Seq-LoRA: Sequential Bayesian Low-Rank Adaptation for Large Language Models

Boming Chen
Department of Statistics
University of Connecticut
boming.chen@uconn.edu

Xiaojing Wang
Department of Statistics
University of Connecticut
xiaojing.wang@uconn.edu

Abstract

Low-Rank Adaptation (LoRA) enables efficient fine-tuning of large language models, but adapted models often become overconfident, especially in low-data settings and under distribution shift. Existing uncertainty-aware LoRA methods typically construct a static posterior over a fixed adaptation set and do not model how local posterior evidence varies across structured subsets of the training data. We propose **Seq-LoRA**, a post-hoc Bayesian framework for sequentially aggregating slice-wise posterior evidence in LoRA space. Starting from a shared maximum a posteriori LoRA anchor, Seq-LoRA builds slice-wise Kronecker-factored quadratic surrogates, projects them into a shared curvature-informed low-dimensional subspace, rewrites the resulting latent quadratic forms as Gaussian pseudo-observations, and performs exact inference in an induced linear–Gaussian state-space model via Kalman filtering. Rather than relying on a particular curriculum order, Seq-LoRA couples heterogeneous slice-wise local posterior evidence through a random-walk prior, forming a terminal Bayesian posterior for prediction. Across the in-distribution ScienceQA test split and six out-of-distribution (OOD) reasoning targets, Seq-LoRA substantially reduces deterministic LoRA overconfidence under shift. Among fine-tuned methods, it achieves the best negative log-likelihood on five OOD targets and the best expected calibration error on four, while preserving most task accuracy. Code is available at https://github.com/toori999/seq-lora.

1 Introduction

Low-Rank Adaptation (LoRA) has become a standard approach to parameter-efficient fine-tuning (PEFT) of large language models (LLMs), enabling downstream adaptation with substantially lower memory and optimization costs than full fine-tuning [1–8]. However, strong task accuracy alone does not guarantee reliable predictive uncertainty. In low-data and distribution-shifted settings, LoRA-adapted LLMs can become overconfident, producing predictive probabilities that are poorly aligned with empirical correctness [9–15]. Trustworthy deployment therefore requires calibrated probabilities—commonly evaluated using negative log-likelihood (NLL) and expected calibration error (ECE)—in addition to high accuracy [9, 16–18].

Recent uncertainty-aware PEFT methods address this issue by placing approximate posteriors over the low-rank adaptation space rather than the full LLM parameter space [13–15, 19–21]. This restriction makes posterior approximation tractable, but most existing Bayesian LoRA methods remain *static*: they construct a single posterior approximation over a fixed adaptation set and treat the training data as a homogeneous, unordered collection. This static view overlooks a common form of structure in downstream data: examples are often naturally partitioned into ordered slices.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In many practical settings, adaptation data are not perfectly homogeneous. Even before encountering external out-of-distribution (OOD) benchmarks, the training distribution may contain weak structured variation across slices, arising from temporal acquisition, streaming batches, curriculum stages, or interpretable metadata such as grade levels in educational data. Curriculum learning typically uses such structured variation to shape deterministic optimization and improve the final maximum a posteriori (MAP) model [22–26]. Seq-LoRA asks a complementary question: after a MAP adapter has already been trained, can the same slice-level heterogeneity be converted into posterior evidence for uncertainty estimation? We view each slice as inducing a local pull around the shared MAP anchor, and use these local pulls as weak signals of how the adapted model may drift under distributional change. This raises our central question: **can training-set slice variation yield less overconfident posterior predictions under external shift?**

We propose **Seq-LoRA**, a post-hoc Bayesian framework that models sliced LoRA adaptation as sequential inference over slice-dependent deviations from a shared MAP anchor. Given a partition $\{\mathcal{D}_t\}_{t=1}^{T}$, Seq-LoRA builds slice-wise quadratic surrogates of the adapter loss around the MAP solution using Kronecker-factored curvature [27–29]. These local surrogates are then projected into a shared curvature-informed latent subspace, preserving PEFT efficiency while placing all slices in a common coordinate system. In this latent space, the slice-wise quadratic terms admit an equivalent Gaussian pseudo-observation interpretation, yielding a linear–Gaussian state-space model (LGSSM). This enables closed-form sequential posterior inference via Kalman filtering [30, 31]. A random-walk transition prior provides lightweight coupling across adjacent slices without imposing a fixed monotonic trend.

Empirically, Seq-LoRA improves predictive calibration and likelihood while keeping accuracy close to deterministic LoRA, with particularly clear gains under distribution shift. Across in-distribution (ID) and OOD science question-answering benchmarks, Seq-LoRA reduces the overconfidence induced by deterministic LoRA adaptation and achieves a strong likelihood–calibration trade-off among deterministic, ensemble-based, and Bayesian LoRA baselines. Semantic order is a useful, interpretable inductive bias, not a requirement: in our ablations, multiple tested slice constructions, including random partitions, yield useful slice-wise posterior evidence, while LGSSM coupling provides a structured way to aggregate this evidence and stabilize predictive uncertainty.

Contributions. Our contributions are threefold:

- We introduce *sequential aggregation of slice-structured local evidence* for uncertainty-aware PEFT, framing posterior construction over training slices as state-space inference rather than a static posterior approximation.
- We propose **Seq-LoRA**, a tractable post-hoc Bayesian LoRA framework that combines slice-wise Kronecker-factored surrogates, a shared curvature-aligned latent subspace, Gaussian pseudo-observations, and an LGSSM posterior solved by Kalman filtering.
- We demonstrate that the LGSSM formulation offers a structured posterior construction principle for PEFT: slice-wise local evidence supplies posterior diversity, while sequential LGSSM coupling converts this evidence into a stable terminal posterior that improves calibration and NLL under distribution shift.

2 Related Work

Uncertainty-aware LoRA and Bayesian PEFT. LoRA reduces the cost of adapting large language models by freezing the pretrained weights and learning low-rank update matrices [4]. This efficiency makes the adapter space an attractive target for uncertainty estimation. General-purpose uncertainty baselines such as Monte Carlo Dropout (MC-Dropout) and deep ensembles can be applied to adapted models [32, 33], but they either rely on stochastic regularization or require multiple independently trained adapters. Bayesian LoRA methods instead introduce posterior uncertainty directly in the adapter parameters. Laplace-LoRA applies a post-hoc Laplace approximation around a trained LoRA adapter using curvature information [13, 27, 34, 35]. BLoB learns variational posterior moments during fine-tuning [14]. TFB transforms trained adapters into Bayesian ones without additional training by searching for an acceptable posterior variance [21]. C-LoRA introduces contextual, input-dependent posterior parameterization for uncertainty-aware adaptation [15]. These methods improve calibration, but they typically construct one global posterior over the adaptation data. In

contrast, Seq-LoRA constructs slice-wise local evidence and couples it through a sequential state-space posterior. LoRA-MCL [36] also exploits LoRA-space diversity, but jointly trains multiple LoRA hypotheses with a (relaxed) Winner-Takes-All objective for diverse generation, rather than building a post-hoc posterior around one MAP adapter for calibration under shift.

Curriculum learning and data ordering. Curriculum learning studies how the order or weighting of examples affects optimization and generalization. Early work frames curriculum learning as a "starting small" strategy and relates it to continuation methods for non-convex optimization [22]. Self-paced learning selects easier examples according to the current model state [23], while competence-based and cross-review methods develop task-adaptive curricula for neural machine translation and natural language understanding [24, 37]. Recent LLM studies also show that curriculum or strategic data ordering can affect instruction tuning and post-training [25, 26]. At the same time, effective curricula need not be monotonic: discovered curricula can be non-monotonic, and simple easy-to-hard or hard-to-easy schedules may underperform [38, 39]. Seq-LoRA is related to this literature in its use of ordered partitions, but differs in objective. We do not rely on order primarily as a training schedule for improving the MAP adapter. Instead, we use ordered slices after MAP training as structured posterior evidence for uncertainty estimation.

3 Preliminaries

We briefly introduce the notation and background on Low-Rank Adaptation and Bayesian posterior approximation used throughout the paper. We use lowercase letters for scalars and vectors and uppercase letters for matrices, except for counts such as M, T, and L.

3.1 Low-Rank Adaptation

Consider M LoRA-instrumented modules. For module m, the adapted weight is

$$W^{(m)} = W_0^{(m)} + B^{(m)} A^{(m)}, \tag{1}$$

where $W_0^{(m)}$ is the frozen pre-trained weight, and $B^{(m)} \in \mathbb{R}^{d_{\text{out}}^{(m)} \times r^{(m)}}$ and $A^{(m)} \in \mathbb{R}^{r^{(m)} \times d_{\text{in}}^{(m)}}$ are low-rank adaptation factors with $r^{(m)} \ll \min\big(d_{\text{out}}^{(m)}, d_{\text{in}}^{(m)}\big)$. For notational simplicity, we absorb the LoRA scaling factor into $B^{(m)}$. This parameterization substantially reduces the number of trainable parameters relative to full fine-tuning while retaining the efficiency advantages of parameter-efficient adaptation [4].

As in prior Bayesian LoRA methods, we place uncertainty only over the trainable $A^{(m)}$ factors while keeping $W_0^{(m)}$ and $B^{(m)}$ fixed [13, 14]. Intuitively, fixing $B^{(m)}$ preserves the learned low-rank basis, while the posterior quantifies uncertainty over the corresponding low-dimensional adaptation coefficients. Let $a \in \mathbb{R}^d$ denote the concatenated vectorization of all trainable LoRA-A parameters across the M modules. Throughout, $\text{vec}_{\text{row}}(A)$ denotes row-major vectorization of A, matching the implementation's flatten/view convention, and $\text{mat}^{(m)}(\cdot)$ denotes the inverse reshape for module m.

3.2 Bayesian Prediction and Local Curvature Approximation

Given a test input ξ, Bayesian prediction marginalizes over the posterior distribution of the LoRA parameters [19]:

$$p(y \mid \xi, \mathcal{D}) = \int p(y \mid \xi, a)\, p(a \mid \mathcal{D})\, da. \tag{2}$$

For modern neural networks, this posterior is generally intractable, so practical Bayesian methods rely on approximate posterior representations.

A standard route is to approximate a local objective, such as the empirical risk or the negative log posterior, by a second-order expansion around a reference point a_{ref}:

$$\mathcal{L}(a_{\text{ref}} + \Delta a) \approx \mathcal{L}(a_{\text{ref}}) + g^\top \Delta a + \frac{1}{2} \Delta a^\top \Lambda \Delta a, \tag{3}$$

where $g := \nabla_a \mathcal{L}(a_{\text{ref}})$ is the local gradient and $\Lambda \succeq 0$ is a positive semi-definite curvature approximation, typically based on the Fisher information or generalized Gauss–Newton matrix [27, 29].

This local quadratic form underlies scalable Laplace-style Bayesian inference [34] and motivates the curvature-based posterior approximation used later in our method.

Seq-LoRA specializes this Bayesian setup to ordered data slices and a sequential latent-state model, as described in Section 4.

4 Seq-LoRA: Sequential Bayesian Adaptation

Seq-LoRA performs post-hoc Bayesian inference over slice-dependent LoRA-A deviations around a shared MAP adapter. The key idea is to treat each ordered data slice as providing local quadratic evidence about an unobserved adapter displacement, then aggregate this evidence through a low-dimensional linear–Gaussian state-space model. Figure 1 summarizes the four-stage construction, and the following subsections give the formal details. Additional derivations and implementation details are deferred to Appendix A.

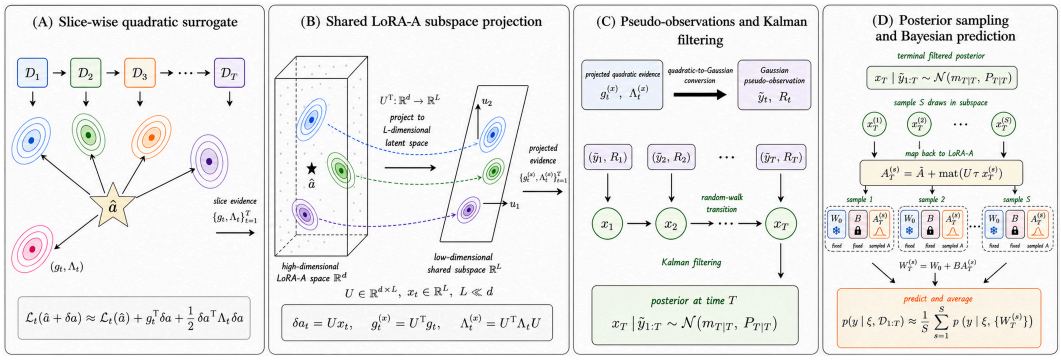

Figure 1: Overview of Seq-LoRA. (A) Slice-wise gradients and curvatures define local evidence about the LoRA-A deviation from the MAP anchor. (B) This evidence is projected into a shared curvature-aligned subspace. (C) The projected surrogates become Gaussian pseudo-observations in a random-walk LGSSM, yielding $p(x_T \mid \widetilde{y}_{1:T})$ by Kalman filtering. (D) Samples are mapped back to LoRA weights for Bayesian model averaging.

4.1 Ordered Slice-wise Surrogates

Following Section 3, let $a \in \mathbb{R}^d$ denote the concatenated vectorization of all trainable LoRA-A parameters across the LoRA-instrumented modules, and let $\hat{a}$ denote a shared MAP LoRA anchor obtained by standard fine-tuning. For ordered slices $\{\mathcal{D}_t\}_{t=1}^T$, Seq-LoRA models slice-dependent deviations $\delta a_t := a_t - \hat{a}$. For each slice t, we approximate the slice-wise empirical risk $\mathcal{L}_t(a)$ around the common anchor by

$$\mathcal{L}_t(\hat{a} + \delta a) \approx \mathcal{L}_t(\hat{a}) + g_t^\top \delta a + \frac{1}{2} \delta a^\top \Lambda_t \delta a, \tag{4}$$

where $g_t := \nabla_a \mathcal{L}_t(\hat{a})$ is the slice-wise gradient and $\Lambda_t \succeq 0$ is a positive semi-definite curvature approximation. Using a shared anchor keeps all slice-wise surrogates in a common coordinate system. Thus, a slice surrogate is local Newton-style evidence at the shared MAP anchor, not a Laplace approximation around a slice-specific optimum. Slices enter only post hoc: $\hat{a}$ is fine-tuned once on the full training set without per-slice adapters, and the partition determines only the local evidence constructed around this shared anchor. Natural slice axes include grade or difficulty levels, data-acquisition batches, and curriculum stages; without such metadata, slices can be formed from training-only base-model NLL or as a size-balanced random partition in a fixed, semantics-free order (Appendix A.7).

For module m, recall the LoRA parameterization $W^{(m)} = W_0^{(m)} + B^{(m)} A^{(m)}$. Let $h^{(m)}$ denote the module input and $g_{\text{out}}^{(m)}$ the gradient with respect to the module output under the slice loss. Since $\Delta W^{(m)} = B^{(m)} A^{(m)}$, the gradient with respect to $A^{(m)}$ is

$$\nabla_{A^{(m)}} \mathcal{L}_t = \left(B^{(m)}\right)^\top g_{\text{out}}^{(m)} \left(h^{(m)}\right)^\top. \tag{5}$$

To make the slice-wise surrogate tractable in the LoRA-A space, we use a module-wise Kronecker-factored approximate curvature (K-FAC) approximation of the classification generalized Gauss–Newton (GGN), restricted to the LoRA-A parameters [27, 29]. With $q^{(m)} := (B^{(m)})^\top g_{\text{out}}^{(m)}$, Equation (5) is the separable outer product $q^{(m)}(h^{(m)})^\top$, whose row-major vectorization is $q^{(m)} \otimes h^{(m)}$. Pulling the classification output GGN through this separable Jacobian and applying K-FAC yields the module-wise Kronecker-factored curvature approximation

$$\Lambda_t^{(m)} \approx G_{B,t}^{(m)} \otimes H_t^{(m)}, \tag{6}$$

where $H_t^{(m)} := \mathbb{E}_{\mathcal{D}_t}[h^{(m)}(h^{(m)})^\top]$ and $G_{B,t}^{(m)} := (B^{(m)})^\top C_{t,\text{out}}^{(m)} B^{(m)}$, with $C_{t,\text{out}}^{(m)}$ the classification GGN block at the module-m output; Appendix A.1 gives the full definition, the row-major Kronecker order, and why this is a classification GGN rather than an empirical Fisher. Across modules, we use the block-diagonal approximation $\Lambda_t \approx \text{blkdiag}(\Lambda_t^{(1)}, \ldots, \Lambda_t^{(M)})$, which preserves dominant intra-module second-order structure while discarding cross-module curvature terms.

4.2 Shared Curvature-Informed Subspace

Even with Kronecker structure, inference in the full LoRA space remains costly. We therefore restrict each slice-dependent deviation to a shared low-dimensional subspace [40]. For module m, define the average curvature factors

$$\bar{H}^{(m)} = \frac{1}{T}\sum_{t=1}^{T} H_t^{(m)}, \qquad \bar{G}_B^{(m)} = \frac{1}{T}\sum_{t=1}^{T} G_{B,t}^{(m)}. \tag{7}$$

Let $\bar{H}^{(m)} u_i^{(m)} = \alpha_i^{(m)} u_i^{(m)}$ and $\bar{G}_B^{(m)} v_j^{(m)} = \beta_j^{(m)} v_j^{(m)}$ be the corresponding eigendecompositions. We select the top-L_m Kronecker directions according to the products $\alpha_i^{(m)}\beta_j^{(m)}$ and assemble the resulting basis into $U^{(m)} \in \mathbb{R}^{d_m \times L_m}$, where d_m is the number of LoRA-A parameters in module m. Stacking all modules yields the global projection $U = \text{blkdiag}(U^{(1)}, \ldots, U^{(M)}) \in \mathbb{R}^{d \times L}$, with $L = \sum_{m=1}^{M} L_m$.

