PRESERVING ANCHORED CORES IN TASK-VECTORS FOR MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging has emerged as a training-free alternative to multi-task learning, aiming to combine multiple task-specific fine-tuned models into a single multi-task model. Most existing model merging approaches follow the *Task Arithmetic* paradigm, which decomposes fine-tuned weights into pre-trained parameters and task vectors, and performs merging exclusively in the task-vector space. The effectiveness of this paradigm implicitly relies on the assumption that task-specific knowledge is encoded solely within task vectors. In this paper, we argue that this assumption generally does not hold due to the intrinsic task preferences of pre-trained models. Specifically, we identify **Load-Bearing Wall (LBW) dimensions**, namely some task-critical knowledge that remains embedded in the pre-trained weights rather than being fully transferred into task vectors. We characterize LBW dimensions from both scalar-weight and subspace perspectives, thereby covering the major paradigms of existing model merging methods. Our analysis reveals that, by ignoring LBW dimensions, task-vector-based approaches fail to fully resolve task conflicts and may inadvertently damage task-specific knowledge encoded in the pre-trained model, leading to degradation. To address this issue, we propose **P**reserve **A**nchored **C**ores in **T**ask-vectors (`PACT`), which preserves the anchored task-specific cores (i.e., LBW components) within task vectors by aligning their orthogonal complements with the subspace of the pre-trained weights. These aligned subspace components are then removed from the task vectors before applying existing model merging algorithms. Furthermore, we develop an efficient variant based on randomized SVD to improve scalability. As `PACT` addresses model merging from a complementary perspective, it can be seamlessly integrated with existing methods. Extensive experiments across vision and NLP benchmarks demonstrate that `PACT` consistently improves mainstream model merging methods by considerable margins and establishes new state-of-the-art performance.

1 INTRODUCTION

Pre-trained models (PTMs) have become the foundation of modern machine learning systems, providing strong general-purpose representations across a wide range of domains and tasks (Carion et al., 2020; Radford et al., 2021; Caron et al., 2021). To achieve high performance on downstream applications, PTMs are typically specialized through task-specific fine-tuning (Wortsman et al., 2022b; Ilharco et al., 2022b). However, extending such specialization to the multi-task setting often requires complex optimization procedures, substantial computational resources, and access to large amounts of task-specific training data (Wei et al., 2024). Model merging has recently emerged as an attractive training-free alternative, aiming to combine multiple task-specific fine-tuned models into a single multi-task model without additional training (Li et al., 2023; Yang et al., 2026).

Existing model merging methods can be broadly categorized into scalar weight-based and subspace-based approaches (Yang et al., 2026; Ruan et al., 2025). Scalar weight-based methods (Ilharco et al., 2022a; Yadav et al., 2023; Nguyen et al., 2025) treat individual parameter coordinates as the basic unit of task knowledge and perform coordinate-wise operations on model weights or task vectors. In contrast, subspace-based approaches (Stoica et al., 2025; Gargiulo et al., 2025; Marczak et al., 2025) represent task knowledge as a collection of important directions in parameter space and merge models by identifying, preserving, or recombining task-relevant subspaces. Despite their

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

methodological differences, both paradigms operate primarily in the space of *task vectors*, namely the parameter updates induced by fine-tuning.

The widespread adoption of task vectors is largely motivated by the insights of Task Arithmetic (Ilharco et al., 2022a), which showed that task vectors corresponding to different tasks are often approximately orthogonal. This observation suggests that task-specific knowledge is largely encoded in task vectors and can therefore be manipulated independently. Consequently, most existing model merging methods implicitly assume that task vectors provide a sufficiently complete representation of task-specific knowledge, while the pre-trained model serves merely as a shared initialization point. In this paper, we challenge this assumption by asking a simple yet fundamental question:

Is task-specific and -critical knowledge fully captured by task vectors?

Our answer is negative. Through a series of controlled experiments spanning both scalar weight-based and subspace-based merging paradigms, we identify a previously overlooked phenomenon: a portion of task-specific and -critical knowledge remains anchored in the pre-trained model itself. We refer to such dimensions as **Load-Bearing Wall (LBW)** dimensions. Although these dimensions often exhibit only small changes during fine-tuning and therefore contribute little to the task vector, they play a disproportionately important role in maintaining task performance. As a result, task vectors alone provide an incomplete representation of task-specific knowledge.

This observation has important implications for model merging. Since existing approaches operate exclusively on task vectors, they are unable to explicitly protect LBW dimensions during merging. Consequently, updates introduced by other task vectors may overwrite or distort the pre-trained structures on which a task critically depends, leading to performance degradation after merging. Through destructive perturbation experiments, we verify both the existence of LBW components and their vulnerability during model aggregation, as illustrated in Figure 1.

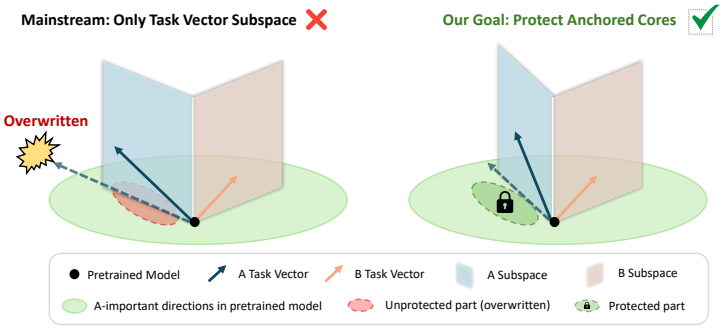


Figure 1: Illustration of LBW dimensions in pre-trained model and its impact on model merging.

Building upon these findings, we propose PACT, a data-free framework for protecting task-specific LBW knowledge during model merging. The key idea is to identify and preserve the task-anchored components associated with LBW dimensions while filtering out task-vector components that interfere with the corresponding pre-trained subspace. Since PACT operates directly on task vectors, it can be seamlessly integrated with existing scalar weight-based and subspace-based merging algorithms. To further improve scalability, we develop an efficient variant based on randomized SVD with substantially reduced computational complexity. Our contributions are summarized as follows:

- We revisit a fundamental assumption underlying Task Arithmetic and investigate whether task vectors fully capture task-specific knowledge. We empirically identify the existence of LBW dimensions and demonstrate their critical role in model merging.
- We propose PACT, a data-free framework that protects task-specific LBW knowledge from interference during model merging. We further develop an efficient variant with improved scalability through randomized SVD.
- Extensive experiments on vision and language benchmarks, covering both full fine-tuning and LoRA fine-tuning settings, demonstrate that PACT consistently improves mainstream model merging methods and establishes new state-of-the-art (SOTA) performance.

2 Related Work

Model merging aims to integrate multiple task-specific fine-tuned expert models into a single multi-task model without accessing their original training data, thereby avoiding the computational overhead and data privacy concerns of traditional multi-task learning. The task vector lies at the core of most static merging methods. Task Arithmetic (Ilharco et al., 2022a) performs multi-task merging by scaling and summing task vectors, establishing the plug-and-play merging paradigm. TIES-Merging (Yadav et al., 2023) sequentially applies magnitude pruning, sign disambiguation, and disjoint aggregation to alleviate parameter conflicts. Consensus Merging (Wang et al., 2024) eliminates "selfish" and "catastrophic" weights that benefit only a single task at the expense of others. Concrete Merging (Tang et al., 2023) uses meta-learning to generate binary masks that suppress conflicting parameters. AWD (Xiong et al., 2024) reduces interference by promoting orthogonality among task vectors. DARE (Yu et al., 2024) randomly drops part of the parameters as a dropout-style preprocessing step. MAP (Li et al., 2025) leverages a second-order Taylor expansion and linear regression to estimate the Hessian, providing loss-based guidance for merging. Both MetaGPT (Zhou et al., 2024) and TATR (Sun et al., 2025) explicitly model the loss gap: the former derives a closed-form solution for the merging coefficient, while the latter models the loss gap as the product of gradients and task vectors to quantify knowledge conflicts.

Recent approaches exploit SVD to capture the low-rank structure of task matrices, operating within subspaces to align, orthogonalize, or equalize components so as to preserve shared knowledge while suppressing interference. KnOTS (Stoica et al., 2025), designed for LoRA fine-tuning, merges adapters by aligning and averaging their right singular vectors. TSV (Gargiulo et al., 2025) applies SVD to each task matrix and whitens the resulting subspaces to reduce inter-task interference. Iso-C (Marczak et al., 2025) replaces the singular values of the summed task matrix with their mean, flattening the singular spectrum to expand the effective subspace and significantly improve the subspace alignment ratio. Iso-CTS (Marczak et al., 2025) extends this by injecting task-specific singular directions, balancing shared and task-specific knowledge. DOGE TA (Wei et al., 2025) formulates merging as a constrained optimization problem, constructs a shared subspace, and optimizes a correction vector via projected gradient descent with adaptive task-aware merging coefficients, further narrowing the performance gap to individual task models.

However, all these model merging methods rest on an implicit assumption: that the task-specific and -critical knowledge resides solely within the corresponding task vector and is independent of the pre-trained base model. Consequently, they overlook the LBW parameters within the pre-trained parameters that are critical for each specific task. Our study is pioneering to examine the role of pre-trained parameters for individual tasks and to protect the pre-trained core subspace by filtering LBW dimensions from task vectors.

Connection and Distinction with Counterparts. Concurrent work CSP (Sun et al., 2026) also recognizes the importance of pre-trained weights in model merging, but differs fundamentally in motivation. CSP builds on their near-optimality and preserves a global, task-agnostic core subspace. In contrast, `PACT` identifies task-specific knowledge anchored in pre-trained parameters despite negligible fine-tuning updates, termed task-specific LBW dimensions. Accordingly, CSP applies a static projection based on a globally shared subspace, whereas `PACT` constructs dynamic, mutually orthogonal shields to protect heterogeneous LBW subspaces from cross-task overwriting. Integrated with Iso-C, `PACT` consistently outperforms CSP, with gains reaching +3.0 points in the 20-task ViT-B/16 setting. Moreover, static core filtering (S-`PACT`) yields almost no gain, confirming the necessity of task-conditioned shielding (Appendix Sec. D.1 and Sec. E.1).

3 Motivation and Observation

3.1 Preliminary

Most model merging methods operate on task vectors $\Delta_t = \theta_t - \theta_0$, implicitly treating them as complete representations of task-specific knowledge. We challenge this assumption by considering parameter dimensions that receive negligible fine-tuning updates but remain critical to task performance. We call such dimensions Load-Bearing Wall (LBW) dimensions: for task t, dimension i is an LBW dimension if $|\Delta_t^{(i)}| \approx 0$, while perturbing the corresponding pre-trained parameter $\theta_0^{(i)}$

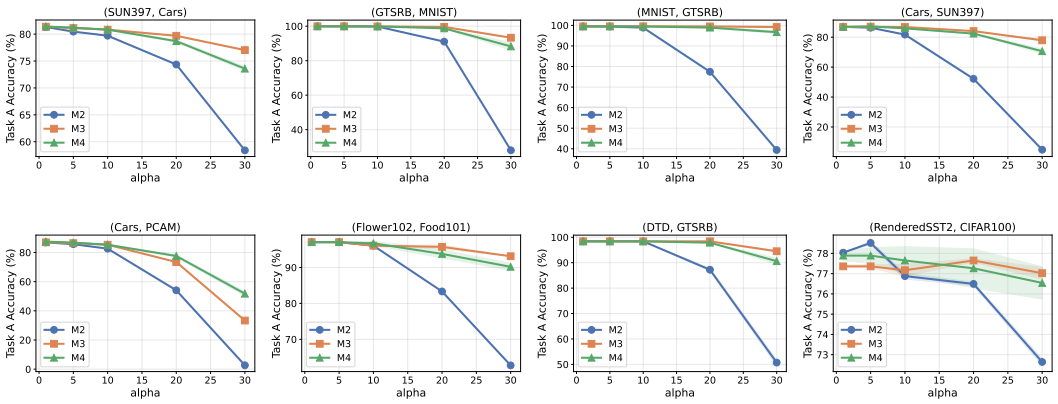


Figure 2: Accuracy (%) under varying attack strengths with respective to different masks. Generally, $M2$ could be easily attacked while $M3$ and $M4$ are rather stable, indicating the importance of parameters within crucial mask. For more results, please refer to Appendix Sec. C.1.

substantially degrades task performance. Because these dependencies are weakly represented in Δ_t, updates from other tasks may overwrite them without exhibiting conventional task-vector conflicts.

3.2 Identifying and Validating LBW Dimensions

3.2.1 LBW Dimensions in Scalar Weight

To validate our hypothesis, we first introduce two metrics for identifying potential LBW dimensions, which are used throughout the subsequent experiments:

$$F_\theta = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial \mathcal{L}i}{\partial \theta}\right)^2, \qquad E_B = \frac{d}{n}\frac{\sum_{\tau_B \in [\widehat{\Delta_A}]} \tau_B^2}{\sum_{\tau_B \in [\Delta_B]} \tau_B^2}. \tag{1}$$

Here, $N = 1000$ is the number of samples used to estimate the Fisher information (Kirkpatrick et al., 2017). A large value of F_θ indicates that the loss is sensitive to the corresponding parameter, suggesting that the parameter is important for the task. In the definition of E_B (Subramanian et al., 2005), $\widehat{\Delta_A}$ denotes a selected subset of the task-A vector Δ_A, while n and d denote the number of parameters in this subset and the total number of parameters in the corresponding layer, respectively. The score compares the average squared update magnitude within the selected subset with that of the entire layer. Therefore, $E_B > 1$ indicates that task B updates the dimensions selected from task A more strongly than it updates a typical dimension in the task-B layer.

Filter Process. For each layer, we select the 30% smallest-magnitude parameters in Δ_A and retain the 30% with the largest Fisher values, yielding approximately 9% of the layer parameters. We then retain dimensions with statistically significant update enrichment from task B, forming the *crucial mask*. This mask contains fewer than 2% of the model parameters and targets dimensions that are weakly updated by task A, important to its performance, and likely to be modified by task B. We construct equally sized *safe* and *random* masks using low-Fisher parameters and uniform sampling, respectively, followed by the same enrichment filtering.

Do LBW Parameters Exist? We investigate LBW parameters using 16 task pairs (A, B) from standard benchmarks (Wang et al., 2024). For each pair, we construct task A's crucial mask and perturb its dimensions using task B's task vector. Let $M0$ denote the performance of task A's fine-tuned model. Restoring the crucial-mask parameters to their pre-trained values yields $M1$, testing whether their functionality originates from the pre-trained model. We further apply $\theta_0 + \alpha\tau_B$, with $\alpha \in \{1, 5, 10, 20, 30\}$, to the crucial, safe, and random masks, producing $M2$, $M3$, and $M4$, respectively. Here, τ_B denotes the corresponding parameters in Δ_B. All experiments use ViT-B/16.

(A,B)	$M0$	$M1$	$M2$	$M3$	$M4$
(Cars, SUN397)	86.49	86.61	5.04	77.93	70.64
(GTSRB, MNIST)	99.92	99.92	28.04	93.39	90.47
(Food101, Flowers102)	89.56	89.60	55.29	84.01	83.16
(DTD, GTSRB)	98.40	98.40	50.71	94.50	90.60
(OxfordIIITPet, SUN397)	94.84	94.84	51.54	93.03	91.94

Table 1: Accuracy (%) under different parameter modifications. The value of α is set to 30.0.

(A, B)	$\theta_0 + \Delta_A + \Delta_B$	θ_0 on crucial	Δ_{acc}
(SUN397, Cars)	70.97	71.78	+0.81
(Cars, SUN397)	78.62	79.36	+0.74
(GTSRB, MNIST)	91.55	93.47	+1.91
(Cars, PACM)	74.69	77.03	+2.33
(DTD, GTSRB)	92.55	94.68	+2.13

Table 2: Δ_{acc} represents the accuracy (%) change due to the crucial mask replacement.

As shown in Figure 2 and Table 1, $M0$ and $M1$ perform nearly identically, indicating that the selected dimensions derive their functionality primarily from the pre-trained model. In contrast, $M2$ suffers substantially greater degradation than $M3$ and $M4$ under equal-magnitude perturbations. Performance decreases monotonically with α for all masks, with the crucial mask consistently affected most severely, providing direct evidence for LBW parameters.

