Accurate and Efficient Low-Rank Model Merging in Core Space

Aniello Panariello[1*] **Daniel Marczak**[2,3 *]

Simone Magistri[4] **Angelo Porrello**[1] **Bartłomiej Twardowski**[5,6]

Andrew D. Bagdanov[4] **Simone Calderara**[1] **Joost van de Weijer**[6]

[1]AImageLab, University of Modena and Reggio Emilia, Italy
[2]Warsaw University of Technology, Poland [3]IDEAS NCBR, Warsaw, Poland
[4]Media Integration and Communication Center (MICC), University of Florence, Italy
[5]IDEAS Research Institute, Warsaw, Poland
[6]Computer Vision Center, Universitat Autònoma de Barcelona, Spain

Abstract

In this paper, we address the challenges associated with merging low-rank adaptations of large neural networks. With the rise of parameter-efficient adaptation techniques, such as Low-Rank Adaptation (LoRA), model fine-tuning has become more accessible. While fine-tuning models with LoRA is highly efficient, existing merging methods often sacrifice this efficiency by merging fully-sized weight matrices. We propose the *Core Space* merging framework, which enables the merging of LoRA-adapted models within a common alignment basis, thereby preserving the efficiency of low-rank adaptation while substantially improving accuracy across tasks. We further provide a formal proof that projection into Core Space ensures no loss of information and provide a complexity analysis showing the efficiency gains. Extensive empirical results demonstrate that Core Space significantly improves existing merging techniques and achieves state-of-the-art results on both vision and language tasks while utilizing a fraction of the computational resources. Codebase is available at `https://github.com/apanariello4/core-space-merging`.

1 Introduction

In recent years, the size of neural networks has grown substantially [2, 4, 9, 13, 51], increasing the economic and computational costs associated with training from scratch and fine-tuning. As a consequence, efficient low-rank adaptation techniques have emerged, which enable broader access to these powerful models [15, 16, 21, 26, 56]. Techniques like Low-Rank Adaptation (LoRA) [16] reparameterize model updates to significantly reduce the number of trainable parameters. This makes it feasible for a broader range of users to fine-tune large architectures on their specific tasks.

At the same time, the advent of model hubs such as Hugging Face [17] has simplified the diffusion of pre-trained and fine-tuned models, opening new opportunities for collaborative and multi-task learning by allowing users to acquire and build upon existing models easily. In this context, model merging, which aims to combine multiple specialized models into one capable of handling various tasks, has been gaining interest [32, 37, 48, 27, 39]. However, most prior works focus on fully fine-tuned models [6, 12, 18, 28, 34, 46, 50, 38]. While this is practical for smaller architectures, fully fine-tuned versions of larger models are rare due to their high memory and compute costs. As model

*Equal Contribution. `aniello.panariello@unimore.it, daniel.marczak.dokt@pw.edu.pl`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-2066

sizes grow, new strategies are needed to efficiently merge fine-tuned adaptations without incurring the prohibitive overhead of merging full models.

In [42], the authors observe that directly applying existing merging techniques [18, 50, 55] to updates derived by multiplying low-rank components leads to suboptimal results. To address this, they introduce an alignment space that improves update compatibility. However, merging in this alignment space requires abandoning the low-rank representation and performing a singular value decomposition (SVD) on the horizontally concatenated full space updates. This suffers from two significant drawbacks: it eliminates the efficiency benefits of low-rank adaptation, and becomes prohibitively expensive as the base model size increases.

To overcome such limitations, we propose *Core Space*, a novel parameter-efficient subspace that supports arbitrary merging techniques while retaining the benefits of low-rank adaptation. Core Space provides a common alignment basis for all task-specific low-rank components without loss of information. Notably, its dimensionality depends solely on the number of tasks and the LoRA rank, remaining tractable regardless of the base model size. Beyond its advantages in terms of efficiency, merging in Core Space also consistently improves the performance of existing merging strategies. We evaluate three setups: (1) ■ first multiplying low-rank matrices and then applying merging techniques, (2) ■ merging in the KnOTS space [42], and (3) ■ merging in our proposed Core Space. Across both vision and language domains, merging in Core Space achieves the best results – demonstrated on ViT-B/32, ViT-L/14, and Llama 3 8B backbones – highlighting that our approach not only preserves parameter efficiency but also leads to improved generalization and task performance (see Fig. 1).

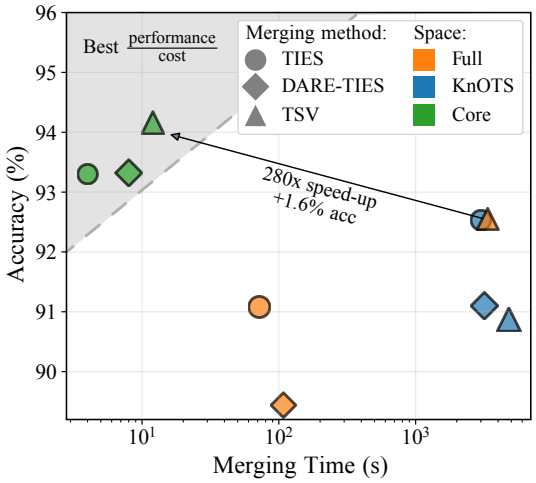

Figure 1: Merging in full space is fast but suboptimal (bottom center). Merging in KnOTS space or using strong merging methods (*e.g.*, TSV) improves performance but increases cost by orders of magnitude (right). **Core Space merging is effective and efficient (top left).** Results on Llama 3 8B.

The main contributions of this paper are the following:

- We introduce *Core Space Merging*, a framework to merge LoRA-adapted models in a shared low-rank basis, avoiding costly full space operations, while improving accuracy. Our approach can be easily integrated with existing merging methods.
- We prove that projection into Core Space ensures **no loss of information**, and provide a complexity analysis demonstrating the **efficiency gain** of merging in the proposed space.
- We present an extensive empirical evaluation showing **state-of-the-art results** achieved at a **fraction of the computational cost** of competing methods by merging in Core Space for vision and language tasks, including experiments on ViT-B/32, ViT-L/14, and Llama 3 8B.

2 Related Work

Parameter-efficient fine-tuning (PEFT). Pre-trained models serve as a starting point for training experts specialized in various downstream tasks [10, 36]. As the size of frontier models grows, the cost of fully fine-tuning such models increases accordingly. Therefore, several parameter-efficient fine-tuning (PEFT), updating a small fraction of parameters, have been proposed, including adapters [15], prefix tuning [25], and prompt tuning [24]. Nowadays, LoRA [16] and its variants [21, 26], which rely on low-rank updates, have emerged as one of the most popular PEFT techniques.

Model merging. The abundance of available expert models inspired the fundamental question behind model merging [32]: *how can we integrate knowledge from multiple expert models into a single multi-task model?* Task Arithmetic [18] proposed to construct *task vectors* (*i.e.*, the parameter difference between expert and base model) and aggregate them via scaled addition, creating a multi-task expert. Since a significant performance gap between the single-task and the merged models remains, many approaches were proposed to address this issue [1, 19, 29, 31, 43, 52, 54, 35]. TIES-Merging [50] focuses on reducing sign conflicts between the parameters of expert models. Model Breadcrumbs [8] removes outliers from the task vectors, while Consensus Merging [46] eliminates catastrophic and selfish weights. Most recent methods, like TSV [12] and Iso-C [28], rely on singular value decomposition (SVD) of weight update matrices to reduce task interference when merging models. However, most methods are designed for merging fully fine-tuned models.

Merging LoRA-adapted models. Methods designed to merge fully fine-tuned models do not necessarily transfer well to merging LoRA-adapted models [44]. The authors of [44] proposed and improved a method to merge LoRAs. However, their approach relies on altering the fine-tuning procedure. KnOTS [42] proposes to merge LoRA updates in the shared subspace, achieving a significant improvement. However, KnOTS performs SVD on the concatenation of full-size matrices instead of leveraging their decomposed update representations, making it costly, especially for large weight matrices. Therefore, finding a method that effectively and efficiently merges LoRA-adapted models remains an open challenge.

3 Preliminaries

LoRA fine-tuning. Low-Rank Adaptation (LoRA) [16] is a technique for efficient fine-tuning of large pre-trained models. Instead of updating the full model weights $W \in \mathbb{R}^{m\times n}$, LoRA introduces two learnable matrices $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{m\times r}$ (where $r \ll \min(m, n)$), and modifies the weight update as $W = W_0 + \Delta W = W_0 + BA$, where W_0 is the original weight matrix. This significantly reduces the number of parameters that need to be updated during fine-tuning.

Model merging. Given a set of T parameters $\{W_1, \dots, W_T\}$, for a common architecture trained on T different tasks, a basic model merging approach calculates task vectors $\Delta W_i = W_i - W_0$, and computes a weighted sum $W_{\text{merged}} = W_0 + \alpha \sum_{i=1}^{T} \Delta W_i$. When dealing with LoRA-adapted models we obtain a common W_0 and a set of decomposed, low-rank updates $\{\Delta W_i = B_i A_i\}_{i=1}^{T}$. However, merging such weight matrices obtained from LoRA leads to suboptimal performance as shown in [42], since LoRA-adapted models are less aligned w.r.t. to their fully fine-tuned counterparts.

4 The Core Space Merging Framework

In this section, we introduce *Core Space Merging* (see Fig. 2), a framework designed to identify an effective and efficient subspace – referred to as the *Core Space* – in which model merging for LoRA-adapted models can be performed while remaining in the low-rank regime. Core Space is designed to be reversible – it ensures no loss of information when projecting into Core Space and back to the original space – while being as compact as possible. Compactness allows for the use of state-of-the-art merging methods relying on Singular Value Decomposition (SVD) of weight matrices, which are highly costly to perform in the original space for large models.

4.1 Model Merging in Core Space

Let $A^{(t)} \in \mathbb{R}^{r\times n}$ and $B^{(t)} \in \mathbb{R}^{m\times r}$ denote the low-rank matrices for task t, derived from a shared pre-trained base model W_0. Each task update $\Delta W^{(t)} = B^{(t)}A^{(t)}$ can be reconstructed from the SVD of the matrices:

$$\begin{aligned} A^{(t)} = U_A^{(t)}\Sigma_A^{(t)}V_A^{(t)\top}, \quad B^{(t)} = U_B^{(t)}\Sigma_B^{(t)}V_B^{(t)\top}, \\ \Delta W^{(t)} = U_B^{(t)}\Sigma_B^{(t)}V_B^{(t)\top}U_A^{(t)}\Sigma_A^{(t)}V_A^{(t)\top}, \end{aligned} \tag{1}$$

where the shapes of the matrices in the decomposition are: $U_A^{(t)} \in \mathbb{R}^{r\times r}$, $\Sigma_A^{(t)} \in \mathbb{R}^{r\times r}$, $V_A^{(t)} \in \mathbb{R}^{n\times r}$, $U_B^{(t)} \in \mathbb{R}^{m\times r}$, $\Sigma_B^{(t)} \in \mathbb{R}^{r\times r}$, and $V_B^{(t)} \in \mathbb{R}^{r\times r}$.

https://doi.org/10.52202/085713-2066

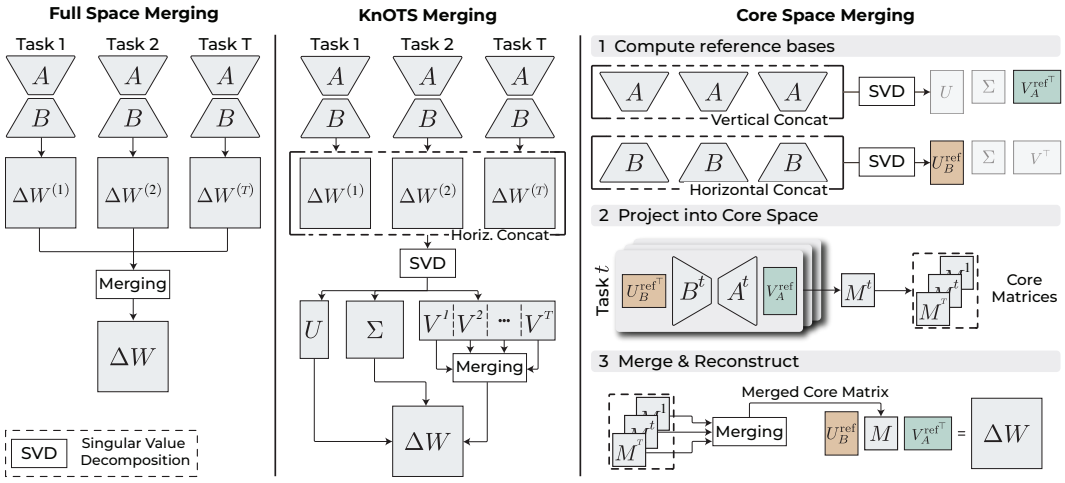

Figure 2: **Full Space Merging** (left) firstly reconstructs full space matrices $\Delta W^{(t)} = B^{(t)}A^{(t)}$, and then performs merging in the full space to obtain ΔW. **KnOTS Merging** concatenates the $\Delta W^{(t)}$ matrices, and performs a costly SVD on this high-dimensional matrix. Then, the $V^{(t)}$ matrices are merged and used to obtain the final ΔW. The proposed **Core Space Merging** (right) performs SVD on a concatenation of low-dimensional $A^{(t)}$ and $B^{(t)}$ matrices to obtain reference bases $(V_A^{\text{ref}}, U_B^{\text{ref}})$. Afterwards, it projects each update into the Core Space to obtain the core matrices $\{M^{(t)}\}_{t=1}^{T}$. It then performs merging in the Core Space and reconstructs to obtain the final ΔW.

