FRET: FIXED-REFERENCE TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation learns through a compact projection of the model's input features. Normalizing this projection controls the contribution of each input channel, yet a learned rotation redistributes these gains through correlations between projection columns. We introduce Fixed-Reference Tuning (FRET), which imposes unit column norms on the effective projection in the layer's input coordinates. Its coupled realization projects the raw factor onto this constraint at each forward pass; its decoupled realization learns the constrained projection through independent coordinates. We characterize gain redistribution and show that coupled normalization removes the radial component of a rotation update at identity. The resulting layer combines a rotated frozen base with a normalized low-rank residual and admits exact dense export. Across six mathematical-reasoning benchmarks and three paired training seeds, both realizations exceed the naive moving-frame combination: FRET-C by 0.43 ± 0.05 macro points, with no negative benchmark–seed difference, while the two realizations differ by 0.12 points, within seed variation.

1 INTRODUCTION

Large language models increasingly acquire task-specific capabilities through supervised finetuning and reinforcement learning. Their scale makes compact adaptation valuable: a small set of trainable parameters reduces optimizer storage and supports multiple adaptations of a shared pretrained model (Hu et al., 2021; Yin et al., 2025). Low-rank residuals learn additive corrections, while structured rotations reorient the pretrained weights (Qiu et al., 2023; Liu et al., 2024b). Combining these transformations makes the geometry of the residual's input projection central to learning.

Consider a low-rank residual $s\boldsymbol{B}\boldsymbol{A}$ initialized with $\boldsymbol{B} = \boldsymbol{0}$. The pretrained function is preserved, and the first update to $\boldsymbol{B}$ depends on the features produced by $\boldsymbol{A}$. Column normalization assigns equal projection length to unit changes in each input channel (Kang et al., 2026). A learned input rotation changes this geometry. The residual receives $\mathcal{N}_r(\boldsymbol{A})\boldsymbol{R}^\top \boldsymbol{x}$, whose effective projection is $\mathcal{N}_r(\boldsymbol{A})\boldsymbol{R}^\top$. Correlations between columns allow rotation to redistribute their squared lengths while preserving their sum. Thus, normalization attached to the stored factor regulates a *moving frame*; the gains seen along the original input coordinates evolve with the rotation.

Those input coordinates have a concrete architectural meaning. Each is a feature channel passed into the adapted layer. For layers receiving RMS-normalized features, learned channelwise gains already distinguish these axes (Zhang & Sennrich, 2019). Studies of transformer activations also identify extreme responses concentrated in specific feature dimensions (Sun et al., 2024). These observations motivate keeping the projection constraint in the *fixed input frame*, where its effect on individual channels stays explicit throughout adaptation.

We propose **Fixed-Reference Tuning** (FRET): constrain the effective down-projection $\boldsymbol{C}$ to have unit columns in that frame. The frozen base can rotate, and the residual learns within the resulting product of unit spheres. For a raw factor $\boldsymbol{A}$ used after $\boldsymbol{R}^\top$, the nearest feasible realized factor is

$$\boldsymbol{F}_{\mathrm{FRET}} = \mathcal{N}_r(\boldsymbol{A}\boldsymbol{R}^\top)\boldsymbol{R}, \qquad \boldsymbol{C}_{\mathrm{FRET}} = \mathcal{N}_r(\boldsymbol{A}\boldsymbol{R}^\top). \tag{1}$$

Here $\mathcal{N}_r$ normalizes each column along the rank dimension. This *coupled* realization, FRET-C, expresses the factor in input coordinates, normalizes it, and returns it to the learned frame (Figure 1). A *decoupled* realization, FRET-D, learns an independent raw projection $\boldsymbol{Z}$ and sets $\boldsymbol{C} = \mathcal{N}_r(\boldsymbol{Z})$. Both represent the same constrained layers, with distinct gradient paths through the rotation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

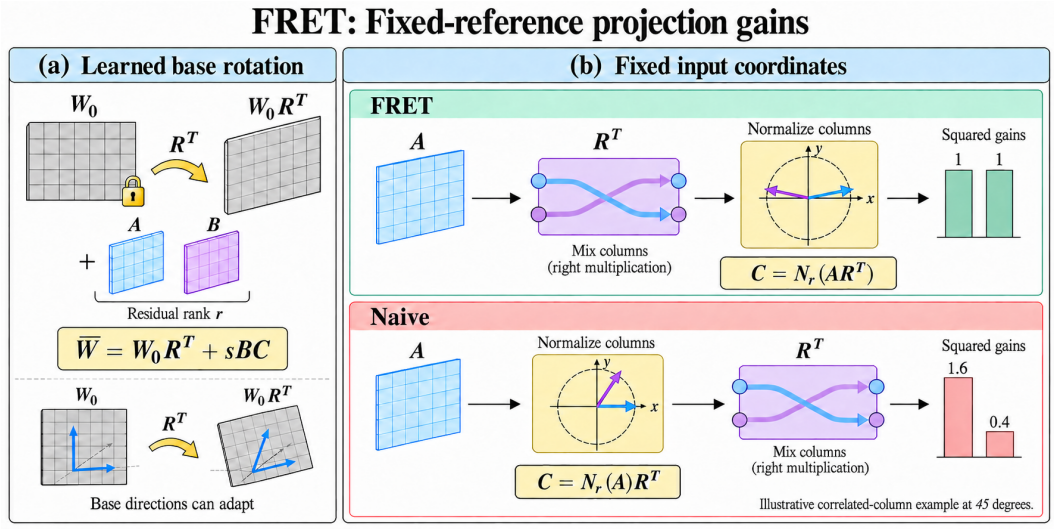


Figure 1: **Fixed-reference projection geometry.** (a) The unrotated base $\boldsymbol{W}_0$ and its learned orientation $\boldsymbol{W}_0\boldsymbol{R}^\top$ share a low-rank residual $s\boldsymbol{BC}$. (b) FRET-C (top) mixes columns before normalization; Naive (bottom) normalizes before mixing. The squared gains $(1, 1)$ and $(1.6, 0.4)$ use the analytic example in Figure 4. Vector sketches illustrate the operations. The full layer applies the shared row scale Γ_m to the displayed core $\overline{\boldsymbol{W}}$. Table 2 lists construction and dense export.

Our analysis connects the constraint to the update geometry. At identity rotation, the coupled map removes precisely the radial component responsible for first-order gain drift. Its tangential component retains changes in projection direction. We also derive how within-block column correlations govern finite gain redistribution and how the coupled coordinates feed the residual gradient into the rotation. These results give separate roles to the constraint, its parameterization, and the optimizer.

Our contributions are:

- A fixed-reference projection constraint with coupled and decoupled realizations, a nearest-factor derivation, and an exact dense export.
- A characterization of rotation-induced gain redistribution and its radial component, together with the gradient coupling induced by the two realizations.
- A six-benchmark, three-seed comparison in which both realizations exceed moving-frame normalization (FRET-C by 0.43 ± 0.05 macro points, non-negative on every benchmark and seed) while differing from each other within seed variation, complemented by a closed-form example of the constraint's action.

2 Related Work

Normalized low-rank adaptation. Weight normalization separates a weight vector's magnitude and direction (Salimans & Kingma, 2016). LoRA learns a low-rank residual around a frozen weight (Hu et al., 2021), and DoRA applies magnitude–direction decomposition to that adaptation (Liu et al., 2024a). NoRA normalizes down-projection columns along the rank dimension, persistently or at initialization, and studies their combination with DoRA (Kang et al., 2026). FRET places this column constraint on the effective projection after a learned change of coordinates. Our experimental layers retain DoRA's output-row scaling convention; Section 4.1 and Appendix F.1 define the compared recipes.

Orthogonal adaptation. OFT learns orthogonal transformations of pretrained weights; BOFT and OFTv2 develop structured and computationally efficient realizations (Qiu et al., 2023; Liu et al., 2024b; Qiu et al., 2025b). HRA represents an orthogonal update by a chain of Householder reflections (Yuan et al., 2024). ETHER uses reflection-based transformations (Bini et al., 2024), while Group and Shuffle composes block-diagonal factors and permutations (Gorbunov et al., 2024).

