CORA: Per-Slice Coherent Orthogonal Rotation for SVD-based Low-Rank Adaptation

Pengcheng Wang
Purdue University
wang4495@purdue.edu

Ziran Liu
Shanghai Institute for Mathematics and Interdisciplinary Sciences,
Research Institute of Intelligent Complex Systems, Fudan University,
Shanghai 200433, China
zl1011@nyu.edu

Wei Wang
Futurewei Technologies
wei.wang@futurewei.com

Wei Jiang
Futurewei Technologies
wei.jiang@futurewei.com

Abstract

Parameter-Efficient Fine-Tuning (PEFT) commonly adapts pretrained weights through low-rank updates, and recent methods further exploit the singular value decomposition (SVD) of the base weight for initialization or subspace selection. However, these methods do not explicitly preserve the coupled geometry between the pretrained left and right singular bases. Motivated by recent minimum-perturbation theory, which shows that stable finetuning follows a coherent SVD rotation in which a single orthogonal Q acts on both the left singular basis U_0 and the right singular basis V_0, we prove a per-slice analogue: each row slice of W_0 can be adapted by a shared orthogonal rotation Q_i on its left basis U_i and right basis V_i together with a diagonal spectrum shift. We implement this form as **CORA** (*Coherent Orthogonal Rotation Adaptation*), which applies per-slice orthogonal rotations and a per-layer diagonal scale to the rank-r SVD truncation of W_0. CORA uses $\frac{1}{2}m(r-1)$ trainable parameters per linear layer, about $4\times$ fewer than LoRA at the same rank. CORA outperforms LoRA, DoRA, PiSSA, and MiLoRA on commonsense reasoning and code generation while using about $8\times$ fewer parameters.

1 Introduction

Large Language Models (LLMs) are commonly adapted to downstream tasks through Parameter-Efficient Fine-Tuning (PEFT), since full finetuning is expensive in compute, storage, and deployment. The dominant PEFT method, LoRA [1], freezes a pretrained weight $W_0 \in \mathbb{R}^{m\times k}$ and learns a low-rank additive update

$$\Delta W = BA, \qquad B \in \mathbb{R}^{m\times r},\ A \in \mathbb{R}^{r\times k},\ r \ll \min(m,k).$$

This makes adaptation cheap and modular but leaves open which geometric degrees of freedom the adapter should use to modify a pretrained weight.

SVD-based PEFT methods [2–5] make this choice explicit by decomposing $W_0 = U_0\Sigma_0 V_0^\top$ into three adaptation axes: the left basis U_0, the right basis V_0, and the spectrum Σ_0. They access these

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

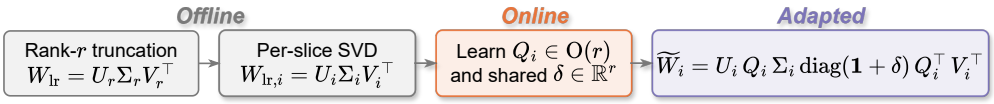

Figure 1: **Overview of CORA.** A weight matrix W is replaced by its rank-r SVD reconstruction W_{lr}, then partitioned into $s = m/r$ row slices. Per-slice SVD factors are precomputed; only the per-slice orthogonal Q_i and a per-layer diagonal scale $\boldsymbol{\delta} \in \mathbb{R}^r$ are learned, with $Q_{U,i} = Q_{V,i} = Q_i$ (Theorem 2). Per-layer trainable parameters are $\frac{1}{2}m(r-1)$, roughly $4\times$ fewer than LoRA at the same r.

axes through additive low-rank updates: PiSSA initializes A and B from the principal singular components [2], while MiLoRA updates only the minor components [3]. Orthogonal finetuning methods instead apply a learned orthogonal R as $W = RW_0$ to rotate the basis without altering the spectrum [6, 7].

Recent theoretical work [8] shows that under stability assumptions on a well-pretrained W_0, the Frobenius minimum-perturbation finetuning takes the structured form

$$\widetilde{W} = U_0 Q(\Sigma_0 + \Delta\Sigma)Q^\top V_0^\top,$$

where $Q_U = Q_V = Q$ is a coherent rotation acting on both sides of the spectrum and $\Delta\Sigma$ is a diagonal shift in singular values. Additive SVD-LoRA methods and orthogonal finetuning methods each realize a strict subset of this form: the former bundles Q and $\Delta\Sigma$ jointly inside a rank-r update $\Delta W = BA$, while the latter learns Q alone with $\Delta\Sigma = 0$.

Existing parameterizations realize the coherent rotation–spectrum form only partially. Additive LoRA-style methods can in principle change both basis and spectrum, but these effects are entangled inside the same rank-r update BA, so rotation and spectrum share a single parameter budget without a way to separate them. Orthogonal finetuning methods make rotation explicit, but strict orthogonality preserves singular values and cannot express spectral shifts without an additional spectral parameterization.

The coherent rotation form is stated for the full weight matrix, while practical adapters operate on structured low-rank matrices to keep the parameter budget small. The global form does not specify how to realize this efficiently.

We address this limitation by extending the coherent rotation to a per-slice adapter. Specifically, we partition a weight matrix into row slices $W_i \in \mathbb{R}^{r\times k}$ and show that, under mild regularity assumptions inherited from W_0, each slice has the analogous minimum-perturbation form

$$W_i^\star = U_i Q_i(\Sigma_i + \Delta\Sigma_i)Q_i^\top V_i^\top, \qquad Q_{U,i} = Q_{V,i} = Q_i,$$

where (U_i, Σ_i, V_i) is the per-slice SVD. We implement this per-slice form as a compact adapter by applying it to the rank-r reconstruction $W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$ and freezing the residual $W_0 - W_{\mathrm{lr}}$. In practice, we parameterize Q_i as a block-diagonal closed-form Cayley map of a learnable skew-symmetric matrix, which keeps Q_i exactly orthogonal throughout training (Section 4.2). The resulting method, **CORA** (Coherent Orthogonal Rotation Adaptation), realizes the coherent rotation form at $\frac{1}{2}\,m\,(r-1)$ parameters per linear layer, roughly $4\times$ fewer than LoRA at the same rank. We evaluate CORA against SVD-based and general PEFT baselines on three task families: commonsense reasoning (8 tasks), code (HumanEval), and math (GSM8K and MATH), across LLaMA-2-7B, LLaMA-3-8B, and Mistral-7B. CORA improves the parameter–accuracy trade-off on commonsense reasoning and code generation, exceeding LoRA, DoRA, PiSSA, and MiLoRA at about $8\times$ fewer parameters.

Our contributions are:

- We prove a per-slice analog of the coherent rotation theorem of [8]: every row slice of W_0 follows the same minimum-perturbation form (Theorem 2), extending it to the per-slice granularity of practical adapters.
- We parameterize each per-slice rotation as $Q_i = U_i^\top R_i U_i$ for a single learnable orthogonal R_i, enforcing the coherent rotation condition $Q_{U,i}^\star = Q_{V,i}^\star$ (Proposition 1) with no constraint or penalty

during training. CORA applies this parameterization to the rank-r reconstruction W_{lr} of W_0, using $\frac{1}{2}m(r-1)$ trainable parameters per linear layer, roughly $1/4$ of LoRA at matched rank.