Are LBW Parameters Damaged During Merging? Having established LBW parameters, we examine whether model merging disrupts them. Starting from the merged model $\theta_0 + \Delta_A + \Delta_B$, we evaluate task A before and after restoring only the crucial-mask parameters to their pre-trained values, while keeping all others unchanged. A performance recovery indicates that these parameters were overwritten during merging and contributed to task-A degradation.

As shown in Table 2, this targeted restoration substantially improves task-A performance despite affecting around 2% of the parameters. These results demonstrate that task-vector-based merging can modify LBW dimensions and impair task-specific knowledge embedded in the pre-trained model. Additional results are provided in Appendix Sec. C.1.

3.2.2 LBW Dimensions in Subspace

We have established the existence of LBW dimensions and their vulnerability during model merging. However, whether a similar phenomenon exists in subspace-based model merging methods (Yang et al., 2026) remains unclear. While the Fisher information analysis reveals the vulnerability of specific parameters at a microscopic level, subspace-based approaches operate on low-rank geometric structures rather than individual weights. Therefore, to understand how task interference arises in these methods, we extend our analysis from parameter space to subspace geometry.

We perform SVD on both the pre-trained weight matrix $\theta_0 \in \mathbb{R}^{m\times n}$ and the task vectors $\Delta_1, \ldots, \Delta_T \in \mathbb{R}^{m\times n}$:

$$\theta_0 = U_0 \Sigma_0 V_0^\top, \quad \text{and} \quad \Delta_t = U_t \Sigma_t V_t^\top, \quad \forall t = 1, \ldots, T, \tag{2}$$

where Σ_* denotes the diagonal matrix of singular values, and U and V denote the left and right singular vectors, respectively. Physically, for any input feature $x \in \mathbb{R}^n$, the forward pass transformation is governed by $\theta_0 x = U_0 \Sigma_0 V_0^\top x$. In this formulation, the right singular vectors V_0 act as an orthonormal basis for the input space, where the projection coordinates $V_0^\top x$ are scaled by the singular values Σ_0. Consequently, the top-K right singular vectors $V_{0,K}$ represent the principal directions that preserve the maximum variance of the representations, thereby capturing the dominant structural foundation of the pre-trained model.

To identify LBW dimensions in the spectral domain, we extract the top-K singular vectors of the pre-trained weights to form the base matrix $V_{0,K} \in \mathbb{R}^{n\times K}$. Similarly, we extract the top-k singular vectors of Δ_t to form the active task matrix $V_{t,k} \in \mathbb{R}^{n\times k}$. Unless otherwise specified, we set $(K, k) = (15, 8)$ throughout all experiments.

We define the *LBW subspace* $V_{t,LBW}$ of task t as the component of the pre-trained core subspace $V_{0,K}$ that lies outside the active update subspace of task t. Formally, it is obtained by projecting $V_{0,K}$ onto the orthogonal complement of $V_{t,k}$ followed by orthonormalization:

$$V_{t,LBW} = \text{orth}\left((I - V_{t,k} V_{t,k}^\top) V_{0,K}\right) \in \mathbb{R}^{n\times d_{LBW}}. \tag{3}$$

Specifically, the projection matrix $I - V_{t,k} V_{t,k}^\top$ filters out components of the pre-trained core $V_{0,K}$ that lie within the active task updates $V_{t,k}$. This projection successfully isolates the pre-trained

directions that are critical for task performance yet receive negligible updates during fine-tuning. The orth operator then orthonormalizes these remaining components to construct a stable, non-redundant basis $V_{t,LBW}$ with dimension d_{LBW}, which is at most K. Conceptually, $V_{t,LBW}$ serves as the spectral analogue of the LBW dimensions, capturing pre-trained feature directions that remain untouched during task adaptation.

Causal Validation via Subspace Ablation. To investigate whether model performance depends on $V_{t,LBW}$, we conduct a global subspace ablation study. Given a task-specific model $\theta_t = \theta_0 + \Delta_t$, we remove the LBW subspace by constructing $\theta_{abla} = \theta_t(I - V_{t,LBW}V_{t,LBW}^\top)$. We reasonably assume Δ_t is approximately orthogonal to $V_{t,LBW}$ by construction (i.e., $\Delta_t V_{t,LBW} \approx 0$), this operation primarily removes the pre-trained components contained in $V_{t,LBW}$ while preserving the task-specific updates:

Task t	θ_t	θ_{abla} ($\Delta_{\text{acc}}^{LBW}$)	θ_{rand} ($\Delta_{\text{acc}}^{rand}$)	$\Delta_{\text{acc}}^{LBW}/\Delta_{\text{acc}}^{rand}$
EMNIST	99.78	17.00 (-82.78)	99.75 (-0.03)	3104.25×
FashionMNIST	95.28	9.78 (-85.50)	95.14 (-0.14)	610.71×
GTSRB	99.92	1.91 (-98.01)	99.86 (-0.06)	1566.60×
KMNIST	99.94	11.92 (-88.02)	99.88 (-0.06)	1467.00×
STL10	99.60	11.00 (-88.60)	99.33 (-0.27)	332.25×
SVHN	96.76	7.40 (-89.36)	96.51 (-0.25)	352.74×

Table 3: Δ_{acc}^* represents the accuracy (%) change due to the subspace filtering. The numbers in parentheses indicate the decrease in accuracy compared to the θ_t.

$$\theta_{abla} \approx \theta_0(I - V_{t,LBW}V_{t,LBW}^\top) + \Delta_t \tag{4}$$

We provide a formal derivation in Appendix Sec. A.3. As a control group, we ablate a random orthogonal subspace V_{rand} of identical dimensionality strictly within the non-core pre-trained space ($V_{0,K}^\perp$): $\theta_{rand} = \theta_t(I - V_{rand}V_{rand}^\top)$. To ensure a fair comparison, the control subspace V_{rand} is constructed under two constraints: it must share the identical dimension d_{pact} and lie strictly within the non-core pre-trained space orthogonal to $V_{0,K}$. Formally, we project a random Gaussian matrix X_{rand} onto the orthogonal complement of the pre-trained core to obtain $(I - V_{0,K}V_{0,K}^\top)X_{rand}$, and then apply QR decomposition to extract the orthonormal basis. From this basis, we randomly select and re-orthonormalize d_{pact} columns to yield V_{rand}. Unlike $V_{t,LBW}$ which preserves the structured, task-essential components derived from the pre-trained core $V_{0,K}$, V_{rand} represents purely unstructured directions within the non-core space $V_{0,K}^\perp$.

Table 3 reports results on 19 tasks using ViT-B/16. Although $V_{t,LBW}$ accounts for only **1.17%** of the total dimensions (780 out of 66,816), its removal leads to severe performance degradation across tasks. For example, EMNIST accuracy drops from **99.78%** to **17.00%** ($\downarrow 82.78\%$), while GTSRB accuracy decreases from **99.92%** to **1.91%** ($\downarrow 98.01\%$). In contrast, ablating a random control subspace has a negligible effect, with EMNIST and GTSRB retaining **99.75%** ($\downarrow 0.03\%$) and **99.86%** ($\downarrow 0.06\%$) accuracy, respectively. Additional results are provided in Appendix Sec. C.2.

On average, the performance degradation caused by removing $V_{t,LBW}$ is **458.05× larger** than that observed for the random control subspace. These results provide strong empirical evidence that the pre-trained components contained in $V_{t,LBW}$ play a disproportionately important role in downstream task performance, supporting our interpretation of $V_{t,LBW}$ as the subspace-level manifestation of LBW dimensions.

Interference of LBW Subspaces During Model Merging. Having established $V_{t,LBW}$ as the LBW subspace, we analyze how standard model merging disrupts these structures by introducing two geometric metrics for any task pair (A, B) to quantify this interference. First, we define the symmetric *Sacred Space Similarity* (Sim) to quantify the geometric overlap between the LBW subspaces of two distinct tasks, and the asymmetric *Hidden Interference Ratio* (Inf) to quantify the direct intrusion of one task's updates into another task's parameters:

$$\text{Sim}(A, B) = \frac{\|(V_{A,LBW})^\top V_{B,LBW}\|_F^2}{\min(r_A, r_B)}, \qquad \text{Inf}(B \to A) = \frac{\|(V_{B,k})^\top V_{A,LBW}\|_F^2}{k} \tag{5}$$

where r_A and r_B denote the rank of their respective LBW subspace. Mathematically, $\text{Sim}(A, B)$ computes the sum of squared cosines of principal angles between two subspaces, a classical metric heavily utilized in Grassmanian manifold alignment (Fernando et al., 2013) and task singular vector synchronization (Marczak et al., 2025). Inf computes the exact proportion of

Task B's explicit active task vector updates ($V_{B,k}$) that invasively project onto the LBW subspace of Task A. Unlike parameter-wise ratios, the Inf metric is purely geometric and scale-invariant, strictly evaluating the directional collisions between Task B's core task vector updates and Task A's implicit dependency on its LBW subspace. This projection-based interference measurement is closely related to the gradient projection metrics developed in continual learning for preventing catastrophic forgetting (Farajtabar et al., 2020; Saha et al., 2021). The empirical severity of these subspace-level collisions is fully exposed in the heatmaps for the foundational Block 0 (Figure 3). On one hand, the Sim heatmap reveals that **the similarity between the LBW subspaces of different tasks is relatively low to some extent**, with values dropping as low as 0.59. This low similarity confirms that different tasks depend on highly heterogeneous subsets of pre-trained parameters, proving that a universal, task-agnostic "safe zone" does not exist; instead, each task carves out a unique, indispensable dependency on its own LBW subspace. On the other hand, the Inf heatmap uncovers a non-negligible geometric collision, demonstrating that **the active updates of task vectors blindly invade the LBW subspaces of other tasks**. While the diagonal remains strictly 0.00—confirming self-orthogonality—the off-diagonal entries skyrocket, showing that the explicit task vector updates of one task can obliterate up to **48%** of the LBW subspace implicitly relied upon by another (for instance, SUN397 severely intrudes upon the LBW subspace of EuroSAT). This profound geometric collision highlights that naive model merging severely corrupts the foundational LBW subspaces of downstream tasks (refer to Appendix Sec. C.3 for more results). This directly mandates a layer-wise mechanism to explicitly shield these LBW subspaces during merging, motivating our design.

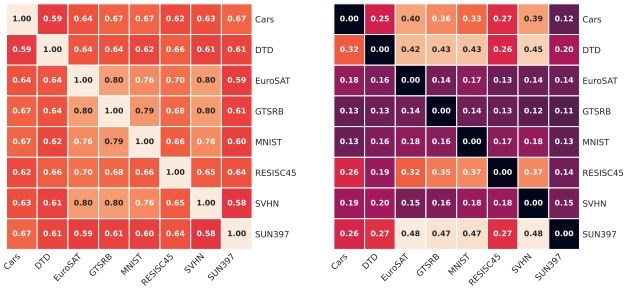


Figure 3: Sim and Inf of the `mlp.c_fc` layer in Block 0.

4 Preserving Anchored Cores in Task-Vectors

While Fisher-based methods can theoretically locate these vital parameters, their heavy computational cost and requirement for training data preclude the possibility of explicitly protecting them in practical, data-free model merging scenarios. We therefore need a merging framework that is task-agnostic, data-free, and capable of geometrically shielding these implicit dependencies layer by layer. To this end, we propose PACT. Instead of directly summing task matrices, PACT operationalizes the isolated "Sacred Spaces" ($V_{t,LBW}$ derived in Eqn. 3) into a universal geometric shield, enforcing strict orthogonal filtering to neutralize hidden interference before integration.

As defined in Section 3, $V_{t,LBW}$ precisely identifies the orthogonal anchor directions that a single task t implicitly relies upon. However, merging T distinct tasks requires a holistic defense mechanism: we must ensure that no task's explicit update interferes with the load-bearing walls of any other task. Simply filtering against each $V_{t,LBW}$ sequentially is mathematically flawed and computationally inefficient, as the sacred spaces across different tasks often share significant overlaps (reflected by the high similarities in Figure 3). To construct a unified, non-redundant protection boundary for a given task t, we aggregate the sacred spaces of all other tasks. We then extract a unified orthonormal column basis using QR decomposition, denoted by the operator $\mathrm{orth}(\cdot)$:

$$V^t_{\text{protect}} = \mathrm{orth}\Big(\big[V_{1,LBW} \;\big|\; \dots \;\big|\; V_{t-1,LBW} \;\big|\; V_{t+1,LBW} \;\big|\; \dots \;\big|\; V_{T,LBW}\big]\Big) \tag{6}$$

This resulting matrix V^t_{protect} serves as the *Global Orthogonal Shield* for task t. It elegantly captures the union of all critical pre-trained dependencies required by the rest of the tasks, while eliminating overlapping basis vectors to prevent over-constraining the update space. Equipped with the global shield V^t_{protect}, we can now surgically sanitize the update matrix of task t. We enforce strict non-interference by subtracting the projection of Δ_t onto the protected subspace:

$$\tilde{\Delta}_t = \Delta_t - \Delta_t V^t_{\text{protect}} (V^t_{\text{protect}})^\top \tag{7}$$

where $V^t_{\text{protect}}(V^t_{\text{protect}})^\top$ is the orthogonal projection operator onto the protection space. Applying it to Δ_t extracts the components of Δ_t that would overwrite the anchor directions of other tasks. Subtracting these components removes such detrimental parts, retaining only the portions of Δ_t that are harmless to other tasks, thereby completing the filtering. In this process, the filtering of each task is jointly determined by the anchor directions of all other tasks, embodying a mutual protection mechanism across tasks.

The complete computational procedure is presented in Algorithm 1 in the Appendix Sec. G. Considering that our algorithm requires performing $T+1$ SVD decompositions for each layer of the model, its time complexity reaches $O(mn \cdot \min(m,n))$. Moreover, it requires storing the full $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$, which imposes substantial memory pressure for large-scale matrices. To address this, we propose an efficient alternative of PACT. Specifically, we can replace the full SVD decomposition in Eqn. 2 with randomized SVD (Halko et al., 2011). Randomized SVD leverages random projections for low-rank approximation, and its ability to efficiently and approximately obtain the top-k singular values and singular vectors is suited to the top-k extraction process of PACT. Meanwhile, the steep singular value spectrum ensures the efficiency of randomized SVD. Through the adaptation of randomized SVD, we can reduce the time complexity to $O(mnk)$, while only needing to store the narrow matrix $Q \in \mathbb{R}^{m\times(k+p)}$ and a few small matrices, thereby substantially alleviating memory pressure as well. The performance of this alternative and its computational complexity analysis are provided in the Appendix Sec. B and Sec. E.5. We also provide the theoretical analysis of PACT in the Appendix Sec. A.

5 Experiments

5.1 Fully Fine-tuned Vision Models

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Zero-shot	48.3	57.2	56.1	55.3	61.3	59.7	64.7	68.2	65.2
Fine-tuned	92.8	90.9	91.3	94.6	92.8	93.2	95.8	94.3	94.7
Weight Averaging	66.3(72.1)	64.3(71.1)	61.0(67.5)	72.2(76.6)	69.5(74.8)	65.3(70.4)	79.6(83.2)	76.7(81.1)	71.6(75.6)
TIES	75.1(81.0)	68.0(74.8)	63.4(69.9)	79.7(84.3)	73.2(78.7)	68.2(73.3)	86.9(90.7)	79.5(84.1)	75.7(79.8)
Consensus TA	75.0(80.8)	70.4(77.4)	65.4(72.0)	79.4(83.9)	74.4(79.9)	69.8(74.9)	86.3(90.1)	82.2(86.9)	79.0(83.2)
Iso-CTS	86.2(92.8)	81.7(89.7)	78.1(85.5)	91.1(96.1)	**86.4(92.8)**	82.4(88.4)	94.7(98.8)	**91.0(96.3)**	90.1(94.9)
Task Arithmetic	70.8(76.5)	65.3(72.1)	60.5(66.8)	75.4(79.6)	70.5(75.9)	65.8(70.8)	84.9(88.7)	79.4(84.0)	74.0(78.1)
PACT-TA	75.2(81.0)	66.4(73.3)	59.9(66.4)	85.5(90.3)	76.8(82.7)	68.7(74.0)	92.8(96.9)	87.0(92.0)	83.0(87.7)
TSV-M	85.9(92.3)	80.1(87.9)	77.1(84.3)	89.0(93.9)	84.6(91.0)	80.6(86.5)	93.0(97.0)	89.2(94.4)	87.7(92.5)
PACT-TSV-M	**88.1(94.8)**	**82.0(89.6)**	**80.3(87.6)**	91.2(96.3)	85.4(91.5)	**83.8(89.6)**	94.4(98.5)	89.5(94.4)	89.7(94.3)
Iso-C	86.3(92.9)	80.3(88.1)	75.5(82.5)	90.6(95.6)	84.8(91.1)	79.6(85.4)	94.2(98.3)	89.3(94.5)	87.6(92.2)
PACT-Iso-C	87.3(94.1)	81.3(89.0)	78.7(86.0)	**92.0(97.2)**	86.1(92.3)	83.4(89.3)	**95.1(98.3)**	90.5(95.5)	**90.3(95.0)**

Table 4: PACT-Iso-C achieves SOTA performance for all backbones on all evaluated scenarios. We present average absolute accuracy and average normalized accuracy (in bracket) in %. The best method in **bold** and the second-best underlined.