These constructions determine the rotation family. FRET specifies the projection constraint coupled to such a rotation and retains a separate low-rank residual. POET and Pion study spectrum-preserving reparameterization and optimization (Qiu et al., 2025a; Shi et al., 2026); Pion also transports optimizer momentum between tangent spaces. FRET's residual and learned output scales allow the complete weight spectrum to adapt.

Coordinates and optimization. LoRA-RITE makes optimization invariant to equivalent factor scalings and rotations (Yen et al., 2025). Bilinear Optimization Divergence relates factor constraints to preservation of effective updates on historical subspaces (Wang et al., 2026). FRET regulates input-coordinate projection gains. Its coupled and decoupled forms share a function class, allowing the reference constraint and the optimization coordinates to be studied separately. Evaluations of parameter-efficient RLVR methods (Yin et al., 2025) and geometrically constrained subspaces (Zhang et al., 2026) motivate examining these choices in mathematical reasoning.

3 Fixed-Reference Tuning

3.1 The projection seen by the residual

For column-vector inputs, let the frozen weight be $\boldsymbol{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d}$, the raw low-rank factors be $\boldsymbol{A} \in \mathbb{R}^{r \times d}$ and $\boldsymbol{B} \in \mathbb{R}^{d_{\text{out}} \times r}$, and the learned rotation be $\boldsymbol{R} \in \mathrm{O}(d)$. Write $\boldsymbol{F}$ for the factor realized from $\boldsymbol{A}$ in the forward pass. The adapted core is

$$\overline{\boldsymbol{W}} = (\boldsymbol{W}_0 + s\boldsymbol{B}\boldsymbol{F})\boldsymbol{R}^\top = \boldsymbol{W}_0\boldsymbol{R}^\top + s\boldsymbol{B}\boldsymbol{C}, \qquad \boldsymbol{C} = \boldsymbol{F}\boldsymbol{R}^\top, \tag{2}$$

where s is the residual scale. The rotation reorients the frozen base; $s\boldsymbol{B}\boldsymbol{C}$ is a rank-at-most-r residual.

These changes provide distinct degrees of freedom. For an identity base of even dimension, a nonzero, non-full-turn rotation in every two-dimensional block produces a full-rank displacement $\boldsymbol{R}^\top - \boldsymbol{I}$. Expressing this displacement additively requires rank d. A low-rank residual, in turn, can change singular values that a pure rotation preserves. The two components therefore cover complementary weight transformations.

The response of the effective down-projection to coordinate j is $\boldsymbol{C}\boldsymbol{e}_j$. We require

$$\boldsymbol{C} \in \mathcal{S}_{\boldsymbol{I}} = \{\boldsymbol{C} : \operatorname{diag}(\boldsymbol{C}^\top\boldsymbol{C}) = \boldsymbol{1}\}, \tag{3}$$

so a unit perturbation to each input channel produces a unit-length latent response. The up-projection $\boldsymbol{B}$ then maps these normalized features into output coordinates. The layer's input channels supply a reference independent of the adapter's internal rotation.

For a raw column norm $q_j = \|\boldsymbol{A}_{:,j}\|_2$, define

$$\mathcal{N}_r(\boldsymbol{A})_{:,j} = \boldsymbol{A}_{:,j}/\rho_\epsilon(q_j). \tag{4}$$

Ideal normalization uses $\rho_0(q) = q$ for $q > 0$. Two numerical choices are the max floor $\max(q, \epsilon)$ and the additive denominator $q + \epsilon$. The accompanying implementation uses the latter with $\epsilon = 10^{-6}$. We derive the unit-gain constraint for ideal normalization; Appendix D gives the exact gain deficit and derivatives for both numerical choices. All folding identities assume dropout-disabled linear computation.

3.2 Nearest feasible factor

At a fixed rotation, choose the factor satisfying Equation (3) with minimum displacement from $\boldsymbol{A}$:

$$\boldsymbol{F}_* = \arg\min_{\boldsymbol{F}:\, \boldsymbol{F}\boldsymbol{R}^\top \in \mathcal{S}_{\boldsymbol{I}}} \|\boldsymbol{F} - \boldsymbol{A}\|_F^2. \tag{5}$$

Set $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{R}^\top$ and $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{R}^\top$. Orthogonality gives $\|\boldsymbol{F} - \boldsymbol{A}\|_F = \|\boldsymbol{C} - \boldsymbol{Z}\|_F$, so the problem separates into independent unit-sphere projections. For a nonzero column $\boldsymbol{z}$ and $\|\boldsymbol{c}\|_2 = 1$,

$$\|\boldsymbol{c} - \boldsymbol{z}\|_2^2 = 1 + \|\boldsymbol{z}\|_2^2 - 2\boldsymbol{c}^\top\boldsymbol{z} \geq (\|\boldsymbol{z}\|_2 - 1)^2. \tag{6}$$

Cauchy–Schwarz gives equality uniquely at $\boldsymbol{c} = \boldsymbol{z}/\|\boldsymbol{z}\|_2$. Returning to factor coordinates yields

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{R}^\top, \qquad \boldsymbol{C}_{\text{FRET}} = \mathcal{N}_r(\boldsymbol{Z}), \qquad \boldsymbol{F}_{\text{FRET}} = \boldsymbol{C}_{\text{FRET}}\boldsymbol{R}. \tag{7}$$

Realization	Learned coordinates	Effective projection $\boldsymbol{C}$	Unit gains
FRET-C	$\boldsymbol{A}, \boldsymbol{R}$	$\mathcal{N}_r(\boldsymbol{A}\boldsymbol{R}^\top)$	Fixed input frame
FRET-D	$\boldsymbol{Z}, \boldsymbol{R}$	$\mathcal{N}_r(\boldsymbol{Z})$	Fixed input frame
Naive	$\boldsymbol{A}, \boldsymbol{R}$	$\mathcal{N}_r(\boldsymbol{A})\boldsymbol{R}^\top$	Moving frame
Unnormalized	$\boldsymbol{A}, \boldsymbol{R}$	$\boldsymbol{A}\boldsymbol{R}^\top$	Unconstrained
No rotation	$\boldsymbol{A}$	$\mathcal{N}_r(\boldsymbol{A})$	Fixed input frame

Table 1: Projection realizations under ideal normalization of nonzero columns. All rows also learn $\boldsymbol{B}$; the no-rotation row freezes $\boldsymbol{R} = \boldsymbol{I}$. FRET-C and FRET-D share the fixed-reference constraint.

This is the coupled realization FRET-C in Figure 1. The nearest point is unique for every nonzero column. At zero, all unit vectors tie; the numerical map uses the continuous floor convention in Appendix D.

The effective projection satisfies

$$\boldsymbol{F}_{\mathrm{FRET}}\boldsymbol{R}^\top = \boldsymbol{C}_{\mathrm{FRET}}, \qquad \|(\boldsymbol{C}_{\mathrm{FRET}})_{:,j}\|_2 = q_j/\rho_\epsilon(q_j), \tag{8}$$

where $q_j = \|\boldsymbol{Z}_{:,j}\|_2$. Thus the core folds to

$$\overline{\boldsymbol{W}} = \boldsymbol{W}_0\boldsymbol{R}^\top + s\boldsymbol{B}\boldsymbol{C}_{\mathrm{FRET}}. \tag{9}$$

The learned base rotation remains in the first term. Since every factor is input-independent, inference can use one dense weight. Section 3.5 gives the shared row-rescaled realization.

For comparison, normalizing the stored factor gives

$$\boldsymbol{C}_{\mathrm{Naive}} = \mathcal{N}_r(\boldsymbol{A})\boldsymbol{R}^\top, \qquad \boldsymbol{C}_{\mathrm{raw}} = \boldsymbol{A}\boldsymbol{R}^\top. \tag{10}$$

In the basis $\boldsymbol{R}$, $\boldsymbol{C}_{\mathrm{Naive}}\boldsymbol{R}$ has unit columns. Its constraint therefore moves with the rotation. Table 1 lists the resulting maps.

3.3 Gain redistribution

Let $\boldsymbol{U}$ contain unit columns and $\boldsymbol{H} = \boldsymbol{U}^\top\boldsymbol{U}$. After rotation, the squared gains are $\mathrm{diag}(\boldsymbol{R}\boldsymbol{H}\boldsymbol{R}^\top)$. The trace remains d, while correlations in the off-diagonal entries of $\boldsymbol{H}$ can change its diagonal. The following result characterizes rotations that preserve the constraint for every unit-column projection.