- CORA reaches $\mathbf{82.16}\%$ on LLaMA-2-7B commonsense at 6.6 M parameters, exceeding PiSSA, MiLoRA, DoRA, and LoRA at $\sim 8\times$ fewer parameters; the same method transfers to LLaMA-3-8B and Mistral-7B on coding, reaching $\mathbf{48.2}$ and $\mathbf{40.2}$ HumanEval Pass@1 respectively at 10.3 M parameters, 4 to $12\times$ fewer than LoRA, DoRA, LoRI, and DiaBlo.

2 Related work

Low-rank and SVD-based adaptation. LoRA [1] learns $\Delta W = BA$ with $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times k}$ while freezing W_0; close variants augment the update with a learnable magnitude (DoRA [9]), adaptive per-layer rank (AdaLoRA [10]), or cross-layer parameter sharing (VeRA [11]). A growing subfamily uses the pretrained SVD $W_0 = U_0\Sigma_0 V_0^\top$ as the design principle: PiSSA [2] and MiLoRA [3] initialize A, B from the principal and minor singular components respectively, CorDA [12] uses an activation-weighted decomposition $W_0XX^\top$ for task-aware initialization, SVFT [4] freezes U_0, V_0 and trains a sparse coupling matrix M so that $\Delta W = U_0MV_0^\top$, KaSA [5] adds knowledge-aware gating on singular values, and LoftQ [13] combines SVD initialization with 4-bit quantization. Optimization-side variants such as LoRA-GA [14] and LoRA-Pro [15] improve initialization and gradient scaling for the same $\Delta W = BA$ parameterization. All of these methods bias the optimizer toward the pretrained spectrum but leave the additive $\Delta W = BA$ block bundling rotation and spectrum adaptation, without enforcing the coherent rotation condition $Q_U = Q_V$ established in [8] (see Section 3.2).

Orthogonal and block-diagonal finetuning. Orthogonal finetuning methods parameterize the update as a rotation of W_0: OFT [6] uses a strictly orthogonal block-diagonal R applied as $\widetilde{W} = RW_0$; BOFT [7] replaces the block structure with a butterfly factorization; OFTv2 [16] moves the rotation to the activation side for throughput; POET [17] extends to a two-sided orthogonal equivalence $\widetilde{W} = PW_0Q^\top$; and PSOFT [18] constrains the rotation to the principal subspace of W_0. These methods operate at substantially higher training cost than the SVD-initialized PEFT family and target distinct deployment regimes. We use them as conceptual context and benchmark CORA against the SVD-initialized PEFT family in Section 5.1. DiaBlo [19] replaces the low-rank product $\Delta W = BA$ with a direct block-diagonal update on W. All of these methods apply at the full-matrix scale; none exploits a per-slice decomposition. **CORA** is the per-slice specialization of this family, with $s = m/r$ independent orthogonal rotations, one per row slice of the rank-r reconstruction $W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$, unifying the additive SVD-initialized and orthogonal-rotation families under a single form (1).

3 Preliminaries

3.1 Notation

Let $W_0 \in \mathbb{R}^{m\times k}$ be a pretrained weight matrix with SVD $W_0 = U_0\Sigma_0V_0^\top$, where $U_0 \in \mathbb{R}^{m\times d}$, $V_0 \in \mathbb{R}^{k\times d}$, $\Sigma_0 = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$, $d = \min(m, k)$, and $\sigma_1 \geq \cdots \geq \sigma_d > 0$. Write $W_{\mathrm{lr}} := U_{0,r}\Sigma_{0,r}V_{0,r}^\top$ for the rank-r truncation, where $U_{0,r}$, $\Sigma_{0,r}$, $V_{0,r}$ keep the leading r singular components.

3.2 Minimum-perturbation form

We first recall the global minimum-perturbation form for SVD-aware finetuning. Under stability assumptions on a well-pretrained W_0 and the Frobenius minimum-perturbation objective, the finetuned weight admits a coherent in-basis rotation, which we restate in our notation.

Theorem 1 (Coherent rotation form, after [8, Prop. 9.4])**.** *Let $W_0 \in \mathbb{R}^{m\times k}$ be a pretrained weight matrix with SVD $W_0 = U_0\Sigma_0V_0^\top$. Under the standard stability assumptions and the Frobenius minimum-perturbation objective, the finetuned weight has the form*

$$W^\star = U_0Q(\Sigma_0 + \Delta\Sigma)Q^\top V_0^\top, \tag{1}$$

where Q is orthogonal and $\Delta\Sigma$ is diagonal. Equivalently, the left and right in-basis rotations satisfy $Q_U^\star = Q_V^\star = Q$.

Among all finetuned weights that achieve a given task objective, the one with smallest $\|\Delta W\|_F$ keeps the pretrained singular coordinate system fixed and acts on Σ_0 by a similarity transform $Q\,(\cdot)\,Q^\top$ in that basis. Any mismatch between the left and right in-basis rotations, i.e., $Q_U \neq Q_V$, breaks this coherent structure and increases the perturbation needed to realize the same adaptation.

Eq. (1) thus exposes two coupled degrees of freedom: a spectral shift $\Delta\Sigma$ and a single coherent rotation Q that acts on both sides of the spectrum. Two existing PEFT families realize this structure only partially: additive SVD-LoRA methods bundle spectrum and basis adaptation inside $\Delta W = BA$, while orthogonal finetuning methods learn Q alone with $\Delta\Sigma = 0$.

This form is stated for the full matrix: Q acts on W_0's entire singular bases. Section 4.1 extends it to the row-slice granularity that practical adapters use.

3.3 Per-slice decomposition

We work with a per-slice decomposition of the source matrix. Given $W \in \mathbb{R}^{m\times k}$, we partition its rows into $s = m/r$ contiguous slices $W_i \in \mathbb{R}^{r\times k}$, so that $W = [W_1^\top \cdots W_s^\top]^\top$, and write each slice's SVD as $W_i = U_i\Sigma_i V_i^\top$. CORA applies per-slice adaptation to the rank-r reconstruction $W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$ while freezing the residual $W_0 - W_{\mathrm{lr}}$. The rank-r truncation keeps the per-slice SVD cache compact and acts as a spectral regularizer on the source matrix.

4 Method

We first extend the coherent rotation form (Theorem 1) from a full weight matrix to row-slice submatrices (Section 4.1). We then introduce a *rotation reparameterization* that realizes the resulting per-slice form with a single learnable matrix per slice (Section 4.2), and finally instantiate this form as the CORA adapter, with and without a learnable spectrum scale (Section 4.3). See Appendix A–G for full proofs and implementation details (Cayley map, parameter accounting); the main text states results and sketches the justifications.

4.1 Per-slice coherent rotation form

Theorem 1 is stated for the full pretrained matrix W_0. We now state its row-slice generalization.

Setup. Following the per-slice decomposition of Section 3.3, let W denote either W_0 or its rank-r reconstruction $W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$, partitioned into $s = m/r$ row slices $W_i \in \mathbb{R}^{r\times k}$ with SVDs $W_i = U_i\Sigma_i V_i^\top$ ($U_i \in \mathbb{R}^{r\times r}$ square orthogonal, Σ_i diagonal, $V_i \in \mathbb{R}^{k\times r}$ column-orthonormal).

