BEYOND THE MERGED VECTOR: FIXED-BUDGET DIRECTIONAL MEMORY FOR DATA-FREE CONTINUAL MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-free continual model merging aims to progressively integrate sequentially arriving task models into a unified model without accessing historical data or retaining previous task models. Existing methods typically construct historical subspaces from the cumulative merged task vector, but such implicit historical representations progressively degrade as tasks accumulate. To address this limitation, we propose **CUSP-Merge** (**CU**mulative **S**hared–**P**rivate Memory with Adaptive Protection for Continual Model Merging), a fixed-budget Shared–Private directional memory framework that replaces the implicit historical subspace with explicitly maintained historical task directions. Specifically, CUSP-Merge consolidates directions repeatedly supported across tasks into Shared Memory, retains task-specific residual directions in Private Memory, and dynamically promotes private directions to Shared Memory according to cumulative cross-task support, thereby enabling fine-grained organization of historical directions. Furthermore, this structured memory enables direction-level adaptive protection to balance stability and plasticity. Extensive experiments across multiple model architectures and task configurations demonstrate that CUSP-Merge achieves the highest average accuracy in all settings, with more pronounced gains on longer task sequences, while maintaining a favorable performance–storage trade-off.

1 INTRODUCTION

In recent years, pretrained models have been widely adapted to diverse downstream tasks, giving rise to a large number of task-specific expert models (Radford et al., 2021; Dosovitskiy, 2020). Model merging provides an effective way to reuse model capabilities by integrating multiple experts without accessing their training data (Ilharco et al., 2022; Yadav et al., 2023). However, in real-world deployment scenarios, task models are often not obtained all at once but emerge progressively as new tasks arise. Due to storage and privacy constraints (Kim et al., 2025; Kairouz & McMahan, 2021), historical task models and their data may not be preserved or accessed again. This motivates the study of continual model merging (Porrello et al., 2025; Dziadzio et al., 2025; Tang et al., 2025), which aims to continuously incorporate new task capabilities while retaining previously acquired knowledge as new tasks arrive. Under data-free and storage-constrained settings, balancing historical stability and new-task plasticity remains a core challenge in this field (Parisi et al., 2019).

Historical knowledge is therefore needed to determine which previously acquired capabilities should be protected when incorporating a new task. Existing data-free continual model merging methods typically use the parameter change of the current merged model relative to the pretrained model, namely the cumulative merged task vector, to represent such knowledge (Tang et al., 2025; Yang et al., 2026). They extract a low-rank historical subspace from this vector and constrain subsequent updates within it to mitigate forgetting. Recent studies further show that the dominant subspace spanned by the principal directions of a task vector, termed *task directions* in this paper, can characterize task-relevant representation directions, making task-vector-based historical representations an effective data-free alternative (Qiu et al., 2026c;b; Yang et al., 2026).

However, the above strategy implicitly relies on a key assumption that has not been sufficiently studied: *the dominant subspace of a single cumulative merged vector can faithfully preserve the*

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.

important directions of all preceding tasks. Our analysis shows that this assumption may gradually break down during continual merging. Further analysis reveals that, compared with an equal-rank reference subspace directly constructed from historical task directions, the historical subspace recovered from the cumulative merged vector incurs substantial additional information loss, and this loss continues to accumulate as the task sequence grows. This suggests that relying solely on the current merged model to reconstruct historical directions may not sufficiently capture reliable historical knowledge in continual merging.

This observation leads to a direct insight: instead of repeatedly recovering historical directions from the current merged model after fusion, historical directions should be explicitly extracted and maintained when each task arrives. However, storing all historical directions causes storage overhead to grow linearly with the number of tasks, while indiscriminately compressing them into a fixed-rank subspace may discard directions that are important for specific tasks. Therefore, under a limited storage budget, historical directions need to be effectively organized to preserve both stable cross-task shared structures and task-specific information that may be overlooked by global compression.

Based on this insight, we propose **CUSP-Merge** (**CU**mulative **S**hared–**P**rivate Memory with Adaptive Protection for Continual Model Merging), a fixed-budget Shared–Private directional memory framework for data-free continual model merging. CUSP-Merge replaces the historical subspace implicitly reconstructed from the cumulative merged vector with an explicit directional memory, and dynamically organizes historical directions into Shared Memory and Private Memory according to their cross-task support. Shared Memory consolidates shared directions consistently supported by multiple tasks, while Private Memory preserves task-specific directions that have not yet received sufficient cross-task support. Furthermore, CUSP-Merge leverages its memory structure to adaptively adjust the protection strength of different historical directions, thereby balancing historical knowledge preservation and new-task adaptation. Empirically, CUSP-Merge demonstrates consistent advantages across diverse settings, outperforming the strongest continual model merging baseline by 1.17 percentage points on average and proving particularly effective for long-sequence continual merging under a fixed memory budget.

Our main contributions are summarized as follows:

- We reveal a key limitation of constructing historical subspaces from the cumulative merged task vector: the merged parameter update may fail to fully preserve important directions of historical tasks, leading to degradation of historical representations as tasks accumulate.
- We propose CUSP-Merge, to the best of our knowledge, the first framework to explicitly maintain a fixed-budget historical directional memory for data-free continual model merging. It organizes historical directions through a Shared–Private structure and adopts direction-level adaptive protection to balance stability and plasticity.
- Extensive experiments across multiple CLIP backbones and continual merging sequences of varying lengths show that CUSP-Merge achieves the highest average accuracy across all settings and a favorable performance–storage trade-off on long task sequences.

2 Related Work

Model Merging. Model merging aims to integrate multiple expert models independently fine-tuned from the same pretrained model into a unified model without additional joint training. Early approaches mainly rely on parameter averaging or linear interpolation (Utans, 1996; Wortsman et al., 2022; Entezari et al., 2021; Ainsworth et al., 2022), while Task Arithmetic (Ilharco et al., 2022) further enables capability composition through algebraic operations on task vectors. Since independently fine-tuned updates often contain redundancy and conflicts, subsequent methods mitigate task interference through parameter selection, sparsification, or sign consistency (Yadav et al., 2023; Yu et al., 2024b; Wang et al., 2024; Marczak et al., 2024). More recently, another line of work has further exploited the low-rank spectral structure of task vectors and leveraged subspace modeling for more fine-grained conflict resolution (Gargiulo et al., 2025; Marczak et al., 2025; Cheng et al., 2025; Stoica et al., 2025). Beyond fully data-free approaches, some methods use test data or input-dependent routing for dynamic merging (Yang et al., 2024; Lu et al., 2024; Tang et al., 2024b), and therefore rely on additional test-time information or dynamic decision making during inference. Moreover, most existing model merging methods assume that all models to be merged are

simultaneously available, which limits their applicability to continual or streaming scenarios. Our work instead considers a continual setting in which task models arrive sequentially and are merged recursively.

Continual Learning. Continual learning studies how to learn a sequence of tasks while mitigating catastrophic forgetting (McCloskey & Cohen, 1989). Representative approaches include regularization-based constraints (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018), memory replay (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Buzzega et al., 2020), dynamic architecture expansion (Rusu et al., 2016; Zhou et al., 2024; Yu et al., 2024a), and gradient projection or subspace constraints designed to reduce interference between old and new tasks (Farajtabar et al., 2020; Saha et al., 2021; Liang & Li, 2024). Although these methods alleviate the stability–plasticity trade-off from different perspectives, they are all designed around a continual training process and typically rely on information such as historical samples, gradients, activations, or training states. This limits their applicability when task data or intermediate training states are unavailable.

Continual Model Merging. Continual model merging (Dziadzio et al., 2025; Tang et al., 2025) provides a complementary paradigm that progressively integrates independently fine-tuned task models without access to training data, relying instead on model parameters and their structural information. Unlike continual model merging methods that rely on test-time adaptation or dynamic expert routing (Qiu et al., 2026d;a), we focus on a strictly data-free setting that accesses neither training nor test data, while retaining the original static inference pipeline. Under this setting, existing methods primarily mitigate task interference during sequential merging through parameter-space constraints. Specifically, OPCM (Tang et al., 2025) projects incoming task updates onto the orthogonal complement of the historical subspace to reduce interference with previously acquired knowledge; DOP (Yang et al., 2026) introduces dual projections for historical knowledge and the current task, thereby mitigating forgetting while maintaining current-task performance; NUFILT (Qiu et al., 2026c) employs null-space filtering based on the singular subspaces of task vectors and incorporates low-rank compensation to enhance plasticity. However, these methods typically characterize historical knowledge indirectly through the accumulated merged state and its induced parameter-space structure. In contrast, we study an explicit representation of historical directions that is continuously maintained under a fixed memory budget.

3 Background and Motivation

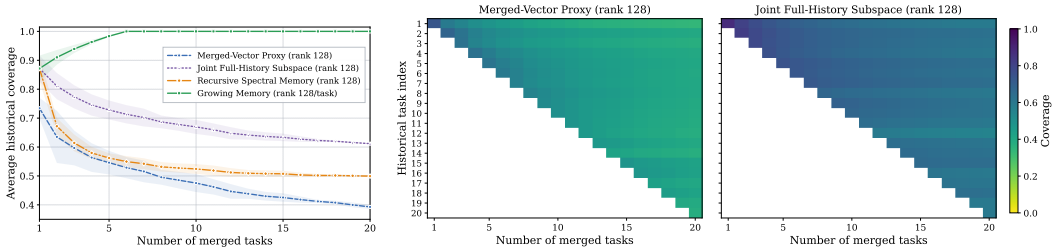
3.1 Problem Setting

We study data-free continual model merging. Given a pretrained base model θ_0, we consider a sequence of T task-specific models $\{\theta_t\}_{t=1}^{T}$, where each θ_t is independently fine-tuned from θ_0 on its corresponding task dataset $\mathcal{D}_t$. Unlike conventional model merging, which typically assumes an offline one-shot setting, continual model merging considers task models arriving sequentially. At stage t, the merger receives the current task model θ_t and updates the previously merged model $\theta_{t-1}^{\text{merged}}$ into a new merged model, without revisiting previous task-specific models or any task data. This process can be written as $\theta_t^{\text{merged}} = \text{ContinualMerge}\left(\theta_t, \theta_{t-1}^{\text{merged}}; \theta_0\right)$, with the initialization $\theta_1^{\text{merged}} = \theta_1$. Our goal is to obtain a unified model that can continuously integrate knowledge from sequential tasks while maintaining strong performance across all previously seen tasks.