Motivation. Under the hypothesis that all tasks share approximately the same common bases (U_B, V_A) such that $\forall t \in \{1, \ldots, T\},\ U_B \approx U_B^{(t)}$ and $V_A \approx V_A^{(t)}$, we have:

$$\Delta W = \sum_{t=1}^{T} \Delta W^{(t)} = \sum_{t=1}^{T} B^{(t)}A^{(t)} \approx U_B \left(\sum_{t=1}^{T} \Sigma_B^{(t)} V_B^{(t)\top} U_A^{(t)} \Sigma_A^{(t)} \right) V_A^{\top}, \tag{2}$$

where $\Sigma_B^{(t)} V_B^{(t)\top} U_A^{(t)} \Sigma_A^{(t)} \in \mathbb{R}^{r \times r}$ encodes the directional transformation applied by the low-rank update of task t. This suggests that, under aligned bases, the sum of low-rank updates (*i.e.*, Task Arithmetic [18]) can be reduced to merging operations in a much smaller $r \times r$ space.

Projecting into the Core Space. In practice, task-specific bases are not aligned, making direct merging of core matrices as in Eq. (2) infeasible. Therefore, we aim to find a shared basis that can represent all tasks without loss of information and that enables merging to be performed in a reduced space compared to the full space $\Delta W^{(t)}$. Intuitively, such a shared basis should span the subspace formed by the union of the individual task subspaces.

Definition 1 (Reference Bases)**.** Given a collection of low-rank matrices $\{A^{(t)}, B^{(t)}\}_{t=1}^{T}$, we define as *reference bases* the orthonormal matrices $U_B^{\text{ref}} \in \mathbb{R}^{m \times Tr}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n \times Tr}$, obtained by performing SVD over the vertically stacked $A^{(t)}$ and horizontally stacked $B^{(t)}$ low-rank matrices across tasks:

$$[B^{(1)}; \ldots; B^{(T)}] = U_B^{\text{ref}} \Sigma_B V_B^{\top}; \quad \begin{bmatrix} A^{(1)} \\ \vdots \\ A^{(T)} \end{bmatrix} = U_A \Sigma_A \left(V_A^{\text{ref}} \right)^{\top}. \tag{3}$$

These bases span a shared latent subspace into which all task-specific updates are projected.

Although the reference bases $(U_B^{\text{ref}}, V_A^{\text{ref}})$ span all task-specific directions, each task t is originally expressed in its own local bases $(U_B^{(t)}, V_A^{(t)})$. To express each update in the common coordinate system for merging, we solve the following least-squares problems:

$$R_B^{(t)} = \underset{R \in \mathbb{R}^{T \cdot r \times r}}{\arg\min} \left\| U_B^{\text{ref}} R - U_B^{(t)} \right\|_F^2, \quad Q_A^{(t)} = \underset{Q \in \mathbb{R}^{T \cdot r \times r}}{\arg\min} \left\| V_A^{\text{ref}} Q - V_A^{(t)} \right\|_F^2, \tag{4}$$

where $V_A^{(t)} \in \mathbb{R}^{n \times r}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n \times T \cdot r}$ (and similarly for $U_B^{(t)}$ and U_B^{ref}). These problems are convex, and since U_B^{ref} and V_A^{ref} are orthonormal, setting the gradients to zero yields the global minimizers (see Sec. A.1 for the full derivation):

$$R_B^{(t)} = {U_B^{\text{ref}}}^\top U_B^{(t)}, \quad Q_A^{(t)} = {V_A^{\text{ref}}}^\top V_A^{(t)}. \tag{5}$$

As we will show in Sec. 4.2, $\|U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)}\|_F^2 = 0$, which allows us to substitute $U_B^{(t)}$ with $U_B^{\text{ref}} R_B^{(t)}$, and similarly $V_A^{(t)}$ with $V_A^{\text{ref}} Q_A^{(t)}$, in Eq. (1), yielding:

$$\Delta W^{(t)} = U_B^{\text{ref}} \underbrace{R_B^{(t)} \Sigma_B^{(t)} {V_B^{(t)}}^\top U_A^{(t)} \Sigma_A^{(t)} {Q_A^{(t)}}^\top}_{\text{task-}t\text{ update in reference coordinates}} {V_A^{\text{ref}}}^\top. \tag{6}$$

By substituting the least-squares solutions from Eq. (5) and using the definitions of $B^{(t)}$ and $A^{(t)}$ from Eq. (1), we can equivalently write:

$$\Delta W^{(t)} = U_B^{\text{ref}} \left({U_B^{\text{ref}}}^\top B^{(t)} A^{(t)} V_A^{\text{ref}} \right) {V_A^{\text{ref}}}^\top. \tag{7}$$

This reformulation expresses each $\Delta W^{(t)}$ in the reference basis, enabling all updates to be compared or merged within a shared coordinate system.

Definition 2 (Core Matrix)**.** We define the *core matrix* $M^{(t)}$ as:

$$M^{(t)} = \left({U_B^{\text{ref}}}^\top B^{(t)} \right) \left(A^{(t)} V_A^{\text{ref}} \right) \in \mathbb{R}^{Tr \times Tr}. \tag{8}$$

This formulation generalizes the middle expression in Eq. (2), where aligned task-specific bases were implicitly assumed. In contrast, the core matrix $M^{(t)}$ is expressed in the reference bases $(U_B^{\text{ref}}, V_A^{\text{ref}})$ and thus does not rely on any alignment assumption. It encodes the directional transformation applied by the low-rank update of task t, providing a compact and lossless representation of each task update in the shared reference space and enabling efficient merging in a reduced $Tr \times Tr$ space.

Reparametrized Model Merging in Core Space. Once task-specific updates $\Delta W^{(t)}$ have been reparameterized into their corresponding core matrices $M^{(t)}$ in the shared reference bases $(U_B^{\text{ref}}, V_A^{\text{ref}})$, Core Space Merging enables model merging to be performed entirely within a compact, aligned, low-rank subspace. Specifically, the merged update is computed by applying a merging operator $\mathcal{M}$ over the set of core matrices:

$$M_{\text{merged}} = \mathcal{M}(\{M^{(t)}\}_{t=1}^T), \tag{9}$$

where $\mathcal{M}$ may be *any merging function*, ranging from simple arithmetic averaging [18] to more advanced, non-linear or geometry-aware techniques [50, 55]. The final update in the original model space, as described also in Algorithm 1, is recovered by projecting M_{merged} back through the reference bases:

$$\Delta W = U_B^{\text{ref}} M_{\text{merged}} {V_A^{\text{ref}}}^\top. \tag{10}$$

Because Core Space is a lossless representation of the original updates for each individual task (see Eq. (7)), merging in this space preserves all relevant task information. Furthermore, when $\mathcal{M}$ is *linear*, such as Task Arithmetic, the merge operation in Core Space is *exactly equivalent* to applying the same merge in the full model space:

$$\mathcal{M}(\{\Delta W^{(t)}\}_{t=1}^T) = \mathcal{M}(\{U_B^{\text{ref}} M^{(t)} {V_A^{\text{ref}}}^\top\}_{t=1}^T) = U_B^{\text{ref}} \mathcal{M}(\{M^{(t)}\}_{t=1}^T) {V_A^{\text{ref}}}^\top. \tag{11}$$

Core Space merging offers key benefits over full space merging:

- **Efficiency.** Core matrices $M^{(t)} \in \mathbb{R}^{Tr \times Tr}$ are significantly smaller than their full space counterparts $\Delta W^{(t)} \in \mathbb{R}^{m \times n}$. This reduction enables high-cost merging algorithms to run at a fraction of the time and memory footprint (see Sec. 4.3).
- **Efficacy.** As shown in Sec. 5.1, Core Space merging improves performance over full space merging when *non-linear* methods are used. In Sec. 5.2, we show that this improvement stems from better alignment and more compact representation of task-specific directions.

https://doi.org/10.52202/085713-2066

Algorithm 1 Core Matrix Alignment and Merging

Require: Low-rank updates $\{(A^{(t)}, B^{(t)})\}_{t=1}^{T}$, merging function $\mathcal{M}(\cdot)$.
1: Stack $A^{(t)}$ vertically, $B^{(t)}$ horizontally
2: Compute V_A^{ref}: stack$(A^{(t)}) = U_A \Sigma_A \underline{V_A^{\text{ref}\top}}$ ▷ reference bases
3: Compute U_B^{ref}: stack$(B^{(t)}) = \underline{U_B^{\text{ref}}} \Sigma_B V_B^{\top}$
4: **for** $t = 1$ to T **do**
5: Compute: $M^{(t)} = (U_B^{\text{ref}\top} B^{(t)})(A^{(t)} V_A^{\text{ref}})$ ▷ Eq. (8)
6: Merge aligned core matrices: $M_{\text{merged}} = \mathcal{M}(\{M^{(t)}\}_{t=1}^{T})$
7: **return** $\Delta W = U_B^{\text{ref}} M_{\text{merged}} V_A^{\text{ref}\top}$ ▷ reconstructed merged model

4.2 No Information Loss in Core Space Representation

Replacing $U_B^{(t)}$ and $V_A^{(t)}$ with $U_B^{\text{ref}} R_B^{(t)}$ and $V_A^{\text{ref}} Q_A^{(t)}$ to obtain Eqs. (6) and (7), which define the final form of the core matrix, requires that the solutions to the least-squares problems in Eq. (4) incur zero alignment error. That is,

$$\left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 = 0, \qquad \left\| V_A^{\text{ref}} Q_A^{(t)} - V_A^{(t)} \right\|_F^2 = 0. \tag{12}$$

In this section, we show that the reference bases U_B^{ref} and V_A^{ref}, obtained via the SVD of the stacked matrices $B^{(t)}$ and $A^{(t)}$ (see Eq. (3)), *minimize* the total alignment error across all T tasks, achieving an error of *exactly zero*. To illustrate this, we first analyze the alignment error for a single task t, focusing on U_B^{ref}. Analogous results hold symmetrically for V_A^{ref}. For clarity, we assume in the following derivations that $T \cdot r \leq m$ and $T \cdot r \leq n$, so that the total LoRA rank does not exceed the maximum possible rank of the target weight matrix. In Sec. A.4, we provide a more general analysis that removes this assumption and demonstrate that the zero alignment error result continues to hold.

Lemma. *Let $U_B^{(t)} \in \mathbb{R}^{m \times r}$ and $U_B^{\text{ref}} \in \mathbb{R}^{m \times T \cdot r}$ be matrices with orthonormal columns, and let $R_B^{(t)} = U_B^{\text{ref}\top} U_B^{(t)} \in \mathbb{R}^{T \cdot r \times r}$ be the optimal solution minimizing the error of the least-square problem. Then, the optimal alignment error is given by:*

$$\varepsilon_U = \left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 = r - \left\| U_B^{(t)\top} U_B^{\text{ref}} \right\|_F^2. \tag{13}$$

The proof, provided in Sec. A.2, leverages the properties of Frobenius norm and the orthonormality of $U_B^{(t)}$ and U_B^{ref}. To formally demonstrate that our chosen reference basis U_B^{ref} minimizes the alignment error across all T tasks (or equivalently maximize $||U_B^{(t)\top} U_B^{\text{ref}}||_F^2$ for each task t), we first formulate the following constrained optimization problem for a single task, and then extend it to the multi-task scenario:

$$\max_{U \in \mathcal{S}} \left\| U_B^{(t)\top} U \right\|_F^2 = \max_{U \in \mathcal{S}} \operatorname{tr}\left(U^{\top} U_B^{(t)} U_B^{(t)\top} U \right), \quad \text{where} \quad \mathcal{S} = \left\{ U \in \mathbb{R}^{m \times Tr} \;\middle|\; U^{\top} U = I_{T \cdot r} \right\} \tag{14}$$

and $\operatorname{tr}(\cdot)$ denotes the trace operator. The optimization domain is restricted to the Stiefel manifold $\mathcal{S}$ (*i.e.*, the set of matrices with orthonormal columns). The following lemma characterizes the solution to the following optimization problem:

Lemma. *A solution U^* to the quadratic program in Eq.* (14) *is given by a basis whose columns include the r eigenvectors corresponding to nonzero eigenvalues of $B^{(t)} B^{(t)\top} \in \mathbb{R}^{m \times m}$ or, equivalently, by the r left singular vectors of the matrix $B^{(t)}$. Moreover, at the optimum, the objective attains its maximum value r, resulting in zero alignment error in Eq.* (13)*.*

We refer the reader to Sec. A.3 for a detailed proof. Briefly, the result follows by applying the method of Lagrange multipliers to augment the optimization objective with the Stiefel manifold constraint and then enforcing stationarity by setting the gradient of the Lagrangian to zero.