We evaluate our approaches over sets of 8, 14, and 20 datasets, following Marczak et al. (2025). We provide the details of the datasets in Appendix Sec. F. We consider three variants of CLIP (Radford et al., 2021) with ViT-B/32,ViT-B/16 and ViT-L/14 as visual encoders (Dosovitskiy et al., 2020). We use the checkpoints fine-tuned on the tasks above, provided in Wang et al. (2024). The hyperparameters (K, k) are set to be (15,8).

We compare our approaches with the following model merging methods: weight averaging (Wortsman et al., 2022a), Task Arithmetic (Ilharco et al., 2022a), TIES-Merging (Yadav et al., 2023), Consensus TA (Wang et al., 2024), TSV-M (Gargiulo et al., 2025), Iso-C and Iso-CTS (Marczak et al., 2025). We include the results of the zero-shot model and fine-tuned models serving as lower- and upper-bound, respectively. We compare the results based on absolute and normalized accuracy following standard practice (Wang et al., 2024; Gargiulo et al., 2025; Marczak et al., 2025).

Table 4 presents the performance of multi-task model merging. PACT consistently improves across all settings over all three base methods: TA, TSV-M, and Iso-C. Notably, PACT-IsoC achieves SOTA results in all 9 settings, and its gain over Iso-C becomes even more pronounced as the

Method	CoLA	SST-2	MRPC	STSB	QQP	MNLI	QNLI	RTE	Avg.	Norm Avg.
Individual	0.5511	0.9472	0.8799	0.8770	0.8946	0.8745	0.9244	0.6643	0.8266	-
Weight Averaging	0.0	0.5528	0.7549	0.4201	0.3993	0.4173	0.5982	0.4729	0.4519	52.54
TA w/ DARE	0.0	0.8073	0.7811	0.3059	0.7124	0.6401	0.7772	0.5560	0.5725	66.19
Ties-Merging	0.1139	0.9117	0.7134	-0.0476	0.7245	**0.7931**	0.8278	0.4152	0.5565	64.54
TIES w/ DARE	0.0656	0.8601	0.7803	0.1464	0.7263	0.666	0.79	0.4946	0.5662	65.67
Task Arithmetic	0.0	0.8314	0.7841	0.2958	0.7025	0.6523	0.7785	0.5343	0.5724	66.05
PACT-TA	0.1763	0.9060	0.7799	0.1255	0.7711	0.77	0.8155	0.5379	0.6103	71.75
TSV-M	0.2020	0.9300	**0.7895**	0.4977	0.8260	0.7456	0.8288	0.5126	0.6665	78.22
PACT-TSV-M	0.4453	**0.9255**	0.7702	**0.6489**	0.7971	0.7542	**0.8541**	0.5126	0.7135	85.62
Iso-C	0.3013	0.9094	0.7755	0.6206	**0.8470**	0.7873	0.8510	0.6390	**0.7164**	85.32
PACT-Iso-C	**0.4960**	0.8899	0.7728	0.6306	0.8110	0.5856	0.7954	**0.6931**	0.7093	**86.46**

Table 6: Results of merging RoBERTa models on eight GLUE tasks. We present average absolute accuracy and average normalized accuracy in %. Best performances are shown in **bold**.

number of tasks increases, surpassing the protection offered by Iso-CTS in the common space. PACT-TSV-M also attains SOTA performance on ViT-B/32. Moreover, PACT-TA delivers highly competitive results, with the three experiments on ViT-L/14 all approaching the SOTA level. These substantial improvements further corroborate the existence of the LBW effect within pre-trained parameters.

5.2 LoRA-adaptive Vision Models

To evaluate our approaches in low-rank adaptation scenario, we follow the evaluation protocol of KnOTS (Stoica et al., 2025), a recent SOTA method for merging LoRA fine-tuned models. We use codebase and checkpoints provided by KnOTS: ViT-B/32 and ViT-L/14 fine-tuned with rank 16 LoRA (Hu et al., 2022) on 8 vision tasks. To adapt our methodology to the low-rank regime, rather than operating on the reconstructed dense matrix $\Delta W_t = B_t A_t$, we simply perform SVD on the small matrix A_t. This is mathematically justified as they share the exact same row space, providing a computationally free trick to extract the task subspace. We compare our approaches with Iso-C, Iso-CTS, TIES and DARE-TIES (Yu et al., 2024) - combined with KnOTS or not – and TA. The hyperparameters (K, k) are set to be (7, 4).

Method	ViT-B/32	ViT-L/14
TIES	63.7	75.2
DARE-TIES	63.7	74.7
KnOTS-TIES	68.0	78.2
KnOTS-DARE-TIES	63.9	75.6
Iso-CTS	73.7	85.3
Task Arithmetic	63.7	74.4
PACT-TA	66.9	77.7
Iso-C	73.6	83.7
PACT-Iso-C	**75.1**	**86.3**

Table 5: Normalized per-task average accuracy. We merge 8 models fine-tuned with LoRA following (Stoica et al., 2025). PACT-Iso-C achieves SOTA performance on all evaluated scenarios.

As presented in Table 5, PACT-Iso-C surpasses all methods in both experiments, achieving SOTA performance, which demonstrates the versatility of our merging approach. Meanwhile, PACT-TA also achieves performance close to KnOTS, a method specifically designed for LoRA fine-tuning, indicating that the LBW effect also exists in LoRA fine-tuning, even though the pre-trained parameters remain untouched during the fine-tuning process.

5.3 Fully Fine-tuned Language Models

To evaluate our approaches on language models, we conduct experiments on GLUE tasks (Wang et al., 2018) with RoBERTa-base (Liu et al., 2019) and GPT-2 (Radford et al., 2019). Due to the space limit, we present the results of RoBERTa-base in the main text, while leaving the GPT-2 results in the Appendix Sec. D.2. Partially following DARE (Yu et al., 2024), we merge all eight RoBERTa-base (Liu et al., 2019) models fine-tuned on each task. Performance is evaluated on eight GLUE (Wang et al., 2018) tasks: CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan & Brockett, 2005), STS-B (Cer et al., 2017), QQP (Iyer et al., 2017), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Giampiccolo et al., 2007). Among these, CoLA uses the Matthews correlation coefficient, STS-B uses the average of Pearson and Spearman correlation coefficients, and the remaining tasks are evaluated by accuracy. As shown in Table 6, PACT outperforms all other merging methods, further validating the LBW effect in model merging for NLP.

AI USE STATEMENT

In this work, we used generative AI tools to assist with searching for and reviewing relevant literature, polishing and rewriting author-written text, and checking the correctness of author-developed theoretical proofs. All identified literature was verified against the original sources, and all AI-assisted edits and proof checks were independently reviewed and validated by the authors. Generative AI tools were not used for any other purpose. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We have provided sufficient details for reproducibility in the Appendix Sec. F, and our implementation is also provided in the supplementary materials.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 1–14, 2017.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*, 2005.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pp. 2960–2967, 2013.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18695–18705, 2025.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9, 2007.

Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, pp. 117–124. Springer, 2013.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53 (2):217–288, 2011.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022a.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277, 2022b.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. First quora dataset release: Question pairs, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lu Li, Tianyu Zhang, Zhiqi Bu, Suyuchen Wang, Huan He, Jie Fu, Yonghui Wu, Jiang Bian, Yong Chen, and Yoshua Bengio. Map: Low-compute model merging with amortized pareto fronts via quadratic approximation. In *International Conference on Learning Representations*, volume 2025, pp. 65032–65064, 2025.

Weishi Li, Yong Peng, Miao Zhang, Liang Ding, Han Hu, and Li Shen. Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost Van De Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *arXiv preprint arXiv:2502.04959*, 2025.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

The-Hai Nguyen, Dang Huu-Tien, Takeshi Suzuki, and Le-Minh Nguyen. Regmean++: Enhancing effectiveness and generalization of regression mean for model merging. *arXiv preprint arXiv:2508.03121*, 2025.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

OM Parkhi. A. vedaldi, a. In *Zisserman, c. Jawahar,"cats and dogs," in 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Wei Ruan, Tianze Yang, Yifan Zhou, Tianming Liu, and Jin Lu. From task-specific models to unified systems: A review of model merging approaches. *arXiv preprint arXiv:2503.08998*, 2025.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pp. 1453–1460. IEEE, 2011.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, volume 2025, pp. 4501–4519, 2025.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the national academy of sciences*, 102(43):15545–15550, 2005.

Wenju Sun, Qingyong Li, Wen Wang, Yangliao Geng, and Boyang Li. Task arithmetic in trust region: A training-free model merging approach to navigate knowledge conflicts. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 5178–5187, 2025.

Wenju Sun, Qingyong Li, Tiancheng Li, Yangliao Geng, and Boyang Li. Revisiting the role of pretrained weights in model merging: On near-optimality within the core subspace. In *Forty-third International Conference on Machine Learning*, 2026.

Anke Tang, Li Shen, Yong Luo, Liang Ding, Han Hu, Bo Du, and Dacheng Tao. Concrete subspace learning based interference elimination for multi-task model fusion. *arXiv preprint arXiv:2312.06173*, 2023.

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Fusionbench: A comprehensive benchmark of deep model fusion. 2024.

Vinita Vasudevan and M Ramakrishna. A hierarchical singular value decomposition algorithm for low rank matrices. *arXiv preprint arXiv:1710.02812*, 2017.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 210–218. Springer, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 2024.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Yongxian Wei, Zixuan Hu, Li Shen, Zhenyi Wang, Yu Li, Chun Yuan, and Dacheng Tao. Task groupings regularization: Data-free meta-learning with heterogeneous pre-trained models. *arXiv preprint arXiv:2405.16560*, 2024.

Yongxian Wei, Anke Tang, Li Shen, Zixuan Hu, Chun Yuan, and Xiaochun Cao. Modeling multi-task model merging as adaptive projective gradient descent. *arXiv preprint arXiv:2501.01230*, 2025.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pp. 1112–1122, 2018.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119 (1):3–22, 2016.

Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. Multi-task model merging via adaptive weight disentanglement. *arXiv preprint arXiv:2411.18729*, 2024.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. Metagpt: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1711–1724, 2024.

Appendix for "Preserving Anchored Cores in Task-vectors for Model Merging"

A Theoretical Properties for PACT **16**
A.1 Proof of Theorem 1 16
A.2 Proof of Corollary 1 17
A.3 Derivation and Proof of the Ablation Approximation 17

B Computational complexity analysis **18**

C Additional Motivation Experiments **19**
C.1 Extended Analysis on Fisher-level Parameter Collisions 19
C.2 Extended Geometric Subspace Analysis and Layer-wise Heterogeneity 21
C.3 Layer-wise and Component-wise Analysis of Pre-trained Subspace Intrusion 23

D Additional Evaluation **27**
D.1 Comparison with Counterparts 27
D.2 Evaluation on Additional Language Benchmarks 28

E Detailed Ablation Studies **28**
E.1 Algorithmic Component Ablation: Dynamic Task Shields vs. Static PACT (S-PACT) . 29
E.2 Layer-wise Depth Ablation: Localized Filtering for Efficiency 29
E.3 Ablation on Component-Wise Projection Filtering 30
E.4 Hyperparameter Sensitivity Analysis 31
E.5 Efficient Alternative 32

F Experimental Details **32**
F.1 Datasets 32
F.2 Additional Notes 34
F.3 Selection of the scaling factor α 34

G Algorithm **36**

H Conclusion & Limitation **36**

A THEORETICAL PROPERTIES FOR PACT

To rigorously justify the empirical success of PACT, we provide a theoretical analysis demonstrating its preservation of individual task performance.

Theorem 1 (Projected First-Order Performance Preservation). *Let $\theta_j = \theta_0 + \Delta_j$ denote the task-specific fine-tuned parameters for task j, and let $P_j = V^j_{\text{protect}}$ be the orthonormal basis of the protection space used by PACT. The filtered update is defined as $\tilde{\Delta}_j = \Delta_j \left(I - P_j P_j^\top\right)$. Assume that $\mathcal{L}_j$ is locally β_j-smooth around θ_j. Then the degradation caused by the projection filtering satisfies*

$$\mathcal{L}_j(\theta_0 + \tilde{\Delta}_j) - \mathcal{L}_j(\theta_0 + \Delta_j) \le \|\nabla \mathcal{L}_j(\theta_j) P_j\|_F \, \|\Delta_j P_j\|_F + \frac{\beta_j}{2} \|\Delta_j P_j\|_F^2 .$$

In particular, the first-order degradation is bounded by

$$\Delta \mathcal{L}_j^{(1)} \le \|\nabla \mathcal{L}_j(\theta_j) P_j\|_F \, \|\Delta_j P_j\|_F . \tag{8}$$

Theorem 1 shows that the self-performance degradation is not determined by the overall norm of the update, but by the discarded update energy and the task sensitivity inside the protection space. Equivalently, defining

$$\rho_{g,j} = \frac{\|\nabla \mathcal{L}_j(\theta_j) P_j\|_F}{\|\nabla \mathcal{L}_j(\theta_j)\|_F}, \qquad \rho_{\Delta,j} = \frac{\|\Delta_j P_j\|_F}{\|\Delta_j\|_F},$$

the first-order term is controlled by

$$\Delta \mathcal{L}_j^{(1)} \le \|\nabla \mathcal{L}_j(\theta_j)\|_F \, \|\Delta_j\|_F \, \rho_{g,j} \rho_{\Delta,j}. \tag{9}$$

Corollary 1 (Active-Subspace Relaxation). *Let $V_{j,k}$ be the orthonormal basis of the active subspace of task j, and define $\cos(\phi_j) = \left\| V_{j,k}^\top P_j \right\|_2$. Suppose that both the gradient and the task update are approximately concentrated in $V_{j,k}$, namely*

$$\left\|\nabla \mathcal{L}_j(\theta_j) \left(I - V_{j,k} V_{j,k}^\top\right)\right\|_F \le \epsilon_g \|\nabla \mathcal{L}_j(\theta_j)\|_F ,$$

and

$$\left\|\Delta_j \left(I - V_{j,k} V_{j,k}^\top\right)\right\|_F \le \epsilon_\Delta \|\Delta_j\|_F .$$

Then the degradation caused by PACT's filtering satisfies

$$\begin{aligned}\mathcal{L}_j(\theta_0 + \tilde{\Delta}_j) - \mathcal{L}_j(\theta_0 + \Delta_j) \le{}& \|\nabla \mathcal{L}_j(\theta_j)\|_F \, \|\Delta_j\|_F \, (\cos(\phi_j) + \epsilon_g)\,(\cos(\phi_j) + \epsilon_\Delta) \\ &+ \frac{\beta_j}{2} \|\Delta_j\|_F^2 \, (\cos(\phi_j) + \epsilon_\Delta)^2 .\end{aligned}$$

In the exact active-subspace case where $\epsilon_g = \epsilon_\Delta = 0$, the first-order degradation reduces to

$$\Delta \mathcal{L}_j^{(1)} \le \|\nabla \mathcal{L}_j(\theta_j)\|_F \, \|\Delta_j\|_F \cos^2(\phi_j). \tag{10}$$

A.1 PROOF OF THEOREM 1

Proof. Let

$$D_j = \Delta_j - \tilde{\Delta}_j = \Delta_j P_j P_j^\top \tag{11}$$

be the component discarded by PACT. Then

$$\theta_0 + \tilde{\Delta}_j = \theta_j - D_j. \tag{12}$$

Since $\mathcal{L}_j$ is locally β_j-smooth around θ_j, we have

$$\mathcal{L}_j(\theta_j - D_j) \le \mathcal{L}_j(\theta_j) - \langle \nabla \mathcal{L}_j(\theta_j), D_j \rangle + \frac{\beta_j}{2} \|D_j\|_F^2. \tag{13}$$

