MINGLE: Mixture of Null-Space Gated Low-Rank Experts for Test-Time Continual Model Merging

Zihuan Qiu[1] **Yi Xu**[2] **Chiyuan He**[1] **Fanman Meng**[1*]
Linfeng Xu[1] **Qingbo Wu**[1] **Hongliang Li**[1]
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Dalian University of Technology, Dalian, China
{zihuanqiu@std., cyhe@std., fmmeng@, lfxu@, qbwu@, hlli@}uestc.edu.cn, yxu@dlut.edu.cn

Abstract

Continual model merging integrates independently fine-tuned models sequentially without access to the original training data, offering a scalable and efficient solution for continual learning. However, existing methods face two critical challenges: parameter interference among tasks, which leads to catastrophic forgetting, and limited adaptability to evolving test distributions. To address these issues, we introduce the task of Test-Time Continual Model Merging (TTCMM), which leverages a small set of unlabeled test samples during inference to alleviate parameter conflicts and handle distribution shifts. We propose MINGLE, a novel framework for TTCMM. MINGLE employs a mixture-of-experts architecture with parameter-efficient, low-rank experts, which enhances adaptability to evolving test distributions while dynamically merging models to mitigate conflicts. To further reduce forgetting, we propose Null-Space Constrained Gating, which restricts gating updates to subspaces orthogonal to prior task representations, thereby suppressing activations on old tasks and preserving past knowledge. We further introduce an Adaptive Relaxation Strategy that adjusts constraint strength dynamically based on interference signals observed during test-time adaptation, striking a balance between stability and adaptability. Extensive experiments on standard continual merging benchmarks demonstrate that MINGLE achieves robust generalization, significantly reduces forgetting, and consistently surpasses previous state-of-the-art methods by 7–9% on average across diverse task orders. Our code is available at: `https://github.com/zihuanqiu/MINGLE`

1 Introduction

Continual learning aims to incrementally adapt machine learning models to new tasks without forgetting previously learned knowledge, addressing the critical challenge of catastrophic forgetting [43]. However, conventional continual learning approaches typically require continuous access to original training data, raising significant concerns about privacy and substantial computational overhead due to retraining efforts, thus limiting their applicability in dynamic, data-sensitive environments.

To address these limitations, recent works have explored an alternative paradigm known as continual model merging (CMM), which sequentially integrates independently fine-tuned models directly in parameter space, without revisiting any training data [38, 51, 70]. CMM typically operates under a "merge-to-transfer" paradigm: given a pretrained model θ_0 and independently fine-tuned models $\{\theta_t\}_{t=1}^{T}$, a unified model is constructed sequentially by combining task-specific weight updates $\Delta\theta_t = \theta_t - \theta_0$ via weighted averaging or projection-based strategies [24, 81, 69].

*Corresponding author

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

Despite its advantages in scalability, data privacy, and distributed training capabilities [44, 14, 57], existing CMM methods still encounter critical issues, notably severe parameter interference between tasks and limited adaptability to evolving test distributions. This parameter interference arises because, as fine-tuned models are incrementally merged, overlapping or conflicting parameter updates accumulate, resulting in severe forgetting of previously learned tasks. To mitigate this interference, recent methods introduce structural constraints such as orthogonal projection [70, 78], model linearization [38, 69], and pruning-based sparsification [86, 90]. However, their effectiveness diminishes as task count grows and interference becomes increasingly entangled. Moreover, models merged across tasks often fail to generalize effectively, particularly when facing unseen or shifting task conditions. These flaws result in severe forgetting of earlier tasks and substantial performance gaps compared to the upper bound achieved by individually fine-tuned models. As shown in Fig. 1, TA [24] suffers from large performance gaps and strong forgetting, reflected by low accuracy and negative backward transfer. OPCM [70] improves over TA via orthogonalized merging but still shows notable degradation.

To overcome these limitations, we propose a novel continual merging paradigm—**Test-Time Continual Model Merging (TTCMM)**—which explicitly introduces the concept of test-time adaptation (TTA) [75, 63] into model merging. Unlike prior TTA-based multi-task merging methods [87, 68, 88], which assume simultaneous availability of models and test data from all tasks, TTCMM utilizes only a small set of unlabeled samples from the current task, making it uniquely suited for realistic continual scenarios where revisiting historical data is often infeasible.

