NEURAL INFORMATION PROCESSING SYSTEMS

SONY

Making LoRA Identifiable: Orthogonal Alignment for Continual learning

Pratik Rakesh Singh, Mohammadi Zaki, Akash Saha, Aneesh Mukkamala, Pankaj Wasnik

Motivation

One LoRA per task



Memory grows with every task: $O(T(m + n)r)$

IncLoRA, O-LoRA and gated variants extend or route across task-specific modules.

One shared LoRA (this work)



Memory stays constant: $O((m + n)r)$

No replay data and no stored task adapters. Every new task is merged into the same adapter.

The recipe

Fine-tune on task t
get new factors B, A

›

Merge into the running adapter
one LoRA after every task

›

Repeat for the next task
old tasks must survive

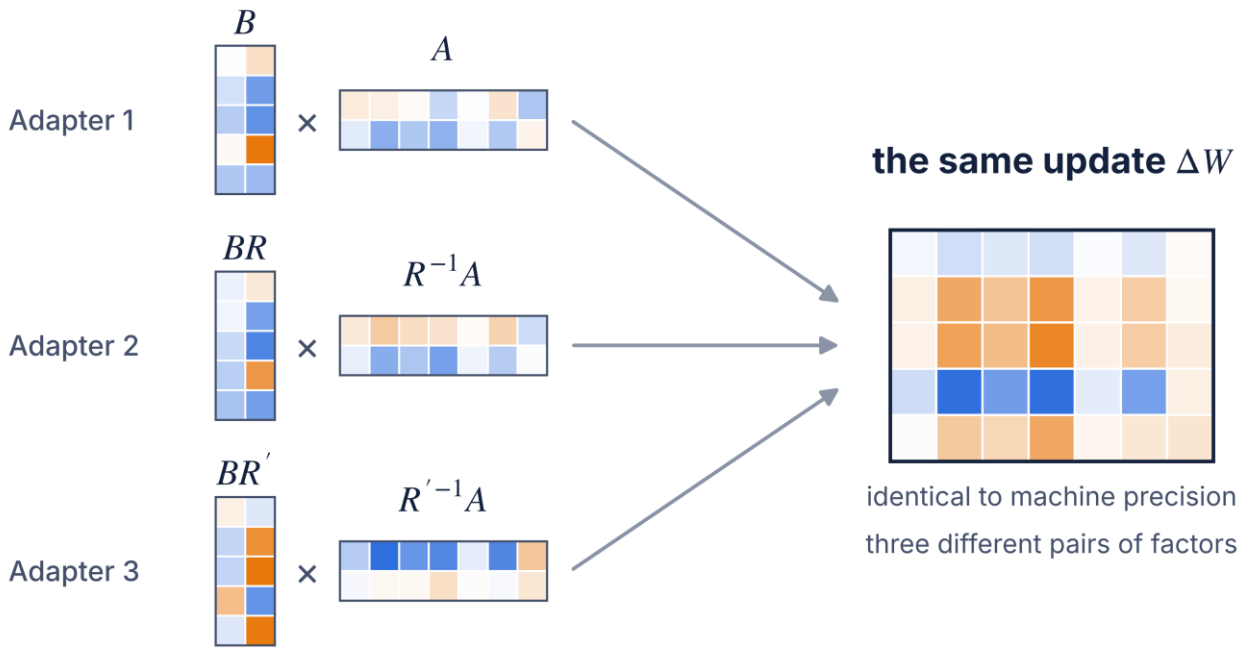
The merge step is where things go wrong.