We write the global latent state as $x_t = \text{concat}_{m=1}^{M} x_t^{(m)}$, where $x_t^{(m)} \in \mathbb{R}^{L_m}$, and parameterize the deviations as $\delta a_t \approx U x_t$. Equivalently, the module-m deviation is $\delta a_t^{(m)} \approx U^{(m)} x_t^{(m)}$. This yields module-wise projected slice-wise gradients and curvatures:

$$g_t^{(x,m)} := \big(U^{(m)}\big)^\top g_t^{(m)}, \qquad \Lambda_t^{(x,m)} := \big(U^{(m)}\big)^\top \Lambda_t^{(m)} U^{(m)}. \tag{8}$$

The explicit latent-space assembly from the Kronecker eigenspace is given in Appendix A.2.

4.3 Gaussian Pseudo-Observations and Sequential Inference

Crucially, the projected slice surrogate admits an exact Gaussian pseudo-observation representation. For each module m, introduce a small damping constant $\lambda > 0$ and define $\widetilde{\Lambda}_t^{(x,m)} := \Lambda_t^{(x,m)} + \lambda I_{L_m}$. The damping makes $\widetilde{\Lambda}_t^{(x,m)} \succ 0$, enabling stable Cholesky factorization. The projected slice surrogate becomes

$$\widetilde{\mathcal{L}}_t^{(m)}(x_t^{(m)}) = \text{const} + \big(g_t^{(x,m)}\big)^\top x_t^{(m)} + \frac{1}{2}\big(x_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)}. \tag{9}$$

By defining the local shift $\mu_t^{(m)} := -(\widetilde{\Lambda}_t^{(x,m)})^{-1} g_t^{(x,m)}$, the Cholesky decomposition $\widetilde{\Lambda}_t^{(x,m)} = (R_t^{(m)})^\top R_t^{(m)}$, and $\widetilde{y}_t^{(m)} := R_t^{(m)} \mu_t^{(m)}$, completing the square gives $\widetilde{\mathcal{L}}_t^{(m)}(x_t^{(m)}) = \frac{1}{2}\|\widetilde{y}_t^{(m)} - R_t^{(m)} x_t^{(m)}\|_2^2 + \text{const}$, i.e., $\exp(-\widetilde{\mathcal{L}}_t^{(m)}(x_t^{(m)})) \propto \mathcal{N}(\widetilde{y}_t^{(m)}; R_t^{(m)} x_t^{(m)}, I_{L_m})$. Consequently, each slice induces a Gaussian pseudo-observation $\widetilde{y}_t^{(m)}$, yielding the observation model:

$$\widetilde{y}_t^{(m)} = R_t^{(m)} x_t^{(m)} + \varepsilon_t^{(m)}, \qquad \varepsilon_t^{(m)} \sim \mathcal{N}(0, I_{L_m}). \tag{10}$$

The full derivation, including Proposition 1, is provided in Appendix A.3. The pseudo-observations are neither additional observed data nor a separately assumed likelihood: they are an exact algebraic rewriting of each damped projected quadratic surrogate via completing the square. Thus, Kalman

inference is exact within the induced LGSSM, while the resulting LoRA posterior remains a structured approximation with respect to the original LLM likelihood. This construction is consistent with generalized Bayesian updating [41], in which beliefs are updated through loss terms—here the projected slice-wise surrogates—rather than the raw likelihood. To couple these ordered slices, we place a Gaussian random-walk prior on each module-specific latent chain, formulating the complete LGSSM:

$$x_1^{(m)} \sim \mathcal{N}\big(m_1^{(m)}, P_1^{(m)}\big), \tag{11}$$

$$x_t^{(m)} = x_{t-1}^{(m)} + \eta_t^{(m)}, \qquad \eta_t^{(m)} \sim \mathcal{N}(0, Q^{(m)}). \tag{12}$$

Here, t indexes the ordered sequence of slice-wise posterior evidence $(R_t^{(m)}, \widetilde{y}_t^{(m)})$ induced by the chosen partition in the shared latent LoRA subspace. The random walk is a first-order smoothing prior over this evidence sequence, not a temporal data-generating model of a static dataset: it couples adjacent latent states while nonzero $Q^{(m)}$ permits slice-specific drift, and it is a simple default, not a uniquely optimal transition (Appendix C.7). When $Q^{(m)} = 0$, all pseudo-observations collapse onto one shared latent state, yielding a pooled-evidence, subspace-Laplace-style internal baseline; Section 4.7.2 compares the random walk with this tied-state pooling and with independent per-slice posteriors without transitions (Ind). We use a time-homogeneous process covariance $Q^{(m)} = s_q q_m I_{L_m}$ within each module, with $s_q = 1$ fixed and q_m computed before filtering from training-side, curvature-normalized adjacent-slice drift, not fitted by expectation–maximization or tuned on evaluation data (Appendix A.5). Since the induced model is linear and Gaussian, inference within it is exact via Kalman filtering [30, 31]. Rauch–Tung–Striebel (RTS) smoothing [42] can recover latent trajectories, but our default prediction uses only the terminal filtered state.

4.4 Posterior Prediction and Complexity

For each module m, we use the terminal filtered posterior under the chosen ordered partition,

$$x_T^{(m)} \mid \widetilde{y}_{1:T}^{(m)} \sim \mathcal{N}\big(m_{T|T}^{(m)}, P_{T|T}^{(m)}\big). \tag{13}$$

RTS smoothing leaves the terminal marginal unchanged; averaging intermediate states would instead mix slice-stage adapters rather than the final posterior. For posterior prediction, we first draw S Monte Carlo samples $x_T^{(m,s)} \sim \mathcal{N}(m_{T|T}^{(m)}, P_{T|T}^{(m)})$. We then map each latent sample back to the LoRA parameter space:

$$A_T^{(m,s)} = \hat{A}^{(m)} + \mathrm{mat}^{(m)}\big(U^{(m)} \tau x_T^{(m,s)}\big), \tag{14}$$

where $\mathrm{mat}^{(m)}(\cdot)$ reshapes the latent vector back to the shape of $A^{(m)}$, and $\tau \in [0, 1]$ is a weight-space posterior-scale shrinkage factor, not a logit-temperature parameter (Appendices A.6 and C.6). The corresponding sampled adapter weight is

$$W_T^{(m,s)} = W_0^{(m)} + B^{(m)} A_T^{(m,s)}. \tag{15}$$

The Bayesian predictive distribution for a test input ξ is approximated by Monte Carlo model averaging [19, 43]:

$$p(y \mid \xi, \mathcal{D}_{1:T}) \approx \frac{1}{S} \sum_{s=1}^{S} p\big(y \mid \xi, \{W_T^{(m,s)}\}_{m=1}^{M}\big). \tag{16}$$

Because the projected model is block-diagonal across modules, inference decomposes naturally into module-wise low-dimensional systems. Once the subspace is fixed, the total Kalman filtering cost is $\mathcal{O}(T \sum_{m=1}^{M} L_m^3)$, with a state memory footprint of $\mathcal{O}(\sum_{m=1}^{M} L_m^2)$, entirely bypassing the need to form or invert full $d \times d$ curvature matrices. The induced low-rank posterior support is discussed in Appendix A.4. A complete algorithmic summary is provided in Algorithm 1 in Appendix A.8.

4.5 Experimental Setup

Backbone and adaptation setup. We use Qwen3-8B-Base [44] as the shared backbone and apply LoRA to the query, value, and language-model-head projections (`q_proj`, `v_proj`, and `lm_head`). All methods use the same backbone, prompt template, answer-label space, LoRA target modules, rank, and scaling. This controlled setup isolates the effect of posterior modeling from differences in model capacity, answer scoring, or decoding protocol. Full experimental details, including hyperparameters and reproducibility notes, are in Appendix B.

Datasets and evaluation protocol. The MAP adapter is trained on closed-choice ScienceQA over grades 2–11 [45], which provides a natural grade-based ordered partition for Seq-LoRA. We evaluate on the held-out ScienceQA grades 2–11 test split as the in-distribution (ID) benchmark. To assess robustness under distribution shift, we further evaluate on ScienceQA Grade12 [45], OpenBookQA (OBQA) [46], ARC-Challenge (ARC-C) [47], two grouped MMLU science benchmarks at high-school (MMLU-H) and college (MMLU-C) levels derived from MMLU [48], and GPQA [49]. Dataset sizes, split usage, answer-choice counts, and ID/OOD groupings are summarized in Table 4.

For each example, we score only the answer-label tokens corresponding to valid choices and apply a softmax over the valid answer set. Invalid answer labels are masked for variable-choice examples. We report Accuracy (ACC), Negative Log-Likelihood (NLL), and Expected Calibration Error (ECE), with ECE computed using 10 equal-width bins. Appendix B.4 formalizes the candidate-answer restricted evaluation protocol used throughout.

Baselines and model selection. We compare Seq-LoRA with the frozen base model, deterministic MAP fine-tuning, MC-Dropout [32], Deep Ensemble [33], Laplace-LoRA [13], BLoB [14], TFB [21], and C-LoRA [15]. Post-hoc methods, including Laplace-LoRA, TFB, and Seq-LoRA, are applied to the same fixed MAP checkpoint. When a method requires an anchor set, curvature fitting, or data-dependent posterior construction, the required data are drawn only from the ScienceQA grades 2–11 training split. No method uses any ID test example or OOD benchmark for checkpoint selection, temperature scaling, posterior-scale selection, or hyperparameter tuning.

Posterior-predictive evaluation. All methods are evaluated with the same tokenization, prompt format, answer-token space, prediction interface, and metric implementation. For posterior-predictive methods, we average predictive probabilities over Monte Carlo samples before computing ACC, NLL, and ECE. Method-specific sample counts and batching details are reported in Appendix B.6.

4.6 Main Results

Table 1 summarizes the ID and OOD results. Seq-LoRA targets posterior-predictive calibration rather than top-1 accuracy maximization, with likelihood and calibration gains concentrated on OOD targets. Its calibration-oriented operating point is set by the posterior-scale parameter τ, chosen automatically (auto-τ) so that the mean KL between MAP and τ-scaled posterior predictions on training anchors matches a fixed target of 0.05 (Appendix A.6; $\tau \in [0.58, 0.67]$ over five seeds at the default $T = 10$). Appendix C.6 (Figures 2–4) shows that auto-τ is not a validation-tuned optimum but balances ID task preservation with OOD likelihood and calibration, and that fixed nonzero scales already improve OOD NLL/ECE over MAP. Relative to MAP, Seq-LoRA changes ID accuracy by -0.25 points, matches ARC-C, improves MMLU-H by 0.21 points, and incurs drops of 0.64–1.61 points on the remaining OOD targets. On ID, its NLL and ECE are slightly worse than Laplace-LoRA's and BLoB's.

Seq-LoRA shows its clearest advantage in likelihood and calibration under distribution shift. Among fine-tuned methods, it achieves the best NLL on five of the six OOD benchmarks and the best ECE on four. The gains are especially pronounced on the more difficult science benchmarks. On ARC-C, MMLU-C, and GPQA, Seq-LoRA reduces MAP NLL/ECE from $0.43/6.93$, $1.09/15.97$, and $2.01/34.13$ to $0.24/1.47$, $0.74/6.43$, and $1.52/20.73$, respectively. The improvement profile reflects a regime-dependent pattern: MAP is near-optimal on ID (so all Bayesian methods cluster tightly), miscalibrated but accurate on smaller and larger OOD shifts (where Seq-LoRA's gains are largest), and unreliable on extreme shift such as GPQA, where posterior averaging can correct overconfidence but cannot supply missing task knowledge. Overall, these results suggest that sequential posterior inference can retain much of the benefit of LoRA adaptation while substantially reducing the overconfidence of a deterministic MAP adapter under distribution shift.

4.7 Ablation Studies

All ablations use the same backbone, LoRA setup, prompt format, answer scoring, and metrics as Section 4.5. We use the same ID/OOD targets and group results into In-Distribution, Smaller Shift, and Larger Shift regimes for readability. The MAP order ablation controls for ordinary curriculum effects: relative to the natural grade exposure order used to train the shared MAP anchor, reverse or random exposure does not consistently improve OOD likelihood or calibration (Appendix

Table 1: Main benchmark results on ID and OOD evaluation. ID denotes the held-out ScienceQA closed-choice test split for grades 2–11. For variable-choice examples, invalid answer labels are masked before computing metrics over valid choices. Except for the frozen base model (†), results are mean $\pm$ standard deviation over five seeds; ACC and ECE are percentages. Best and second-best non-reference results are in **bold** and underlined, with ties after rounding broken by smaller standard deviation.

		In-Dist.	Smaller Shift			Larger Shift		
Metric	**Method**	**SciQA G2–11**	**Grade12**	**OBQA**	**ARC-C**	**MMLU-H**	**MMLU-C**	**GPQA**
ACC ↑	Base†	78.43 ± 0.00	72.90 ± 0.00	78.40 ± 0.00	88.65 ± 0.00	79.07 ± 0.00	68.50 ± 0.00	30.80 ± 0.00
	MAP	94.11 ± 0.31	93.16 ± 0.84	88.36 ± 0.09	91.50 ± 0.50	81.33 ± 0.11	73.53 ± 0.60	**36.65 ± 0.37**
	MC-Drop	94.13 ± 0.33	93.29 ± 1.08	**88.44 ± 0.26**	91.55 ± 0.50	81.39 ± 0.28	73.58 ± 0.67	35.94 ± 0.88
	Ensemble	**94.16 ± 0.12**	93.42 ± 0.18	88.12 ± 0.67	**91.71 ± 0.25**	81.08 ± 0.31	74.16 ± 0.44	36.61 ± 0.52
	Laplace	94.03 ± 0.40	**93.49 ± 0.77**	88.44 ± 0.43	91.56 ± 0.46	81.11 ± 0.50	72.95 ± 0.53	36.47 ± 0.90
	BLoB	93.35 ± 0.16	91.23 ± 0.53	87.92 ± 0.18	91.47 ± 0.48	**81.99 ± 0.27**	**74.34 ± 0.43**	35.09 ± 0.62
	TFB	94.11 ± 0.45	93.03 ± 0.74	88.44 ± 0.30	91.54 ± 0.42	81.57 ± 0.44	72.95 ± 1.15	35.71 ± 1.29
	C-LoRA	92.32 ± 0.20	89.29 ± 1.34	86.36 ± 0.54	91.14 ± 0.22	80.27 ± 0.69	73.53 ± 1.48	35.58 ± 1.14
	Seq-LoRA	93.86 ± 0.17	92.52 ± 0.74	87.52 ± 0.52	91.50 ± 0.21	81.54 ± 0.73	72.08 ± 1.18	35.04 ± 1.17
ECE ↓	Base†	3.93 ± 0.00	10.64 ± 0.00	8.31 ± 0.00	6.17 ± 0.00	3.22 ± 0.00	6.89 ± 0.00	25.13 ± 0.00
	MAP	3.77 ± 0.32	5.53 ± 0.62	8.91 ± 0.24	6.93 ± 0.50	13.36 ± 0.34	15.97 ± 0.85	34.13 ± 0.79
	MC-Drop	3.78 ± 0.35	5.61 ± 0.54	8.71 ± 0.24	6.94 ± 0.28	13.18 ± 0.42	16.07 ± 0.96	34.76 ± 0.88
	Ensemble	3.19 ± 0.06	4.64 ± 0.36	7.50 ± 0.53	6.22 ± 0.24	12.50 ± 0.36	14.64 ± 0.55	32.97 ± 0.66
	Laplace	1.20 ± 0.33	4.28 ± 0.73	3.06 ± 0.66	3.02 ± 0.36	6.93 ± 0.75	8.20 ± 0.69	23.07 ± 1.27
	BLoB	**1.17 ± 0.27**	4.56 ± 0.68	3.58 ± 0.88	3.96 ± 0.47	6.27 ± 0.53	7.59 ± 0.77	20.58 ± 0.57
	TFB	1.82 ± 0.47	**3.39 ± 0.73**	3.46 ± 0.79	3.31 ± 1.06	6.60 ± 0.90	8.46 ± 0.74	20.88 ± 1.20
	C-LoRA	1.42 ± 0.15	3.67 ± 0.62	4.18 ± 1.19	3.76 ± 0.37	7.19 ± 0.35	7.40 ± 0.94	**19.88 ± 1.00**
	Seq-LoRA	1.85 ± 0.48	4.32 ± 0.64	**2.27 ± 0.51**	**1.47 ± 0.23**	**3.67 ± 0.56**	**6.43 ± 1.56**	20.73 ± 1.57
NLL ↓	Base†	0.47 ± 0.00	0.62 ± 0.00	0.61 ± 0.00	0.35 ± 0.00	0.59 ± 0.00	0.77 ± 0.00	1.63 ± 0.00
	MAP	0.19 ± 0.01	0.33 ± 0.01	0.56 ± 0.02	0.43 ± 0.01	0.84 ± 0.02	1.09 ± 0.03	2.01 ± 0.03
	MC-Drop	0.18 ± 0.01	0.34 ± 0.01	0.56 ± 0.01	0.43 ± 0.01	0.84 ± 0.02	1.09 ± 0.03	2.01 ± 0.03
	Ensemble	0.16 ± 0.01	0.30 ± 0.01	0.49 ± 0.01	0.39 ± 0.01	0.79 ± 0.02	1.05 ± 0.02	1.96 ± 0.02
	Laplace	**0.13 ± 0.00**	**0.23 ± 0.01**	0.38 ± 0.01	0.27 ± 0.00	0.61 ± 0.02	0.86 ± 0.02	1.69 ± 0.01
	BLoB	0.13 ± 0.00	0.24 ± 0.01	0.37 ± 0.01	0.27 ± 0.01	0.58 ± 0.01	0.80 ± 0.01	1.59 ± 0.01
	TFB	0.15 ± 0.01	0.25 ± 0.01	0.39 ± 0.01	0.29 ± 0.01	0.59 ± 0.01	0.82 ± 0.02	1.54 ± 0.03
	C-LoRA	0.15 ± 0.00	0.26 ± 0.02	0.41 ± 0.02	0.28 ± 0.01	0.61 ± 0.01	0.80 ± 0.02	1.58 ± 0.01
	Seq-LoRA	0.14 ± 0.00	0.24 ± 0.01	**0.35 ± 0.01**	**0.24 ± 0.00**	**0.53 ± 0.01**	**0.74 ± 0.02**	**1.52 ± 0.02**

Table 12). We therefore focus on the post-hoc Seq-LoRA mechanism: fixed-MAP slice-wise evidence construction and aggregation. Full sweeps are in Appendix C.4, including reverse-grade LGSSM.

