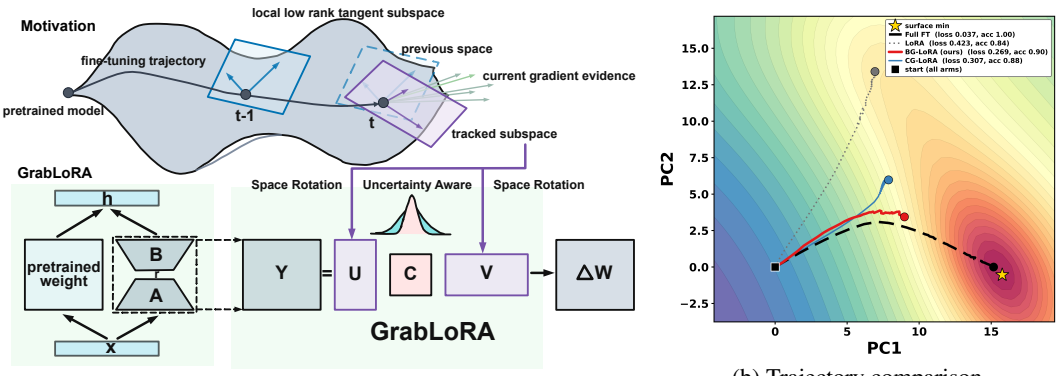
Low-rank adaptation (LoRA) (Hu et al., 2021), as a standard parameter-efficient fine-tuning (PEFT) method (Houlsby et al., 2019), serves as an efficient bridge between the large-scale pretraining and downstream deployment (Howard & Ruder, 2018; Salimbeni et al., 2024; Roe et al., 2026; Devlin et al., 2019) by updating parameters in a low-dimensional space, which substantially reduces the storage, memory, and optimization costs. Owing to its simplicity and efficiency, LoRA has been applied across instruction tuning (Ge et al., 2025; Che et al., 2026), mathematical reasoning (Wang et al., 2025; Li et al., 2026) and code generation (Zhao, 2025).

Recent LoRA variants improve this framework through better rank allocation, sparsity and masking mechanisms (Zhang et al., 2023; Valipour et al., 2023; Mao et al., 2024; Balzano et al., 2026). Most of these methods focus on *how much* the update can express, such as its rank, budget, or the parameters it touches, while leaving *where* the update lives largely unexamined. The subspace in which the low-rank update resides, and the geometry of the latent space it operates in, are treated as fixed or incidental rather than as an object deserving design. Recognizing that this subspace matters, a line of work exploits gradient information at initialization ($t = 0$) to place the low-rank update in a task-relevant subspace (Wang et al., 2024b; Zhang et al., 2025; Zheng & Proutière, 2026), which substantially improves early optimization.

However, downstream adaptation is not a static correction around the pretrained initialization. The pretrained parameters W_0 encode broad corpus-level statistics, whereas downstream tasks may occupy shifted regions of the representation space (Ben-David et al., 2010; Gururangan et al., 2020; Merchant et al., 2020). As fine-tuning progresses, the model moves away from W_0, and the dominant gradient directions can drift accordingly. Thus, a subspace that is well aligned with the initial gradient may become progressively less aligned with the gradient observed at later training steps. Figure 1 (right) illustrates this phenomenon: the task-relevant gradient subspace rotates along the

[1]Code is available in the supplementary material.

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.



(a) Overview of GrabLoRA.

(b) Trajectory comparison

Figure 1: Overview of GrabLoRA. **Left**: Along the fine-tuning trajectory, GRABLORA tracks the task-relevant gradient subspace on Grassmann manifold, propagating the previous geometry while realigning it with newly observed evidence, plus an uncertainty-aware preconditioner at its core. **Right**: 2D PCA projection of trajectories comparing full fine-tuning method (Full FT), vanilla LoRA method (LoRA), an initialization-only LoRA method (CG-LoRA), and ours (GRABLORA) on a training–fine-tuning misaligned synthetic task. At early steps, both CG-LoRA and GRABLORA follow the descent direction. Once the gradient direction rotates, only GRABLORA keeps tracking the shifted latent subspace and converges toward the new optimum while vanilla LoRA and CG-LoRA drift away. Implementation detail is available in Appendix C.

fine-tuning trajectory, while initialization-only methods remain tied to their original subspaces, leading to a slower training loss decrease.

One way to address this is to adapt the low-rank structure during training rather than only at initialization. Several methods make such intermediate changes to the LoRA architecture, including rank pruning (Zhang et al., 2023), reinitialization (Xia et al., 2024), and subspace recomposition (Nikdan et al., 2024; Yang et al., 2025). These operations, however, reshape the structure of the update, its rank, its reset points and its factor composition, without ever tracking the geometry of the latent gradient subspace they act on. As a result, the task-relevant geometry injected at $t = 0$ is neither monitored nor deliberately maintained. It is left to drift as an unintended by-product of these structural edits, rather than treated as a quantity to be preserved and refined. This gap raises two questions: *Does the task-relevant latent gradient subspace rotate as LoRA training proceeds?* and, if yes, then

How to keep track of this rotation without discarding useful geometry already learned?

To answer the second question, we propose **GRABLORA** (Figure 1 (left)), a geometry-aware LoRA framework that treats the latent gradient subspace as an evolving state rather than a one-time initialization. At each refresh point, GRABLORA collects gradient evidence and updates the left and right subspaces as points on Grassmann manifolds, using a closed-form optimization which balances the incoming gradients and previously tracked geometry. To adapt to the changing bases of the rotating subspaces, GRABLORA further introduces an uncertainty-aware preconditioner. Based on Adam, the update along each tracked direction is scaled not only by its estimated gradient signal, but also by the confidence with which that direction has been identified from the available evidence. Thus, GRABLORA couples geometry-aware subspace tracking with uncertainty-aware step size allocation, improving downstream adaptation while preserving task-relevant gradient geometry. We further provide theoretical guarantees characterizing the convergence behavior of GRABLORA.

Roadmap and contributions. Our contributions are summarized as follows:

- **Latent gradient subspace rotation**. We identify and empirically characterize the *rotation* of the task-relevant latent gradient subspace during LoRA fine-tuning. This observation reveals the misalignment between initialization-only subspaces and the dominant gradient directions later in training, motivating the design of GRABLORA (Section 3).
- **Novel Framework**. We propose GRABLORA, a novel LoRA framework that updates the left and right latent gradient subspace as points on Grassmann manifold through a closed-

form optimization, which balances the incoming evidence and the historical geometry. To adapt the low-rank update to this rotating basis, we further introduce an uncertainty-aware preconditioner that scales each direction using both inherited geometric uncertainty and evidence-induced uncertainty (Section 4).

- **Performance.** Across natural language understanding and generation tasks, GRABLORA achieves the strongest overall performance among the baselines while retaining the memory efficiency of LoRA. Also, GRABLORA maintains high coverage of the evolving gradient subspace throughout training (Section 7).
- **Theoretical insights**. We analyze how the Grassmannian update tracks the rotating latent gradient subspace with controlled error and preserves high gradient coverage. Also, we show that an idealized update based on this projected gradient yields descent under standard smoothness assumptions (Section 5).

2 RELATED WORK

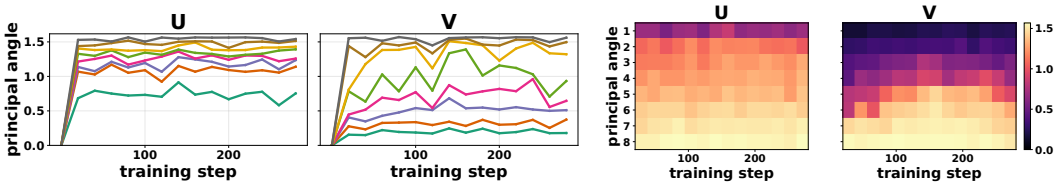
LoRA and its variants. One line of work builds more informative initializations through singular value decomposition (PiSSA (Meng et al., 2025)), first-order (LoRA-GA (Wang et al., 2024b), LoRA-One (Zhang et al., 2025)) and second order information (CG-LoRA (Zheng & Proutière, 2026)). These methods fix the subspace at initialization while do not account for its drift as training proceeds. Another line of work alters LoRA's parameterization during training through direction pruning (AdaLoRA (Zhang et al., 2023)), learning rate decoupling (LoRA+ (Hayou et al., 2024)), sparsity masking (ROSA (Nikdan et al., 2024)). These edits act on structure during training yet without tracking the geometry of the gradient subspace. In contrast, GRABLORA bridges this gap, carrying the information-rich early subspace forward and continually realigning it with the rotating gradient geometry as training proceeds.

Grassmannian geometry. A manifold is a space that may be globally curved yet locally admits ordinary Euclidean coordinates (Hatcher, 2001). Let $U = [u_1, \dots, u_r] \in \mathbb{R}^{d\times r}$ have orthonormal columns (i.e., $U^\top U = I_r$). The Stiefel manifold $\mathrm{St}(r, d) = \left\{U \in \mathbb{R}^{d\times r} : U^\top U = I_r\right\}$ is the set of all such ordered orthonormal r-frames. In contrast, the Grassmann manifold $\mathrm{Gr}(r, d)$ is the set of all r-dimensional linear subspaces of $\mathbb{R}^d$, which, different from Stiefel manifold, is basis-invariant. Taking the benefit of basis-invariance, Grassmann manifolds has become a standard tool for modeling subspaces in machine learning, including computer vision (Yataka et al., 2023; Zhang et al., 2018; Harandi et al., 2011), principal subspaces and dimension reduction (Delogne & Jacques, 2025; Balzano et al., 2011; Saad-Falcon et al., 2024), and signal processing (Wang et al., 2024a; Shi et al., 2014). GRABLORA also models each latent gradient subspace as a point on a Grassmann manifold, where basis ambiguity vanishes and the subspace drift during training is measured and the subspace is updated.

Dynamic subspace tracking. Dynamic subspace tracking has been a long-studied problem in signal processing and machine learning, with applications spanning computer vision (Narayanamurthy & Vaswani, 2018), online learning (Xie et al., 2015) and network sensing (Chi & Fu, 2017). Grassmannian formulations are especially natural since they track the subspace intrinsically rather than a particular basis. GROUSE (Balzano et al., 2011), as a classical instance, recovers and tracks a subspace from highly incomplete observations, much as GRABLORA refines its subspaces from limited, incoming gradient evidence. Later work models the subspace's temporal evolution as motion along Grassmannian geodesics (Blocker et al., 2023; Saad-Falcon et al., 2024). This idea has recently reached large model training. Subtrack (Rajabi & Rambhatla, 2024) and SubTrack++ (Rajabi et al., 2025) track the dominant gradient subspace on the Grassmannian with rank-one geodesic updates, enabling memory-efficient training. GRABLORA differs in its mechanism by jointly inferring the left and right subspaces with preconditioned update, which yields an uncertainty-aware and geometric-aware adaptation.

3 LATENT GRADIENT SUBSPACE ROTATION IN LORA

In this section, we first review the LoRA setting, then introduce a metric to quantify latent gradient subspace rotation and empirically demonstrate this phenomenon during fine-tuning.



(a) Cumulative rotation: principal angles of (U_t, V_t) from initialization (U_0, V_0).

(b) Incremental rotation: principal angles of (U_t, V_t) at a fixed lag $\tau = 20$.

Figure 2: Principal angles of the rank-$r = 8$ left (U) and right (V) gradient subspaces over 300 training steps on the MMLU task (5-shot) of the layer 0 query matrix in Llama-3.1-8B. **(a)** Measured against initialization, the subspace becomes nearly orthogonal to its gradient-informed starting point. **(b)** Measured against $t - \tau$, the angles remain nonzero throughout, so the subspace keeps rotating rather than settling. Both panels reveal that the latent gradient subspaces rotate continuously throughout LoRA fine-tuning.

Review of LoRA. Motivated by the observation that the weight updates required for fine-tuning are intrinsically low-rank (Li et al., 2018; Aghajanyan et al., 2020), LoRA constrains the weight update to a fixed low-rank form,

$$W_t = W_0 + \Delta W_t = W_0 + \eta B_t A_t, \qquad B_t \in \mathbb{R}^{d_{\text{out}} \times r},\ A_t \in \mathbb{R}^{r \times d_{\text{in}}},\ r \ll \min\{d_{\text{out}}, d_{\text{in}}\}$$

While this captures the cumulative update, the loss gradient is more informative. We therefore study the instantaneous gradient of the fine-tuning objective and its dominant subspace:

$$Y_t = -\nabla_w \mathcal{L}(x; w_t), \qquad \widehat{Y}_t = U_t \Sigma_t V_t^\top$$

where $\widehat{Y}_t$ is the rank-r truncated SVD approximation of Y_t, with orthonormal bases $U_t \in \mathbb{R}^{d_{\text{out}} \times r}$ and $V_t \in \mathbb{R}^{d_{\text{in}} \times r}$. We refer to U_t and V_t as the left and right latent gradient subspaces.

Latent gradient space rotation. To address the first question put forward in Section 1: *Does the latent gradient subspace rotate as LoRA training proceeds?*, we track how U_t and V_t move over training. We distinguish two forms of subspace motion. The *cumulative* rotation compares (U_t, V_t) against the initialization (U_0, V_0), measuring how far the subspace has departed from its gradient-informed initialization point. The *incremental* rotation compares (U_t, V_t) against $(U_{t-\tau}, V_{t-\tau})$ at a fixed lag τ, measuring the ongoing, local extent of rotation.

To quantify the cumulative and incremental rotation of the U and V space, we measure the *principal angle* between each subspace and its counterpart at an earlier training step, a basis-invariant characterization of the separation between two equal-rank subspaces. For the subspaces spanned by the columns of U_s and U_t, the principal angles $\theta_1, \ldots, \theta_r$ are obtained from the singular values of $U_s^\top U_t$:

$$U_s^\top U_t = L\, \text{diag}(\sigma_1, ..., \sigma_r) R^\top,\ \theta_k(U_s, U_t) = \arccos(\sigma_k),\ 0 \le \theta_1 \le \cdots \le \theta_r \le \frac{\pi}{2}$$

Measuring cumulative rotation from $t = 0$ (Figure 2a), the principal angles rise steeply and approach $\frac{\pi}{2}$ (up to $\sim$1.5 rad $\approx 85.9°$) within roughly 50 steps, so some directions become nearly orthogonal to their gradient-informed starting point. Incrementally (Figure 2b), even the smallest angle θ_1 stays at $\sim$0.4 rad $\approx 22.9°$ for lag $\tau = 20$, with higher-indexed directions rotating more. Both figures show that the subspace keeps drifting rather than settling after the initial transient. This behavior is not layer-specific (Appendix E.1).

4 METHOD

Section 3 shows that the task-relevant latent gradient subspace rotates continuously during LoRA fine-tuning. Tracking this rotation, rather than fixing the subspace at initialization, keeps the adapter aligned with the directions that drive learning, yielding steeper loss descent and stronger downstream performance while preserving a low-dimensional structure. This motivates GRABLORA. We answer the second question in Section 1: *How to keep track of this rotation, without discarding useful geometry already learned?* The method is organized as follows. We first derive how

Algorithm 1 GRABLORA

Require: pretrained W_0; steps T; rank r; refresh period τ; step size η

Initialize subspaces U_0, V_0 and states $\{P^+_{0,k}, \mu_{0,k}, c_{0,k}\}_{k=1}^r$
for $t = 1, \ldots, T$ **do**
  Get gradient evidence $\{Y_{t,i}\}_{i=1}^{n_t}$, $\bar{Y}_t = \frac{1}{n_t}\sum_i Y_{t,i}$
  if $t \bmod \tau \equiv 0$ **then**
    Rotate the subspaces U_t, V_t by minimizing Eq. 1 ▷ Subspace rotation
  else
    Keep the subspaces: $U_t, V_t \leftarrow U_{t-1}, V_{t-1}$
  end if
  Form the tracked directions $A_{t,k} = u_{t,k} v_{t,k}^\top$
  for $k = 1, \ldots, r$ **do**
    Preconditioned steps update by Eq. 2 ▷ Uncertainty-aware preconditioner
  end for
  Propose ΔW_t by Eq. 3, $W_t \leftarrow W_{t-1} + \Delta W_t$
end for
return W_T

GRABLORA rotates the subspace by trading off geometric continuity against incoming evidence. Then we introduce how GRABLORA adapts the Adam preconditioner to a *rotating basis* by incorporating uncertainty into the coefficient update. Finally, we assemble the full update loop, including the out-of-subspace correction. Algorithm 1 summarizes the full procedure while Algorithm 2 in Appendix A provides a detailed version.