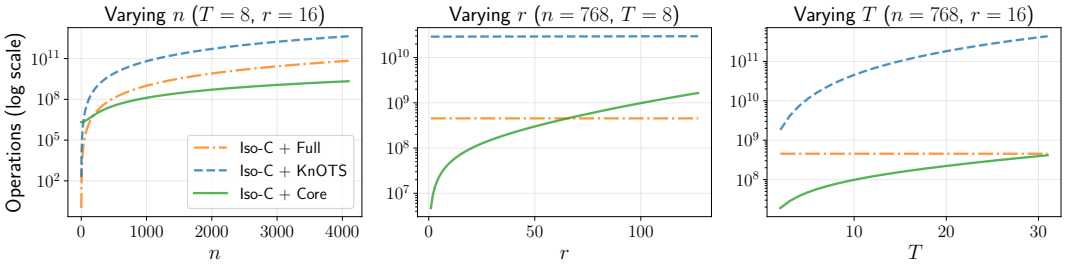

Figure 3: **Core Space merging is more efficient than the previous state-of-the-art KnOTS.** The cost is similar to full space merging, which results in much lower performance. We visualize the number of operations performed to merge T rank r LoRA modules of final shape $n \times n$.

Extension to multiple tasks. Achieving zero reconstruction error for a single model t does not guarantee optimality for any other model $t' \neq t$. Therefore, we aim to identify a reference basis U^* that jointly optimizes Eq. (14) across all T models. We formulate this global problem as:

$$\max_{U \in \mathcal{S}} \sum_{t=1}^{T} \operatorname{tr}(U^\top U_B^{(t)} {U_B^{(t)}}^\top U) = \max_{U \in \mathcal{S}} \operatorname{tr}(U^\top \mathbf{U}_B \mathbf{U}_B^\top U), \tag{15}$$

where $\mathbf{U}_B = [U_B^{(1)},\ U_B^{(2)},\ \ldots,\ U_B^{(T)}]$ denotes the horizontal concatenation of all $U_B^{(t)}$ matrices. The equality in Eq. (15) follows directly from the linearity of the trace operator and the distributivity of matrix multiplication concerning matrix addition: $M^\top A_1 M + M^\top A_2 M = M^\top (A_1 + A_2) M$.

By considerations analogous to the single task-case, a global solution U^* is given by the top $T \cdot r$ left singular vectors of the matrix $\mathbf{B}$, obtained by vertically stacking each matrix $B^{(t)}$, *i.e.*, $U^* = U_B^{\text{ref}}$. This choice ensures zero alignment error simultaneously across all T tasks, consistent with the procedure described in Sec. 4.1.

4.3 Computational Complexity Analysis

We summarize the time complexities of TA, Iso-C, and TSV merged in all three spaces (Full, KnOTS, and our Core) in Tab. 1 and Fig. 3. Details on the derivation of the complexities are in Appendix B. Our approach exhibits a time complexity comparable to that of Task Arithmetic in full space. Our method's additional terms are negligible unless the product $T \cdot r$ becomes significantly large. A key advantage of our method lies in its scalability compared to KnOTS, whose time complexity is super-cubic, driven by a factor that scales cubically with the weight matrix size n. Finally, we emphasize the minimal additional overhead incurred when combining our method with Iso-C or TSV in the Core Space; it introduces a cost substantially lower than its counterpart in full space or KnOTS space.

Table 1: $\mathcal{O}(\cdot)$ time complexities. The cheapest method is highlighted in **bold** ($T, r \ll n$).

Space	TA	Iso-C	TSV
Full	$\boldsymbol{n^2Tr}$	n^3	n^3T
KnOTS	n^3T^2	$n^3T^2 + n^2Tr$	$n^3T^2 + T^3r^2n$
Core	$\boldsymbol{n^2Tr}$	$\boldsymbol{n^2Tr + T^3r^3}$	$\boldsymbol{n^2Tr + T^4r^3}$

5 Experimental Results

Experimental setup. We follow the experimental setup of KnOTS and use the LoRA checkpoints provided by the authors [42]. For the vision experiments, we use two variants of CLIP [36] with ViT-B/32 and ViT-L/14 [11] as vision encoders fine-tuned on a standard set of 8 tasks. We employ Llama 3 8B [13] fine-tuned on 6 NLI tasks for the language experiments. All models are fine-tuned with LoRA [16] with rank 16 applied on all matrices (keys, queries, values, and outputs) across all attention layers. Following [42], we report *normalized accuracy* as a ratio of the accuracy of the merged model on a given task to the accuracy of the model fine-tuned on this task. We also report *absolute accuracy* for the joint-task setting (additional experimental details in Sec. D).

https://doi.org/10.52202/085713-2066

Table 2: Normalized accuracies of merged models on NLI tasks for Llama 3 8B.

Method	Space	SNLI	MNLI	SICK	QNLI	RTE	SCITAIL	Avg (ΔAcc)	Time [s]	Rel. Time
Abs. Accuracy		92.50	90.31	91.58	94.49	89.86	96.52	-	-	-
TA	Full	93.57	95.28	87.96	68.71	100.0	96.73	90.38 (+0.00)	9	-
TIES	Full	95.17	96.19	84.18	74.18	100.0	96.78	91.08 (+0.00)	72	9
	KnOTS	91.82	94.19	92.97	78.57	100.0	97.61	92.53 (+1.45)	3000	375
	Core	92.07	93.51	93.63	83.72	99.19	97.66	**93.30** (+2.22)	**8**	1
DARE-TIES	Full	94.76	96.8	78.39	72.08	98.39	96.20	89.44 (+0.00)	108	13
	KnOTS	91.62	96.72	74.90	84.75	99.48	99.13	91.10 (+1.66)	3180	397
	Core	92.10	93.58	93.70	83.68	99.19	97.66	**93.32** (+3.88)	**8**	1
TSV	Full	95.38	95.12	88.83	76.80	101.61	97.56	92.55 (+0.00)	3360	280
	KnOTS	92.53	95.83	82.77	77.01	100.0	97.08	90.87 (-1.68)	4800	400
	Core	95.86	95.70	89.25	83.89	102.42	97.86	**94.16** (+1.61)	**12**	1
Iso-C	Full	55.00	39.04	76.54	55.90	46.77	69.25	57.08 (+0.00)	540	67
	KnOTS	85.28	52.86	89.43	54.90	75.00	77.73	72.53 (+15.45)	4860	607
	Core	91.54	90.10	87.87	75.85	99.19	97.42	**90.33** (+33.25)	**8**	1

Baseline merging spaces. We compare our proposed Core Space with two alternative merging spaces. **Full Space** operates in space of full reconstructed weight matrices $\Delta W^{(t)} = B^{(t)}A^{(t)} \in \mathbb{R}^{m\times n}$. **KnOTS Space** [42] operates in the space of the right singular vectors of the concatenated reconstructed weight matrices $\{\Delta W^{(t)}\}_{t=1}^{T} \in \mathbb{R}^{m\times nT}$.

Baseline merging methods. We evaluate each merging space using the following merging methods. **Task Arithmetic (TA)** [18] performs a scaled summation of each task matrix $W_{\text{merged}} = W_0 + \alpha \sum_{i=1}^{T} \Delta W_i$. As this is a linear operation, the results of merging in each space are the same (see Eq. (11) for Core and [42] for KnOTS). **TIES** [50] trims low-magnitude parameters and averages parameters with dominating sign, while **DARE** [55] preprocesses task vectors by randomly dropping a fraction of parameters and rescaling the remaining ones. **TSV** [12] concatenates low-rank approximations of task matrices and orthogonalizes them across tasks. **CART** [6] calculates centered task vectors as a difference of fine-tuned weights from the average of all fine-tuned weights and performs task arithmetic on the low-rank approximation of these centered task vectors. **Iso-C** [28] flattens the spectrum of singular values for a model merged with task arithmetic. As the spectrum flattening can be performed on weights merged with any merging technique, we combine Iso with other merging techniques, denoting it with +Iso-C.

5.1 Results

LLMs merging. We present Llama 3 8B results in natural language inference in Tab. 2. In line with our complexity analysis, merging in Core Space is much more efficient than merging in Full or KnOTS space, bringing up to $600\times$ merging speed-up. Moreover, merging in Core Space improves the performance of all tested merging methods. In particular, it elevates TSV to 94.16% average normalized accuracy, achieving state-of-the-art results.

Per-task evaluation in vision setting. We present per-task vision results for ViT-B/32 in Tab. 3. We observe that 8 out of 9 merging methods achieve their highest average accuracy when performed in our proposed Core Space. The best combination – TSV + Iso-C merged in Core Space – achieves state-of-the-art average normalized accuracy of 76.3%. It significantly outperforms the previously reported SoTA of TIES in KnOTS space, achieving 68.0% [42]. Similar conclusions hold for experiments on ViT-L/14 presented in Sec. E.1.

Heterogeneous ranks. While handling LoRA modules with heterogeneous ranks might seem non-trivial, our method supports it seamlessly without modification. Even with different ranks, the modules can be concatenated across tasks to form an aggregate basis spanning the combined subspaces, after which projection and alignment are applied to each local task core matrix. Since SVD makes no assumptions about input ranks, it yields valid orthonormal bases in all cases, enabling our framework to merge variable-rank LoRA modules naturally. We evaluate this setting by assigning rank 16 to half the tasks and rank 64 to the rest; the results reported in Sec. E.2 show that our method still outperforms other approaches.

Table 3: Normalized accuracies of merged models on the vision tasks with ViT-B/32.

Method	Space	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC	SUN397	SVHN	Avg (Δ Acc)
Abs. Accuracy		74.00	58.30	99.00	92.70	99.30	88.40	64.50	96.20	-
TA	Full	81.97	73.72	48.97	42.24	53.12	71.50	97.46	41.25	63.78 (+0.00)
TIES	Full	82.37	72.72	49.91	36.62	57.16	69.38	96.92	44.56	63.70 (+0.00)
	KnOTS	83.75	74.45	50.36	47.31	67.01	71.79	96.51	50.64	67.73 (+4.03)
	Core	84.74	76.46	52.19	50.41	67.36	71.21	96.45	50.18	**68.63 (+4.93)**
DARE-TIES	Full	82.14	73.72	49.35	37.78	56.63	70.14	97.35	42.12	63.65 (+0.00)
	KnOTS	82.01	72.90	44.15	45.54	60.59	70.89	95.56	47.64	64.91 (+1.26)
	Core	84.57	76.09	57.09	51.01	66.64	71.39	96.16	52.14	**69.39 (+5.74)**
TSV	Full	83.44	75.55	50.99	45.03	59.31	73.33	96.40	49.23	**66.66 (+0.00)**
	KnOTS	81.86	74.91	51.25	41.64	53.93	71.64	97.95	40.36	64.19 (-2.47)
	Core	83.86	75.09	52.64	45.39	58.53	72.95	97.63	45.21	66.41 (-0.25)
CART	Full	83.04	81.93	50.39	70.17	59.14	79.11	99.26	49.11	71.52 (+0.00)
	KnOTS	83.94	75.18	52.23	54.48	64.78	74.48	95.88	55.73	69.59 (-1.93)
	Core	80.83	83.94	54.99	73.28	66.25	80.95	98.69	48.57	**73.44 (+1.92)**
TIES +Iso-C	Full	78.86	74.45	60.01	39.02	66.65	70.30	98.39	48.59	67.03 (+0.00)
	KnOTS	78.46	80.38	58.81	64.97	72.10	76.89	98.33	49.78	72.47 (+5.44)
	Core	82.91	84.76	52.41	78.79	71.56	81.43	99.48	52.14	**75.44 (+8.41)**
DARE-TIES +Iso-C	Full	78.71	75.54	50.84	42.86	65.03	71.88	98.92	48.08	66.48 (+0.00)
	KnOTS	82.93	74.18	49.31	46.73	66.64	71.82	96.72	50.57	67.36 (+0.88)
	Core	83.27	83.12	54.55	79.04	71.83	82.08	99.36	52.37	**75.70 (+9.22)**
TSV +Iso-C	Full	79.38	80.38	57.99	65.64	64.22	79.74	98.59	46.49	71.55 (+0.00)
	KnOTS	80.81	83.03	58.25	74.34	67.66	79.69	98.54	49.86	74.02 (+2.47)
	Core	82.98	85.12	50.95	84.25	71.14	84.39	99.06	53.53	**76.43 (+4.88)**
CART +Iso-C	Full	80.33	82.11	57.31	77.38	71.17	81.57	98.72	51.91	75.06 (+0.00)
	KnOTS	82.05	80.47	56.12	64.58	62.40	78.81	99.22	45.05	71.09 (-3.97)
	Core	82.93	84.21	51.14	81.32	72.12	82.83	99.33	55.32	**76.15 (+1.09)**
Iso-C	Full	80.16	83.03	51.44	74.76	70.72	79.89	98.66	50.20	73.60 (+0.00)
	KnOTS	80.33	79.29	57.50	67.60	65.63	79.54	99.26	46.62	71.97 (-1.63)
	Core	83.35	84.30	50.13	81.97	71.07	83.46	99.17	53.90	**75.92 (+2.32)**

Table 4: Joint-task setting absolute accuracy of merged models on the vision tasks with ViT-B/32.