4.7.1 Effect of Slice Granularity

Table 2 shows that moving from a single aggregate posterior ($T = 1$) to multi-slice partitions ($T = 5$ or $T = 10$) substantially reduces ECE and NLL, so multi-slice posterior evidence adds useful uncertainty beyond one aggregate posterior. This suggests that a single pooled surrogate hides useful slice-level heterogeneity in the local curvature and pseudo-observation evidence. The gains saturate by $T = 20$, where finer slices do not consistently improve over $T = 10$, suggesting a granularity trade-off: increasing T exposes more local variation, but also reduces the number of examples per slice and can make curvature and pseudo-observation estimates less stable. Accuracy is comparatively insensitive to T, reinforcing that Seq-LoRA mainly changes the posterior predictive distribution rather than the deterministic decision boundary. The default $T = 10$ is not selected by this ablation; it is the natural grade-aligned ScienceQA partition, chosen a priori from grades 2 through 11. Appendix C.6 extends this to a denser T-sweep up to $T = 40$ (Figure 2): with automatic τ, the metrics show no systematic gain beyond $T \approx 10$, whereas fixed scales keep changing with T (unshrunk $\tau = 1$ increasingly degrades ID ACC, NLL, and ECE), so granularity and posterior scale should be read jointly.

4.7.2 Slice Evidence Aggregation and Coupling

Table 2 shows the value of multi-slice evidence over pooled $T = 1$; Table 3 then fixes $T = 10$ and isolates how this slice evidence is aggregated. Here, "grade" uses natural grade slices, "random" uses size-balanced random slices, "Ind" (Independent) removes LGSSM transition coupling, and $Q = 0$ is the tied-state pooled-evidence limit, a subspace-Laplace-style internal baseline. The $Q = 0$ row improves over MAP (Larger Shift ECE $21.15 \to 17.32$) but remains well behind nonzero-Q LGSSM variants (larger-shift ECE/NLL $17.32/1.12$ vs. $10.28/0.93$ for LGSSM grade). Thus, slice-wise

Table 2: Effect of slice granularity T for Seq-LoRA (mean $\pm$ standard deviation over five seeds). T is the number of disjoint ScienceQA G2–11 training slices: $T = 1$ is pooled, $T = 10$ is the default grade-aligned partition, $T = 5$ merges adjacent grade pairs, and $T = 20$ splits each grade into two near-equal halves. Best and second-best entries are selected before rounding.

		In-Dist.	Smaller Shift			Larger Shift		
Metric	**T**	**SciQA G2–11**	**Grade12**	**OBQA**	**ARC-C**	**MMLU-H**	**MMLU-C**	**GPQA**
ACC ↑	T=1	**94.03 ± 0.31**	**92.97 ± 0.70**	**88.28 ± 0.46**	91.58 ± 0.42	81.05 ± 0.59	**73.76 ± 0.90**	**35.80 ± 1.05**
	T=5	93.94 ± 0.10	92.39 ± 1.04	87.72 ± 0.88	**91.60 ± 0.33**	**81.75 ± 0.42**	72.72 ± 0.95	35.54 ± 0.90
	T=10	93.86 ± 0.17	92.52 ± 0.74	87.52 ± 0.52	91.50 ± 0.21	81.54 ± 0.73	72.08 ± 1.18	35.04 ± 1.17
	T=20	93.97 ± 0.40	92.19 ± 1.15	87.56 ± 0.85	91.34 ± 0.26	81.38 ± 0.52	71.96 ± 1.10	34.64 ± 1.33
ECE ↓	T=1	2.70 ± 0.29	4.59 ± 0.95	6.09 ± 0.29	5.37 ± 0.41	10.29 ± 0.59	12.57 ± 1.00	29.10 ± 1.68
	T=5	**1.57 ± 0.58**	4.53 ± 0.99	2.57 ± 0.71	1.79 ± 0.44	3.88 ± 0.78	7.32 ± 1.07	**20.72 ± 2.04**
	T=10	1.85 ± 0.48	**4.32 ± 0.64**	**2.27 ± 0.51**	1.47 ± 0.23	3.67 ± 0.56	**6.43 ± 1.56**	20.73 ± 1.57
	T=20	1.68 ± 0.27	4.83 ± 1.65	2.48 ± 0.79	**1.44 ± 0.13**	**3.43 ± 0.79**	6.49 ± 0.41	20.78 ± 1.75
NLL ↓	T=1	0.14 ± 0.01	0.26 ± 0.01	0.42 ± 0.01	0.31 ± 0.01	0.66 ± 0.02	0.92 ± 0.04	1.77 ± 0.05
	T=5	0.14 ± 0.01	**0.23 ± 0.01**	**0.34 ± 0.01**	0.24 ± 0.01	0.54 ± 0.02	0.76 ± 0.02	1.54 ± 0.04
	T=10	**0.14 ± 0.00**	0.24 ± 0.01	0.35 ± 0.01	**0.24 ± 0.00**	0.53 ± 0.01	**0.74 ± 0.02**	**1.52 ± 0.02**
	T=20	0.14 ± 0.01	0.24 ± 0.01	0.35 ± 0.01	0.24 ± 0.00	**0.53 ± 0.01**	0.75 ± 0.01	1.53 ± 0.02

Table 3: Key slice-evidence and coupling ablations at $T = 10$ (mean $\pm$ standard deviation over five seeds). Smaller/Larger Shift are unweighted macro-averages over Grade12/OBQA/ARC-C and MMLU-H/MMLU-C/GPQA. Ind (Independent): no transition; one posterior per slice, with predictions averaged over slices (Appendix A.7). Best and second-best entries are in **bold** and underlined, with ties after rounding broken by smaller standard deviation. Per-task results are in Table 15.

Metric	**Method**	**In-Dist.**	**Smaller Shift**	**Larger Shift**
ACC ↑	MAP	94.11 ± 0.31	91.01 ± 0.44	**63.84 ± 0.21**
	LGSSM grade	93.86 ± 0.17	90.51 ± 0.39	62.89 ± 0.98
	LGSSM random	93.89 ± 0.27	90.24 ± 0.63	63.36 ± 0.50
	LGSSM $Q = 0$ pooled	94.07 ± 0.27	**91.12 ± 0.55**	63.68 ± 0.62
	Ind grade	93.80 ± 0.20	90.25 ± 0.51	62.95 ± 0.55
	Ind random	**94.11 ± 0.22**	90.80 ± 0.34	63.39 ± 0.46
ECE ↓	MAP	3.77 ± 0.32	7.12 ± 0.34	21.15 ± 0.56
	LGSSM grade	1.85 ± 0.48	**2.69 ± 0.22**	10.28 ± 1.05
	LGSSM random	1.87 ± 0.52	3.03 ± 0.43	**10.00 ± 0.80**
	LGSSM $Q = 0$ pooled	2.77 ± 0.18	5.69 ± 0.47	17.32 ± 0.66
	Ind grade	1.41 ± 0.36	2.81 ± 0.35	11.38 ± 0.63
	Ind random	**1.22 ± 0.32**	3.90 ± 0.18	13.96 ± 1.00
NLL ↓	MAP	0.19 ± 0.01	0.44 ± 0.01	1.31 ± 0.02
	LGSSM grade	0.14 ± 0.00	**0.28 ± 0.00**	**0.93 ± 0.01**
	LGSSM random	0.14 ± 0.00	0.28 ± 0.01	0.94 ± 0.02
	LGSSM $Q = 0$ pooled	0.14 ± 0.01	0.34 ± 0.01	1.12 ± 0.03
	Ind grade	0.14 ± 0.01	0.28 ± 0.01	0.96 ± 0.01
	Ind random	**0.13 ± 0.00**	0.29 ± 0.01	1.01 ± 0.03

pseudo-observations already contain useful local posterior evidence, but full tied-state pooling is insufficient. This is a structural boundary comparison, not a continuous sweep over s_q.

We read Table 3 as partition $\times$ aggregator (larger-shift ECE/NLL). With independent per-slice posteriors, grade slices outperform random slices (Ind grade $11.38/0.96$ vs. Ind random $13.96/1.01$); Ind random attains the lowest ECE only in-distribution (1.22). Holding the partition and slice-local surrogates fixed within each run, replacing Ind by LGSSM aggregation helps the tested random slices more (ECE $13.96 \to 10.00$, -3.96 points) than the grade slices ($11.38 \to 10.28$, -1.10 points), so LGSSM grade and LGSSM random end up comparable ($10.28/0.93$ vs. $10.00/0.94$): semantic slice structure can help but is not required. We hypothesize that random slices carry weaker individual evidence, which benefits more from aggregation into one terminal posterior. These gains should

not be read as the net effect of coupling: the Ind/LGSSM comparison changes both the transition structure and the terminal-posterior aggregation, and each variant selects its own τ.

Seq-LoRA should therefore be interpreted as a structured aggregator of slice-relative evidence in latent LoRA space, rather than as a search for one privileged curriculum order. Changing the order or partition changes how local evidence is smoothed and propagated to the terminal posterior, so different variants can induce different predictive preferences. Appendix Table 15 reports the full ordering sweep, including reverse-grade, shuffled-grade, and base-NLL orders; these variants all have lower NLL and ECE than MAP on every target, by target-dependent margins. Computed from the five nonzero-Q LGSSM rows of Table 15, larger-shift macro-averages span 10.00–11.68 ECE and 0.93–0.96 NLL, versus 21.15/1.31 for MAP, so the gains are not simply terminal bias toward high-grade slices. At 80% coverage, LGSSM random has lower selective NLL than Ind random on all six OOD targets (e.g., GPQA 1.554 vs. 1.681; Appendix C.5, Table 16). Because OBQA lacks grade-like slice metadata, OBQA-to-ARC/MMLU transfer (Appendix C.3) uses the semantics-free fallback of Appendix A.7. For each backbone (Qwen3-8B-Base and Llama-2-7B), Seq-LoRA's reported mean ECE and NLL are below MAP's on all four of its OOD targets (Tables 13 and 14; the MMLU targets differ between backbones), while Llama-2-7B's ID OBQA ACC drops from 77.93 (MAP) to 76.80 (1.13 points). This suggests that the gains are not specific to one backbone or to grade-aligned slices. Finally, Seq-LoRA's post-hoc fit takes about as long as one MAP training run (1014.61 vs. 933.76 s) and peaks at 26.13 GB reserved GPU memory, versus 94.28 GB for Laplace-LoRA (Appendix C.1).

5 Conclusion

We introduced **Seq-LoRA**, a post-hoc Bayesian framework for uncertainty-aware PEFT that partitions fine-tuning data into slices, builds slice-wise quadratic evidence around a shared MAP LoRA anchor, and performs posterior inference in a curvature-aligned subspace via a linear–Gaussian state-space model. Empirically, it improves NLL and calibration under distribution shift while preserving most task accuracy; ablations support the value of multi-slice evidence and LGSSM aggregation rather than a privileged curriculum order, with the random-walk prior acting as a stable aggregator of slice-wise evidence even with semantics-free random slices.

Limitations. Seq-LoRA has two modeling limitations: a minimal random-walk prior, which may miss abrupt or recurring slice regimes, and rank-L subspace uncertainty, which excludes directions outside U. It is also a one-shot, offline posterior construction method: it compresses a fixed training distribution into one terminal posterior and does not update as new data, tasks, or distributions arrive. Its posterior is conditional on the chosen partition and ordering, whose sensitivity we do not isolate: we claim effectiveness across the tested slice constructions, not partition/order invariance or per-target optimality, and leave a controlled fixed-MAP sensitivity analysis over partitions and orderings to future work. Our evaluation is limited to closed-choice QA with predominantly science-oriented shifts; instruction tuning, commonsense reasoning, and open-ended factual QA remain outside the current evidence. Appendix C.7 discusses these limitations and future directions.

Broader impact. Better-calibrated adapted LLMs can help users decide when to defer or abstain, but improved benchmark calibration does not guarantee reliable confidence in deployment: Seq-LoRA inherits the biases, failure modes, and misuse potential of the underlying LLM and fine-tuning data, and its confidence estimates may invite over-reliance without application-specific validation.

Acknowledgments and Disclosure of Funding

Dr. Wang's research was fully supported by the National Science Foundation Award No. 1848451. The authors declare no competing interests.

References

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

[2] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

[3] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[5] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5:220–235, 2023. doi: 10.1038/s42256-023-00626-4. URL https://doi.org/10.1038/s42256-023-00626-4.

[6] Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning, 2023. URL https://arxiv.org/abs/2303.15647.

[7] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey, 2024. URL https://arxiv.org/abs/2403.14608.

[8] Kaiyang Li, Shaobo Han, Qing Su, Wei Li, Zhipeng Cai, and Shihao Ji. Uni-LoRA: One vector is all you need. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://neurips.cc/virtual/2025/poster/116600.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

[10] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2796–2804. PMLR, 2018. URL https://proceedings.mlr.press/v80/kuleshov18a.html.

[11] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d76621371888657d2eb1d-Abstract.html.

[12] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 15682–15694.

Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8420d359404024567b5aefda1231af24-Abstract.html.

[13] Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FJiUyzOF1m.

[14] Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. BLoB: Bayesian low-rank adaptation by backpropagation for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 67758–67794, 2024. doi: 10.52202/079017-2164. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7d53575463291ea6b5a23cf6e571f59b-Abstract-Conference.html.

[15] Amir Hossein Rahmati, Sanket Jantre, Weifeng Zhang, Yucheng Wang, Byung-Jun Yoon, Nathan M. Urban, and Xiaoning Qian. C-LoRA: Contextual low-rank adaptation for uncertainty estimation in large language models, 2025. URL https://arxiv.org/abs/2505.17773.

[16] Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973. doi: 10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2.

[17] A. Philip Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. doi: 10.1080/01621459.1982.10477856.

[18] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, pages 2901–2907, 2015. doi: 10.1609/aaai.v29i1.9602.

[19] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992. doi: 10.1162/neco.1992.4.3.448.

[20] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622. PMLR, 2015. URL https://proceedings.mlr.press/v37/blundell15.html.

[21] Haizhou Shi, Yibin Wang, Ligong Han, Huan Zhang, and Hao Wang. Training-free bayesianization for low-rank adapters of large language models, 2024. URL https://arxiv.org/abs/2412.05723.

[22] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA, 2009. Association for Computing Machinery. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[23] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pages 1189–1197. Curran Associates, Inc., 2010. URL https://papers.nips.cc/paper_files/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

[24] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1119. URL https://aclanthology.org/N19-1119/.

[25] Bruce W. Lee, Hyunsoo Cho, and Kang Min Yoo. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1281–1309, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.82. URL https://aclanthology.org/2024.findings-naacl.82/.

[26] Jisu Kim and Juhwan Lee. Strategic data ordering: Enhancing large language model performance through curriculum learning, 2024. URL https://arxiv.org/abs/2405.07490.

[27] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2408–2417. PMLR, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

[28] Roger Grosse and James Martens. A Kronecker-factored approximate Fisher matrix for convolution layers. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 573–582. PMLR, 2016. URL https://proceedings.mlr.press/v48/grosse16.html.

[29] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 557–565. PMLR, 2017. URL https://proceedings.mlr.press/v70/botev17a.html.

[30] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

[31] Simo S"arkk"a and Lennart Svensson. *Bayesian Filtering and Smoothing*, volume 17 of *Institute of Mathematical Statistics Textbooks*. Cambridge University Press, 2 edition, 2023. doi: 10.1017/9781108917407. URL https://doi.org/10.1017/9781108917407.

[32] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

[33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

[34] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux—effortless bayesian deep learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 20089–20103, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a7c9585703d275249f30a088cebba0ad-Abstract.html.

[35] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skdvd2xAZ.

[36] Victor Letzelter, Hugo Malard, Mathieu Fontaine, Gaël Richard, Slim Essid, Andrei Bursuc, and Patrick Pérez. Multiple choice learning of low-rank adapters for language modeling. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 65628–65667. PMLR, 2026. URL https://proceedings.mlr.press/v306/letzelter26a.html.

[37] Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.542. URL https://aclanthology.org/2020.acl-main.542/.

[38] Mohamed Elgaar and Hadi Amiri. HuCurl: Human-induced curriculum discovery. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1862–1877, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.104. URL https://aclanthology.org/2023.acl-long.104/.

[39] Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. What makes a good curriculum? disentangling the effects of data ordering on LLM mathematical reasoning, 2025. URL https://arxiv.org/abs/2510.19099.

[40] Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, volume 115 of *Proceedings of Machine Learning Research*, pages 1169–1179. PMLR, 2020. URL https://proceedings.mlr.press/v115/izmailov20a.html.

[41] Pier Giovanni Bissiri, Chris C. Holmes, and Stephen G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016. doi: 10.1111/rssb.12158. URL https://doi.org/10.1111/rssb.12158.

[42] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965. doi: 10.2514/3.3166.

[43] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*, volume 33, pages 4697–4708, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/322f62469c5e3c7dc3e58f5a4d1ea399-Abstract.html.

[44] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[45] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pages 2507–2521, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html.

[46] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

[47] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

[48] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2020. URL https://arxiv.org/abs/2009.03300.