To further characterize the parameter-space structure, we define the task vector of task t as $\tau_t = \theta_t - \theta_0$ (Ilharco et al., 2022). Accordingly, the continual model merging process can be further written as $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + \widetilde{\tau}_t$, where $\widetilde{\tau}_t = \mathcal{F}_t(\tau_t)$ denotes the effective update obtained by transforming the current task vector through the merging function $\mathcal{F}_t$.

3.2 Limitations of Merged-Vector-Based Historical Subspaces

In our data-free continual model merging setting, historical task models are inaccessible after each stage, while the merged model's displacement from the pretrained model provides a compact summary of accumulated historical updates. A natural strategy is therefore to represent historical knowledge using this parameter change, yielding the cumulative merged task vector (Tang et al., 2025):



(a) (b)

Figure 1: Historical direction coverage on the 20-task ViT-B/32 setting over ten random task orders. (a) Average historical coverage under different history representations. (b) Task-wise coverage heatmaps of the Merged-Vector Proxy and Joint Full-History Subspace.

$\widetilde{\tau}_{\leq t-1}^{(l)} = \theta_{t-1}^{\mathrm{merged},(l)} - \theta_0^{(l)}$. Recent studies have further revealed a close connection between task vectors and data representations (Cheng et al., 2025; Yang et al., 2026). In particular, NUFILT (Qiu et al., 2026c) shows both empirically and theoretically that the dominant right-singular subspace of a task vector is approximately aligned with the corresponding data-representation subspace, and uses it as a data-free proxy for task-relevant representation directions in constructing optimization objectives. However, since individual historical task vectors $\{\tau_i\}_{i<t}$ are no longer accessible during continual merging, the historical subspace must instead be reconstructed from the cumulative merged task vector: $H_{t-1}^{(l)} = \mathrm{TopV}_r\left(\widetilde{\tau}_{\leq t-1}^{(l)}\right)$. The limitation arises because the cumulative merged task vector essentially integrates multiple historical updates into a single parameter update: $\{\tau_1, \tau_2, \ldots, \tau_{t-1}\} \longrightarrow \widetilde{\tau}_{\leq t-1}$. During this many-to-one aggregation, some historical directions may be weakened or even lost due to cross-task interference. Consequently, even if the right-singular subspace of each task vector provides a good proxy for its task-relevant directions, the subspace reconstructed from the cumulative merged vector may still provide an incomplete representation of historical knowledge.

To quantitatively analyze the above phenomenon, we define *historical direction coverage*. Given the historical subspace $H_t^{(l)}$ at stage t, its directional coverage of task i is defined as

$$C_{t,i}^{(l)} = \frac{\left\| \tau_i^{(l)} H_t^{(l)} H_t^{(l)\top} \right\|_F^2}{\left\| \tau_i^{(l)} \right\|_F^2}, \qquad 1 \leq i \leq t \leq T. \tag{1}$$

This metric measures the fraction of task-vector energy of task i captured by the historical subspace. Averaging over all observed tasks and L analyzed layers gives the *average historical coverage*: $\mathrm{AHC}_t = \frac{1}{tL} \sum_{i=1}^{t} \sum_{l=1}^{L} C_{t,i}^{(l)}$.

We conduct the analysis on the 20-task ViT-B/32 setting (Tang et al., 2025) using ten random task orders and report the mean and standard deviation. As shown in Fig. 1a, we use the cumulative merged task vector of NUFILT (Qiu et al., 2026c) as the Merged-Vector Proxy and follow its configuration of using the top-128 right singular directions to construct the historical subspace. Its average historical coverage continuously decreases from 0.7342 at the first stage to 0.3929 at stage 20, indicating progressive degradation of historical-direction representation during continual merging.

To distinguish whether this degradation arises from the fixed-rank subspace constraint or from information loss in the cumulative merged vector itself, we construct a Joint Full-History Subspace. At each stage, it accesses all historical task vectors and jointly computes a rank-128 reference subspace. Its coverage reaches 0.6118 at stage 20, approximately 21.88 percentage points higher than the Merged-Vector Proxy. Since the two subspaces have the same rank, this gap shows that the fixed-rank constraint alone cannot explain the coverage degradation. Instead, using the cumulative merged vector as a historical proxy introduces additional information loss. Fig. 1b further shows that this loss is not confined to a few tasks, but is widespread across the historical task sequence.

As an approximate upper bound, Growing Memory retains 128 directions for each task without cross-task compression. Its final coverage approaches 1.0, showing that explicitly storing task di-

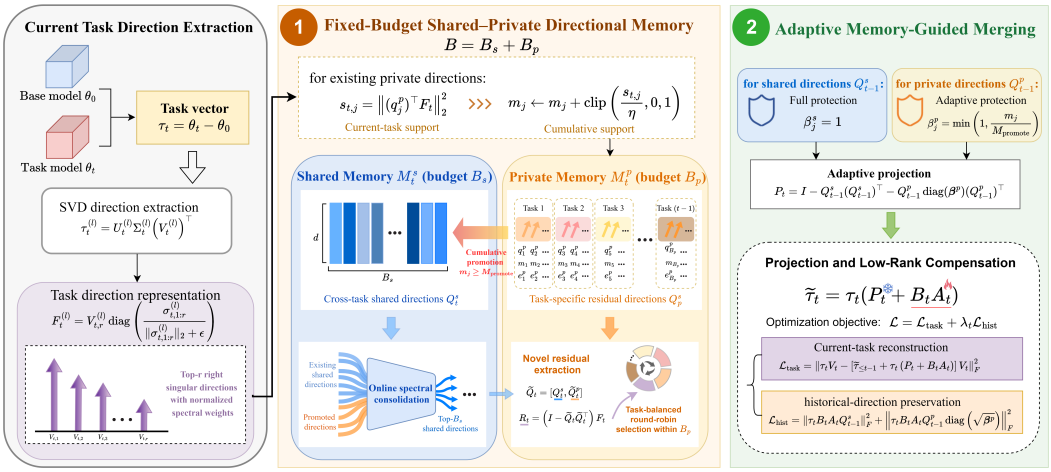


Figure 2: Overview of CUSP-Merge.

rections can preserve nearly all historical task-vector energy. However, the storage requirements of both reference branches grow linearly with the number of tasks. To further examine explicit directional memory under a fixed budget, we introduce Recursive Spectral Memory, which recursively merges the current task directions with the spectral representation retained from the previous stage and truncates the result back to rank 128 at each stage. Its coverage reaches 0.4994 at stage 20, 10.65 percentage points above the Merged-Vector Proxy, showing that even with the same subspace rank, explicitly maintaining historical directions is preferable to reconstructing them from the cumulative merged vector. Nevertheless, its final coverage remains 11.23 percentage points below the Joint Full-History Subspace and well below Growing Memory, indicating that indiscriminately compressing all historical directions into a single fixed-capacity spectral space still loses important structure. This motivates a fixed-budget memory that organizes historical directions into consolidatable cross-task shared directions and task-specific directions vulnerable to global compression.

4 Method

Building on the above observations, we propose CUSP-Merge, a data-free continual model merging method that explicitly maintains a fixed-budget Shared–Private directional memory and leverages its structure for direction-level adaptive protection. The overall framework is illustrated in Fig. 2.

4.1 Fixed-Budget Shared–Private Directional Memory

To replace the historical subspace implicitly estimated from the cumulative merged vector, we construct a fixed-budget Shared–Private directional memory that explicitly preserves historical task directions. The memory separately maintains cross-task shared directions and task-specific directions, and is continuously updated under a fixed capacity.

Directional Memory Construction. We extract the principal directions from the task vector τ_t, providing a compact representation of the current task update. For the l-th linear layer, singular value decomposition is applied to the task-vector matrix $\tau_t^{(l)}$, with its right singular vectors characterizing the principal input directions along which the layer-wise parameter update acts. We select the top-r right singular directions and weight them by the normalized singular values, obtaining the task direction representation of this layer: $F_t^{(l)} = V_{t,r}^{(l)} \operatorname{diag}\left(\frac{\sigma_{t,1:r}^{(l)}}{\|\sigma_{t,1:r}^{(l)}\|_2 + \epsilon}\right)$. This normalization reduces the effect of scale differences across task vectors while preserving the relative spectral strengths of the principal directions. For notational simplicity, we omit the layer index l hereafter.

Based on the above task direction representation, we maintain a fixed-budget Shared–Private directional memory: $\mathcal{M}_t = \{\mathcal{M}_t^s, \mathcal{M}_t^p\}$. The Shared Memory $\mathcal{M}_t^s$ stores shared directions repeatedly

supported by multiple tasks, while the Private Memory $\mathcal{M}_t^p$ preserves task-specific residual directions that have not been sufficiently reused. Each memory atom records a direction q_j and its spectral energy e_j; for private directions, we additionally maintain a support mass m_j for promotion and the source task index. The detailed procedure is described below. To prevent linear growth with the number of tasks, a fixed budget $B = B_s + B_p$ is split between the Shared and Private Memories.