Grassmannian Subspace Rotation. Suppose that, at step t, we are given the previous left and right subspaces U_{t-1} and V_{t-1}, together with a batch of gradient evidence $Y_{t,i} = -\nabla_w \mathcal{L}(x_i; w_t)$, $i = 1, \ldots, n_t$. To update the latent gradient subspace projectors $P_t^U = U_t U_t^\top$ and $P_t^V = V_t V_t^\top$ without either discarding the geometry learned so far, or overfitting to the new evidence, we therefore optimize the objective function

$$J(P^U, P^V) = \sum_{i=1}^{n_t} \|Y_{t,i} - P^U Y_{t,i} P^V\|_F^2 + \frac{\kappa_U}{2}\|P^U - P_{t-1}^U\|_F^2 + \frac{\kappa_V}{2}\|P^V - P_{t-1}^V\|_F^2 \quad (1)$$

where the first term encourages P_t^U and P_t^V to capture most of the new evidence $Y_{t,i}$ while the projection-distance penalties encourage the subspaces to remain aligned with the learned directions, which helps preserve the geometric continuity on the Grassmann manifold. Thus, κ_U and κ_V control the trade-off between adapting to new evidence and retaining the previously preserved geometry. Beyond the interpretation of Equation 1, this optimization problem also admits closed-form alternating updates. Fixing P^V, the updated left subspace is given by the top-r eigenspace of $\sum_{i=1}^{n_t} Y_{t,i} P^V Y_{t,i}^\top + \kappa_U P_{t-1}^U$, with the right subspace updated symmetrically. Details are provided in Section B.1. After obtaining U_t and V_t, we form a tracked coordinate subspace using the paired rank-one directions $A_{t,k} = u_{t,k} v_{t,k}^\top$, $k = 1, \ldots, r$, which serve as the coordinate basis for the preconditioned update in the following.

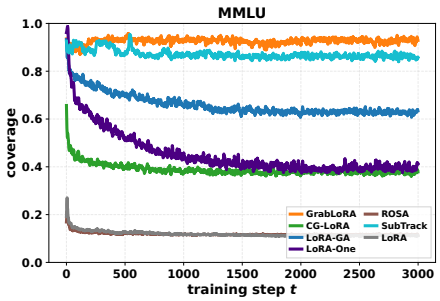


Figure 3: MMLU (5-shot) coverage of all methods over 3000 training steps on Llama-3.1-8B. GRABLORA achieves the highest coverage.

Figure 3 previews the effectiveness of our design. On MMLU (5-shot) task with model Llama-3.1-8B, GRABLORA maintains high coverage of the task-relevant gradient subspace throughout training compared with different baselines. The full definition and evaluation of coverage are given in Section 7.

Uncertainty-aware preconditioner. Adam (Kingma & Ba, 2017) scales each coordinate by the inverse square root of an empirical second moment, which is well defined only when the coordinate system is fixed, so that every step refers to the same direction. In GRABLORA, the k-th coefficient is instead tied to a moving rank-one direction $A_{t,k} = u_{t,k} v_{t,k}^\top$ from optimization in Equation 1. As the subspace rotates, the moment accumulated in the old basis no longer corresponds to the new

direction. Therefore, we replace Adam's empirical second moment with a rotation-aware uncertainty estimate. To capture the geometry variation, we use $\rho_{t,k} = |\langle A_{t,k}, A_{t-1,k}\rangle|_F \in [0,1]$ to denote the overlap between consecutive directions. For the new incoming evidence $Y_{t,i}$, whose new coordinate is $z_{t,i,k} = \langle A_{t,k}, Y_{t,i}\rangle_F$ through Frobenius inner product. We denote its batch-averaged variance to be:

$$\mathrm{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-) = P_{t,k}^- + R_{t,k} = \rho_{t,k}^2 P_{t-1,k}^+ + q_0 + q_1\big(1-\rho_{t,k}^2\big) + (n_t\beta_{t,k})^{-1} \tag{2}$$

where $\mathcal{F}_t^-$ is the σ-algebra of the information available before observing the current batch evidence at time t. Appendix A provides a rigorous model setup, definitions and assumptions.

Here, we provide some insights. $P_{t,k}^- = \rho_{t,k}^2 P_{t-1,k}^+ + q_0 + q_1\big(1-\rho_{t,k}^2\big)$ combines uncertainty $p_{t-1,k}^+$ inherited from time $t-1$ with bases process noise q_0 and rotation-induced uncertainty $q_1(1-\rho_{t,k}^2)$. When direction k is stable (i.e., the rotation is small and the overlap is close to one, $\rho_{t,k} \to 1$), the previous uncertainty is mostly inherited ($P_{t,k}^- \to P_{t-1,k}^+ + q_0$), while when it rotates substantially ($\rho_{t,k} \to 0$), the term $q_1(1-\rho_{t,k}^2)$ adds extra uncertainty for the mismatch between the old history and the new basis. Adding the incoming batch evidence contributes an additional noise variance $R_{t,k} = (n_t\beta_{t,k})^{-1}$, so the predictive uncertainty combines uncertainty in the tracked geometry with noise in the new evidence.

Taking advantage of the composition of uncertainty in $\mathrm{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-)$, we use its square root as the denominator in Adam, while keeping Adam's first-momentum update unchanged. Intuitively, the denominator accounts for both geometric uncertainty $P_{t,k}^-$ and incoming evidence noise $R_{t,k}$. Therefore, higher uncertainty increases the denominator and reduces the step magnitude along direction k, making the update more conservative.

Empirical insight. According to Equation 2, we decompose $\mathrm{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-)$ into observation noise $(n_t\beta_{t,k})^{-1}$ (noise), inherited geometric uncertainty $\rho_{t,k}^2 P_{t-1,k}^+$ (inherit), base process noise q_0 (base), and rotation-induced uncertainty $q_1(1-\rho_{t,k}^2)$ (rotation). Figure 4 reports their relative contributions, using the GLUE SST-2 task with $\kappa_U = \kappa_V = 0.1$ across rotation quantiles Q_1 to Q_5 as an example, where larger quantiles correspond to stronger subspace rotation. Across the configurations with $q = q_0 = q_1 \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, there are some regular findings. As rotation increases, the contribution of inherited uncertainty decreases, and the denominator becomes increasingly dominated by current observation noise. The rotation-induced uncertainty is most visible in the intermediate regime, where the basis has moved enough to make past coefficient history less reliable, but not so much that the update is driven almost entirely by new evidence. Thus, this preconditioner is rotation-aware in its uncertainty accounting.

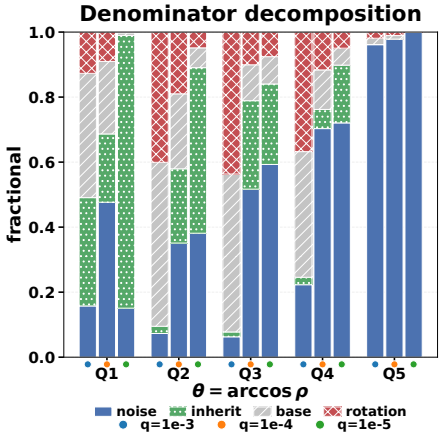


Figure 4: Preconditioner with uncertainty aware denominator decomposition into observation noise, inherited uncertainty, base process noise, and rotation uncertainty on GLUE-SST2, across rotation bins from low (Q_1) to high (Q_5).

Full update loop The preconditioned coefficient update acts on the core connecting P_t^U and P_t^V. However, when the subspace rotates, the current gradient may also contain first-order components that move into or out of the left and right subspaces. These tangent components, $(I-P_t^U)\bar{Y}_t P_t^V$ and $P_t^U \bar{Y}_t (I-P_t^V)$, are not associated with the inherited coefficient posterior, so we use the current batch evidence directly for them. The resulting update to the adapter is

$$\Delta W_t \;=\; \eta\left[\sum_{k=1}^{r} \hat{c}_{t,k} A_{t,k} \;+\; (I-P_t^U)\bar{Y}_t P_t^V + P_t^U \bar{Y}_t (I-P_t^V)\right] \tag{3}$$

where $\hat{c}_{t,1}, \ldots, \hat{c}_{t,r}$ are the uncertainty-preconditioned coefficients for rank-1 direction $A_{t,k}$. More details are available in Appendix A.

Method	CoLA	MRPC	QNLI	QQP	RTE	SST-2	Avg
Full FT	$79.73_{\pm 0.38}$	$86.44_{\pm 0.51}$	$91.96_{\pm 0.07}$	$89.33_{\pm 1.00}$	$79.78_{\pm 1.30}$	$93.58_{\pm 0.90}$	86.80
LoRA	$78.43_{\pm 1.15}$	$86.03_{\pm 0.65}$	$\underline{92.46}_{\pm 0.63}$	$68.13_{\pm 4.22}$	$67.63_{\pm 4.93}$	$\underline{93.54}_{\pm 0.07}$	81.04
LoRA-GA	$78.07_{\pm 0.29}$	$\underline{87.09}_{\pm 1.13}$	$91.69_{\pm 0.28}$	$71.07_{\pm 9.32}$	$65.82_{\pm 0.75}$	$93.31_{\pm 0.18}$	81.17
LoRA-One	$79.20_{\pm 1.48}$	$78.92_{\pm 1.36}$	$\mathbf{92.49}_{\pm 0.25}$	$\underline{85.13}_{\pm 1.46}$	$\underline{73.16}_{\pm 2.12}$	$93.50_{\pm 0.58}$	$\underline{83.74}$
CG-LoRA	$\underline{79.33}_{\pm 0.50}$	$80.64_{\pm 3.61}$	$92.06_{\pm 0.28}$	$82.73_{\pm 0.95}$	$73.16_{\pm 1.99}$	$93.43_{\pm 0.13}$	83.56
ROSA	$69.90_{\pm 0.82}$	$74.84_{\pm 0.62}$	$92.42_{\pm 0.51}$	$79.13_{\pm 6.12}$	$63.66_{\pm 4.91}$	$93.46_{\pm 0.30}$	78.90
SubTrack[2]	$67.67_{\pm 2.06}$	$74.84_{\pm 1.86}$	$63.23_{\pm 2.57}$	$78.87_{\pm 2.48}$	$55.96_{\pm 3.25}$	$87.73_{\pm 2.79}$	71.38
GrabLoRA	$\mathbf{79.50}_{\pm 0.98}$	$\mathbf{87.17}_{\pm 1.90}$	$92.01_{\pm 0.53}$	$\mathbf{89.70}_{\pm 0.62}$	$\mathbf{76.77}_{\pm 3.07}$	$\mathbf{93.69}_{\pm 0.41}$	**86.47**

Table 1: T5-Base model on GLUE tasks, accuracy is metric, average over 3 seeds. Bold marks the best, underline marks the second best. GrabLoRA overall outperforms other baseline methods.

5 THEORY

In this section, we provide theoretical insight into GRABLORA from two perspectives. First, we analyze how the Grassmannian update tracks the rotating latent gradient subspace and preserves high coverage of the population gradient. Second, we show that an *idealized* update based on the projected population gradient yields descent under standard smoothness assumptions.

Assumption 5.1 (Model assumption)**.** Motivated by low-rank approximations of gradient matrices (Hu et al., 2021; Wang et al., 2024b), we assume that the population level gradient $G_t = -\mathbb{E}[\nabla \mathcal{L}(W_t)]$ lies in r-dimensional left and right subspaces. Their orthogonal projectors P_t^U and P_t^V satisfy $G_t = P_t^U G_t P_t^V$.

Notation. Let $\widehat{P}_t^U$ and $\widehat{P}_t^V$ be the projectors estimated by GRABLORA, which forms the idealized gradient evidence $\widehat{G}_t = \widehat{P}_t^U G_t + G_t \widehat{P}_t^V - \widehat{P}_t^U G_t \widehat{P}_t^V$. We measure the tracking errors $e_t^U = \|\widehat{P}_t^U - P_t^U\|_2$ and $e_t^V = \|\widehat{P}_t^V - P_t^V\|_2$. The corresponding per-step subspace drifts are $\delta_t^U = \|P_t^U - P_{t-1}^U\|_2$, $\delta_t^V = \|P_t^V - P_{t-1}^V\|_2$. We measure the perturbation between the empirical and population operators by $\xi_t^U = \|\frac{1}{n_t} Y_t \widehat{P}_t^V \bar{Y}_t^\top - \mathbb{E}[Y_{t,i} P_t^V Y_{t,i}^\top \mid \mathcal{F}_t^-]\|_2$. The spectral separation between the population subspace and its orthogonal complement is

$$\Delta_t^U = \lambda_{\min}(C_t C_t^\top + U_t^\top \mathbb{E}[E_{t,i} P_t^V E_{t,i}^\top \mid \mathcal{F}_t^-] U_t) - \lambda_{\max}(U_{t,\perp}^\top \mathbb{E}[E_{t,i} P_t^V E_{t,i}^\top \mid \mathcal{F}_t^-] U_{t,\perp})$$

The quantities for the right subspace are defined analogously. Define $d_{t,k} = (\sqrt{\mathrm{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-)} + \varepsilon)^{-1}$ as preconditioner. Range of those $d_{t,k}$ are $0 < d_- \le \min_{k\in[r]}\{1, d_{t,k}\} \le \max_{k\in[r]}\{1, d_{t,k}\} \le d_+$.

Theorem 5.2 (Coverage)**.** *If $\Delta_t^U > 0$ and $\xi_t^U + \frac{\kappa_U}{n_t}(e_{t-1}^U + \delta_t^U) < \frac{1}{2}(\Delta_t^U + \frac{\kappa_U}{n_t})$ and likewise for V, then*

$$\frac{\|G_t - \widehat{G}_t\|_F}{\|G_t\|_F} \le e_t^U e_t^V \le \frac{4\left[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\right]\left[\xi_t^V + \bar{\kappa}_V(e_{t-1}^V + \delta_t^V)\right]}{(\Delta_t^U + \bar{\kappa}_U)(\Delta_t^V + \bar{\kappa}_V)}$$

The proof is in Appendix B.2. Intuitively, the theorem shows that the coverage error is controlled by the product of the left and right subspace tracking errors, which increases with empirical perturbation ξ_t, accumulated tracking error e_{t-1}, and latent subspace drift δ_t, but decreases with the eigengap Δ_t, which measures how well the task-relevant subspace is separated from noise.

Theorem 5.3 (Convergence)**.** *Suppose the loss function $\mathcal{L}$ is L-smooth. If the learning rate η satisfies $0 < \eta \le \frac{d_-[1-(e_t^U e_t^V)^2]}{L d_+^2}$, then*

$$L(W_{t+1}) \le \mathcal{L}(W_t) - \frac{\eta d_-}{2}\left[1 - (e_t^U e_t^V)^2\right] \|\nabla \mathcal{L}(W_t)\|_F^2$$

The proofs are available in Appendix B.3. Intuitively, this result connects coverage to optimization. When the tracked subspaces cover the population gradient well, the product $e_t^U e_t^V$ is small, so the update direction preserves most of the descent information in $-\nabla \mathcal{L}(W_t)$.

[2]SubTrack is included for its relevance to subspace tracking. Its original focus on wall-clock training efficiency differs from our evaluation setting, so the reported results should be interpreted within our setup and hyperparameter search. Implementation details and the search ranges are provided in Appendix D.

Method	**GSM8K**	**Humaneval**	**MMLU**
Metric	*accuracy*	*pass@1*	*accuracy*
LoRA	$61.1_{\pm 3.2}$	$\underline{37.0}_{\pm 2.0}$	$61.5_{\pm 4.1}$
LoRA-GA	$62.8_{\pm 1.1}$	$30.3_{\pm 5.9}$	$60.1_{\pm 0.8}$
LoRA-One	$\underline{63.4}_{\pm 0.4}$	$34.6_{\pm 2.5}$	$\underline{64.0}_{\pm 0.6}$
CG-LoRA	$61.5_{\pm 0.9}$	$35.2_{\pm 2.5}$	$60.1_{\pm 0.0}$
ROSA	$63.3_{\pm 0.7}$	$35.1_{\pm 1.4}$	$60.6_{\pm 0.7}$
SubTrack	$52.8_{\pm 1.4}$	$29.9_{\pm 1.1}$	$59.0_{\pm 0.0}$
Ours	$\mathbf{64.2}_{\pm 0.8}$	$\mathbf{37.4}_{\pm 0.9}$	$\mathbf{65.3}_{\pm 0.1}$

Table 2: Performance of Llama-3.1-8B on GSM8K, Humaneval, MMLU tasks averaged over three seeds. Bold denotes the best, underline the second best. GrabLoRA consistently outperforms other baseline methods across these three tasks.

Model	Task	train time ours (s)	Peak GPU mem. (GB)	
			GrabLoRA	LoRA
T5-base	CoLA	317	0.93	0.90
	MRPC	318	1.25	1.25
	QNLI	750	5.06	5.06
	QQP	320	2.20	2.21
	RTE	319	2.15	2.16
	SST-2	453	2.01	2.02
Llama-3.1-8B	Math	8738	33.46	33.49
	Code	2047	33.50	33.54
	MMLU	1183	33.41	33.44

Table 3: Training time and peak GPU memory averaged over 3 seeds. GrabLoRA has similar storage savings as LoRA.

6 Experiments

Baselines. We compare GrabLoRA against seven baselines: the vanilla LoRA method (Hu et al., 2021) (**vanilla**), five representative LoRA variants and the full fine-tuning (**Full FT**) as a reference. To compare against gradient-based initialization-only methods, we include **LoRA-GA** (Wang et al., 2024b), **LoRA-One** (Zhang et al., 2025) and **CG-LoRA** (Zheng & Proutière, 2026). To compare against a method that augments low-rank updates with sparsity during training, we add **ROSA** (Nikdan et al., 2024). To compare against a method that tracks subspaces on the Grassmann manifold, we add **SubTrack** (Rajabi et al., 2025).