[49] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-Proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

[50] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10248–10259. PMLR, 2020. URL https://proceedings.mlr.press/v119/wenzel20a.html.

[51] Chris C. Holmes and Stephen G. Walker. Assigning a value to a power likelihood in a general bayesian model. *Biometrika*, 104(2):497–503, 2017. doi: 10.1093/biomet/asx010. URL https://doi.org/10.1093/biomet/asx010.

[52] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[53] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. Continual learning for large language models: A survey. *CoRR*, abs/2402.01364, 2024. doi: 10.48550/arXiv.2402.01364. URL https://arxiv.org/abs/2402.01364.

[54] Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025. doi: 10.1145/3735633. URL https://doi.org/10.1145/3735633.

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

[56] Muhammad Fawi. CURLoRA: Stable LLM continual fine-tuning and catastrophic forgetting mitigation. *CoRR*, abs/2408.14572, 2024. doi: 10.48550/arXiv.2408.14572. URL https://arxiv.org/abs/2408.14572.

[57] Zeinab Borhanifard, Heshaam Faili, and Yadollah Yaghoobzadeh. Combining replay and LoRA for continual learning in natural language understanding. *Computer Speech & Language*, 90: 101737, 2025. doi: 10.1016/j.csl.2024.101737. URL https://doi.org/10.1016/j.csl.2024.101737.

A Additional Method Details for Seq-LoRA

This appendix provides derivations and implementation details deferred from Section 4. In particular, we make the classification GGN factor explicit, expand the latent-space projection induced by the Kronecker eigenspace, prove the Gaussian pseudo-observation representation of the projected slice surrogate, derive the Kalman filtering posterior and the induced LoRA-space posterior, describe the adaptive estimator used for the process covariance, and summarize the full algorithmic pipeline.

A.1 Classification GGN Curvature Factor

We make explicit the curvature factor used in Section 4.1. Let z denote the model logits at the scored answer-token position, $s^{(m)}$ the output of LoRA module m, and $J_{z,s^{(m)}}$ the Jacobian of z with respect to $s^{(m)}$. Let $p_{\hat{a}}$ be the predictive distribution over the valid answer set under the MAP adapter $\hat{a}$. The classification GGN block at the module-m output is

$$C_{t,\text{out}}^{(m)} = \mathbb{E}_{x\in\mathcal{D}_t}\left[(J_{z,s^{(m)}})^\top \left(\text{diag}(p_{\hat{a}}) - p_{\hat{a}}p_{\hat{a}}^\top\right) J_{z,s^{(m)}}\right], \tag{17}$$

where $\text{diag}(p_{\hat{a}}) - p_{\hat{a}}p_{\hat{a}}^\top$ is the Hessian of the softmax cross-entropy with respect to the logits. The LoRA-A Kronecker factor used in Equation (6) is then $G_{B,t}^{(m)} = (B^{(m)})^\top C_{t,\text{out}}^{(m)} B^{(m)}$. Because this construction uses the loss Hessian rather than the empirical gradient outer product, $G_{B,t}^{(m)}$ is a classification GGN factor restricted to LoRA-A and not an empirical-Fisher approximation. The Kronecker order in Equation (6) follows the row-major LoRA-A vectorization used throughout the paper: for $q^{(m)} = (B^{(m)})^\top g_{\text{out}}^{(m)}$, $\text{vec}_{\text{row}}(q^{(m)}(h^{(m)})^\top) = q^{(m)} \otimes h^{(m)}$, which gives $G_{B,t}^{(m)} \otimes H_t^{(m)}$. Under column-major vectorization, the same curvature block would be written as $H_t^{(m)} \otimes G_{B,t}^{(m)}$. The Kronecker form follows from the same separability. For a single module input–output pair, the Jacobian of $s^{(m)}$ with respect to $\text{vec}_{\text{row}}(A^{(m)})$ is $B^{(m)} \otimes (h^{(m)})^\top$, so by the mixed-product rule its contribution to the LoRA-A GGN is $(B^{(m)})^\top \Gamma^{(m)} B^{(m)} \otimes h^{(m)}(h^{(m)})^\top$, where $\Gamma^{(m)}$ denotes the term inside the expectation defining $C_{t,\text{out}}^{(m)}$. Following K-FAC [27], we approximate the slice expectation of this Kronecker product by the Kronecker product of the slice expectations, which yields $G_{B,t}^{(m)} \otimes H_t^{(m)}$.

A.2 Projection into the Global Kronecker Eigenspace

For each module m, let

$$\bar{H}^{(m)} u_i^{(m)} = \alpha_i^{(m)} u_i^{(m)}, \qquad \bar{G}_B^{(m)} v_j^{(m)} = \beta_j^{(m)} v_j^{(m)} \tag{18}$$

be the eigendecompositions of the averaged slice-wise factors. Under the row-major LoRA-A vectorization used throughout the paper, the averaged Kronecker block is $\bar{G}_B^{(m)} \otimes \bar{H}^{(m)}$. Since the eigendirections of a Kronecker product are Kronecker products of the factor eigendirections, its eigenpairs are given by

$$\lambda_{ij}^{(m)} = \alpha_i^{(m)} \beta_j^{(m)}, \qquad w_{ij}^{(m)} = v_j^{(m)} \otimes u_i^{(m)}. \tag{19}$$

Let $\mathcal{I}_m = \{(i_\ell, j_\ell)\}_{\ell=1}^{L_m}$ denote the selected index pairs corresponding to the top-L_m eigenvalue products $\alpha_i^{(m)}\beta_j^{(m)}$. The corresponding projection matrix is

$$U^{(m)} = \left[w_{i_1 j_1}^{(m)}, \ldots, w_{i_{L_m} j_{L_m}}^{(m)}\right] \in \mathbb{R}^{d_m \times L_m}. \tag{20}$$

The projected curvature can be computed without materializing the full Kronecker block. Using the identity

$$(a \otimes b)^\top (C \otimes D)(a' \otimes b') = (a^\top C a')(b^\top D b'), \tag{21}$$

we obtain, for any $\ell, \ell' \in \{1, \ldots, L_m\}$,

$$\left[\Lambda_t^{(x,m)}\right]_{\ell,\ell'} = \left(u_{i_\ell}^{(m)}\right)^\top H_t^{(m)} u_{i_{\ell'}}^{(m)} \cdot \left(v_{j_\ell}^{(m)}\right)^\top G_{B,t}^{(m)} v_{j_{\ell'}}^{(m)}. \tag{22}$$

Equivalently, this can be written as a Hadamard product $\Lambda_t^{(x,m)} = M_{H,t}^{(m)} \odot M_{G,t}^{(m)}$, where

$$\big[M_{H,t}^{(m)}\big]_{\ell,\ell'} = \Big(u_{i_\ell}^{(m)}\Big)^\top H_t^{(m)} u_{i_{\ell'}}^{(m)}, \qquad \big[M_{G,t}^{(m)}\big]_{\ell,\ell'} = \Big(v_{j_\ell}^{(m)}\Big)^\top G_{B,t}^{(m)} v_{j_{\ell'}}^{(m)}. \tag{23}$$

This establishes the latent-space assembly rule referred to in Section 4.2. For memory efficiency, we retain only the top `n_kfac=8` eigendirections of each Kronecker factor before forming the Kronecker product directions.

A.3 Quadratic Surrogate as a Gaussian Pseudo-Observation

We restate and prove the key equivalence used in Section 4.3.

Proposition 1. *For each module m and slice t, let*

$$\widetilde{\Lambda}_t^{(x,m)} = \big(R_t^{(m)}\big)^\top R_t^{(m)}, \qquad \mu_t^{(m)} = -\big(\widetilde{\Lambda}_t^{(x,m)}\big)^{-1} g_t^{(x,m)}, \qquad \widetilde{y}_t^{(m)} = R_t^{(m)} \mu_t^{(m)}. \tag{24}$$

Then there exists a constant $C_t^{(m)}$ (independent of $x_t^{(m)}$) such that

$$\big(g_t^{(x,m)}\big)^\top x_t^{(m)} + \frac{1}{2}\big(x_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)} = \frac{1}{2}\left\|\widetilde{y}_t^{(m)} - R_t^{(m)} x_t^{(m)}\right\|_2^2 + C_t^{(m)}. \tag{25}$$

Consequently, the exponentiated surrogate takes the form of a Gaussian observation:

$$\exp\Big(-\widetilde{\mathcal{L}}_t^{(m)}(x_t^{(m)})\Big) \propto \mathcal{N}\big(\widetilde{y}_t^{(m)};\, R_t^{(m)} x_t^{(m)},\, I_{L_m}\big). \tag{26}$$

Proof. Expanding the squared Euclidean norm yields:

$$\begin{aligned}\frac{1}{2}\left\|\widetilde{y}_t^{(m)} - R_t^{(m)} x_t^{(m)}\right\|_2^2 &= \frac{1}{2}\big(\widetilde{y}_t^{(m)}\big)^\top \widetilde{y}_t^{(m)} - \big(\widetilde{y}_t^{(m)}\big)^\top R_t^{(m)} x_t^{(m)} + \frac{1}{2}\big(x_t^{(m)}\big)^\top \big(R_t^{(m)}\big)^\top R_t^{(m)} x_t^{(m)} \\ &= \frac{1}{2}\big(\widetilde{y}_t^{(m)}\big)^\top \widetilde{y}_t^{(m)} - \big(\mu_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)} + \frac{1}{2}\big(x_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)}.\end{aligned} \tag{27}$$

By substituting the definition of the local shift, $\widetilde{\Lambda}_t^{(x,m)} \mu_t^{(m)} = -g_t^{(x,m)}$, the cross-term simplifies:

$$-\big(\mu_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)} = \big(g_t^{(x,m)}\big)^\top x_t^{(m)}. \tag{28}$$

Therefore,

$$\frac{1}{2}\left\|\widetilde{y}_t^{(m)} - R_t^{(m)} x_t^{(m)}\right\|_2^2 = \big(g_t^{(x,m)}\big)^\top x_t^{(m)} + \frac{1}{2}\big(x_t^{(m)}\big)^\top \widetilde{\Lambda}_t^{(x,m)} x_t^{(m)} + \frac{1}{2}\big(\widetilde{y}_t^{(m)}\big)^\top \widetilde{y}_t^{(m)}. \tag{29}$$

Rearranging the terms completes the proof, where the constant is explicitly given by

$$C_t^{(m)} = -\frac{1}{2}\big(\widetilde{y}_t^{(m)}\big)^\top \widetilde{y}_t^{(m)}. \tag{30}$$

□

A.4 Kalman Filtering Posterior and Induced LoRA-Space Posterior

The pseudo-observation result above converts each projected slice surrogate into a Gaussian likelihood term. For each module m, Seq-LoRA therefore uses the linear–Gaussian model

$$p\Big(\widetilde{y}_t^{(m)} \mid x_t^{(m)}\Big) = \mathcal{N}\Big(\widetilde{y}_t^{(m)}; R_t^{(m)} x_t^{(m)}, I_{L_m}\Big), \tag{31}$$

$$x_1^{(m)} \sim \mathcal{N}\Big(m_1^{(m)}, P_1^{(m)}\Big), \tag{32}$$

$$x_t^{(m)} \mid x_{t-1}^{(m)} \sim \mathcal{N}\Big(x_{t-1}^{(m)}, Q^{(m)}\Big), \qquad t = 2, \ldots, T. \tag{33}$$

In the main experiments, $m_1^{(m)} = 0$ and $P_1^{(m)} = I_{L_m}$; the exact values are reported in Appendix B.5. The process covariance $Q^{(m)}$ is specified from training-slice quantities as described in Appendix A.5.

The ID and OOD evaluation sets are reserved for final evaluation and are not used to construct the posterior, choose $Q^{(m)}$, or select the posterior-scale correction τ.

Let $m_{t|t}^{(m)}$ and $P_{t|t}^{(m)}$ denote the filtered mean and covariance of $x_t^{(m)}$ after observing $\widetilde{y}_{1:t}^{(m)}$. For $t = 1$, the prediction step uses the initial prior:

$$m_{1|0}^{(m)} = m_1^{(m)}, \qquad P_{1|0}^{(m)} = P_1^{(m)}. \tag{34}$$

For $t \geq 2$, the random-walk transition gives

$$m_{t|t-1}^{(m)} = m_{t-1|t-1}^{(m)}, \qquad P_{t|t-1}^{(m)} = P_{t-1|t-1}^{(m)} + Q^{(m)}. \tag{35}$$

The measurement update is

$$S_t^{(m)} = R_t^{(m)} P_{t|t-1}^{(m)} \big(R_t^{(m)}\big)^\top + I_{L_m}, \tag{36}$$

$$K_t^{(m)} = P_{t|t-1}^{(m)} \big(R_t^{(m)}\big)^\top \big(S_t^{(m)}\big)^{-1}, \tag{37}$$

$$m_{t|t}^{(m)} = m_{t|t-1}^{(m)} + K_t^{(m)} \left(\widetilde{y}_t^{(m)} - R_t^{(m)} m_{t|t-1}^{(m)}\right), \tag{38}$$

$$P_{t|t}^{(m)} = \left(I_{L_m} - K_t^{(m)} R_t^{(m)}\right) P_{t|t-1}^{(m)}. \tag{39}$$

In implementation we symmetrize covariance matrices after the update and add a small jitter to $S_t^{(m)}$ before solving the linear system for $K_t^{(m)}$. The precision matrices $\widetilde{\Lambda}_t^{(x,m)}$ are damped by λI_{L_m} before Cholesky factorization, which makes the pseudo-observation construction numerically stable.

After processing all T slices, the module-wise terminal filtered posterior is

$$x_T^{(m)} \mid \widetilde{y}_{1:T}^{(m)} \sim \mathcal{N}\Big(m_{T|T}^{(m)}, P_{T|T}^{(m)}\Big). \tag{40}$$

Stacking the module states gives

$$x_T \mid \widetilde{y}_{1:T} \sim \mathcal{N}\big(m_{T|T}, P_{T|T}\big), \tag{41}$$

where

$$m_{T|T} = \mathrm{concat}_{m=1}^{M}\, m_{T|T}^{(m)}, \qquad P_{T|T} = \mathrm{blkdiag}_{m=1}^{M}\, P_{T|T}^{(m)}.$$

This block-diagonal covariance follows from the module-wise curvature approximation and independent module-wise state-space filters.

Let $U = \mathrm{blkdiag}(U^{(1)}, \ldots, U^{(M)})$ be the global latent projector and let $\hat{a}$ denote the concatenated MAP LoRA-A parameter vector. The raw terminal latent posterior induces the following approximate posterior in the LoRA-A parameter space:

$$a_T \mid \mathcal{D}_{1:T} \approx \mathcal{N}\big(\hat{a} + U m_{T|T}, U P_{T|T} U^\top\big). \tag{42}$$

This LoRA-space Gaussian is degenerate: its covariance has support only in the shared curvature-aligned subspace spanned by U, while orthogonal LoRA-A directions are held fixed at the MAP anchor. This low-rank support is the explicit efficiency approximation that makes posterior sampling tractable. The posterior-scale correction used for prediction rescales this displacement:

$$a_T^{(\tau)} \mid \mathcal{D}_{1:T} \approx \mathcal{N}\big(\hat{a} + \tau U m_{T|T}, \tau^2 U P_{T|T} U^\top\big), \qquad 0 \leq \tau \leq 1. \tag{43}$$

Equation (42) is useful for interpreting the approximation, but the implementation does not form the full $d \times d$ covariance $U P_{T|T} U^\top$. Instead, it samples in the low-dimensional latent space and maps each sample back to the LoRA-A factors:

$$x_T^{(m,s)} \sim \mathcal{N}\Big(m_{T|T}^{(m)}, P_{T|T}^{(m)}\Big), \qquad A_T^{(m,s)} = \hat{A}^{(m)} + \mathrm{mat}^{(m)}\Big(U^{(m)} \tau x_T^{(m,s)}\Big). \tag{44}$$

The corresponding sampled adapter weight is $W_T^{(m,s)} = W_0^{(m)} + B^{(m)} A_T^{(m,s)}$, and Bayesian prediction averages the predictive probabilities over Monte Carlo samples as in Section 4.4. We use the terminal filtered posterior for the main predictions because it summarizes the posterior evidence accumulated from all slices. RTS smoothing can recover the entire latent trajectory, but it is not needed for the final adapted adapter used in our experiments.

A.5 Process Covariance Specification

Seq-LoRA uses a random-walk prior to couple the slice-dependent latent states. The process covariance controls the slack for coupling local posterior evidence; $Q = 0$ gives the pooled-evidence limit. In the main experiments, the process covariance is time-homogeneous within each module and isotropic in the selected latent subspace:

$$Q_t^{(m)} = s_q q_m I_{L_m}, \qquad t = 2, \dots, T, \tag{45}$$

where $s_q > 0$ is a global process-noise scale and $q_m > 0$ is a module-specific relative scale. The first process covariance $Q_1^{(m)}$ is not added to the initial prior; the Kalman filter starts from $x_1^{(m)} \sim \mathcal{N}(m_1^{(m)}, P_1^{(m)})$, and process noise is applied only to transitions into later slices.

The role of q_m is to allow different LoRA modules to have different prior drift scales while keeping the process prior simple and stable. We construct q_m from the geometry of the slice-wise pseudo-observations rather than from an EM or smoothing estimate. Let

$$\mu_t^{(m)} = -\big(\widetilde{\Lambda}_t^{(x,m)}\big)^{-1} g_t^{(x,m)}$$

be the local pseudo-observation shift for module m, and recall that $\widetilde{\Lambda}_t^{(x,m)} = (R_t^{(m)})^\top R_t^{(m)}$. We define the average coordinate-wise projected curvature

$$\bar{\lambda}_\ell^{(m)} = \left[\frac{1}{T}\sum_{t=1}^{T} (R_t^{(m)})^\top R_t^{(m)}\right]_{\ell\ell}. \tag{46}$$

For each module, we measure the curvature-normalized slice-to-slice drift

$$d_m = \mathrm{median}_{t=2,\dots,T}\, \mathrm{median}_{\ell=1,\dots,L_m} \left[\bar{\lambda}_\ell^{(m)} \big(\mu_{t,\ell}^{(m)} - \mu_{t-1,\ell}^{(m)}\big)^2\right]. \tag{47}$$

This statistic is large when the slice-local quadratic optima move substantially in directions that are also well identified by the projected curvature.