Assumption 1 (Slice-level regularity)**.** *Each slice W_i has non-degenerate spectrum ($\sigma_{\min}(W_i) > 0$), bounded condition number ($\kappa(W_i) < \infty$), and small stable perturbation ($\|\Delta W_i\|_F \ll \|W_i\|_F$).*

Assumption 1 requires each slice to inherit the regularity of the parent matrix; it holds generically because row-slicing preserves the column-space rank, and the singular-value interlacing theorem [20] ensures $\sigma_{\min}(W_i) \geq \sigma_{\min}(W) > 0$ for generic slice boundaries.

Theorem 2 (Per-slice coherent rotation form)**.** *Let $W_i = U_i\Sigma_i V_i^\top$ be a row slice satisfying Assumption 1. Under the slice-level Frobenius minimum-perturbation objective, the finetuned slice has the form*

$$W_i^\star = U_i Q_i(\Sigma_i + \Delta\Sigma_i)Q_i^\top V_i^\top, \tag{2}$$

where $Q_i \in \mathrm{O}(r)$ is orthogonal and $\Delta\Sigma_i$ is diagonal. Equivalently, the left and right per-slice in-basis rotations satisfy $Q_{U,i}^\star = Q_{V,i}^\star = Q_i$.

Proof sketch. The theorem is a slice-local statement: a single slice is analyzed under its own Frobenius minimum-perturbation objective, lifting the global argument to local SVD coordinates. Two ingredients enable this lift. (i) The Frobenius norm is row-partition additive, $\|\Delta W\|_F^2 = \sum_{i=1}^s \|\Delta W_i\|_F^2$, so the per-slice cost $\|\Delta W_i\|_F$ is the natural local objective. (ii) Each slice inherits the regularity of W via singular-value interlacing (Assumption 1), so the local problem is well-posed in (U_i, Σ_i, V_i). Within each slice, the spectrum-fixed Frobenius-distance minimization has a unitary similarity solution by classical Schur–Horn / Von Neumann arguments [21]. The full proof is in Appendix A.

Source matrix. Theorem 2 applies whenever the source matrix W is square orthogonal in its column basis. CORA takes $W = W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$ and freezes the residual $W_0 - W_{\mathrm{lr}}$. The rank-r truncation removes tail singular directions and acts as a spectral regularizer.

4.2 Rotation reparameterization

Theorem 2 specifies the desired per-slice form, but a direct implementation would require parameterizing each Q_i on the orthogonal manifold $\mathrm{O}(r)$, which involves expensive retraction or projection steps. We instead reparameterize each Q_i through an unconstrained learnable matrix R_i per slice, related by a fixed basis-conjugation in the per-slice SVD frame. This is analogous to weight-normalization-style reparameterizations [22]: the learned R_i lives in an unconstrained ambient space, while the target rotation Q_i is recovered through a fixed map.

Specifically, for each slice, let $R_i \in \mathbb{R}^{r\times r}$ be a learnable matrix (orthogonal by default; see Section 4.3), and define $Q_i := U_i^\top R_i U_i$ (a similarity transform of R_i by the per-slice left singular basis U_i). Substituting into Eq. (2) and using the identity $U_iU_i^\top = I_r$ (since U_i is square orthogonal), $U_iQ_i = (U_iU_i^\top)R_iU_i = R_iU_i$, which gives the equivalent reparameterized form:

$$\widetilde{W}_i \;=\; R_i\, U_i\, \Sigma_i\, \mathrm{diag}(\mathbf{1}+\boldsymbol{\delta})\, Q_i^\top\, V_i^\top, \qquad Q_i^\top = U_i^\top R_i^\top U_i, \tag{3}$$

where $\boldsymbol{\delta} \in \mathbb{R}^r$ is a per-layer scale vector applied to every slice i of the layer. Only R_i per slice (and optionally the shared $\boldsymbol{\delta}$ per layer) is learned; the $Q_i^\top$ factor is computed from R_i at evaluation time.

Proposition 1 (Rotation reparameterization realizes the coherent rotation form)**.** *Let $W_i = U_i\Sigma_iV_i^\top$ be a slice with $U_i \in \mathbb{R}^{r\times r}$ square orthogonal, Σ_i diagonal, and $V_i \in \mathbb{R}^{k\times r}$ column-orthonormal. Let $R_i \in \mathrm{O}(r)$ be orthogonal and $\boldsymbol{\delta} \in \mathbb{R}^r$ a scale vector. Define $Q_i = U_i^\top R_iU_i$ and form $\widetilde{W}_i$ via Eq.* (3)*. Then:*

(i) $\widetilde{W}_i = U_iQ_i\Sigma_i\,\mathrm{diag}(\mathbf{1}+\boldsymbol{\delta})\,Q_i^\top V_i^\top$; in particular Q_i is orthogonal.

(ii) The SVD of $\widetilde{W}_i$ satisfies the coherent rotation condition $Q^\star_{U,i} = Q^\star_{V,i}$.

Proof sketch. Part (i) follows by the substitution above and the identity $U_iU_i^\top = I_r$. Part (ii) follows by writing $\widetilde{W}_i = B_iC_i$ with $B_i = R_i(U_i\Sigma_i\,\mathrm{diag}(\mathbf{1}+\boldsymbol{\delta})\,U_i^\top)R_i^\top$ symmetric and $C_i = U_iV_i^\top$ row-orthonormal. The SVD of $\widetilde{W}_i$ is then determined by the eigendecomposition of B_i, which induces the same in-basis rotation on the left and right. The full proof is in Appendix B.

Remark. Proposition 1 reduces the per-slice parameterization to a single orthogonal $R_i \in \mathrm{O}(r)$ instead of parameterizing $Q_{U,i}$ and $Q_{V,i}$ separately. The coherent rotation condition $Q^\star_{U,i} = Q^\star_{V,i}$ holds identically. The shared scale $\boldsymbol{\delta} \in \mathbb{R}^r$ is an optional component, which scales the per-slice singular values entrywise by $\mathbf{1}+\boldsymbol{\delta}$. When $\boldsymbol{\delta}$ is omitted, the per-slice spectrum stays at Σ_i.

4.3 CORA adapter

Given a pretrained linear layer $y = W_0x$, CORA adapts it to $\widetilde{y} = \widetilde{W}x$ using the per-slice parameterization above. CORA has three offline-precomputable components and two learnable components.

Offline components. For each adapted layer, we precompute (i) the SVD $W_0 = U_0\Sigma_0V_0^\top$, (ii) the source matrix $W_{\mathrm{lr}} = U_{0,r}\Sigma_{0,r}V_{0,r}^\top$ obtained by truncating to the top-r singular components, and (iii) the per-slice SVD factors (U_i, Σ_i, V_i) obtained by partitioning W_{lr} into $s = m/r$ row slices.