Tasks and metrics. We evaluate on two task families. For *natural language understanding*, we use a subset of the GLUE benchmarks (Wang et al., 2019) with accuracy as the metric: CoLA, SST-2, MRPC, QNLI, RTE, QQP. For *generative and instruction-following tasks*, we consider three settings: mathematical reasoning (train on 20K MetaMathQA (Yu et al., 2024) samples, evaluate on 1K GSM8K (Cobbe et al., 2021) samples, using accuracy), code generation (train on 20K Code-Feedback (Zheng et al., 2025) samples, evaluate on 164 HumanEval (Chen et al., 2021) problems using Pass@1), and instruction following (train on 20K Alpaca (Taori et al., 2023) samples, evaluate on 1K MMLU (Hendrycks et al., 2021) samples in the 5-shot setting, using accuracy). Detailed experimental configurations and training settings are provided in Appendix D.

Models and random seeds. For the GLUE tasks, we fine-tune the T5-base model (Raffel et al., 2023). For the generative and instruction-following tasks, we use Llama-3.1-8B (Grattafiori et al., 2024) unless otherwise stated. For a unified comparison across all methods, including the Full FT baseline, we update only the query and value projection matrices, while all other model parameters remain frozen. Unless otherwise stated, we report the mean and standard deviation over seeds 1,2,3.

7 Results

Performance. Table 1 reports the performance of fine-tuning on a T5-base model on six GLUE tasks, with a task-specific training budget as fully described in Appendix D.1. GrabLoRA achieves the highest score among parameter-efficient methods on CoLA, MRPC, QQP, RTE, SST-2, five out of the six tasks, with LoRA-One leading on QNLI. The average score is 86.47%, exceeding the strongest parameter-efficient baseline, LoRA-One, by 2.73% (a 3.26% relative improvement). GrabLoRA also achieves higher mean scores than Full FT on MRPC, QNLI, QQP, and SST-2 though the overall average remains slightly below full fine-tuning. Table 2 reports the performance of fine-tuning on Llama-3.1-8B model on mathematical reasoning (GSM8K), code generation (HumanEval), and knowledge evaluation (MMLU). GrabLoRA obtains the highest reported score on all three benchmarks: 64.2% on GSM8K, 37.4% on Humaneval, and 65.3% on MMLU, exceeding the strongest baseline on each task by 0.8%, 0.4%, and 1.3% point respectively.

Coverage. A central motivation of this paper is that the task-relevant gradient subspace may rotate during fine-tuning. If it does, a subspace chosen only at initialization may gradually lose its ability

Method	CoLA	MRPC	QNLI	QQP	RTE	SST-2	Avg
GrabLoRA	$79.50_{\pm 0.98}$	$87.17_{\pm 1.90}$	$92.01_{\pm 0.53}$	$89.70_{\pm 0.62}$	$76.77_{\pm 3.07}$	$93.69_{\pm 0.41}$	86.47
No Precond	$73.13_{\pm 1.06}$	$\underline{80.39}_{\pm 0.65}$	$\underline{86.27}_{\pm 0.40}$	$87.57_{\pm 0.55}$	$\mathbf{61.61}_{\pm 3.51}$	$89.14_{\pm 0.24}$	$\underline{79.69}$
Init freeze	$\mathbf{68.43}_{\pm 0.51}$	$80.88_{\pm 1.36}$	$\mathbf{82.13}_{\pm 0.06}$	$\underline{86.53}_{\pm 1.99}$	$72.20_{\pm 0.36}$	$\underline{87.00}_{\pm 0.70}$	**79.53**
no rediag	$73.07_{\pm 1.11}$	$83.09_{\pm 0.25}$	$88.03_{\pm 0.21}$	$87.67_{\pm 0.55}$	$74.49_{\pm 0.21}$	$89.07_{\pm 0.24}$	82.57
intermediate frozen	$77.27_{\pm 0.50}$	$86.76_{\pm 0.88}$	$87.10_{\pm 0.62}$	$\mathbf{77.87}_{\pm 2.21}$	$\underline{70.88}_{\pm 8.86}$	$\mathbf{84.06}_{\pm 3.46}$	80.66
no strips	$\underline{69.27}_{\pm 0.84}$	$\mathbf{77.45}_{\pm 1.70}$	$86.30_{\pm 1.54}$	$89.17_{\pm 1.02}$	$74.13_{\pm 1.85}$	$88.07_{\pm 0.23}$	80.73

Table 4: Ablation study of GrabLoRA across GLUE tasks, reporting accuracy (%) averaged over 3 seeds. Bold and underlined entries indicate the largest and second-largest decreases from the full GrabLoRA method within each column. Removing any component reduces the average performance across tasks.

to capture the gradients encountered later in training. To quantify this effect, we introduce *coverage*, the fraction of a reference gradient subspace captured by the update subspace estimator associated with a method. Let U^a and V^a denote orthogonal basis for method a's left and right subspaces, with projectors $P^{U,a} = U^a U^{a\top}$ and $P^{V,a} = V^a V^{a\top}$. For a reference gradient G^b, we define

$$\text{Coverage}(a, b) = \frac{\|P^{U,a} G^b + G^b P^{V,a} - P^{U,a} G^b P^{V,a}\|_F}{\|G^b\|_F} \in [0, 1]$$

Thus, a higher value means that method a's updated subspace capture more of the reference gradient G^b. In our experiments, G^b is the full fine-tuning gradient, which we use as a reference for the gradient directions available during training. Figure 3 in Section 4 further illustrates how coverage evolves over the course of training. More results are listed in Appendix E.

Complexity. We conduct the experiments using eight NVIDIA A100 GPUs, each with 40GB of memory. Table 3 reports the training time of GRABLORA in seconds, and compares its peak GPU memory usage with that of LoRA. Across all evaluated tasks, GRABLORA maintains peak GPU memory usage comparable to LoRA, indicating that tracking subspace rotation and maintaining uncertainty estimates introduce little additional peak memory overhead in these experiments.

8 DISCUSSION

In this paper, we first identify the *rotation* of the latent gradient subspace during training, an underexplored phenomenon that challenges the assumption that the directions found at initialization remain sufficient throughout training. Then, to answer the question: *How to keep track of this rotation, without discarding useful geometry already learned?* We develop GRABLORA, which incorporates rotation-aware subspace tracking and a preconditioner designed for the changing subspace into the training loop. Next, we report the ablation study and discuss future directions for GRABLORA.

Ablations. Table 4 reports the ablation results for the T5-base model on six GLUE tasks. Freezing the initial subspace (Init freeze), which sets $U_t = U_0$ and $V_t = V_0$ throughout training without updating and rotation, causes the largest average drop, from 86.47% to 79.53%. This suggests that updating the subspaces during training to track their rotation matters. Replacing the adaptive update with normalized steps gives the second lowest average score, at 79.69%. This variant replaces adaptive step size allocation with normalized steps, and its lower performance supports the benefit of the adaptive update used by GRABLORA. Without re-diagonalization (no rediag), the average falls to 82.57%. Updating the core only when the bases U_t and V_t are refreshed (intermediate frozen) reduces it to 80.66%, suggesting that evidence arriving between refreshes is useful even when the subspaces are held fixed. Removing the tangent strips (no strips), which replaces $\Delta W_t = U_t C_t V_t^\top + (I - P_t^U)\bar{Y}_t P_t^V + P_t^U \bar{Y}_t (I - P_t^V)$ with $\Delta W_t = U_t C_t V_t^\top$ during the update, reduces the average to 80.73%, showing that updates outside the current core also matter.

Limitations and future work. Several limitations suggest directions for further work. First, we currently refresh the subspaces on a fixed schedule, regardless of how quickly they move. An adaptive schedule could use measured rotation to refresh only when the geometry changes substantially, balancing performance against computation. Second, GRABLORA uses a fixed rank of $r = 8$ across layers, although different layers may need different capacities (Zhang et al., 2024; Yu et al., 2026). Therefore, examining the relationship between rotation and rank allocation during training can further guide dynamic rank allocation across layers in a geometry-informed manner.

STATEMENT OF LLM USAGE

In this work, we used generative AI tools for checking grammar and spelling in writing, as well as for experiment coding and visualization. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main paper provides a simplified version of the algorithm in Section 4. Detailed pseudocode is provided in Appendix A, and the code is included in the supplementary ZIP file. We list the key configurations in Appendix D. Theoretical proofs and calculation details are available in Appendix B.

ETHICS STATEMENT

This work studies parameter-efficient fine-tuning using publicly available datasets and pretrained models. It does not involve human subjects, private user data, or the release of a new dataset. We document the experimental settings and provide code and configuration details for reproducibility. Downstream applications should assess the safety, privacy, and bias risks of the adapted methods.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020. URL https://arxiv.org/abs/2012.13255.

Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information, 2011. URL https://arxiv.org/abs/1006.4046.

Laura Balzano, Tianjiao Ding, Benjamin D. Haeffele, Soo Min Kwon, Qing Qu, Peng Wang, Zhangyang Wang, and Can Yaras. An overview of low-rank structures in the training and adaptation of large models: From implicit low-dimensionality to efficient training and fine-tuning [special issue on the mathematics of deep learning]. *IEEE Signal Processing Magazine*, 43(3):52–70, May 2026. ISSN 1558-0792. doi: 10.1109/msp.2026.3666749. URL http://dx.doi.org/10.1109/MSP.2026.3666749.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, May 2010. ISSN 1573-0565. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4.

Cameron J. Blocker, Haroon Raja, Jeffrey A. Fessler, and Laura Balzano. Dynamic subspace estimation with grassmannian geodesics, 2023. URL https://arxiv.org/abs/2303.14851.

Chang Che, Ziqi Wang, Pengwan Yang, Qi Wang, Hui Ma, and Zenglin Shi. Lora in lora: Towards parameter-efficient architecture expansion for continual visual instruction tuning, 2026. URL https://arxiv.org/abs/2508.06202.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Yuejie Chi and Haoyu Fu. Subspace learning from bits, 2017. URL https://arxiv.org/abs/1407.6288.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 2026/09/20/ 1970. ISSN 00361429. URL http://www.jstor.org/stable/2949580. Full publication date: Mar., 1970.

Rémi Delogne and Laurent Jacques. Random features for grassmannian kernels, 2025. URL https://arxiv.org/abs/2504.21533.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Chendi Ge, Xin Wang, Zeyang Zhang, Hong Chen, Jiapei Fan, Longtao Huang, Hui Xue, and Wenwu Zhu. Dynamic mixture of curriculum loRA experts for continual multimodal instruction tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zpGK1bOlHt.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks, 2020. URL https://arxiv.org/abs/2004.10964.

Mehrtash T. Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C. Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR 2011*, pp. 2705–2712, 2011. doi: 10.1109/CVPR.2011.5995564.

Allen Hatcher. *Algebraic topology*. Cambridge university press, New York, 2001. ISBN 978-0-521-79540-1.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models, 2024. URL https://arxiv.org/abs/2402.12354.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. URL https://arxiv.org/abs/1902.00751.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018. URL https://arxiv.org/abs/1801.06146.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes, 2018. URL https://arxiv.org/abs/1804.08838.

Zongqian Li, Yixuan Su, Han Zhou, Zihao Fu, and Nigel Collier. Flexi-lora with input-adaptive ranks: Efficient finetuning for speech and reasoning tasks, 2026. URL https://arxiv.org/abs/2605.01959.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7), December 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40663-9. URL http://dx.doi.org/10.1007/s11704-024-40663-9.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models, 2025. URL https://arxiv.org/abs/2404.02948.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to bert embeddings during fine-tuning?, 2020. URL https://arxiv.org/abs/2004.14448.

Praneeth Narayanamurthy and Namrata Vaswani. Nearly optimal robust subspace tracking, 2018. URL https://arxiv.org/abs/1712.06061.

Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. Rosa: Accurate parameter-efficient fine-tuning via robust adaptation, 2024. URL https://arxiv.org/abs/2401.04679.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Sahar Rajabi and Sirisha Rambhatla. Optimizing fine-tuning efficiency: Gradient subspace tracking on grassmann manifolds for large language models. In *NeurIPS 2024 Workshop on Mathematics of Modern Machine Learning*, 2024. URL https://openreview.net/forum?id=ZP7VzpXmKR.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Subtrack++ : Gradient subspace tracking for scalable LLM training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6geRIdlFWJ.

Zephaniah Roe, Jack Sanderson, Dang Nguyen, Julian Huang, Todd Nief, Aryan Shrivastava, Chenhao Tan, and Ari Holtzman. Iterative finetuning is mostly idempotent, 2026. URL https://arxiv.org/abs/2605.01130.

Alex Saad-Falcon, Brighton Ancelin, and Justin Romberg. Subspace tracking with dynamical models on the grassmannian, 2024. URL https://arxiv.org/abs/2402.10352.

Etienne Salimbeni, Francesco Craighero, Renata Khasanova, Milos Vasic, and Pierre Vandergheynst. BEYOND FINE-TUNING: LORA MODULES BOOST NEAR- OOD DETECTION AND LLM SECURITY. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=H7Q5hHcvZE.

Hailong Shi, Hao Zhang, Gang Li, and Xiqin Wang. Stable embedding of grassmann manifold via gaussian random matrices, 2014. URL https://arxiv.org/abs/1312.3263.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation, 2023. URL https://arxiv.org/abs/2210.07558.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Rui Wang, Chen Hu, Ziheng Chen, Xiao-Jun Wu, and Xiaoning Song. A grassmannian manifold self-attention network for signal classification. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 5099–5107. International Joint Conferences on Artificial Intelligence Organization, 8 2024a. doi: 10.24963/ijcai.2024/564. URL https://doi.org/10.24963/ijcai.2024/564. Main Track.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via lora, 2025. URL https://arxiv.org/abs/2504.15777.

Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation, 2024b. URL https://arxiv.org/abs/2407.05000.

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning, 2024. URL https://arxiv.org/abs/2401.04151.

Yao Xie, Ruiyang Song, Hanjun Dai, Qingbin Li, and Le Song. Online supervised subspace tracking, 2015. URL https://arxiv.org/abs/1509.00137.

Haodong Yang, Lei Wang, and Md Zakir Hossain. Srlora: Subspace recomposition in low-rank adaptation via importance-based fusion and reinitialization, 2025. URL https://arxiv.org/abs/2505.12433.

Ryoma Yataka, Kazuki Hirashima, and Masashi Shiraishi. Grassmann manifold flows for stable shape generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=H2udtfMbl4.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024. URL https://arxiv.org/abs/2309.12284.

Xin Yu, Cong Xie, Xunmei Liu, Tiantian Fan, Lingzhou Xue, and Zhi Zhang. Train large, deploy compact: Structured compression for compact low-rank adaptation, 2026. URL https://arxiv.org/abs/2510.00192.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the davis–kahan theorem for statisticians, 2014. URL https://arxiv.org/abs/1405.0680.

Jiayao Zhang, Guangxu Zhu, Robert W. Heath Jr., and Kaibin Huang. Grassmannian learning: Embedding geometry awareness in shallow and deep learning, 2018. URL https://arxiv.org/abs/1808.02229.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. Loraprune: Structured pruning meets low-rank parameter-efficient fine-tuning, 2024. URL https://arxiv.org/abs/2305.18403.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning, 2023. URL https://arxiv.org/abs/2303.10512.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. Lora-one: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently, 2025. URL https://arxiv.org/abs/2502.01235.

Yuanfan Zhao. Improving code generation based on lora fine-tuning. *Applied and Computational Engineering*, 218:126–134, 12 2025. doi: 10.54254/2755-2721/2026.TJ30955.

Frédéric Zheng and Alexandre Proutière. Curvature-guided lora: Matching full fine-tuning in function space, 2026. URL https://arxiv.org/abs/2603.29824.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement, 2025. URL https://arxiv.org/abs/2402.14658.