Proposition 1 (Preservation for every projection)**.** *Fix $r \geq 1$ and $d \geq 2$. For $\boldsymbol{R} \in \mathrm{O}(d)$, $\mathrm{diag}(\boldsymbol{U}^\top\boldsymbol{U}) = \mathbf{1}$ implies $\mathrm{diag}(\boldsymbol{R}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{R}^\top) = \mathbf{1}$ for every $\boldsymbol{U} \in \mathbb{R}^{r\times d}$ if and only if $\boldsymbol{R}$ is a signed permutation.*

Proof. Signed permutations reorder columns and change their signs. For necessity, take $\boldsymbol{U} = \boldsymbol{u}\boldsymbol{\sigma}^\top$ with $\|\boldsymbol{u}\|_2 = 1$ and every sign vector $\boldsymbol{\sigma} \in \{-1, 1\}^d$. Any row $\boldsymbol{v}^\top$ of a universally preserving rotation must satisfy

$$(\boldsymbol{v}^\top\boldsymbol{\sigma})^2 = \sum_j v_j^2 + 2\sum_{k<\ell} v_k v_\ell \sigma_k \sigma_\ell = 1. \tag{11}$$

Multiply by $\sigma_k\sigma_\ell$ and average over independent uniform signs. This gives $v_k v_\ell = 0$ for $k \neq \ell$, so each row has at most one nonzero entry. Orthogonality makes that entry ± 1 and places it in a different column for each row. □

For a learned rotation close to identity, the useful quantity is its local interaction with the current column correlations. The next result gives this derivative and a separate random-rotation reference value.

Proposition 2 (Correlation-to-gain redistribution)**.** *Fix a block $\boldsymbol{U}_b \in \mathbb{R}^{r\times b}$ with unit columns, and write $\boldsymbol{H} = \boldsymbol{U}_b^\top\boldsymbol{U}_b$ and $\boldsymbol{q}(\boldsymbol{R}_b) = \mathrm{diag}(\boldsymbol{R}_b\boldsymbol{H}\boldsymbol{R}_b^\top)$. For $\boldsymbol{R}_b(t) = \exp(t\boldsymbol{\Omega})$, $\boldsymbol{\Omega}^\top = -\boldsymbol{\Omega}$,*

$$\dot{q}_j(0) = 2\sum_k \Omega_{jk}H_{jk}. \tag{12}$$

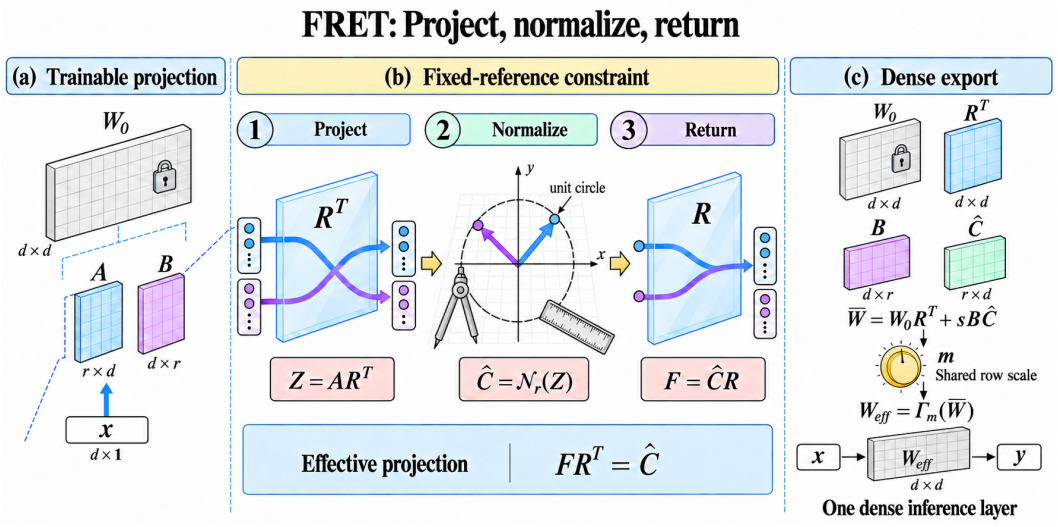


Figure 2: **FRET-C construction and dense export.** (a) Frozen and trainable factors for a square layer. (b) Project, normalize, and return, yielding $\boldsymbol{F}\boldsymbol{R}^\top = \widehat{\boldsymbol{C}}$; $\boldsymbol{Z}$ denotes the intermediate $\boldsymbol{A}\boldsymbol{R}^\top$ here. (c) Apply the shared row scale $\Gamma_{\boldsymbol{m}}$ to the composed core for one dense inference layer.

For an independent Haar-uniform $\boldsymbol{R}_b \in \mathrm{O}(b)$,

$$\mathbb{E}_{\boldsymbol{R}_b}\left[\frac{\|\boldsymbol{q}(\boldsymbol{R}_b) - \mathbf{1}\|_2^2}{b}\right] = \frac{2\sum_{j\neq k} H_{jk}^2}{b(b+2)}. \tag{13}$$

When $r < b$*, this expectation is at least* $2(b-r)/[r(b+2)]$.

Proof. The derivative of $\boldsymbol{R}_b\boldsymbol{H}\boldsymbol{R}_b^\top$ at zero is $\boldsymbol{\Omega}\boldsymbol{H} - \boldsymbol{H}\boldsymbol{\Omega}$, whose jth diagonal entry gives Equation (12). The expectation follows from the sphere's fourth moment; Appendix H gives the calculation and rank bound. □

The local derivative couples a rotation direction to existing column correlations. The Haar expectation averages this interaction over independent random rotations. When $r \geq b$, the block can have orthonormal columns, giving $\boldsymbol{H} = \boldsymbol{I}$ and zero redistribution under every rotation. When $r < b$, the rank bound enforces nonzero within-block correlation energy.

A finite-displacement bound also follows by setting $\boldsymbol{K} = \boldsymbol{H} - \boldsymbol{I}$:

$$\|\boldsymbol{q}(\boldsymbol{R}_b) - \mathbf{1}\|_\infty \leq 2\|\boldsymbol{R}_b - \boldsymbol{I}\|_2\|\boldsymbol{H} - \boldsymbol{I}\|_2. \tag{14}$$

Indeed, $\mathrm{diag}(\boldsymbol{K}) = \mathbf{0}$ and $\boldsymbol{R}_b\boldsymbol{K}\boldsymbol{R}_b^\top - \boldsymbol{K} = (\boldsymbol{R}_b - \boldsymbol{I})\boldsymbol{K}\boldsymbol{R}_b^\top + \boldsymbol{K}(\boldsymbol{R}_b^\top - \boldsymbol{I})$. The bound separates rotation displacement from the projection's correlation norm. For $r < b$, the Gram matrix has a zero eigenvalue, so $\|\boldsymbol{H} - \boldsymbol{I}\|_2 \geq 1$; small displacement can still keep their product small. Both quantities can be read directly from a checkpoint.

A two-channel example. For two unit columns with inner product ρ, take

$$\boldsymbol{H} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \qquad \boldsymbol{R}(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{15}$$

Their squared gains after moving-frame normalization are

$$q_1(\theta) = 1 + \rho\sin(2\theta), \qquad q_2(\theta) = 1 - \rho\sin(2\theta). \tag{16}$$

The maximum gain error is $|\rho\sin(2\theta)|$: it vanishes for uncorrelated columns and peaks at $45°$. Fixed-reference normalization restores both gains to one for nonzero mixed columns. At $0°$ and $90°$, both maps agree because the rotation is a signed permutation.

3.4 FUNCTION CLASS AND OPTIMIZATION COORDINATES

Two coordinates for the constraint. Consider a separate trainable factor $\boldsymbol{Z}$ and the layer

$$\overline{\boldsymbol{W}}_{\mathrm{dec}} = \boldsymbol{W}_0\boldsymbol{R}^\top + s\boldsymbol{B}\mathcal{N}_r(\boldsymbol{Z}). \tag{17}$$

This gives the decoupled realization FRET-D. For every $\boldsymbol{A}$ and orthogonal $\boldsymbol{R}$, set $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{R}^\top$; conversely, set $\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{R}$. Thus the decoupled layer and FRET-C have exactly the same attainable weights, including under a shared outer row scaling and the same numerical denominator. The nearest-factor construction uses coupled coordinates $(\boldsymbol{A}, \boldsymbol{R})$; the decoupled construction uses $(\boldsymbol{Z}, \boldsymbol{R})$. FRET-C versus Naive tests the constraint placement, while FRET-C versus FRET-D tests the coordinates within the fixed-reference constraint.