Cumulative Memory Evolution. As tasks arrive sequentially, CUSP-Merge dynamically partitions the Shared–Private Memory through a cumulative promotion mechanism that identifies directions with stable cross-task support. For an existing private direction q_j^p, we define its current-task support as $s_{t,j} = \left\|(q_j^p)^\top F_t\right\|_2^2$, which measures the subspace overlap between the current task directions and the historical private direction. Its cumulative support m_j is then updated according to the current support: $m_j \leftarrow m_j + \text{clip}\left(\frac{s_{t,j}}{\eta}, 0, 1\right)$. Here, $m_j^{\text{init}} = 1$ is the initial support value, and the parameter η controls the normalization scale of the support evidence contributed by a single task. When $m_j \geq M_{\text{promote}}$, the direction q_j^p is promoted to the Shared Memory. This mechanism allows either strong support or weaker support accumulated across multiple tasks to drive promotion, while preventing accidental weak overlap from immediately triggering promotion.

To maintain a compact historical representation under a fixed memory budget, we further conduct online memory consolidation. For the Shared Memory, we combine the existing shared directions with the newly promoted directions and perform spectral consolidation via singular value decomposition, retaining the principal directions within the budget B_s to obtain a compressed representation of cross-task shared structures. The squared singular value of each retained shared direction q_j^s is recorded as its spectral energy e_j^s.

For the Private Memory, after promotion and Shared-Memory consolidation, we denote the remaining private directions as $\widetilde{Q}_t^p$ and define the intermediate historical direction matrix as $\widetilde{Q}_t = [Q_t^s, \widetilde{Q}_t^p]$. The novel residual of the current task is then given by $R_t = \left(I - \widetilde{Q}_t \widetilde{Q}_t^\top\right) F_t$. We perform singular value decomposition on R_t and add its principal residual directions to $\widetilde{Q}_t^p$ as new private candidates, recording their spectral energy e_j^p and source task index. When the Private budget B_p is exceeded, directions are grouped by source task, ranked by e_j^p within each group, and selected in a task-balanced round-robin manner to prevent a few tasks from dominating the memory at the expense of others. After budget selection, the final historical direction matrix is $Q_t = [Q_t^s, Q_t^p]$.

4.2 Adaptive Memory-Guided Merging

To turn the stored directional memory into effective constraints during continual merging, we leverage its Shared–Private structure to adaptively regulate the protection of historical directions, balancing historical knowledge preservation and current-task plasticity.

Asymmetric Direction Protection. When merging task t, CUSP-Merge uses the historical directional memory accumulated from the first $t-1$ tasks and assigns different protection strengths to historical directions according to their Shared–Private identities. Since shared directions have been repeatedly supported by multiple tasks and represent stable cross-task structures, they are always fully protected: $\beta_j^s = 1$. In contrast, private directions have not yet received sufficient cross-task support, and fully protecting them may overly restrict the current-task update and reduce plasticity. Therefore, their protection strengths are adaptively determined by the cumulative support m_j: $\beta_j^p = \min\left(1, \frac{m_j}{M_{\text{promote}}}\right)$. As a direction receives more support from subsequent tasks, m_j increases and its protection strength becomes progressively stronger. Once $m_j \geq M_{\text{promote}}$, the direction is promoted to the Shared Memory and becomes fully protected.

Projection and Low-Rank Compensation. Let Q_{t-1}^s and Q_{t-1}^p denote the shared and private direction matrices constructed from the first $t-1$ tasks, respectively. To protect historical knowledge while preserving directional flexibility, we construct the following projection operator using the protection strengths defined above:

$$P_t = I - Q_{t-1}^s (Q_{t-1}^s)^\top - Q_{t-1}^p \,\text{diag}(\boldsymbol{\beta}^p)(Q_{t-1}^p)^\top. \tag{2}$$

This projection fully filters the shared directions and partially filters the private directions according to their protection strengths, yielding the projected update $\tau_t P_t$. Since the projection may also remove useful components required by the current task, we further adopt the low-rank compensation formulation of (Qiu et al., 2026c) and learn a compensation term $B_t A_t$, yielding the effective update $\widetilde{\tau}_t = \tau_t(P_t + B_t A_t)$. The trainable low-rank factors A_t and B_t recover current-task capabilities lost during projection, while historical-direction constraints prevent reintroducing components that interfere with previous tasks.

Specifically, we optimize the following objective: $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_t \mathcal{L}_{\text{hist}}$. To preserve current-task capability, we perform singular value decomposition on τ_t and take its top-r_g right singular vectors to form V_t. The task reconstruction loss encourages the resulting merged update to remain consistent with the original task vector within the principal subspace of the current task:

$$\mathcal{L}_{\text{task}} = \left\| \tau_t V_t - \left[\widetilde{\tau}_{\leq t-1} + \tau_t \left(P_t + B_t A_t \right) \right] V_t \right\|_F^2 . \tag{3}$$

To prevent the low-rank compensation from bypassing the projection-based protection and reintroducing interference with historical tasks, we define the history preservation loss as

$$\mathcal{L}_{\text{hist}} = \left\| \tau_t B_t A_t Q_{t-1}^s \right\|_F^2 + \left\| \tau_t B_t A_t Q_{t-1}^p \operatorname{diag}\left(\sqrt{\boldsymbol{\beta}^p} \right) \right\|_F^2 . \tag{4}$$

Here, shared directions receive full-strength constraints, while private directions are adaptively constrained according to their protection strengths. Finally, the weight of the history preservation loss is scheduled according to the number of observed tasks: $\lambda_t = \sqrt{\frac{t-1}{t}}$. A detailed derivation of the optimization objective is provided in Appendix A.1.

5 Experiments

5.1 Experimental Setup

Models and Datasets. Following existing continual model merging setups (Tang et al., 2025), we adopt CLIP-ViT (Radford et al., 2021) as the visual backbone and evaluate ViT-B/32, ViT-B/16, and ViT-L/14 on task sequences with 8, 14, and 20 tasks. All experiments are based on publicly available task-specific fine-tuned checkpoints (Tang et al., 2024a) and are repeated over 10 random task orders (seeds 42–51), with averaged results reported. See Appendix A.2 for detailed settings.

Implementation Details. We perform low-rank fusion optimization on the attention and MLP layers of the CLIP vision encoder, using a unified set of hyperparameters across all model architectures and task sequences. CUSP-Merge adopts a fixed directional memory budget of $B = 320$ for each linear layer, with capacities of $B_s = 256$ and $B_p = 64$ for the Shared and Private Memories, respectively. The support threshold and cumulative promotion threshold are set to $\eta = 1/512$ and $M = 2$, respectively. Starting from the second task, we optimize the low-rank compensation for 100 iterations using Adam with a learning rate of 5×10^{-4}. More implementation details are included in Appendix A.3.

Metrics and Baselines. To evaluate our merged models, we adopt average accuracy (ACC) and backward transfer (BWT) as the primary metrics (Lin et al., 2022). Detailed formulations are provided in Appendix A.4. We compare CUSP-Merge with three categories of baselines: (1) non-merging paradigms, including pretrained models, independently fine-tuned experts, and continual fine-tuning; (2) continual extensions of conventional model merging methods, including SWA (Izmailov et al., 2018), Task Arithmetic (Ilharco et al., 2022), TIES-Merging (Yadav et al., 2023), and other representative approaches; (3) continual model merging methods, including OPCM (Tang et al., 2025), DOP (Yang et al., 2026), and NUFILT (Qiu et al., 2026c). The complete set of baselines is listed in Appendix A.5.

5.2 Main Results

Table 1 shows that CUSP-Merge achieves the highest ACC across all nine combinations of model architectures and task-sequence lengths. Compared with NUFILT, the strongest existing continual model merging baseline, CUSP-Merge improves ACC by an average of 1.17 percentage points

Table 1: Continual model merging results over ten task orders (mean ± standard deviation). Best and second-best results are shown in **bold** and underlined, respectively. "C." denotes "Continual".