CONTENTS

A Detailed Algorithm **15**

A.1 Notation . . . 15

A.2 model setup . . . 15

A.3 Tracking latent subspace . . . 16

A.4 Uncertainty-aware preconditioner . . . 16

A.5 The whole loop . . . 17

B Proof **18**

B.1 closed form of latent space optimization . . . 18

B.2 Coverage. . . . 20

B.3 Convergence . . . 23

C Figure details **26**

D Experiments Details **26**

D.1 Training details . . . 26

D.2 Hyperparameter Configuration . . . 27

D.3 Computation Resources . . . 27

D.4 Prompt Design . . . 27

E additional results **29**

E.1 Misalignment . . . 29

E.2 Coverage . . . 29

Notation	Meaning
W_t	weight matrix, (e.g. q v block) in transfomrer architecture
W_0	pretrained fixed know weight
r	rank
n_t	batch size of evidence collected at time t
κ_u, κ_v	Grassmann prior for U and V
q_0, q_1	baseline / rotation process-noise constants
β_1	first order momentum in Adam
η	learning rate
$j_{\max}$	blockwise optimization loops
ϵ	Adam floor
$Y_{t,i} = -\nabla\mathcal{L}(x_i, w_t)$	Gradient evidence from sample i at time t
$\mathcal{F}_t^- = \sigma(D_{0:t-1})$	all information at time t before this step evidence
$\mathcal{F}_t^+ = \sigma(D_{0:t})$	all information at time t before this step evidence
$U_t = [u_{t,1} \dots u_{t,r}]$, V_t	whose columns span the left, right subspaces
C_t	diaganal matrix connecting left and right subspace U_t and V_t
P_t^U, P_t^V	subspace projectors
$A_{t,k} = u_{t,k} v_{t,k}^\top$	k^{th} rank-one matrix direction
$z_{t,i,k} = u_{t,k}^\top Y_{t,i}\, v_{t,k}$	projected coordinate of sample i on $A_{t,k}$
$\beta_{t,k}$	coordinate precision $= 1/\operatorname{var}(z_{t,i,k})$
$R_{t,k} = (n\beta_{t,k})^{-1}$	noise variance of the averaged coordinate $\bar{z}_{t,k}$
$\rho_{t,k} \in [0,1]$	overlap of $A_{t,k}$ with its previous direction $A_{t-1,k}$
$q_{t,k} = q_0 + q_1(1-\rho_{t,k}^2)$	rotation-driven variance
$P_{t,k}^-$	prior uncertainty of $c_{t,k}$ at time t before receiving evidence from $\bar{Y}_t$
$P_{t,k}^+$	posterior uncertainty of $c_{t,k}$ after receiving evidence from $\bar{Y}_t$
$s_{t,k} = P_{t,k}^- + R_{t,k}$	prior predictive variance of $\bar{z}_{t,k}$
$\mu_{t,k}$, $\hat{\mu}_{t,k}$	first moment in Adam, and its unbiased form
ΔW_t	proposed weight update at time t
$\mathcal{C}$	refresh set, steps in $\mathcal{C}$ will update rotation of space
$\langle , \rangle_F$	Frobenius norm
$\langle , \rangle_2$	spectral norm

Table 5: Notations in this paper.

A Detailed Algorithm

A.1 Notation

Table 5 presents the notation in this paper.

A.2 model setup

GrabLoRA tracks the rotating rank-r subspace with changing cordinates. Therefore, after basis rotation, the cordiates of each evidence lives also change. The following model tries to build the rotation and connection at time $t-1$ and time t.

Suppose $Y_{t,i} = -\nabla_w\mathcal{L}(x_i; w_t) = G_t + E_{t,i}$ can be captured by a low dimensional population gradient $G_t = \mathbb{E}_{x_i}[-\nabla_w\mathcal{L}(x_i; w_t)]$ with a white noise term $E_{t,i}$.

$$Y_{t,i} = G_t^\top + E_{t,i},\ z_{t,i,k} = \langle A_{t,k}, Y_{t,i}\rangle_F = c_{t,k} + e_{t,i,k},\ \mathbb{E}[e_{t,i,k}] = 0,\ \operatorname{Var}(e_{t,i,k}) = \beta_{t,k}^{-1}, \quad \text{(A.1)}$$

Where $c_{t,k} = \langle A_{t,k}, G_{t,k}\rangle_F$, $e_{t,i,k} = \langle A_{t,k}, E_{t,i,k}\rangle_F$. Denote $C_t = \operatorname{diag}(c_{t,1}, \dots, c_{t,r})$ and $e_{t,i,k}$ uncorrelated across i with averaged noise variance $R_{t,k} = (n\hat{\beta}_{t,k})^{-1}$. Here, $c_{t,k}$ is the latent *coefficient* along direction $A_{t,k}$ and $z_{t,i,k}$ is the empirical *evidence* of this coefficient $c_{t,k}$ from sample i. Moreover, the averaged coordinate $\bar{z}_{t,k}$ is an estimate of $c_{t,k}$, the latent population gradient onto the current rotated rank-1 coordinate $A_{t,k}$. We further assume each $c_{t,k}$ weight varies smoothly with the subspace rotation,

$$c_{t,k} = \rho_{t,k}\, c_{t-1,k} + w_{t,k}, \qquad \mathbb{E}[w_{t,k}] = 0, \quad \operatorname{Var}(w_{t,k}) = q_{t,k} = q_0 + q_1(1-\rho_{t,k}^2), \quad \text{(A.2)}$$

So a direction that barely turns ($\rho_{t,k} \to 1$) inherits its previous geometry uncertainty, whereas a direction that turns sharply is treated as newly uncertain. This uncertainty is what preconditions.

After the setup, we can derive the variance update. At time t before receiving the incoming evidence $Y_{t,i}$ for $i \in \{1, \dots, n_t\}$,

$$P^-_{t,k} = \mathrm{var}(c_{t,k} \mid \mathcal{F}^-_t) = \rho^2_{t,k}\,\mathrm{var}(c_{t-1,k} \mid \mathcal{F}^-_t) + \mathrm{var}(w_{t,k} \mid \mathcal{F}^-_t) = \rho^2_{t,k} P^+_{t-1,k} + q_{t,k}, \quad \text{(A.3)}$$

and when new evidence update the $c_{t,k}$, using the linear minimum mean-square error principle.

$$\left(P^+_{t,k}\right)^{-1} = \left(P^-_{t,k}\right)^{-1} + n\hat{\beta}_{t,k}. \quad \text{(A.4)}$$

A.3 Tracking latent subspace

To prevent over-fitting, and to update the calculation speed and efficiency, we update the rotation space in s refresh set $\mathcal{C}$. At a refresh $t \in \mathcal{C}$ we update the projectors by trading current evidence against geometric continuity,

$$J(P^U, P^V) = \sum_{i=1}^{n} \left\|Y_{t,i} - P^U Y_{t,i} P^V\right\|^2_F + \tfrac{\kappa_U}{2}\left\|P^U - P^U_{t-1}\right\|^2_F + \tfrac{\kappa_V}{2}\left\|P^V - P^V_{t-1}\right\|^2_F \quad \text{(A.5)}$$

over $P^U = UU^\top$, $P^V = VV^\top$. This is nonconvex optimization. However, if we update P^U and P^V iteratively (blockwise in following discussion), then this has closed form. Fixing P^V, the U-problem is with closed form solution:

$$\max_U \mathrm{tr}[UU^\top(\sum_i Y_i P^V Y_i^\top + \kappa_U P^U_{t-1})] \quad \text{(A.6)}$$

whose solution is the top-r left singular vectors of $[\,Y_1 V \;\cdots\; Y_n V \mid \sqrt{\kappa_U}\,U_{t-1}\,]$. This is symmetrically for V. Calculation details are available in Apendix B.1.

We initialize the iterative optimization at (U_{t-1}, V_{t-1}) and perform up to $j_{\max}$ iterations, obtaining the subspace projectors P^U_t, P^V_t and corresponding orthonormal bases $U^{\mathrm{raw}}_t, V^{\mathrm{raw}}_t$. Although the optimization determines the subspaces, their bases remain free to rotate within them. Consequently, the projected evidence has a generally non-diagonal core:

$$\widehat{Y}_t = P^U_t \bar{Y}_t P^V_t = U^{\mathrm{raw}}_t C^{\mathrm{raw}}_t V^{\mathrm{raw}\top}_t, \qquad C^{\mathrm{raw}}_t = U^{\mathrm{raw}\top}_t \bar{Y}_t V^{\mathrm{raw}}_t.$$

Here, $\widehat{Y}_t$ is the projection of the batch evidence onto the estimated left and right subspaces and has rank at most r. To express this evidence through r paired rank-1 directions, we re-diagonalize the core using its singular value decomposition:

$$C^{\mathrm{raw}}_t = Q_U C_t Q_V^\top, \qquad Q_U^\top Q_U = Q_V^\top Q_V = I_r, \qquad C_t = \mathrm{diag}(c_{t,1}, \dots, c_{t,r}),$$

Defining $U_t = U^{\mathrm{raw}}_t Q_U$ and $V_t = V^{\mathrm{raw}}_t Q_V$, we obtain

$$\widehat{Y}_t = U^{\mathrm{raw}}_t C^{\mathrm{raw}}_t V^{\mathrm{raw}\top}_t = (U^{\mathrm{raw}}_t Q_U)\left(Q_U^\top C^{\mathrm{raw}}_t Q_V\right)(V^{\mathrm{raw}}_t Q_V)^\top = U_t C_t V_t^\top \quad \text{(A.7)}$$

This re-diagonalization preserves the estimated Grassmann points because $U_t U_t^\top = U^{\mathrm{raw}}_t Q_U Q_U^\top U^{\mathrm{raw}\top}_t = U^{\mathrm{raw}}_t U^{\mathrm{raw}\top}_t = P^U_t$, indicating U_t and U^{raw}_t span the same space. Selecting data-adapted directions $A_{t,k} = u_{t,k} v_{t,k}^\top$ satisfies $\bar{z}_{t,k} = \langle A_{t,k}, \bar{Y}_t\rangle_F = c_{t,k} + \bar{e}_{t,k}$. Thus, the re-diagonalization acts on the empirical core and leaves the strip terms unchanged.

Adding constraints of core matrix to be diagonal resolves the non-identifiability of Grassmann optimization and distinguish each rank-1 matrix direction $A_{t,k} = u_{t,k} v_{t,k}^\top$. And we can measure the closedness of current k^{th} direction with the last step:

$$\rho_{t,k} = \left|\langle u_{t-1,k} v_{t-1,k}^\top,\, u_{t,k} v_{t,k}^\top\rangle\right| = \left|(U_{t-1}^\top U_t)_{kk}\,(V_{t-1}^\top V_t)_{kk}\right| \in [0,1]. \quad \text{(A.8)}$$

A.4 Uncertainty-aware preconditioner

Since the basis varies per updating, therefore, each basis balances between old geometry with new coming evidence. The new coming evidence brings uncertainty which contain huge information. Our design for precondition is stated as below: The predictive variance of the new incoming averaged coordinates is achieved when

$$s_{t,k} = \mathrm{var}(\bar{z}_{t,k} \mid \mathcal{F}^-_t) = P^-_{t,k} + R_{t,k} = \rho^2_{t,k} P^+_{t-1,k} + q_0 + q_1(1 - \rho^2_{t,k}) + (n_t \beta_{t,k})^{-1} \quad \text{(A.9)}$$

Algorithm 2 GRABLORA (single run with indices dataset d, seed s, layer l suppressed)

Require: pretrained W_0, rank r, refresh set $\mathcal{C}$, evidence budget n_t, Grassmann prior κ_u, κ_v, Block-wise optimization loops $j_{\max}$, process noise q_0, q_1, momentum β_1, floor ϵ, step cap $c_{\max}$, learning rate η

$\mu_{0,k} \leftarrow 0$
for $t = 0$ **to** T **do**
  $Y_{t,i} \leftarrow -\nabla_w \mathcal{L}(x_i; w_t)$; $\bar{Y}_t \leftarrow \frac{1}{n}\sum_i Y_{t,i}$ ▷ n=1 if $t \notin \mathcal{C}$
  (a) Subspace
  if $t = 0$ **then**
    $(U_0, \Sigma_0, V_0) \leftarrow \text{TRUNCSVD}(\bar{Y}_0, r)$; $\rho_{0,k} \leftarrow 1$ ▷ Eq. A.12
  else if $t \in \mathcal{C}$ **then**
    $(P^U, P^V) \leftarrow \arg\min J$ (Eq. A.5); $(U_t, V_t) \leftarrow$ re-diagonalize (Eq. A.7); $\rho_{t,k}$ (Eq. A.8)
  else
    $(U_t, V_t) \leftarrow (U_{t-1}, V_{t-1})$; $\rho_{t,k} \leftarrow 1$ ▷ Eq. A.14
  end if
  (b) Preconditioner
    (b.1) incoming evidence
    $z_{t,i,k} \leftarrow u_{t,k}^\top Y_{t,i} v_{t,k}$; $\beta_{t,k} \leftarrow 1/\operatorname{var}(z_{t,i,k})$; $R_{t,k} \leftarrow (n\beta_{t,k})^{-1}$ ▷ Eq. A.1
    (b.2) uncertainty propagation
    $q_{t,k}$ (Eq. A.2); $P^-_{t,k}$ (Eq. A.3); $s_{t,k}$ (Eq. A.9); $P^+_{t,k}$ (Eq. A.4)
    (b.3) preconditioned update
    $\mu_{t,k}$; $\hat{\mu}_{t,k}$; $c_{t,k} \leftarrow \text{clip}\big(\hat{\mu}_{t,k}/(\sqrt{s_{t,k}} + \epsilon), \pm 1/c_{\max}\big)$; ▷ Eq. A.10
  (c) Weight proposal
  $\Delta W_t \leftarrow U_t C_t V_t^\top + (I - P_t^U)\bar{Y}_t P_t^V + P_t^U \bar{Y}_t (I - P_t^V)$ ▷ Eq. A.11
  $w_{t+1} \leftarrow w_t + \eta\, \Delta W_t$
end for
return w_{T+1}

measures how the weight of each direction $A_{t,k} = u_{t,k} v_{t,k}^\top$ could have moved plus how noisy this step's estimate is. If the newly rank-1 direction $A_{t,k}$ captures most of the new evidences' directions, then $\bar{e}_{t,k}$ has a lower variance. $\operatorname{var}(\bar{e}_{t,k}) = R_{t,k} = (n_t \beta_{t,k})^{-1}$ is small, hence $s_{t,k}$ is small. By dividing $s_{t,k}$ in place of Adam's empirical second moment:

$$\mu_{t,k} = \beta_1 \mu_{t-1,k} + (1-\beta_1)\bar{z}_{t,k}, \quad \hat{\mu}_{t,k} = \frac{\mu_{t,k}}{1-\beta_1^t}, \quad \hat{c}_{t,k} = \text{clip}\Big(\frac{\hat{\mu}_{t,k}}{\sqrt{s_{t,k}} + \epsilon}, \pm\tfrac{1}{c_{\max}}\Big) \tag{A.10}$$

We realize the uncertainty aware precondition. And the whole updating assembly to be:

$$\Delta W_t \;= \sum_{k=1}^{r} \hat{c}_{t,k} A_{t,k} + \; (I - P_t^U)\bar{Y}_t P_t^V + P_t^U \bar{Y}_t (I - P_t^V) \tag{A.11}$$

which gives a rotation-aware preconditioner. If a direction remains stable and the batch evidence is reliable (low variance), then $\operatorname{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-)$ is small and GRABLORA takes a confident step. If the direction rotates sharply or the projected evidence is noisy, $\operatorname{var}(\bar{z}_{t,k} \mid \mathcal{F}_t^-)$ increases and the coefficient update becomes more conservative. Thus, the step size is controlled not only by gradient magnitude (numerator), but also by how trustworthy the current coordinate is under the evolving subspace.

A.5 THE WHOLE LOOP

Initialization. There is no previous subspace at $t = 0$, so the frames are seeded directly from the evidence and each mode weight is given one observation's worth of prior uncertainty. We take the truncated top-r SVD of the averaged evidence,

$$\widehat{Y}_0 = U_0 \Sigma_0 V_0^\top, \qquad (U_0, V_0, \Sigma_0) = \text{TRUNCSVD}(\bar{Y}_0, r), \tag{A.12}$$

$\Sigma_0 = \operatorname{diag}\{\sigma_{0,1}, \ldots, \sigma_{0,r}\}$. Projection onto U_0 and V_0 form the coordinates $z_{0,i,k} = u_{0,k}^\top Y_{0,i} v_{0,k}$, and estimate the precision $\hat{\beta}_{0,k}$ from their sample variance. We then set

$$\rho_{0,k} = 1, \qquad P^+_{0,k} = P^-_{0,k} = (n\hat{\beta}_{0,k})^{-1}, \qquad \mu_{0,k} = 0, \tag{A.13}$$

and take a single coefficient step: $s_{0,k} = P^-_{0,k} + R_{0,k} = 2(n\hat{\beta}_{0,k})^{-1}$ by Equation A.9, then $c_{0,k}$ by Equation A.10 (with $\hat{\mu}_{0,k} = \bar{z}_{0,k}$ after bias correction) and $P^+_{0,k}$ by Equation A.4. The singular

values $\sigma_{0,k}$ enter only through the initial frames. The coefficient starts from the momentum, not from $\sigma_{0,k}$.

Frozen steps. Between refreshes the subspace is held fixed and a single full-batch gradient is used, so no rotation and no per-sample spread are computed:

$$U_t = U_{t-1}, \quad V_t = V_{t-1}, \quad \rho_{t,k} = 1, \quad n = 1, \quad \hat{\beta}_{t,k} = \hat{\beta}_{t-1,k}, \quad q_{t,k} = q_0. \tag{A.14}$$

Because $\rho_{t,k} = 1$, the prediction Equation A.3 reduces to $P^-_{t,k} = P^+_{t-1,k} + q_0$ and the predictive variance to $s_{t,k} = P^-_{t,k} + \hat{\beta}^{-1}_{t,k}$. The coordinate is read from the single gradient, $\bar{z}_{t,k} = u^\top_{t,k} \bar{Y}_t v_{t,k}$, the momentum still advances, and $c_{t,k}, P^+_{t,k}$ follow Equation A.10,Equation A.4 unchanged. On $t \notin \mathcal{C}$ the subspace is fixed and a single full-batch gradient is used: n=1, $\rho_{t,k}$=1, $q_{t,k}$=q_0, $\hat{\beta}_{t,k}$ carried from the last refresh. The momentum still advances and Equation A.10, Equation A.4 apply unchanged.