For a nonzero raw column $\boldsymbol{z}$, write $q = \|\boldsymbol{z}\|_2$ and $\boldsymbol{u} = \boldsymbol{z}/q$. Its normalization Jacobian is

$$\boldsymbol{J}(\boldsymbol{z}) = \frac{\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top}{q}. \tag{18}$$

The Jacobian removes changes in length and retains changes in direction, scaled by $1/q$. Small raw columns can therefore have large derivatives even when their normalized forward response has unit length.

Rotation updates at identity. Take unit-column $\boldsymbol{A}$ and $\boldsymbol{R}(t) = \exp(t\boldsymbol{\Omega})$. At $t = 0$, both normalized maps equal $\boldsymbol{A}$. Define $\boldsymbol{V} = -\boldsymbol{A}\boldsymbol{\Omega}$. Their derivatives are

$$\dot{\boldsymbol{C}}_{\mathrm{Naive},:,j} = \boldsymbol{V}_{:,j}, \qquad \dot{\boldsymbol{C}}_{\mathrm{FRET},:,j} = (\boldsymbol{I} - \boldsymbol{A}_{:,j}\boldsymbol{A}_{:,j}^\top)\boldsymbol{V}_{:,j}. \tag{19}$$

Write $\boldsymbol{a}_j = \boldsymbol{A}_{:,j}$ and $\boldsymbol{H} = \boldsymbol{A}^\top\boldsymbol{A}$. The radial coefficient in the Naive update satisfies

$$\boldsymbol{a}_j^\top \boldsymbol{V}_{:,j} = \sum_k H_{jk}\Omega_{jk} = \tfrac{1}{2}\dot{q}_j(0). \tag{20}$$

Consequently,

$$\dot{\boldsymbol{C}}_{\mathrm{FRET},:,j} = \dot{\boldsymbol{C}}_{\mathrm{Naive},:,j} - \tfrac{1}{2}\dot{q}_j(0)\boldsymbol{a}_j. \tag{21}$$

FRET-C removes exactly the radial term responsible for first-order gain drift, retaining the tangential change in feature direction. For the planar example, these removed coefficients are ρ and $-\rho$ per unit rotation angle at identity. The two maps thus share their initial forward value and differ in their rotation derivatives. Freezing $\boldsymbol{R} = \boldsymbol{I}$ throughout training gives an equality control for both the forward and backward maps.

Raw-coordinate coupling. Let $\boldsymbol{G}_Z = \partial\mathcal{L}/\partial\boldsymbol{Z}$ and let $\boldsymbol{G}_R^{\mathrm{direct}}$ denote dependence on $\boldsymbol{R}$ with $\boldsymbol{Z}$ fixed. Because $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{R}^\top$,

$$\begin{aligned} d\boldsymbol{Z} &= d\boldsymbol{A}\,\boldsymbol{R}^\top + \boldsymbol{A}\,d\boldsymbol{R}^\top, \\ \nabla_{\boldsymbol{A}}\mathcal{L} = \boldsymbol{G}_Z\boldsymbol{R}, &\qquad \nabla_{\boldsymbol{R}}\mathcal{L} = \boldsymbol{G}_R^{\mathrm{direct}} + \boldsymbol{G}_Z^\top\boldsymbol{A}. \end{aligned} \tag{22}$$

The rotation parameterization pulls back the last expression to its trainable coordinates. In the coupled realization, a change to $\boldsymbol{R}$ also changes the residual features through $\boldsymbol{Z}$. With independent $(\boldsymbol{Z}, \boldsymbol{R})$, the second gradient contains only $\boldsymbol{G}_R^{\mathrm{direct}}$. This is an optimization distinction between two representations of the same constrained layer. Both uses of $\boldsymbol{R}$ in Equation (7) must remain in the computation graph to implement the stated raw-coordinate method.

At a frozen $\boldsymbol{R}$, plain SGD on $\boldsymbol{A}$ maps to the same SGD step on $\boldsymbol{Z}$, since $(\boldsymbol{G}_Z\boldsymbol{R})\boldsymbol{R}^\top = \boldsymbol{G}_Z$. Elementwise Adam updates generally lose this equivalence: rotating a gradient before squaring its entries changes the second-moment state. When $\boldsymbol{R}$ is also trained, the additional differential term in Equation (22) remains even for SGD. These identities locate the trajectory difference in the residual-to-rotation gradient and the optimizer's coordinate-dependent preconditioner.

Connection to the weight update. For the additive core with $\boldsymbol{C}$ held fixed, let $\boldsymbol{G} = \partial\mathcal{L}/\partial\overline{\boldsymbol{W}}$. An SGD step on $\boldsymbol{B}$ gives

$$\nabla_{\boldsymbol{B}}\mathcal{L} = s\boldsymbol{G}\boldsymbol{C}^\top, \qquad s\Delta\boldsymbol{B}\boldsymbol{C} = -\eta s^2\boldsymbol{G}\boldsymbol{C}^\top\boldsymbol{C}. \tag{23}$$

Step	Training and dense export
Input	Frozen $\boldsymbol{W}_0$; trainable $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{m}$ and rotation parameters; input $\boldsymbol{x}$; residual scale s; floor ρ_ϵ.
1	Construct orthogonal blocks $\boldsymbol{R}_k$ and retain their computation graph.
2	Compute $\boldsymbol{Z}_k = \boldsymbol{A}_k \boldsymbol{R}_k^\top$ and $\boldsymbol{C}_{\mathrm{FRET},k} = \mathcal{N}_r(\boldsymbol{Z}_k)$.
3	Return $\boldsymbol{F}_k = \boldsymbol{C}_{\mathrm{FRET},k} \boldsymbol{R}_k$ and assemble $\boldsymbol{U} = \boldsymbol{W}_0 + s\boldsymbol{B}\boldsymbol{F}_{\mathrm{FRET}}$.
4	Evaluate $\boldsymbol{y} = \Gamma_{\boldsymbol{m}}(\boldsymbol{U})\boldsymbol{R}^\top \boldsymbol{x}$; add the unchanged bias separately.
5	Backpropagate through normalization and both uses of the rotation.
Export	Store $\Gamma_{\boldsymbol{m}}(\boldsymbol{W}_0 \boldsymbol{R}^\top + s\boldsymbol{B}\boldsymbol{C}_{\mathrm{FRET}})$ and the bias.

Table 2: FRET-C training and export. Steps 2–3 construct the factor from parameters. Step 4 applies it to the rotated activation $\boldsymbol{R}^\top \boldsymbol{x}$.

The rightmost matrix mixes and scales input-coordinate gradients. Its diagonal sets each coordinate's direct contribution, and its off-diagonal entries couple coordinates. The fixed-reference constraint sets the diagonal contribution to one, while the learned column correlations determine cross-coordinate contributions.

At $\boldsymbol{B} = \boldsymbol{0}$, the first gradient through the residual into the down-projection is zero, so the displayed identity also describes the residual component of the first plain-SGD update. A simultaneous base rotation supplies another component. With output-row scaling, the gradient $\boldsymbol{G}$ must first be pulled back through that map; with Adam, the elementwise preconditioner changes the update.

3.5 Trainable layer and dense export

Blockwise rotation. Partition the d input channels into blocks of width b, with $b \mid d$. Each block has an independent skew-symmetric parameter $\boldsymbol{S}_k$, stored through its $b(b-1)/2$ upper-triangular entries. We use the Cayley map

$$\boldsymbol{R} = \mathrm{blockdiag}(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_{d/b}), \qquad \boldsymbol{R}_k = (\boldsymbol{I} - \boldsymbol{S}_k)(\boldsymbol{I} + \boldsymbol{S}_k)^{-1}. \tag{24}$$

The inverse exists for real skew-symmetric $\boldsymbol{S}_k$; direct multiplication gives $\boldsymbol{R}_k^\top \boldsymbol{R}_k = \boldsymbol{I}$. The implementation evaluates each independent block by a direct linear solve. Setting $\boldsymbol{S}_k = \boldsymbol{0}$ initializes the rotation at identity. The earlier identities apply to every orthogonal member of this blockwise rotation family.