Learnable parameters. CORA learns (i) per slice, a block-diagonal orthogonal rotation $R_i = \mathrm{blkdiag}(\phi(A_i^{(1)}), \ldots, \phi(A_i^{(r/b)}))$, where each $A_i^{(g)} \in \mathbb{R}^{b\times b}$ is skew-symmetric, b is the inner block size with $b \mid r$, and $\phi : \mathrm{skew}(b) \to \mathrm{O}(b)$ is a Cayley-type map (Appendix D); and (ii) optionally, per layer, a single shared scale vector $\boldsymbol{\delta} \in \mathbb{R}^r$ applied multiplicatively as $\Sigma_i \mapsto \Sigma_i\,\mathrm{diag}(\mathbf{1}+\boldsymbol{\delta})$ to every slice i of that layer. Tying $\boldsymbol{\delta}$ across slices reduces the spectrum-correction cost from m to r scalars per layer while preserving layer-wide singular-value structure. By default, CORA learns the per-slice rotation R_i together with the shared scale $\boldsymbol{\delta}$, a configuration we denote **CORA+$\boldsymbol{\delta}$** and use throughout our main experiments. Setting $\boldsymbol{\delta} = 0$ recovers the rotation-only configuration, which we refer to as

plain **CORA**. Both configurations satisfy the coherent rotation form by Proposition 1, and they differ only in whether the spectrum is allowed to scale.

Parameter accounting. With the default full-block setting ($b = r$), each slice's Cayley generator $A_i \in \mathbb{R}^{r\times r}$ is skew-symmetric with $r(r-1)/2$ free parameters. Summing over the m/r slices of a weight matrix $W \in \mathbb{R}^{m\times k}$,

$$|\theta^{(W)}_{\text{CORA}}| \;=\; \tfrac{1}{2}\, m\,(r-1), \tag{4}$$

independent of the column dimension k (the optional shared scale adds r scalars per layer). LoRA at the same rank r uses $|\theta^{(W)}_{\text{LoRA}}| = r(m+k)$, so for a large rank r

$$\frac{|\theta^{(W)}_{\text{CORA}}|}{|\theta^{(W)}_{\text{LoRA}}|} \;\xrightarrow{r\to\infty}\; \frac{m}{2(m+k)}, \tag{5}$$

which equals exactly $1/4$ for square projections ($m = k$, e.g., the attention query/key/value layers in LLaMA-2-7B) and remains close to $1/4$ on the weighted average across LLaMA-2-7B's full target set. Per-tier counts ($r \in \{16, 32, 64, 128\}$ giving 6.6, 13.6, 27.6, 55.7 M parameters) are reported in Table 1.

Implementation summary. At inference, the adapted weight is computed slice by slice via Eq. (3) and reassembled. We instantiate ϕ as the *closed-form Cayley map* $\phi(A) = (I - A)(I + A)^{-1}$, which keeps each block $\phi(A_i^{(g)})$ exactly orthogonal and imposes no convergence-radius constraint on $A_i^{(g)}$, so no auxiliary regularizer is needed. Implementation details and parameter counts are summarized in Appendix D and Section 5.

5 Experiments

5.1 Setup

Models, data, and metrics. We evaluate CORA across three task families. For commonsense reasoning on LLaMA-2-7B and LLaMA-3-8B, we train on commonsense_170k [23] and report arithmetic-mean accuracy over the eight standard benchmarks (BoolQ, PIQA, SIQA, HellaSwag, WinoGrande, ARC-Easy, ARC-Challenge, OpenBookQA), following the protocol used by DoRA, MiLoRA, and DiaBlo [3, 9, 19]. For mathematical reasoning on LLaMA-2-7B, we train on MetaMathQA-395K [24] and evaluate on GSM8K [25] and MATH [26] via Exact Match. For code generation on LLaMA-3-8B and Mistral-7B, we train on CodeAlpaca [27] and report HumanEval Pass@1 [28]. We adapt the attention query/key/value and FFN up/down projections on LLaMA-2-7B, extending to the full set (adding the attention output and FFN gate projections) on LLaMA-3-8B and Mistral-7B, matching DoRA [9]'s target modules.

Baselines. Our primary baselines are the SVD-based PEFT methods that publish on the corresponding benchmark: PiSSA [2] and MiLoRA [3] on commonsense and math, plus LoRA [1] and DoRA [9] as architectural reference. For code generation, because HumanEval results are not reported for the SVD-based baselines, we additionally include LoRI [29] and DiaBlo [19]. We take baseline numbers from the original publications and from our HuggingFace-Trainer reproductions, and note the source for each row in the corresponding table caption. See Appendix G for training hyperparameters.

5.2 Main results on commonsense reasoning

Table 1 reports per-task accuracy on LLaMA-2-7B and LLaMA-3-8B at four parameter tiers ($r \in \{8, 16, 32, 64\}$). CORA cells use the default variant (closed-form Cayley with shared scale $\boldsymbol{\delta}$; see Table 4).

CORA improves over the SVD-based PEFT family at every parameter tier on both models. On LLaMA-2-7B, CORA r=16 at 6.6 M parameters reaches 82.16%, 8.4 points above PiSSA r=32 (56 M) and 3.0 points above MiLoRA r=32 (56 M). On LLaMA-3-8B, CORA r=16 at 10.3 M parameters reaches 86.65%, 11.3 points above PiSSA and 4.8 points above MiLoRA, again at roughly $6\times$ fewer parameters. PiSSA and MiLoRA constrain A and B but still adapt through an

Table 1: Per-task commonsense accuracy on LLaMA-2-7B and LLaMA-3-8B (commonsense_170k, 8-task protocol). Baselines: LoRA / DoRA from [9]; PiSSA / MiLoRA from [3]; Full FT from [19]. Underlined: SVD-family. Best in **bold** (Full FT excluded).

Method	r/N	#Params	BoolQ	PIQA	SIQA	HellaS	WinoG	ARC-e	ARC-c	OBQA	AVG
LLaMA-2-7B											
Full FT	N/A	6.7 B (100%)	73.3	85.7	81.0	90.2	86.9	88.6	77.4	85.2	83.5
LoRA	r=32	56 M (0.84%)	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
DoRA	r=32	57 M (0.85%)	71.8	83.7	76.0	**89.1**	82.6	83.7	68.2	82.4	79.7
PiSSA	r=32	56 M (0.84%)	67.6	78.1	78.4	76.6	78.0	75.8	60.2	75.6	73.8
MiLoRA	r=32	56 M (0.84%)	67.6	83.8	80.1	88.2	82.0	82.8	68.8	80.6	79.2
CORA (ours)	r=64	27.6 M (0.41%)	70.8	83.9	80.2	81.3	84.5	85.8	71.1	82.0	79.95
CORA (ours)	r=32	13.6 M (0.20%)	72.2	**85.5**	81.5	87.2	**86.7**	86.4	72.8	84.2	82.06
CORA (ours)	r=16	6.6 M (0.10%)	**73.2**	84.6	**81.8**	88.6	84.9	**87.1**	**73.5**	83.6	**82.16**
CORA (ours)	r=8	3.1 M (0.05%)	69.7	84.2	81.3	88.5	85.2	**87.1**	73.1	**85.2**	81.79
LLaMA-3-8B											
Full FT	N/A	8 B (100%)	76.4	89.7	82.5	95.5	89.6	92.9	84.3	89.2	87.5
LoRA	r=32	63 M (0.79%)	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
DoRA	r=32	63 M (0.79%)	**74.6**	89.3	79.9	95.5	85.6	90.5	80.4	85.8	85.2
PiSSA	r=32	63 M (0.79%)	67.1	81.1	77.2	83.6	78.9	77.7	63.2	74.6	75.4
MiLoRA	r=32	63 M (0.79%)	68.8	86.7	77.2	92.9	85.6	86.8	75.5	81.8	81.9
CORA (ours)	r=64	43.3 M (0.54%)	72.2	88.2	82.7	96.1	88.2	92.2	83.1	87.0	86.21
CORA (ours)	r=32	21.3 M (0.27%)	72.0	89.2	**83.2**	96.2	**88.5**	**93.5**	83.2	**88.2**	**86.75**
CORA (ours)	r=16	10.3 M (0.13%)	72.9	**89.9**	82.6	**96.3**	88.2	92.6	**83.5**	87.2	86.65
CORA (ours)	r=8	4.8 M (0.06%)	72.5	89.0	81.1	95.6	87.6	93.1	82.1	87.0	86.00

additive low-rank update; this update cannot in general realize the per-slice coherent rotation that CORA derives in closed form (Theorem 2).