	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
	Pre-Trained	48.1	56.9	55.6	55.4	62.0	59.8	64.9	69.1	65.6
	Individual	90.4	89.3	89.8	92.4	91.3	91.6	94.3	93.4	93.5
	C. Fine-Tuned	79.8	67.4	62.6	82.9	72.2	68.2	90.0	70.9	77.7
ACC (%) ↑	Average (SWA)	$66.3_{\pm 0.0}$	$65.4_{\pm 0.0}$	$61.1_{\pm 0.0}$	$72.3_{\pm 0.0}$	$69.7_{\pm 0.0}$	$64.8_{\pm 0.0}$	$80.0_{\pm 0.0}$	$77.5_{\pm 0.0}$	$71.1_{\pm 0.0}$
	C. Task Arithmetic	$67.5_{\pm 0.0}$	$66.5_{\pm 0.0}$	$60.0_{\pm 0.0}$	$77.1_{\pm 0.0}$	$70.9_{\pm 0.6}$	$64.2_{\pm 0.0}$	$82.1_{\pm 0.0}$	$77.9_{\pm 0.0}$	$70.3_{\pm 0.0}$
	C. TIES-Merging	$49.0_{\pm 10.2}$	$66.2_{\pm 0.6}$	$59.9_{\pm 0.7}$	$66.8_{\pm 3.7}$	$70.5_{\pm 0.8}$	$63.0_{\pm 1.6}$	$64.3_{\pm 7.0}$	$78.0_{\pm 0.6}$	$68.3_{\pm 0.9}$
	MagMax-IND	$70.7_{\pm 0.0}$	$67.0_{\pm 0.0}$	$61.2_{\pm 0.0}$	$76.7_{\pm 1.8}$	$67.0_{\pm 0.0}$	$62.5_{\pm 0.0}$	$83.4_{\pm 0.0}$	$71.2_{\pm 0.0}$	$71.2_{\pm 0.0}$
	C. LW AdaMerging	$53.4_{\pm 3.2}$	$59.8_{\pm 1.6}$	$59.7_{\pm 7.4}$	$59.9_{\pm 2.3}$	$64.3_{\pm 1.2}$	$61.5_{\pm 1.1}$	$68.8_{\pm 2.9}$	$73.1_{\pm 5.7}$	$66.9_{\pm 1.1}$
	C. LoRA-WEMoE	$68.8_{\pm 7.8}$	$63.8_{\pm 3.4}$	$49.6_{\pm 15.4}$	$72.6_{\pm 3.7}$	$67.9_{\pm 2.9}$	$55.0_{\pm 7.0}$	$75.6_{\pm 7.8}$	$74.0_{\pm 5.0}$	$56.9_{\pm 19.8}$
	C. WUDI-Merging	$74.7_{\pm 6.6}$	$67.0_{\pm 6.9}$	$63.7_{\pm 3.8}$	$81.0_{\pm 4.7}$	$75.0_{\pm 4.1}$	$69.6_{\pm 4.7}$	$87.5_{\pm 3.3}$	$84.2_{\pm 3.7}$	$78.1_{\pm 2.8}$
	C. Iso-C	$71.7_{\pm 1.2}$	$73.2_{\pm 1.8}$	$67.6_{\pm 0.8}$	$78.5_{\pm 1.2}$	$79.7_{\pm 1.3}$	$73.0_{\pm 1.1}$	$86.9_{\pm 0.5}$	$86.9_{\pm 1.8}$	$80.9_{\pm 0.8}$
	C. KnOTS-TIES	$54.4_{\pm 6.9}$	$67.8_{\pm 0.4}$	$60.5_{\pm 1.4}$	$57.9_{\pm 8.4}$	$71.6_{\pm 0.3}$	$62.7_{\pm 1.1}$	$68.3_{\pm 5.7}$	$78.8_{\pm 0.3}$	$69.7_{\pm 0.8}$
	C. TSV-M	$68.2_{\pm 4.8}$	$63.3_{\pm 4.8}$	$58.8_{\pm 3.3}$	$75.4_{\pm 4.0}$	$69.2_{\pm 3.1}$	$63.1_{\pm 1.3}$	$82.2_{\pm 3.6}$	$78.1_{\pm 3.6}$	$70.5_{\pm 1.2}$
	OPCM	$75.5_{\pm 0.5}$	$71.9_{\pm 0.3}$	$65.7_{\pm 0.2}$	$81.8_{\pm 0.3}$	$77.1_{\pm 0.5}$	$70.3_{\pm 0.2}$	$87.0_{\pm 0.4}$	$83.5_{\pm 0.2}$	$76.0_{\pm 0.2}$
	DOP	$78.3_{\pm 1.6}$	$73.0_{\pm 1.2}$	$68.5_{\pm 1.6}$	$84.5_{\pm 0.8}$	$76.2_{\pm 1.2}$	$72.2_{\pm 1.8}$	$88.3_{\pm 0.7}$	$85.5_{\pm 0.8}$	$80.7_{\pm 0.9}$
	NUFILT	$83.6_{\pm 0.2}$	$78.0_{\pm 0.2}$	$71.0_{\pm 0.9}$	$87.3_{\pm 0.1}$	$83.1_{\pm 0.3}$	$78.1_{\pm 0.9}$	$91.6_{\pm 0.1}$	$89.2_{\pm 0.1}$	$84.7_{\pm 0.8}$
	CUSP (Ours)	$\mathbf{84.1}_{\pm 0.8}$	$\mathbf{78.2}_{\pm 0.6}$	$\mathbf{72.9}_{\pm 1.1}$	$\mathbf{89.0}_{\pm 0.3}$	$\mathbf{83.7}_{\pm 0.5}$	$\mathbf{80.7}_{\pm 1.1}$	$\mathbf{92.0}_{\pm 0.3}$	$\mathbf{89.7}_{\pm 0.2}$	$\mathbf{86.8}_{\pm 0.6}$
BWT (%) ↑	Average (SWA)	$-11.5_{\pm 2.2}$	$-8.0_{\pm 1.3}$	$-7.1_{\pm 2.1}$	$-9.7_{\pm 1.5}$	$-7.1_{\pm 1.4}$	$-7.3_{\pm 1.7}$	$-7.3_{\pm 1.4}$	$-5.8_{\pm 1.0}$	$-6.4_{\pm 1.5}$
	C. Task Arithmetic	$-9.6_{\pm 1.5}$	$\underline{-1.3}_{\pm 1.6}$	$-3.4_{\pm 1.0}$	$-4.2_{\pm 1.0}$	$-1.3_{\pm 0.4}$	$-3.6_{\pm 0.4}$	$-7.1_{\pm 0.8}$	$-1.8_{\pm 0.3}$	$-3.3_{\pm 0.3}$
	C. TIES-Merging	$-15.3_{\pm 8.0}$	$\mathbf{1.9}_{\pm 0.6}$	$\underline{-1.5}_{\pm 0.7}$	$-5.5_{\pm 0.4}$	$\mathbf{1.4}_{\pm 0.7}$	$\mathbf{-1.5}_{\pm 1.2}$	$-13.0_{\pm 5.7}$	$\underline{-1.1}_{\pm 0.4}$	$\underline{-2.9}_{\pm 1.0}$
	MagMax-IND	$-8.3_{\pm 1.3}$	$-7.4_{\pm 1.4}$	$-7.2_{\pm 1.6}$	$-6.1_{\pm 1.3}$	$-7.4_{\pm 2.0}$	$-8.0_{\pm 2.2}$	$-5.0_{\pm 0.8}$	$-6.0_{\pm 2.1}$	$-6.5_{\pm 2.1}$
	C. LW AdaMerging	$-32.5_{\pm 3.6}$	$-24.1_{\pm 1.7}$	$-22.7_{\pm 4.3}$	$-27.8_{\pm 2.7}$	$-22.1_{\pm 1.4}$	$-21.4_{\pm 1.2}$	$-24.3_{\pm 3.3}$	$-19.6_{\pm 1.7}$	$-21.7_{\pm 1.1}$
	C. LoRA-WEMoE	$-20.4_{\pm 9.0}$	$-20.2_{\pm 3.9}$	$-24.5_{\pm 10.0}$	$-18.0_{\pm 6.2}$	$-18.8_{\pm 3.4}$	$-25.8_{\pm 7.9}$	$-17.8_{\pm 5.9}$	$-16.8_{\pm 5.3}$	$-27.9_{\pm 17.2}$
	C. WUDI-Merging	$-17.0_{\pm 7.5}$	$-22.8_{\pm 7.3}$	$-26.0_{\pm 4.1}$	$-12.6_{\pm 5.4}$	$-16.9_{\pm 4.4}$	$-18.5_{\pm 14.2}$	$-7.3_{\pm 3.7}$	$-9.4_{\pm 4.0}$	$-15.8_{\pm 2.9}$
	C. Iso-C	$-10.2_{\pm 1.2}$	$-10.4_{\pm 1.9}$	$-10.3_{\pm 1.4}$	$-6.7_{\pm 0.5}$	$-7.1_{\pm 1.1}$	$-9.9_{\pm 1.7}$	$-3.7_{\pm 0.6}$	$-3.7_{\pm 1.7}$	$-5.5_{\pm 1.3}$
	C. KnOTS-TIES	$-12.6_{\pm 3.9}$	$\mathbf{1.9}_{\pm 0.5}$	$\mathbf{-1.3}_{\pm 0.7}$	$-13.5_{\pm 5.5}$	$\underline{1.0}_{\pm 0.1}$	$\underline{-2.3}_{\pm 0.6}$	$-11.6_{\pm 3.6}$	$\mathbf{0.3}_{\pm 0.3}$	$\mathbf{-2.3}_{\pm 0.7}$
	C. TSV-M	$-24.0_{\pm 5.6}$	$-26.7_{\pm 4.9}$	$-31.7_{\pm 3.4}$	$-18.5_{\pm 4.6}$	$-22.7_{\pm 3.1}$	$-28.2_{\pm 1.8}$	$-13.0_{\pm 4.0}$	$-15.6_{\pm 3.9}$	$-23.3_{\pm 1.3}$
	OPCM	$-6.3_{\pm 1.1}$	$-6.0_{\pm 1.0}$	$-7.8_{\pm 1.5}$	$-4.8_{\pm 0.7}$	$-5.1_{\pm 1.4}$	$-6.3_{\pm 2.2}$	$-2.6_{\pm 1.0}$	$-4.3_{\pm 0.7}$	$-6.5_{\pm 1.8}$
	DOP	$-6.1_{\pm 2.3}$	$-8.8_{\pm 2.1}$	$-12.5_{\pm 2.8}$	$-3.9_{\pm 1.0}$	$-7.5_{\pm 2.2}$	$-11.4_{\pm 2.8}$	$-3.2_{\pm 1.2}$	$-4.6_{\pm 1.0}$	$-8.4_{\pm 1.4}$
	NUFILT	$\mathbf{-2.7}_{\pm 0.7}$	$-5.7_{\pm 0.9}$	$-8.9_{\pm 2.3}$	$\mathbf{-1.6}_{\pm 0.5}$	$-3.5_{\pm 0.6}$	$-7.1_{\pm 1.9}$	$\mathbf{-1.1}_{\pm 0.3}$	$-2.0_{\pm 0.3}$	$-4.6_{\pm 0.7}$
	CUSP (Ours)	$\underline{-4.9}_{\pm 1.2}$	$-7.6_{\pm 0.9}$	$-12.0_{\pm 1.8}$	$\underline{-2.6}_{\pm 0.6}$	$-5.4_{\pm 0.7}$	$-8.1_{\pm 1.6}$	$\underline{-1.7}_{\pm 0.6}$	$-2.5_{\pm 0.3}$	$-5.2_{\pm 0.7}$

across all settings. Its advantage becomes more pronounced as the task sequence grows: under the 20-task setting, CUSP-Merge outperforms NUFILT by 1.9, 2.6, and 2.1 percentage points on ViT-B/32, ViT-B/16, and ViT-L/14, respectively. These consistent improvements demonstrate that CUSP-Merge better preserves previously acquired capabilities over long merging sequences and scales effectively across vision backbones of different sizes.