Space	TA	TIES	DARE-TIES	TSV	CART	TIES +Iso-C	DARE-TIES +Iso-C	TSV +Iso-C	CART +Iso-C	Iso-C
Full	43.5	43.6	44.0	**45.4**	44.8	43.5	44.3	48.3	44.8	52.1
KnOTS	43.5	46.8	45.2	44.6	44.7	40.5	44.8	51.4	52.6	52.9
Core	43.5	**47.4**	**47.6**	44.5	**49.6**	**54.1**	**54.0**	**55.7**	**55.6**	**55.9**

Additional PEFT methods Our method can also be applied to other PEFT methods, such as VeRA [21]. In VeRA, $\Delta W = \Lambda_b B \Lambda_d A$, where $A \in \mathbb{R}^{r \times n}$, $B \in \mathbb{R}^{m \times r}$, $\Lambda_b \in \mathbb{R}^{1 \times m}$, and $\Lambda_d \in \mathbb{R}^{r \times 1}$. Unlike LoRA, in VeRA the A and B matrices are randomly chosen, frozen, and shared across the network, while only the two scaling vectors Λ are learned for each layer. To adapt VeRA to our Core Space merging, we absorb the scaling vectors into the matrices, *i.e.*, $\tilde{B} = \Lambda_b B$ and $\tilde{A} = \Lambda_d A$, and then treat $\tilde{A}$ and $\tilde{B}$ as the LoRA A and B matrices. To confirm that our method also works with VeRA, we report additional experiments in Sec. E.3, which show that our method outperforms other baselines also in this setting.

Joint-task evaluation in vision setting. We also evaluate vision models in the challenging joint-task setting introduced in [42], in which the task ID is unknown during inference. Instead of performing a multi-task evaluation, it evaluates the merged model on the union of all classes, requiring the model to distinguish between classes from all tasks. We present the results in Tab. 4. Core Space facilitates merging with almost all methods, achieving state-of-the-art results when combined with Iso-C.

5.2 Analysis

Truncation. We herein compare the utilization of full space and Core Space for models merged with TA. Firstly, we calculate the SVD of the merged matrices: ΔW_{merged} for full space and M_{merged} for Core Space. Then, we truncate a fraction of p least significant values, *i.e.*, $\sigma_i = 0$ for $i > (1-p) * \dim(\Sigma)$, and observe a drop in the accuracy of the merged model after truncation. As shown in Fig. 4, in full space we can truncate a fraction up to $p = 0.8$ values without performance loss, while in Core Space, truncation of any component results in a performance drop. It shows that

https://doi.org/10.52202/085713-2066

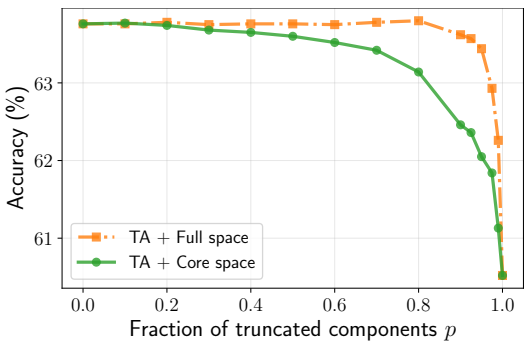

Figure 4: Most components in full space are irrelevant when doing Task-Arithmetic (TA). Removing any components from the Core Space results in a performance drop, showing that it is an information-dense space. We report the results on vision tasks with ViT-B/32.

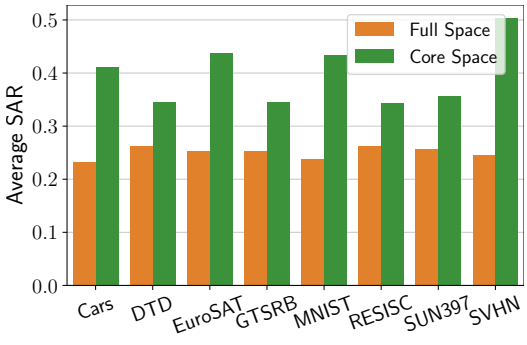

Figure 5: **Subspace Alignment Ratio** (SAR) [28]. Each bar shows the average SAR between LoRA task matrices, Full, and Core Space. In Core Space, task matrices exhibit higher SAR. The associated performance gains suggest that better alignment facilitates more effective merging.

the Core Space is dense while the full space contains many unused or redundant components. We hypothesize that the compactness of Core Space facilitates model merging as it extracts only the relevant components.

Core Space improves subspace alignment. In this Section, we evaluate the Subspace Alignment Ratio (SAR) [28] between each pair of LoRA updates fine-tuned on different tasks. The SAR measures how much of the subspace of one task is contained in another and correlates with post-merge performance. We compute SAR in full and Core Space. Fig. 5 shows that Core Space yields consistently higher alignment. We argue that this result is because the Core Space enforces a shared basis across tasks, which filters out task-specific noise and promotes alignment. In Sec. C.1 we show that higher SAR correlates with lower merging interference.

Choice of the reference basis. To evaluate the reference bases choice, we assess the performance when adopting different reference bases, and compute the alignment error ε_U defined in Eq. (13) (averaged over all layers and tasks). We present the results in Tab. 5. In row (1), we evaluate using the basis of the first task as a reference basis. In row (2), we set the reference basis to a random orthonormal basis of the same dimensionality. These two bases perform much less than our reference basis in row (3). Moreover, we confirm that the optimal reference basis from row (3) achieves zero alignment error. Additionally, we verified experimentally that even in the extreme case where $T \cdot r > \min(m, n)$ (*e.g.*, $Tr = 2048 > 768$ for merging 8 ViT-B/32 LoRA models), the reconstruction error defined in Eq. (13) remains exactly zero, consistent with the generalized theoretical result in Sec. A.4.

Table 5: Ablations on the choice of reference basis. Our basis (3) achieves higher results than the single-task basis (1) and the random orthonormal basis of the same dimensionality (2). We proceed with V_A^{ref} analogously to U_B^{ref}. We report the TIES-Core results on vision tasks with ViT-B/32.

	Reference Basis U_B^{ref}	**Shape**	**Avg. Acc.**	**Avg.** ε_U
(1)	$U_B^{(1)}$ (first task)	$m \times r$	60.4	13.4
(2)	Random orthonormal	$m \times Tr$	61.6	13.3
(3)	Concatenation (Eq. (3))	$m \times Tr$	**68.6**	**0.0**

6 Conclusion

We propose Core Space, an efficient and effective method for merging LoRA modules. By projecting task-specific LoRA updates into a common subspace, Core Space reduces alignment error, leading to consistent accuracy improvements and SOTA results in both vision and language settings, while remaining computationally efficient. Our evaluations across vision and language domains confirm its scalability and strong performance in practical settings. We believe that Core Space can contribute to more efficient and accessible model adaptation in multi-task settings, particularly for large models.

https://doi.org/10.52202/085713-2066

Acknowledgments

We acknowledge the Spanish project PID2022-143257NB-I00, financed by MCIN/AEI/10.13039/501100011033 and ERDF/EU, and Funded by the European Union ELLIOT project. This work was supported by the Fortissimo Plus (FFplus) project Grant Agreement No. 101163317), under the European HighPerformance Computing Joint Undertaking (EuroHPC JU) and the Digital Europe Programme. The authors gratefully acknowledge access to compute resources enabled by FFplus. Funded by the European Union. Aniello Panariello acknowledges travel support from ELIAS (GA no 101120237). Daniel Marczak is supported by National Centre of Science (NCN, Poland) Grant No. 2021/43/O/ST6/02482. Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I and National Centre of Science (NCN, Poland) Grant No. 2023/51/D/ST6/02846.

References

[1] T. Akiba, M. Shing, Y. Tang, Q. Sun, and D. Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 2025.

[2] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Alhammadi, M. Daniele, D. Heslow, J. Launay, Q. Malartic, B. Noune, B. Pannier, and G. Penedo. The falcon series of language models: Towards open frontier models. *arXiv preprint arXiv: 2311.16867*, 2023.

[3] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *Empirical Methods in Natural Language Processing*, 2015.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.

[5] G. Cheng, J. Han, and X. Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 2017.

[6] J. Choi, D. Kim, C. Lee, and S. Hong. Revisiting weight averaging for model merging. *arXiv preprint arXiv:2412.12153*, 2024.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing Textures in the Wild. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014.

[8] M.-J. Davari and E. Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *Proceedings of the European Conference on Computer Vision*, 2024.

[9] DeepSeek-AI. Deepseek-v3 technical report. *arXiv preprint arXiv: 2412.19437*, 2024.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2019.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] A. A. Gargiulo, D. Crisostomi, M. S. Bucarelli, S. Scardapane, F. Silvestri, and E. Rodolà. Task singular vectors: Reducing task interference in model merging. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2025.

[13] A. Grattafiori, A. Dubey, A. Jauhri, and A. P. *et al.* The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*, 2024.

 https://doi.org/10.52202/085713-2066

[14] P. Helber, B. Bischke, A. Dengel, and D. Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[15] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019.

[16] J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2021.

[17] Hugging Face. https://huggingface.co, 2024.

[18] G. Ilharco, M. T. Ribeiro, M. Wortsman, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

[19] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations*, 2023.

[20] T. Khot, A. Sabharwal, and P. Clark. SciTail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[21] D. J. Kopiczko, T. Blankevoort, and Y. M. Asano. Vera: Vector-based random matrix adaptation. *International Conference on Learning Representations*, 2023.

[22] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops*, 2013.

[23] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online].*, 2, 2010.

[24] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[25] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[26] S. Liu, C. Wang, H. Yin, P. Molchanov, Y. F. Wang, K. Cheng, and M. Chen. Dora: Weight-decomposed low-rank adaptation. In *International Conference on Machine Learning*, 2024.

[27] G. Mancusi, M. Bernardi, A. Panariello, A. Porrello, S. Calderara, and R. Cucchiara. Is multiple object tracking a matter of specialization? In *Advances in Neural Information Processing Systems*, 2024.

[28] D. Marczak, S. Magistri, S. Cygert, B. Twardowski, A. D. Bagdanov, and J. van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *International Conference on Machine Learning*, 2025.

[29] D. Marczak, B. Twardowski, T. Trzci'nski, and S. Cygert. Magmax: Leveraging model merging for seamless continual learning. *Proceedings of the European Conference on Computer Vision*, 2024.

[30] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2014.

[31] I. E. Marouf, S. Roy, E. Tartaglione, and S. Lathuilière. Weighted ensemble models are strong continual learners. *arXiv preprint arXiv: 2312.08977*, 2023.

[32] M. Matena and C. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2021.

[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Neural Information Processing Systems Workshops*, 2011.

[34] G. Ortiz-Jiménez, A. Favero, and P. Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023.

[35] A. Panariello, E. Frascaroli, P. Buzzega, L. Bonicelli, A. Porrello, and S. Calderara. Modular embedding recomposition for incremental learning. *Brit. Mach. Vis. Conf.*, 2025.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[37] A. Rame, M. Kirchmeyer, T. Rahier, A. Rakotomamonjy, P. Gallinari, and M. Cord. Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35, 2022.

[38] F. Rinaldi, G. Capitani, L. Bonicelli, D. Crisostomi, F. Bolelli, E. Ficarra, E. Rodolà, S. Calderara, and A. Porrello. Update your transformer to the latest release: Re-basin of task vectors. *International Conference on Machine Learning*, 2025.

[39] A. Soutif, S. Magistri, J. v. d. Weijer, and A. D. Bagdanov. An empirical analysis of forgetting in pre-trained models with incremental low-rank updates. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, Proceedings of Machine Learning Research, 2025.

[40] C. . C. Staff. Singular value decompositions, 2020. Course Notes, CS 357: Numerical Methods I, Fall 2020.

[41] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

[42] G. Stoica, P. Ramesh, B. Ecsedi, L. Choshen, and J. Hoffman. Model merging with svd to tie the knots. *International Conference on Learning Representations*, 2025.

[43] D. Tam, M. Bansal, and C. Raffel. Merging by matching models in task subspaces. *arXiv preprint arXiv: 2312.04339*, 2023.

[44] A. Tang, L. Shen, Y. Luo, Y. Zhan, H. Hu, B. Du, Y. Chen, and D. Tao. Parameter efficient multi-task model fusion with partial linearization. In *International Conference on Learning Representations*, 2023.

[45] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

[46] K. Wang, N. Dimitriadis, G. Ortiz-Jiménez, F. Fleuret, and P. Frossard. Localizing task information for improved model merging and compression. In *International Conference on Machine Learning*, 2024.