We convert these raw module drift statistics into relative scales by normalizing across modules:

$$\tilde{q}_m = \left(\frac{d_m}{\mathrm{median}_{m'}\, d_{m'}}\right)^{\beta}, \qquad 0 \le \beta \le 1, \tag{48}$$

and then clip the result to a fixed range:

$$q_m = \mathrm{clip}\big(\tilde{q}_m, q_{\min}, q_{\max}\big). \tag{49}$$

The exponent β shrinks the module-specific scales toward a shared constant prior: $\beta = 0$ gives $q_m = 1$ for all modules, while $\beta = 1$ uses the full curvature-normalized relative drift. The clipping bounds prevent a small number of unstable modules from dominating the state evolution.

This construction should be interpreted as a prior specification, not as an attempt to learn the prior from the posterior trajectory. It preserves the random-walk LGSSM structure while allowing modules with larger curvature-normalized slice drift to receive proportionally larger process variance. All reported nonzero-Q runs use $s_q = 1$, $\beta = 0.05$, and $[q_{\min}, q_{\max}] = [0.5, 2.0]$ (Table 8); no continuous sweep over s_q is reported (Appendix C.7).

A.6 Posterior-Scale Correction

Seq-LoRA uses a scalar posterior-scale correction $\tau \in [0, 1]$ to control the absolute magnitude of the approximate posterior displacement during posterior prediction. It is a weight-space operation, not an output-space calibration transform. Logit temperature scaling leaves the model parameters fixed and rescales logits after the forward pass. In contrast, τ is applied before prediction by replacing each sampled adapter perturbation $\hat{\theta} + \Delta\theta^{(s)}$ with $\hat{\theta} + \tau\Delta\theta^{(s)}$. Thus, $\tau = 0$ recovers the MAP adapter, while $\tau = 1$ uses the raw Seq-LoRA posterior scale.

This distinction is important for interpretation. The slice-wise LGSSM posterior determines the covariance structure and sampled adapter perturbations; τ only controls their global sampling radius. It cannot create a new posterior family or directly reshape predictive probabilities in the way a logit

temperature can. Since $\tau \leq 1$, this correction can only shrink the raw posterior displacement scale and never amplify it.

The role of τ is to enforce an accuracy–calibration operating point. Stronger posterior sampling can improve uncertainty estimates by increasing predictive diversity, but overly large weight-space perturbations may move sampled adapters away from the MAP solution and reduce top-1 accuracy. Seq-LoRA therefore chooses τ from a fixed training-anchor predictive-KL budget. It is not selected to maximize validation NLL or ECE; instead, it limits calibration-oriented posterior averaging so that it does not destroy the task competence of the MAP adapter.

Related scale-control mechanisms also appear in approximate Bayesian inference, including posterior tempering and cold posteriors [50], generalized Bayesian learning-rate parameters [41, 51], and posterior variance or prior-precision selection in Laplace-style and training-free Bayesianization methods [13, 21, 34]. The closest analogy is TFB, which determines an acceptable posterior variance scale on an anchor set; Seq-LoRA differs in that τ only rescales the structured sequential posterior rather than constructing an isotropic posterior family.

Let $\mathcal{A} = \{\xi_i\}_{i=1}^{N_a}$ be an anchor set drawn only from the training distribution. For an input ξ_i, let $p_0(\cdot \mid \xi_i)$ denote the deterministic MAP predictive distribution over the valid answer choices. Given raw Seq-LoRA posterior displacements $\Delta\theta^{(s)}$ around the MAP adapter $\hat{\theta}$, the scale-corrected posterior predictive distribution is

$$p_\tau(y \mid \xi_i) = \frac{1}{S} \sum_{s=1}^{S} p\big(y \mid \xi_i, \hat{\theta} + \tau \Delta\theta^{(s)}\big). \tag{50}$$

Thus, $\tau = 0$ recovers the deterministic MAP adapter and $\tau = 1$ uses the raw Seq-LoRA posterior displacement scale.

We select τ by matching a fixed training-anchor predictive-KL budget, rather than by optimizing ACC, NLL, or ECE on validation or test benchmarks. Define

$$K_{\mathcal{A}}(\tau) = \frac{1}{N_a} \sum_{i=1}^{N_a} \mathrm{KL}(p_0(\cdot \mid \xi_i) \,\|\, p_\tau(\cdot \mid \xi_i)). \tag{51}$$

The reference target is κ_{tar}; in the main experiments we use $\kappa_{\mathrm{tar}} = 0.05$. We also use a narrow acceptance window $[\kappa_{\min}, \kappa_{\max}] = [0.05, 0.0525]$ as an early stopping criterion. This window is not a separate tuning objective; it only indicates that the posterior predictive drift is sufficiently close to the fixed reference budget. The ID test split and all OOD benchmarks are used only for final evaluation.

The quadratic update used for τ is motivated by the local behavior of the predictive KL around MAP. Since $p_{\tau=0}(\cdot \mid \xi) = p_0(\cdot \mid \xi)$, we have $K_{\mathcal{A}}(0) = 0$. Under differentiability of $p_\tau(\cdot \mid \xi)$ in τ, the first-order term of $\mathrm{KL}(p_0 \| p_\tau)$ vanishes at $\tau = 0$ because probabilities sum to one. Therefore,

$$K_{\mathcal{A}}(\tau) = C\tau^2 + o(\tau^2), \qquad C \geq 0, \tag{52}$$

which provides a local scale model for estimating a τ near the target KL.

In practice, we first evaluate a probe value

$$\tau_p = 0.5. \tag{53}$$

If $K_{\mathcal{A}}(\tau_p) \in [\kappa_{\min}, \kappa_{\max}]$, we accept τ_p. Otherwise, using the local approximation $K_{\mathcal{A}}(\tau) \approx C\tau^2$, we estimate

$$\hat{\tau} = \mathrm{clip}\left(\tau_p \sqrt{\frac{\kappa_{\mathrm{tar}}}{\max\{K_{\mathcal{A}}(\tau_p), \epsilon_K\}}}, 0, 1\right), \tag{54}$$

where $\epsilon_K > 0$ is a small numerical constant. We then evaluate $K_{\mathcal{A}}(\hat{\tau})$. If this value lies in the acceptance window, we select $\hat{\tau}$.

If the corrected candidate is still outside the window, we perform one additional quadratic refinement:

$$\hat{\tau}_{\mathrm{ref}} = \mathrm{clip}\left(\hat{\tau} \sqrt{\frac{\kappa_{\mathrm{tar}}}{\max\{K_{\mathcal{A}}(\hat{\tau}), \epsilon_K\}}}, 0, 1\right), \tag{55}$$

and evaluate $K_{\mathcal{A}}(\hat{\tau}_{\mathrm{ref}})$. When two evaluated candidates bracket the target, with $K_{\mathrm{low}} \le \kappa_{\mathrm{tar}} \le K_{\mathrm{high}}$, we use the secant interpolation

$$\tau_{\mathrm{interp}} = \tau_{\mathrm{low}} + \frac{\kappa_{\mathrm{tar}} - K_{\mathrm{low}}}{K_{\mathrm{high}} - K_{\mathrm{low}}}(\tau_{\mathrm{high}} - \tau_{\mathrm{low}}) \tag{56}$$

as a low-cost estimate of a scale near the fixed target. If no evaluated or interpolated candidate falls near the target, we choose the evaluated candidate whose training-anchor KL is closest to κ_{tar}.

Overall, τ serves as a conservative posterior-scale correction. It standardizes the predictive perturbation magnitude of the approximate posterior under a fixed training-anchor KL budget, while preserving the posterior directions and sequential structure learned by Seq-LoRA. A dense ablation over fixed and automatic posterior scales is provided in Appendix C.6.

A.7 Unordered Slices and Independent-Slice Variant

Seq-LoRA is most naturally applied when the training data admit a meaningful slice axis, such as time, curriculum level, domain, or user segment. Such semantic order is an optional, interpretable inductive bias rather than a requirement: Seq-LoRA is a structured evidence-aggregation method conditional on a chosen slice construction. An indexed slice sequence is needed only because the LGSSM transition prior is defined between adjacent latent states (Section 4.3), and the evidence Seq-LoRA aggregates is the variation in projected slice-local gradients and curvatures around the shared MAP anchor: different finite-sample slices can induce different local preferred shifts and precisions. When a dataset provides no canonical order, slices can be formed from training-only base-model NLL (Table 15), and we consider two diagnostic variants. The first keeps the Seq-LoRA LGSSM unchanged but randomly partitions the training indices into size-balanced slices (sizes differ by at most one example), without labels, topics, difficulty annotations, or evaluation data; the partition is regenerated from each run seed, and the slices are processed in a fixed, semantics-free order that we do not interpret as a hidden curriculum. This preserves the sequential posterior construction while testing whether slice-level posterior evidence remains useful without a hand-designed semantic order; it is evaluated as LGSSM random on ScienceQA (Table 3) and with OBQA as the adaptation dataset (Appendix C.3). The second removes the LGSSM transition coupling and forms an independent-slice posterior ensemble. Comparing it with the LGSSM on the same slices tests what LGSSM aggregation as a whole (the transition together with the terminal-posterior construction) adds beyond multi-slice posterior diversity, rather than isolating the transition alone. These variants are used as fallbacks and ablations; the default Seq-LoRA predictor remains the terminal LGSSM posterior.

For the independent-slice variant, we use the same slice-wise quadratic surrogates and the same shared curvature-informed subspace as Seq-LoRA. For slice t and module m, the damped projected surrogate defines

$$\widetilde{\mathcal{L}}_t^{(m)}(x) = \mathrm{const} + \big(g_t^{(x,m)}\big)^\top x + \frac{1}{2} x^\top \widetilde{\Lambda}_t^{(x,m)} x, \tag{57}$$

where $\widetilde{\Lambda}_t^{(x,m)} = \Lambda_t^{(x,m)} + \lambda I_{L_m}$. As in Seq-LoRA, define $R_t^{(m)}$ and $\widetilde{y}_t^{(m)}$ by

$$\widetilde{\Lambda}_t^{(x,m)} = (R_t^{(m)})^\top R_t^{(m)}, \qquad \widetilde{y}_t^{(m)} = R_t^{(m)} \mu_t^{(m)}.$$

The independent-slice variant conditions each slice separately on the same initial latent prior used by the LGSSM, $x^{(m)} \sim \mathcal{N}(m_1^{(m)}, P_1^{(m)})$, and the one-slice pseudo-observation likelihood $\widetilde{y}_t^{(m)} \mid x^{(m)} \sim \mathcal{N}(R_t^{(m)} x^{(m)}, I)$. Therefore,

$$q_t^{(m)}(x) = \mathcal{N}\Big(\bar{m}_t^{(m)}, \bar{P}_t^{(m)}\Big), \tag{58}$$

with

$$\bar{P}_t^{(m)} = \Big((P_1^{(m)})^{-1} + (R_t^{(m)})^\top R_t^{(m)}\Big)^{-1}, \tag{59}$$

$$\bar{m}_t^{(m)} = \bar{P}_t^{(m)} \Big((P_1^{(m)})^{-1} m_1^{(m)} + (R_t^{(m)})^\top \widetilde{y}_t^{(m)}\Big). \tag{60}$$

Unlike Seq-LoRA, this variant does not introduce a latent Markov chain over $\{x_t^{(m)}\}_{t=1}^T$. Each slice posterior is sampled independently, so there is no process covariance $Q^{(m)}$, no Kalman filtering,

and no cross-slice smoothing. This is also distinct from the $Q = 0$ tied-state boundary case: $Q = 0$ conditions one shared latent state on all slice pseudo-observations, whereas the independent variant forms separate local posteriors and ensembles their predictions. Thus, $Q = 0$ pools evidence into one state, whereas the independent variant ensembles separate slice posteriors without smoothing.

At prediction time, we draw S_t samples from each slice posterior, $x_t^{(m,s)} \sim q_t^{(m)}$, and apply the posterior-scale correction τ, selected for this variant by the same training-anchor KL rule as in Seq-LoRA (Appendix A.6):

$$A_t^{(m,s)} = \hat{A}^{(m)} + \mathrm{mat}^{(m)}\Big(U^{(m)} \tau x_t^{(m,s)}\Big). \tag{61}$$

The predictive distribution averages over both slices and Monte Carlo samples:

$$p(y \mid \xi) \approx \frac{1}{T S_t} \sum_{t=1}^{T} \sum_{s=1}^{S_t} p\Big(y \mid \xi, \{W_t^{(m,s)}\}_{m=1}^{M}\Big), \tag{62}$$

where

$$W_t^{(m,s)} = W_0^{(m)} + B^{(m)} A_t^{(m,s)}.$$

This construction preserves the same slice-level posterior evidence and subspace projection as Seq-LoRA while removing the sequential prior. Therefore, under the same slice construction, the comparison between Seq-LoRA and the independent-slice variant indicates the contribution of LGSSM coupling and terminal-posterior aggregation beyond multi-slice posterior decomposition alone.

A.8 Algorithm

Algorithm 1 Seq-LoRA

Require: Ordered data slices $\{\mathcal{D}_t\}_{t=1}^{T}$, MAP LoRA anchor $\hat{a}$, subspace dimensions $\{L_m\}_{m=1}^{M}$
Require: Hyperparameters: damping $\lambda > 0$, process-noise scale $s_q > 0$, MC samples S

// Phase 1: Curvature extraction and shared subspace construction
Compute slice-wise gradients g_t and module-wise K-FAC factors $\{H_t^{(m)}, G_{B,t}^{(m)}\}$ for all t
for each module $m = 1$ to M **do**
  Compute averaged factors $\bar{H}^{(m)} = \frac{1}{T}\sum_t H_t^{(m)}$ and $\bar{G}_B^{(m)} = \frac{1}{T}\sum_t G_{B,t}^{(m)}$
  Extract top-L_m Kronecker eigendirections from the averaged factors to build projector $U^{(m)}$
// Phase 2: Pseudo-observations and sequential latent inference
for each module $m = 1$ to M **do**
  for each slice $t = 1$ to T **do**
    Project the slice surrogate to obtain $g_t^{(x,m)}$ and $\Lambda_t^{(x,m)}$
    Set $\widetilde{\Lambda}_t^{(x,m)} \leftarrow \Lambda_t^{(x,m)} + \lambda I_{L_m}$
    Compute $\mu_t^{(m)} \leftarrow -(\widetilde{\Lambda}_t^{(x,m)})^{-1} g_t^{(x,m)}$
    Factorize $\widetilde{\Lambda}_t^{(x,m)} = (R_t^{(m)})^\top R_t^{(m)}$
    Form pseudo-observation $\widetilde{y}_t^{(m)} \leftarrow R_t^{(m)} \mu_t^{(m)}$
  Specify module process covariance $Q^{(m)} = s_q q_m I_{L_m}$
  Run Kalman filtering with observations $\{(R_t^{(m)}, \widetilde{y}_t^{(m)})\}_{t=1}^{T}$ to obtain $\mathcal{N}(m_{T|T}^{(m)}, P_{T|T}^{(m)})$
// Phase 3: Posterior-scale correction and Bayesian prediction
Select posterior-scale correction $\tau \in [0, 1]$ on a training anchor using Appendix A.6
for sample $s = 1$ to S **do**
  for each module $m = 1$ to M **do**
    Sample $x_T^{(m,s)} \sim \mathcal{N}(m_{T|T}^{(m)}, P_{T|T}^{(m)})$
    Set $A_T^{(m,s)} \leftarrow \hat{A}^{(m)} + \mathrm{mat}^{(m)}\big(U^{(m)} \tau x_T^{(m,s)}\big)$
    Restore adapter weight $W_T^{(m,s)} \leftarrow W_0^{(m)} + B^{(m)} A_T^{(m,s)}$
return $p(y \mid \xi, \mathcal{D}_{1:T}) \approx \frac{1}{S}\sum_{s=1}^{S} p\big(y \mid \xi, \{W_T^{(m,s)}\}_{m=1}^{M}\big)$

Table 4: Dataset and split statistics for the main text-only closed-choice science QA benchmark suite. OOD datasets are used only for final evaluation.

Benchmark	#Train	#Eval	#Choices	Split/source
ScienceQA G2–11	5991	2063	2–4	train / test
ScienceQA Grade12	–	310	2–4	all Grade12 splits
OBQA	–	500	4	test
ARC-C	–	1169	3–4	test, filtered
MMLU-H	–	664	4	test subjects
MMLU-C	–	346	4	test subjects
GPQA	–	448	4	`gpqa_main`

Notes. ScienceQA G2–11 uses the ScienceQA train split for MAP fine-tuning and the ScienceQA test split for ID evaluation. ScienceQA Grade12 uses all available ScienceQA splits filtered to grade 12. MMLU-H concatenates high-school physics, chemistry, and biology; MMLU-C concatenates college physics, chemistry, and biology. ARC-C has 1172 raw test examples; three five-choice examples are excluded by the four-label evaluation protocol. For ScienceQA G2–11 test, the 2/3/4-choice counts are 1409/428/226; for Grade12 they are 260/29/21; for ARC-C they are 4/1165 for 3/4 choices after filtering.

Table 5: Existing models and datasets used by the experiments.