Against the LoRA family, CORA matches or exceeds the published numbers at $4\times$ to $9\times$ fewer parameters. CORA r=16 exceeds DoRA r=32 by 2.5 points on LLaMA-2-7B and 1.5 points on LLaMA-3-8B. At the smallest budget, CORA r=8 uses 3.1 M parameters on LLaMA-2-7B (81.79%) and 4.8 M on LLaMA-3-8B (86.00%), between 5% and 8% of LoRA r=32's budget while remaining above LoRA r=32 on both models.

5.3 Beyond commonsense: code and math

We evaluate CORA on two additional task families to test transfer beyond commonsense. For code generation we finetune LLaMA-3-8B and Mistral-7B on CodeAlpaca and report HumanEval Pass@1 in Table 2, with reference baselines from DiaBlo [19] Tab. 3 and LoRI [29] Tab. 2. For mathematical reasoning we finetune LLaMA-2-7B on MetaMathQA-395K and report GSM8K and MATH accuracy in Table 3, with reference baselines drawn from MiLoRA [3] Tab. 2 and DiaBlo [19] Tab. 2. We use the same hyperparameters as the commonsense setup (Appendix G).

On code, CORA r=16 at 10.3 M reaches 48.2 HumanEval Pass@1 on LLaMA-3-8B, 5.0 points above DiaBlo N=64 at 121 M and LoRI r=32 at 45 M. On Mistral-7B, CORA r=32 and r=64 reach 40.9, $\sim$ 6–7 points above LoRA, DoRA, and the DiaBlo variants at $\sim$ 60–120 M; CORA r=16 at 10.3 M retains 40.2, and CORA r=8 at 4.8 M retains 45.1/40.2 on the two models. The lower-rank tiers degrade slightly faster on Mistral-7B than on the two LLaMA models, an effect we tentatively attribute to its grouped-query attention layout, which couples the K/V projections across heads and likely concentrates more task-relevant signal into the small fraction of slices each R_i adapts.

On math, CORA r=64 at 27.6 M reaches 62.0 GSM8K, 1.4 points above LoRA r=64 at 113 M; on MATH, CORA r=64 reaches 12.7, below MiLoRA r=64's 17.8.

Overall, principal-component methods tend to underperform on MATH. We hypothesize that this gap reflects the *spectral locality* of mathematical knowledge in the pretrained backbone. If mathematical reasoning is underrepresented in general-purpose pretraining relative to natural-language tasks (a common assumption motivating math-specific instruction-tuning corpora such as MetaMathQA [24]), then the principal singular directions of W_0 should encode comparatively little MATH-relevant

Table 2: Code generation on LLaMA-3-8B and Mistral-7B (CodeAlpaca, HumanEval Pass@1). Baselines from DiaBlo [19] and LoRI [29]. Best in **bold**.

	LLaMA-3-8B		Mistral-7B	
Config	Params	HumanEval	Params	HumanEval
LoRA r=32 (ref)	90 M	34.7	91 M	33.8
DoRA r=32 (ref)	90 M	33.1	91 M	33.7
LoRI r=32 (ref)	45 M	43.2	46 M	33.8
DiaBlo N=128 (ref)	61 M	39.4	61 M	34.0
DiaBlo N=64 (ref)	121 M	43.2	122 M	34.4
CORA r=64 (ours)	43.3 M	47.0	43.3 M	**40.9**
CORA r=32 (ours)	21.3 M	45.1	21.3 M	**40.9**
CORA r=16 (ours)	10.3 M	**48.2**	10.3 M	40.2
CORA r=8 (ours)	4.8 M	45.1	4.8 M	40.2

Table 3: Mathematical reasoning on LLaMA-2-7B (GSM8K + MATH). Best in **bold**.

Config	Params	GSM8K	MATH
Full FT (reference)	6.74B	66.5	19.8
LoRA r=64	113 M	60.6	16.9
PiSSA r=64	113 M	58.2	15.8
MiLoRA r=64	113 M	63.5	**17.8**
CORA r=128 (W_0)	55.7 M	62.4	14.4
CORA r=64 (W_{lr})	27.6 M	62.0	12.7

structure. Adapters that operate within the top-r principal subspace (PiSSA and the default CORA that applies the rotation to W_{lr} and discards the tail singular directions on which MATH relies) inherit this limitation directly. MiLoRA, which freezes the principal components and trains the minor ones, has access to precisely the residual directions where math-specific structure is more likely to concentrate, consistent with its own framing of minor components as task-adaptive [3]. DiaBlo, which adapts entries of W_0 directly, also performs well on MATH at higher parameter budgets (20.4 at N=32, 141 M).

To test this hypothesis, in Section 5.4, we evaluate a variant of CORA using the full W_0 as the source matrix instead of W_{lr}, which retains these tail directions and is able to close part of the performance gap. This indicates that further studies can be conducted to improve MATH. For example, a controlled comparison that varies only the source spectrum within CORA's slice-wise framework (e.g., rotating the bottom-r subspace as a direct counterpart to MiLoRA) would isolate the effect, and we leave this to future work.

5.4 Ablations

Shared spectrum scale. We ablate whether the shared spectrum scale $\boldsymbol{\delta}$ is enabled (Section 4.3). Table 4 reports the LLaMA-2-7B 8-task average across the four CORA tiers. Closed-form solve Cayley is used throughout. The shared scale gains 0.8 to 2.4 points at $r \in \{16, 32, 64\}$ and rescues the r=8 configuration from collapse ($+13.8$ points). Therefore $\boldsymbol{\delta}$ is enabled as the default.

Rotation sharing across slice pairs. By default, CORA learns one orthogonal R_i per slice. We can halve the rotation parameter count by sharing one R_i across pairs of consecutive slices, the *paired* variant. The relevant comparison is at matched parameter budget rather than at matched rank: at a fixed budget, paired with rank r has the same parameter count as default with rank $r/2$, so paired effectively trades half the rotation degrees of freedom for doubled per-slice spectral capacity. Table 5 reports this matched-budget comparison on LLaMA-2-7B math. Paired sharing wins both metrics at both budgets: at ~ 27.6 M parameters, paired r=128 exceeds default r=64 by 2.3 points on GSM8K and 1.5 on MATH; at ~ 13.6 M, paired r=64 exceeds default r=32 by 2.0 on GSM8K and 0.5 on MATH. We adopt no sharing as the default in our main results; the paired variant is a drop-in upgrade when doubling the per-slice rank within a fixed budget is preferred.