Furthermore, BWT measures the performance change relative to when each task is first incorporated and is directly comparable only among methods with similar overall accuracy. Although CUSP-Merge achieves lower BWT than NUFILT, stage-wise performance decomposition shows that this difference mainly arises from its higher initial task accuracy. At the end of the sequence, CUSP-Merge still achieves higher absolute accuracy on historical tasks overall. Therefore, a lower BWT does not imply that CUSP-Merge retains less historical knowledge. Detailed analysis is provided in Appendix B.1.

5.3 Ablation and Analysis Results

Ablation on Core Components. As shown in Table 2, the full method achieves the highest ACC and BWT. Replacing cumulative promotion with one-shot promotion reduces ACC by 1.97 percentage points, demonstrating that cross-task evidence helps reliably identify shared directions. Fixing private protection to either $\beta^p = 0.5$ or 1.0 causes an ACC drop of approximately 2 points, indicating that adaptive protection provides a better stability–plasticity balance than uniform soft or hard protection. Although the recursively compressed Single Memory achieves an average ACC close to CUSP-Merge, this aggregate result masks substantial task-level differences. CUSP-Merge improves the final accuracy over the first ten arrival positions by 2.37 points on average and effectively mitigates the degradation of forgetting-prone tasks. Further analysis is provided in Appendix B.2.

Table 2: Core component ablation on ViT-B/32 with 20 tasks.

Variant	S/P	Cumulative	Adaptive	ACC (%) ↑	BWT (%) ↑
Single Memory	–	–	–	$72.28_{\pm 1.85}$	$-14.62_{\pm 2.49}$
w/o Cumulative Promotion	✓	–	–	$70.96_{\pm 2.15}$	$-15.36_{\pm 2.78}$
w/o Adaptive ($\beta^p = 0.5$)	✓	✓	–	$70.92_{\pm 2.27}$	$-15.46_{\pm 2.86}$
w/o Adaptive ($\beta^p = 1.0$)	✓	✓	–	$70.80_{\pm 2.32}$	$-15.55_{\pm 2.92}$
Full CUSP	✓	✓	✓	$\mathbf{72.93}_{\pm 1.14}$	$\mathbf{-12.04}_{\pm 1.82}$

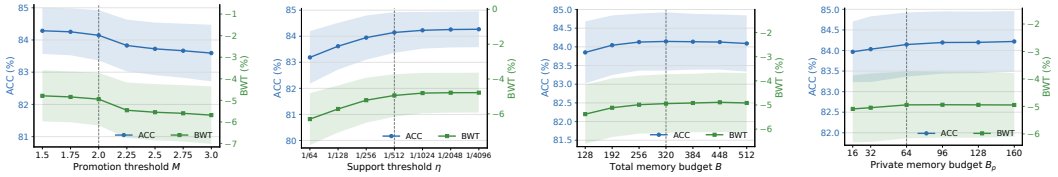


Figure 3: Hyper-parameter sensitivity of CUSP-Merge on ViT-B/32 with 8 tasks. From left to right: M, η, B, and B_p. Shading denotes standard deviation, and dashed lines indicate default values.

Hyper-parameter Sensitivity. We analyze the sensitivity to M, η, B, and B_p on ViT-B/32 with 8 tasks over ten random task orders in Fig. 3. A larger M delays the promotion of private directions and degrades performance. $M = 2$ is only 0.14 percentage points below the best result while avoiding overly aggressive promotion. Decreasing η improves both ACC and BWT, but the gains saturate beyond $1/512$, where weaker overlaps may be treated as valid support. Performance also plateaus around $B = 320$, with little benefit from a larger budget. With $B = 320$ fixed, varying B_p results in only minor changes, indicating robustness to the Shared–Private capacity allocation. The same trends hold for settings with 14 and 20 tasks (Appendix B.3). Balancing performance, promotion reliability, and storage efficiency, we set $M = 2$, $\eta = 1/512$, $B = 320$, and $B_p = 64$.

Fixed-Budget Performance vs. Growing Memory. On the ViT-B/32 20-task setting, we compare CUSP-Merge with a fixed memory budget of $B = 320$ against Growing Memory using different per-task ranks, reporting ACC and additional storage relative to the backbone model in Table 3. Under nearly the same storage, CUSP-Merge outperforms Growing Memory with $r = 16$ by 0.97 percentage points. Moreover, it achieves performance close to the best Growing Memory result while using only about one quarter of the additional storage of the $r = 64$ setting, demonstrating a favorable performance–storage trade-off. Full performance–storage results are in Appendix B.4. Coverage analyses further confirm efficient budget utilization and the complementary role of Private Memory (Appendix B.5).

Table 3: Performance–memory trade-off.

Method	ACC (%) ↑	Extra Mem. ↓
Growing ($r = 8$)	$71.56_{\pm 1.39}$	15.2%
Growing ($r = 16$)	$71.96_{\pm 1.35}$	30.3%
Growing ($r = 32$)	$72.50_{\pm 1.23}$	60.7%
Growing ($r = 64$)	$73.81_{\pm 1.14}$	121.4%
Growing ($r = 128$)	$73.50_{\pm 1.45}$	242.8%
CUSP-Merge	$72.93_{\pm 1.14}$	30.4%

6 Conclusion

In this work, we study data-free continual model merging and identify a key limitation: historical subspaces derived from the cumulative merged task vector progressively lose important historical directions as new tasks arrive. To address this issue, we propose CUSP-Merge, which explicitly maintains historical knowledge using a fixed-budget Shared–Private directional memory and applies direction-level adaptive protection based on cross-task support, thereby balancing stability and plasticity. Experiments show that CUSP-Merge consistently outperforms existing methods in average accuracy across different model scales and task-sequence lengths, with more pronounced gains on longer sequences. It also achieves a favorable performance–storage trade-off, providing an effective solution for scalable continual model merging.

AI USE STATEMENT

In this work, generative AI tools were used only to assist in drafting and polishing parts of the manuscript. Specifically, they were used to organize the presentation, improve clarity and coherence, reduce redundancy, and check selected LaTeX formatting, table descriptions, and figure captions. Literature review, research conception and execution, method implementation, mathematical reasoning, experimental data processing, and result analysis were conducted by the authors. This work did not involve synthetic dataset construction, questionnaires or interviews, material transcription, or qualitative thematic analysis.

All AI-assisted text was thoroughly reviewed and revised by the authors. The research methods, mathematical formulations, and technical conclusions were developed and determined by the authors. All experimental results were obtained from experiments conducted by the authors, and citation information was manually verified against the relevant literature. No unverified generated content was directly adopted. The authors take full responsibility for the final content of this work, including any text or other material drafted or polished with the assistance of generative AI.

ETHICS STATEMENT

This work studies data-free continual model merging using publicly available datasets and model checkpoints. It does not involve human subjects, collect personal or sensitive information, or release new datasets, and therefore does not require institutional review board approval. All datasets and pretrained models are used solely for research purposes in accordance with their applicable terms of use.

REPRODUCIBILITY STATEMENT

To support reproducibility, Section 4 describes the directional memory update mechanism and merging objective of CUSP-Merge, while Section 5.1 specifies the models, task settings, and main hyperparameters. The appendix further presents the derivation of the optimization objective (Appendix A.1), dataset and task configurations (Appendix A.2), complete implementation details (Appendix A.3), and evaluation metrics (Appendix A.4). All experiments use specified random seeds and report statistics over multiple task orders. The source code will be publicly released upon publication to facilitate reproducibility.

REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European conference on computer vision*, pp. 144–161. Springer, 2018.

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.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

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.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. *arXiv preprint arXiv:2503.08099*, 2025.

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.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sebastian Dziadzio, Vishaal Udandarao, Karsten Roth, Ameya Prabhu, Zeynep Akata, Samuel Albanie, and Matthias Bethge. How to merge your multimodal models over time? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20479–20491. IEEE, 2025.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

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.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, 2025.

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.

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.

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*, 2022.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

Youngeun Kim, Seunghwan Lee, Aecheon Jung, Bogon Ryu, and Sungeun Hong. Task vector quantization for memory-efficient model merging. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20105–20115. IEEE, 2025.

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.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647. IEEE, 2024.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177, 2022.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pp. 379–395. Springer, 2024.

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.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

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.

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.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Angelo Porrello, Lorenzo Bonicelli, Pietro Buzzega, Monica Millunzi, Simone Calderara, and Rita Cucchiara. A second-order perspective on model compositionality and incremental learning. In *International Conference on Learning Representations*, volume 2025, pp. 75098–75128, 2025.

Haiyun Qiu, Xingyu Wu, and Kay Chen Tan. Towards adaptive continual model merging via manifold-aware expert evolution. *arXiv preprint arXiv:2604.22464*, 2026a.

Zihuan Qiu, Zhiyang Liao, Chiyuan He, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Safe-merge: Data-free continual model merging with general knowledge preservation. *arXiv preprint arXiv:2608.01184*, 2026b.

Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving stability, promoting plasticity. In *International Conference on Learning Representations*, volume 2026, pp. 128492–128518, 2026c.

Zihuan Qiu, Yi Xu, Chiyuan He, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Mingle: Mixture of null-space gated low-rank experts for test-time continual model merging. *Advances in Neural Information Processing Systems*, 38:143841–143878, 2026d.

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.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

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. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

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.

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

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. *arXiv preprint arXiv:2402.00433*, 2024b.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Merging models on the fly without retraining: A sequential approach to scalable continual model merging. *arXiv preprint arXiv:2501.09522*, 2025.

Joachim Utans. Weight averaging for neural networks and local resampling schemes. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models. AAAI Press*, pp. 133–138. Citeseer, 1996.

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.

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.

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, 2022.

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, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

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, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. *Advances in Neural Information Processing Systems*, 38:39275–39305, 2026.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23219–23230. IEEE, 2024a.

Linyan 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 *Icml*, volume 2, pp. 21, 2024b.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23554–23564. IEEE, 2024.