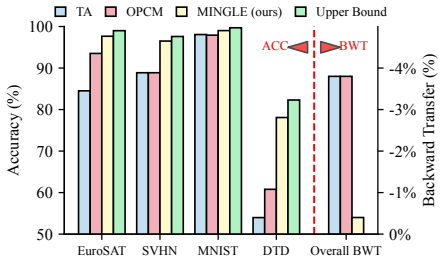


Figure 1: After 8-task continual merging: accuracy on first four tasks and overall BWT.

In this paper, we propose MINGLE (**MI**xture of **N**ull-Space **G**ated **L**ow-Rank **E**xperts), a method designed to continually merge independently fine-tuned models at test-time while preserving prior knowledge. MINGLE employs a mixture-of-experts architecture [27, 45] composed of lightweight LoRA-based [21] experts, enabling efficient and flexible test-time adaptation. To robustly prevent interference from previously learned tasks, we introduce a novel **Null-Space Constrained Gating** mechanism, restricting gating updates to task-orthogonal subspaces. Additionally, we propose an **Adaptive Relaxation Strategy** to dynamically modulate constraint strength based on test-time interference feedback during adaptation.

Extensive experiments on standard continual learning benchmarks show that MINGLE consistently outperforms previous state-of-the-art approaches by 7–9% on average, achieving robust generalization and strong resistance to catastrophic forgetting across diverse continual learning scenarios. Remarkably, these improvements are achieved entirely without any access to original training data, demonstrating the effectiveness of our TTCMM paradigm and the power of test-time adaptation in continual learning.

Our contributions are summarized as follows:

- We formalize test-time continual model merging (TTCMM), a novel task that leverages unlabeled test samples to merge independently fine-tuned models.
- We propose MINGLE, a TTCMM framework with Adaptive Null-Space Constrained Gating to effectively balance stability and plasticity.
- Extensive experiments show that MINGLE achieves state-of-the-art performance, consistently outperforming prior methods in accuracy, robustness and resistance to forgetting.

2 Related Work

Continual Learning. Continual learning (CL) seeks to mitigate catastrophic forgetting [43], where learning new tasks overwrites prior knowledge. Regularization-based methods constrain updates with importance weights [31, 92, 2, 30, 82], while distillation aligns outputs to preserve knowledge [20, 12, 60, 52]. Replay methods store exemplars or generate surrogates with prompts, prototypes, or generators [55, 39, 79, 61, 53], and dynamic architectures expand capacity via growth or ensembling [35, 98, 42]. Recent work leverages lightweight adapters or prompts in pre-trained models for efficient transfer [89, 23, 80]. Model merging offers an alternative route. Some methods remain close to

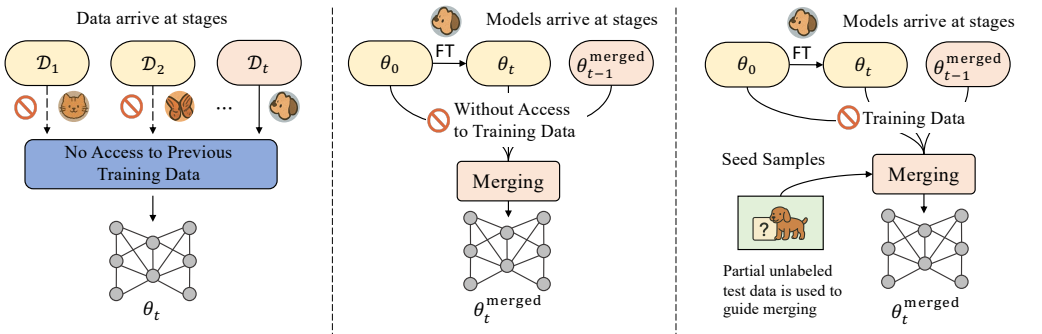


Figure 2: Comparison of three continual learning paradigms. (a) Conventional Continual Learning trains models sequentially with data arriving in stages, without access to previous task data. (b) Continual Model Merging continually fuses independently trained models, without access to any training data. (c) Test-Time Continual Model Merging improves merging by leveraging a few unlabeled test samples from the current task.

conventional CL by sequentially fine-tuning and merging models to reduce forgetting [41, 42, 16], typically requiring training data. In contrast, continual model merging [28, 38, 5, 51, 70] merges *independently fine-tuned models* without revisiting training data, enabling greater scalability and privacy.