[47] A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *Naacl*, 2018.

[48] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022.

[49] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 2016.

[50] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

 https://doi.org/10.52202/085713-2066

[51] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[52] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv: 2408.07666*, 2024.

[53] E. Yang, L. Shen, Z. Wang, G. Guo, X. Chen, X. Wang, and D. Tao. Representation surgery for multi-task model merging. *International Conference on Machine Learning*, 2024.

[54] E. Yang, Z. Wang, L. Shen, S. Liu, G. Guo, X. Wang, and D. Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024.

[55] L. Yu, B. Yu, H. Yu, F. Huang, and Y. Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, 2024.

[56] Q. Zhang, M. Chen, A. Bukharin, N. Karampatziakis, P. He, Y. Cheng, W. Chen, and T. Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

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: We clearly state our main contributions in both the abstract and the introduction, specifically proposing the Core Space merging framework for LoRA-adapted models. These claims are supported by both theoretical analysis and extensive empirical validation in the main body of the paper.

   Guidelines:

   - The answer NA 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 NA 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: We include a discussion of limitations in the conclusion, noting that while Core Space can handle varying LoRA ranks, we do not empirically explore this capability. We also highlight the restriction to same-architecture merging and leave extensions to heterogeneous backbones and alternative PEFT methods (e.g., VeRA) as future work.

   Guidelines:

   - The answer NA 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.

https://doi.org/10.52202/085713-2066

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: We present full derivations and assumptions for our theoretical results, including proofs of zero alignment error and properties of the Core Space in the main text and supplementary material.

   Guidelines:

   - The answer NA 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: We detail the datasets, models (ViT-B/32, ViT-L/14, Llama 3 8B), LoRA rank configuration, and baselines used. The experimental settings closely follow those established in KnOTS, and we clarify this alignment.

   Guidelines:

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

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

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

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

   Answer: [Yes]

   Justification: We do not produce any new data. The checkpoints used are from *Model merging with SVD to tie the Knots*. The code used to generate our results will be shared in the supplementary materials and released upon acceptance, along with detailed instructions for reproducing our experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all essential settings such as LoRA rank, number of tasks, model types, and the exact layers modified. Supplementary material provides additional configuration details.

   Guidelines:

   - The answer NA 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]

https://doi.org/10.52202/085713-2066

Justification: Due to the computational cost of running multiple repetitions across dozens of tasks and large models (e.g., Llama 3 8B), we do not include error bars or statistical significance tests. Instead, we focus on reporting consistent trends across a wide range of datasets and provide detailed per-task results. Given the stable performance across runs and the deterministic nature of the merging process (given fixed seeds and checkpoints), we believe the reported results reliably support our main claims.

Guidelines:

- The answer NA 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: We describe our experimental setup in supplementary materials, specifying the exact GPU models used (NVIDIA L40S and RTX 4080), memory capacity, and the consistency of hardware across timing benchmarks. Merging times and relative computational costs are reported in Table 2, allowing for a fair comparison.

   Guidelines:

   - The answer NA 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: Our research complies with the NeurIPS Code of Ethics. We use only publicly available datasets and models, and our contributions do not involve sensitive or private data.

   Guidelines:

- The answer NA 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: We address broader impacts in the conclusion, noting that Core Space can contribute to more efficient and accessible model adaptation in multi-task settings.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not involve the release of models or data that pose a high risk for misuse.

    Guidelines:

    - The answer NA 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**

https://doi.org/10.52202/085713-2066

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: We list all datasets used along with their licenses in the supplementary material. We also ensure any third-party checkpoints (e.g., from KnOTS) are properly cited.

Guidelines:

- The answer NA 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: [NA]

Justification: We do not release new datasets or models as part of this work.

Guidelines:

- The answer NA 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: [NA]

Justification: Our paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA 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: [NA]

    Justification: Our research does not involve human subjects or require IRB approval.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Large language models were not used in the methodology, experiments, or theoretical analysis. Any LLM usage was limited to minor editing assistance and did not affect the scientific content.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-2066

Appendix

Contents of the Appendix

A No Information Loss in Core Space Representation

A.1 Least Squares Solutions

Let $U_A^{(t)} \in \mathbb{R}^{r\times r}$, $\Sigma_A^{(t)} \in \mathbb{R}^{r\times r}$, $V_A^{(t)} \in \mathbb{R}^{n\times r}$, $U_B^{(t)} \in \mathbb{R}^{m\times r}$, $\Sigma_B^{(t)} \in \mathbb{R}^{r\times r}$, and $V_B^{(t)} \in \mathbb{R}^{r\times r}$ be the components of two low-rank matrices $A^{(t)}$ and $B^{(t)}$ represented in SVD form. Let $U_B^{\text{ref}} \in \mathbb{R}^{m\times T\cdot r}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n\times T\cdot r}$ be the shared *reference bases*, obtained by taking the left and the right singular vectors from the SVD of the horizontally and vertically stacked low-rank matrices $B^{(t)}$ and $A^{(t)}$, respectively (see Eq. 4 in the main paper). We assume $T \cdot r \leq m$ and $T \cdot r \leq n$, where T is the number of tasks and r is LoRA rank, as both are typically small relative to the feature dimension. Then, the solutions to the least square problems:

$$R_B^{(t)} = \underset{R\in\mathbb{R}^{T\cdot r\times r}}{\arg\min} \left\| U_B^{\text{ref}} R - U_B^{(t)} \right\|_F^2, \quad Q_A^{(t)} = \underset{Q\in\mathbb{R}^{T\cdot r\times r}}{\arg\min} \left\| V_A^{\text{ref}} Q - V_A^{(t)} \right\|_F^2, \tag{16}$$

are given by:

$$R_B^{(t)} = {U_B^{\text{ref}}}^\top U_B^{(t)}, \quad Q_A^{(t)} = {V_A^{\text{ref}}}^\top V_A^{(t)}. \tag{17}$$

Proof. Since $V_A^{(t)}$ and $U_B^{(t)}$ come from the SVDs of $A^{(t)}$ and $B^{(t)}$, they are orthonormal:

$${V_A^{(t)}}^\top V_A^{(t)} = I_r \qquad {U_B^{(t)}}^\top U_B^{(t)} = I_r,$$

where I_r is the $r \times r$ identity matrix. Similarly, the reference bases U_B^{ref} and V_A^{ref} also have orthonormal columns:

$${V_A^{\text{ref}}}^\top V_A^{\text{ref}} = I_{T\cdot r}, \qquad {U_B^{\text{ref}}}^\top U_B^{\text{ref}} = I_{T\cdot r},$$

where $I_{T\cdot r}$ is the $T \cdot r \times T \cdot r$ and $T \cdot r \times T \cdot r$ identity matrix.

Consider the problems in Eq. (16). Each objective is convex and admits a unique global minimizer, as U_B^{ref} and V_A^{ref} have full column rank due to their orthonormality. To solve the first problem, we compute the gradient of the objective function with respect to R:

$$\frac{\partial \left\| U_B^{\text{ref}} R - U_B^{(t)} \right\|_F^2}{\partial R} = 2{U_B^{\text{ref}}}^\top (U_B^{\text{ref}} R - U_B^{(t)}).$$

Setting the gradient to zero and solving the resulting equation gives:

$$U_B^{\text{ref}\top} U_B^{\text{ref}} R_B^{(t)} = U_B^{\text{ref}\top} U_B^{(t)} \quad \Rightarrow \quad R_B^{(t)} = U_B^{\text{ref}\top} U_B^{(t)},$$

since $U_B^{\text{ref}\top} U_B^{\text{ref}} = I_{T\cdot r}$.

Similarly, for the second problem in Eq. (16):

$$Q_A^{(t)} = V_A^{\text{ref}\top} V_A^{(t)}$$

□

A.2 Alignment Error Quantification

Lemma. *Let $U_B^{(t)} \in \mathbb{R}^{m\times r}$ and $U_B^{\text{ref}} \in \mathbb{R}^{m\times T\cdot r}$ be matrices with orthonormal columns where $T \cdot r = \text{rank}([B^{(1)}, \ldots, B^{(T)}]) \leq m$, and let $R_B^{(t)} = U_B^{\text{ref}\top} U_B^{(t)} \in \mathbb{R}^{T\cdot r\times r}$ be the optimal solution minimizing the error of the least-square problem. Then, the optimal alignment error is given by:*

$$\varepsilon_U = \left\| U_B^{\text{ref}} R_B^{(t)} - U_B^{(t)} \right\|_F^2 = r - \left\| U_B^{(t)\top} U_B^{\text{ref}} \right\|_F^2. \tag{18}$$

Proof. To derive the alignment error, we simplify the notation by temporarily omitting the dependency on the model index t. Substituting the definition $R_B = U_B^{\text{ref}\top} U_B$ from Eq. (17), we can write:

$$\begin{aligned}
\varepsilon_U &= ||U_B^{\text{ref}} U_B^{\text{ref}\top} U_B - U_B||_F^2 = ||(U_B^{\text{ref}} U_B^{\text{ref}\top} - I_m) U_B||_F^2 \\
&= \text{tr}\left(\left[(U_B^{\text{ref}} U_B^{\text{ref}\top} - I_m) U_B \right]^\top \left[(U_B^{\text{ref}} U_B^{\text{ref}\top} - I_m) U_B \right] \right) \\
&= \text{tr}\left(U_B^\top (U_B^{\text{ref}} U_B^{\text{ref}\top} - I_m)^2 U_B \right) \\
&= \text{tr}\left(U_B^\top (U_B^{\text{ref}} U_B^{\text{ref}\top} U_B^{\text{ref}} U_B^{\text{ref}\top} - 2U_B^{\text{ref}} U_B^{\text{ref}\top} + I_m) U_B \right) \\
&= \text{tr}\left(U_B^\top (U_B^{\text{ref}} U_B^{\text{ref}\top} - 2U_B^{\text{ref}} U_B^{\text{ref}\top} + I_m) U_B \right) \qquad \text{(by using } U_B^{\text{ref}\top} U_B^{\text{ref}} = I\text{)} \\
&= \text{tr}\left(U_B^\top (I_m - U_B^{\text{ref}} U_B^{\text{ref}\top}) U_B \right) = \\
&= \text{tr}\left(U_B{}^\top U_B \right) - \text{tr}\left(U_B^\top U_B^{\text{ref}} U_B^{\text{ref}\top} U_B \right) \qquad \text{(by linearity of trace)} \\
&= r - \left\| U_B^\top U_B^{\text{ref}} \right\|_F^2,
\end{aligned}$$

where $\text{tr}(U_B^\top U_B) = \|U_B\|_F^2 = r$, since U_B has orthonormal columns, and $\text{tr}(U_B^\top U_B^{\text{ref}} U_B^{\text{ref}\top} U_B) = \|U_B^\top U_B^{\text{ref}}\|_F^2$ by the cyclic property of the trace and definition of the Frobenius norm. □

A.3 Optimal Reference Bases

Lemma. *A solution U^* to the quadratic program:*

$$\max_{U\in\mathcal{S}} \left\| U_B^{(t)\top} U \right\|_F^2 = \max_{U\in\mathcal{S}} \text{tr}\left(U^\top U_B^{(t)} U_B^{(t)\top} U \right), \quad \mathcal{S} = \left\{ U \in \mathbb{R}^{m\times T\cdot r} \;\middle|\; U^\top U = I_{T\cdot r} \right\},$$

is given by any orthonormal basis whose columns include the top r eigenvectors corresponding to nonzero eigenvalues of $B^{(t)} B^{(t)\top} \in \mathbb{R}^{m\times m}$ or, equivalently, by the top r left singular vectors of $B^{(t)}$. At the optimum, the objective achieves the maximum value of r, yielding zero alignment error in ε_U as defined in Eq. (18)*.*

Proof. The proof proceeds in two steps. First, we show that any orthonormal basis U_* containing the top r eigenvectors of $U_B^{(t)} U_B^{(t)\top}$ solves the optimization problem. Second, we establish that the eigenvectors of $U_B^{(t)} U_B^{(t)\top}$ are the same as those of $B^{(t)} B^{(t)\top}$.

https://doi.org/10.52202/085713-2066

Step 1: Solving the constrained optimization. By leveraging the method of Lagrange multipliers, we write the augmented objective function:

$$\mathcal{L}(U, \Lambda) = \text{tr}(U^\top U_B^{(t)} U_B^{(t)\top} U) - \text{tr}\left(\Lambda(U^\top U - I_{T\cdot r})\right), \tag{19}$$

where $\Lambda \in \mathbb{R}^{T\cdot r \times T\cdot r}$ is a matrix of Lagrange multipliers. Taking gradients with respect to U and Λ, we obtain:

$$\begin{aligned} \nabla_U \mathcal{L}(U, \Lambda) &= \frac{\partial}{\partial U}\text{tr}(U^\top U_B^{(t)} U_B^{(t)\top} U) - \frac{\partial}{\partial U}\text{tr}(\Lambda(U^\top U - I_{T\cdot r})) \\ &= 2U_B^{(t)} U_B^{(t)\top} U - 2U\Lambda \end{aligned} \tag{20}$$

$$\nabla_\Lambda \mathcal{L}(U, \Lambda) = -\frac{\partial}{\partial \Lambda}\text{tr}(\Lambda(U^\top U - I_{T\cdot r})) = U^\top U - I_{T\cdot r}. \tag{21}$$

Setting the gradients to zero gives the two necessary optimality conditions:

$$U_B^{(t)} U_B^{(t)\top} U_* = U_* \Lambda_*, \tag{22}$$

$$U_*^\top U_* = I_{T\cdot r}. \tag{23}$$

The second condition, Eq. (23), holds by construction: we explicitly choose $U_* \in \mathbb{R}^{m\times T\cdot r}$ to be an orthonormal matrix. Specifically, we define $U_* = [v_1, \ldots, v_r, p_1, \ldots, p_{(T-1)\cdot r}]$, where $\{v_i\}_{i=1}^r$ are orthonormal eigenvectors of $U_B^{(t)} U_B^{(t)\top}$ associated with its nonzero eigenvalues $\lambda_1, \ldots, \lambda_r$, and $\{p_j\}_{j=1}^{(T-1)\cdot r}$ are additional orthonormal vectors chosen to complete the basis.