A Additional Descriptions

A.1 Derivation of the Projection-Aware Objective

Data-Free Surrogate Objectives. NUFILT (Qiu et al., 2026c) shows that the representation subspace of task data approximately aligns with the principal right-singular subspace of the corresponding task vector. Let ζ denote their subspace misalignment. When ζ is small, the data-dependent stability and plasticity objectives are governed by the following parameter-space quantities:

$$\begin{aligned} \mathcal{L}_{\text{stab}}^{\text{proxy}} &= \left\| (\widetilde{\tau}_{\leq t} - \widetilde{\tau}_{\leq t-1})\, \widehat{V}_{\leq t-1} \right\|_F^2, \\ \mathcal{L}_{\text{plas}}^{\text{proxy}} &= \left\| (\widetilde{\tau}_{\leq t} - \tau_t)\, \widehat{V}_t \right\|_F^2. \end{aligned} \tag{5}$$

Here, $\widehat{V}_{\leq t-1}$ and $\widehat{V}_t$ denote the principal right-singular subspaces of the historical and current tasks, respectively. The former limits disturbance to the historical task subspace, whereas the latter encourages the merged result to approximate the current task vector within its principal subspace. They therefore serve as data-free surrogates for stability and plasticity.

CUSP-Merge adopts the above plasticity surrogate. Let V_t contain the top r_g right singular vectors of τ_t. Using the cumulative update $\widetilde{\tau}_{\leq t} = \widetilde{\tau}_{\leq t-1} + \tau_t(P_t + B_t A_t)$, the plasticity surrogate in Eq. 5 directly yields the current-task reconstruction loss in Eq. 3.

Constraints on Shared Directions. According to the projection operator in Eq. 2, shared directions receive full protection and therefore satisfy $P_t Q_{t-1}^s = 0$. The incremental update along these directions can thus be expanded as

$$\begin{aligned}(\widetilde{\tau}_{\leq t} - \widetilde{\tau}_{\leq t-1})\, Q_{t-1}^s &= \tau_t (P_t + B_t A_t) Q_{t-1}^s \\ &= \tau_t B_t A_t Q_{t-1}^s.\end{aligned} \tag{6}$$

Accordingly, the stability surrogate along shared directions reduces to

$$\mathcal{L}_{\text{shared}} = \left\| \tau_t B_t A_t Q_{t-1}^s \right\|_F^2. \tag{7}$$

This constraint prevents the low-rank compensation from reintroducing historical components already removed by the projection operator.

Constraints on Private Directions. Private directions receive partial protection determined by their cumulative support. Let $D_\beta = \operatorname{diag}(\boldsymbol{\beta}^p)$. Since the memory update maintains mutually orthogonal shared and private directions, Eq. 2 gives $P_t Q_{t-1}^p = Q_{t-1}^p (I - D_\beta)$. The complete incremental update along private directions is therefore

$$\begin{aligned}(\widetilde{\tau}_{\leq t} - \widetilde{\tau}_{\leq t-1})\, Q_{t-1}^p &= \tau_t (P_t + B_t A_t) Q_{t-1}^p \\ &= \tau_t Q_{t-1}^p (I - D_\beta) + \tau_t B_t A_t Q_{t-1}^p.\end{aligned} \tag{8}$$

The first term in Eq. 8 is the current-task component intentionally retained by the soft projection to preserve plasticity, whereas the second is introduced by the low-rank compensation. Directly applying the complete stability surrogate would minimize

$$\left\| \tau_t Q_{t-1}^p (I - D_\beta) + \tau_t B_t A_t Q_{t-1}^p \right\|_F^2. \tag{9}$$

This would also penalize the update intentionally retained by the soft projection and could drive the low-rank compensation to cancel that component, thereby weakening adaptive projection.

We therefore use P_t to determine the proportion of the current-task update allowed along each private direction, while constraining only the low-rank compensation in the history preservation loss:

$$\mathcal{L}_{\text{private}} = \left\| \tau_t B_t A_t Q_{t-1}^p \operatorname{diag}\left(\sqrt{\boldsymbol{\beta}^p}\right) \right\|_F^2. \tag{10}$$

The square root ensures that the loss weight of the j-th private direction is β_j^p after expanding the Frobenius norm. Directions with stronger cumulative support are therefore constrained more strongly, whereas those with weaker support retain greater flexibility for current-task adaptation.

Combining the Shared and Private constraints yields the history preservation loss in Eq. 4. It is optimized jointly with the current-task loss using the scheduled weight $\lambda_t = \sqrt{\frac{t-1}{t}}$.

A.2 Details of Datasets and Configurations

Models and Datasets. Following existing data-free continual model merging setups (Tang et al., 2025), we conduct experiments with three CLIP-ViT backbones (Radford et al., 2021), including ViT-B/32, ViT-B/16, and ViT-L/14. We directly use publicly available task-specific expert models (Tang et al., 2024a), where each expert is independently fine-tuned from the corresponding pre-trained CLIP model on a single downstream task. Therefore, during continual merging, only the currently arriving task model is required, without retraining task models or accessing their original training data.

Our vision task pool consists of 20 image classification datasets covering diverse domains, including scene recognition, fine-grained object recognition, remote sensing, digit and traffic-sign recognition, texture classification, medical imaging, facial expression recognition, and rendered-text classification. Specifically, the task pool includes SUN397 (Xiao et al., 2010), Cars (Krause et al., 2013), RESISC45 (Cheng et al., 2017), EuroSAT (Helber et al., 2019), SVHN (Netzer et al., 2011), GTSRB (Stallkamp et al., 2012), MNIST (LeCun et al., 1998), DTD (Cimpoi et al., 2014), Flowers102 (Nilsback & Zisserman, 2008), PCAM (Veeling et al., 2018), FER2013 (Goodfellow et al., 2013), OxfordIIITPet (Parkhi et al., 2012), STL10 (Coates et al., 2011), CIFAR100 and CIFAR10 (Krizhevsky

et al., 2009), Food101 (Bossard et al., 2014), FashionMNIST (Xiao et al., 2017), EMNIST (Cohen et al., 2017), KMNIST (Clanuwat et al., 2018), and RenderedSST2 (Socher et al., 2013; Zhai et al., 2019). These datasets vary substantially in visual distributions, numbers of classes, and recognition granularity, providing a diverse testbed for evaluating continual model merging across heterogeneous task sequences.

Task Configurations. To evaluate performance under different task-sequence scales, we construct three continual merging settings with 8, 14, and 20 tasks from the above task pool. The task sets are progressively expanded as follows:

- **8-task setting:** SUN397, Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD.
- **14-task setting:** the above 8 tasks together with Flowers102, PCAM, FER2013, OxfordIIITPet, STL10, and CIFAR100.
- **20-task setting:** the above 14 tasks together with CIFAR10, Food101, FashionMNIST, EMNIST, KMNIST, and RenderedSST2.

For each task-set size, we randomly permute the corresponding tasks to construct different continual merging orders. Specifically, we use 10 random task orders generated with seeds 42–51 for each setting. Different seeds produce different task permutations, while for a given seed, all compared methods use the same task set and task order to ensure a fair comparison. Unless otherwise specified, results are averaged over these 10 task orders, and standard deviations are additionally reported when analyzing variation across runs.

A.3 Additional Implementation Details

CUSP-Merge is applied to the linear layers in the attention and MLP modules, namely `attn`, `fc1`, `fc2`, and `mlp`. For each layer, we extract the top 128 right singular directions from each incoming task vector. After removing the components represented by the existing memory, we select the minimum residual rank that preserves 95% of the residual spectral energy as private candidates. The total directional memory budget is fixed at $B = 320$, with capacities of $B_s = 256$ and $B_p = 64$ for Shared Memory and Private Memory, respectively. The support threshold and cumulative promotion threshold are set to $\eta = 1/512$ and $M = 2$, respectively. Shared Memory is updated by retaining the principal spectral directions, while Private Memory uses task-balanced direction selection under its fixed capacity.

For fusion optimization, the low-rank compensation rank is set to 256, and the principal subspace rank for current-task reconstruction is set to $r_g = 64$. At stage t, the effective weight of the history preservation loss is scheduled as $\lambda_t = \sqrt{\frac{t-1}{t}}$. Starting from the second task, we optimize the low-rank compensation for 100 iterations using Adam with a learning rate of 5×10^{-4}. The same configuration is used across all model architectures, task-sequence lengths, and task orders.

A.4 Evaluation Metrics

We use average accuracy (ACC) and backward transfer (BWT) to assess the performance of the merged models (Lin et al., 2022). ACC measures the average performance of the final merged model across all tasks:

$$\mathrm{ACC} = \frac{1}{T}\sum_{i=1}^{T} A_i\left(\theta_T^{\mathrm{merged}}\right), \tag{11}$$

where $A_i(\cdot)$ denotes the accuracy on task i. A higher ACC indicates better overall merging performance. BWT measures forgetting of previously learned tasks:

$$\mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left[A_i\left(\theta_T^{\mathrm{merged}}\right) - A_i\left(\theta_i^{\mathrm{merged}}\right)\right]. \tag{12}$$

Under comparable overall accuracy, a higher BWT indicates less forgetting of historical knowledge.

Table 4: Stage-wise decomposition of BWT on ViT-B/32 over ten task orders (mean ± standard deviation). $\mathrm{LAcc}_{\mathrm{old}}$ and $\mathrm{FAcc}_{\mathrm{old}}$ denote historical-task accuracy at task incorporation and after the complete sequence, respectively. Higher values are better.