Problem

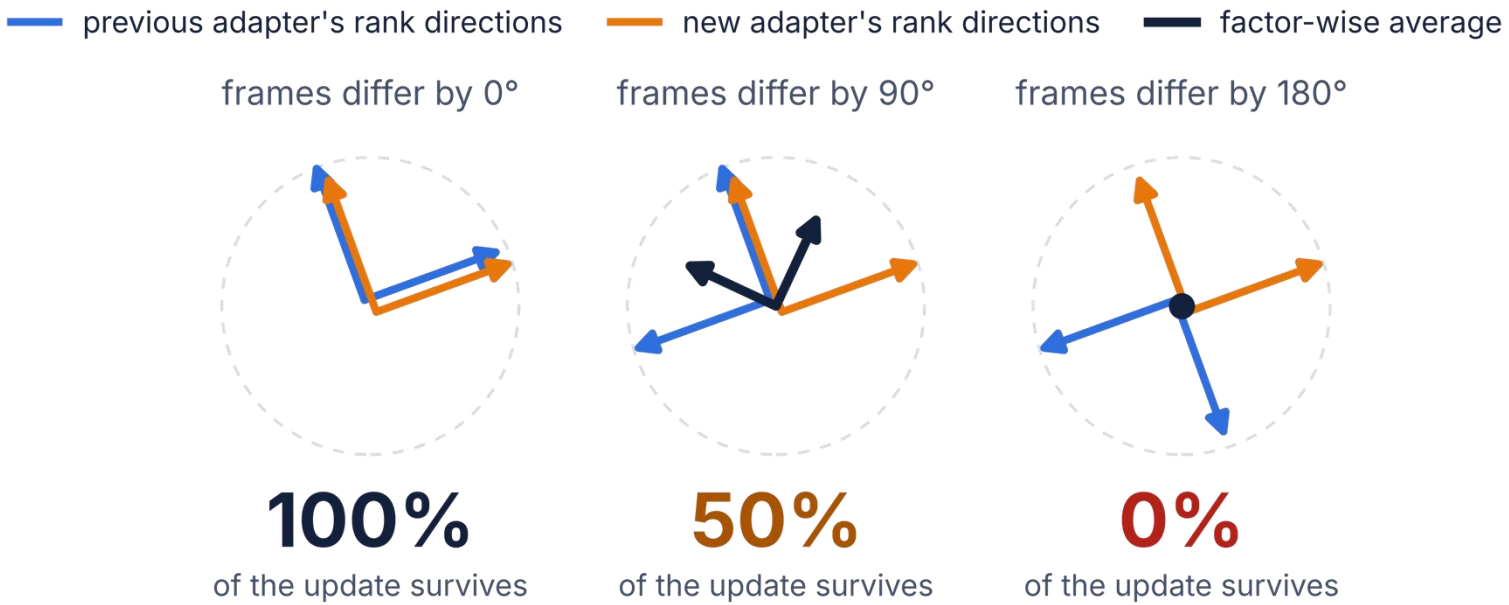
$$\Delta W = BA = (BR)(R^{-1}A)$$

for every invertible $R \in \mathrm{GL}(r)$

- Infinitely many factor pairs give the same update.
- Nothing in training picks a canonical one.
- Adapters from consecutive tasks can sit in **unrelated rank coordinates**.
- Merging A with A, or B with B, then mixes directions that are not comparable.



Naive merging problem



TOY EXAMPLE, r = 2

- Take one adapter.
- Write a second copy in a frame rotated by θ. Same function.
- Average the factors.

$$\bar{B}\bar{A} = \tfrac{1}{2}(1 + \cos\theta) \cdot \Delta W$$

What is lost has nothing to do with the tasks.

Interference in a merged LoRA can come from mismatched coordinates, not from real conflict between tasks.