The empirical layer applies the shared output-row convention attributed in Section 2. For a nonzero row of $\boldsymbol{U}$, define

$$[\Gamma_{\boldsymbol{m}}(\boldsymbol{U})]_{i,:} = m_i \boldsymbol{U}_{i,:} / \|\boldsymbol{U}_{i,:}\|_2, \qquad \boldsymbol{W}_{\mathrm{eff}} = \Gamma_{\boldsymbol{m}}(\overline{\boldsymbol{W}}). \tag{25}$$

For orthogonal $\boldsymbol{R}$, every row obeys $\|\boldsymbol{u}_i \boldsymbol{R}^\top\|_2 = \|\boldsymbol{u}_i\|_2$. Hence row scaling commutes with this rotation:

$$\Gamma_{\boldsymbol{m}}(\boldsymbol{U})\boldsymbol{R}^\top = \Gamma_{\boldsymbol{m}}(\boldsymbol{U}\boldsymbol{R}^\top). \tag{26}$$

Substituting $\boldsymbol{U} = \boldsymbol{W}_0 + s\boldsymbol{B}\boldsymbol{C}_{\mathrm{FRET}}\boldsymbol{R}$ gives

$$\boldsymbol{W}_{\mathrm{eff}} = \Gamma_{\boldsymbol{m}}(\boldsymbol{W}_0 + s\boldsymbol{B}\boldsymbol{C}_{\mathrm{FRET}}\boldsymbol{R})\boldsymbol{R}^\top = \Gamma_{\boldsymbol{m}}(\boldsymbol{W}_0 \boldsymbol{R}^\top + s\boldsymbol{B}\boldsymbol{C}_{\mathrm{FRET}}). \tag{27}$$

This proves exact deterministic export of the full layer. Output-row norms equal $|m_i|$ when the direction rows are nonzero. The projection constraint operates inside this shared outer map.

Table 2 retains the existing rotated-input layer interface using block products on the thin factor. Alternatively, evaluating the right-hand side of Equation (27) directly removes the return product $\boldsymbol{C}_{\mathrm{FRET}}\boldsymbol{R}$ from the residual path. This algebraic simplification retains the same coupled variables $(\boldsymbol{A}, \boldsymbol{R})$ and their gradients when all parameter dependencies are differentiated. It is distinct from training an independent $\boldsymbol{Z}$. Inference uses the exported dense matrix and the original layer interface. During training, row-denominator derivatives and dropout placement affect the backward map; Appendix F states these conventions. Appendix D gives the parameter count and block-product cost.

Function-preserving initialization. Initialize $\boldsymbol{B} = \boldsymbol{0}$, $\boldsymbol{R} = \boldsymbol{I}$, and $m_i = \|\boldsymbol{W}_{0,i,:}\|_2$ for nonzero base rows. Then $\boldsymbol{W}_{\mathrm{eff}} = \boldsymbol{W}_0$, regardless of the initial down-projection. The first residual update is determined by the initialized effective projection through $\nabla_{\boldsymbol{B}}\mathcal{L} = s\boldsymbol{G}\boldsymbol{C}^\top$ for the additive core. Once $\boldsymbol{B}$ becomes nonzero, the normalization Jacobian and rotation coupling contribute to subsequent updates.

Resetting a rotation. Absorbing a rotation into the frozen base between training segments requires the following coordinate change to preserve the FRET-C function:

$$\boldsymbol{W}_0^+ = \boldsymbol{W}_0\boldsymbol{R}^\top, \qquad \boldsymbol{A}^+ = \boldsymbol{A}\boldsymbol{R}^\top, \qquad \boldsymbol{R}^+ = \boldsymbol{I}. \tag{28}$$

With $\boldsymbol{B}$ and $\boldsymbol{m}$ unchanged, $\mathcal{N}_r(\boldsymbol{A}^+) = \boldsymbol{C}_{\mathrm{FRET}}$, so both the additive core and the row-scaled dense weight are preserved. The reference remains the original layer-input channels. In FRET-D, the independent raw projection $\boldsymbol{Z}$ stays fixed while the base absorbs the rotation. A reset's training-state transition also specifies the treatment of optimizer moments. First moments transform linearly; a full second-moment matrix transforms on both axes. Elementwise Adam retains its diagonal, making the post-reset state a coordinate-dependent choice. A reset schedule and its state transition together define a training recipe. Appendix F.1 distinguishes that label from the constraint.

Input statistics and the constraint. Input-coordinate normalization separates marginal channel energy from cross-channel interactions. For a projection Gram matrix $\boldsymbol{G} = \boldsymbol{C}^\top\boldsymbol{C}$ with unit diagonal and input second moment $\boldsymbol{\Sigma}_x = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$,

$$\mathbb{E}\|\boldsymbol{C}\boldsymbol{x}\|_2^2 = \mathrm{tr}(\boldsymbol{\Sigma}_x) + \sum_{j\neq k} G_{jk}(\Sigma_x)_{jk}. \tag{29}$$

Zero cross-moments eliminate the second term. With correlated inputs it is bounded in magnitude by $\|\,\mathrm{offdiag}(\boldsymbol{G})\|_F\|\,\mathrm{offdiag}(\boldsymbol{\Sigma}_x)\|_F$. Thus coordinate normalization fixes the diagonal contribution while leaving correlation-dependent responses available to the task. For diagonal $\boldsymbol{\Sigma}_x$, the total projected energy equals the input energy even when the channel variances differ. More generally, the off-diagonal term measures the interaction between input correlations and projection correlations. Appendix E extends the construction to any fixed orthogonal reference basis.

4 Experiments

4.1 Setup and reporting

We compare seven recipes that share the base model, training data, optimizer, rank, residual scale, and normalization floor, and differ only in the layer components of Appendix F.1: the frozen base (Base); matched-cost DoRA without rotation (M); a learned rotation with reset and an unnormalized projection (Rot-reset); a normalized projection with the rotation frozen at identity (Norm-noR); moving-frame normalization with a learned rotation (Naive); and FRET-C and FRET-D. Every trained recipe uses three paired training seeds that share data order, evaluation questions, and decoding budget.

Evaluation uses six benchmarks with N questions and K samples per question: AIME24 (N=30, K=32), AIME25 (30, 32), AMC23 (40, 32), HMMT25 (30, 32), MATH-500 (500, 4), and Minerva (272, 4). Avg. is the fraction of correct samples among the NK generations; Pass is the fraction of questions solved by at least one sample. The macro average weights the six Avg. values equally,

$$\bar{a} = \frac{1}{6}\sum_{b\in\mathcal{B}} a_b. \tag{30}$$

Tables report the mean and sample standard deviation over seeds. Because the seeds are paired, contrasts between recipes are computed per seed and summarized by their mean and spread. Appendix C lists all Avg. and Pass values and the per-seed counts behind them.

4.2 Benchmark results

FRET-C reaches a macro average of 40.70 ± 0.86, FRET-D 40.58 ± 0.92, and Naive 40.27 ± 0.82 (Table 3). The paired FRET-C$-$Naive difference is $+0.42$, $+0.39$, and $+0.48$ points in the three seeds (0.43 ± 0.05), and none of the eighteen benchmark–seed differences is negative (Figure 3a). FRET-D$-$Naive is 0.31 ± 0.18, positive in every seed. The two realizations differ by 0.12 ± 0.20 with mixed sign across seeds: at three seeds the coordinates within the fixed-reference constraint are not distinguishable, whereas the constraint placement is.