Model Merging. Early work merged models via direct parameter averaging [72, 58], later refined by linear mode connectivity [13, 1]. Wortsman *et al.* [81] showed that weight averaging can also enhance robustness and out-of-distribution generalization. Task Arithmetic (TA) [24] views models as task vectors to be summed, but relies on weight disentanglement [49], often violated under standard fine-tuning, motivating structured training [28, 65]. Beyond averaging, interference-aware methods reweight or sparsify parameters [86, 90], or fuse models via distillation and clustering [88, 76]. LoRA-based tuning [21] introduces additional entanglement challenges, spurring gradient-free or retrieval-based strategies [22, 94, 95]. More recently, dynamic merging [68, 40] adapts parameters conditioned on inputs, achieving higher flexibility and performance, but remains limited to multi-task fusion and unexplored in continual settings.

Test-Time Adaptation. TTA adapts models at inference to mitigate distribution shift. Early approaches used self-supervised objectives [66], entropy minimization [75], or regularized updates [63]. Online TTA adapts continuously [25], while batch-wise variants ignore temporal structure [17]. To enhance stability, later work introduced confidence filtering [48], EMA [11], partial updates [91], test-time augmentation [93], and adaptive BatchNorm [56]. For vision–language models, TTA often employs prompts or adapters [59, 15, 36]. We draw on TTA to guide merging, aligning fused models with evolving test distributions.

Relation to Prior Work. Most related to our work are MoE-Adapter [89] and WEMOE [68], both built on MoE architectures [27, 29]. MoE-Adapter follows conventional continual learning, embedding expert modules that are jointly trained across tasks. In contrast, we adopt a model-merging paradigm, where experts are extracted from independently fine-tuned models and inserted without further training. WEMOE incorporates test-time adaptation but targets multitask learning, assuming simultaneous access to all models and data. By contrast, MINGLE is tailored for the more challenging continual merging setting.

3 MINGLE: Mixture of Null-Space Gated Low-Rank Experts

3.1 Preliminaries

Problem Setting. We study continual learning in a model merging setting, where a sequence of task-specific models $\{\theta_1, \ldots, \theta_T\}$ are independently fine-tuned from a shared pre-trained model θ_0, each using a dataset $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}$ with label space $\mathcal{C}_i \subset \mathcal{Y}$. The goal is to construct a unified model θ_T^{merged} that generalizes across the combined label space $\mathcal{C}_{1:T} = \bigcup_{i=1}^{T} \mathcal{C}_i$.

Unlike conventional continual learning, we assume *no access* to training data during merging. All adaptation happens directly in parameter space. This paradigm is relevant in scenarios where only final fine-tuned models are retained, while original training data is discarded due to privacy, storage, or accessibility constraints.

To contextualize this, we compare with two related paradigms in Fig. 2:

- **Conventional Continual Learning.** A single model θ is sequentially updated on $\mathcal{D}_1, \ldots, \mathcal{D}_T$, discarding previous data. It requires direct training data access and extensive retraining.
- **Continual Model Merging.** A sequence of models are merged incrementally in parameter space without access to training data and earlier models: $\theta_t^{\text{merged}} = \text{Merge}(\theta_{t-1}^{\text{merged}}, \theta_t)$.
- **Test-Time Continual Model Merging.** An extension of the above where a small unlabeled subset $\mathcal{D}_t^{\text{seed}} \subset \mathcal{D}_t^{\text{test}}$ (*e.g.*, 5 samples per class) is available at each stage to provide lightweight task-specific guidance. We refer to $\mathcal{D}_t^{\text{seed}}$ as the *seed samples* of task t.

Existing Continual Merging Strategies. Let θ_0 denote the parameters of a pre-trained model. The corresponding task vector is defined as $\Delta\theta_t = \theta_t - \theta_0$.

- **Continual Task Arithmetic (C. TA).** A simple additive merge [24]: $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + \lambda\Delta\theta_t$, where λ is a scalar. While training-free, it is sensitive to λ and prone to task interference.
- **Orthogonal Projection-based Continual Merging (OPCM).** Tang et al. [70] propose projecting each $\Delta\theta_t$ onto the orthogonal complement of previous directions: $\theta_t^{\text{merged}} = \theta_0 + \frac{1}{\lambda_t}\left[\lambda_{t-1}\Delta\theta_{t-1}^{\text{merged}} + \mathcal{P}^{(t-1)}(\Delta\theta_t)\right]$, where $\mathcal{P}^{(t-1)}$ retains components orthogonal to previous updates. This reduces interference but ignores adaptation to task distributions.