Substituting $D_j = \Delta_j P_j P_j^\top$ gives

$$\begin{aligned} - \langle \nabla \mathcal{L}_j(\theta_j), D_j \rangle &\le \left| \left\langle \nabla \mathcal{L}_j(\theta_j), \Delta_j P_j P_j^\top \right\rangle \right| \\ &= \left| \langle \nabla \mathcal{L}_j(\theta_j) P_j, \Delta_j P_j \rangle \right| \\ &\le \|\nabla \mathcal{L}_j(\theta_j) P_j\|_F \, \|\Delta_j P_j\|_F , \end{aligned} \tag{14}$$

where the last step follows from Cauchy–Schwarz. Moreover, since P_j has orthonormal columns,

$$\|D_j\|_F = \|\Delta_j P_j P_j^\top\|_F = \|\Delta_j P_j\|_F. \tag{15}$$

Combining the above inequalities yields

$$\begin{aligned}\mathcal{L}_j(\theta_0 + \tilde{\Delta}_j) - \mathcal{L}_j(\theta_0 + \Delta_j) &= \mathcal{L}_j(\theta_j - D_j) - \mathcal{L}_j(\theta_j) \\ &\le \|\nabla\mathcal{L}_j(\theta_j)P_j\|_F \|\Delta_j P_j\|_F + \frac{\beta_j}{2}\|\Delta_j P_j\|_F^2 .\end{aligned} \tag{16}$$

□

A.2 Proof of Corollary 1

Proof. Let $P_j = V_{j,k}V_{j,k}^\top$. By the triangle inequality,

$$\begin{aligned}\|\nabla\mathcal{L}_j(\theta_j)P_j\|_F &\le \|\nabla\mathcal{L}_j(\theta_j)P_jP_j\|_F + \|\nabla\mathcal{L}_j(\theta_j)(I-P_j)P_j\|_F \\ &\le \|\nabla\mathcal{L}_j(\theta_j)\|_F \left\|V_{j,k}^\top P_j\right\|_2 + \epsilon_g \|\nabla\mathcal{L}_j(\theta_j)\|_F \\ &= \|\nabla\mathcal{L}_j(\theta_j)\|_F (\cos(\phi_j) + \epsilon_g) .\end{aligned} \tag{17}$$

Similarly,

$$\|\Delta_j P_j\|_F \le \|\Delta_j\|_F (\cos(\phi_j) + \epsilon_\Delta) . \tag{18}$$

Substituting these two inequalities into Theorem 1 gives Eqn. 10. When $\epsilon_g = \epsilon_\Delta = 0$, the first-order term becomes

$$\Delta\mathcal{L}_j^{(1)} \le \|\nabla\mathcal{L}_j(\theta_j)\|_F \|\Delta_j\|_F \cos^2(\phi_j). \tag{19}$$

□

A.3 Derivation and Proof of the Ablation Approximation

To justify the ablation approximation presented in Eqn. 4, we present a unified mathematical derivation based on the construction of the LBW subspace. Recall that the task-specific model parameter is decomposed as $\theta_t = \theta_0 + \Delta_t$, and the ablation operation is defined as:

$$\theta_{abla} = \theta_t(I - V_{t,LBW}V_{t,LBW}^\top) \tag{20}$$

where the LBW subspace basis $V_{t,LBW} \in \mathbb{R}^{n\times d_{LBW}}$ is constructed via Eqn.3.

By this construction, $V_{t,LBW}$ lies within the column space of $(I - V_{t,k}V_{t,k}^\top)V_{0,K}$. Consequently, we can express $V_{t,LBW} = (I - V_{t,k}V_{t,k}^\top)V_{0,K}R^{-1}$ for some invertible matrix R. Left-multiplying this relation by $V_{t,k}^\top$ yields:

$$\begin{aligned}V_{t,k}^\top V_{t,LBW} &= V_{t,k}^\top(I - V_{t,k}V_{t,k}^\top)V_{0,K}R^{-1} \\ &= (V_{t,k}^\top - V_{t,k}^\top V_{t,k}V_{t,k}^\top)V_{0,K}R^{-1}\end{aligned} \tag{21}$$

Substituting the orthonormality condition $V_{t,k}^\top V_{t,k} = I$, the expression simplifies to:

$$\begin{aligned}V_{t,k}^\top V_{t,LBW} &= (V_{t,k}^\top - I\cdot V_{t,k}^\top)V_{0,K}R^{-1} \\ &= 0\cdot V_{0,K}R^{-1} = 0\end{aligned} \tag{22}$$

This establishes that the constructed LBW subspace $V_{t,LBW}$ is orthogonal to the principal task-specific subspace $V_{t,k}$.

Next, we assume that the task-specific parameter update Δ_t primarily aligns with the subspace spanned by its principal components $V_{t,k}$. This relationship can be formally expressed as the approximation $\Delta_t \approx \Delta_t V_{t,k}V_{t,k}^\top$. Projecting the update Δ_t onto the LBW subspace then yields:

$$\begin{aligned}\Delta_t V_{t,LBW} &\approx \left(\Delta_t V_{t,k}V_{t,k}^\top\right) V_{t,LBW} \\ &= \Delta_t V_{t,k}\left(V_{t,k}^\top V_{t,LBW}\right)\end{aligned} \tag{23}$$

Applying the strict orthogonality $V_{t,k}^\top V_{t,LBW} = 0$ derived above, we immediately obtain:

$$\Delta_t V_{t,LBW} \approx 0 \quad \text{and thus} \quad \Delta_t V_{t,LBW} V_{t,LBW}^\top \approx 0 \tag{24}$$

Finally, by substituting the decomposition $\theta_t = \theta_0 + \Delta_t$ into the ablation definition in Eqn. 20 and expanding the terms, we have:

$$\begin{aligned} \theta_{abla} &= (\theta_0 + \Delta_t)(I - V_{t,LBW} V_{t,LBW}^\top) \\ &= \theta_0 (I - V_{t,LBW} V_{t,LBW}^\top) + \Delta_t - \Delta_t V_{t,LBW} V_{t,LBW}^\top \end{aligned} \tag{25}$$

Using the approximation $\Delta_t V_{t,LBW} V_{t,LBW}^\top \approx 0$ from Eqn. 24, the expression simplifies to:

$$\theta_{abla} \approx \theta_0 (I - V_{t,LBW} V_{t,LBW}^\top) + \Delta_t \tag{26}$$

This completes the derivation, demonstrating that the ablation operation selectively filters out pre-trained components within the LBW subspace while leaving the task-specific updates largely intact.

B Computational complexity analysis

In this Section, we analyze the computational complexity of our proposed `PACT-Iso-C` and `PACT-TSV-M` merging algorithms, alongside their highly efficient RSVD variants (Halko et al., 2011). We compare them with the baseline Iso-C and the SOTA subspace merging methods, TSV-M (Gargiulo et al., 2025) and Iso-CTS (Marczak et al., 2025).

Following the conventions established in previous works, let $\Delta_t \in \mathbb{R}^{n \times n}$, and let T and L be the number of tasks and network layers, respectively. For simplicity, assume that each layer consists of a single squared $n \times n$ matrix.

In our analysis, we focus on the number of Singular Value Decompositions (SVDs) performed by each algorithm, as this is by far the most costly component of each pipeline. The complexity of a single full SVD on $\Delta_t \in \mathbb{R}^{n \times n}$ is $\mathcal{O}(n^3)$ (Vasudevan & Ramakrishna, 2017). Below, we detail the total computational complexity for each merging method and summarize them in Table 7:

- **Iso-C** performs a single SVD on the aggregated Δ_{TA} per layer, with total complexity:

$$\mathcal{O}(\text{Iso-C}) = \mathcal{O}(Ln^3) \tag{27}$$

- **TSV-M** performs:
  - T SVDs per layer on each task matrix, yielding $\mathcal{O}(TLn^3)$.
  - Two additional SVDs per layer for subspace alignment, yielding $\mathcal{O}(2Ln^3)$.

  Yielding the total complexity:

$$\mathcal{O}(\text{TSV-M}) = \mathcal{O}(TLn^3 + 2Ln^3) = \mathcal{O}\big((T+2)Ln^3\big) = \mathcal{O}(TLn^3) \tag{28}$$

- **Iso-CTS** performs:
  - One SVD on Δ_{TA} per layer, with complexity $\mathcal{O}(Ln^3)$.
  - One SVD on each Δ_t, for all T tasks per layer, with complexity $\mathcal{O}(TLn^3)$.
  - Two SVDs on two constructed matrices $U^*, V^* \in \mathbb{R}^{n \times n}$ per layer, yielding $\mathcal{O}(2Ln^3)$.

  Therefore, the total complexity equals:

$$\mathcal{O}(\text{Iso-CTS}) = \mathcal{O}(Ln^3 + TLn^3 + 2Ln^3) = \mathcal{O}\big((T+3)Ln^3\big) = \mathcal{O}(TLn^3) \tag{29}$$

- `PACT-Iso-C` (Ours) integrates our orthogonal projection filtering with the Iso-C framework. It performs:
  - $(T+1)$ SVDs for the `PACT` filtering phase (one on θ_0 and T on task matrices Δ_t): $\mathcal{O}\big((T+1)Ln^3\big)$.
  - One SVD on the aggregated filtered matrix $\sum \tilde{\Delta}_j$ for the final Iso-C merging: $\mathcal{O}(Ln^3)$.

Yielding the total complexity:

$$\mathcal{O}(\texttt{PACT-Iso-C}) = \mathcal{O}\big((T+1)Ln^3 + Ln^3\big) = \mathcal{O}\big((T+2)Ln^3\big) = \mathcal{O}(TLn^3) \quad (30)$$

(Note: The QR decompositions required to build the protection spaces have a complexity of $\mathcal{O}(n(TK)^2)$, which is strictly dominated by the $\mathcal{O}(n^3)$ SVD terms since $K \ll n$.)

- `PACT-TSV-M` (Ours) integrates `PACT` filtering with TSV-M merging. It performs:
  - $(T+1)$ SVDs for the `PACT` filtering phase: $\mathcal{O}\big((T+1)Ln^3\big)$.
  - $(T+2)$ SVDs applied to the filtered matrices $\tilde{\Delta}_t$ for the TSV-M merging phase: $\mathcal{O}\big((T+2)Ln^3\big)$.

  Yielding the total complexity:

$$\mathcal{O}(\texttt{PACT-TSV-M}) = \mathcal{O}\big((T+1)Ln^3 + (T+2)Ln^3\big) = \mathcal{O}\big((2T+3)Ln^3\big) = \mathcal{O}(TLn^3) \quad (31)$$

Efficient Variants with Randomized SVD (RSVD):
Since extracting task-specific features only requires the top-k singular vectors, RSVD can approximate these components using random projections, strictly dropping the decomposition complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 k_{max})$, where k_{max} is the maximum rank retained. However, SVDs utilized for final subspace alignment or full aggregation must remain exact.

- **Efficient TSV-M (RSVD)** replaces the first T task-specific SVDs, but the 2 final alignment SVDs cannot be substituted:

$$\mathcal{O}(\text{Efficient TSV-M}) = \mathcal{O}\big(TLn^2 k_{max} + 2Ln^3\big) \quad (32)$$

- **Efficient `PACT-Iso-C` (RSVD)** replaces all $(T+1)$ SVDs in the `PACT` phase, leaving only the final Iso-C aggregation SVD exact:

$$\mathcal{O}(\text{Efficient } \texttt{PACT-Iso-C}) = \mathcal{O}\big((T+1)Ln^2 k_{max} + Ln^3\big) \quad (33)$$

- **Efficient `PACT-TSV-M` (RSVD)** replaces the $(T+1)$ SVDs in the `PACT` phase and the T task SVDs in the TSV-M phase, while the 2 final TSV-M alignment SVDs remain exact:

$$\mathcal{O}(\text{Efficient } \texttt{PACT-TSV-M}) = \mathcal{O}\big((2T+1)Ln^2 k_{max} + 2Ln^3\big) \quad (34)$$

Comparison Summary. As highlighted in the analysis, all exact subspace methods (TSV-M, Iso-CTS, `PACT-Iso-C`, and `PACT-TSV-M`) share the same asymptotic complexity scaling of $\mathcal{O}(TLn^3)$. Notably, our exact `PACT-Iso-C` strictly requires $(T+2)$ full SVDs, posing slightly less overhead than the $(T+3)$ required by Iso-CTS. Exact `PACT-TSV-M` unsurprisingly carries the highest constant factor with $(2T+3)$ SVDs due to the sequential combination of two distinct algorithms.

More importantly, our **Efficient RSVD Variants** fundamentally decouple the cubic complexity barrier from the number of tasks T. While a small, constant number of full SVDs (e.g., 1 for Iso-C, 2 for TSV-M) remains inescapable for the final aggregation steps, substituting the T-dependent operations with $\mathcal{O}(n^2 k_{max})$ practically eliminates the scaling bottleneck. Because the number of active vectors k_{max} is a tiny fraction of the matrix dimension ($k_{max} \ll n$), the computational cost scaling with T is reduced by orders of magnitude. This makes the heavily piped Efficient `PACT-TSV-M` highly scalable and drastically cheaper to compute than traditional exact merging pipelines on large-scale models.

C Additional Motivation Experiments

C.1 Extended Analysis on Fisher-level Parameter Collisions

Our parameter-level perturbation experiments, as detailed in Section 3.2.1, are extended here across 16 diverse task pairs (A, B) to substantiate the universality of the LBW hypothesis. Table 8 presents the downstream performance of task A under different modification schemes with an attack strength of $\alpha = 30$. We observe a consistent trend across all evaluated pairs:

Method	**# Full SVDs** ($\mathcal{O}(n^3)$)	**# RSVDs** ($\mathcal{O}(n^2 k_{max})$)	**Total Complexity**
Exact Methods (Baselines & Ours)			
Iso-C (Marczak et al., 2025)	1	0	$\mathcal{O}(Ln^3)$
TSV-M (Gargiulo et al., 2025)	$T+2$	0	$\mathcal{O}\big((T+2)Ln^3\big)$
Iso-CTS (Marczak et al., 2025)	$T+3$	0	$\mathcal{O}\big((T+3)Ln^3\big)$
PACT-Iso-C (Ours)	$T+2$	0	$\mathcal{O}\big((T+2)Ln^3\big)$
PACT-TSV-M (Ours)	$2T+3$	0	$\mathcal{O}\big((2T+3)Ln^3\big)$
Efficient Variants with Randomized SVD (RSVD)			
Efficient TSV-M	2	T	$\mathcal{O}\big(TLn^2k_{max} + 2Ln^3\big)$
Efficient PACT-Iso-C (Ours)	1	$T+1$	$\mathcal{O}\big((T+1)Ln^2k_{max} + Ln^3\big)$
Efficient PACT-TSV-M (Ours)	2	$2T+1$	$\mathcal{O}\big((2T+1)Ln^2k_{max} + 2Ln^3\big)$

Table 7: Comparison of computational complexity among different merging methods. T and L denote the number of tasks and layers, respectively. n is the dimension of the square layer matrices, and k_{max} ($k_{max} \ll n$) is the maximum number of vectors retained in RSVD approximations.

(A,B)	$M0$	$M1$	$M2$	$M3$	$M4$
(SUN397, Cars)	81.32	81.22	58.40	77.03	73.58
(Cars, SUN397)	86.49	86.61	5.04	77.93	70.64
(GTSRB, MNIST)	99.92	99.92	28.04	93.39	90.47
(MNIST, GTSRB)	99.50	99.50	39.52	99.20	96.69
(DTD, RESISC45)	98.40	98.40	92.02	97.69	97.34
(RESISC45, DTD)	97.04	97.09	74.94	96.38	94.94
(Cars, PCAM)	86.49	86.36	2.70	33.37	51.76
(PCAM, Cars)	97.86	97.84	94.05	97.76	97.80
(Food101, Flower102)	89.56	89.60	55.29	84.01	83.16
(Flower102, Food101)	97.06	97.06	62.75	93.14	90.20
(GTSRB, DTD)	99.92	99.92	77.03	99.49	99.77
(DTD, GTSRB)	98.40	98.40	50.71	94.50	90.60
(SUN397, OxfordIIITPet)	81.32	81.17	55.83	78.32	77.45
(OxfordIIITPet, SUN397)	94.84	94.84	51.54	93.03	91.94
(CIFAR100, RenderedSST2)	91.08	91.12	87.40	90.25	90.06
(RenderedSST2, CIFAR100)	77.31	77.46	72.64	77.02	76.88

Table 8: Accuracy (%) under different parameters modifications. The value of α is set to 30.0.