Table 4: Shared spectrum scale ablation on LLaMA-2-7B, commonsense 8-task average.

CORA tier	w/o $\boldsymbol{\delta}$	w/ $\boldsymbol{\delta}$
r=64 (27.6 M)	77.57	79.95
r=32 (13.6 M)	81.23	82.06
r=16 (6.6 M)	81.38	**82.16**
r=8 (3.1 M)	68.00	81.79

Table 5: Rotation sharing across slice pairs on LLaMA-2-7B math (GSM8K + MATH) at matched parameter budgets. Paired at rank r has the same parameter count as default at rank $r/2$. Each cell reports GSM8K/MATH; Δ is paired minus default.

Params	default	paired	Δ
$\sim$ 27.6 M	62.0/12.7 (r=64)	64.3/14.2 (r=128)	$+2.3/+1.5$
$\sim$ 13.6 M	58.3/11.5 (r=32)	60.3/12.0 (r=64)	$+2.0/+0.5$

Source-matrix variant (W_{lr} vs W_0). Theorem 2 applies to any source matrix with a per-slice SVD. We compare the rank-r reconstruction W_{lr} (default) against the full pretrained weight W_0 on commonsense (Table 6). On both LLaMA models, W_0 at r=128 ($\sim$ 56 M) approaches the W_{lr} tier band (within 1 point on LLaMA-2-7B, within band on LLaMA-3-8B) at 4–8$\times$ the parameter cost. The rank-r truncation in W_{lr} acts as a spectral regularizer in CORA's target regime; W_0 retains the tail singular directions of W_0 that the truncation discards and applies on tasks that rely on those directions (e.g., MATH).

U-side rotation only. Proposition 1 shows that CORA's default form realizes $Q_{U,i} = Q_{V,i} = Q_i$ from a single learnable R_i via the identity $Q_i = U_i^\top R_i U_i$, with $R_i U_i = U_i Q_{U,i}$ acting on the U-side and $Q_i^\top V_i^\top$ rotating the V-side. To verify that the V-side rotation $Q_{V,i}$ is necessary, we drop it and keep only $Q_{U,i}$: the U-only variant uses $\widetilde{W}_i = R_i U_i \Sigma_i \,\mathrm{diag}(\mathbf{1} + \boldsymbol{\delta})\, V_i^\top$ in place of the default $\widetilde{W}_i = R_i U_i \Sigma_i \,\mathrm{diag}(\mathbf{1} + \boldsymbol{\delta})\, Q_i^\top V_i^\top$; the skew-parameter count of R_i is unchanged. Table 7 reports HumanEval Pass@1 on Mistral-7B. Removing $Q_{V,i}$ costs 2.4 to 9.8 points across ranks; the loss is largest at r=128 where the F variant relies most on the two-sided structure. This confirms that realizing the coherent rotation form on both singular bases is a load-bearing part of CORA's design.

Table 6: Source-matrix variant on commonsense 8-task average. Two variants: W_{lr} (default) and W_0. L2-7B / L3-8B: averages on LLaMA-2-7B / LLaMA-3-8B.

Variant	r	#Params	L2-7B	L3-8B
W_{lr}	16	6.6 M	82.16	86.65
W_{lr}	32	13.6 M	82.06	86.75
W_0	128	55.7 M	81.30	86.70

Table 7: U-side rotation only on Mistral-7B HumanEval Pass@1. *default*: two-sided + $\boldsymbol{\delta}$; *U-only* + $\boldsymbol{\delta}$: V-side dropped; *U-only*: V-side and $\boldsymbol{\delta}$ dropped.

r	default	U-only + $\boldsymbol{\delta}$	U-only
r=128 (W_0)	40.9	31.1	29.3
r=64	40.9	35.4	38.4
r=32	40.9	38.4	40.9
r=16	40.2	37.8	35.4
r=8	40.2	36.0	39.6

6 Conclusion

We extended the coherent rotation form of finetuning from the full weight matrix to its row slices, and showed that the resulting per-slice condition reduces to an algebraic identity via $Q_i = U_i^\top R_i U_i$, removing the need for an explicit orthogonality constraint during optimization. CORA learns one orthogonal R_i per slice of the rank-r reconstruction W_{lr}, uses $\frac{1}{2}m(r-1)$ trainable parameters per linear layer independent of the column dimension k, and reassembles into a dense weight of the same shape as W_0 at deployment. Empirically, CORA r=16 at 6.6 M parameters reaches 82.16% commonsense accuracy on LLaMA-2-7B; the same method transfers to LLaMA-3-8B (commonsense and code) and Mistral-7B (code).

Limitations. CORA has several limitations. First, every adapted layer requires an offline SVD of W_0, and the per-slice factors (U_i, Σ_i, V_i) must be computed and cached on disk before training begins. Second, the slice-wise forward $\widetilde{W}_i = R_i U_i \Sigma_i Q_i^\top V_i^\top$ adds matmuls per layer at training time relative to LoRA's $W_0 + BA$ form. Third, our largest evaluated backbone is 8B (LLaMA-3-8B), and we have not validated CORA at $\geq$ 30B scale (e.g., LLaMA-3-70B, Mixtral 8×22B). Fourth, evaluation covers commonsense reasoning, math, and code generation under instruction tuning; vision-language, multilingual, long-context, and continual-learning settings remain unexplored.

References

[1] 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 *ICLR*, 2022.

[2] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024.

[3] Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *NAACL*, 2025.

[4] Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski, and Sujay Sanghavi. SVFT: Parameter-efficient fine-tuning with singular vectors. In *NeurIPS*, 2024.

[5] Fan Wang, Juyong Jiang, Chansung Park, Sunghun Kim, and Jing Tang. KaSA: Knowledge-aware singular-value adaptation of large language models. In *ICLR*, 2025.

[6] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In *NeurIPS*, 2023.

[7] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In *ICLR*, 2024.

[8] Ziran Liu, Wei Wang, Jinhao Wang, Pengcheng Wang, Xinyi Sui, Cihan Ruan, Nam Ling, and Wei Jiang. Geometric and spectral alignment for deep neural network II. *arXiv preprint arXiv:2605.02111*, 2026. URL `https://arxiv.org/abs/2605.02111`.

[9] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *ICML*, 2024.

[10] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*, 2023.

[11] Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *ICLR*, 2024.

[12] Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *NeurIPS*, 2024.

[13] Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-fine-tuning-aware quantization for large language models. In *ICLR*, 2024.

[14] Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *NeurIPS*, 2024.

[15] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-Pro: Are low-rank adapters properly optimized? In *ICLR*, 2025.

[16] Zeju Qiu, Weiyang Liu, Adrian Weller, and Bernhard Schölkopf. Orthogonal finetuning made scalable. In *EMNLP*, 2025.

[17] Zeju Qiu, Simon Buchholz, Tim Z. Xiao, Maximilian Dax, Bernhard Schölkopf, and Weiyang Liu. Reparameterized LLM training via orthogonal equivalence transformation. In *NeurIPS*, 2025.

[18] Fei Wu, Jia Hu, Geyong Min, and Shiqiang Wang. Efficient orthogonal fine-tuning with principal subspace adaptation. In *ICLR*, 2026.