Next, to verify the first condition reported in Eq. (22), we define the diagonal matrix $\Lambda_* = \text{diag}(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0) \in \mathbb{R}^{T\cdot r \times T\cdot r}$, where the trailing zeros correspond to the eigenvalues associated with the orthogonal complement. Since each v_i is an eigenvector of $U_B^{(t)} U_B^{(t)\top}$ with eigenvalue λ_i, and each p_j lies in the nullspace of that matrix, we have:

$$U_B^{(t)} U_B^{(t)\top} v_i = \lambda_i v_i \quad \text{for } i = 1, \ldots, r, \quad \text{and} \quad U_B^{(t)} U_B^{(t)\top} p_j = 0 \quad \text{for } j = 1, \ldots, (T-1)\cdot r.$$

Therefore:

$$U_B^{(t)} U_B^{(t)\top} U_* = [\lambda_1 v_1, \ldots, \lambda_r v_r, 0, \ldots, 0] = U_* \Lambda_*,$$

which confirms that the first-order condition in Eq. (22) is satisfied.

Next, we substitute this result into the original expression to obtain the optimal value:

$$\mathcal{L}^* = \text{tr}(U_*^\top U_B^{(t)} U_B^{(t)\top} U_*) = \text{tr}(U_*^\top U_* \Lambda_*) = \text{tr}(\Lambda_*) = \sum_{i=1}^r \lambda_i = \text{tr}(U_B^{(t)} U_B^{(t)\top}) = r,$$

proving that the maximum value of the quadratic problem is r, and the corresponding alignment error ε_U for U_* is zero.

Step 2: Equivalence of eigenspaces. Now we show that, given the matrix $B^{(t)} \in \mathbb{R}^{n\times r}$ from a LoRA-based adaptation, if v is an eigenvector of $B^{(t)} B^{(t)\top}$, then v is also, by construction, an eigenvector of $U_B^{(t)} U_B^{(t)\top}$, where $B^{(t)} = U_B^{(t)} \Sigma^{(t)} V_B^{(t)\top}$:

$$\begin{aligned} v \text{ is an eigenvector of } B^{(t)} B^{(t)\top} &\implies B^{(t)} B^{(t)\top} v = \lambda v \\ &\implies U_B^{(t)} \Sigma^{2(t)} U_B^{(t)\top} v = \lambda v \\ &\implies (B^{(t)} B^{(t)\top}) U_B^{(t)} \Sigma^{2(t)} U_B^{(t)\top} v = (B^{(t)} B^{(t)\top}) \lambda v \\ &\implies U_B^{(t)} \Sigma^{2(t)} U_B^{(t)\top} v = \lambda B^{(t)} B^{(t)\top} v. \end{aligned}$$

From the second and fourth rows, we obtain:

$$\lambda B^{(t)} B^{(t)\top} v = \lambda v \tag{24}$$

If $\lambda > 0$, we obtain that v is an eigenvector of $B^{(t)}B^{(t)\top}$ and its corresponding eigenvalue is $= 1$. If $\lambda = 0$, then we have:

$$\begin{aligned} v \text{ is an eigenvector of } B^{(t)}B^{(t)\top}, \lambda = 0 &\implies B^{(t)}B^{(t)\top}v = 0 \\ &\implies U_B^{(t)}\Sigma^{2(t)}U_B^{(t)\top}v = 0 \\ &\implies v^\top U_B^{(t)}\Sigma^{2(t)}U_B^{(t)\top}v = 0 \end{aligned}$$

Since the matrix B has rank r, it has exactly r strictly positive singular values, while all other singular values beyond rank r are zero. Thus, the matrix $\Sigma^{2(t)}$ is positive definite. Hence, it must be:

$$v^\top U_B^{(t)}\Sigma^{2(t)}U_B^{(t)\top}v = 0 \implies U_B^{(t)\top}v = 0 \implies U_B^{(t)}U_B^{(t)\top}v = 0, \tag{25}$$

which confirms that if $\lambda = 0$, then v is an eigenvector of $U_B^{(t)}U_B^{(t)\top}$ with eigenvalue 0. □

Generalization to Multiple Tasks. This result generalizes directly to the multi-task setting, as defined in the main paper, by applying it to the matrix $\mathbf{B} \in \mathrm{R}^{m\times T\cdot r}$ obtained by horizontally stacking the LoRA updates $B^{(t)}$ from all tasks T. Then the optimal reference basis $U_* = U_B^{\text{ref}}$ (as done in our approach) is given by the left singular vectors of $\mathbf{B}$, obtained via SVD. This matrix satisfies both orthonormality and the optimality conditions derived above. Defining $\Lambda_* = \mathrm{diag}(\lambda_1, \ldots, \lambda_{T\cdot r})$, where λ_i are the eigenvalues of $\mathbf{B}\mathbf{B}^\top$, guarantees perfect reconstruction and zero alignment error for all tasks.

A.4 Generalization Beyond the $T \cdot r \leq m, n$ Assumption

The derivations in Secs. A.1 to A.3 assumed that the total LoRA rank $T \cdot r$ is less than both m and n. We now show that this assumption is not necessary, and that **the reconstruction error remains zero even when $T \cdot r > m$ or $T \cdot r > n$.**

Intrinsic rank. When $T \cdot r > m$ (for B) or $T \cdot r > n$ (for A), stacking the LoRA matrices still yields reference bases with intrinsic dimensions:

$$d_U = \mathrm{rank}([B^{(1)}, \ldots, B^{(T)}]) \leq m, \qquad d_V = \mathrm{rank}([A^{(1)}, \ldots, A^{(T)}]^\top) \leq n,$$

since the number of linearly independent directions cannot exceed the number of rows or columns

We can therefore replace the reference bases $U_B^{\text{ref}} \in \mathbb{R}^{m\times T\cdot r}$ and $V_A^{\text{ref}} \in \mathbb{R}^{n\times T\cdot r}$ with truncated orthonormal bases:

$$U_B^{\text{ref}} \in \mathbb{R}^{m\times d_U}, \qquad V_A^{\text{ref}} \in \mathbb{R}^{n\times d_V},$$

whose columns span the full LoRA update space.

Least-Squares Solution. Rewriting the least-squares problem (Eq. (16)) in terms of the truncated U_B^{ref} gives:

$$R_B^{(t)} = \underset{R\in\mathbb{R}^{d_U\times r}}{\arg\min} \|U_B^{\text{ref}}R - U_B^{(t)}\|_F^2.$$

Following the same derivation as in Sec. A.1, the optimal solution is

$$R_B^{(t)} = (U_B^{\text{ref}})^\top U_B^{(t)},$$

with an analogous expression for $Q_A^{(t)}$.

Alignment Error. Substituting this solution into the error expression of Sec. A.2 shows that the optimal alignment error remains

$$\varepsilon_U = r - \|(U_B^{(t)})^\top U_B^{\text{ref}}\|_F^2,$$

which achieves zero when U_B^{ref} spans the column space of all $B^{(t)}$. Hence, the theoretical guarantees extend unchanged to the case $T \cdot r > m, n$.

https://doi.org/10.52202/085713-2066

B Computational Complexity Analysis

Iso-C Complexity. To assess the time complexity of our approach, we begin by analyzing that of Iso-C [28], to establish a baseline for comparison. For each layer, the Iso-C procedure can be broken down into the following steps:

- *LoRA → full space*: Compute $\Delta^{(t)} = B^{(t)}A^{(t)}$ for $t = 1, \ldots, T$, given the matrices $B^{(t)}$ and $A^{(t)}$. The resulting complexity is $\mathcal{O}(T \cdot r \cdot n \cdot m)$.
- *Summation*: Applying task arithmetic in the full space, $\Delta_{\text{TA}} = \sum_t^T \Delta^{(t)}$, involves a cost of $\mathcal{O}(T \cdot n \cdot m)$.
- *SVD computation*: Computing the decomposition $\Delta_{\text{TA}} = U\Sigma V^\top$ for an $m \times n$ matrix has a complexity of $\mathcal{O}(m^2 \cdot n + n^3)$ [40].
- *Isotropization*: The final step, $\Delta_{\text{Iso-C}} = U\Sigma_{\text{avg}}V^\top$, is dominated by $\mathcal{O}(m^2 \cdot n)$.

Overall, the total cost of Iso-C is dominated by $\mathcal{O}(T \cdot r \cdot n \cdot m + m^2 \cdot n + n^3)$. Assuming that $m = n$, then the time complexity of Iso-C is approximately cubic: $\mathcal{O}(n^3 + T \cdot r \cdot n^2)$ with respect to the number of features.

KnOTS Complexity. Secondly, we analyze the computational cost of the **KnOTS** method [42]:

- *LoRA → full space*: As in Iso-C, this step has complexity $\mathcal{O}(T \cdot r \cdot n \cdot m)$.
- *Concatenation*: Stacking all weight matrices $\Delta W = \left[\Delta W^{(1)}, \ldots, \Delta W^{(T)}\right]$ as block columns of a global matrix has a time complexity of $\mathcal{O}(T \cdot n \cdot m)$.
- *SVD computation*: This is performed on a matrix of size $m \times (n \cdot T)$, resulting in a complexity of $\mathcal{O}(n^2 \cdot T^2 \cdot m + m^3)$, by making use of the transpose trick[2].
- *Merge*: Assuming simple task arithmetic is performed in the $V^\top$ space, T blocks of n columns each are summed, yielding a complexity of $\mathcal{O}(T^2 \cdot r \cdot n)$ to compute $V^\top_{\text{merge}}$.
- *Reconstruction*: The final step $\Delta_{\text{KNOTS}} = U\Sigma V^\top_{\text{merge}}$ has a complexity of $\mathcal{O}(m \cdot n \cdot T \cdot r + T \cdot r \cdot n)$.

Overall, the total cost of KNOTS is dominated by $\mathcal{O}(m^3 + T \cdot n(2r \cdot m + m + n \cdot T \cdot m + T \cdot r + r))$. Assuming $m = n$, the time complexity simplifies to $\mathcal{O}(n^3T^2)$. Compared to Iso-C, the time complexity of KNOTS remains cubic with respect to the number of features. However, it includes an additional T^2 factor that scales quadratically with the number of tasks being merged.

TSV Complexity. Then we analyze the cost of TSV [12]:

- *LoRA → full space*: As previously, this step has complexity $\mathcal{O}(T \cdot r \cdot n \cdot m)$.
- *SVD Computation*: In this step SVD is performed on T matrices of size $m \times n$ resulting in a complexity of $\mathcal{O}(T \cdot (m^2 \cdot n + n^3))$.
- *Concatenation*: Stacking the first k components of left and right singular vectors for all tasks results in $\mathcal{O}(T \cdot k(n + m))$.
- *Global SVD Computation*: The SVD performed on the stacked matrices $n \times m$ requires $\mathcal{O}(2 \cdot (m^2 \cdot n + n^3))$.
- *Obtaining orthogonal matrices*: This step requires $\mathcal{O}(2 \cdot m^2 \cdot n)$.
- *Merge*: The final merge has a complexity of $\mathcal{O}(m \cdot n \cdot T \cdot r + T \cdot r \cdot n)$.

The overall cost of TSV is thus $\mathcal{O}(T{\cdot}r{\cdot}m{\cdot}n{+}T{\cdot}r{\cdot}n{+}T{\cdot}m^2{\cdot}n{+}T{\cdot}n^3{+}T{\cdot}k{\cdot}m{+}T{\cdot}k{\cdot}n{+}m^2{\cdot}n{+}n^3)$. Assuming that $m = n$ and $T, r, k \ll n$, the computational cost is dominated by $\mathcal{O}(T \cdot n^3)$.

Core Space Complexity. Finally, we analyze the cost of our approach in **Core Space**.