Severe Degradation under Crucial Attack ($M2$) Modifying the parameters within the crucial mask ($M2$) leads to a drastic performance drop. For instance, in the (Cars, SUN397) pair, the accuracy of task A (Cars) collapses from a baseline of 86.49% to 5.04%. Similarly, for (Cars, PACM), the accuracy drops to 2.70%. This massive collapse indicates that these small-magnitude, high-Fisher parameters are highly sensitive and hold critical task-specific information.

Robustness under Safe and Random Attacks ($M3$, $M4$) Conversely, modifying the parameters in the safe mask ($M3$) or the random mask ($M4$) with the same intensity yields considerably milder degradation. For example, under $M3$ and $M4$, the performance on Cars in the (Cars, SUN397) pair remains at 77.93% and 70.64%, respectively. These results confirm that the "load-bearing" property is highly localized and successfully captured by our Fisher-based identification method.

To confirm whether model merging interferes with these LBW dimensions, we evaluate the effect of restoring the crucial mask parameters to their pre-trained values θ_0 after a naive merging of task vectors $\theta_0 + \Delta_A + \Delta_B$. As shown in Table 9, across almost all task pairs, simply resetting the crucial mask parameters (which constitute only $0.80\% - 2.97\%$ of the total parameters) back to θ_0 yields a measurable performance recovery Δ_{acc}. Specifically, we observe an improvement of $+2.33\%$ for (Cars, PACM) and $+2.13\%$ for (DTD, GTSRB). This empirical recovery demonstrates that the task vector of the counterpart task (Δ_B) indeed introduces destructive interference to the crucial pre-trained weights of task A during naive addition.

(A,B)	crucial mask ratio(%)	$\theta_0 + \Delta_A + \Delta_B$	θ_0 on crucial	Δ_{acc}
(SUN397, Cars)	1.28	70.97	71.78	0.81
(Cars, SUN397)	1.52	78.62	79.36	0.74
(GTSRB, MNIST)	2.00	91.55	93.47	1.91
(MNIST, GTSRB)	2.97	99.08	99.14	0.06
(DTD, RESISC45)	1.19	94.15	94.68	0.53
(RESISC45, DTD)	1.48	94.29	94.60	0.32
(Cars, PCAM)	1.89	74.69	77.03	2.33
(PCAM, Cars)	0.80	97.50	97.46	0.04
(Food101, Flower102)	2.20	84.92	85.46	0.54
(Flower102, Food101)	1.12	93.14	93.14	0.00
(GTSRB, DTD)	1.91	99.89	99.89	0.00
(DTD, GTSRB)	2.17	92.55	94.68	2.13
(SUN397, OxfordIIITPet)	1.28	76.38	76.88	0.50
(OxfordIIITPet, SUN397)	1.87	92.66	92.66	0.00
(CIFAR100, RenderedSST2)	0.92	89.02	88.98	0.04
(RenderedSST2, CIFAR100)	1.26	76.16	76.01	0.14

Table 9: Δ_{acc} represents the accuracy (%) change due to the crucial mask replacement, with improvement in red and decrease in blue.

To thoroughly evaluate the sensitivity of the identified load-bearing parameters, we visualize the downstream accuracy of task A under varying attack strengths $\alpha \in [1.0, 5.0, 10.0, 20.0, 30.0]$ across all 16 evaluated task pairs in Figure 4. The curves show a systematic and consistent divergence. Across all 16 pairs, the $M2$ curve (representing the targeted attack on the crucial mask) drops precipitously as the attack strength α increases. In contrast, the $M3$ (safe mask) and $M4$ (random mask) curves exhibit a much gentler decline, remaining relatively stable even under high perturbation levels. This uniform behavior across diverse task combinations suggests that the load-bearing wall phenomenon is not an artifact of a specific dataset pair, but rather a structural characteristic inherent to pre-trained vision models.

C.2 Extended Geometric Subspace Analysis and Layer-wise Heterogeneity

This subsection provides the complete, expanded experimental results for the global subspace ablation analysis introduced in Section 3.2.2 of the main text. While Table 3 in Section 3.2.2 presents a concise subset of six representative tasks due to space constraints, we extend this evaluation here to the full set of 19 diverse downstream datasets on the ViT-B-16 model, as summarized in Table 10.

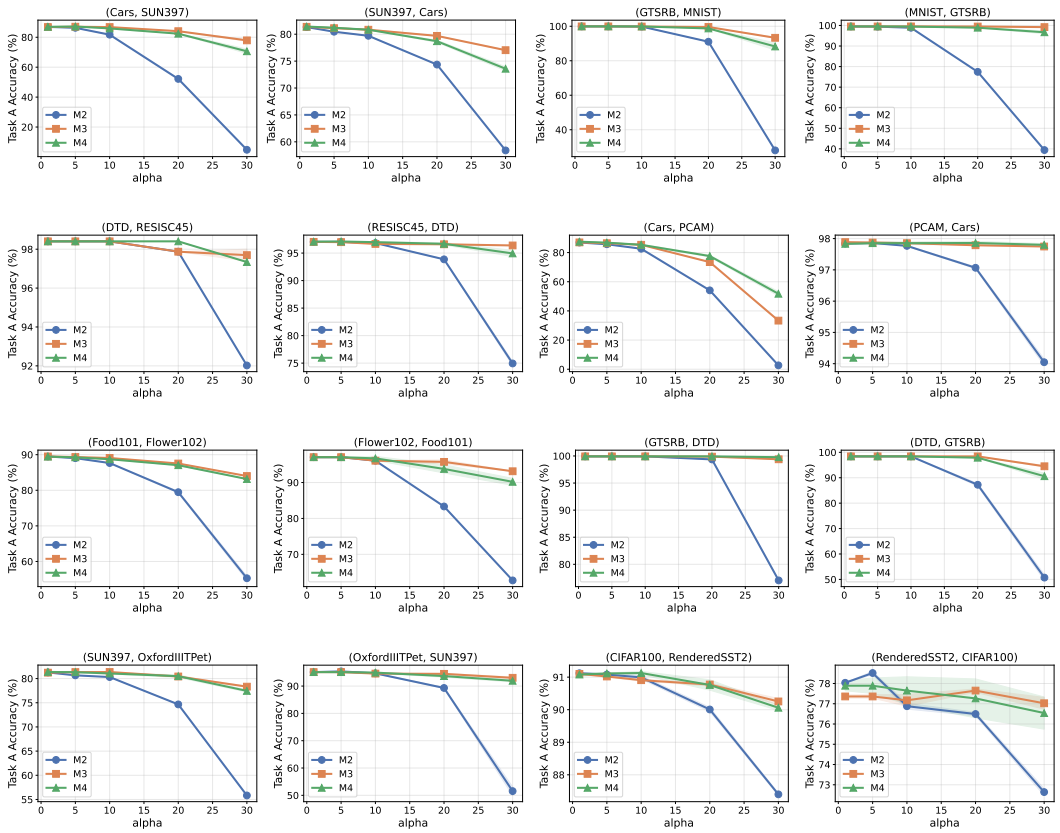


Figure 4: Accuracy (%) under varying attack strengths with respective to different masks across 16 task pairs. Generally, $M2$ is could be easily attacked while $M3$ and $M4$ are rather stable, indicating the importance of parameters within crucial mask.

The results consistently indicate that surgically removing the low-dimensional $V_{t,LBW}$ subspace (which represents on average only 1.17% of the total parameter dimensions) leads to a severe degradation of downstream performance across all 18 tasks. For instance, CIFAR100 accuracy falls from 91.08% to 1.88%, and Flowers102 drops from 97.06% to 0.98%. In contrast, ablating a random orthogonal control subspace of the identical dimensionality (V_{rand}) causes negligible performance changes, retaining 88.15% and 91.18% accuracy respectively. The ratio of the resulting performance drops ($\Delta_{acc}^{LBW}/\Delta_{acc}^{rand}$) ranges from $13.75\times$ to over $3104\times$. This notable asymmetry provides supportive empirical evidence indicating that these low-rank pre-trained subspaces are highly critical for sustaining downstream capabilities.

To analyze the underlying geometric conflicts causing these performance drops, we supplement the main text experiments by evaluating the pairwise Sacred Space Similarity (Sim, Eqn. 5) and Hidden Interference Ratio (Inf, Eqn. 5). When these metrics are averaged globally across all network layers (Figures 5 and 6), task similarities appear deceptively high (> 0.91), and the pairwise interference appears mild ($4\% - 11\%$).

However, we find that evaluating these metrics globally hides severe local conflicts due to a distinct, non-uniform layer-wise distribution. When we rank the layers and isolate the top-6 worst-affected layers (Figures 7 and 8), a starkly different geometric landscape emerges. In these critical foundational layers, the similarity (Sim) drops significantly to the $0.75 - 0.87$ range, and the Hidden Interference (Inf) skyrockets, indicating that explicit updates from one task can overwrite up to 25% of the implicit pre-trained dependencies of another (e.g., SUN397 and Cars invading SVHN's space).

Task t	baseline θ_t	θ_{abla} (Δ_{acc}^{LBW})	θ_{rand} (Δ_{acc}^{rand})	$\Delta_{acc}^{LBW}/\Delta_{acc}^{rand}$
CIFAR100	91.08	1.88 (89.20)	88.15 (2.93)	30.48×
CIFAR10	98.42	10.82 (87.60)	97.89 (0.53)	166.33×
Cars	86.49	0.37 (86.12)	80.22 (6.27)	13.75×
DTD	98.40	3.19 (95.21)	98.05 (0.35)	268.50×
EMNIST	99.78	17.00 (82.78)	99.75 (0.03)	3104.25×
FER2013	44.43	16.34 (28.08)	43.47 (0.95)	29.49×
FashionMNIST	95.28	9.78 (85.50)	95.14 (0.14)	610.71×
Flowers102	97.06	0.98 (96.08)	91.18 (5.88)	16.33×
Food101	89.56	0.96 (88.60)	85.99 (3.57)	24.84×
GTSRB	99.92	1.91 (98.01)	99.86 (0.06)	1566.60×
KMNIST	99.94	11.92 (88.02)	99.88 (0.06)	1467.00×
OxfordIIITPet	94.84	2.17 (92.66)	93.75 (1.09)	85.25×
PCAM	97.86	49.10 (48.76)	96.76 (1.10)	44.33×
RESISC45	97.04	4.18 (92.86)	95.98 (1.06)	87.75×
RenderedSST2	77.31	54.91 (22.40)	75.63 (1.69)	13.29×
STL10	99.60	11.00 (88.60)	99.33 (0.27)	332.25×
SUN397	81.32	0.56 (80.77)	76.19 (5.13)	15.74×
SVHN	96.76	7.40 (89.36)	96.51 (0.25)	352.74×

Table 10: Δ_{acc}^{*} represents the accuracy (%) change due to the subspace filtering. The random subspace filtering is averaged across three random seeds. The numbers in parentheses indicate the decrease in accuracy compared to the θ_t.

This sharp contrast reveals that the degree of pairwise task conflict is highly dependent on layer depth, with early foundational layers absorbing severe, high-density interference. This pronounced layer-wise variation suggests that a uniform, global merging approach is insufficient, motivating a dedicated investigation into how these conflicts are distributed layer-by-layer across the pre-trained architecture.

C.3 Layer-wise and Component-wise Analysis of Pre-trained Subspace Intrusion

To systematically investigate how task updates interact with the pre-trained architecture under a subspace geometric view, we conduct a layer-wise analysis. This investigation requires a metric that directly quantifies how task-specific updates modify the shared, pre-trained representation space, rather than measuring pairwise task-to-task relationships. To this end, motivated by the projection strength metrics (Saha et al., 2021) and subspace alignment formulations (Fernando et al., 2013), we define the **Intrusion Energy (E_{in})** for a specific task t at layer ℓ as:

$$E_{in}(t, \ell) = \frac{\|\Delta_t V_{0,K}\|_F^2}{\|\Delta_t\|_F^2} \tag{35}$$

where $V_{0,K} \in \mathbb{R}^{n \times K}$ represents the top-K right singular vectors of the pre-trained weights θ_0, capturing the core coordinates of general pre-trained knowledge. The objective of E_{in} is to measure the exact proportion of a task's explicit update energy (Δ_t) that projects directly onto this shared pre-trained core. A higher E_{in} indicates that fine-tuning has aggressively reshaped the foundational pre-trained representation, creating a high-energy "intrusion" that is vulnerable to overwrite when merging.

C.3.1 Global Layer-wise Intrusion Dynamics

Figure 10 plots the average E_{in} across the transformer blocks of ViT-B-16, revealing a clear layer-wise heterogeneity:

Foundational Bottlenecks in Early Blocks. In early blocks (blocks 0 to 2), which extract fundamental visual representations (e.g., textures, edges), E_{in} is heavily elevated. For instance, the

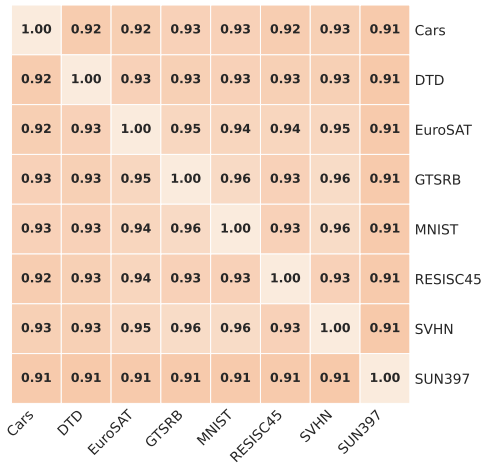


Figure 5: Sim (Global Avg)

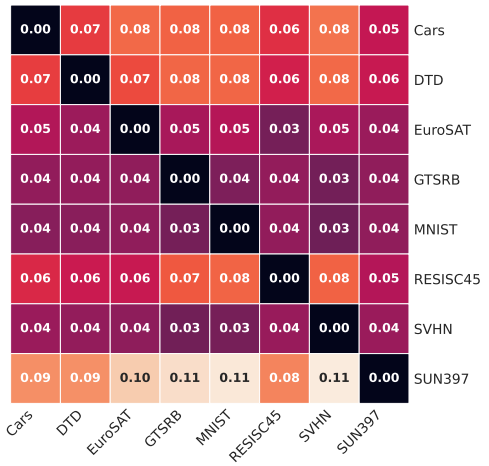


Figure 6: Inf (Global Avg)

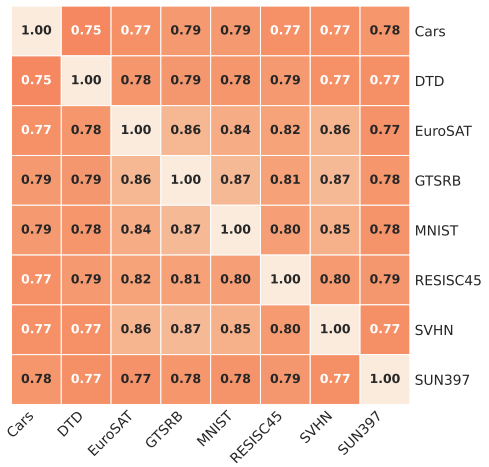


Figure 7: Sim (Top-6 Layers)

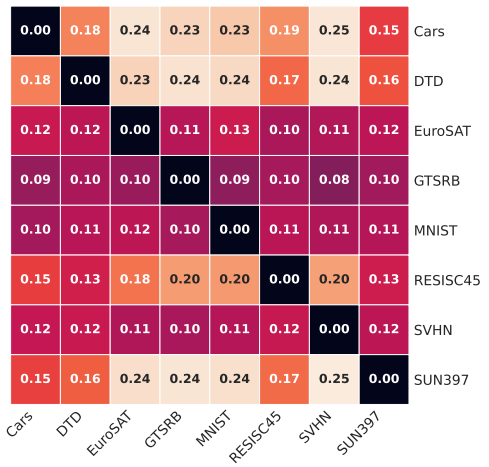


Figure 8: Inf (Top-6 Layers)

Figure 9: Comparison of global averages versus the top-6 worst-affected layers for Sacred Space Similarity (Sim) and Hidden Interference Ratio (Inf).