[19] Selcuk Gurses, Aozhong Zhang, Yanxia Deng, Xun Dong, Xin Li, Naigang Wang, Penghang Yin, and Zi Yang. DiaBlo: Diagonal blocks are sufficient for finetuning. *arXiv preprint arXiv:2506.03230*, 2025.

[20] G. W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[21] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[22] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, 2016.

[23] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *EMNLP*, 2023.

[24] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

[25] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

[27] Sahil Chaudhary. Code Alpaca: An instruction-following LLaMA model for code generation. `https://github.com/sahil280114/codealpaca`, 2023.

[28] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[29] Juzheng Zhang, Jiacheng You, Ashwinee Panda, and Tom Goldstein. LoRI: Reducing cross-task interference in multi-task low-rank adaptation. In *COLM*, 2025.

A Proof of Theorem 2

Proof. The Frobenius norm of a row-partitioned perturbation decomposes as

$$\|\Delta W\|_F^2 = \sum\nolimits_{i=1}^{s} \|\Delta W_i\|_F^2.$$

By Assumption 1, each slice $W_i = U_i \Sigma_i V_i^\top$ satisfies the standard stability hypotheses (non-degenerate spectrum, bounded condition number, small Frobenius perturbation). The per-slice Frobenius norm is invariant under the SVD coordinates:

$$\|\Delta W_i\|_F = \|U_i^\top \Delta W_i V_i\|_F,$$

since U_i is orthogonal and V_i has orthonormal columns on the slice subspace. The slice-level minimum-perturbation problem therefore reduces, in (U_i, Σ_i, V_i) coordinates, to a spectrum-fixed Frobenius minimization on the local core.

Write the slice perturbation in local coordinates as $\Delta W_i = U_i P_i V_i^\top$, for some $P_i \in \mathbb{R}^{r \times r}$. Then $W_i^\star = U_i(\Sigma_i + P_i)V_i^\top$. Let $\Sigma_i + P_i = Q_{U,i}^\star D_i (Q_{V,i}^\star)^\top$ be the SVD of the local adapted core, with D_i diagonal. By the Schur–Horn / Von Neumann trace-inequality argument [21, §4.3 and §7.4], the constrained minimum of $\|(\Sigma_i + P_i) - \Sigma_i\|_F$ over P_i, at fixed spectrum of $\Sigma_i + P_i$, is attained by a unitary similarity transform of Σ_i. Therefore,

$$Q_{U,i}^\star = Q_{V,i}^\star = Q_i, \qquad D_i = \Sigma_i + \Delta\Sigma_i.$$

Substituting yields $W_i^\star = U_i Q_i (\Sigma_i + \Delta\Sigma_i) Q_i^\top V_i^\top$. □

B Proof of Proposition 1

Proof. Part (i) is the algebraic identity $\widetilde{W}_i = U_i Q_i \Sigma_i \operatorname{diag}(\mathbf{1}+\boldsymbol{\delta})\, Q_i^\top V_i^\top$ established in Lemma 1; we focus on (ii). For brevity we present the argument for $\boldsymbol{\delta} = \mathbf{0}$; replacing Σ_i throughout with the diagonal matrix $\Sigma_i \operatorname{diag}(\mathbf{1}+\boldsymbol{\delta})$ leaves every step unchanged, since the spectral and orthogonality structure used below is preserved.

Substituting $Q_i = U_i^\top R_i U_i$ into $\widetilde{W}_i = R_i U_i \Sigma_i Q_i^\top V_i^\top$ gives

$$\widetilde{W}_i = R_i U_i \Sigma_i U_i^\top R_i^\top U_i V_i^\top = \underbrace{R_i (U_i \Sigma_i U_i^\top) R_i^\top}_{B_i} \underbrace{U_i V_i^\top}_{C_i}.$$

Since $U_i \Sigma_i U_i^\top$ is symmetric positive semidefinite, B_i is also symmetric positive semidefinite. Moreover, $C_i C_i^\top = U_i V_i^\top V_i U_i^\top = U_i U_i^\top = I_r$, so C_i has orthonormal rows.

Let $B_i = E_i \Lambda_i E_i^\top$, with $\Lambda_i = \operatorname{diag}(\mu_1, \ldots, \mu_r)$, $\mu_\ell \geq 0$. Then $\widetilde{W}_i \widetilde{W}_i^\top = B_i C_i C_i^\top B_i^\top = B_i^2 = E_i \Lambda_i^2 E_i^\top$, so the left singular vectors of $\widetilde{W}_i$ are $\widetilde{U}_i = E_i$. The corresponding right singular vectors are $\widetilde{V}_i = C_i^\top E_i = V_i U_i^\top E_i$. Therefore, $Q^\star_{U,i} = U_i^\top \widetilde{U}_i = U_i^\top E_i$, and $Q^\star_{V,i} = V_i^\top \widetilde{V}_i = V_i^\top V_i U_i^\top E_i = U_i^\top E_i$. Hence $Q^\star_{U,i} = Q^\star_{V,i}$, proving (ii). □

C Algebraic identity for CORA

The main text's Theorem 1 ("coherent rotation form") is the $s = 1$ specialization of Theorem 2, whose per-slice proof above applies directly; the parameterization $W = U Q_U \Sigma Q_V^\top V^\top$ matches [8, Defn. 9.3]. Direct calculation shows that for any orthogonal R_i, CORA's parameterization satisfies Eq. (1) exactly. The argument uses no optimization assumption and verifies the method in Section 4.3.

Lemma 1 (Coherent rotation form holds identically)**.** *Let $W_i = U_i \Sigma_i V_i^\top$ be a slice with SVD, $U_i \in \mathbb{R}^{r\times r}$ square orthogonal, $\Sigma_i \in \mathbb{R}^{r\times r}$ diagonal, $V_i \in \mathbb{R}^{k\times r}$ with orthonormal columns. Let $R_i \in \mathbb{R}^{r\times r}$ be orthogonal, and set $Q_i := U_i^\top R_i U_i$. Then:*

(i) Q_i is orthogonal.

(ii) The CORA parameterization $\widetilde{W}_i := R_i U_i \Sigma_i Q_i^\top V_i^\top$ equals

$$\widetilde{W}_i \;=\; U_i\, Q_i\, \Sigma_i\, Q_i^\top\, V_i^\top,$$

i.e., the coherent rotation form (1) *with $\Delta\Sigma = 0$.*

*(iii) Allowing a per-layer multiplicative scale $\Sigma_i \mapsto \Sigma_i \operatorname{diag}(\mathbf{1}+\boldsymbol{\delta})$ with $\boldsymbol{\delta} \in \mathbb{R}^r$ (the **CORA+δ** variant) produces $\widetilde{W}_i = U_i Q_i \Sigma_i \operatorname{diag}(\mathbf{1}+\boldsymbol{\delta})\, Q_i^\top V_i^\top$. This still satisfies Eq.* (1)*, though the spectrum shift it spans is a one-parameter family rather than the full diagonal $\Delta\Sigma_i$ allowed by Theorem 2.*

Proof. (i) Since U_i is square orthogonal, $U_i U_i^\top = U_i^\top U_i = I_r$. Hence $Q_i^\top Q_i = (U_i^\top R_i U_i)^\top (U_i^\top R_i U_i) = U_i^\top R_i^\top (U_i U_i^\top) R_i U_i = U_i^\top R_i^\top R_i U_i = U_i^\top U_i = I_r$.