Asset	Version/source used	License or terms
Qwen3-8B-Base [44]	Hugging Face model `Qwen/Qwen3-8B-Base`; main backbone.	Apache-2.0.
Llama 2-7B [52]	Hugging Face model `meta-llama/Llama-2-7b-hf`; additional backbone where reported.	Llama 2 Community License and provider access terms.
ScienceQA [45]	ScienceQA text-only closed-choice subset, accessed through `tcallens/scienceqa-text-only`; grades 2–11 for training/ID evaluation and grade 12 for OOD evaluation.	Dataset: CC BY-NC-SA 4.0; code: MIT.
OpenBookQA [46]	`allenai/openbookqa`, `main` configuration, test split.	Upstream AllenAI repository is Apache-2.0; the Hugging Face dataset card lists the license as unknown. We do not redistribute the raw data.
AI2 ARC [47]	`allenai/ai2_arc`, ARC-Challenge and ARC-Easy configurations, test split.	CC BY-SA 4.0.
MMLU [48]	`cais/mmlu`, high-school and college physics, chemistry, and biology test subjects.	MIT License.
GPQA [49]	`Idavidrein/gpqa`, `gpqa_main`.	CC BY 4.0, with provider terms requesting that examples not be revealed publicly to reduce benchmark leakage.

B Experimental Details

B.1 Datasets and Splits

Table 4 reports the datasets and splits used in the main text-only closed-choice science QA benchmark suite. Counts are after the preprocessing and filtering used by our evaluation protocol.

B.2 Existing Assets, Versions, and Licenses

Table 5 lists the external model and dataset assets used in the experiments. We use these assets for training or evaluation only and do not redistribute the raw model weights or benchmark data in the paper or in the released code. Users should obtain gated assets, such as Llama 2, directly from the original providers and comply with the corresponding use restrictions.

Table 6: Shared MAP fine-tuning configuration.

Item	Value
Backbone	Qwen/Qwen3-8B-Base
Training dataset	`tcallens/scienceqa-text-only`
Training subset	ScienceQA, closed-choice only, grades 2–11
Training exposure order	grades 2→11 in ascending order, examples shuffled within each grade (alternatives in Table 12)
Tokenizer padding side	left
Maximum sequence length	300
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	0.01
Warmup ratio	0.06
Maximum training steps	2,000
Checkpoint save interval	every 1,000 steps
Monitoring interval	every 100 steps; not used for checkpoint selection
Micro batch size	4
Gradient accumulation	2
Effective train batch size	8
Train-time evaluation batch size	32
Gradient clipping	1.0
Gradient checkpointing	disabled
LoRA target modules	`q_proj`, `v_proj`, `lm_head`
LoRA rank r	8
LoRA α	16
LoRA dropout	0.05
LoRA bias	none
Task type	causal language modeling with candidate-answer restricted scoring

B.3 Shared Fine-Tuning Configuration

Table 6 summarizes the shared MAP fine-tuning setup used to produce the deterministic checkpoint on top of which Laplace-LoRA, TFB, and Seq-LoRA are applied. This checkpoint also defines the common LoRA architecture used across the post-hoc Bayesian baselines.

B.4 Candidate-Answer Restricted Evaluation

This appendix gives the formal definition of the restricted answer-space evaluation used in Section 4.5. For an example with question text q and valid choices $\{c_j\}_{j=1}^{K}$, we construct a prompt with labels $\texttt{A}, \texttt{B}, \ldots$ up to the number of valid choices. We extract the logits at the final non-padding position and normalize only over the valid answer-label tokens:

$$p(y = j \mid \xi) = \frac{\exp(\ell_j(\xi))}{\sum_{k \in \mathcal{C}(\xi)} \exp(\ell_k(\xi))}, \qquad j \in \mathcal{C}(\xi), \tag{63}$$

where $\mathcal{C}(\xi)$ is the valid answer-label set for example ξ. Invalid answer labels are masked for variable-choice examples. NLL and ACC are computed on this restricted answer space. ECE is computed using $B = 10$ equal-width confidence bins:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{n} \left| \mathrm{acc}(B_b) - \mathrm{conf}(B_b) \right|, \tag{64}$$

where confidence is the maximum valid-choice probability. For posterior-predictive methods, we first average predictive probabilities over Monte Carlo samples and then compute ACC, NLL, and ECE. Table 7 shows the prompt template used for all candidate-answer evaluations.

B.5 Seq-LoRA Configuration

Table 8 reports the key Seq-LoRA hyperparameters used during Bayesian post-processing and posterior prediction in the main benchmark. The posterior-scale correction is determined using only

Table 7: Prompt template used for candidate-answer restricted evaluation.

Component	Template
Instruction	`Select one of the choices that answers the following question:`
Question	`{question}`
Choices	`Choices: A. {choice_A}. B. {choice_B}....`
Answer prefix	`Answer:`

Table 8: Seq-LoRA posterior construction and prediction configuration.

Item	Value
K-FAC batch size	4
K-FAC rank budget `n_kfac`	8
Posterior statistics dtype	`float32`
Subspace dimension per module L_m	64
Damping λ	10^{-4}
μ-observation batches per slice	32
μ-observation scale	2.0
Initial state covariance P_1	$1.0I$
Process covariance mode	`module_constant`, $Q^{(m)} = s_q q_m I$
Global process scale s_q	1.0
Module Q clip range	$q_m \in [0.5, 2.0]$
Module Q shrink exponent β	0.05
Posterior MC samples	32
Posterior-scale mode	train-anchor KL scale correction
Posterior-scale range	$\tau \in [0, 1]$
Train-anchor size for τ	500 examples
τ anchor MC samples	32
KL reference target κ_{tar}	0.05
KL acceptance window	$[0.05, 0.0525]$
Selected τ range, $T = 10$	0.58–0.67 over five seeds

a training anchor set; ID and OOD evaluation splits are not used for selecting the scale. Curvature factors are extracted with K-FAC block factorization of the classification GGN restricted to LoRA-A; they are not empirical-Fisher gradient outer products.

B.6 Baselines and Reporting

Table 9 summarizes the method-specific evaluation configuration. For post-hoc Bayesian methods, all curvature, anchor, and scale-fitting data are drawn only from the ScienceQA G2–11 training split. In the ScienceQA main experiments, BLoB, C-LoRA, and the Deep Ensemble members (MAP adapters) are trained with the same grade 2→11 exposure order as the shared MAP adapter (Table 6), with examples shuffled within each grade according to the run seed; per-example orders need not coincide across methods. MC-Dropout reuses the MAP adapter, with dropout kept active at evaluation.

B.7 Reproducibility Notes

The implementation uses left-padded tokenization and evaluates answer-choice probabilities at the final non-padding token position. For ScienceQA, the default Seq-LoRA slices are determined by the natural grade ordering from grades 2 through 11. Seq-LoRA accumulates K-FAC statistics only on the LoRA-A modules, constructs module-wise Kronecker eigenspaces, and performs module-wise Kalman inference in the resulting latent subspaces. Posterior-statistics caches store the K-FAC, subspace, and pseudo-observation statistics; changing the posterior scale, Monte Carlo sample count, or evaluation task does not require recomputing these cached statistics. The code, including launch scripts that fix the seeds and settings of the reported runs, is available at `https://github.com/toori999/seq-lora`.

Table 9: Method-specific evaluation protocol in the main benchmark.

Method	Stage	Checkpoint	Bayesian-stage data	MC	Batch	Notes
Base	none	frozen base	none	1	48	No adaptation.
MAP	none	MAP adapter	none	1	48	Deterministic LoRA.
MC-Drop	dropout inference	MAP-style	none beyond MAP	32	48	Dropout on at eval.
Deep Ensemble	ensemble	25 MAP adapters	none beyond MAP	5	48	5 ensembles of 5.
Laplace-LoRA	post-hoc Laplace	shared MAP	train split	1000	16	Curvature posterior.
BLoB	Bayesian training	method-specific	train split	32	48	Official settings.
TFB	post-hoc Bayes	shared MAP	train anchor	32	48	Official settings.
C-LoRA	Bayesian training	method-specific	train split	32	48	Official settings.
Seq-LoRA	post-hoc LGSSM	shared MAP	train split	32	48	K-FAC + LGSSM.

Table 10: Training and Bayesianization cost for Qwen3-8B-Base LoRA on a single NVIDIA RTX PRO 6000 GPU. MAP reports standard deterministic LoRA fine-tuning. For post-hoc methods (Laplace-LoRA, TFB, and Seq-LoRA), time denotes the additional fitting or Bayesianization stage after the shared MAP checkpoint has been trained. For BLoB and C-LoRA, time denotes end-to-end Bayesian adapter training. Batch denotes the batch size used for the measured stage. Time is reported in seconds, and peak allocated/reserved GPU memory is reported in GB.

Method	Stage measured	Batch	Time (s) ↓	Peak Alloc. (GB) ↓	Peak Reserv. (GB) ↓
MAP	LoRA training	8	933.76	19.50	20.10
Laplace-LoRA	post-hoc fit	32	693.16	60.06	94.28
BLoB	Bayesian training	8	1923.14	28.85	44.45
TFB	post-hoc Bayesianization	48	138.22	14.60	16.01
C-LoRA	Bayesian training	8	2444.77	23.98	30.51
Seq-LoRA	post-hoc fit	4	1014.61	23.77	26.13

C Additional Tables

C.1 Efficiency and Practicality

All efficiency measurements are taken on a single NVIDIA RTX PRO 6000 GPU with 96GB memory, using the same logging and measurement protocol as in the main experiments. Table 10 separates standard MAP LoRA training from the additional cost required to obtain a Bayesian adapter. TFB has the smallest Bayesianization cost because it is training-free and does not require curvature fitting or Bayesian retraining. Among methods that require an additional fitting or training stage, Seq-LoRA is practical: its post-hoc posterior construction costs an amount comparable to one MAP LoRA training run, while avoiding the end-to-end Bayesian retraining required by BLoB and C-LoRA. Seq-LoRA is also substantially more memory-efficient than Laplace-LoRA, reducing peak reserved memory from 94.28GB to 26.13GB.

Table 11 reports inference cost under the same hardware setting. Seq-LoRA keeps peak memory in the same range as other low-rank posterior-sampling methods and far below Laplace-LoRA. In wall-clock time, Seq-LoRA is not uniformly fastest but remains practical: it is among the faster methods on smaller-shift benchmarks and is faster on average than TFB and C-LoRA, while being modestly slower than BLoB. Overall, Seq-LoRA provides a practical post-hoc route to Bayesian prediction: it requires a moderate one-time fit after MAP training, but avoids full Bayesian retraining and the high memory cost of Laplace-LoRA.

C.2 MAP training-order ablation.

Table 12 isolates the effect of exposure order on the deterministic MAP adapter before any Bayesian posterior construction. We compare MAP adapters trained with the natural grade order, the reverse grade order, and a random exposure order, while keeping the backbone, LoRA configuration, training data, and evaluation protocol fixed. The results show that training order changes the MAP solution, but the effect is task-dependent: reverse or random exposure can slightly improve ID accuracy, whereas the natural grade order is generally stronger for OOD likelihood and calibration. Thus, Seq-LoRA's

Table 11: Inference-time cost for posterior-predictive evaluation with Qwen3-8B-Base on a single NVIDIA RTX PRO 6000 GPU. We report average wall-clock time and peak GPU memory over 5 seeds on each evaluation set. Because practical inference settings differ by method, these numbers reflect deployable evaluation cost rather than a fully matched microbenchmark. In particular, Laplace is evaluated with MC=1000, chunk size=128, and batch size=16 due to its substantially larger memory footprint, while the remaining sampling-based methods use MC=32 and batch size=48. Lower is better in all columns; the best value in each task-wise metric column is shown in bold.

	Laplace			BLoB			TFB			C-LoRA			Seq-LoRA		
Task	**Time**	**Alloc.**	**Res.**	**Time**	**Alloc.**	**Res.**	**Time**	**Alloc.**	**Res.**	**Time**	**Alloc.**	**Res.**	**Time**	**Alloc.**	**Res.**
ID	445.20	48.16	94.13	319.85	14.99	17.87	357.00	**14.95**	17.80	390.83	15.23	**17.20**	**316.85**	15.51	22.28
Grade12	70.32	41.80	82.50	**54.44**	14.67	**15.90**	71.97	**14.63**	17.80	74.06	14.80	15.91	65.82	15.20	22.28
OBQA	110.19	37.37	82.50	77.45	14.51	**15.48**	76.38	**14.48**	15.84	88.30	14.58	15.51	**68.81**	15.04	16.94
ARC-C	257.48	50.73	94.25	**198.04**	14.82	17.82	261.27	**14.78**	**17.76**	267.98	14.99	17.90	233.40	15.35	21.07
MMLU-High	148.54	52.99	94.16	**144.13**	15.25	19.15	194.98	**15.22**	19.08	190.65	15.55	**19.06**	175.14	15.79	21.07
MMLU-College	77.89	49.62	94.22	**74.52**	15.01	17.07	100.01	**14.97**	19.08	99.40	15.24	**16.92**	90.39	15.55	20.08
GPQA	**103.64**	67.47	94.24	197.90	15.62	20.01	266.45	**15.58**	19.94	243.85	16.01	**19.92**	233.72	16.14	20.55

Evaluation settings. Laplace: MC=1000, chunk size=128, eval batch size=16 BLoB: sample prediction, MC=32, eval batch size=48 TFB-LoRA: MC=32, eval batch size=48 C-LoRA: sample prediction, MC=32, eval batch size=48 Seq-LoRA: MC=32, eval batch size=48.

Table 12: Effect of MAP training-exposure order on the deterministic adapter, full per-task breakdown. All rows evaluate MAP-only LoRA adapters trained on the same ScienceQA G2–11 training set with different exposure orders (Grade order, Reverse grade, Random order). Results are mean $\pm$ standard deviation over 5 seeds. ACC and ECE are percentages.

Metric	**Variant**	**ID**	**Grade12**	**OBQA**	**ARC-C**	**MMLU-H**	**MMLU-C**	**GPQA**
ACC	Grade order	94.11 ± 0.31	93.16 ± 0.84	88.36 ± 0.09	91.50 ± 0.50	81.33 ± 0.11	73.53 ± 0.60	36.65 ± 0.37
	Reverse grade	94.63 ± 0.22	93.74 ± 0.78	87.64 ± 0.61	91.93 ± 0.25	82.05 ± 0.44	72.48 ± 0.99	35.40 ± 0.56
	Random order	94.63 ± 0.21	93.49 ± 0.70	86.72 ± 0.48	91.33 ± 0.32	80.78 ± 0.39	72.43 ± 0.75	35.67 ± 0.93
ECE	Grade order	3.77 ± 0.32	5.53 ± 0.62	8.91 ± 0.24	6.93 ± 0.50	13.36 ± 0.34	15.97 ± 0.85	34.13 ± 0.79
	Reverse grade	3.31 ± 0.28	5.42 ± 0.50	9.42 ± 0.52	6.72 ± 0.10	13.14 ± 0.61	18.19 ± 0.66	35.28 ± 0.28
	Random order	3.40 ± 0.20	5.84 ± 0.72	10.42 ± 0.62	7.25 ± 0.30	14.02 ± 0.47	17.21 ± 1.25	35.05 ± 1.30
NLL	Grade order	0.19 ± 0.01	0.33 ± 0.01	0.56 ± 0.02	0.43 ± 0.01	0.84 ± 0.02	1.09 ± 0.03	2.01 ± 0.03
	Reverse grade	0.17 ± 0.01	0.34 ± 0.02	0.61 ± 0.04	0.44 ± 0.01	0.84 ± 0.02	1.16 ± 0.05	2.05 ± 0.03
	Random order	0.18 ± 0.01	0.35 ± 0.01	0.64 ± 0.04	0.47 ± 0.01	0.87 ± 0.03	1.13 ± 0.06	2.05 ± 0.04

gains should not be interpreted as simply coming from a universally better curriculum-trained MAP anchor.

C.3 Backbone Generalization on the OBQA-to-ARC/MMLU Transfer Benchmark

Tables 13 and 14 report OBQA$\rightarrow$ARC/MMLU transfer results on two different backbones, Qwen3-8B-Base and Llama-2-7B [52], following the canonical Bayesian LoRA transfer protocol established by Laplace-LoRA [13], BLoB [14], and TFB [21], including their training set, OOD targets, and number of seeds. Because OBQA does not provide a canonical, grade-like slice axis, both tables evaluate Seq-LoRA's semantics-free fallback (Appendix A.7), a size-balanced random partition of the OBQA training set processed in a fixed order, complementing the grade-ordered ScienceQA setup of Section 4.6. Across both backbones, Seq-LoRA improves NLL and ECE on the OOD transfer targets, including the best ECE on ARC-Easy (ARC-E) and ARC-C under Llama-2-7B. On Llama-2-7B ID OBQA, Seq-LoRA trades about 1.1 ACC points relative to MAP (76.80 vs. 77.93) for OBQA ECE comparable to BLoB and the ARC-E/ARC-C ECE gains above; this is the same calibration-oriented operating point set by τ in Section 4.6, applied here without per-backbone retuning to preserve the no-test-set-selection protocol. The overall pattern is consistent across the two backbones and across grade-ordered and semantics-free slice constructions, in line with semantic ordering being an optional inductive bias rather than a requirement (Appendix A.7).

Scope of the OBQA partition evidence. The balanced random partition is regenerated from each run seed, and the MAP fine-tuning seed also changes across runs, so the standard deviations in Tables 13 and 14 combine variation from different random partitions with MAP fine-tuning and other run-level randomness; these results are not an isolated partition- or ordering-sensitivity study. We also do not directly quantify the geometry of the OBQA slices: the process-scale construction

Table 13: Transfer results on the OBQA→ARC/MMLU benchmark with Qwen3-8B-Base. ID denotes the in-distribution OBQA evaluation split. Results are mean ± standard deviation over three runs. Best results are in **bold**; second-best are underlined. Ties after rounding are broken by the unrounded mean and then by standard deviation.