Tasks	Method	$\mathrm{LAcc}_{\mathrm{old}} \uparrow$	$\mathrm{FAcc}_{\mathrm{old}} \uparrow$	ACC ↑	BWT ↑
8	NUFILT	$86.5_{\pm 1.3}$	$83.8_{\pm 1.8}$	$83.6_{\pm 0.2}$	$\mathbf{-2.7}_{\pm 0.7}$
	CUSP-Merge	$\mathbf{89.0}_{\pm 1.5}$	$\mathbf{84.1}_{\pm 2.5}$	$\mathbf{84.1}_{\pm 0.8}$	$-4.9_{\pm 1.2}$
14	NUFILT	$83.9_{\pm 1.4}$	$\mathbf{78.2}_{\pm 1.1}$	$78.1_{\pm 0.2}$	$\mathbf{-5.7}_{\pm 0.8}$
	CUSP-Merge	$\mathbf{85.8}_{\pm 1.5}$	$\mathbf{78.2}_{\pm 1.5}$	$\mathbf{78.2}_{\pm 0.6}$	$-7.6_{\pm 0.9}$
20	NUFILT	$80.2_{\pm 2.0}$	$71.3_{\pm 0.9}$	$71.1_{\pm 0.9}$	$\mathbf{-8.9}_{\pm 2.3}$
	CUSP-Merge	$\mathbf{84.9}_{\pm 1.3}$	$\mathbf{72.9}_{\pm 1.3}$	$\mathbf{72.9}_{\pm 1.1}$	$-12.0_{\pm 1.8}$

A.5 Baseline Details

For completeness, we list the full set of baselines used in our evaluation. The reference paradigms comprise the pretrained model, independently fine-tuned experts, and continual fine-tuning. The conventional merging methods evaluated under the continual protocol include SWA (Izmailov et al., 2018), Task Arithmetic (Ilharco et al., 2022), TIES-Merging (Yadav et al., 2023), MagMax-IND (Marczak et al., 2024), Layer-Wise AdaMerging (Yang et al., 2024), LoRA-WEMoE (Tang et al., 2024b), WUDI-Merging (Cheng et al., 2025), Iso-C (Marczak et al., 2025), KnOTS-TIES (Stoica et al., 2025), and TSV-M (Gargiulo et al., 2025). We also compare with methods designed specifically for continual model merging, including OPCM (Tang et al., 2025), DOP (Yang et al., 2026), and NUFILT (Qiu et al., 2026c).

B Additional Results

B.1 Stage-Wise Performance Decomposition of BWT

To contextualize BWT, we decompose it into historical-task accuracy at task incorporation and after the complete sequence:

$$\mathrm{LAcc}_{\mathrm{old}} = \frac{1}{T-1}\sum_{i=1}^{T-1} A_i\left(\theta_i^{\mathrm{merged}}\right), \tag{13}$$

$$\mathrm{FAcc}_{\mathrm{old}} = \frac{1}{T-1}\sum_{i=1}^{T-1} A_i\left(\theta_T^{\mathrm{merged}}\right), \tag{14}$$

such that $\mathrm{BWT} = \mathrm{FAcc}_{\mathrm{old}} - \mathrm{LAcc}_{\mathrm{old}}$. Therefore, BWT depends not only on final historical-task retention but also on the accuracy achieved when each task is first incorporated.

As shown in Table 4, CUSP-Merge consistently achieves higher $\mathrm{LAcc}_{\mathrm{old}}$ than NUFILT. It also obtains higher final historical-task accuracy in the 8- and 20-task settings and comparable performance in the 14-task setting. Therefore, its more negative BWT mainly reflects a higher task-learning reference point rather than weaker final retention of historical knowledge.

B.2 Further Comparison with Single Memory

Although Single Memory achieves an overall average accuracy close to that of CUSP-Merge, the aggregate result does not fully reveal their differences across task-arrival positions and task-level tail performance. We therefore conduct further analyses on the ViT-B/32 20-task setting from two perspectives: early-task retention and worst-task robustness.

Retention of Early Tasks. We align tasks according to their arrival positions in each sequence. Since tasks introduced early undergo more subsequent merging steps, their final accuracy provides a direct measure of long-term historical retention. As shown in Fig. 4, CUSP-Merge achieves a positive average improvement at each of the first ten positions, with an average gain of 2.37 percentage

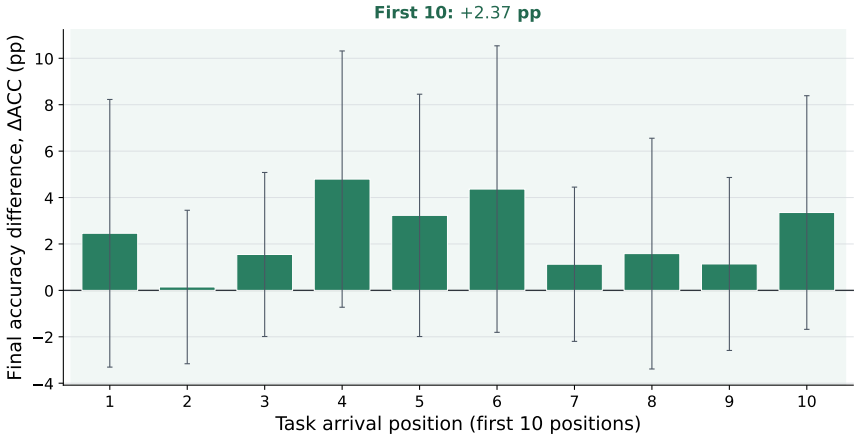


Figure 4: Final accuracy differences between CUSP-Merge and Single Memory for the first ten task-arrival positions. Bars and error bars denote the paired mean difference and standard deviation over ten task orders, respectively. Positive values indicate higher final retention accuracy for CUSP-Merge at the corresponding positions.

Table 5: Worst-task performance of CUSP-Merge and Single Memory on the ViT-B/32 20-task setting. Results are reported as the mean and standard deviation over ten task orders.

Metric	Single Memory	CUSP-Merge	Δ
Overall ACC	$72.28_{\pm 1.85}$	$72.93_{\pm 1.14}$	$+0.66$
Worst-task ACC	$35.43_{\pm 8.32}$	$\mathbf{43.52}_{\pm 7.07}$	$+8.08$
ACC on Single's worst task	$35.43_{\pm 8.32}$	$\mathbf{44.41}_{\pm 7.30}$	$+8.97$

points. A two-sided paired t-test on the per-order averages over these positions confirms that the improvement is statistically significant ($p = 0.006$). When all 20 positions are considered, task-arrival position is strongly negatively correlated with the accuracy gain of CUSP-Merge (Pearson's $r = -0.74$), suggesting that tasks introduced earlier and exposed to longer subsequent merging trajectories benefit more from the structured directional memory.

Task-wise Robustness. We further compare the final accuracy of the two methods on individual tasks, ordered by the average accuracy of Single Memory from low to high. As shown in Fig. 5, the differences are not uniformly distributed across tasks, with larger improvements on some low-accuracy tasks. This result suggests that aggregate accuracy may obscure substantial task-level variation. To quantify task-level tail performance, Table 5 reports the worst-task accuracy of both methods. For Worst-task ACC, we independently take the minimum accuracy among the 20 tasks for each method and each task order. For ACC on Single's worst task, we first identify the worst-performing task of Single Memory in each order and then evaluate both methods on the same task. Despite an overall ACC difference of only 0.66 percentage points, CUSP-Merge increases the minimum task accuracy in each task order from 35.43 ± 8.32 to 43.52 ± 7.07. On the same tasks where Single Memory performs worst, CUSP-Merge improves final accuracy by 8.97 percentage points on average. These results demonstrate that CUSP-Merge substantially reduces the risk of severe task-level degradation under a fixed memory budget.

B.3 Extended Hyper-parameter Sensitivity

Figure 6 extends the sensitivity analysis to the ViT-B/32 settings with 14 and 20 tasks over ten task orders. The overall trends are consistent with the 8-task results: performance decreases with an overly large M, saturates when η decreases beyond $1/512$, and shows limited gains when B exceeds 320. Performance is also relatively stable across different B_p values. These results further support the default settings $M = 2$, $\eta = 1/512$, $B = 320$, and $B_p = 64$.

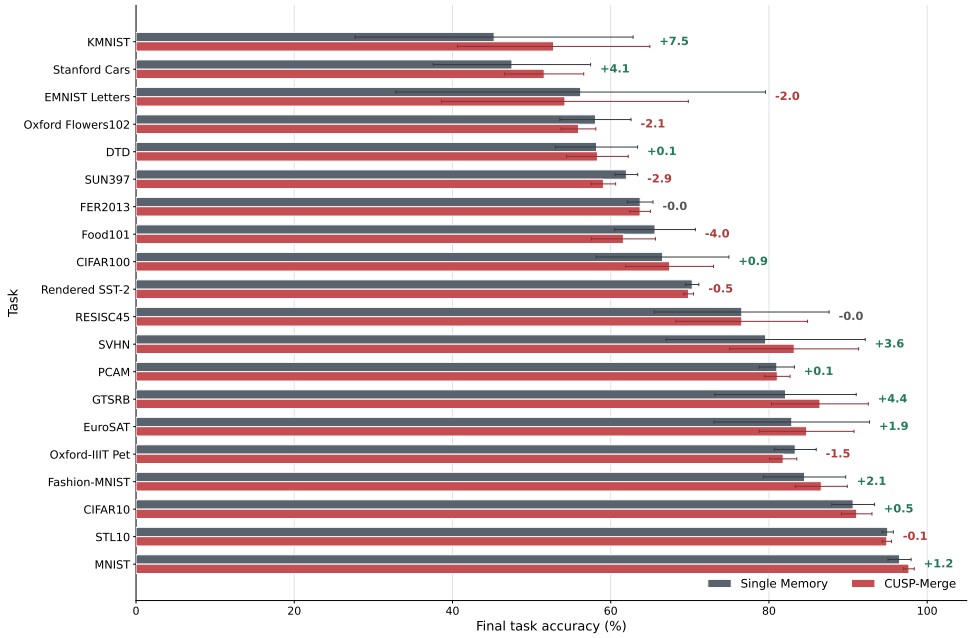


Figure 5: Task-wise final accuracy of CUSP-Merge and Single Memory on the ViT-B/32 20-task setting. Bars and error bars denote the mean and standard deviation over ten task orders, respectively. Values on the right report $\Delta = \mathrm{CUSP} - \mathrm{Single}$. Tasks are ordered by the average accuracy of Single Memory from low to high.