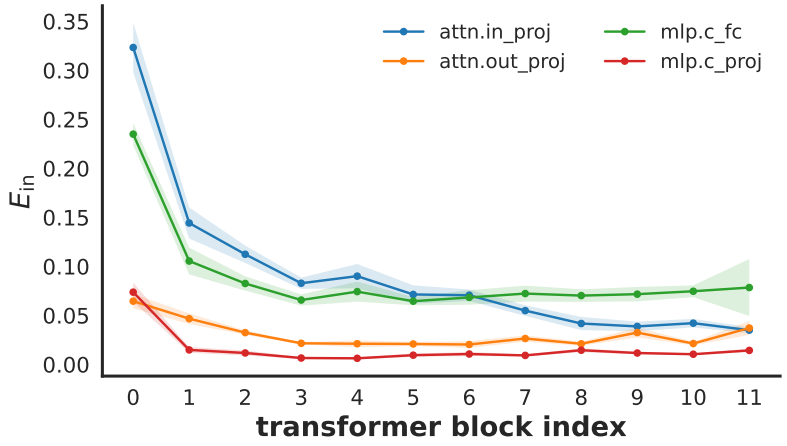


Figure 10: Layer-wise Intrusion Energy E_{in} across the transformer blocks of ViT-B-16.

intrusion energy in the input projection layers exceeds 30%. This indicates that downstream specialization forces task updates to aggressively modify these early representation bottlenecks, rendering them highly sensitive to cross-task interference.

Orthogonal Routing in Deeper Blocks: Conversely, in deeper layers (blocks 5 to 11), E_{in} drops significantly and plateaus below 5%. This confirms that deeper layers safely route high-level semantic updates into task-specific orthogonal subspaces with minimal perturbation of the core general-purpose representations.

This layer-wise distribution highlights the risk of "the averaging illusion": a 48% structural corruption at foundational block 0 cannot be resolved by deeper blocks. Treating all layers uniformly allows high-energy foundational collisions to proceed unchecked, leading to downstream performance degradation.

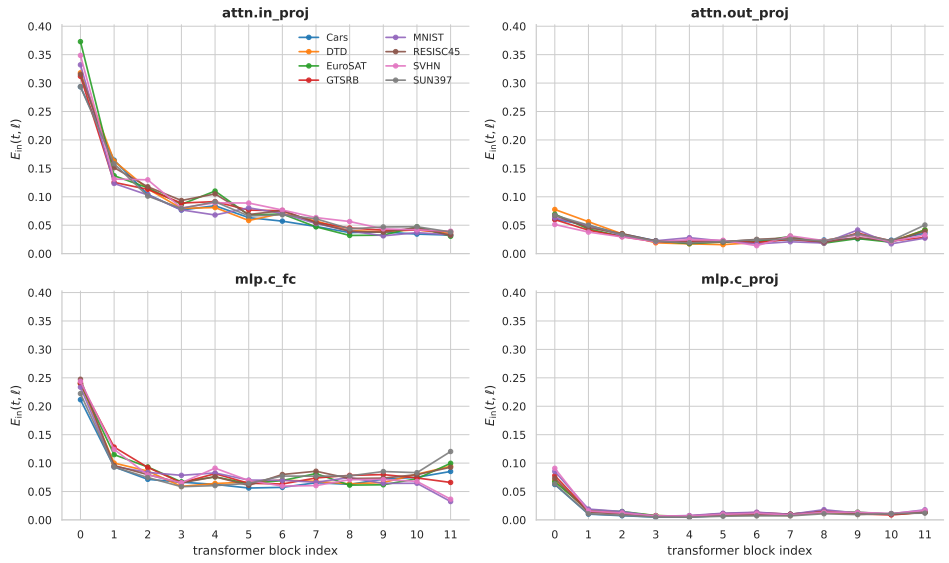


Figure 11: Task-specific and Component-wise Intrusion Energy E_{in} across different transformer components.

C.3.2 Component-wise and Task-wise Intrusion Dynamics

To evaluate these dynamics at a more granular scale, Figure 11 decomposes E_{in} across individual tasks and specific transformer components (`attn.in_proj`, `attn.out_proj`, `mlp.c_fc`, and `mlp.c_proj`).

We observe a strong component-wise heterogeneity. The input-facing projection layers (`attn.in_proj` and `mlp.c_fc`) absorb nearly 40% of the update energy at block 0, acting as the primary bottleneck of task intrusion. In contrast, the output projection layers (`attn.out_proj` and `mlp.c_proj`) remain flat and low (mostly below 5%), confirming they act primarily as feature routers that do not perturb the shared core. Importantly, the tightly clustered curves across 8 diverse downstream datasets demonstrate that this early-layer collision is a universal geometric characteristic of ViT fine-tuning rather than an artifact of a specific task.

This universality is further illustrated in the exhaustive heatmap in Figure 12. Across all evaluated tasks, the earliest blocks (e.g., `b00.attn.in_proj`, `b00.mlp.c_fc`) manifest as prominent dark bands of intense, high-energy intrusion into the pre-trained space. Beyond block 4, the heatmap transitions consistently to light hues, confirming that the pre-trained core remains undisturbed in deeper blocks.

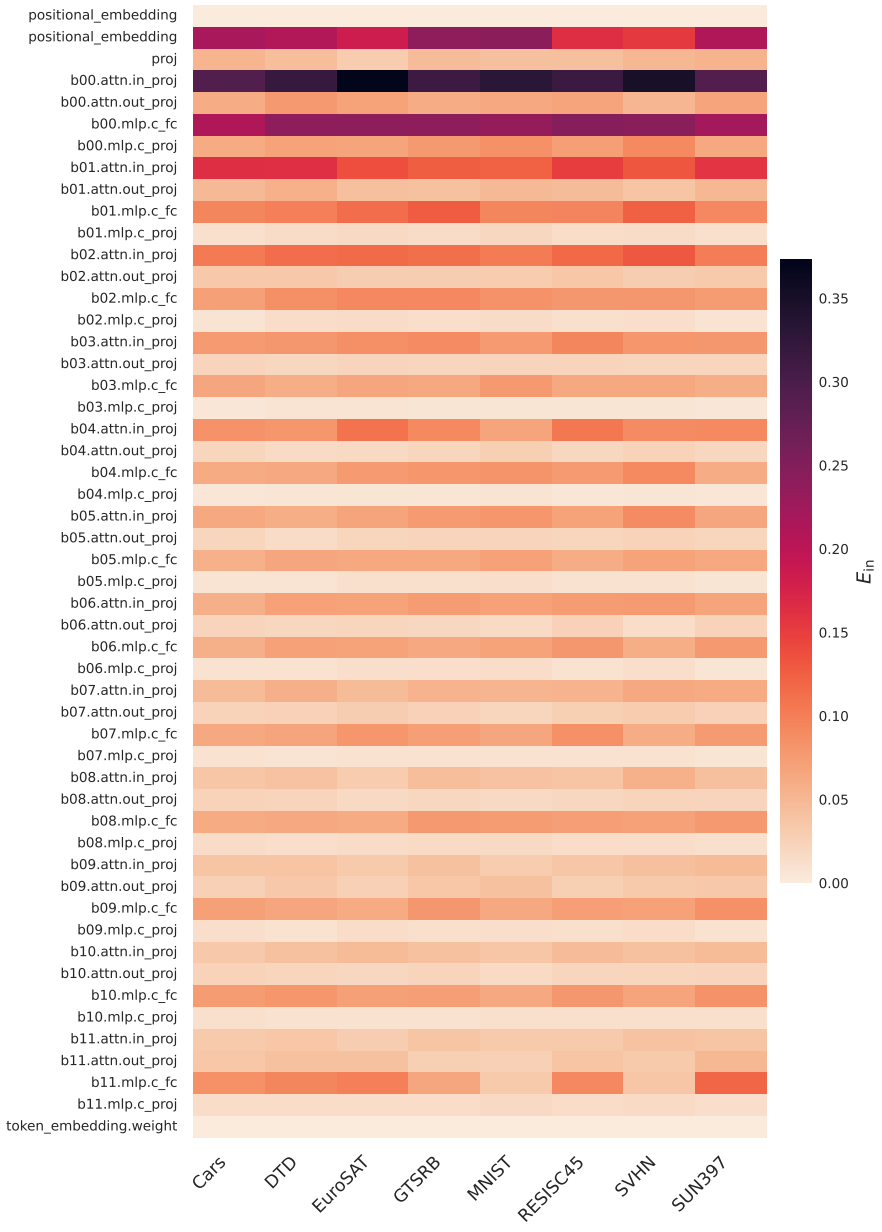


Figure 12: Comprehensive heatmap of Intrusion Energy (E_{in}) across all layers and tasks.

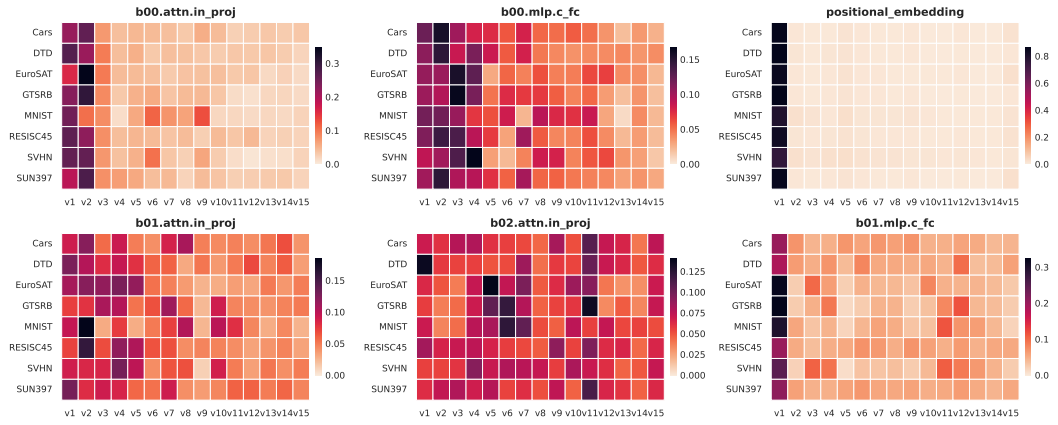


Figure 13: Fractional Intrusion Energy distributed across individual right singular vectors (v_1 to v_{15}) for select top-affected layers.

C.3.3 Microscopic Intrusion Distribution Along Principal Axes

To understand the geometric mechanics of this intrusion at the finest scale, we analyze which specific singular vectors within the pre-trained core subspace ($K = 15$) are affected. Figure 13 visualizes the fractional intrusion energy distributed across the individual right singular vectors (v_1 to v_{15}) for the most heavily affected layers.

The visualization reveals that task updates do not uniformly perturb the core space. Instead, they target specific principal axes in a highly task-dependent manner. For example, in the foundational layer `b00.mlp.c_fc`, SVHN heavily modifies v_4, whereas EuroSAT targets v_3. Additionally, in the `positional_embedding` layer, the intrusion across all tasks is almost exclusively concentrated on the single most dominant vector (v_1), indicating a universal shift in positional priors during downstream specialization.

This vector-wise heterogeneity highlights the necessity of PACT's design. Because different tasks rely on and explicitly attack different combinations of principal axes, simply discarding a fixed, task-agnostic set of singular vectors is insufficient. The adaptive, QR-decomposition-based global shield implemented in PACT is essential to dynamically map and protect these highly heterogeneous, task-specific attack vectors.

D Additional Evaluation

D.1 Comparison with Counterparts

While both CSP and PACT recognize the critical role of pretrained weights in model merging, they diverge fundamentally in their core insights. CSP is grounded in the observation that pretrained weights are already near-optimal within a global, task-agnostic core subspace defined by dominant singular vectors. It posits that updates within this subspace are unlikely to improve performance and may instead disrupt shared generalization capabilities. In contrast, PACT challenges the implicit assumption that task-specific knowledge is fully captured by task vectors alone. We identify LBW dimensions—task-critical parameters that remain anchored in the pretrained model with negligible fine-tuning updates—and demonstrate that these heterogeneous, task-specific structures are vulnerable to being overwritten during merging.

This distinction leads to qualitatively different protection targets. CSP treats interference as a global drift away from the pretrained optimum and protects a single, universal core subspace shared across all tasks. While effective at preserving broad generalization, this approach cannot distinguish between harmful drift and task-specific anchoring, potentially discarding vital task knowledge or failing to prevent pairwise collisions among tasks whose critical pretrained dependencies overlap. PACT, by contrast, frames interference as a geometric collision between the explicit updates of one

task and the implicit LBW subspaces of others. Consequently, PACT explicitly models and protects the distinct LBW subspaces required by each individual task, rather than enforcing a single global safe zone.

The differing perspectives also manifest in fundamentally different operational mechanisms. CSP mitigates interference through a static, post-hoc projection that uniformly constrains all task updates outside the shared core subspace. This global projection acts as a blanket filter applied identically to every task vector. PACT instead constructs a dynamic, task-aware global orthogonal shield for each expert by aggregating the LBW subspaces of all other tasks. This mutual shielding mechanism surgically filters out only those components of a task vector that would invade the load-bearing structures of its peers, thereby preserving task-specific knowledge that CSP's global projection might inadvertently remove.

Ultimately, PACT operates at a finer granularity than CSP, actively respecting the heterogeneous pretrained foundations upon which individual tasks depend. Rather than assuming a uniform safe subspace suffices for all tasks, PACT acknowledges that different tasks rely on different subsets of pretrained parameters and designs its protection mechanism accordingly. This task-aware, geometrically precise approach addresses interference patterns that global preservation alone cannot resolve, establishing PACT as a complementary advancement that builds upon—but goes beyond—the foundational insights of CSP.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Task Arithmetic	70.8(76.5)	65.3(72.1)	60.5(66.8)	75.4(79.6)	70.5(75.9)	65.8(70.8)	84.9(88.7)	79.4(84.0)	74.0(78.1)
CSP-TA	72.6(78.1)	67.1(74.1)	62.6(69.2)	77.6(82.0)	71.9(77.5)	67.0(72.1)	86.7(90.5)	81.7(86.5)	76.0(80.3)
PACT-TA	75.2(81.0)	66.4(73.3)	59.9(66.4)	85.5(90.3)	76.8(82.7)	68.7(74.0)	92.8(96.9)	87.0(92.0)	83.0(87.7)
Iso-C	86.3(92.9)	80.3(88.1)	75.5(82.5)	90.6(95.6)	84.8(91.1)	79.6(85.4)	94.2(98.3)	89.3(94.5)	87.6(92.2)
CSP-Iso-C	87.1(93.9)	80.3(88.1)	76.0(83.1)	91.2(96.4)	85.1(91.4)	80.4(86.2)	95.0(99.2)	89.7(94.8)	88.6(93.2)
PACT-Iso-C	87.3(94.1)	81.3(89.0)	78.7(86.0)	92.0(97.2)	86.1(92.3)	83.4(89.3)	95.1(98.3)	90.5(95.5)	90.3(95.0)
TSV-M	85.9(92.3)	80.1(87.9)	77.1(84.3)	89.0(93.9)	84.6(91.0)	80.6(86.5)	93.0(97.0)	89.2(94.4)	87.7(92.5)
CSP-TSV-M	86.8(93.4)	79.9(87.3)	77.6(84.8)	89.8(94.8)	84.0(90.0)	81.0(86.7)	93.9(98.0)	89.4(93.2)	88.0(92.6)
PACT-TSV-M	88.1(94.8)	82.0(89.6)	80.3(87.6)	91.2(96.3)	85.4(91.5)	83.8(89.6)	94.4(98.5)	89.5(94.4)	89.7(94.3)

Table 11: Comparison of PACT against CSP on various methods. We present average absolute accuracy and average normalized accuracy (in bracket) in %.

D.2 Evaluation on Additional Language Benchmarks

Following FusionBench (Tang et al., 2024), we merge GPT-2 (Radford et al., 2019) models on seven GLUE tasks, each with a different classification head, where all tasks are evaluated solely by accuracy. As shown in Table 12, PACT outperforms all other merging methods, further validating the LBW effect in model merging for NLP.