		In-Dist.	Smaller Shift		Larger Shift	
Metric	**Method**	**ID**	**ARC-E**	**ARC-C**	**MMLU-H**	**MMLU-C**
ACC ↑	MAP	91.93 ± 0.12%	96.98 ± 0.38%	92.19 ± 0.10%	81.23 ± 0.57%	72.25 ± 0.29%
	MC-Drop	91.87 ± 0.31%	97.01 ± 0.29%	92.13 ± 0.17%	81.43 ± 0.32%	72.74 ± 1.09%
	Ensemble	91.93 ± 0.50%	97.06 ± 0.12%	92.10 ± 0.13%	81.02 ± 0.15%	**73.89 ± 0.17%**
	Laplace	92.13 ± 0.42%	97.12 ± 0.19%	**92.27 ± 0.22%**	81.48 ± 0.46%	72.26 ± 0.50%
	BLoB	92.20 ± 0.20%	97.18 ± 0.13%	91.93 ± 0.05%	**82.03 ± 0.57%**	72.83 ± 0.58%
	TFB	91.93 ± 0.12%	96.94 ± 0.38%	92.22 ± 0.00%	81.33 ± 0.66%	72.64 ± 1.09%
	C-LoRA	91.33 ± 0.50%	96.83 ± 0.05%	91.79 ± 0.00%	81.28 ± 0.23%	70.71 ± 1.02%
	Seq-LoRA	**92.60 ± 0.40%**	**97.34 ± 0.19%**	92.07 ± 0.22%	81.43 ± 0.77%	71.48 ± 0.60%
ECE ↓	MAP	6.57 ± 0.25%	2.00 ± 0.28%	6.04 ± 0.10%	12.57 ± 0.40%	16.10 ± 0.28%
	MC-Drop	6.48 ± 0.26%	2.06 ± 0.38%	6.02 ± 0.11%	12.53 ± 0.06%	15.72 ± 1.08%
	Ensemble	6.07 ± 0.23%	1.65 ± 0.10%	5.71 ± 0.25%	11.86 ± 0.29%	14.70 ± 0.10%
	Laplace	2.27 ± 0.70%	1.72 ± 0.17%	2.51 ± 0.32%	5.89 ± 0.36%	8.55 ± 0.51%
	BLoB	3.54 ± 0.53%	0.74 ± 0.05%	3.08 ± 0.21%	5.13 ± 0.34%	8.67 ± 1.27%
	TFB	4.34 ± 0.44%	1.08 ± 0.43%	3.80 ± 0.04%	8.57 ± 0.35%	10.13 ± 0.84%
	C-LoRA	2.94 ± 0.18%	**0.63 ± 0.06%**	2.46 ± 0.11%	4.87 ± 0.44%	8.61 ± 0.82%
	Seq-LoRA	**1.99 ± 0.43%**	1.66 ± 0.51%	**1.60 ± 0.63%**	**4.13 ± 0.34%**	**8.26 ± 1.07%**
NLL ↓	MAP	0.37 ± 0.02	0.14 ± 0.01	0.34 ± 0.01	0.77 ± 0.01	0.96 ± 0.02
	MC-Drop	0.37 ± 0.02	0.14 ± 0.01	0.34 ± 0.01	0.77 ± 0.01	0.96 ± 0.03
	Ensemble	0.34 ± 0.01	0.13 ± 0.00	0.32 ± 0.01	0.74 ± 0.01	0.93 ± 0.01
	Laplace	0.24 ± 0.02	0.11 ± 0.01	0.23 ± 0.00	0.58 ± 0.01	0.78 ± 0.02
	BLoB	0.23 ± 0.01	**0.09 ± 0.00**	**0.21 ± 0.00**	0.54 ± 0.01	0.74 ± 0.00
	TFB	0.28 ± 0.02	0.12 ± 0.01	0.25 ± 0.01	0.60 ± 0.00	0.78 ± 0.02
	C-LoRA	0.26 ± 0.01	0.10 ± 0.00	0.22 ± 0.00	0.55 ± 0.00	0.75 ± 0.00
	Seq-LoRA	**0.23 ± 0.01**	0.10 ± 0.01	0.21 ± 0.00	**0.53 ± 0.01**	**0.73 ± 0.03**

uses a curvature-normalized measure of slice-to-slice variation (Appendix A.5), but no dedicated slice-geometry diagnostic is reported. Our claim is therefore limited to the effectiveness of the evaluated semantics-free fallback across the reported runs and two backbones, not latent semantic structure in the OBQA slices, reliability across arbitrary partitions, partition/order invariance, per-target optimality, or a characterization of when semantics-free slicing is reliable; a controlled analysis is left to future work (Appendix C.7).

C.4 Slice Evidence and Coupling: Full Ablation and Mechanistic Discussion

Table 15 reports the full per-task slice ablation, extending the macro-averaged Table 3 in the main text with the reverse, shuffled, and base-NLL ordering variants, the random-partition variant, the $Q = 0$ tied-state limit, and both independent-slice variants. The four discussion paragraphs below interpret these results as boundary cases and sanity checks of the LGSSM aggregation mechanism.

Mechanistic interpretation. The full ablation separates several boundary cases of Seq-LoRA. The $Q = 0$ row is the tied-state limit of the LGSSM. In this case, the random-walk slack is removed and all latent states are forced to be equal, $x_1 = \cdots = x_T = x$. The posterior therefore reduces to a pooled pseudo-observation posterior,

$$p(x \mid \widetilde{y}_{1:T}) \propto p(x) \prod_{t=1}^{T} p(\widetilde{y}_t \mid x).$$

This boundary case keeps the pseudo-observation construction and Kalman update interface, but removes slice-specific latent drift. Its improvement over MAP indicates that slice-wise pseudo-observations preserve useful local evidence from the original training loss. Its weaker NLL and ECE relative to nonzero-Q LGSSM variants show that simple tied-state pooling does not explain the full gain of Seq-LoRA.

Table 14: Additional Llama-2-7B results on OBQA-to-ARC/MMLU transfer. All methods are trained on OBQA and evaluated on the OBQA test set and OOD science benchmarks. Results are mean ± standard deviation over three seeds; ACC and ECE are percentages. Best and second-best results are in **bold** and underlined, respectively, with ties after rounding broken by smaller standard deviation.

		In-Dist.	OOD Transfer			
Metric	**Method**	**OBQA**	**ARC-E**	**ARC-C**	**MMLU-Chem**	**MMLU-Phy**
ACC ↑	MAP	77.93 ± 0.31	76.64 ± 0.07	64.28 ± 0.34	43.67 ± 1.15	27.12 ± 1.13
	MC-Drop	77.67 ± 0.58	**76.74 ± 0.13**	64.19 ± 0.52	**45.33 ± 1.53**	28.10 ± 1.50
	Ensemble	78.07 ± 0.31	76.73 ± 0.11	64.51 ± 0.18	44.67 ± 0.58	28.10 ± 1.50
	Laplace	**79.47 ± 0.12**	75.91 ± 1.19	**66.00 ± 0.77**	43.67 ± 1.15	27.45 ± 0.98
	BLoB	79.13 ± 0.64	76.39 ± 0.36	64.13 ± 0.52	43.06 ± 1.20	27.78 ± 3.18
	TFB	77.42 ± 0.53	74.06 ± 0.49	63.07 ± 1.31	41.67 ± 2.76	**31.60 ± 2.62**
	C-LoRA	74.60 ± 0.35	75.97 ± 0.44	63.11 ± 0.20	41.00 ± 3.00	25.82 ± 0.57
	Seq-LoRA	76.80 ± 1.20	75.08 ± 0.41	63.08 ± 0.55	44.67 ± 1.53	31.05 ± 2.04
ECE ↓	MAP	6.12 ± 0.72	6.20 ± 0.13	11.55 ± 0.27	12.50 ± 0.51	23.55 ± 1.75
	MC-Drop	6.32 ± 0.63	6.09 ± 0.11	11.64 ± 0.42	9.96 ± 0.94	22.62 ± 2.29
	Ensemble	6.21 ± 0.61	5.90 ± 0.15	11.01 ± 0.23	15.93 ± 1.29	22.23 ± 1.70
	Laplace	**3.11 ± 1.18**	5.49 ± 0.88	7.43 ± 1.38	9.55 ± 2.49	21.06 ± 1.22
	BLoB	4.01 ± 1.28	5.26 ± 0.57	10.41 ± 0.70	14.62 ± 1.77	25.70 ± 3.49
	TFB	9.81 ± 0.42	7.65 ± 0.27	4.86 ± 1.32	**8.88 ± 1.33**	**14.94 ± 2.98**
	C-LoRA	9.42 ± 1.14	5.57 ± 0.57	11.35 ± 0.42	17.27 ± 2.18	26.11 ± 0.88
	Seq-LoRA	4.07 ± 1.07	**3.60 ± 0.60**	**4.16 ± 0.69**	9.99 ± 2.15	16.37 ± 2.40
NLL ↓	MAP	0.59 ± 0.00	0.67 ± 0.00	0.95 ± 0.01	1.37 ± 0.00	1.49 ± 0.01
	MC-Drop	0.59 ± 0.00	0.67 ± 0.00	0.95 ± 0.01	1.37 ± 0.00	1.49 ± 0.01
	Ensemble	0.58 ± 0.00	0.67 ± 0.00	0.94 ± 0.00	1.36 ± 0.00	1.48 ± 0.00
	Laplace	**0.54 ± 0.01**	**0.63 ± 0.00**	**0.87 ± 0.01**	1.36 ± 0.01	1.46 ± 0.01
	BLoB	0.56 ± 0.01	0.66 ± 0.00	0.93 ± 0.00	1.37 ± 0.02	1.57 ± 0.03
	TFB	0.66 ± 0.01	0.73 ± 0.00	0.96 ± 0.01	1.36 ± 0.07	1.42 ± 0.02
	C-LoRA	0.68 ± 0.02	0.67 ± 0.00	0.98 ± 0.00	1.42 ± 0.04	1.51 ± 0.01
	Seq-LoRA	0.60 ± 0.01	0.66 ± 0.01	0.90 ± 0.01	**1.30 ± 0.01**	**1.40 ± 0.01**

The independent variants provide a second structural boundary case, not the other end of a continuous Q sweep. They remove the transition model and average posterior predictions from slice-wise posteriors, rather than forming a terminal filtered posterior. Ind grade performs well, which shows that grade-based partitioning induces useful posterior diversity even without sequential coupling. In contrast, Ind random is weaker under larger distribution shift, consistent with each random slice being closer to a small noisy view of the global training distribution, with less distinct slice-local evidence; we do not test this explanation directly. LGSSM random remains much closer to the grade-ordered LGSSM, suggesting that the LGSSM can aggregate weaker local evidence into a stable terminal posterior. Within each run, LGSSM random and Ind random share the random partition, MAP checkpoint, and slice-local surrogates, and each selects its own τ by the same training-anchor rule; the comparison therefore does not separate the transition from the terminal-posterior construction.

The order and partition variants should therefore not be interpreted as testing whether the raw data order is the true data-generating sequence. Seq-LoRA operates on projected slice-wise posterior evidence $(R_t, \widetilde{y}_t)$, not directly on the raw example sequence. Changing the partition changes the slice-local evidence itself, whereas changing the order changes the coupling prior over that evidence and how it is propagated to the terminal posterior (Appendix C.7, item (iv)). The complete table shows that reverse, shuffled, random, and base-NLL partitions all remain useful, but they are not identical: different variants induce different per-target trade-offs. This supports the view that Seq-LoRA does not depend on a single privileged curriculum order, while the chosen partition still modulates the terminal posterior geometry. Reverse-grade LGSSM directly checks the high-grade terminal-bias explanation: it still improves over MAP in NLL and ECE, but is not uniformly better than the natural grade order.

As a sanity check, the $Q = 0$ tied-state row in Table 15 closely matches the $T = 1$ aggregate posterior in Table 2. This is expected because removing random-walk slack forces all slice evidence to update

Table 15: Full slice-structure ablation. In addition to the main-text variants, LGSSM reverse reverses the natural grade order, LGSSM shuffled randomly permutes the same grade-based slices, and LGSSM base-NLL orders slices by base-model difficulty. Results are mean ± standard deviation over five seeds. ACC and ECE are percentages. The † marks the frozen base model. Best and second-best non-reference entries are in **bold** and underlined; ties use the unrounded mean and then standard deviation.

		In-Dist.	Smaller Shift			Larger Shift		
Metric	**Method**	**SciQA G2–11**	**Grade12**	**OBQA**	**ARC-C**	**MMLU-H**	**MMLU-C**	**GPQA**
ACC ↑	Base†	78.43 ± 0.00	72.90 ± 0.00	78.40 ± 0.00	88.65 ± 0.00	79.07 ± 0.00	68.50 ± 0.00	30.80 ± 0.00
	MAP	94.11 ± 0.31	93.16 ± 0.84	88.36 ± 0.09	91.50 ± 0.50	81.33 ± 0.11	73.53 ± 0.60	**36.65 ± 0.37**
	LGSSM grade	93.86 ± 0.17	92.52 ± 0.74	87.52 ± 0.52	91.50 ± 0.21	81.54 ± 0.73	72.08 ± 1.18	35.04 ± 1.17
	LGSSM reverse	93.73 ± 0.25	88.58 ± 1.56	87.72 ± 0.83	91.50 ± 0.28	81.39 ± 0.80	73.01 ± 0.83	35.00 ± 1.04
	LGSSM shuffled	93.82 ± 0.25	91.04 ± 0.53	88.04 ± 0.74	91.52 ± 0.14	**81.66 ± 0.41**	73.01 ± 0.83	34.60 ± 0.42
	LGSSM base-NLL	93.93 ± 0.34	91.74 ± 0.87	87.96 ± 0.79	91.31 ± 0.13	81.27 ± 0.71	72.77 ± 0.69	34.33 ± 0.58
	LGSSM random	93.89 ± 0.27	91.36 ± 1.76	87.92 ± 0.66	91.44 ± 0.32	81.38 ± 0.84	73.47 ± 0.80	35.22 ± 0.84
	LGSSM $Q = 0$ pooled	94.07 ± 0.27	**93.42 ± 1.15**	**88.40 ± 0.32**	91.53 ± 0.48	81.36 ± 0.29	**73.87 ± 1.05**	35.80 ± 0.72
	Ind grade	93.80 ± 0.20	91.03 ± 1.28	88.08 ± 0.64	**91.64 ± 0.32**	81.20 ± 0.84	73.24 ± 1.25	34.42 ± 0.68
	Ind random	**94.11 ± 0.22**	92.52 ± 0.62	88.24 ± 0.52	91.63 ± 0.22	81.66 ± 0.42	73.41 ± 0.98	35.09 ± 0.69
ECE ↓	Base†	3.93 ± 0.00	10.64 ± 0.00	8.31 ± 0.00	6.17 ± 0.00	3.22 ± 0.00	6.89 ± 0.00	25.13 ± 0.00
	MAP	3.77 ± 0.32	5.53 ± 0.62	8.91 ± 0.24	6.93 ± 0.50	13.36 ± 0.34	15.97 ± 0.85	34.13 ± 0.79
	LGSSM grade	1.85 ± 0.48	4.32 ± 0.64	2.27 ± 0.51	**1.47 ± 0.23**	**3.67 ± 0.56**	6.43 ± 1.56	20.73 ± 1.57
	LGSSM reverse	**1.15 ± 0.20**	3.80 ± 0.98	3.20 ± 0.87	2.26 ± 0.58	5.01 ± 1.13	8.37 ± 0.94	21.65 ± 2.29
	LGSSM shuffled	1.41 ± 0.36	4.88 ± 0.28	2.16 ± 0.60	1.80 ± 0.60	4.60 ± 0.39	7.01 ± 1.07	21.81 ± 0.81
	LGSSM base-NLL	1.57 ± 0.61	4.89 ± 0.76	**2.14 ± 0.92**	1.98 ± 0.31	3.72 ± 1.26	6.62 ± 1.32	21.74 ± 2.02
	LGSSM random	1.87 ± 0.52	5.05 ± 1.07	2.34 ± 0.27	1.71 ± 0.61	3.82 ± 0.61	**5.88 ± 0.83**	**20.31 ± 1.30**
	LGSSM $Q = 0$ pooled	2.77 ± 0.18	4.72 ± 1.00	6.71 ± 0.49	5.65 ± 0.28	10.39 ± 0.27	12.59 ± 0.74	28.99 ± 1.14
	Ind grade	1.41 ± 0.36	**3.58 ± 0.97**	2.81 ± 0.98	2.04 ± 0.48	4.69 ± 0.79	7.59 ± 0.89	21.87 ± 0.69
	Ind random	1.22 ± 0.32	4.04 ± 0.55	3.74 ± 0.42	3.91 ± 0.20	7.16 ± 0.78	9.35 ± 0.64	25.38 ± 2.07
NLL ↓	Base†	0.47 ± 0.00	0.62 ± 0.00	0.61 ± 0.00	0.35 ± 0.00	0.59 ± 0.00	0.77 ± 0.00	1.63 ± 0.00
	MAP	0.19 ± 0.01	0.33 ± 0.01	0.56 ± 0.02	0.43 ± 0.01	0.84 ± 0.02	1.09 ± 0.03	2.01 ± 0.03
	LGSSM grade	0.14 ± 0.00	**0.24 ± 0.01**	0.35 ± 0.01	**0.24 ± 0.00**	**0.53 ± 0.01**	**0.74 ± 0.02**	**1.52 ± 0.02**
	LGSSM reverse	0.15 ± 0.00	0.28 ± 0.02	0.35 ± 0.01	0.25 ± 0.01	0.56 ± 0.01	0.77 ± 0.02	1.56 ± 0.03
	LGSSM shuffled	0.15 ± 0.01	0.25 ± 0.02	0.35 ± 0.01	0.24 ± 0.01	0.55 ± 0.02	0.76 ± 0.01	1.55 ± 0.02
	LGSSM base-NLL	0.14 ± 0.00	0.25 ± 0.01	0.35 ± 0.01	0.24 ± 0.00	0.54 ± 0.01	0.77 ± 0.02	1.56 ± 0.06
	LGSSM random	0.14 ± 0.00	0.25 ± 0.01	**0.34 ± 0.01**	0.24 ± 0.00	0.54 ± 0.01	0.75 ± 0.01	1.53 ± 0.04
	LGSSM $Q = 0$ pooled	0.14 ± 0.01	0.27 ± 0.01	0.43 ± 0.01	0.33 ± 0.01	0.66 ± 0.02	0.92 ± 0.03	1.76 ± 0.04
	Ind grade	0.14 ± 0.01	0.26 ± 0.01	0.35 ± 0.01	0.24 ± 0.01	0.55 ± 0.01	0.77 ± 0.01	1.56 ± 0.01
	Ind random	**0.13 ± 0.00**	0.25 ± 0.01	0.36 ± 0.01	0.26 ± 0.01	0.58 ± 0.02	0.82 ± 0.04	1.64 ± 0.05

one shared latent state. We use this only to verify implementation consistency, not as a separate source of empirical gain.