[2] $\text{SVD}(P^\top) = U'\Sigma'V'^\top \rightarrow P = (U'\Sigma'V'^\top)^\top = V'\Sigma U'^\top$, thus, if $r \ll n$, this will reduce the number of operations. We will apply the transpose trick throughout.

- *Stacking $A^{(t)}$ and $B^{(t)}$*: Stacking two sequences of T matrices – one with each matrix of shape $r \times n$ and the other with each matrix of shape $m \times r$ – results in a cost of $\mathcal{O}(T \cdot r(n+m))$.
- *SVD computation*: The stacked global A matrix has shape $(T \cdot r) \times n$; hence, the cost of its SVD is $\mathcal{O}(n^2 \cdot (T \cdot r) + (T \cdot r)^3)$, by using the transpose trick. The stacked B matrix has shape $m \times (T \cdot r)$, with a cost of $\mathcal{O}(m^2 \cdot (T \cdot r) + (T \cdot r)^3)$. The overall cost of this step is $\mathcal{O}((m^2+n^2) \cdot (T \cdot r) + 2 \cdot (T \cdot r)^3)$.
- *Low-rank loop*: In the optimized version of the low-rank loop, we only compute the matrix multiplication to obtain the aligned matrices. In this case the total cost is $\mathcal{O}(T \cdot (Tmr^2 + Tr^2n + T^2r^3))$ (using the optimal matrix multiplication order $(U_B^{\text{ref}}B)(AV_A^{\text{ref}})$). Assuming that $T, r << n, m$, the cost is dominated by $\mathcal{O}(T^2r^2(m+n))$.
- *Merge:* Assuming simple task arithmetic is performed in the aligned Core Space, the cost is $\mathcal{O}(T^3 \cdot r^2)$.
- *Isotropization:* Optionally, Iso-C can be applied in the Core Space; since the Core Space is defined within a square matrix of dimension $T \cdot r \times T \cdot r$, this step adds an additional time complexity of $\mathcal{O}(T^3r^3)$.
- *Reconstruction:* The final step requires $\mathcal{O}(m \cdot T^2 \cdot r^2 + m \cdot T \cdot r \cdot n)$.

To sum up, our approach involves:

$$\mathcal{O}(\underbrace{Tr(n+m)}_{\text{Stacking}} + \underbrace{Tr(m^2+n^2) + 2(Tr)^3}_{\text{SVD refs.}} + \underbrace{(T^2r^2(m+n))}_{\text{Low-rank loop.}} + \underbrace{Tr(T^2r + Trm + mn))}_{\text{Merge \& Rec.}} =$$

$$\mathcal{O}(Tr(m+n+2m^2+2n^2+mn+r^2) + T^2r^2(2m+n+T) + T^3r^3)$$

If we assume that $m = n$, the total time complexity simplifies to:

$$\mathcal{O}(Tr(2n+5n^2+r^2) + T^2r^2(3n+T) + T^3r^3), \tag{26}$$

which, if we assume $T, r \ll n$, is dominated by:

$$\mathcal{O}(Trn^2) \tag{27}$$

C Additional Analysis

C.1 High subspace alignment leads to lower interference

In this Section, we experimentally show that merging in Core Space reduces interference when merging models. We follow [53, 28] and measure the interference as the L1 distance between the final embeddings of task-specific models and the merged one. We compare the interference when merging with TSV + Iso-C in Full Space versus Core Space. For each dataset, we collect the activations from the final layer (*i.e.*, the projection to a common vision-language space) of both the task-specific model and the merged model. We present the average distance across all the samples in the test set. We observe lower interference when merging in Core Space, highlighting its effectiveness. Note that Full Space merging, which causes higher interference, also exhibits higher SAR in Sec. 5.2.

C.2 Rank of the merged update matrices

Consider $T = 8$ ViT-B/32 models fine-tuned with LoRA of rank $r = 16$. The merged update matrices ΔW resulting from different merging methods and spaces can have different effective ranks $r_{\Delta W}$. Table 6 reports the average rank of ΔW across all layers.

In most cases, the target rank of ΔW is equal to $Tr = 128$. The only exception is merging with TIES in Full space, where for weight matrices $W \in \mathbb{R}^{m \times n}$ the effective rank approaches the dimensionality of the matrices $d = \min(m, n) = 768$. This phenomenon arises because TIES performs trimming on the reconstructed weight matrices $\Delta W_t = BA$, which destroys the low-rank structure. In contrast, both Core and KnOTS operate directly in a constrained Tr-dimensional space, ensuring that the merged ΔW maintains the intended rank.

https://doi.org/10.52202/085713-2066

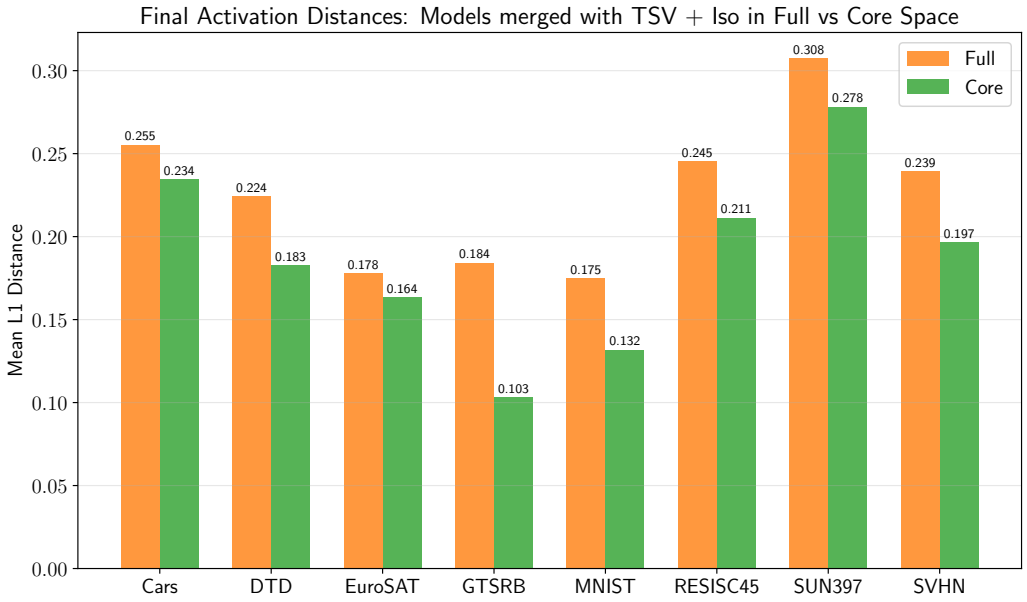

Figure 6: Mean L1 distance between the final embeddings of task-specific models and the merged one using TSV + Iso in Full and Core Space. We used ViT-B/16 model.

Table 6: Average rank $r_{\Delta W}$ of merged update matrices ΔW obtained by merging 8 ViT-B/32 LoRA models with $r = 16$ (so $Tr = 128$).

Merging Space	**TA**	**TSV**	**TIES**
Full	128.00	128.00	766.25
KnOTS	128.00	128.00	128.00
Core	128.00	128.00	128.00

D Additional Experiment Details

Licenses of Used Datasets and Models

In our research, we employed publicly available datasets and models, each governed by specific licenses. Below, we outline the sources and associated licenses for each:

- **KnOTS LoRA Checkpoints** [42]: The KnOTS repository, which provides LoRA-adapted model checkpoints and training scripts, is licensed under the MIT License. This permissive license allows for reuse and modification with proper attribution.
- **Cars196** [22]: The Cars196 dataset is available for non-commercial research purposes. Specific licensing details are not explicitly provided.
- **Describable Textures Dataset (DTD)** [7]: The DTD is made available to the computer vision community for research purposes. The dataset is licensed under the Creative Commons Attribution 4.0 License (CC BY 4.0).
- **EuroSAT** [14]: The EuroSAT dataset is licensed under the MIT License.
- **German Traffic Sign Recognition Benchmark (GTSRB)** [41]: The GTSRB dataset is licensed under the Creative Commons Zero (CC0) Public Domain Dedication.
- **MNIST** [23]: The MNIST dataset is publicly available for research purposes. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **NWPU-RESISC45** [5]: The NWPU-RESISC45 dataset is licensed under the Creative Commons Attribution 4.0 License (CC BY 4.0).

- **SUN397** [49]: The SUN397 dataset is available for research purposes only. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **Street View House Numbers (SVHN)** [33]: The SVHN dataset is available for non-commercial use only.
- **Stanford Natural Language Inference (SNLI)** [3]: The SNLI dataset is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).
- **Multi-Genre Natural Language Inference (MNLI)** [47]: The MNLI dataset is released under the Open American National Corpus (OANC) license, which permits free use, modification, and sharing under permissive terms.
- **Sentences Involving Compositional Knowledge (SICK)** [30]: The SICK dataset is distributed under a Creative Commons Attribution-NonCommercial-ShareAlike license.
- **Question Natural Language Inference (QNLI)** [45]: The QNLI dataset is part of the GLUE benchmark. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **Recognizing Textual Entailment (RTE)** [45]: The RTE dataset is part of the GLUE benchmark. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **SciTail** [20]: The SciTail dataset is licensed under the Apache License 2.0.

D.1 Experimental Environment

The language experiments with Llama 3 8B were performed with a single 48G NVIDIA L40S. In contrast, the more affordable vision experiments were executed using a single 16G NVIDIA RTX 4080. To keep things fair, the reported times for the language experiments all refer to experiments performed on the same machine.

Our implementation builds directly on the KnOTS codebase [42] and uses the exact LoRA checkpoints they released. For full details on the original training and adaptation procedures, please refer to [42].

D.2 Hyperparameter Search

To find optimal hyperparameters for each model, we adopt the widely used validation holdout strategy [42, 28, 12, 50]. Specifically, we perform a linear search for hyperparameters on the validation set, starting from a defined minimum value and incrementally increasing it until performance declines, indicating the optimal range. The identified optimal hyperparameters are then applied to the test set. We use the following search settings:

- Scaling factor α starts at 0.1, increasing in increments of 0.1. This is used for every approach.
- The top-K parameter for TIES and DARE-TIES begins at 10 and increases in increments of 10.
- The pruning factor p for DARE-TIES starts at 0.1 and increases in increments of 0.1.
- For CART, the pruning rank is searched over the set $\{0.04, 0.08, 0.16, 0.32\}$, following the methodology of the original paper. Additionally, CART includes an extra scaling factor λ in its merging formulation. Specifically, the merged weights are computed as $W_{\text{merged}} = W_0 + \alpha(\theta_{\text{avg}} + \lambda \sum_{t=1}^{T} \bar{\tau}_t)$, where θ_{avg} denotes the average of the updates and $\bar{\tau}_t$ represents the centered task vector for task t. For further details, we refer the reader to [6].

In Tab. 7, we report the parameters used for the various merging methods in Core Space across all backbones. Note that we search for the optimal parameters for all methods across all spaces to maintain fairness. For the natural-language inference experiments, we omit CART as its hyperparameter tuning proved prohibitively expensive and exclude +Iso-C variants, as they consistently degraded performance.

 https://doi.org/10.52202/085713-2066

Table 7: Optimal hyperparameters for Core-Space merging on each backbone and merging strategy, including +Iso-C variants.

Backbone	Merging Method	α	Top-K	Pruning p	CART rank	λ
ViT-B/32	TIES-Core	0.6	10	-	-	-
	TIES-Core+Iso-C	2.0	30	-	-	-
	DARE-TIES-Core	0.6	10	0.1	-	-
	DARE-TIES-Core+Iso-C	2.0	30	0.1	-	-
	TSV-Core	0.2	-	-	-	-
	TSV-Core+Iso-C	0.9	-	-	-	-
	CART-Core	0.4	-	-	0.32	5.8
	CART-Core+Iso-C	0.7	-	-	0.04	2.6
	Iso-C-Core	0.9	-	-	-	-
ViT-L/14	TIES-Core	0.4	10	-	-	-
	TIES-Core+Iso-C	2.4	20	-	-	-
	DARE-TIES-Core	0.4	10	0.1	-	-
	DARE-TIES-Core+Iso-C	2.4	20	0.2	-	-
	TSV-Core	0.2	-	-	-	-
	TSV-Core+Iso-C	0.9	-	-	-	-
	CART-Core	0.1	-	-	0.04	6.5
	CART-Core+Iso-C	1.0	-	-	0.08	2.0
	Iso-C-Core	0.9	-	-	-	-
Llama 3 8B	TIES-Core	1.1	80	-	-	-
	DARE-TIES-Core	1.1	80	0.1	-	-
	TSV-Core	0.5	-	-	-	-
	Iso-C-Core	2.8	-	-	-	-

D.3 Code Availability and Pseudocode

Our Core Space merging implementation is released at `https://github.com/apanariello4/core-space-merging`.

Listing 1 gives PyTorch-style pseudocode illustrating how to apply any merging strategy within the Core Space framework.

E Additional Results

E.1 Per-task evaluation in vision setting for ViT-L/14

We provide in Tab. 8 per-task results vision model merging using the ViT-L/14 backbone. Similarly to what we observed for other backbones, in this case, performing the merging in Core Space yields consistent improvements across all methods, resulting in new state-of-the-art results.