Whole loop. Each step draws evidence $Y_{t,i} = -\nabla_w \mathcal{L}(x_i; w_t)$ and forms $\bar{Y}_t = \frac{1}{n}\sum_i Y_{t,i}$. On a refresh $t \in \mathcal{C}$ the subspace is rotated by Equation A.5, re-diagonalized Equation A.7, and the per-atom overlap $\rho_{t,k}$ read off by Equation A.8, otherwise the frozen shortcut Equation A.14 applies. Given the frames, coordinates $z_{t,i,k}$ and precision $\hat{\beta}_{t,k}$ fix the noise $R_{t,k} = (n\hat{\beta}_{t,k})^{-1}$. The uncertainty is propagated by Equation A.3, Equation A.4 to give the preconditioner $s_{t,k}$ Equation A.9. The coefficient $m_{t,k}$ follows from Equation A.10 and the proposal Equation A.11 adds the filtered core and the two tangent strips before the weight step. Initialization Equation A.12, Equation A.13 supplies the $t = 0$ state. Algorithm 2 collects the loop.

The coefficients fill the tracked core; the two first-order tangent strips add the part of the evidence just outside the subspace (the doubly-orthogonal block is dropped):

$$\Delta W_t = \sum_{k=1}^{r} \hat{c}_{t,k} A_{t,k} + (I - P^U_t)\bar{Y}_t P^V_t + P^U_t \bar{Y}_t (I - P^V_t), \qquad w_{t+1} = w_t + \eta\, \Delta W_t. \tag{A.15}$$

B Proof

B.1 Closed form of latent space optimization

After rediagonalization and reordering at each step, columns of U_t and U_{t-1} are consistently matched, which makes it convenient to consider the specific k^{th} dimension. As designed, we try to minimize the loss function $J(P^U, P^V)$ which balances between the current evidence $Y_{t,i}$ and the last step geometry.

Proposition B.1 (Loss function closed form)**.** *Let $Y_{t,i} \in \mathbb{R}^{p\times q}$, $i = 1, \ldots, n$, and let $P^U \in \mathbb{R}^{p\times p}$, $P^V \in \mathbb{R}^{q\times q}$ range over rank-r orthogonal projectors, Consider*

$$J(P^U, P^V) = \sum_{i=1}^{n} \left\| Y_{t,i} - P^U Y_{t,i} P^V \right\|_F^2 + \frac{\kappa_U}{2} \left\| P^U - P^U_{t-1} \right\|_F^2 + \frac{\kappa_V}{2} \left\| P^V - P^V_{t-1} \right\|_F^2 \tag{B.1}$$

If P^U and P^V are updated alternately, each subproblem admits a closed-form solution given by a top-r eigendecomposition.

Proof. Fix i, by definition of Frobenius norm, and utilize the properties of projection matrix,

$$\begin{aligned}
\left\| Y_{t,i} - P^U Y_{t,i} P^V \right\|_F^2 &= \|Y_{t,i}\|_F^2 - 2\,\mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \left\| P^U Y_{t,i} P^V \right\|_F^2 \\
&= \|Y_{t,i}\|_F^2 - 2\,\mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \mathrm{tr}\big(P^U Y_{t,i} P^V (P^V)^\top Y_{t,i}^\top (P^U)^\top\big) \\
&= \|Y_{t,i}\|_F^2 - 2\,\mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \mathrm{tr}\big(P^U Y_{t,i} P^V Y_{t,i}^\top\big) \\
&= \|Y_{t,i}\|_F^2 - 2\,\mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) \\
&= \|Y_{t,i}\|_F^2 - \mathrm{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big)
\end{aligned} \tag{B.2}$$

For any rank-r orthogonal projector P and P_{t-1}, $\mathrm{tr}(P^\top P) = \mathrm{tr}(P) = r$, so

$$\|P - P_{t-1}\|_F^2 = \mathrm{tr}(P) + \mathrm{tr}(P_{t-1}) - 2\,\mathrm{tr}(P P_{t-1}) = 2r - 2\,\mathrm{tr}(P P_{t-1})$$

Therefore, suppose P^V is fixed and collect the P^U-dependent term:

$$\begin{aligned}
\arg\min_{P^U} J(P^U) &= \arg\min_{P^U} \sum_{i=1}^{n} \left\| Y_{t,i} - P^U Y_{t,i} P^V \right\|_F^2 \\
&= \arg\min_{P^U} \left\{ \sum_{i=1}^{n} \|Y_{t,i}\|_F^2 - \sum_{i=1}^{n} \operatorname{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \frac{\kappa_U}{2}\big(2r - 2\operatorname{tr}(P^U P_{t-1}^U)\big) \right\} \\
&= \arg\min_{P^U} \left\{ -\sum_{i=1}^{n} \operatorname{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) - \kappa_U \operatorname{tr}(P^U P_{t-1}^U) \right\} \\
&= \arg\max_{P^U} \left\{ \sum_{i=1}^{n} \operatorname{tr}\big(Y_{t,i}^\top P^U Y_{t,i} P^V\big) + \kappa_U \operatorname{tr}(P^U P_{t-1}^U) \right\} \\
&= \arg\max_{P^U} \left\{ \sum_{i=1}^{n} \operatorname{tr}\big(P^U Y_{t,i} P^V Y_{t,i}^\top\big) + \kappa_U \operatorname{tr}(P^U P_{t-1}^U) \right\} \\
&= \arg\max_{P^U} \left\{ \operatorname{tr}\big[P^U \big(\sum_{i=1}^{n} Y_{t,i} P^V Y_{t,i}^\top + \kappa_U P_{t-1}^U \big)\big] \right\} \\
&:= \arg\max_{P^U} \left\{ \operatorname{tr}\big(P^U M_t^U\big) \right\}
\end{aligned} \tag{B.3}$$

Lemma B.2 (Rayleigh-Ritz / Ky Fan maximum principle)**.** *Let $M \in \mathbb{R}^{d\times d}$ be symmetric matrix, with eigenvalues $\lambda_1(M) \geq \cdots \geq \lambda_d(M)$, then*

$$\max_{\substack{U\in\mathbb{R}^{d\times r} \\ U^\top U = I_r}} \operatorname{tr}(U^\top M U) = \sum_{j=1}^{r} \lambda_j(M)$$

Maximum is attained when $U = [q_1, \ldots, q_r]$ where $q_1, \ldots, q_r$ are eigen vectors corresponding to the largest r eigenvalues of M.

Since $M_t^U = \sum_{i=1}^n Y_{t,i} P^V Y_{t,i}^\top + \kappa_U P_{t-1}^U$ is symmetric,

$$\max_{U_t} \operatorname{tr}(P^U M_t^U) = \max_{U_t} \operatorname{tr}(U_t U_t^\top M_t^U) = \max_{U_t} \operatorname{tr}(U_t^\top M_t^U U_t) \tag{B.4}$$

is attaiend when $U_t = [u_{t,1}, \ldots, u_{t,r}]$ with $u_{t,1}, \ldots, u_{t,r}$ top r eigen vectors of M_t^U. Since

$$\begin{aligned}
M_t^U &= \sum_{i=1}^{n} Y_{t,i} P_{t-1}^V Y_{t,i}^\top + \kappa_U P_{t-1}^U \\
&= \sum_{i=1}^{n} \left(Y_{t,i} V_{t-1}\right)\left(Y_{t,i} V_{t-1}\right)^\top + \left(\sqrt{\kappa_U} U_{t-1}\right)\left(\sqrt{\kappa_U} U_{t-1}\right)^\top \\
&= \left[\, Y_{t,1} V_{t-1} \;\cdots\; Y_{t,n} V_{t-1} \mid \sqrt{\kappa_U}\, U_{t-1} \,\right]\left[\, Y_{t,1} V_{t-1} \;\cdots\; Y_{t,n} V_{t-1} \mid \sqrt{\kappa_U}\, U_{t-1} \,\right]^\top \\
&:= B_t B_t^\top
\end{aligned} \tag{B.5}$$

Doing SVD over $B_t^U \in \mathbb{R}^{p\times(nq+p)}$:

$$B_t^U = Q \Sigma R^\top, \quad M_t^U = B_t^U B_t^{U\top} = Q \Sigma \Sigma^\top Q^\top$$

So M_t^U's first r eigen vectors are the first r columns of Q. Therefore,

$$U_t = \left(\text{left singular vectors} \left(\left[\, Y_{t,1} V_{t-1} \;\cdots\; Y_{t,n} V_{t-1} \mid \sqrt{\kappa_U}\, U_{t-1} \,\right] \right) \right)_{[:,1:r]}$$

is what we want.

By iteratively setting

$$\begin{aligned}
U_t &= \left(\text{left singular vectors} \left(\left[\, Y_{t,1} V_{t-1} \;\cdots\; Y_{t,n} V_{t-1} \mid \sqrt{\kappa_U}\, U_{t-1} \,\right] \right) \right)_{[:,1:r]} \\
V_t &= \left(\text{left singular vectors} \left(\left[\, Y_{t,1}^\top U_{t-1} \;\cdots\; Y_{t,n}^\top U_{t-1} \mid \sqrt{\kappa_U}\, V_{t-1} \,\right] \right) \right)_{[:,1:r]}
\end{aligned}$$

the loss function $J(P^U, P^V)$ is decreasing. Since J is multi-objective nonconvex function, we utilize the closed form of blockwise updating in code. □

B.2 COVERAGE.

Different from Appendix A, here we distinguish the unobserved population parameter U_t, P_t^U and the observed estimate $\widehat{U}_t$, $\widehat{P}_t^U$ with hat notation. GRABLORA obtains $\widehat{P}_t^U, \widehat{P}_t^V$ as the rank-r minimizers of the temporally regularized Grassmann objective from Equation B.1. Denote the idealized estimator of unobservable G_t is:

$$\begin{aligned}\widehat{G}_t &= \widehat{P}_t^U G_t + G_t \widehat{P}_t^V - \widehat{P}_t^U G_t \widehat{P}_t^V \\ &= \widehat{P}_t^U G_t \widehat{P}_t^V + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V)\end{aligned}$$

Definition B.3 (Estimation error)**.** Estimation error e_t^U measures the distance (principal angle) between the estimated space $\widehat{P}_t^U$ and unobserved P_t^U.

$$e_t^U = \|\widehat{P}_t^U - P_t^U\|_2 = \sin\theta_{\max}(\widehat{\mathcal{U}}_t, \mathcal{U}_t)$$

Definition B.4 (Latent space drift)**.** The unobserved real drift between spaces at different time P_{t-1}^U and P_t^U.

$$\delta_t^U = \|P_t^U - P_{t-1}^U\|_2$$

Definition B.5 (Operator and its perturbation)**.** Recall in Equation B.5, $M_t^U = \sum_{i=1}^n Y_{t,i} P_{t-1}^V Y_{t,i}^\top + \kappa_U P_{t-1}^U$, the former part we denote it with A_t^U. To be specific: A_t^U converts the noisy matrix-valued gradient into an operator whose leading (top-r) eigenspace and its empirical estimate $\widehat{A}_t^U$.

$$A_t^U = \mathbb{E}\big[Y_{t,i}\, P_t^V\, Y_{t,i}^\top \mid \mathcal{F}_t^-\big], \quad \widehat{A}_t^U = \tfrac{1}{n}\sum_{i=1}^n Y_{t,i}\, \widehat{P}_t^V\, Y_{t,i}^\top$$

The distance between A_t^U and $\widehat{A}_t^U$ is:

$$\xi_t^U = \|\widehat{A}_t^U - A_t^U\|_2$$

Definition B.6 (Subspace-preserving projected noise)**.** $\Omega_t = \mathbb{E}[E_{t,i} P_t^V E_{t,i}^\top \mid \mathcal{F}_t^-]$

Lemma B.7 (Alignment)**.** *Assume the spectral separation conditions*

$$\Delta_t^U = \lambda_{min}(C_t C_t^\top + U_t^\top \Omega_t^U U_t) - \lambda_{max}(U_{t,\perp}^\top \Omega_t^U U_{t,\perp}) > 0 \tag{B.6}$$

Then the leading r-dimensional eigen space of A_t^U is $\mathrm{range}(U_t)$.

Proof. $Y_{t,i} = G_t + E_{t,i}$ gives:

$$\begin{aligned}A_t^U &= \mathbb{E}[(G_t + E_{t,i}) P_t^V (G_t + E_{t,i})^\top \mid \mathcal{F}_t^-] \\ &= G_t P_t^V G_t^\top + G_t P_t^V \mathbb{E}[E_{t,i}^\top \mid \mathcal{F}_t^-] + \mathbb{E}[E_{t,i} \mid \mathcal{F}_t^-] P_t^V G_t^\top + \Omega_t^U \\ &= G_t P_t^V G_t^\top + \Omega_t^U \\ &= G_t G_t^\top + \Omega_t^U \\ &= U_t C_t V_t^\top V_t C_t^\top U_t^\top + \Omega_t^U \\ &= U_t C_t C_t^\top U_t^\top + \Omega_t^U\end{aligned}$$

Suppose columns of $U_{t,\perp}$ are the orthonormal basis of the space $(P_t^U)^\perp$, then in the orthonormal basis $[\,U_t\ U_{t,\perp}\,]$, A_t^U can be decomposed to

$$A_t^U = [\,U_t\ U_{t,\perp}\,] \begin{bmatrix} C_t C_t^\top + U_t^\top \Omega_t^U U_t & 0 \\ 0 & U_{t,\perp}^\top \Omega_t^U U_{t,\perp} \end{bmatrix} [\,U_t\ U_{t,\perp}\,]^\top \tag{B.7}$$

By Equation B.6, every eigenvalue of the first block is larger than every eigenvalue of the second block. Therefore, the leading r-dimensional eigen space of A_t^U is $\mathrm{range}(U_t)$. □

Corollary B.8. *Let*

$$M_t^U = A_t^U + \bar{\kappa}_U P_t^U, \qquad \widehat{M}_t^U = \widehat{A}_t^U + \bar{\kappa}_U \widehat{P}_{t-1}^U$$

The eigen gap of M_t^U is

$$\lambda_r(M_t^U) - \lambda_{r+1}(M_t^U) = \Delta_t^U + \bar{\kappa}_U$$

Proof. From the decomposition of Equation B.7, A_t^U's eigen space are reresented by columns of $[\, U_t \; U_{t,\perp} \,] = [\, u_{t,1}, \ldots, u_{t,r}, u_{t,r+1}, \ldots, u_{t,p} \,]$, therefore, the spectral decomposition of A_t^U is:

$$A_t^U = \sum_{j=1}^{p} \lambda_j(A_t^U) u_{t,j} u_{t,j}^\top$$

with $\lambda_1(A_t^U) \geq \cdots \geq \lambda_p(A_t^U)$. Since $P_t^U = U_t U_t^\top = \sum_{j=1}^{r} u_{t,j} u_{t,j}^\top$,

$$\begin{aligned} M_t^U &= A_t^U + \bar{\kappa}_U P_t^U \\ &= \sum_{j=1}^{p} \lambda_j(A_t^U) u_{t,j} u_{t,j}^\top + \bar{\kappa}_U \sum_{j=1}^{r} u_{t,j} u_{t,j}^\top \\ &= \sum_{j=1}^{r} \big[\lambda_j(A_t^U) + \bar{\kappa}_U\big] u_{t,j} u_{t,j}^\top + \sum_{j=r+1}^{p} \lambda_j(A_t^U) u_{t,j} u_{t,j}^\top \end{aligned} \tag{B.8}$$

Therefore, the eigenvalues of M_t^U forms a gap:

$$\lambda_j(M_t^U) = \begin{cases} \lambda_j(A_t^U) + \bar{\kappa}_U, & 1 \leq j \leq r \\ \lambda_j(A_t^U), & r+1 \leq j \leq p \end{cases}$$

with $\bar{\kappa}_U \geq 0$. Hence, P_t^U is the leading rank-r spectral projector of M_t^U and the eigen pag between

$$\gamma_t^U = \lambda_r(M_t^U) - \lambda_{r+1}(M_t^U) = \lambda_r(A_t^U) + \bar{\kappa}_U - \lambda_{r+1}(A_t^U) = \Delta_t^U + \bar{\kappa}_U > 0 \tag{B.9}$$

Intuition: Since P_t^U is the leading rank-r spectral projector of A_t^U. Adding $\bar{\kappa}_U P_t^U$ increases each of the leading r eigenvalues by $\bar{\kappa}_U$ while leaving the remaining eigenvalues unchanged. Consequently, P_t^U is also the leading rank-r spectral projector of M_t^U. □

Lemma B.9 (Davis–Kahan $\sin\Theta$ bound (Davis & Kahan, 1970; Yu et al., 2014))**.** *Let $M, \widehat{M} \in \mathbb{R}^{d\times d}$ be symmetric matrices with eigenvalues arranged in nonincreasing order, $\lambda_1(M) \geq \cdots \geq \lambda_d(M)$ Let P and $\widehat{P}$ be the orthogonal spectral projectors onto the leading r-dimensional eigenspaces of M and $\widehat{M}$ respectively. Define the population eigengap and perturbation size by*

$$\gamma = \lambda_r(M) - \lambda_{r+1}(M) > 0, \qquad \varepsilon = \|\widehat{M} - M\|_2$$

If $\varepsilon < \frac{1}{2}\gamma < \gamma$,

$$\|\widehat{P} - P\|_2 = \|\sin\Theta(\widehat{P}, P)\|_2 \leq \frac{\varepsilon}{\gamma - \varepsilon} \leq \frac{2\varepsilon}{\gamma}$$

The proofs are follow the original papers. Here, we use if for proving Lemma B.10 stated as below.