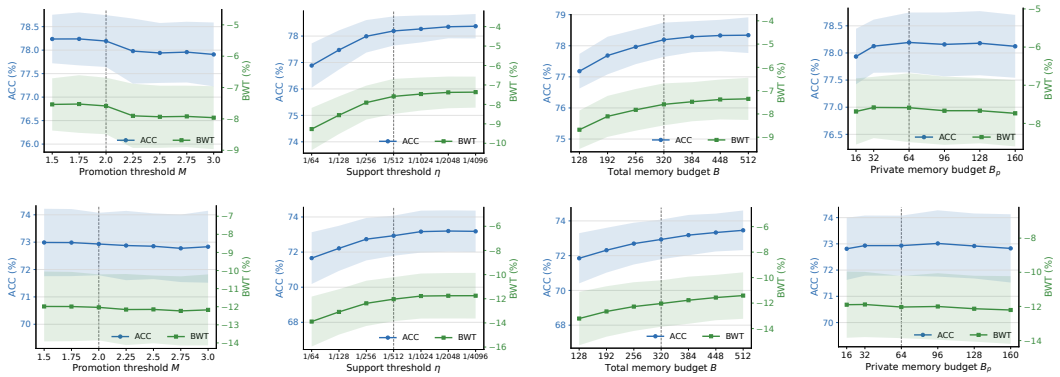


Figure 6: Hyper-parameter sensitivity of CUSP-Merge on ViT-B/32 with 14 tasks (top) and 20 tasks (bottom). From left to right: M, η, B, and B_p. Shading denotes standard deviation, and dashed lines indicate default values.

B.4 Complete Performance–Storage Comparison

Table 6 extends the comparison in the main text to task sequences of different lengths. Growing Memory retains r directions for every task, causing its storage to increase linearly with the number of tasks, whereas CUSP-Merge maintains a fixed budget of $B = 320$. Increasing r generally improves Growing Memory up to $r = 32$ or 64, but $r = 128$ provides no further benefit despite substantially higher storage.

The advantage of a fixed budget becomes more pronounced as the sequence grows. With 20 tasks, CUSP-Merge outperforms Growing Memory with $r = 16$ by 0.97 ACC points under nearly identical storage. Compared with the best-performing $r = 64$ variant, CUSP-Merge trails by only 0.88 points while using 25.0% of its additional memory. With 14 tasks, it achieves nearly identical ACC (78.19 vs. 78.23) using only 35.8% of the additional memory. These results demonstrate that CUSP-

Table 6: Complete performance–storage comparison on ViT-B/32 over ten task orders. ACC is reported in percent (mean ± standard deviation). Mem. denotes the additional persistent memory relative to the backbone parameters.

	8 Tasks		14 Tasks		20 Tasks	
Method	ACC	Mem.	ACC	Mem.	ACC	Mem.
Growing ($r = 8$)	$83.43_{\pm 0.75}$	6.1%	$76.46_{\pm 0.72}$	10.6%	$71.56_{\pm 1.39}$	15.2%
Growing ($r = 16$)	$84.23_{\pm 0.53}$	12.1%	$77.23_{\pm 0.83}$	21.2%	$71.96_{\pm 1.35}$	30.3%
Growing ($r = 32$)	$84.75_{\pm 0.59}$	24.3%	$77.75_{\pm 0.80}$	42.5%	$72.50_{\pm 1.23}$	60.7%
Growing ($r = 64$)	$84.70_{\pm 0.80}$	48.6%	$78.23_{\pm 0.95}$	85.0%	$73.81_{\pm 1.14}$	121.4%
Growing ($r = 128$)	$83.07_{\pm 1.46}$	97.1%	$77.54_{\pm 1.24}$	170.0%	$73.50_{\pm 1.45}$	242.8%
CUSP-Merge	$84.15_{\pm 0.78}$	30.4%	$78.19_{\pm 0.55}$	30.4%	$72.93_{\pm 1.14}$	30.4%

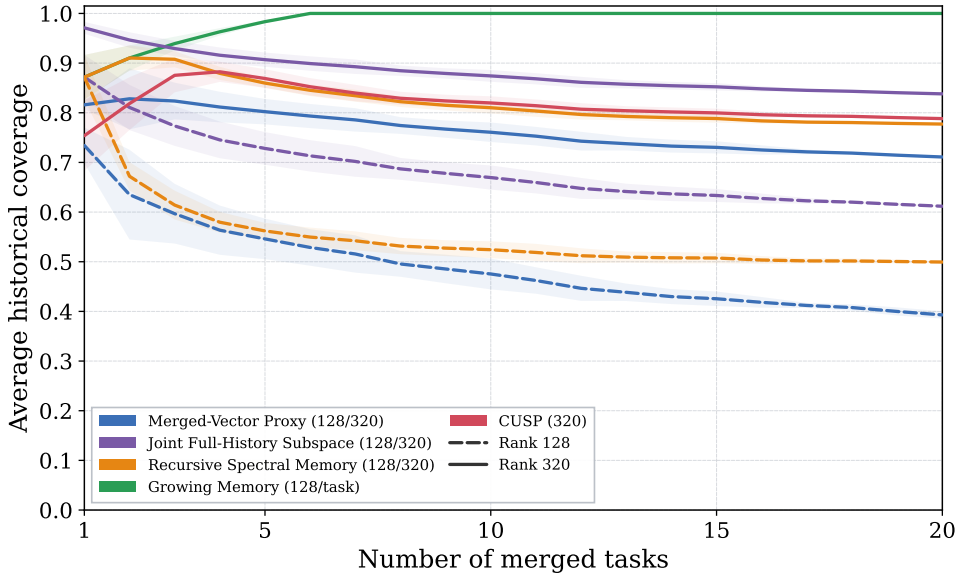


Figure 7: Historical direction coverage on ViT-B/32 with 20 tasks over ten task orders. Colors distinguish historical representations; dashed and solid lines denote ranks 128 and 320, respectively. Shaded regions indicate the standard deviation across task orders.

Merge provides a favorable performance–storage trade-off, particularly for longer continual merging sequences.

B.5 Analysis of Historical Direction Preservation

Extended Historical Direction Coverage. To further evaluate the ability of CUSP to preserve historical directions, we extend the motivation analysis to rank 320 on ViT-B/32 with 20 tasks over ten task orders, aligning the reference subspace size with the fixed directional budget $B = 320$ of CUSP. All coverage values are computed over the full task vectors and averaged across the analyzed `attn` and `fc1` layers. As shown in Fig. 7, increasing the rank consistently improves historical coverage. At stage 20, the coverage of the Merged-Vector Proxy, Joint Full-History Subspace, and Recursive Spectral Memory increases from 0.3929, 0.6118, and 0.4994 at rank 128 to 0.7109, 0.8380, and 0.7772 at rank 320, respectively.

Nevertheless, the rank-320 Merged-Vector Proxy remains 12.71 percentage points below the corresponding Joint Full-History Subspace and 6.63 points below Recursive Spectral Memory. This indicates that increasing the subspace rank alleviates coverage degradation but cannot eliminate the information loss caused by reconstructing historical directions from the cumulative merged vector. Under the same fixed capacity of 320 directions, CUSP achieves a coverage of 0.7882 ± 0.0014,

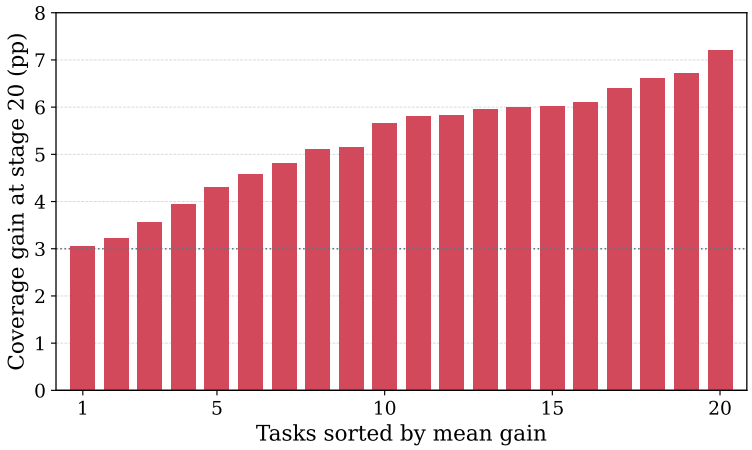


Figure 8: Task-wise coverage gains contributed by Private Memory at stage 20 on ViT-B/32 with 20 tasks. Bars denote the mean gain over ten task orders and are sorted in ascending order. The dotted line marks a gain of three percentage points.

outperforming the rank-320 Merged-Vector Proxy and Recursive Spectral Memory by 7.73 and 1.10 percentage points, respectively, while reducing the gap to the rank-320 Joint Full-History Subspace to 4.98 points. These results demonstrate that, compared with the cumulative merged vector or uniform recursive spectral compression, CUSP preserves historical directions more effectively through its structured fixed-budget memory.

Shared–Private Direction Preservation. To examine whether Private Memory complements historical directions missed by Shared Memory, we compute the task-wise Private coverage gain at stage 20, defined as the difference in full-task-vector coverage between Shared+Private Memory and Shared Memory alone. The directional budgets of Shared and Private Memory are 256 and 64, respectively, and coverage is averaged over the `attn` and `fc1` layers across ten task orders. As shown in Fig. 8, Private Memory improves coverage for all 20 tasks. The task-wise gains range from 3.05 to 7.21 percentage points, with an average gain of 5.30 points. Accordingly, the overall coverage at stage 20 increases from 0.7352 ± 0.0015 with Shared Memory alone to 0.7882 ± 0.0014 with Shared+Private Memory. These consistent gains show that the contribution of Private Memory is not concentrated on a small subset of tasks; instead, it continuously complements historical directions not represented by Shared Memory, confirming the non-redundant role of the Private component in the Shared–Private decomposition.