SOAR: fix the coordinates, then merge

Standard LoRA

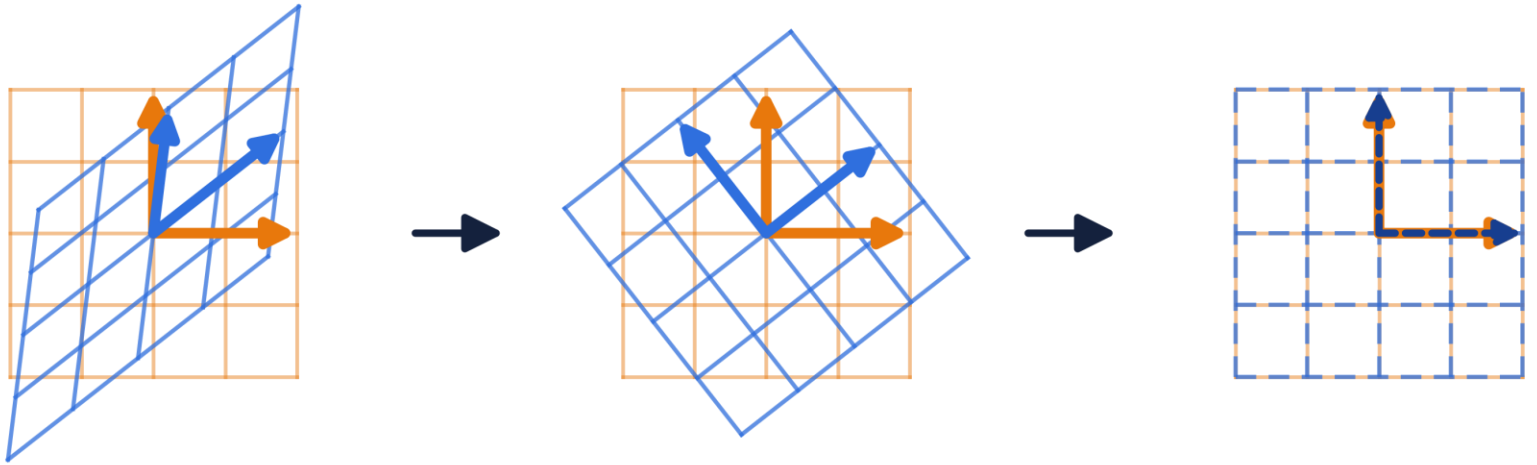
$R \in \mathrm{GL}(r)$

A on the Stiefel manifold

$R \in \mathrm{O}(r)$

After Procrustes

$R^* = UV^\top$



rotation + scaling + shear

rotation only

frames coincide

rank coordinates of the previous adapter

rank coordinates of the new adapter

Step 1 Constrain A: only a rotation is left

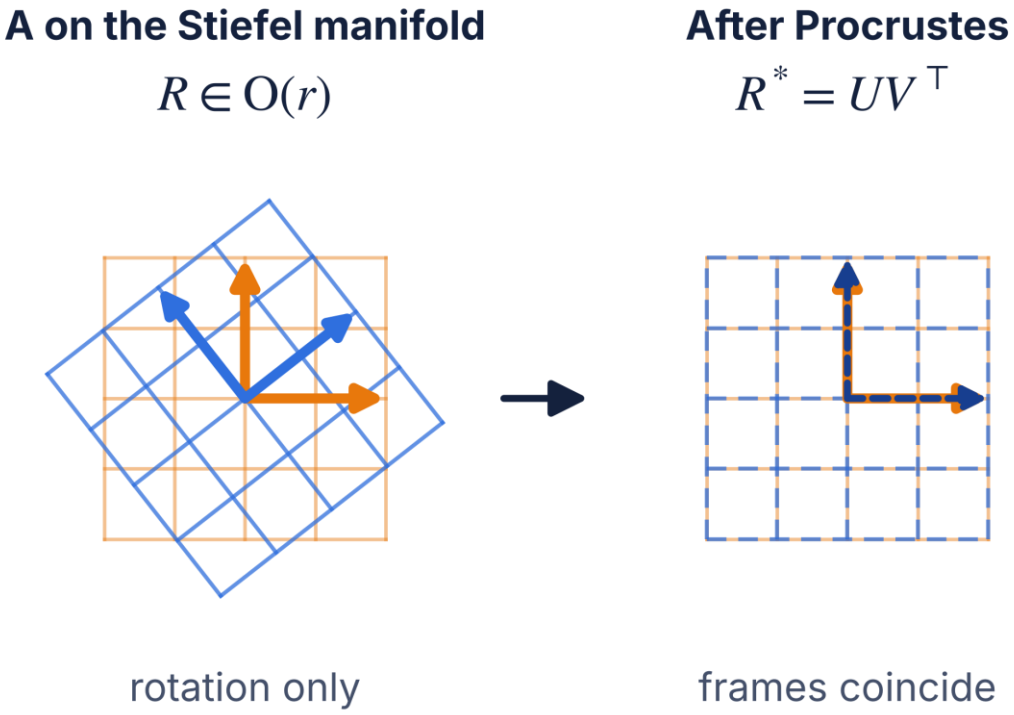
Step 2 Procrustes: undo that rotation

Then Merge B one rank direction at a time

Step 2: rotate the old B onto the new B

$$R^* = \operatorname{argmin}_{R \in O(r)} \|B_{new} - B_{old} R^T\|_F^2 = UV^T$$

- Closed form: $B_{new}{}^T B_{old} = U\Sigma V^T$, one r × r SVD per task.
- Aligned factor: $\tilde{B}_{old} = B_{old} R^{*T}$, now in the new adapter's frame.
- No hyperparameters, negligible cost at r = 32.
- **A is not merged.** We keep the newly trained A; all consolidation happens in B.



After this step, column i of the old B and column i of the new B describe the same rank coordinate.