C.5 Selective Prediction under Slice-Structure Ablations

Table 16 reports selective NLL at 80% coverage for the main slice-structure variants. For each method and target dataset, examples are ranked by posterior predictive confidence and NLL is computed on the most confident 80% of examples. The results show that the same mechanism observed in the standard NLL/ECE metrics also holds for confidence-conditional prediction: tied-state pooling improves over MAP, while nonzero-Q LGSSM variants provide the strongest selective NLL under larger distribution shifts. In particular, LGSSM random has lower selective NLL than Ind random on all six OOD targets (e.g., GPQA 1.554 vs. 1.681) and remains close to LGSSM grade, whereas Ind random is lowest only on the ID split (0.025). Because the Ind/LGSSM comparison changes both the transition structure and the terminal-posterior aggregation, and each variant selects its own τ (Appendix C.4), this supports the value of LGSSM aggregation as a whole, rather than of coupling in isolation, for high-confidence predictions under shift with the tested random partition.

Table 16: Selective prediction under the main slice-structure ablations. Selective NLL is computed on the 80% most confident examples for each target dataset; lower is better. Results are mean ± standard deviation over five seeds.

Method	**SciQA G2–11**	**Grade12**	**OBQA**	**ARC-C**	**MMLU-H**	**MMLU-C**	**GPQA**
MAP	0.035 ± 0.012	0.261 ± 0.013	0.336 ± 0.045	0.207 ± 0.017	0.675 ± 0.035	0.921 ± 0.037	2.071 ± 0.048
LGSSM grade	0.033 ± 0.005	**0.166 ± 0.016**	0.195 ± 0.014	0.099 ± 0.005	**0.397 ± 0.009**	**0.548 ± 0.015**	**1.536 ± 0.017**
LGSSM random	0.034 ± 0.004	0.167 ± 0.013	**0.183 ± 0.016**	0.099 ± 0.007	0.398 ± 0.012	0.555 ± 0.009	1.554 ± 0.044
LGSSM $Q = 0$ pooled	0.028 ± 0.008	0.236 ± 0.011	0.247 ± 0.017	0.149 ± 0.013	0.532 ± 0.022	0.751 ± 0.041	1.844 ± 0.043
Ind grade	0.033 ± 0.006	0.170 ± 0.012	0.192 ± 0.018	**0.095 ± 0.008**	0.418 ± 0.019	0.581 ± 0.020	1.592 ± 0.008
Ind random	**0.025 ± 0.003**	0.179 ± 0.010	0.197 ± 0.016	0.106 ± 0.005	0.446 ± 0.026	0.627 ± 0.041	1.681 ± 0.054



Figure 2: Dense T sweep comparing MAP, fixed posterior scales, and automatic posterior-scale correction on ScienceQA. All runs use the same Qwen3-8B-Base MAP checkpoint, LoRA architecture, candidate-answer evaluation protocol, and 32 posterior samples. We exclude $T = 1$ because it is the pooled aggregate posterior rather than the multi-slice regime.

C.6 Posterior-Scale Ablation: τ as Posterior-Scale Control

Appendix A.6 defines τ and its train-anchor KL selection rule. Here we provide diagnostic evidence for the interpretation used in the main text. The purpose of τ is not to optimize a test metric, but to choose an operating scale for posterior averaging: too large a scale can move sampled adapters far enough from the MAP solution to hurt ID task accuracy, while too small a scale can underuse the posterior and weaken OOD likelihood and calibration gains.

Taken together, Figures 2, 3, and 4 support this operating-point interpretation. In the dense T sweep, unshrunk sampling ($\tau = 1$) becomes unsafe as the number of slices grows: ID ACC drops and ID NLL/ECE worsen because the raw posterior drift becomes too large. Reducing the scale, for example to $\tau = 0.5$, pulls predictions back toward the MAP adapter and recovers ID stability while still preserving OOD NLL/ECE improvements over MAP. However, $\tau = 0.5$ is only a conservative fixed choice. At $T = 10$, the fixed-τ sweep shows that smaller scales better preserve ID behavior, whereas larger scales usually give stronger OOD posterior averaging. Thus, using a fixed small value would also leave calibration gains unused on shifted tasks. Automatic τ addresses this by controlling the training-anchor predictive KL: it keeps the functional perturbation within a fixed budget instead of choosing the best point on any evaluation curve. The fact that OOD metrics and task-level KL still vary across T and target task under automatic scaling further indicates that τ does not erase posterior geometry; it controls only the perturbation magnitude. Importantly, fixed nonzero scales already improve OOD NLL/ECE over MAP, so the benefit is not introduced by the automatic scale rule alone. The automatic τ instead chooses a reproducible compromise scale: it is not per-task optimal, but it avoids both the near-MAP regime of overly small perturbations and the over-perturbed regime that harms ID task competence.

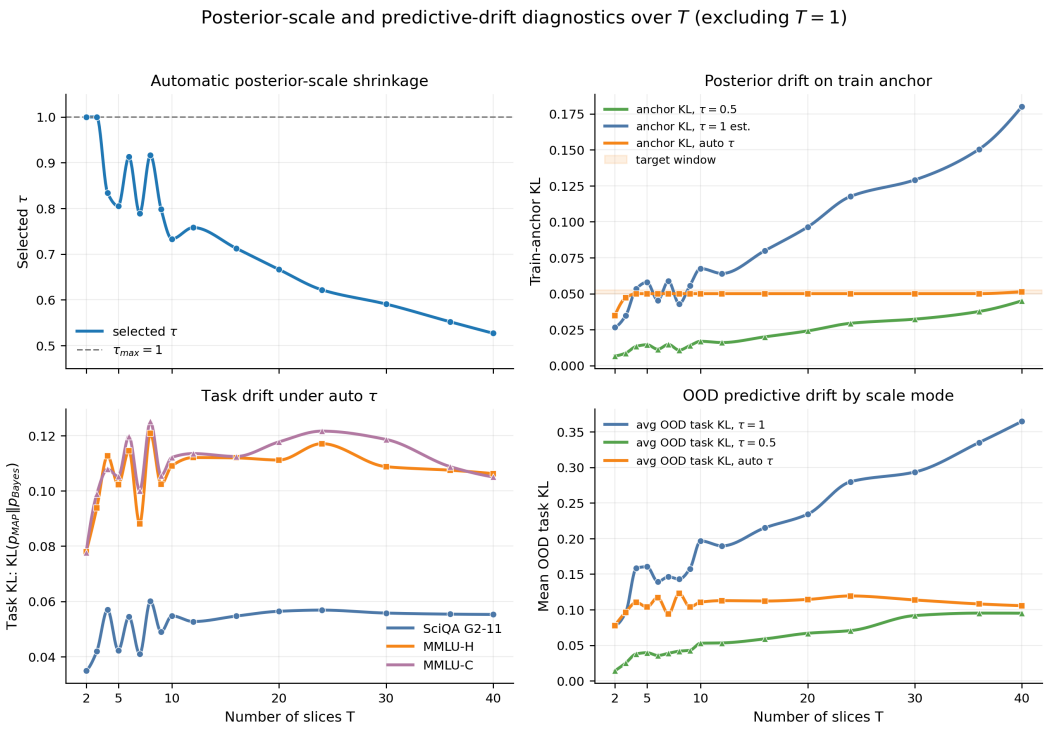

Figure 3: Posterior-drift diagnostics for the same dense T sweep. Panels report the selected posterior scale, training-anchor KL, and task-level predictive KL as functions of slice granularity.

C.7 Limitations and Future Work

This appendix expands on the limitations summarized in Section 5. Seq-LoRA currently relies on a simple transition model, a low-rank curvature subspace, a one-shot offline fitting protocol, and a chosen slice construction.

(i) Minimal-smoothness random-walk transition. The transition $x_t = x_{t-1} + \eta_t$ imposes only a unit relative-drift smoothness prior between adjacent slices, so abrupt regime shifts, recurring regimes, or multi-scale dynamics along the slice axis are not explicitly modeled.

(ii) Subspace-degenerate posterior. The induced LoRA-A posterior covariance $UP_{T|T}U^\top$ is rank-L on a d-dimensional parameter space with $L \ll d$; posterior uncertainty is exactly zero in directions orthogonal to U, and U is selected from *average* K-FAC eigendirections, so once it is fixed, evidence falling outside it cannot be recovered.

(iii) One-shot offline fitting. The slice axis and granularity T must be chosen before posterior fitting. Section 4.7.1 reports, under automatic τ, an empirical saturation regime around $T \approx 10$–20 for ScienceQA, but this regime is established against a static distribution and a fixed semantic axis. The resulting terminal posterior does not update when new data, tasks, or distributions arrive.

(iv) Dependence on the slice construction. Seq-LoRA is conditional on the chosen partition and ordering by construction: the partition determines the slice-wise gradients, curvatures, preferred shifts, and precisions, while the ordering determines latent-state adjacency and, through the adjacent-slice drift statistic d_m (Appendix A.5), can affect the module-relative process scales. Grade, reverse-grade, shuffled-grade, base-NLL, and size-balanced random constructions all retain shifted NLL/ECE gains over MAP, with construction-dependent per-target trade-offs (Table 15). However, the random partitions and shuffled grade orders in that table are regenerated from each run seed, so their standard deviations mix construction randomness with MAP fine-tuning seeds and other run-level randomness (the grade-based and base-NLL partitions are fixed across runs); the same holds for the OBQA fallback (Appendix C.3). These results are therefore not an isolated fixed-MAP partition- or ordering-

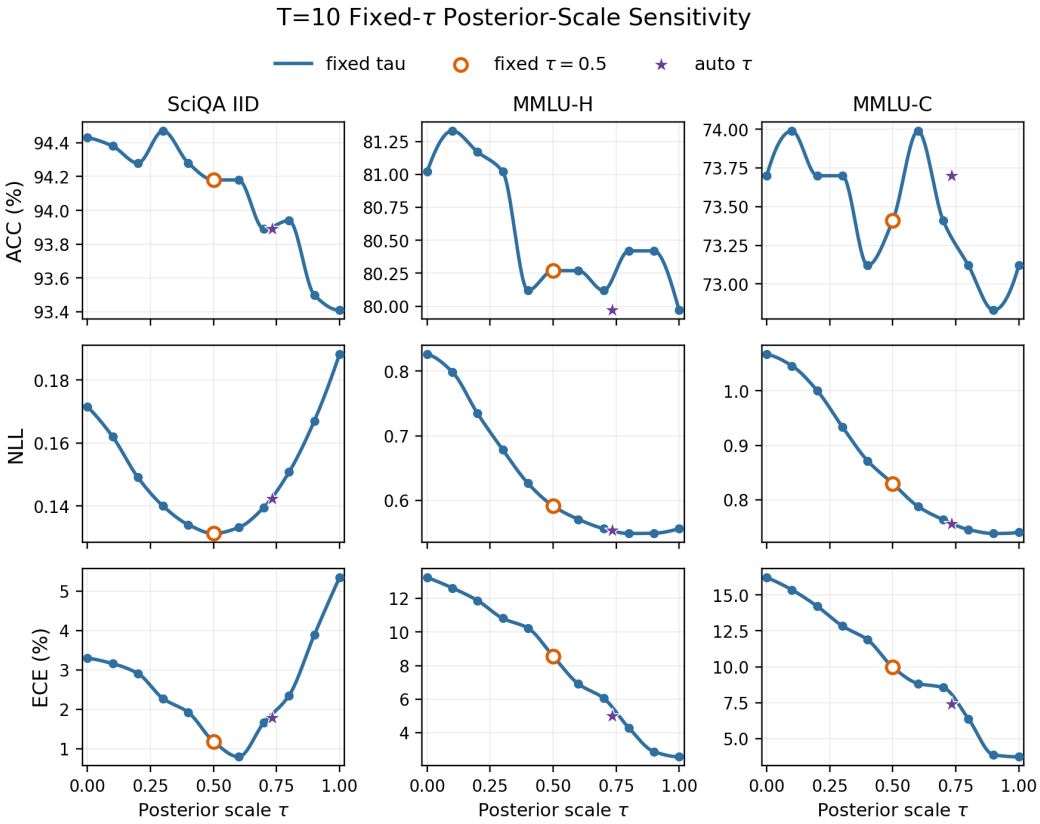

Figure 4: Fixed-τ sensitivity at the default $T = 10$ ScienceQA grade-slice partition. Blue curves sweep fixed posterior scales, orange circles mark the fixed $\tau = 0.5$ operating point, and purple stars denote the actual automatically selected τ run using only the training-anchor KL rule.

sensitivity study, and we claim effectiveness across the tested constructions, not partition/order invariance or per-target optimality.

(v) Process-scale coverage. Apart from the $Q = 0$ boundary case, the global process scale is fixed at $s_q = 1$ and is not swept continuously; the $Q = 0$ versus nonzero-Q comparison (Table 3) is a structural boundary comparison, not a sensitivity analysis over s_q. Sensitivity to the posterior scale τ is analyzed separately in Appendix C.6.

Future work: controlled partition analysis. A controlled analysis that fixes the MAP adapter and varies only the partition and the ordering, in particular for datasets without natural metadata such as OBQA, together with slice-geometry diagnostics of the projected slice-local evidence, would separate partition sensitivity from other run-level randomness and help characterize when semantics-free slicing provides reliable posterior evidence.

Future work: continual Bayesian PEFT. A natural next step is to relax the one-shot / offline assumption by treating the slice axis as an open-ended sequence and updating the LGSSM posterior online as each new slice arrives, rather than refitting curvature, subspace, and Kalman trace from scratch. This is closely related to continual learning for LLMs [53, 54] and parameter-efficient continual adaptation methods such as orthogonal LoRA, CURLoRA, and replay-enhanced LoRA variants [55–57]; the state-space posterior structure used by Seq-LoRA already provides a natural scaffold for online posterior tracking, but the concrete design of the streaming update, the choice of richer transition dynamics, and a forgetting-aware extension are left to future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the method, scope, and empirical claims; Section 5 summarizes the findings and points to Appendix C.7 for limitations and future directions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The Limitations paragraph of Section 5 and Appendix C.7 together discuss the random-walk dynamics assumption, the rank-L subspace posterior, one-shot offline fitting and slice granularity, dependence on the slice construction (no isolated partition- or ordering-sensitivity study), the fixed process scale without a continuous s_q sweep, and the closed-choice, predominantly science-oriented evaluation scope.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Section 4 states the modeling assumptions, and Appendix A provides the deferred derivations, pseudo-observation construction, process-covariance estimator, and algorithm.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The method, evaluation protocol, metrics, baselines, and dataset splits are specified in Section 4.5 and Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The experiments use public benchmarks and publicly available (in some cases gated) backbones, listed with their access terms in Appendix B.2. Code is publicly available at `https://github.com/toori999/seq-lora` under the MIT license, with installation and data-access instructions and launch scripts that fix the seeds and settings of the reported runs; it does not redistribute model weights or benchmark data. The MAP training script for the Qwen3-8B-Base OBQA transfer setting is not included; those adapters follow the same MAP recipe as ScienceQA (Appendix B.3).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Appendix B.3, Appendix B.5, and Appendix B.6 report training, posterior-construction, baseline, and evaluation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes].

Justification: The main tables report mean and standard deviation over five seeds, and the captions define the reported uncertainty.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix C.1 and the remaining tables of Appendix C report GPU type, memory, runtime, batch sizes, and posterior-predictive inference settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The authors have read the NeurIPS Code of Ethics and are not aware of any special circumstances requiring a deviation from it. The work uses public benchmark datasets and existing model backbones, collects no new human-subject or crowdsourced data, does not release a new pretrained model or dataset, and documents third-party asset terms in Appendix B.2.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The Broader impact paragraph of Section 5 discusses potential benefits (better-calibrated confidence that supports deferral or abstention) and risks (over-reliance on model confidence, and inherited biases, failure modes, and misuse potential of the underlying LLM and data).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release a new pretrained language model or a high-risk generative system; it evaluates a posterior-estimation method for existing models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: Appendix B.2 lists the external model and dataset assets used in the experiments, including provider identifiers or versions, licenses or access terms, and citations to the asset creators. Neither the paper nor the released code redistributes raw benchmark data or model weights; such assets must be obtained from the original providers under their terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The only new asset is the released code (`https://github.com/toori999/seq-lora`), documented in its README (installation, data access, per-experiment launch scripts, and implementation notes) and released under the MIT license. No new dataset or pretrained model is introduced, and no raw benchmark data or model weights are redistributed.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The work does not collect new human-subject or crowdsourced data.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The work does not involve new human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: The paper describes the use of large language model backbones as experimental subjects, including Qwen3-8B-Base in the main experiments and Llama 2 in additional experiments where reported. Section 4.5 and Appendix B.2 specify their role, access points, licenses or terms, and evaluation protocol. No additional LLMs or agents are used for data generation, evaluation, hyperparameter selection, or method construction beyond ordinary writing, editing, formatting, and basic code-assistance workflows.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.