E.2 Experiments with Heterogeneous Ranks

In the main paper, we discussed that Core Space merging naturally extends to the heterogeneous rank setting (Tab. 9). Here we provide additional details.

When tasks are fine-tuned with different LoRA ranks, we horizontally and vertically stack the $B^{(t)}$ and $A^{(t)}$ matrices across tasks as usual. The resulting aggregate matrices have rank equal to the dimension of the union of all task subspaces. Performing SVD on these aggregates yields orthonormal reference bases U_B^{ref} and V_A^{ref} that span the combined subspaces, regardless of how individual task ranks vary.

Projection into these reference bases followed by task-specific alignment (see Sec. A.1) guarantees that reconstruction is lossless. This explains why the results in Tab. 9 show that heterogeneous ranks incur no additional degradation in performance under our framework, whereas baselines that lack such alignment struggle with mismatched subspace dimensions.

Listing 1 Basic PyTorch pseudocode for model merging in Core Space.

```
from torch.linalg import svd

A_list, B_list = ..., ...

r, n = A_list[0].shape
m, _ = B_list[0].shape

A_stack = torch.cat(A_list, dim=0)  # (T*r, n)
B_stack = torch.cat(B_list, dim=1)  # (m, T*r)

# Calculate reference bases
Vh_A_ref = svd(A_stack, full_matrices=False)[2]  # (T*r, n)
U_B_ref = svd(B_stack, full_matrices=False)[0]   # (m, T*r)

M_list = []
for A, B in zip(A_list, B_list):
     M_aligned = (U_B_ref.T @ B) @ (A @ Vh_A_ref.T)
     M_list.append(M_aligned)

if merge_strategy == 'TA':
        M_merged = torch.stack(M_list).sum(dim=0)
elif merge_strategy == 'ties':
        M_merged = ties_merging(M_list)
elif merge_strategy == '...':
        M_merged = ...

# Reconstruct delta W
delta_W = U_B_ref @ M_merged @ Vh_A_ref
```

E.3 Extension to VeRA

We also evaluated the applicability of Core Space merging beyond LoRA, specifically on VeRA [21] (Tab. 10). In VeRA, the decomposition $\Delta W = \Lambda_b B \Lambda_d A$ differs structurally from LoRA since A and B are fixed random matrices, and only the scaling vectors Λ_b, Λ_d are trainable.

To apply our method, we absorb the scaling vectors into the low-rank matrices:

$$\tilde{B} = \Lambda_b B, \qquad \tilde{A} = \Lambda_d A,$$

and then treat $(\tilde{A}, \tilde{B})$ as if they were standard LoRA components. Since the subsequent steps (stacking, SVD, projection, and alignment) are agnostic to how A and B were obtained, the derivations in Sec. A apply without modification.

The empirical results in Tab. 10 confirm this reasoning: Core Space merging consistently outperforms other approaches even in the VeRA setting, validating that the framework is general to low-rank adaptation methods beyond LoRA.

https://doi.org/10.52202/085713-2066

Table 8: Accuracies of merged models normalized against fine-tuned models on the vision datasets with ViT-L/14.

Method	Space	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC	SUN397	SVHN	Avg (Δ Acc)
Abs. Accuracy		99.70	70.00	98.50	97.20	99.50	95.70	79.60	97.7	-
TA	Full	80.01	79.50	65.59	59.98	82.20	79.55	86.71	64.74	74.79 (+0.00)
TIES	Full	79.65	78.28	64.43	61.10	83.82	79.42	87.45	69.94	75.51 (+0.00)
	KnOTS	82.47	80.26	64.65	68.85	88.48	82.37	88.18	76.63	**78.99** (+3.48)
	Core	81.94	80.03	65.78	68.94	87.44	81.85	88.48	73.75	78.53 (+3.02)
DARE-TIES	Full	79.70	78.82	64.99	60.63	83.92	79.32	87.07	69.84	75.53 (+0.00)
	KnOTS	80.41	79.65	64.84	62.95	85.33	79.73	87.55	71.00	76.43 (+0.90)
	Core	82.97	80.03	65.66	68.34	87.75	82.48	88.80	75.88	**78.99** (+3.46)
TSV	Full	82.38	80.11	66.12	68.18	85.46	83.02	87.89	70.76	77.99 (+0.00)
	KnOTS	80.17	79.95	66.94	60.24	83.76	79.35	86.85	65.59	75.36 (-2.63)
	Core	82.10	79.80	67.05	66.92	87.50	82.32	87.71	71.89	**78.16** (+0.17)
CART	Full	91.88	88.53	75.51	80.88	68.99	92.77	88.17	64.81	81.44 (+0.00)
	KnOTS	79.65	79.73	64.39	58.28	80.42	78.52	86.52	63.24	73.84 (-7.60)
	Core	86.02	86.33	72.39	82.82	91.03	85.93	88.46	72.67	**83.21** (+1.77)
TIES +Iso-C	Full	84.11	83.07	74.94	76.66	92.88	87.88	88.58	74.61	82.84 (+0.00)
	KnOTS	86.44	88.23	78.74	78.94	94.27	87.94	88.62	72.24	84.43 (+1.59)
	Core	91.13	90.58	79.53	87.67	86.47	90.46	90.05	72.28	**86.02** (+3.18)
DARE-TIES +Iso-C	Full	83.25	81.78	73.03	77.25	86.81	86.78	88.24	74.44	81.45 (+0.00)
	KnOTS	87.32	87.78	74.12	81.50	94.21	89.48	88.88	70.87	84.27 (+2.82)
	Core	90.84	91.12	79.15	88.14	90.19	90.73	89.92	72.05	**86.52** (+5.07)
TSV +Iso-C	Full	86.44	89.07	82.49	84.68	90.76	90.03	87.99	67.98	84.93 (+0.00)
	KnOTS	88.47	90.58	77.69	83.91	87.48	90.00	88.76	67.49	84.30 (-0.63)
	Core	91.54	91.34	80.24	86.79	87.39	91.51	89.59	71.30	**86.21** (+1.28)
CART +Iso-C	Full	88.78	90.43	78.59	87.04	91.96	90.96	89.32	75.18	86.53 (+0.00)
	KnOTS	88.72	89.60	80.77	79.84	87.18	89.50	88.38	68.07	84.01 (-2.52)
	Core	92.08	92.48	81.22	88.96	89.80	91.97	89.81	73.56	**87.49** (+0.96)
Iso-C	Full	86.83	86.94	80.65	77.99	92.09	87.88	88.50	68.69	83.70 (+0.00)
	KnOTS	88.27	89.75	78.36	85.41	91.65	90.93	88.85	70.97	85.52 (+1.82)
	Core	91.23	90.28	80.28	85.29	89.71	90.96	89.58	70.66	**86.00** (+2.30)

Table 9: Normalized accuracies of merged models on the vision tasks with ViT-B/32 with LoRA mixed ranks 16 (Cars, EuroSAT, MNIST, SVHN) and 64 (DTD, GTSRB, RESISC, SUN397).

Method	Space	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC	SUN397	SVHN	Avg (Δ Acc)
Abs. Accuracy		74.00	68.03	99.00	98.00	99.30	93.85	70.85	96.20	-
TA	-	81.97	65.83	47.06	51.66	57.18	72.91	89.75	48.37	64.34 (+0.00)
TIES	Full	80.94	64.74	42.01	53.91	55.41	72.16	89.76	49.09	63.50 (+0.00)
	KnOTS	82.70	67.08	49.01	55.00	57.39	73.74	90.14	45.97	65.13 (+1.63)
	Core	79.74	67.71	41.38	71.86	69.59	75.02	91.43	67.97	70.59 (+7.09)
DARE-TIES	Full	80.81	65.60	41.11	54.89	56.14	72.45	89.74	49.70	63.81 (+0.00)
	KnOTS	82.98	67.08	45.57	56.34	62.34	75.65	90.06	53.53	66.69 (+2.88)
	Core	80.98	68.80	44.26	68.07	64.43	74.45	91.44	62.99	69.43 (+5.62)
TSV	Full	80.98	67.94	50.95	59.10	64.65	75.82	90.76	53.37	67.95 (+0.00)
	KnOTS	82.51	65.91	43.66	48.98	60.52	71.47	89.22	51.29	64.20 (-3.75)
	Core	80.46	69.51	51.44	58.33	61.03	76.70	89.37	52.43	67.41 (-0.54)
TIES +Iso-C	Full	78.94	69.90	54.02	53.13	66.83	75.11	90.64	46.63	66.90 (+0.00)
	KnOTS	81.42	68.49	60.19	45.09	56.70	70.83	90.43	40.11	64.16 (-2.74)
	Core	79.32	75.84	58.66	64.09	80.14	76.76	90.52	55.13	72.56 (+5.66)
DARE-TIES +Iso-C	Full	80.33	70.76	60.98	51.48	61.58	74.16	90.41	47.26	67.12 (+0.00)
	KnOTS	82.11	67.71	58.92	41.44	57.92	69.69	90.37	38.79	63.37 (-3.75)
	Core	78.75	75.37	58.47	64.98	80.83	77.00	90.49	55.53	72.68 (+5.56)
TSV +Iso-C	Full	76.46	73.65	61.02	56.41	69.02	74.04	90.01	49.12	68.72 (+0.00)
	KnOTS	78.88	72.79	66.59	54.61	78.83	72.21	89.03	50.28	70.40 (+1.68)
	Core	79.30	78.26	63.94	58.11	77.28	74.40	89.85	50.97	71.51 (+2.79)
Iso-C	Full	78.48	75.61	63.37	63.05	77.14	77.10	90.22	51.50	72.06 (+0.00)
	KnOTS	78.92	76.31	56.19	63.50	75.11	77.08	90.53	52.44	71.26 (-0.80)
	Core	81.17	79.20	59.71	70.86	81.90	78.93	90.81	56.60	74.90 (+2.84)

Table 10: Normalized accuracies of merged models on the vision tasks with ViT-B/32 with VeRA rank 16.

Method	Space	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC	SUN397	SVHN	Avg (Δ Acc)
Abs. Accuracy		62.79	57.07	96.55	90.85	98.60	88.50	62.79	93.10	-
TA	-	95.78	77.73	47.26	39.77	49.10	70.61	100.74	37.29	64.78 (+0.00)
TIES	Full	95.53	77.73	44.76	39.07	48.25	69.92	100.58	35.62	63.93 (+0.00)
	KnOTS	96.64	77.91	50.13	39.61	50.46	70.52	100.85	35.93	65.26 (+1.33)
	Core	97.06	77.73	44.30	41.98	50.35	71.63	100.44	38.95	65.31 **(+1.39)**
DARE-TIES	Full	94.44	77.73	47.26	42.09	49.82	69.37	100.05	38.28	64.88 (+0.00)
	KnOTS	96.52	77.54	52.13	40.88	50.46	70.25	100.06	37.24	65.63 **(+0.75)**
	Core	97.09	77.82	44.27	42.02	50.41	71.75	100.48	38.97	65.35 (+0.47)
TSV	Full	93.35	77.54	45.11	37.44	53.50	69.40	99.70	33.85	63.74 (+0.00)
	KnOTS	92.73	78.19	54.85	38.77	56.20	69.10	98.82	34.02	65.33 (+1.59)
	Core	93.75	76.51	60.15	37.94	54.32	68.38	98.77	34.63	65.56 **(+1.82)**
TIES +Iso-C	Full	94.79	77.26	46.84	36.11	48.78	68.19	100.81	34.03	63.35 (+0.00)
	KnOTS	94.81	77.26	47.33	35.49	48.67	68.10	100.67	33.74	63.26 (-0.09)
	Core	95.43	77.63	50.79	37.14	50.61	69.06	100.62	33.77	64.38 **(+1,03)**
DARE-TIES +Iso-C	Full	93.82	76.98	46.34	36.80	49.47	68.26	100.44	34.69	63.35 (+0.00)
	KnOTS	94.88	77.17	47.37	35.52	48.64	68.13	100.66	33.75	63.27 (-0.08)
	Core	94.14	77.26	53.01	37.45	52.90	68.13	100.09	31.80	64.35 **(+1.00)**
TSV +Iso-C	Full	93.37	76.70	48.29	35.92	51.47	68.11	100.30	34.35	63.56 (+0.00)
	KnOTS	92.58	75.30	52.51	36.47	58.16	67.36	99.83	35.43	64.71 (+1.15)
	Core	93.57	76.89	63.33	39.27	57.53	68.63	99.18	34.07	66.56 **(+3.00)**
Iso-C	Full	93.50	77.26	52.05	37.42	49.49	68.06	100.62	34.22	64.08 (+0.00)
	KnOTS	94.59	77.35	47.76	35.52	48.73	68.27	100.52	33.68	63.30 (-0.78)
	Core	91.77	75.77	63.60	38.53	58.53	67.81	97.67	36.50	66.27 **(+2.19)**

https://doi.org/10.52202/085713-2066