(ii) Rewrite $R_i = U_i (U_i^\top R_i U_i) U_i^\top = U_i Q_i U_i^\top$ using $U_i U_i^\top = I_r$. Substituting,

$$\widetilde{W}_i \;=\; R_i U_i \Sigma_i Q_i^\top V_i^\top \;=\; (U_i Q_i U_i^\top)\, U_i \Sigma_i Q_i^\top V_i^\top \;=\; U_i Q_i (U_i^\top U_i) \Sigma_i Q_i^\top V_i^\top \;=\; U_i Q_i \Sigma_i Q_i^\top V_i^\top.$$

(iii) Immediate from (ii) by replacing Σ_i with $\Sigma_i \operatorname{diag}(\mathbf{1}+\boldsymbol{\delta})$ (a diagonal operation; per-layer across slices). □

Lemma 1 shows that a single rotation R_i on the left singular basis is enough to recover the coherent rotation form; turning on the shared scale $\boldsymbol{\delta}$ adds a one-parameter spectrum shift inside Eq. (1).

D Cayley implementation

The map $\phi : \mathrm{skew}(b) \to \mathrm{O}(b)$ used in Section 4.3 converts each skew-symmetric block $A_i^{(g)}$ into an orthogonal rotation via the closed-form Cayley map $\phi(A) = (I - A)(I + A)^{-1}$, computed with a linear solve. It produces an exactly orthogonal rotation up to numerical precision and imposes no convergence-radius constraint on $A_i^{(g)}$, so no auxiliary regularizer is needed.

E Per-slice coherence diagnostic

We measure per-slice coherence $\cos(Q_{U,i}, Q_{V,i})$ on 32×32 row slices, averaged over the 160 target layers of LLaMA-2-7B, to verify that adapters satisfying Theorem 2 keep this quantity close to 1. Full FT, PiSSA, MiLoRA, and CORA all stay at or above 0.995, confirming that the SVD-family adapters realize the per-slice coherent rotation form.

Table 8: Per-slice coherence $\cos(Q_{U,i}, Q_{V,i})$ at 32×32 row slices, averaged over 160 layers of LLaMA-2-7B. CORA satisfies $Q_{U,i} = Q_{V,i}$ identically by Lemma 1.

Method	Per-slice $\cos(Q_U, Q_V)$
Full FT	0.9994
PiSSA r=128	0.9995
MiLoRA r=32	0.9950
CORA (ours)	**0.9999**

F Spectral-fingerprint diagnostic

The two adaptation axes (spectrum shift $\Delta\Sigma$ and coherent rotation Q with $Q_U = Q_V$) can be measured directly against a Full-FT reference. This appendix defines the diagnostic, reports it across representative PEFT methods, and discusses what the numbers reveal about how closely each method satisfies Eq. (1).

Two quantities. Given a finetuned weight W with SVD $W = U\Sigma V^\top$, we measure, relative to W_0:

(i) the relative singular-value shift $\|\Delta\Sigma/\Sigma\|$; and

(ii) the spectral coherence $\cos(Q_U, Q_V)$ with $Q_U = U_0^\top U_{:,r}$ and $Q_V = V_0^\top V_{:,r}$.

Quantity (i) measures *how much* the adapter moves the spectrum; quantity (ii) measures whether the left and right subspaces move *together*, which is the defining condition of the coherent rotation form.

Table 9: Global-SVD spectral fingerprint of representative finetuning methods on LLaMA-2-7B (commonsense_170k, 3 epochs, averaged over the 160 target layers).

Method	$\lvert\Delta\Sigma/\Sigma\rvert$	$\cos(Q_U, Q_V)$	Avg acc.
Full FT	0.001	0.99999	83.5
PiSSA r=128	0.006	0.99993	81.2
DoRA r=128	0.391	0.99074	77.2

Coherence tracks accuracy. $\cos(Q_U, Q_V) \geq 0.9999$ holds for Full FT; an adapter that approximates the minimum-perturbation form (Theorem 1) inherits the same coherence. PiSSA at r=128 comes within 10^{-5} of Full FT on this axis and is within 2.3 points in accuracy. DoRA at the same parameter budget drifts to $\cos(Q_U, Q_V) = 0.99074$ and trails by ~ 6 points. The diagnostic separates adapters that satisfy the coherent rotation form from those that depart from it; CORA, by Lemma 1, satisfies Eq. (1) exactly.

G Hyperparameter summary

Table 10: Shared hyperparameters across main-text experiments.

Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)
LR schedule	Cosine decay, 100 warmup steps
Batch size	16 (commonsense / math), 32 (code)
Micro-batch (H100)	16, no gradient checkpointing
Micro-batch (L40S)	4 + gradient checkpointing
Weight decay	0
Precision	bf16 mixed
`torch.compile`	Enabled
L_2 on $\boldsymbol{\delta}$	10^{-3}

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 Section 1 state the per-slice coherent rotation theorem (Theorem 2), the CORA parameterization, and the empirical claims, all of which are supported in Sections 4 and 5.

   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 in Section 6 discusses constraints inherent to SVD-reparameterized PEFT (offline SVD caching, additional training-time matmuls), the 7–8B backbone scope, and the limited task coverage (commonsense, math, code).

   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: Assumption 1 is stated explicitly in Section 4.1; full proofs of Theorem 2, Proposition 1, and Lemma 1 are provided in Appendices A, B, and C.

   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: Section 5.1 lists models, datasets, target modules, and evaluation metrics; Section 4.3 fully specifies the algorithm; Appendix G reports training hyperparameters.

   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: [No]

   Justification: All datasets and base models used are publicly available with citations in Section 5.1; the training and evaluation code will be released upon publication under CC BY 4.0.

   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: Section 5.1 reports models, datasets, target modules, and evaluation metrics; Appendix G provides optimizer, schedule, batch size, precision, and regularization.

   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: [No]

   Justification: Each configuration trains a 7–8B-parameter model, which makes multi-seed runs prohibitive within the available compute budget; we report single-seed numbers consistent with the DoRA, MiLoRA, and DiaBlo evaluation protocols.

   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 G reports hardware (H100 / L40S), micro-batch, gradient checkpointing, and precision used across runs.

   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 work uses only publicly released models and datasets, involves no human subjects, and conforms to the NeurIPS Code of Ethics in every respect.

   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: Like LoRA and DoRA, CORA lowers the cost of finetuning, which can broaden access for users with limited compute. It introduces no new dual-use risks beyond those already inherent in the underlying pretrained models, and we release no new pretrained weights or datasets.

    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 any new high-risk pretrained model or scraped dataset.

    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: The base models (LLaMA-2-7B, LLaMA-3-8B, Mistral-7B) and datasets (commonsense_170k, MetaMathQA-395K, GSM8K, MATH, CodeAlpaca, HumanEval) are cited in Section 5.1 with their original sources, and we comply with their published terms of use.

    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: [N/A]

    Justification: The paper does not release new datasets or pretrained models with this submission.

    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 involve crowdsourcing or research with human subjects.

    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 human subjects research.

    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: [N/A]

    Justification: LLMs were not used as a component of the core methods; the per-slice coherent rotation theorem, the parameterization, and the algorithm were derived and designed by the authors. Per the question's exemption clause, declaration is not required.

    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.