The remaining contrasts order the design choices (Figure 3b). Adding a learned rotation to the normalized projection contributes 0.83 ± 0.28 points (Naive$-$Norm-noR), and adding normalization

Method	AIME24	AIME25	AMC23	HMMT25	MATH-500	Minerva	Macro
Base	20.0±1.3	20.0±3.9	64.4±3.2	7.2±1.2	80.8±0.1	26.7±1.2	36.52±0.98
M (matched-cost)	21.6±1.0	22.3±4.3	66.5±3.6	8.4±0.8	83.7±0.4	28.0±0.9	38.43±1.08
Rot-reset	22.8±1.5	23.0±4.6	68.9±1.5	9.7±0.9	84.8±0.5	28.6±0.9	39.64±0.85
Norm-noR	22.7±1.6	23.0±4.6	68.8±1.5	9.0±1.4	84.4±0.4	28.6±0.9	39.44±0.90
Naive (moving frame)	23.4±1.7	23.4±4.4	70.6±2.3	9.7±0.9	85.7±0.6	28.9±0.9	40.27±0.82
FRET-D (fixed frame)	**24.4**±1.0	23.4±4.5	71.2±2.4	**9.8**±1.0	85.7±0.6	29.0±0.9	40.58±0.92
FRET-C (fixed frame)	23.5±1.6	**23.8**±4.4	**71.5**±2.4	**9.8**±1.0	**86.4**±0.5	**29.2**±0.9	**40.70**±0.86

Table 3: Benchmark Avg. (%), mean±standard deviation over three paired training seeds. Macro equally weights the six benchmarks. Bold marks the highest mean among trained recipes in each column.

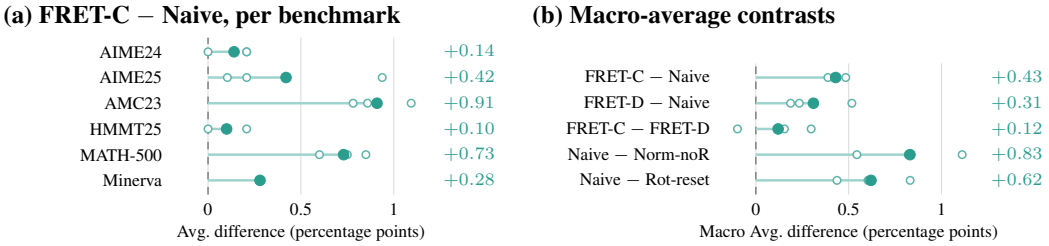


Figure 3: **Paired differences over three training seeds.** Hollow circles are individual seeds; filled circles are seed means (values at right). (a) FRET-C minus Naive Avg. on each benchmark: no benchmark–seed difference is negative. (b) Macro-average contrasts isolating constraint placement (FRET-C and FRET-D minus Naive), coordinates within the constraint (FRET-C minus FRET-D), and the rotation and normalization components (Naive minus Norm-noR and Naive minus Rot-reset).

to the rotated layer contributes 0.62 ± 0.20 (Naive−Rot-reset), and the fixed-reference constraint adds 0.43 on top of their combination. These gains are small in absolute terms. FRET-C and Naive differ by 33 to 38 correct samples per seed out of $7{,}248$, solve the same number of questions on every 30- and 40-question benchmark in every seed, and differ in Pass only on MATH-500 and Minerva, by at most two questions.

4.3 Mechanism check and scope

The analytic example in Figure 4 evaluates the projection itself: at $45°$, Naive has effective Gram diagonal $(1.6, 0.4)$ and FRET-C $(1, 1)$, so the diagonal error of Equation (34) is 0.6 and 0. Connecting this identity to the trained layers requires three checkpoint measurements that the present runs do not record: within-block correlation energy $\sum_{j \neq k} H_{jk}^2$, rotation displacement $\|\boldsymbol{R}_b - \boldsymbol{I}\|_2$, and realized gain error $\| \operatorname{diag}(\boldsymbol{C}^\top \boldsymbol{C}) - \mathbf{1} \|_\infty$. A frozen random reference basis (Appendix E) remains the untested control for the choice of axes.

The evidence is a consistent but small margin: three seeds bound the spread of a 0.4-point effect only coarsely, four benchmarks contain 30–40 questions, and the cost of the two block products is not measured. The comparison orders the design choices under one shared recipe; it does not yet attribute the margin to the gain mechanism of Section 3.

5 Conclusion

FRET fixes the column-norm constraint in the layer's input coordinates. Its coupled realization is the nearest feasible factor under a learned rotation; its decoupled realization expresses the same constrained layer through independent projection coordinates. The analysis identifies column correlations as the source of gain redistribution and shows how coupled normalization removes its radial component at identity. Both realizations support compact training and dense export. Over three paired seeds on six mathematical-reasoning benchmarks, both realizations exceed moving-frame normalization—FRET-C by 0.43 ± 0.05 macro points, with no negative benchmark–seed difference—while differing from each other within seed variation.

AI USE STATEMENT

Generative AI tools assisted with drafting, literature discovery, automated self-review, LaTeX preparation, diagram source code, and mathematical and code–manuscript checks. Quantitative plots use the stated aggregates or analytic formulas. Section 4 describes the experimental data.

REFERENCES

Massimo Bini, Karsten Roth, Zeynep Akata, and Anna Khoreva. ETHER: Efficient finetuning of large-scale models with hyperplane reflections. In *ICML*, 2024.

Mikhail Gorbunov, Nikolay Yudin, Vera Soboleva, Aibek Alanov, Alexey Naumov, and Maxim Rakhuba. Group and shuffle: Efficient structured orthogonal parametrization. In *NeurIPS*, 2024.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Jiale Kang, Ziyin Yue, Zheng Zhan, Yangyi Huang, and Weiyang Liu. Normalized low-rank adaptation (NoRA). *arXiv preprint arXiv:2608.31036*, 2026.

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. *arXiv preprint arXiv:2402.09353*, 2024a.

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*, 2024b.

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.

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

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

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

Kexuan Shi, Hanxuan Li, Zeju Qiu, Yandong Wen, Simon Buchholz, and Weiyang Liu. Pion: A spectrum-preserving optimizer via orthogonal equivalence transformation. *arXiv preprint arXiv:2605.12492*, 2026.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *COLM*, 2024. URL https://arxiv.org/abs/2402.17762.

YongShun Wang, JianLin Su, and Yong Ma. Bilinear optimization divergence: Diagnosing factor-constrained LoRA continual learning. *arXiv preprint arXiv:2609.23594*, 2026. URL https://arxiv.org/abs/2609.23594.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S. Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.20625.

Qingyu Yin, Yulun Wu, Zhennan Shen, Sunbowen Li, Zhilin Wang, Yanshu Li, Chak Tou Leong, Jiale Kang, and Jinjin Gu. Evaluating parameter efficient methods for RLVR. *arXiv preprint arXiv:2512.23165*, 2025.

Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. In *NeurIPS*, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019. URL https://arxiv.org/abs/1910.07467.

Jiaying Zhang, Lei Shi, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, and Renqing He. GeoRA: Geometry-aware low-rank adaptation for RLVR. In *ACL*, 2026. URL https://arxiv.org/abs/2601.09361.

A Numerical Error and Verification

Extreme redistribution. For the block in Proposition 2, each gain lies in $[0, b]$ and the gains sum to b. The pattern $(b, 0, \ldots, 0)$ is attainable: take $\boldsymbol{U}_b = \boldsymbol{u}\mathbf{1}^\top$ with $\|\boldsymbol{u}\|_2 = 1$ and choose a rotation mapping $\mathbf{1}/\sqrt{b}$ to the first coordinate. This establishes the possible range for a correlated projection.

Numerical checks. The accompanying tests check the nearest-factor characterization, signed-permutation preservation, normalization floors, the inexact-rotation bound, deterministic row-rescaled folding, and identity-rotation derivatives. They also verify the radial coefficient in Equation (20) against a finite-difference derivative. A separate Monte Carlo check compares the Haar expectation with Equation (13).

B Reproducibility Protocol

Section 4 reports three paired training seeds per recipe. The following fields identify each run:

- Model revision, dataset and split hashes, source-code and dependency versions, adapter ranks, scales, rotation blocks, floor, and reset rule.
- Optimizer, scheduler, KL, clipping, dropout, batch construction, accumulation, training seed, evaluation seed, and checkpoint-selection rule.
- Question and rollout identifiers, prompts, evaluator, answer extraction, sample counts, truncation policy, and completeness checks.

Parameter count, processed tokens, and measured GPU time are separate cost-matching criteria.