E Detailed Ablation Studies

To systematically evaluate the contribution of each design component in PACT, we conduct extensive ablation studies. This section is organized into five parts: (1) an algorithmic component ablation comparing PACT against a static core-filtering baseline; (2) a layer-wise depth ablation exploring the performance of localized filtering; (3) a component-wise filtering selection ablation; (4) a sensitivity analysis of the hyper-parameters; and (5) an efficient alternative for PACT. To more objectively demonstrate the effects of various ablation studies, we report the performance of all baseline methods used in this section based on our own reproduction. For instance, the performance of the Iso-C baseline exhibits slight variations.

Method	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	Avg.	Norm Avg.
Individual	76.8	82.1	80.4	88.3	89.6	65.3	91.2	82.0	-
Weight Averaging	55.0	55.1	51.0	57.6	76.7	44.8	52.5	56.1	69.2
Fisher	54.8	58.0	39.5	63.3	81.5	49.1	64.7	58.7	71.4
RegMean	61.7	70.4	65.4	69.7	78.8	56.0	79.7	68.8	83.9
Iso-C	68.1	61.2	**70.1**	60.8	85.5	57.4	78.3	68.8	84.1
TIES	68.4	71.4	68.4	69.6	82.4	47.7	81.8	70.0	85.0
Task Arithmetic	68.7	68.6	69.6	**70.5**	81.8	47.3	83.6	70.0	85.0
PACT-TA	**68.7**	70.7	65.9	70.1	84.0	48.7	85.7	**70.5**	**85.6**
TSV-M	65.6	75.4	58.6	64.4	86.2	55.6	85.7	70.2	85.5
PACT-TSV-M	63.0	**77.4**	50.0	70.0	**86.5**	**59.2**	**86.1**	70.3	**85.6**

Table 12: Results of merging GPT-2 models on seven GLUE tasks. We present average absolute accuracy and average normalized accuracy in %. Best performances are shown in **bold**.

E.1 Algorithmic Component Ablation: Dynamic Task Shields vs. Static PACT (S-PACT)

To verify the necessity of dynamically extracting the active task subspaces (the top-k task-specific SVD) and constructing the mutual protection shields, we compare PACT against a simplified, static baseline which we term S-PACT (Static PACT).

S-PACT bypasses the task-vector SVD and mutual projection phases entirely. Instead, it only performs a single SVD on the pre-trained weight matrix θ_0 to extract the top-K core directions $V_{0,K}$, and then directly projects each task matrix onto its orthogonal complement:

$$\tilde{\Delta}_j = \Delta_j(I - V_{0,K}V_{0,K}^\top) \tag{36}$$

This baseline effectively filters task updates against the pre-trained foundation but ignores the distinct, task-specific subspaces of other expert models. S-PACT is computationally cheaper as it requires only one SVD per layer and no subsequent QR decompositions.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
TA	70.8(76.5)	65.3(72.1)	60.5(66.8)	75.4(79.6)	70.5(75.9)	65.8(70.8)	84.9(88.7)	79.4(84.0)	74.0(78.1)
S-PACT-TA	70.8(76.3)	65.8(72.7)	61.7(68.3)	76.2(80.5)	71.2(76.7)	66.4(71.6)	85.0(88.7)	80.3(85.1)	74.7(78.9)
PACT-TA	75.2(81.0)	66.4(73.3)	59.9(66.4)	85.5(90.3)	76.8(82.7)	68.7(74.0)	92.8(96.9)	87.0(92.0)	83.0(87.7)
Iso-C	86.6(93.3)	79.5(87.3)	75.1(82.2)	90.8(96.0)	84.4(90.7)	79.7(85.4)	94.7(98.9)	89.2(94.3)	87.7(92.4)
S-PACT-Iso-C	86.5(93.2)	79.5(87.2)	74.8(81.7)	90.8(96.0)	84.2(90.4)	79.7(85.6)	94.7(98.9)	89.3(94.4)	87.7(92.4)
PACT-Iso-C	87.3(94.1)	81.3(89.0)	78.7(86.0)	92.0(97.2)	86.1(92.3)	83.4(89.3)	95.1(98.3)	90.5(95.5)	90.3(95.0)

Table 13: Algorithmic ablation study comparing PACT against the core-only S-PACT baseline across different model backbones. We present average absolute accuracy and average normalized accuracy (in bracket) in %.

Table 13 summarizes the merging performance of S-PACT against TA, Iso-C, and full PACT. The empirical results reveal that merely performing static core-filtering (S-PACT) yields limited performance improvements. For instance, on ViT-B/16 (14 tasks), S-PACT-TA only marginally improves over TA (71.2% vs. 70.5%), while S-PACT-Iso-C slightly degrades the original Iso-C baseline (84.2% vs. 84.4%).

In contrast, PACT-TA and PACT-Iso-C achieve substantial performance improvements, reaching 76.8% and 86.1% respectively on the same setting. This difference indicates that simply protecting the static pre-trained core is insufficient. To prevent task updates from blindly overwriting each other's distinct downstream manifolds, it is critical to dynamically extract the active task dimensions and construct mutually orthogonal protection shields.

E.2 Layer-wise Depth Ablation: Localized Filtering for Efficiency

In Appendix C.3, we demonstrated that the energy intrusion effect is highly localized within the early foundational layers of the model, whereas deeper layers route semantics in mutually orthogonal

directions. Based on this observation, we perform an ablation on the depth of layers filtered by PACT to identify potential efficiency-performance trade-offs. We compare applying PACT globally to 100% of the layers versus applying it to various percentages of early foundational layers.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
TA	70.8(76.5)	65.3(72.1)	60.5(66.8)	75.4(79.6)	70.5(75.9)	65.8(70.8)	84.9(88.7)	79.4(84.0)	74.0(78.1)
25% PACT-TA	72.8(78.6)	**67.3(74.3)**	**62.3(68.9)**	77.9(82.3)	72.5(78.0)	66.9(72.1)	86.5(90.2)	81.7(86.5)	76.1(80.3)
100% PACT-TA	**75.2(81.0)**	66.4(73.3)	59.9(66.4)	**85.5(90.3)**	**76.8(82.7)**	**68.7(74.0)**	**92.8(96.9)**	**87.0(92.0)**	**83.0(87.7)**
TSV-M	85.9(92.3)	80.1(87.9)	77.1(84.3)	89.0(93.9)	84.6(91.0)	80.6(86.5)	93.0(97.0)	89.2(94.4)	87.7(92.5)
25% PACT-TSV-M	86.5(93.1)	79.8(87.2)	77.2(84.2)	89.5(94.4)	83.7(89.8)	80.7(86.3)	93.6(97.6)	88.0(92.9)	87.6(92.2)
100% PACT-TSV-M	**88.1(94.8)**	**82.0(89.6)**	**80.3(87.6)**	**91.2(96.3)**	**85.4(91.5)**	**83.8(89.6)**	**94.4(98.5)**	**89.5(94.4)**	**89.7(94.3)**
Iso-C	86.6(93.3)	79.5(87.3)	75.1(82.2)	90.8(96.0)	84.4(90.7)	79.7(85.4)	94.7(98.9)	89.2(94.3)	87.7(92.4)
25% PACT-Iso-C	86.9(93.6)	80.0(87.7)	75.6(82.7)	90.9(96.1)	84.7(90.9)	80.0(85.8)	94.8(99.0)	89.4(94.5)	88.0(92.7)
50% PACT-Iso-C	**87.8(94.7)**	81.8(89.6)	78.8(85.9)	91.6(96.9)	85.8(92.1)	82.2(88.0)	95.0(99.2)	90.0(95.1)	89.2(93.9)
75% PACT-Iso-C	87.7(94.4)	**82.2(90.1)**	**79.1(86.2)**	**92.1(97.4)**	**86.5(92.9)**	83.2(89.1)	**95.2(99.3)**	**90.5(95.6)**	90.0(94.8)
100% PACT-Iso-C	87.3(94.1)	81.3(89.0)	78.7(86.0)	92.0(97.2)	86.1(92.3)	**83.4(89.3)**	95.1(98.3)	90.5(95.5)	**90.3(95.0)**

Table 14: Ablation study on filtering depth comparing localized early-layer PACT filtering under various percentages versus 100% global PACT filtering. We present average absolute accuracy and average normalized accuracy (in bracket) in %. Best performances are shown in **bold**.

Table 14 presents the comparison across three base merging algorithms (TA, Iso-C, TSV-M). Overall, integrating PACT systematically improves the multi-task accuracy of all three baseline methods across all evaluated backbones and task scales. Specifically, while baseline methods suffer from performance degradation as the task load increases, the PACT-enhanced variants successfully mitigate this drop, with the best-performing configurations generally concentrated at 75% or 100% filtering depths. To rigorously trace the depth-wise trajectory of this base-protection effect, we focus our analysis primarily on PACT-IsoC, which provides a complete gradient of filtering depths (25%, 50%, 75%, and 100%).

For PACT-IsoC, we observe a non-monotonic performance curve as the filtering depth increases. Moving from the baseline Iso-C to 25% and 50% PACT-IsoC, the accuracy steadily climbs across all configurations, confirming that shielding the load-bearing walls in the early-to-mid layers is highly effective. Interestingly, the performance peaks at different depths depending on model capacity and task scale. For the majority of 8-task and 14-task scenarios on ViT-B/16 and ViT-L/14, the accuracy peaks at 75% depth (e.g., reaching 92.1% and 95.2% on 8 tasks) and slightly plateaus or declines at 100%. This indicates that while protecting the first 75% of the layers is essential due to high-energy visual bottlenecks, the deepest 25% of layers route high-level semantic representations in naturally orthogonal directions, making strict orthogonal constraints in these final blocks redundant or over-constraining.

This peak behavior dynamically shifts when scaling the backbone capacity or task load. For the highly parameter-constrained ViT-B/32 backbone, the optimal performance peaks earlier—at 50% depth for 8 tasks (87.8%) and at 75% depth for more tasks. Conversely, under the heaviest task load of 20 tasks on larger backbones (ViT-B/16 and ViT-L/14), the peak shifts to 100% global filtering (achieving 83.4% and 90.3%). This shift reveals that under extreme multi-task interference, the necessity of safeguarding the pre-trained core extends globally across all layers, and larger backbones possess sufficient dimensional capacity to accommodate the global orthogonal constraints without suffering from dimensionality lock.

E.3 Ablation on Component-Wise Projection Filtering

In Appendix C.3, we observed that task intrusion energy is highly concentrated in the input-facing projection layers, specifically attn.in_proj and mlp.c_fc. To further explore the underlying mechanism of the LBW effect, we investigate whether the critical pre-trained dependencies are strictly confined to these highly intruded layers, or if they are distributed globally across all projection components. To this end, we conduct an ablation experiment where we restrict PACT filtering exclusively to attn.in_proj and mlp.c_fc within each block, leaving the output-facing layers attn.out_proj and mlp.c_proj unfiltered.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
TSV-M	85.9(92.3)	80.1(87.9)	77.1(84.3)	89.0(93.9)	84.6(91.0)	80.6(86.5)	93.0(97.0)	89.2(94.4)	87.7(92.5)
PACT-TSV-M (select)	86.3(92.9)	79.3(86.7)	76.9(84.0)	89.5(94.5)	83.4(89.3)	80.5(86.2)	93.4(97.4)	87.8(92.6)	87.5(92.0)
PACT-TSV-M	**88.1(94.8)**	**82.0(89.6)**	**80.3(87.6)**	**91.2(96.3)**	**85.4(91.5)**	**83.8(89.6)**	**94.4(98.5)**	**89.5(94.4)**	**89.7(94.3)**
Iso-C	86.6(93.3)	79.5(87.3)	75.1(82.2)	90.8(96.0)	84.4(90.7)	79.7(85.4)	94.7(98.9)	89.2(94.3)	87.7(92.4)
PACT-Iso-C (select)	87.1(93.8)	80.5(88.2)	76.6(83.7)	91.2(96.4)	85.0(91.3)	80.6(86.4)	94.9(99.1)	89.9(94.9)	89.0(93.7)
PACT-Iso-C	**87.3(94.1)**	**81.3(89.0)**	**78.7(86.0)**	**92.0(97.2)**	**86.1(92.3)**	**83.4(89.3)**	**95.1(98.3)**	**90.5(95.5)**	**90.3(95.0)**

Table 15: Ablation study on layer selection PACT filtering. We present average absolute accuracy and average normalized accuracy (in bracket) in %. Best performances are shown in **bold**.

Table 15 presents the evaluation results under this restricted filtering setup. Restricting the filtering to only the high-intrusion layers consistently outperforms the baseline TSV-M and Iso-C methods. This outcome confirms that input-facing projection layers indeed constitute the primary bottleneck of task intrusion. However, compared to full PACT filtering, a persistent performance gap remains, especially under larger task scales. For instance, on the 20-task scenario of ViT-B/16, restricting the filtering yields an absolute accuracy of 80.5% for TSV-M and 80.6% for Iso-C, whereas full PACT achieves 83.8% and 83.4%, respectively.

This gap reveals that the LBW effect is a global phenomenon across the model's structural components rather than being entirely isolated within a few peak layers. Although the task intrusion energy is lower in output-facing projection layers, these components still contain critical pre-trained coordinates that require protection. Consequently, while restricting the filtering to high-intrusion layers shields the most vulnerable bottlenecks, safeguarding the entire core subspace globally across all projection components is essential to fully preserve task performance during merging.

E.4 Hyperparameter Sensitivity Analysis

(K, k)	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
(0,0)	86.6(93.3)	79.5(87.3)	75.1(82.2)	90.8(96.0)	84.4(90.7)	79.7(85.4)	94.7(98.9)	89.2(94.3)	87.7(92.4)
α	1.40	0.90	0.90	1.40	1.00	0.80	1.60	1.10	1.00
(38,32)	86.2(92.9)	79.5(88.1)	70.2(77.1)	91.8(97.0)	85.5(91.9)	80.0(85.8)	95.2(99.4)	90.4(95.5)	90.2(94.9)
α	2.30	2.80	4.30	2.50	2.70	5.20	2.40	2.00	3.20
(30,24)	86.6(93.4)	80.4(88.1)	76.6(83.7)	91.9(97.1)	86.1(92.4)	82.9(88.8)	95.2(99.4)	90.4(95.5)	90.2(94.9)
α	2.00	2.30	3.00	2.20	2.30	3.10	2.40	1.80	2.30
(23,16)	87.1(93.9)	81.2(88.8)	78.2(85.4)	92.0(97.2)	**86.4(92.7)**	83.1(89.0)	**95.2(99.4)**	90.5(95.6)	90.2(95.0)
α	1.90	1.90	2.20	2.00	1.80	2.30	2.20	1.60	1.90
(15,8)	**87.3(94.1)**	**81.3(89.0)**	**78.7(86.0)**	**92.0(97.2)**	86.1(92.3)	**83.4(89.3)**	95.1(98.3)	**90.5(95.5)**	**90.3(95.0)**
α	1.70	1.70	1.60	1.80	1.60	1.60	2.00	1.50	1.50

Table 16: Performance of PACT-Iso-C under different hyperparameters (K, k) and the corresponding scaling coefficient, where (0,0) represents the results of our reproduced Iso-C. We present average absolute accuracy and average normalized accuracy (in brackets) in %. The best performance is in **bold**.

To study the sensitivity of PACT to its primary hyperparameters, we refer to the detailed evaluation presented in Table 16. Table 16 examines the effects of varying the pre-trained core dimension K and the active task vector dimension k, alongside the corresponding base merging scaling coefficients α.

The sensitivity analysis indicates that PACT is relatively robust to different hyperparameter configurations. Specifically, the parameter pair $(K, k) = (15, 8)$ consistently yields favorable absolute and normalized accuracy across diverse dataset scales (8, 14, and 20 tasks) and various model backbones (ViT-B/32, ViT-B/16, and ViT-L/14). This stable behavior suggests that a moderate, low-rank projection configuration is sufficient to successfully segregate the shared general-purpose representation from the task-specific update manifolds, simplifying practical deployments.

In addition, we report the hyperparameter configurations on NLP experiments for reproduction. On RoBERTa, the hyperparameters (K, k) are set to be (1, 1) for TA, (15, 8) for TSV-M and (15, 10) for Iso-C. On GPT-2, the hyperparameters (K, k) are set to be (1, 1) for TA and (3, 2) for TSV-M.