Then decide, per rank direction, how much to keep

DIRECTIONAL AGREEMENT

$$c_i = \cos(b_{\text{new,i}}, \tilde{b}_{\text{old,i}})$$

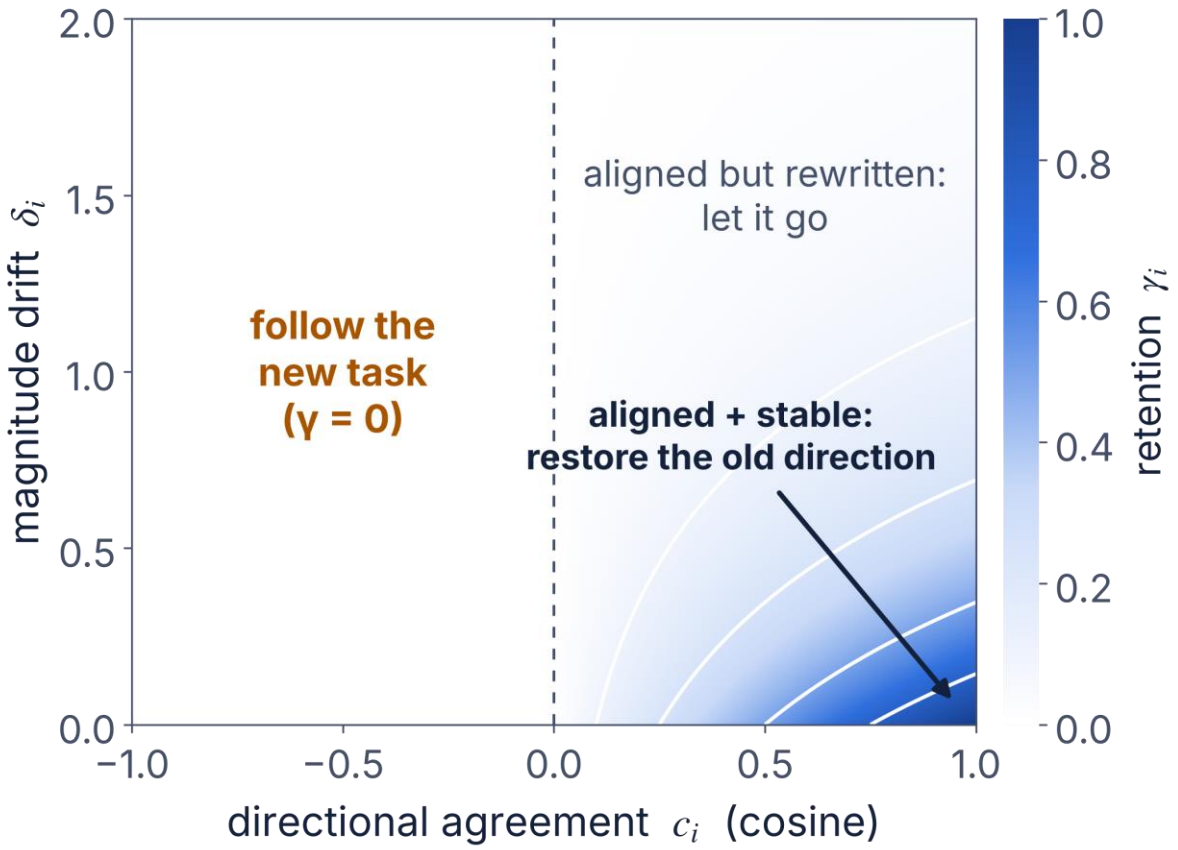
MAGNITUDE DRIFT

$$\delta_i = \|b_{\text{new,i}} - \tilde{b}_{\text{old,i}}\|^2 / \|\tilde{b}_{\text{old,i}}\|^2$$