Paired training seeds share data order, evaluation questions, and decoding budget. Development data determine checkpoint selection. Selected and final checkpoints are evaluated separately, and each run retains its completion or failure status. Paired-seed spread measures training variability; question resampling estimates evaluation variability. The contrasts in Section 4 isolate constraint placement (FRET-C and FRET-D versus Naive), coordinates within the constraint (FRET-C versus FRET-D), and the rotation and normalization components (Naive versus Norm-noR and Rot-reset); a fixed random reference basis remains the untested control.

Engineering validation compares a complete accumulated optimizer update with an explicit reference and checks forward and backward equivalence of optimized kernels, merge/unmerge, and sampler weight synchronization. The test suite includes zero and near-zero columns, numerical precision, and checkpoint restoration.

C Additional Results

Table 4 reports Avg. and Pass for all seven recipes as mean and standard deviation over the three seeds, and Table 5 lists the per-seed counts of correct samples and solved questions from which every reported value is computed.

D Numerical Realization and Cost

With the max floor, the squared gain is one for $q_j \geq \epsilon$ and $(q_j/\epsilon)^2$ below it. For the additive denominator $q_j + \epsilon$, the squared-gain deficit satisfies

$$1 - \left(\frac{q_j}{q_j + \epsilon}\right)^2 = \frac{2q_j\epsilon + \epsilon^2}{(q_j + \epsilon)^2} \leq \frac{2\epsilon}{q_j + \epsilon}. \tag{31}$$

Zero columns remain zero under either convention.

Floor-dependent derivatives. Above the max floor, the Jacobian is Equation (18); below it, the map is $\boldsymbol{z}/\epsilon$ with Jacobian $\boldsymbol{I}/\epsilon$. At $q = \epsilon$, the one-sided radial derivatives differ, so the autodifferentiation boundary convention must be specified. For an additive denominator and $q > 0$,

$$\boldsymbol{J}_+(\boldsymbol{z}) = \frac{\boldsymbol{I}}{q + \epsilon} - \frac{\boldsymbol{z}\boldsymbol{z}^\top}{q(q + \epsilon)^2}. \tag{32}$$

Recipe	AIME24@32 ($N = 30$) Avg.	AIME24@32 ($N = 30$) Pass	AIME25@32 ($N = 30$) Avg.	AIME25@32 ($N = 30$) Pass	AMC23@32 ($N = 40$) Avg.	AMC23@32 ($N = 40$) Pass
Base	20.0±1.3	52.2±3.8	20.0±3.9	53.3±8.8	64.4±3.2	90.8±2.9
M	21.6±1.0	53.3±3.3	22.3±4.3	55.6±10.2	66.5±3.6	91.7±3.8
Rot-reset	22.8±1.5	54.4±5.1	23.0±4.6	55.6±10.2	68.9±1.5	93.3±1.4
Norm-noR	22.7±1.6	54.4±5.1	23.0±4.6	55.6±10.2	68.8±1.5	93.3±1.4
Naive	23.4±1.7	54.4±5.1	23.4±4.4	55.6±10.2	70.6±2.3	94.2±2.9
FRET-D	24.4±1.0	55.6±3.8	23.4±4.5	55.6±10.2	71.2±2.4	94.2±2.9
FRET-C	23.5±1.6	54.4±5.1	23.8±4.4	55.6±10.2	71.5±2.4	94.2±2.9

Recipe	HMMT25@32 ($N = 30$) Avg.	HMMT25@32 ($N = 30$) Pass	MATH-500@4 ($N = 500$) Avg.	MATH-500@4 ($N = 500$) Pass	Minerva@4 ($N = 272$) Avg.	Minerva@4 ($N = 272$) Pass
Base	7.2±1.2	22.2±5.1	80.8±0.1	96.1±0.2	26.7±1.2	47.2±1.5
M	8.4±0.8	24.4±3.8	83.7±0.4	97.1±0.5	28.0±0.9	48.0±1.1
Rot-reset	9.7±0.9	26.7±3.3	84.8±0.5	97.5±0.6	28.6±0.9	48.7±0.8
Norm-noR	9.0±1.4	25.6±5.1	84.4±0.4	97.5±0.6	28.6±0.9	48.7±0.8
Naive	9.7±0.9	26.7±3.3	85.7±0.6	97.7±0.5	28.9±0.9	48.8±1.1
FRET-D	9.8±1.0	26.7±3.3	85.7±0.6	97.7±0.5	29.0±0.9	48.8±1.1
FRET-C	9.8±1.0	26.7±3.3	86.4±0.5	98.0±0.3	29.2±0.9	49.0±1.2

Table 4: Avg. and Pass (%) for all recipes, mean±standard deviation over three paired seeds. Avg. is the fraction of correct samples among the NK generations; Pass is the fraction of the N questions solved by at least one of the K samples. The macro analysis uses Avg.

Recipe	Seed	AIME24 $N = 30, K = 32$	AIME25 $N = 30, K = 32$	AMC23 $N = 40, K = 32$	HMMT25 $N = 30, K = 32$	MATH-500 $N = 500, K = 4$	Minerva $N = 272, K = 4$
Base	1	185/15	235/19	856/37	78/8	1617/480	276/127
	2	184/15	167/14	778/35	73/7	1615/482	299/133
	3	206/17	175/15	838/37	56/5	1617/480	298/125
M	1	200/15	260/20	874/37	84/8	1668/483	293/129
	2	204/16	179/14	798/35	87/8	1673/487	307/134
	3	219/17	203/16	881/38	72/6	1683/487	313/129
Rot-reset	1	211/15	270/20	892/37	102/9	1685/484	301/131
	2	210/16	186/14	860/37	94/8	1704/489	314/135
	3	236/18	207/16	893/38	84/7	1699/490	319/131
Norm-noR	1	211/15	270/20	891/37	99/9	1680/484	301/131
	2	208/16	186/14	859/37	89/8	1694/488	314/135
	3	236/18	207/16	892/38	72/6	1692/490	319/131
Naive	1	213/15	272/20	907/37	102/9	1702/486	303/131
	2	218/16	191/14	873/37	92/8	1724/490	319/136
	3	243/18	210/16	931/39	84/7	1713/490	322/131
FRET-D	1	233/16	273/20	915/37	103/9	1702/486	305/131
	2	225/16	191/14	879/37	94/8	1724/490	319/136
	3	245/18	211/16	939/39	84/7	1713/490	324/131
FRET-C	1	215/15	274/20	918/37	103/9	1719/488	306/132
	2	220/16	192/14	883/37	94/8	1739/491	322/137
	3	243/18	219/16	945/39	84/7	1725/491	325/131

Table 5: Per-seed counts of correct samples / solved questions. Each entry lists the number of correct generations among NK and the number of questions with at least one correct generation among N; Avg. and Pass in Tables 3 and 4 are computed from these counts.

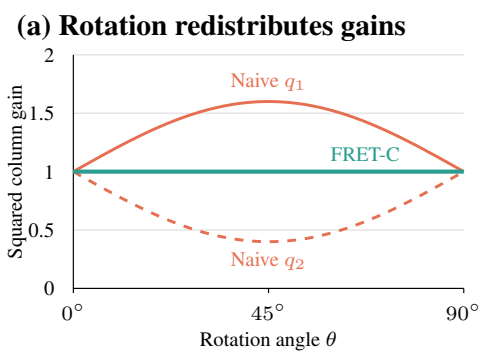
Its radial eigenvalue is $\epsilon/(q+\epsilon)^2$; each tangential eigenvalue is $1/(q+\epsilon)$. At zero, its derivative exists and equals $\boldsymbol{I}/\epsilon$. These expressions specify how the two floors handle small columns and where finite-difference tests encounter a kink.

Inexact rotations. If the implemented rotation is $\widetilde{\boldsymbol{R}}$, define $\boldsymbol{C}_{\mathrm{FRET}} = \mathcal{N}_r(\boldsymbol{A}\widetilde{\boldsymbol{R}}^\top)$. The returned factor produces $\boldsymbol{C}_{\mathrm{FRET}}\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top$, so

$$\|\boldsymbol{C}_{\mathrm{FRET}}\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top - \boldsymbol{C}_{\mathrm{FRET}}\|_F \le \|\boldsymbol{C}_{\mathrm{FRET}}\|_F \|\widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top - \boldsymbol{I}\|_2. \tag{33}$$

This follows from $\|\boldsymbol{X}\boldsymbol{Y}\|_F \le \|\boldsymbol{X}\|_F\|\boldsymbol{Y}\|_2$ with $\boldsymbol{X} = \boldsymbol{C}_{\mathrm{FRET}}$ and $\boldsymbol{Y} = \widetilde{\boldsymbol{R}}\widetilde{\boldsymbol{R}}^\top - \boldsymbol{I}$. The orthogonality residual bounds the return-path discrepancy and complements the floor deficit and dense-export error.