Lemma B.10 (Latent-space tracking)**.** *If $\Delta_t^U > 0$ and $\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U) < \frac{1}{2}(\Delta_t^U + \bar{\kappa}_U)$ then*

$$e_t^U \leq \frac{2\big[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\big]}{\Delta_t^U + \bar{\kappa}_U}$$

Proof. The perturbation ε_t^U of M_t^U is

$$\begin{aligned} \varepsilon_t^U = \|\widehat{M}_t^U - M_t^U\|_2 &= \|\widehat{A}_t^U - A_t^U + \bar{\kappa}_U(\widehat{P}_{t-1}^U - P_t^U)\|_2 \\ &\leq \|\widehat{A}_t^U - A_t^U\|_2 + \bar{\kappa}_U \|\widehat{P}_{t-1}^U - P_t^U\|_2 \\ &\leq \xi_t^U + \bar{\kappa}_U \left(\|\widehat{P}_{t-1}^U - P_{t-1}^U\|_2 + \|P_{t-1}^U - P_t^U\|_2\right) \\ &= \xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U). \end{aligned}$$

The population eigen gap of M_t^U is $\gamma_t^U = \Delta_t^U + \bar{\kappa}_U$ according to Equation B.9. According to Lemma B.9, if $\varepsilon_t^U < \frac{1}{2}\gamma_t^U$, then

$$e_t^U = \|\widehat{P}_t^U - P_t^U\|_2 \leq \frac{2\varepsilon_t^U}{\gamma_t^U} = \frac{2\big[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\big]}{\Delta_t^U + \bar{\kappa}_U}$$

□

Proof. Because $\widehat{P}_t^U$ and $\widehat{P}_t^V$ are orthogonal projectors, the population gradient admits the four-block decomposition (mix the population level G_t with the sample level $\widehat{P}_t^U$ and $\widehat{P}_t^V$).

$$G_t = \widehat{P}_t^U G_t \widehat{P}_t^V + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V) + (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V) \tag{B.10}$$

and the actual estimator is

$$\widehat{G}_t = \widehat{P}_t^U G_t \widehat{P}_t^V + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V) \tag{B.11}$$

Equation B.13-Equation B.11 gives

$$\begin{aligned}
\|G_t - \widehat{G}_t\|_F^2 &= \| + (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V)\|_F^2 \\
&= \|(I - \widehat{P}_t^U) P_t^U G_t P_t^V (I - \widehat{P}_t^V)\|_F \\
&\leq \|(I - \widehat{P}_t^U) P_t^U\|_2 \|G_t\|_F \|P_t^V (I - \widehat{P}_t^V)\|_2 \\
&= \|\widehat{P}_t^U - P_t^U\|_2 \|G_t\|_F \|\widehat{P}_t^V - P_t^V\|_2 \\
&= e_t^U e_t^V \|G_t\|_F
\end{aligned}$$

and therefore

$$\frac{\|G_t - \widehat{G}_t\|_F}{\|G_t\|_F} \leq e_t^U e_t^V \leq \frac{2\big[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\big]}{\Delta_t^U + \bar{\kappa}_U} \cdot \frac{2\big[\xi_t^V + \bar{\kappa}_V(e_{t-1}^V + \delta_t^V)\big]}{\Delta_t^U + \bar{\kappa}_V} \tag{B.12}$$

and the last inequality is achieved from Lemma B.10. □

Theorem 5.2 (Coverage). *If $\Delta_t^U > 0$ and $\xi_t^U + \frac{\kappa_U}{n_t}(e_{t-1}^U + \delta_t^U) < \frac{1}{2}(\Delta_t^U + \frac{\kappa_U}{n_t})$ and likewise for V, then*

$$\frac{\|G_t - \widehat{G}_t\|_F}{\|G_t\|_F} \leq e_t^U e_t^V \leq \frac{4\big[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\big]\big[\xi_t^V + \bar{\kappa}_V(e_{t-1}^V + \delta_t^V)\big]}{(\Delta_t^U + \bar{\kappa}_U)(\Delta_t^V + \bar{\kappa}_V)}$$

Proof. Because $\widehat{P}_t^U$ and $\widehat{P}_t^V$ are orthogonal projectors, the population gradient admits the four-block decomposition (mix the population level G_t with the sample level $\widehat{P}_t^U$ and $\widehat{P}_t^V$).

$$G_t = \widehat{P}_t^U G_t \widehat{P}_t^V + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V) + (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V) \tag{B.13}$$

and the actual estimator is

$$\widehat{G}_t = \widehat{P}_t^U G_t \widehat{P}_t^V + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V) \tag{B.14}$$

Equation B.13-Equation B.11 gives

$$\begin{aligned}
\|G_t - \widehat{G}_t\|_F^2 &= \| + (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V)\|_F^2 \\
&= \|(I - \widehat{P}_t^U) P_t^U G_t P_t^V (I - \widehat{P}_t^V)\|_F \\
&\leq \|(I - \widehat{P}_t^U) P_t^U\|_2 \|G_t\|_F \|P_t^V (I - \widehat{P}_t^V)\|_2 \\
&= \|\widehat{P}_t^U - P_t^U\|_2 \|G_t\|_F \|\widehat{P}_t^V - P_t^V\|_2 \\
&= e_t^U e_t^V \|G_t\|_F
\end{aligned}$$

and therefore

$$\frac{\|G_t - \widehat{G}_t\|_F}{\|G_t\|_F} \leq e_t^U e_t^V \leq \frac{2\big[\xi_t^U + \bar{\kappa}_U(e_{t-1}^U + \delta_t^U)\big]}{\Delta_t^U + \bar{\kappa}_U} \cdot \frac{2\big[\xi_t^V + \bar{\kappa}_V(e_{t-1}^V + \delta_t^V)\big]}{\Delta_t^U + \bar{\kappa}_V} \tag{B.15}$$

and the last inequality is achieved from Lemma B.10. □

B.3 CONVERGENCE

Definition B.11 (Uncertainty-aware preconditioner)**.** If the preconditioned update is

$$\mathcal{D}_t\widehat{G}_t = \sum_{k=1}^{r} d_{t,k}\ \hat{\mu}_{t,k}\widehat{u}_{t,k}\widehat{v}_{t,k}^\top + (I-\widehat{P}_t^U)G_t\widehat{P}_t^V + \widehat{P}_t^U G_t(I-\widehat{P}_t^V)$$

where $d_{t,k}$ is the preconditioner for $\hat{\mu}_{t,k}$, with

$$d_{t,k} = \frac{1}{\sqrt{s_{t,k}}+\epsilon}$$

and the total update is

$$W_{t+1} = W_t + \eta\mathcal{D}_t\widehat{G}_t$$

Denote for $k \in \{1,\dots,r\}$,

$$0 < d_- \le \min\{1, d_{t,1},\dots,d_{t,r}\} \le \max\{1, d_{t,1},\dots,d_{t,r}\} \le d_+$$

Assumption B.12 (Smoothness)**.** The population loss function $\mathcal{L}$ is L-smooth:

$$\mathcal{L}(W+D) \le \mathcal{L}(W) + \langle\nabla\mathcal{L}(W), D\rangle + \frac{L}{2}\|D\|_F^2 \tag{B.16}$$

Theorem 5.3 (Convergence)**.** *Suppose the loss function $\mathcal{L}$ is L-smooth. If the learning rate η satisfies $0 < \eta \le \frac{d_-[1-(e_t^U e_t^V)^2]}{Ld_+^2}$, then*

$$L(W_{t+1}) \le \mathcal{L}(W_t) - \tfrac{\eta d_-}{2}\big[1-(e_t^U e_t^V)^2\big]\|\nabla\mathcal{L}(W_t)\|_F^2$$

Proof. We need the result of inner product and Frobenius norm:

$$\langle\widehat{G}_t, \mathcal{D}_t\widehat{G}_t\rangle_F \ge d_-\|\widehat{G}_t\|_F^2$$

$$\langle\widehat{A}_{t,k}, \widehat{A}_{t,l}\rangle_F = \mathrm{tr}(\hat{v}_{t,k}\hat{u}_{t,k}^\top\hat{u}_{t,l}\hat{v}_{t,l}^\top) = (\hat{u}_{t,k}^\top\hat{u}_{t,l})(\hat{v}_{t,k}^\top\hat{v}_{t,l}) = \delta_{k,l}$$

$$\hat{u}_{t,k}^\top(I-\widehat{P}_t^U) = 0, \quad \hat{v}_{t,k}^\top(I-\widehat{P}_t^V) = 0$$

Since $\widehat{P}_t^U$ is the projection space, so$\widehat{P}_t^U(I-\widehat{P}_t^U) = 0$. Therefore,

$$\langle\widehat{P}_t^U G_t\widehat{P}_t^V,\ (I-\widehat{P}_t^U)G_t\widehat{P}_t^V\rangle_F = \mathrm{tr}\left[\widehat{P}_t^V G_t^\top\widehat{P}_t^U(I-\widehat{P}_t^U)G_t\widehat{P}_t^V\right] = 0$$

Similarly, since $\widehat{P}_t^V(I-\widehat{P}_t^V) = 0$,

$$\langle\widehat{P}_t^U G_t\widehat{P}_t^V,\ \widehat{P}_t^U G_t(I-\widehat{P}_t^V)\rangle_F = \mathrm{tr}\left[\widehat{P}_t^U G_t\widehat{P}_t^V(I-\widehat{P}_t^V)G_t^\top\widehat{P}_t^U\right] = 0$$

and combining both,

$$\langle(I-\widehat{P}_t^U)G_t\widehat{P}_t^V,\ \widehat{P}_t^U G_t(I-\widehat{P}_t^V)\rangle_F = \mathrm{tr}\left[\widehat{P}_t^V G_t^\top(I-\widehat{P}_t^U)\widehat{P}_t^U G_t(I-\widehat{P}_t^V)\right] = 0$$

Therefore,

$$\|\widehat{G}_t\|_F^2 = \sum_{k=1}^{r}\hat{\mu}_{t,k}^2 + \|(I-\widehat{P}_t^U)G_t\widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t(I-\widehat{P}_t^V)\|_F^2$$

$$\langle\widehat{G}_t, \mathcal{D}_t\widehat{G}_t\rangle = \sum_{k=1}^{r} d_{t,k}\hat{\mu}_{t,k}^2 + \|(I-\widehat{P}_t^U)G_t\widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t(I-\widehat{P}_t^V)\|_F^2$$

$$\|\mathcal{D}_t\widehat{G}_t\|_F^2 = \sum_{k=1}^{r} d_{t,k}^2\hat{\mu}_{t,k}^2 + \|(I-\widehat{P}_t^U)G_t\widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t(I-\widehat{P}_t^V)\|_F^2$$

since $1 \geq d_-$, $d_{t,k} \geq d_-$ for $k \in \{1, \ldots, r\}$, therefore,

$$\begin{aligned}\langle \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle &\geq d_- \sum_{k=1}^{r} \hat{\mu}_{t,k}^2 + d_- \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + d_- \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \\ &= d_- \left[\sum_{k=1}^{r} \hat{\mu}_{t,k}^2 + \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \right] \\ &= d_- \|\widehat{G}_t\|_F^2\end{aligned}$$

Similarly, by using $d_+ \geq 1$, and $d_+ \geq d_{t,k}$ for $k \in \{1, \ldots, r\}$,

$$\begin{aligned}\langle \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle &\leq d_+ \sum_{k=1}^{r} \hat{\mu}_{t,k}^2 + d_+ \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + d_- \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \\ &= d_+ \left[\sum_{k=1}^{r} \hat{\mu}_{t,k}^2 + \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \right] \\ &= d_+ \|\widehat{G}_t\|_F^2\end{aligned}$$

To prove $\|\mathcal{D}_t \widehat{G}_t\|_F \leq d_+ \|\widehat{G}_t\|_F$,

$$\begin{aligned}\|\mathcal{D}_t \widehat{G}_t\|_F^2 &= \sum_{k=1}^{r} d_{t,k}^2 \hat{\mu}_{t,k}^2 + \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \\ &\leq d_+^2 \left[\sum_{k=1}^{r} \hat{\mu}_{t,k}^2 + \|(I - \widehat{P}_t^U) G_t \widehat{P}_t^V\|_F^2 + \|\widehat{P}_t^U G_t (I - \widehat{P}_t^V)\|_F^2 \right] \\ &= d_+^2 \|\widehat{G}_t\|_F^2\end{aligned} \tag{B.17}$$

to prove $\langle G_t - \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle = 0$,

$$\begin{aligned}\langle G_t - \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle_F &= \langle (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V),\ \sum_{k=1}^{r} d_{t,k} \hat{\mu}_{t,k} \widehat{u}_{t,k} \widehat{v}_{t,k}^\top + (I - \widehat{P}_t^U) G_t \widehat{P}_t^V + \widehat{P}_t^U G_t (I - \widehat{P}_t^V) \rangle_F \\ &= \langle (I - \widehat{P}_t^U) G_t (I - \widehat{P}_t^V),\ \sum_{k=1}^{r} d_{t,k} \hat{\mu}_{t,k} \widehat{u}_{t,k} \widehat{v}_{t,k}^\top \rangle_F \\ &= 0\end{aligned}$$

The last term is achieved since $\hat{u}_{t,k}^\top (I - \widehat{P}_t^U) = 0, \quad \hat{v}_{t,k}^\top (I - \widehat{P}_t^V) = 0.$

Finally, we could prove

$$d_- \|\widehat{G}_t\|_F^2 \leq \langle G_t, \mathcal{D}_t \widehat{G}_t \rangle_F = \langle \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle_F + \langle G_t - \widehat{G}_t, \mathcal{D}_t \widehat{G}_t \rangle_F \leq d_+ \|\widehat{G}_t\|_F^2 \tag{B.18}$$

Since Theorem 5.2 gives

$$\frac{\|G_t - \widehat{G}_t\|_F}{\|G_t\|_F} \leq e_t^U e_t^V \leq \frac{4\left[\xi_t^U + \bar{\kappa}_U (e_{t-1}^U + \delta_t^U)\right]\left[\xi_t^V + \bar{\kappa}_V (e_{t-1}^V + \delta_t^V)\right]}{\left(\Delta_t^U + \bar{\kappa}_U\right)\left(\Delta_t^U + \bar{\kappa}_V\right)} \tag{B.19}$$

$$\begin{gathered}\|\widehat{G}_t\|_F^2 - \|G_t\|_F^2 \leq \|\widehat{G}_t - G_t\|_F^2 \leq (e_t^U e_t^V)^2 \|G_t\|_F^2 \\ [1 - (e_t^U e_t^V)^2] \|G_t\|_F^2 \leq \|\widehat{G}_t\|_F^2\end{gathered}$$

Therefore, back to the Equation B.18,

$$\langle G_t, \mathcal{D}_t \widehat{G}_t \rangle_F \geq d_- \|\widehat{G}_t\|_F^2 \geq d_- [1 - (e_t^U e_t^V)^2] \|G_t\|_F^2 \tag{B.20}$$

From the Assumption B.12, suppose $W_{t+1} = W_t + \eta \mathcal{D}_t \widehat{G}_t$, by plugging in the Equation B.16,

$$\begin{aligned}
\mathcal{L}(W_{t+1}) &\leq \mathcal{L}(W_t) + \eta \langle \nabla \mathcal{L}(W_t), \mathcal{D}_t \widehat{G}_t \rangle + \frac{L\eta^2}{2} \|\mathcal{D}_t \widehat{G}_t\|_F^2 \\
&= \mathcal{L}(W_t) - \eta \langle G_t, \mathcal{D}_t \widehat{G}_t \rangle + \frac{L\eta^2}{2} \|\mathcal{D}_t \widehat{G}_t\|_F^2 \\
&= \mathcal{L}(W_t) - \eta \langle G_t, \mathcal{D}_t \widehat{G}_t \rangle + \frac{L\eta^2}{2} d_+^2 \|\widehat{G}_t\|_F^2 \\
&= \mathcal{L}(W_t) - \eta d_- [1 - (e_t^U e_t^V)^2 \|G_t\|_F^2] + \frac{L\eta^2}{2} d_+^2 \|G_t\|_F^2 \\
&\leq \mathcal{L}(W_t) - \left[\eta d_- [1 - (e_t^U e_t^V)^2] - \frac{L\eta^2}{2} d_+^2 \right] \|G_t\|_F^2
\end{aligned}$$

where the last inequality is from Equation B.20 and Equation B.17. By setting the $\frac{\eta d_- [1-(e_t^U e_t^V)^2]}{2} \geq \frac{L\eta^2}{2} d_+^2$, which is equivalent to condition of η in Theorem 5.3,

$$\eta d_- [1 - (e_t^U e_t^V)^2] - \frac{L\eta^2}{2} d_+^2 \geq \frac{\eta d_- [1 - (e_t^U e_t^V)^2]}{2}$$

Therefore,

$$\mathcal{L}(W_{t+1}) \leq \mathcal{L}(W_t) - \frac{\eta d_- [1 - (e_t^U e_t^V)^2]}{2} \|G_t\|_F^2 = \mathcal{L}(W_t) - \frac{\eta d_- [1 - (e_t^U e_t^V)^2]}{2} \|\nabla \mathcal{L}(W_t)\|_F^2$$

□

Corollary B.13 (Convergence to a stationary point)**.** *Suppose Theorem 5.3 holds, $\exists\, \bar{e} < 1$, such that for $\forall t \in \{0, \ldots, T-1\}$, $e_t^U e_t^V \leq \bar{e}$. Assume $\inf \mathcal{L}(W) = \mathcal{L}_{inf}$ exist, and*

$$0 < \eta \leq \frac{d_-(1 - \bar{e}^2)}{L d_+^2}$$

then $\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}(W_t)\|_F^2 \leq \frac{2[\mathcal{L}(W_0) - \mathcal{L}_{inf}]}{\eta d_-(1-\bar{e}^2)T}$ and consequently,

$$\min_{0 \leq t \leq T} \|\nabla \mathcal{L}(W_t)\|_F^2 = O(T^{-1})$$

Proof. Since $e_t^U e_t^V \leq \bar{e}$, Theorem 5.3 gives

$$\mathcal{L}(W_{t+1}) \leq \mathcal{L}(W_t) - \frac{\eta d_-}{2}(1 - \bar{e}^2) \|\nabla \mathcal{L}(W_t)\|_F^2$$

And summing over $t = 0, \ldots, T-1$, we have

$$\frac{\eta d_-}{2}(1 - \bar{e}^2) \sum_{t=1}^{T-1} \|\nabla \mathcal{L}(W_t)\|_F^2 \leq \mathcal{L}(W_0) - \mathcal{L}(W_T) \leq \mathcal{L}(W_0) - \mathcal{L}_{\text{inf}}$$

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}(W_t)\|_F^2 \leq \frac{2[\mathcal{L}(W_0) - \mathcal{L}_{\text{inf}}]}{\eta d_-(1 - \bar{e}^2)T}$$

Therefore,

$$\min_{0 \leq t \leq T} \|\nabla \mathcal{L}(W_t)\|_F^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}(W_t)\|_F^2 = O(T^{-1})$$

□

	Hyperparameter	Value
Adam optimizer	β_1	0.9
	β_2	0.999
	ϵ	1×10^{-8}
	Weight decay λ	0.0
	Learning rate Schedule	Cosine
	Warmup steps	20

Table 6: Optimizer settings.