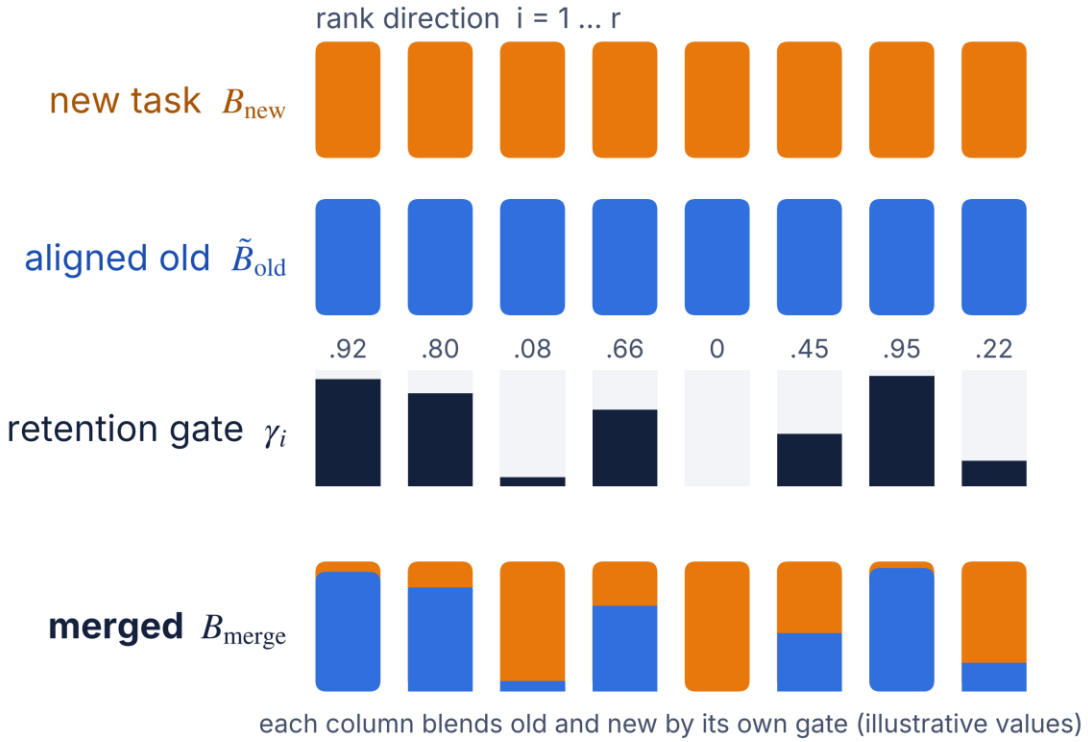
RETENTION GATE

$$\gamma_i = \max(c_i, 0) \cdot \exp(-\delta_i / \tau)$$

Both signals are measured after alignment, so they reflect the tasks rather than the coordinates.



The merge keeps what still agrees



$$b_{merge,i} = (1 - \gamma_i)\, b_{new,i} + \gamma_i\, \tilde{b}_{old,i}$$

- **γ ≈ 0:** weak evidence, keep the new direction as trained.
- **γ ≈ 1:** aligned and stable, restore the old direction.
- Anchored at the new task: plastic by default, stable where it is safe.