E.5 Efficient Alternative

We evaluate the effect of replacing the full SVD in PACT with randomized SVD (RSVD) (Halko et al., 2011), as reported in Table 17. Under exactly the same hyperparameters (K, k, α), RSVD achieves performance on par with full SVD, and in one-third of the experiments it even surpasses the full SVD results, which is quite counterintuitive. We attribute this to the implicit denoising effect of random projection (Halko et al., 2011). Full SVD faithfully extracts all directions, including the potential overfitting noise hidden in Δ_t; in contrast, the random Gaussian projection used in RSVD inherently acts as a low-pass filter. It robustly captures the main load-bearing wall components while naturally smoothing out high-frequency noise. Thus, RSVD provides a "free lunch" for PACT: as a structural regularizer, it reduces the complexity by several orders of magnitude while simultaneously enhancing the generalization ability of the merged model. We summarize and compare the computational complexity of PACT and its variant against these baselines in Appendix Sec. B.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
TSV-M	85.8(92.3)	78.6(85.9)	76.3(83.4)	88.8(93.8)	83.1(89.1)	80.0(85.6)	93.0(97.1)	87.7(92.6)	87.0(91.5)
RSVD	85.5(92.0)	**78.8(86.2)**	76.1(83.4)	88.7(93.6)	82.9(88.9)	79.8(85.4)	93.0(97.0)	87.6(92.5)	86.9(91.5)
PACT-TA	75.2(81.0)	66.4(73.3)	59.9(66.4)	85.5(90.3)	76.8(82.7)	68.7(74.0)	92.8(96.9)	87.0(92.0)	83.0(87.7)
RSVD	**75.6(81.5)**	66.4(73.3)	59.9(66.4)	**85.6(90.5)**	76.8(82.7)	68.7(74.0)	**92.9(97.0)**	**87.3(92.2)**	83.0(87.6)
PACT-Iso-C	87.3(94.1)	81.3(89.0)	78.7(86.0)	92.0(97.2)	86.1(92.3)	83.4(89.3)	95.1(98.3)	90.5(95.5)	90.3(95.0)
RSVD	87.3(94.1)	**81.9(89.7)**	78.7(86.0)	92.0(97.3)	**86.3(92.7)**	83.2(89.1)	**95.2(99.3)**	90.5(95.5)	90.3(95.0)
PACT-TSV-M	88.1(94.8)	82.0(89.6)	80.3(87.6)	91.2(96.3)	85.4(91.5)	83.8(89.6)	94.4(98.5)	89.5(94.4)	89.7(94.3)
RSVD	88.1(94.8)	81.8(89.5)	80.0(87.2)	91.1(96.2)	85.3(91.4)	**83.9(89.7)**	**94.5(98.6)**	89.5(94.3)	89.7(94.3)

Table 17: Results of replacing SVD with RSVD. We present average absolute accuracy and average normalized accuracy (in bracket) in %. Performance improvements are shown in **bold**.

F Experimental Details

In this Appendix, we provide the dataset and implementation details used to carry out the experiments presented in the paper.

F.1 Datasets

The 8-dataset benchmark consists of: Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (LeCun et al., 1998), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011).

The 14-dataset benchmark builds on the preceding one, incorporating six additional datasets: CIFAR100 (Krizhevsky et al., 2009), STL10 (Coates et al., 2011), Flowers102 (Nilsback & Zisserman, 2008), OxfordIIITPet (Parkhi, 2012), PCAM (Veeling et al., 2018), and FER2013 (Goodfellow et al., 2013).

Finally, the 20-dataset benchmark includes the preceding 14 plus the following six: EMNIST (Cohen et al., 2017), CIFAR10 (Krizhevsky et al., 2009), Food101 (Bossard et al., 2014), FashionMNIST (Xiao et al., 2017), RenderedSST2 (Radford et al., 2021), and KMNIST (Clanuwat et al., 2018).

- **SUN397** (Xiao et al., 2016) contains more than $100,000$ images of 397 categories for benchmarking scene understanding. The number of images varies across categories, but there are at least 100 images each.
- **Stanford Cars (Cars)** (Krause et al., 2013) has $16,185$ images in total of 196 types of cars and is evenly split for training and testing sets.

- **RESISC45** (Cheng et al., 2017) is developed for remote sensing image scene classification. This dataset covers 45 scene classes with 700 images of size 256×256 for each.
- **EuroSAT** (Helber et al., 2019) is used for land use and land cover classification using Sentinel-2 satellite images of size 64×64, consisting of $27,000$ images covering 10 classes.
- **SVHN** (Netzer et al., 2011) is a street view house number classification benchmark, containing more than $600,000$ RGB images of 10 printed digits in size 32×32 cropped from house number plates.
- **GTSRB** (Stallkamp et al., 2011) is a German traffic sign recognition benchmark consisting of over $50,000$ images of 43 classes of traffic signs in varying light and background conditions.
- **MNIST** (LeCun et al., 1998) is a well-known classical dataset for hand-written digit classification with $60,000$ training and $10,000$ testing images of size 28×28 in 10 classes of numbers.
- **DTD** (Cimpoi et al., 2014) is a collection of $5,640$ images across 47 categories of textures in the wild, annotated with human-centric attributes.
- **Flowers102** (Nilsback & Zisserman, 2008) contains 102 flower categories that are popular in the United Kingdom, with $1,020$ training and $6,149$ testing images. The images have varying poses and light conditions.
- **PCAM (PatchCamelyon)** (Veeling et al., 2018) consists of more than 300k color images in size of 96×96 pixels extracted from histopathologic scans of lymph node sections. Each of them is annotated with a binary class indicating the presence of metastatic tissue.
- **FER2013** (Goodfellow et al., 2013) is developed for facial expression recognition. The images are grayscale and have a size of 48×48 pixels, describing seven different kinds of emotions. The training and testing split consists of $28,709$ and $7,178$ samples, respectively.
- **OxfordIIITPet** (Parkhi, 2012) is a 37-category pet dataset with roughly 200 images for each category, and is equally divided for both training and testing splits. The images vary in scale, pose, and lighting conditions.
- **STL10** (Coates et al., 2011) is primarily built for unsupervised image recognition tasks covering 10 classes. Hence, the number of labeled images is quite small: 500 training and 800 testing images for each class. All of them are in 96×96 pixel resolution.
- **CIFAR100** (Krizhevsky et al., 2009) consists of color images categorized in 100 general classes, each class contains 600 images, and each image is in size 32×32. There are $50,000$ training images and $10,000$ testing images.
- **CIFAR10** (Krizhevsky et al., 2009) is similar to CIFAR100, except it has 10 classes.
- **Food101** (Bossard et al., 2014) contains 101 food categories, with $101,000$ images. For each class, 750 images are for training and 250 are for testing. Only the testing images are manually reviewed. The training images contain noise mostly from intense colors, and sometimes are mislabelled.
- **FashionMNIST** (Xiao et al., 2017) is designed as a drop-in replacement benchmark for the original MNIST, thereby inheriting the same structure as MNIST.
- **EMNIST** (Cohen et al., 2017) is an extended version of MNIST. EMNIST contains images of both characters and digits. We choose to use only the EMNIST Letters split, which contains around $145,000$ images evenly distributed in 26 classes of the alphabet letters.
- **KMNIST** (Clanuwat et al., 2018), yet another version of MNIST, represents 10 Japanese Hiragana characters.
- **RenderedSST2** (Radford et al., 2021) is used for evaluating the models' capability on optical character recognition. The images are rendered from sentences in the Stanford Sentiment Treebank v2 (Socher et al., 2013), with black texts on a white background in 448×448 resolution. Each image is labeled as positive or negative based on the mood expressed in the text, and the number of images for both classes is nearly balanced. There are $6,920$ training and $1,821$ testing images.

The GLUE benchmark consists of: CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan & Brockett, 2005), STS-B (Cer et al., 2017), QQP (Iyer et al., 2017), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Giampiccolo et al., 2007).

- **CoLA** (Warstadt et al., 2019) consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated as either grammatically acceptable or unacceptable. Following standard practice, we evaluate performance using the Matthews Correlation Coefficient (MCC), which accounts for class imbalance in this binary classification task.
- **SST-2** (Socher et al., 2013) contains sentences extracted from movie reviews along with human annotations of their sentiment polarity. We use the binary (positive/negative) sentence-level classification split and evaluate model performance using accuracy.
- **MRPC** (Dolan & Brockett, 2005) comprises sentence pairs automatically extracted from online news sources, with human annotations indicating whether the sentences in each pair are semantically equivalent. Due to the imbalanced class distribution, we report both accuracy and F1 score for this paraphrase detection task.
- **STS-B** (Cer et al., 2017) is a collection of sentence pairs sourced from news headlines, video and image captions, and natural language inference data. Each pair is assigned a human-annotated similarity score ranging from 1 to 5. As a regression task, performance is evaluated using the average of Pearson and Spearman correlation coefficients.
- **QQP** (Iyer et al., 2017) consists of question pairs collected from the community question-answering platform Quora. The objective is to determine whether two questions are semantically equivalent. Given the unbalanced label distribution, we evaluate this task using both accuracy and F1 score.
- **MNLI** (Williams et al., 2018) is a large-scale, crowd-sourced corpus of sentence pairs annotated for textual entailment across ten diverse genres, including transcribed speech, fiction, and government reports. The task requires predicting whether a premise entails, contradicts, or is neutral with respect to a hypothesis. We evaluate accuracy on both the matched (in-domain) and mismatched (cross-domain) test sets.
- **QNLI** (Rajpurkar et al., 2016) converts the reading comprehension task into a sentence-pair classification problem. Each example pairs a question with a sentence from the corresponding Wikipedia context, and the model must determine whether the sentence contains the answer to the question. Performance is measured by accuracy.
- **RTE** (Giampiccolo et al., 2007) aggregates data from the RTE1 through RTE5 challenges, comprising text pairs sourced from news articles and Wikipedia. All examples have been converted to a binary classification format where "neutral" and "contradiction" labels are collapsed into "not entailment." We evaluate this task using accuracy.

F.2 Additional Notes

Our method relies on SVD and RSVD, which is defined for two-dimensional matrices $\Delta \in \mathbb{R}^{m\times n}$. However, some weights of the neural networks are represented by vectors $\delta \in \mathbb{R}^n$, e.g. bias vectors and parameters of layer normalization (Ba et al., 2016). Therefore, these parameters bypass the filtering process of the `PACT` algorithm and directly enter the subsequent merging process.

F.3 Selection of the scaling factor α

In Figure 14, we present the relationship between validation accuracy and the scaling factor α on the 8-task ViT-B/16 model. TA is sensitive to the choice of α, whereas Iso-C and TSV-M are relatively robust, which is consistent with previous findings. Due to the cleaning and filtering effect of `PACT`, the optimal α values increase for all three methods when combined with `PACT`, which also aligns with theoretical predictions. Notably, `PACT-TA` becomes robust to the choice of the optimal α, though the value itself becomes substantially large. For reproducibility, in Table 18 we provide the optimal α value chosen on the held-out validation set for each model and number of tasks. It should be noted that, in the optimal α search procedures, the early stopping step is set to be 3.

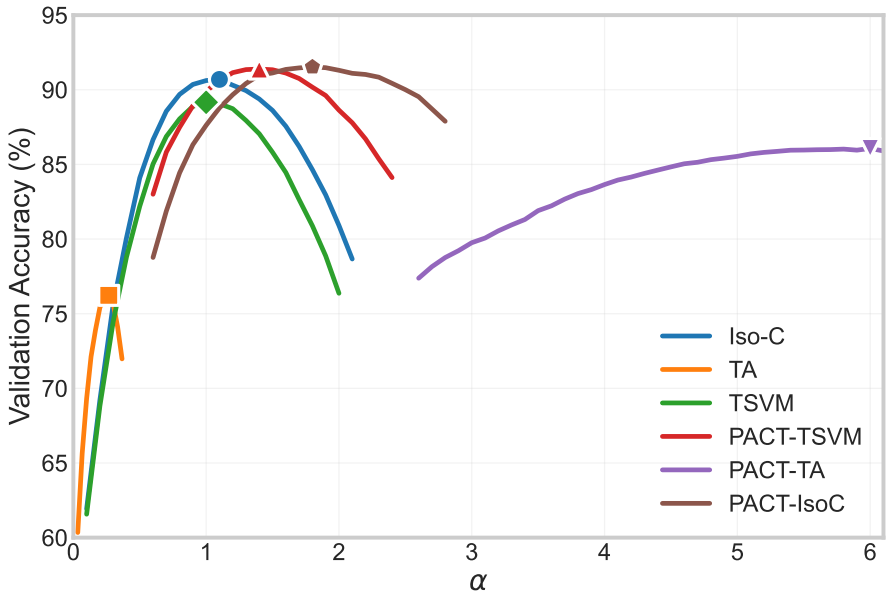


Figure 14: The α is chosen based on the best average performance on the validation set across tasks. Each point denotes the optimal α for each method. The model is ViT-B/16.

Furthermore, we report the optimal scaling factor α selected for the LoRA and NLP-related experiments, with the results presented in Table 19 to facilitate reproducibility. For the LoRA experiments, the scaling factor α starts at 1.0 and increases in increments of 0.5 until the accuracy plateaus. For the NLP experiments, the scaling factor α starts at 0.6 and increases in increments of 0.2 until the accuracy plateaus. The only exception is TA, which starts at 0.1 and increases in increments of 0.2. The early stopping patience is also set to 3.

method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
TA	0.27	0.10	0.70	0.27	0.13	0.10	0.30	0.17	0.10
PACT-TA	5.40	5.30	4.20	6.00	6.70	8.10	5.70	7.10	10.10
Iso-C	1.40	0.90	0.90	1.40	1.00	0.80	1.60	1.10	1.00
PACT-Iso-C	1.70	1.70	1.60	1.80	1.60	1.60	2.00	1.50	1.50
TSV-M	1.00	1.00	0.90	1.00	1.00	0.90	1.10	1.00	1.00
PACT-TSV-M	1.40	1.50	1.70	1.40	1.60	1.80	1.40	1.70	1.70

Table 18: Optimal α value chosen on a held-out validation set for different model types and numbers of tasks.

method	LoRA		NLP	
	ViT-B/32	ViT-L/14	RoBERTa	GPT-2
TA	0.10	0.10	0.30	0.50
PACT-TA	2.00	1.50	4.60	4.20
Iso-C	4.50	7.00	2.00	1.20
PACT-Iso-C	5.50	8.50	3.80	-
TSV-M	-	-	1.60	1.00
PACT-TSV-M	-	-	2.40	1.20

Table 19: Optimal α value chosen on LoRA and NLP experiments.

G ALGORITHM

Here we provide PACT's algorithm in Algorithm 1 for ease of understanding.

Algorithm 1 The Process of PACT

Input: Pre-trained model θ_0, task vectors $\{\Delta_t\}_{t=1}^{T}$
Output: Filtered task vectors $\{\tilde{\Delta}_t\}_{t=1}^{T}$
Compute the principal subspace $V_{0,K}$ of θ_0.
for $t = 1, \ldots, T$ **do**
    Compute the task subspace $V_{t,k}$.
    Compute the anchored core subspace $V_{t,LBW}$ using Eqn. 3.
end for
for $t = 1, \ldots, T$ **do**
    Construct the protected subspace V_{protect}^{t} using Eqn. 6.
    Filter the task vector using Eqn. 7.
end for

H CONCLUSION & LIMITATION

We challenge the assumption that pre-trained parameters are inert in task-vector merging. Fisher and SVD analyses reveal LBW dimensions: low-dimensional, high-energy pre-trained structures unchanged by fine-tuning yet essential for task inference. Disrupting only 1.17% of these structures catastrophically degrades performance. To protect LBW subspaces from cross-task collisions, we propose PACT, a data-free, modular orthogonal shield that removes interfering updates. Across 20 vision tasks, ViT-B/ViT-L backbones, and full/LoRA fine-tuning, PACT consistently improves task-vector merging. Its randomized SVD variant removes cubic dependence on task count, improving scalability. These results reframe merging as protecting pre-trained structures alongside resolving update conflicts. Future work will extend PACT to LLMs and many-task settings, and explore nonlinear projections and adaptive, importance-aware LBW dimensionalities.

Limitations. Although PACT approaches model merging from a new perspective, namely the identification and protection of LBW dimensions, and achieves substantial improvements over existing methods, several limitations remain. First, despite the proposed efficient variant, PACT still introduces additional computational overhead. Second, there may exist more efficient and accurate strategies for identifying LBW dimensions. We leave these directions for future work.