Task	# Train	# Val	# Test	Steps	Model selection	Batch Size	Max Length
CoLA	8,051	500	1,000	3,000	max validation set accuracy, no patience	8	512
SST-2	66,349	1,000	872	4,200		32	128
QNLI	103,743	1,000	5,463	6,500		32	128
MRPC	3,168	500	408	3,000		8	512
RTE	1,990	500	277	3,000		8	512
QQP	20,000	500	1,000	3,000		8	512

Table 7: Training details of GLUE subset tasks on the T5-base model.

C FIGURE DETAILS

Implementation of the right figure of Figure 1. The synthetic task of teacher–student classification. The student is pretrained under teacher T and fine-tuned under a rotated teacher $T' = (1-\gamma)T + \gamma TR$ where R is a rotation matrix inducing task shift. # of classification $K = 8$, with 32 MLP hidden width. $n = 1200$ samples, drawn separtedly for pretrain and fine-tune. Shift =0.06 of the rotation strength. Seed=0. The MLP architecture: $d_{\text{in}} = 24$, one linear and then tanh, then another linear layer, and prediction on $K = 8$. Pretraining: full batch adam, with learning rate $\eta = 0.01$, pretraining steps=250. Fine tuning: Lora variants use $r = 4$. All methods presented use learning rate $\eta = 0.03$.

Implementation of Figure 2. On Llama-3.1-8B, the full fine-tuning method, we compare the gradient space of query block at layer 0 by measuring principal angle at $t = 0$ and at each training steps t for Figure 2 (a). More replication of subfigure (a) with different layers on Llama-3.1-8B is available in Appendix E.1.

Figure 3 presents the coverage of different methods on the task MMLU (5 shots). Other results are available in Appendix E.2.

Figure 4 presents the preconditioner denominator part the $s_{t,k} = \rho_{t,k}^2 P_{t-1,k}^+ + q_0 + q_1(1 - \rho_{t,k}^2) + \frac{1}{n_t \beta_{t,k}}$ on task GSM8K using Llama-3.1-8B model, with configurations $k = 0.1$, $q \in \{10^{-3}, 10^{-4}, 10^{-6}\}$, seed= 6.

D EXPERIMENTS DETAILS

D.1 TRAINING DETAILS

Table 6 presents the optimizer settings for both the Adam optimizer used by the baselines and our adaptation of Adam. All methods share the first-moment decay β_1, numerical stability constant ϵ, weight decay, learning-rate schedule and warmup duration. β_2 applies only to methods retaining Adam's second-moment estimate.

Table 7 presents the training details for the selected GLUE tasks on the T5-base model. For every task, the checkpoint used for final evaluation is selected from checkpoints saved every 50 steps based on the highest validation accuracy, without early stopping.

Task	Train set	# Train	Test set	# Test	Steps	Model selection	Batch Size
Math	MetaMathQA		GSM8K	1,000			4^3
Code	CodeFeedback	20,000	HumanEval	164	3,000	last step configuration	8
General	Alpaca		MMLU	1,000			8

Table 8: Training details for natural language generation tasks in math, code, and general understanding domains on Llama-3.1-8B model.

	Parameter	CoLA	SST-2	MRPC	QNLI	RTE	QQP	GSM8K	HumanEval	MMLU
Method	κ	1	1	5	1	0.1	0.1	0.5	0.1	1
	q	1	10	0.01	10	0.01	0.1	10^{-4}	1	1
Training	η	0.05	0.5	0.1	0.1	0.05	0.01	2×10^{-3}	5×10^{-5}	5×10^{-5}

Table 9: Hyperparameter configuration of GRABLORA for GLUE tasks on T5-base model and for natural language generation tasks on Llama-3.1-8B model.

Table 8 presents the training details for natural language generation tasks in math, code, and general understanding domains on the Llama-3.1-8B model. Each domain is trained on $20{,}000$ examples from a dedicated instruction dataset, MetaMathQA (math), CodeFeedback (code), and Alpaca (general understanding), and evaluated on GSM8K, HumanEval, and MMLU, respectively. All models are fine-tuned for $3{,}000$ steps with a batch size of 4 or 8, and the final checkpoint is taken from the last training step.

D.2 HYPERPARAMETER CONFIGURATION

Table 9 presents the hyperparameter configuration of GRABLORA for GLUE tasks on T5-base model and math, code, general understanding tasks on Llama-3.1-8B model. The hyperparameters $\kappa_U = \kappa_V = \kappa$, $q_0 = q_1 = q$ and η are tuned per task. Because of GRABLORA introduces a specific preconditioner designed for the evolving basis, we tune its learning rate over a different range from that used for initialization-based LoRA variants.

Our implementation uses a unified framework with the same training setting for initialization-only methods, ROSA, SubTrack, vanilla LoRA, and Full FT, all of which update only the query and value projection matrices. Therefore, we use the settings in the original papers as references and conduct a learning-rate grid search over ranges informed by LoRA-GA and LoRA-One, as well as additional ranges suited to our setup. Table 10 presents the hyperparameter used for GLUE tasks. Table 11 presents the hyperparameter used for natural language generation tasks.

D.3 COMPUTATION RESOURCES

We conduct our experiments on 8 NVIDIA A100-40GB gpus.

D.4 PROMPT DESIGN

For each task domain we report the prompt used at training and at evaluation. Bracketed slots (e.g. `[QUESTION]`) denote data-specific fields substituted at run time.

GLUE

```
# SST-2 (sentiment)
Review: [SENTENCE]
Is the sentiment positive? Answer yes or no:

# MRPC (paraphrase)
Sentence 1: [SENTENCE 1]
```

[3]On the math task, batch size = 8 causes CG-LoRA and ROSA to exceed the memory capacity of an NVIDIA A100-40GB GPU. To make a fair comparison across all methods, we use a batch size of 4 for the math task.

Method	CoLA	SST-2	MRPC	QNLI	RTE	QQP
LoRA-GA	2e-4	5e-4	2e-4	5e-4	5e-5	3e-5
LoRA-One	2e-4	5e-4	8e-5	5e-4	8e-5	5e-5
CG-LoRA	2e-4	5e-4	8e-5	1e-4	5e-5	3e-5
ROSA	5e-5	5e-4	5e-5	5e-4	5e-5	4e-5
SubTrack	5e-5	5e-5	5e-5	5e-4	5e-5	5e-5
Vanilla LoRA	2e-4	5e-4	2e-4	5e-4	6e-5	2e-5
Full	1e-4	5e-4	1e-4	5e-4	1e-4	1e-4

Table 10: Selected learning rate for each baseline on the GLUE subset tasks (T5-base model), chosen from the learning rate sweep. We roughly sweep over $5e-6, 1e-5, 5e-5, 1e-4, 2e-4, 5e-4$, with a local search for possible higher value of the baselines.

Method	GSM8K	HumanEval	MMLU
LoRA-One	2e-4, **5e-5**	**2e-4**, 1e-4, 5e-5	**2e-4**, 1e-4, 5e-5
CG-LoRA	2e-4, **1e-5**	**2e-4**, 5e-5, 1e-4	2e-4, **5e-5**, 1e-4
LoRA-GA	2e-4, **1e-5**	2e-4, **5e-4**	**1e-4**, 2e-4, 5e-4
ROSA	**2e-5**, 1e-5, 5e-6	2e-4, **1e-4**, 5e-4	2e-4, 5e-5, **1e-4**
SubTrack	2e-4, 5e-5, **1e-5**	1e-5, **5e-5**	**1e-5**, 5e-5
Full	**1e-5**, 5e-6	**1e-5**	**1e-5**
Vanilla LoRA	2e-4, **2e-5**	**2e-4**, 5e-5, 1e-4	**2e-4**, 5e-4, 1e-4
Ours	2e-4, **2e-3**, 5e-5	1e-5, **5e-5**	1e-5, **5e-5**

Table 11: Learning rate grid search for each baseline on the GSM8K, HumanEval, and MMLU tasks on Llama-3.1-8B model. Each entry lists the set of learning rates searched and the bold value denotes the configuration selected for the final results by using seed = 45 as a pivot run.

```
Sentence 2: [SENTENCE 2]
Do they mean the same thing? Answer yes or no:

# CoLA (linguistic acceptability)
Sentence: [SENTENCE]
Is this sentence grammatical? Answer yes or no:

# QNLI (question--sentence entailment)
Question: [QUESTION]
Sentence: [SENTENCE]
Does the sentence answer the question? Answer yes or no:

# RTE (textual entailment)
Premise: [SENTENCE 1]
Hypothesis: [SENTENCE 2]
Does the premise entail the hypothesis? Answer yes or no:

# QQP (question equivalence)
Question 1: [QUESTION 1]
Question 2: [QUESTION 2]
Are these two questions duplicates? Answer yes or no:
```

Math (GSM8K)

```
Below is a grade-school math problem. Solve it step by step
and give the final answer after '####'.

Question: [QUESTION]
Answer:
```

Code (HumanEval)

```
Below is a programming problem. Write a correct solution.

### Problem:
[PROBLEM]

### Solution:
```

General Understanding (MMLU)

```
Below is an instruction that describes a task. Write a response
that appropriately completes the request.

### Instruction:
[INSTRUCTION]

### Response:
```

Evaluation uses a 5-shot multiple-choice format.

```
The following are multiple choice questions (with answers)
about [SUBJECT].

[QUESTION]
A. [CHOICE A]
B. [CHOICE B]
C. [CHOICE C]
D. [CHOICE D]
Answer:
```

E ADDITIONAL RESULTS

E.1 MISALIGNMENT

E.2 COVERAGE

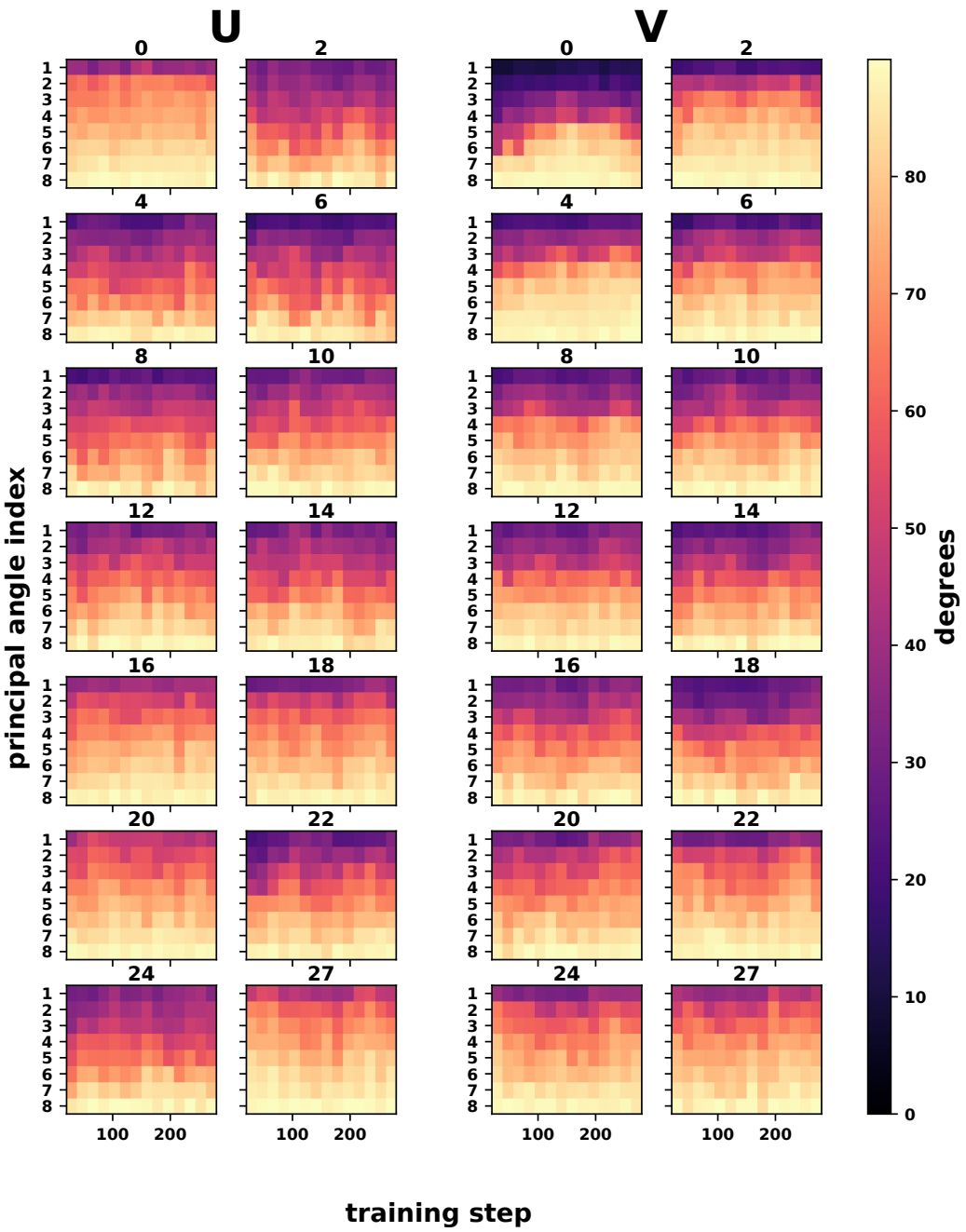


Figure 5: Llama-3.1-8B 0-27 14 layers of query block rotation. Left: U space incremental rotation comparing current state with last $\tau = 20$ steps. Right: V space incremental rotation comparing current state with last $\tau = 20$ steps.