For each new task

Train
A on Stiefel, B free

›

Keep the new A
no merging of A

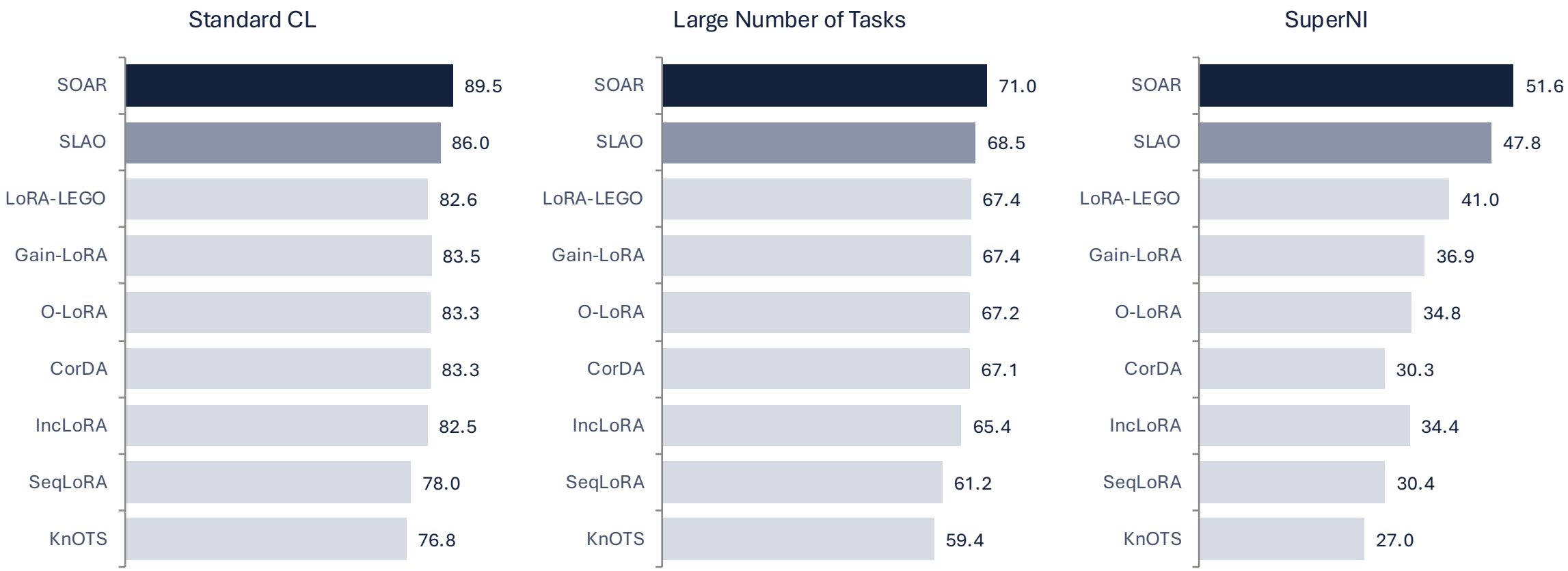
›

Align
old B onto new B

›

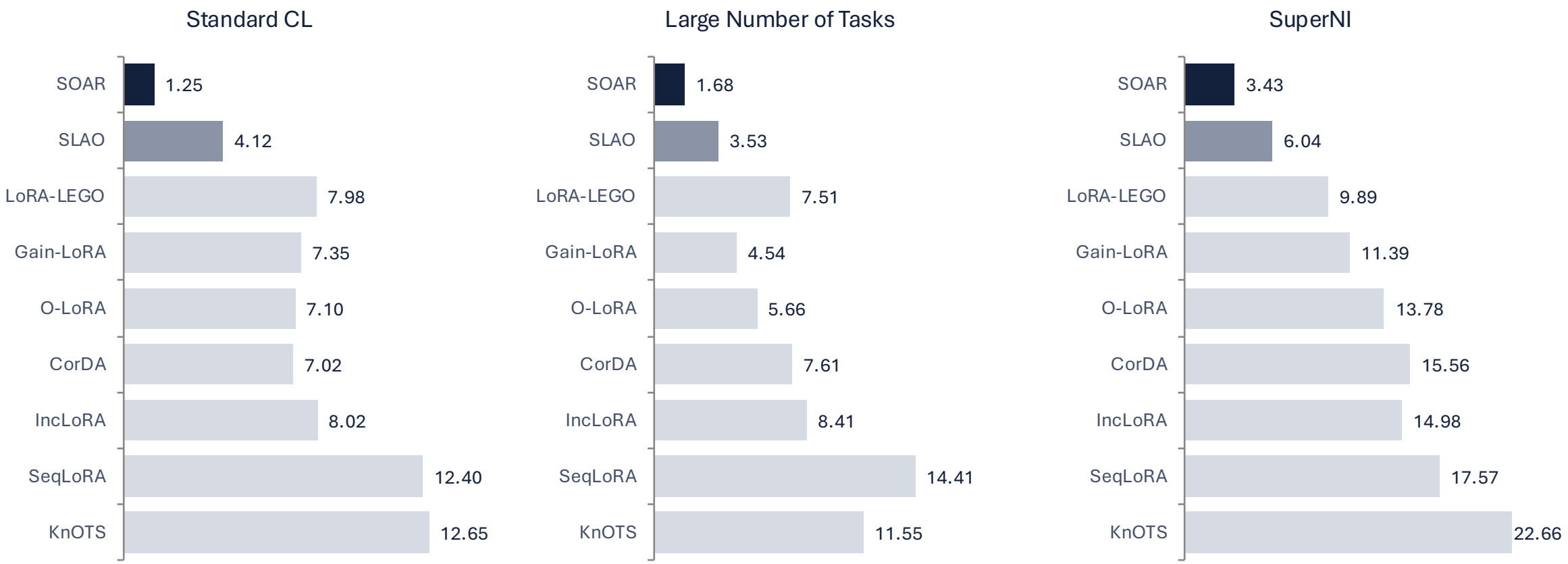
Gate and merge
one γ per rank

SOAR scores highest on all three benchmarks



Over SLAO, the strongest baseline: +3.49 · +2.47 · +3.75 points Final average score (%), Qwen3-8B

And it forgets the least



Less forgetting than SLAO: 70% · 52% · 43% Forgetting = −BWT (points, lower is better), Qwen3-8B

why A is replaced rather than merged



MEAN PRINCIPAL-ANGLE DRIFT OF A

3.50°	O1 · Standard CL
5.04°	O4 · Large # Tasks
12.59°	O7 · SuperNI

Visit us at ***poster session number 5*** *on* ***10th Dec*** *from* ***10AM to 1AM****.*