For equal rotation blocks of size b, the explicit construction uses two $O(rdb)$ block products and $O(rd)$ column normalization. Its trainable arrays match those of the moving-frame layer. With independent skew-symmetric block generators and one magnitude per output row, both contain $r(d + d_{\mathrm{out}}) + d_{\mathrm{out}} + d(b-1)/2$ trainable scalars, excluding biases. Block sharing changes the



(b) Effective gains at $45°$

$$\boldsymbol{A} = \begin{pmatrix} 2 & 0.3 \\ 0 & 0.4 \end{pmatrix}, \quad \rho = 0.6$$

FRET-C $\quad \operatorname{diag}(\boldsymbol{C}^\top \boldsymbol{C}) = (1,\ 1)$

Naive $\quad \operatorname{diag}(\boldsymbol{C}^\top \boldsymbol{C}) = (1.6,\ 0.4)$

Both traces equal 2.
The fixed-frame constraint sets each gain.

Figure 4: **Correlation becomes gain drift.** For the raw factor in (b), normalized columns have correlation 0.6. Naive's gains follow $1 \pm 0.6 \sin(2\theta)$; FRET-C maintains unit gains under ideal normalization. The example separates redistribution between coordinates from preservation of their total energy.

last term. Training-time comparisons must additionally account for backward kernels, row-norm computation, communication, and rollout lengths.

The effective-projection check is

$$\mathcal{E}_{\text{coord}}(\boldsymbol{C}) = \| \operatorname{diag}(\boldsymbol{C}^\top \boldsymbol{C}) - \mathbf{1} \|_\infty. \tag{34}$$

This verifies the gain constraint subject to the floor. With input second moment $\boldsymbol{\Sigma}_x$, the expected latent response is $\operatorname{tr}(\boldsymbol{C}^\top \boldsymbol{C} \boldsymbol{\Sigma}_x)$, which also depends on off-diagonal correlations. The full network's response further depends on $\boldsymbol{B}$ and subsequent layers.

E Alternative Reference Bases

For any orthogonal $\boldsymbol{Q}$, define

$$\mathcal{S}_{\boldsymbol{Q}} = \{\boldsymbol{C} : \operatorname{diag}(\boldsymbol{Q}^\top \boldsymbol{C}^\top \boldsymbol{C} \boldsymbol{Q}) = \mathbf{1}\}. \tag{35}$$

The projection then has unit response to each basis vector $\boldsymbol{Q}\boldsymbol{e}_j$. For ideal normalization, the map

$$\boldsymbol{C}_{\boldsymbol{Q}} = \mathcal{N}_r(\boldsymbol{A}\boldsymbol{R}^\top \boldsymbol{Q}) \boldsymbol{Q}^\top, \qquad \boldsymbol{F}_{\boldsymbol{Q}} = \boldsymbol{C}_{\boldsymbol{Q}} \boldsymbol{R} \tag{36}$$

satisfies this constraint. Choosing $\boldsymbol{Q} = \boldsymbol{I}$ gives FRET; choosing the moving $\boldsymbol{Q} = \boldsymbol{R}$ gives Naive. A frozen random $\boldsymbol{Q}$ tests the selection of the fixed reference frame. Dense reference matrices add computation that should be accounted for in this control.

Equation (29) applies in basis $\boldsymbol{Q}$ after replacing the input second moment by $\boldsymbol{Q}^\top \boldsymbol{\Sigma}_x \boldsymbol{Q}$ and the projection by $\boldsymbol{C}\boldsymbol{Q}$.

F Training Conventions

Implementation conventions. The accompanying implementation divides each rank-dimensional column by $q + 10^{-6}$, accumulating its norm in FP32. Its rotation module uses independent Cayley blocks, configurable block width (default 32), zero-initialized skew parameters, and a direct FP32 linear solve. These describe code defaults.

Differentiating the row denominator in $\Gamma_{\boldsymbol{m}}$ and detaching it give identical forward values with different gradients. A paired comparison holds this choice fixed. The accompanying layer applies low-rank dropout after the input rotation. A channelwise mask acts in that rotated frame, so its placement is part of the training definition. Dense export uses dropout-disabled computation.

Reset and checkpoint state. Equation (28) preserves the FRET-C forward map. A reproducible reset specifies its trigger, the base and factor updates, and the treatment of optimizer moments, scheduler, and random-number state. A complete training checkpoint stores these states alongside the raw factors. A dense export stores the inference operator.

Storage orientation. For implementations storing transposed weights, the operators and normalization axes transpose together. The residual scale s is shared between training and export, and both appearances of the learned rotation remain differentiable. Finite-precision verification compares the unmerged and exported forward values under the target dtype.

F.1 Recipe labels and mathematical controls

Table 6 lists the layer components of the seven compared recipes and of the remaining untested control. The shared output-row map is $\Gamma_{\boldsymbol{m}}$ from Equation (25). The label M denotes the matched-cost DoRA baseline; Rot-reset absorbs its learned rotation into the base according to its reset schedule; Norm-noR freezes the rotation at identity and is the frozen-identity control of Table 1.

Label	Base rotation	Projection	Reference
Compared recipes			
Base	Identity	None	—
M	Identity	Raw low-rank	Free
Rot-reset	Learned	Raw low-rank	Free
Norm-noR	Identity	Normalized	Fixed input
Naive	Learned	Factor normalized	Moving
FRET-D	Learned	Independently normalized	Fixed input
FRET-C	Learned	Coupled normalized	Fixed input
Untested control			
Fixed basis	Learned	Normalized in $\boldsymbol{Q}$	Fixed $\boldsymbol{Q}$

Table 6: Layer components of the compared recipes. Norm-noR is the frozen-identity control; the fixed random basis of Appendix E is the only control without scores.

G Normalization Derivatives

For $q = \|\boldsymbol{z}\|_2 > 0$ and $\boldsymbol{u} = \boldsymbol{z}/q$, differentiation gives $d\boldsymbol{u} = (\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^\top)d\boldsymbol{z}/q$, proving Equation (18).

For an additive denominator, the quotient rule yields Equation (32). Applying this matrix to $\boldsymbol{z}$ and to any vector orthogonal to $\boldsymbol{z}$ gives the radial and tangential eigenvalues in Section D. At zero, the remainder after subtracting $\boldsymbol{z}/\epsilon$ has norm $q^2/[\epsilon(q+\epsilon)] = o(q)$, establishing differentiability.

H Random-Rotation Reference Calculation

For Proposition 2, a row $\boldsymbol{v}^\top$ of a Haar-uniform orthogonal matrix is uniform on the sphere, with fourth moment

$$\mathbb{E}[v_i v_j v_k v_\ell] = \frac{\delta_{ij}\delta_{k\ell} + \delta_{ik}\delta_{j\ell} + \delta_{i\ell}\delta_{jk}}{b(b+2)}. \tag{37}$$

Contracting with two copies of $\boldsymbol{H}$ gives

$$\mathbb{E}[(\boldsymbol{v}^\top \boldsymbol{H}\boldsymbol{v})^2] = \frac{(\operatorname{tr}\boldsymbol{H})^2 + 2\operatorname{tr}(\boldsymbol{H}^2)}{b(b+2)}. \tag{38}$$

Since $\operatorname{tr}\boldsymbol{H} = b$ and $\mathbb{E}[\boldsymbol{v}^\top\boldsymbol{H}\boldsymbol{v}] = 1$, subtracting one yields Equation (13). At most r eigenvalues of $\boldsymbol{H}$ are nonzero, and their sum is b, so Cauchy–Schwarz gives $\|\boldsymbol{H}\|_F^2 \geq b^2/r$. Substituting proves the stated lower bound when $r < b$. The expectation is over rotations independent of $\boldsymbol{H}$.