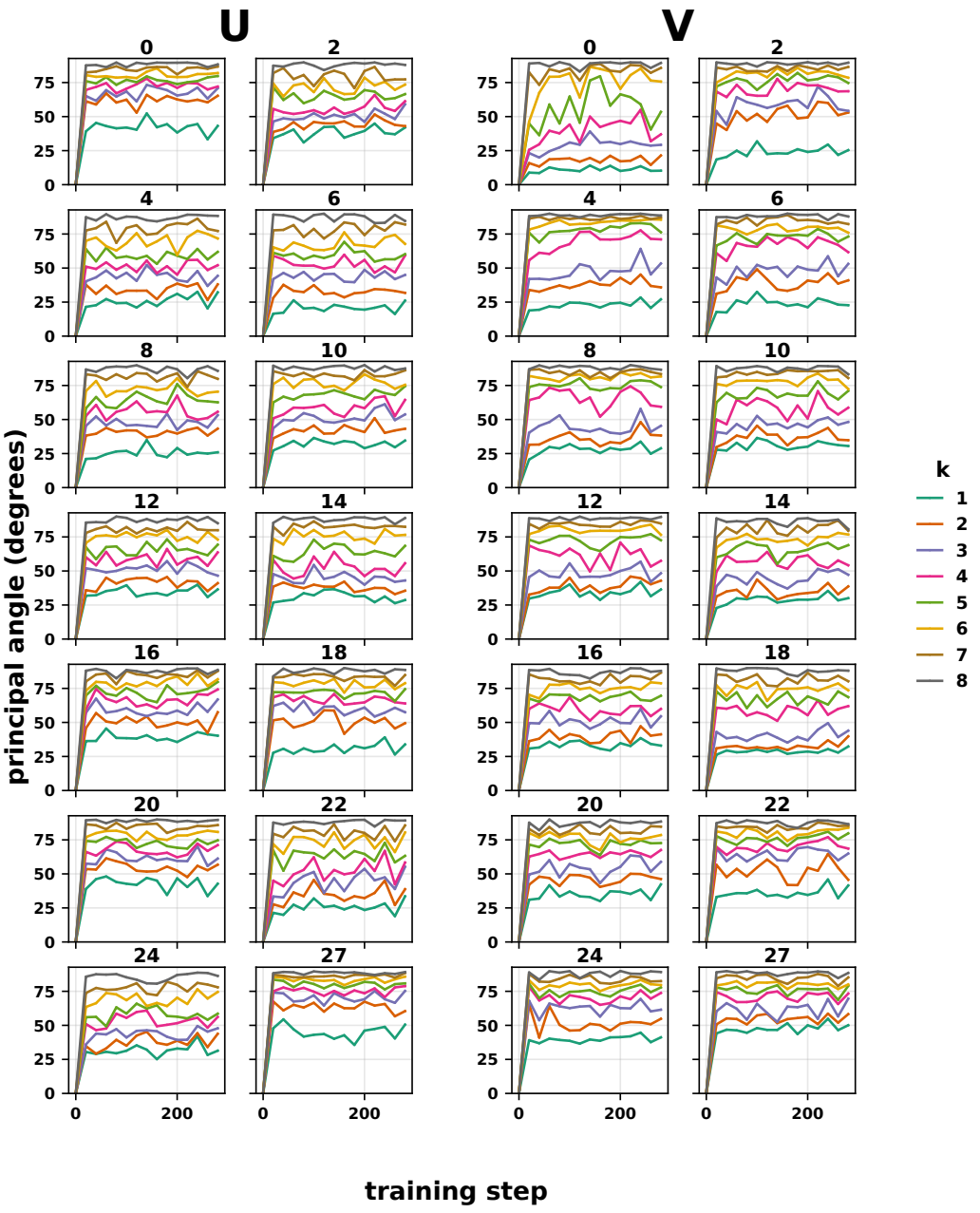


Figure 6: Llama-3.1-8B 0-27 14 layers of query block rotation. Left: U space cumulative rotation comparing current state with last $\tau = 20$ steps. Right: V space cumulative rotation comparing current state with last $\tau = 20$ steps.

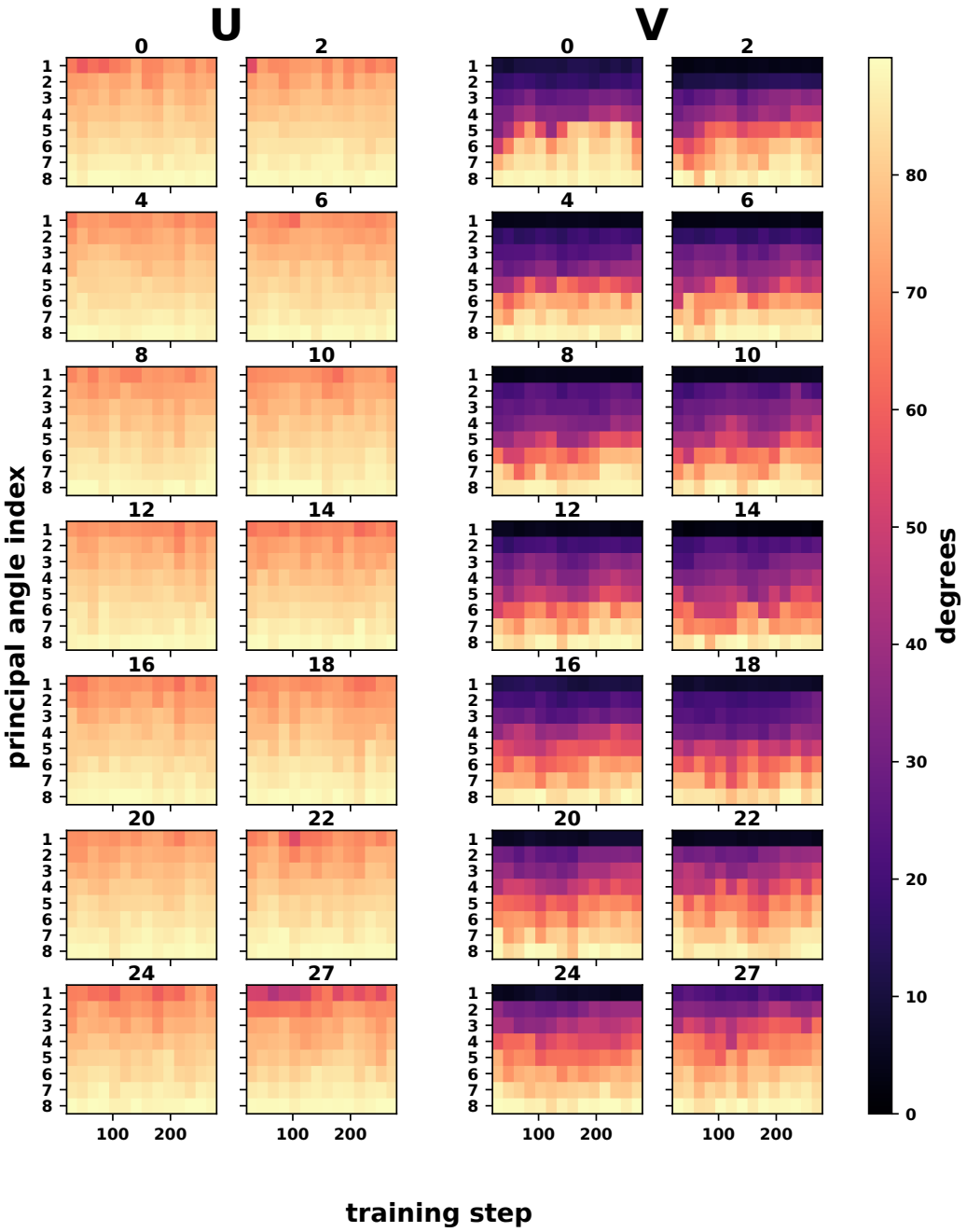


Figure 7: Llama-3.1-8B 0-27 14 layers of value block rotation. Left: U space incremental rotation comparing current state with last $\tau = 20$ steps. Right: V space incremental rotation comparing current state with last $\tau = 20$ steps.

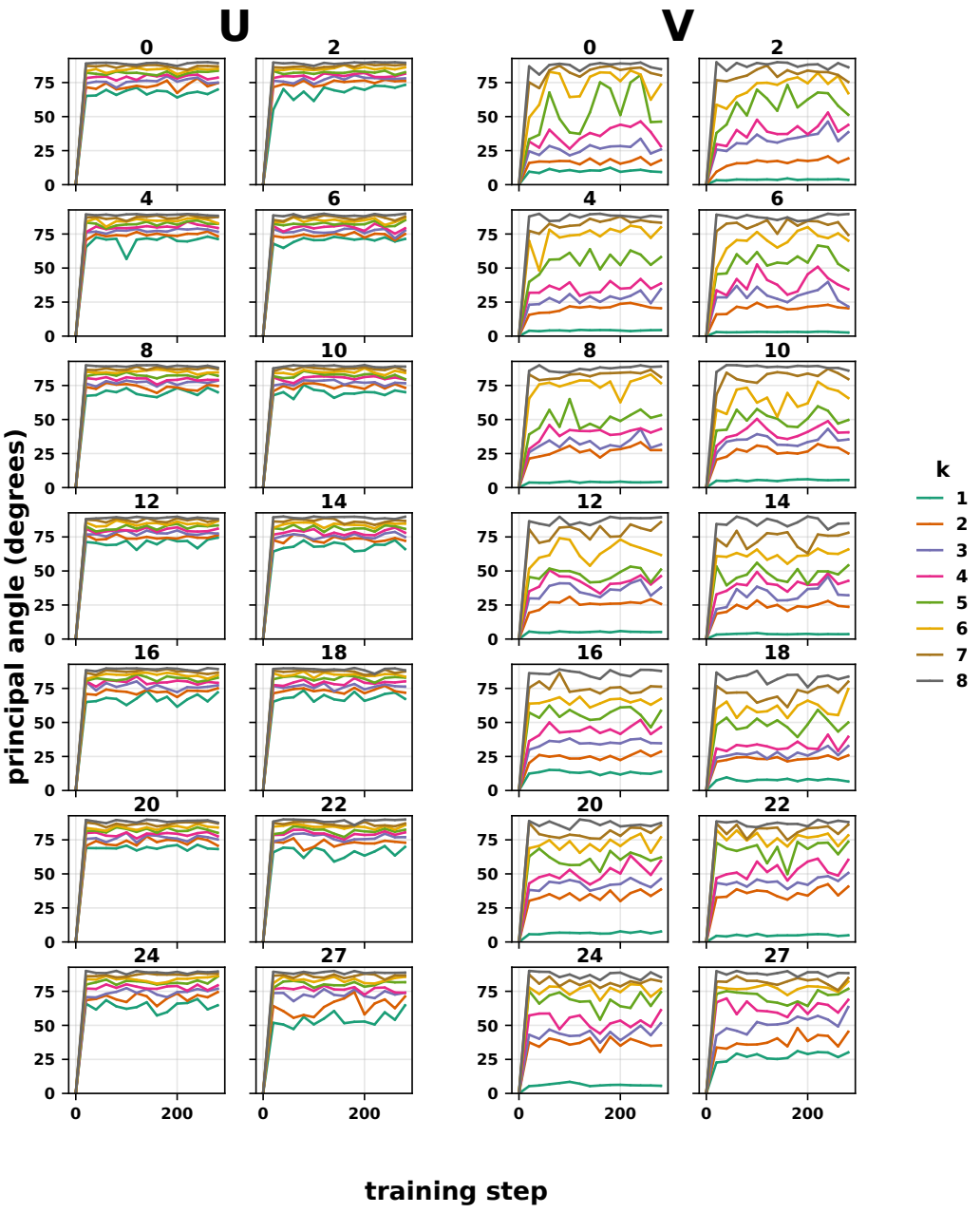


Figure 8: Llama-3.1-8B 0-27 14 layers of value block rotation. Left: U space cumulative rotation comparing current state with last $\tau = 20$ steps. Right: V space cumulative rotation comparing current state with last $\tau = 20$ steps.

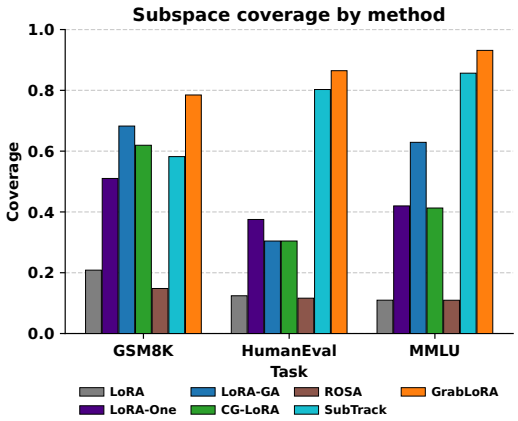


Figure 9: Natural language generation tasks final coverage of all methods during 3000 steps training average over seed=1,2,3 on Llama-3.1-8B model.

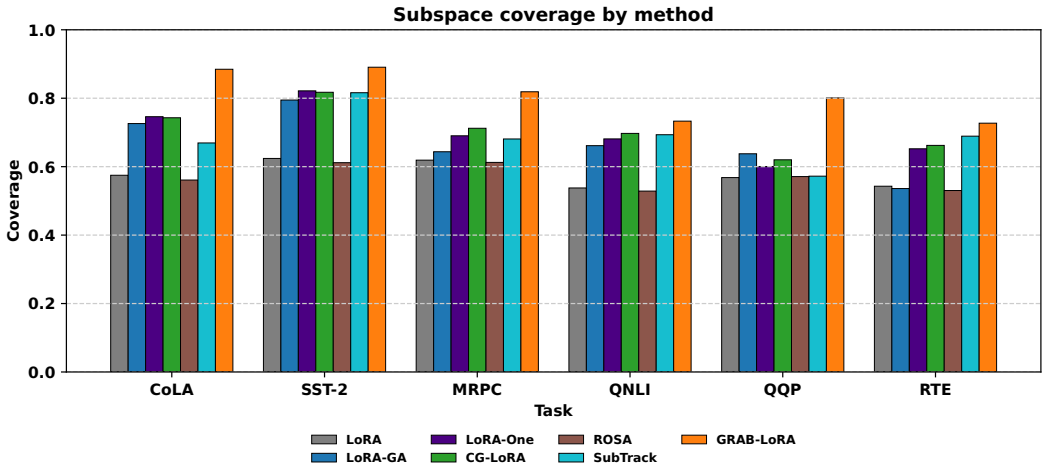


Figure 10: GLUE final coverage of all methods during 3000 steps training average over seed=1,2,3 on T5-base model.

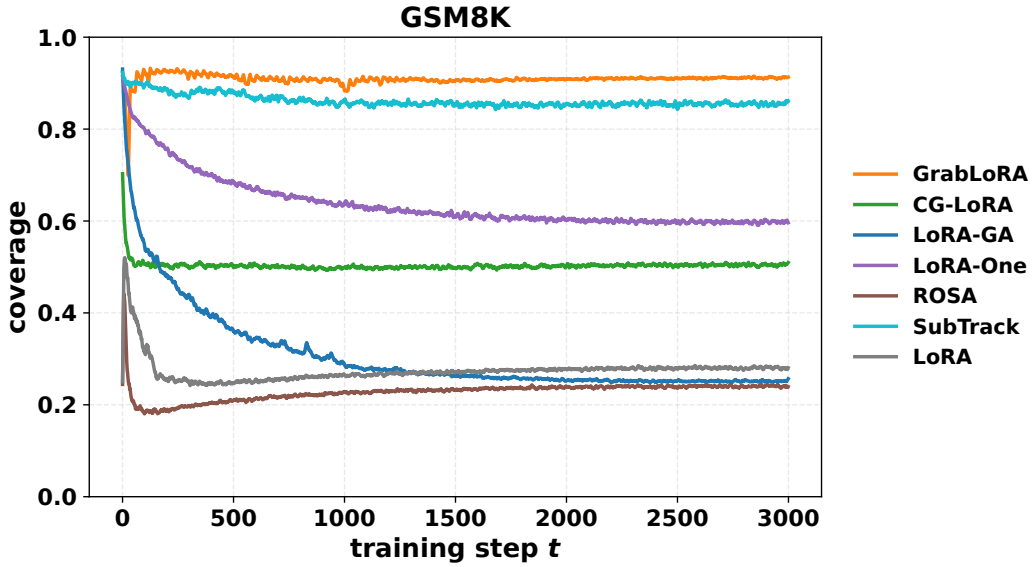


Figure 11: GSM8K coverage of all methods during 3000 steps training average over seed=1,2,3 3 seeds on Llama-3.1-8B.

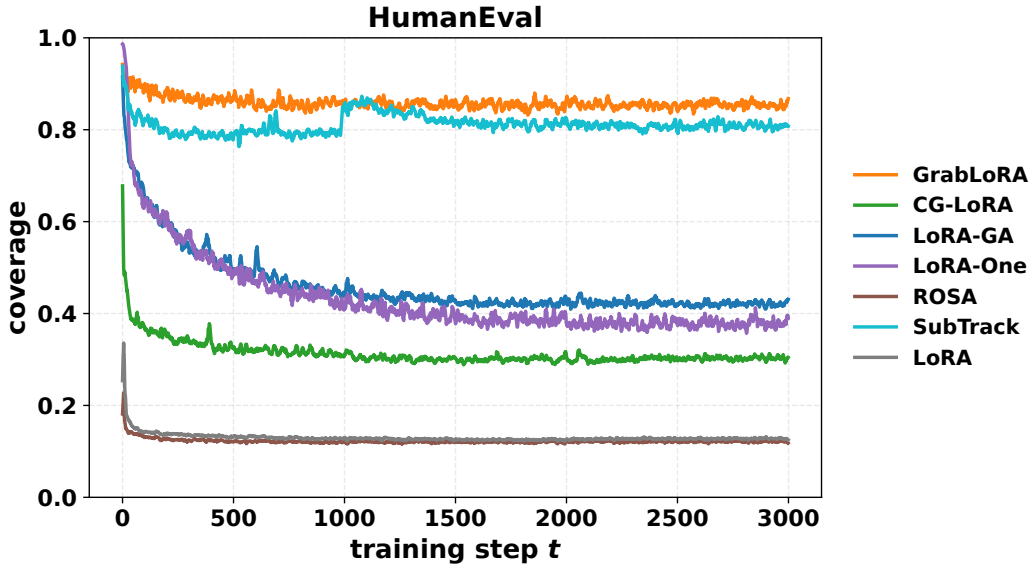


Figure 12: GSM8K coverage of all methods during 3000 steps training average over seed=1,2,3 3 seeds on Llama-3.1-8B.