To address these issues, we present MINGLE, which leverages $\mathcal{D}_t^{\text{seed}}$ to modulate the integration of θ_t at test-time, enhancing alignment to test distribution and mitigating task interference.

3.2 Motivation and Theoretical Analysis

Most existing continual model merging methods combine fine-tuned models via static averaging, where each expert is assigned fixed coefficients, thereby enforcing the *same* mixing rule across the whole input space. Consequently, it cannot specialize to regions where one expert is clearly superior. In contrast, a Mixture-of-Experts (MoE) equips every input with a *data-dependent* gate $g(x) = (g_1(x), \ldots, g_T(x))$ that selects or re-weights experts on-the-fly. We give a formal comparison between static averaging and dynamic MoE under a noisy-routing scenario. [2]

Theorem 1 (Dynamic MoE versus Static Averaging)**.** *Let $\{(D_t, f_t)\}_{t=1}^T$ be T independent tasks with priors $P(t)$ and per-task risks $R_t(i)$. For any static mixture $h_{\text{static}}(x) = \sum_{i=1}^T \alpha_i f_i(x)$ and any hard-routed MoE $h_{\text{MoE}}(x) = f_{i^\star(x)}(x)$ with task-specific routing errors ε_t:*

$$R(h_{\text{MoE}}) \;=\; R_{\text{ideal}} + \sum_{t=1}^{T} P(t)\,\varepsilon_t\big(R_{\textit{wrong},t} - R_t(t)\big), \tag{1}$$

where $R_{\text{ideal}} = \sum_t P(t)R_t(t)$ and $R_{\textit{wrong},t} = \frac{1}{T-1}\sum_{i\neq t} R_t(i)$. Moreover,

1. (Perfect routing) *If $\varepsilon_t = 0$ for all t, then $\inf_g R(h_{\text{MoE}}) \;<\; \inf_{\boldsymbol{\alpha}} R(h_{\text{static}})$ whenever at least two tasks disagree on their best expert.*
2. (Noisy routing) *If $\sum_t P(t)\varepsilon_t\big(R_{\textit{wrong},t} - R_t(t)\big) < R^*_{\text{static}} - R_{\text{ideal}}$, where $R^*_{\textit{static}} = \inf_{\boldsymbol{\alpha}} R(h_{\textit{static}})$, then the MoE still attains lower risk than any static mixture.*

The theory above motivates a design that (i) keeps experts specialized and (ii) prevents interference between tasks. Our **MINGLE** framework achieves both goals by combining

- **Low-rank experts** f_t that capture task-specific variations with minimal parameters, and
- **Null-space constrained gating** that projects gradient updates away from subspaces spanned by previously activated features, keeping ε_t small without harming earlier experts.

[2] Symbols and proofs are deferred to App. A

3.3 Low-Rank Expert Mixture for Continual Model Merging

We adopt MoE framework for continual model merging, in which each task i is equipped with a low-rank expert f_i and an associated input-dependent gating function g_i. These components are injected into the linear layers of the backbone (*e.g.*, CLIP visual encoder). The gate g_i modulates expert activation based on the input features, allowing for fine-grained, localized task specialization.

Mixture of Low-Rank Expert. When a new task t arrives, a dedicated expert f_t and its gate g_t are appended to the model. The output of a given l-th layer[3] can be formulated as follows:

$$\theta_t^{\text{merged},(l)}(X) = \theta_{t-1}^{\text{merged},(l)}(X) + g_t^{(l)}(X) \cdot f_t^{(l)}(X) = \theta_0^{(l)}(X) + \sum_{i=1}^{t} g_i^{(l)}(X) \cdot f_i^{(l)}(X). \quad (2)$$

where only the gate g_t is adaptable during testing, while all experts $\{f_i\}_{i=1}^{t}$ and old gates $\{g_i\}_{i=1}^{t-1}$ remain frozen to preserve prior knowledge. To construct expert f_t, we first project the task vector $\Delta\theta_t$ onto the orthogonal complement of previously learned directions, following OPCM [70]:

$$\mathcal{P}^{(t-1)}(\Delta\theta_t) = \sum_{p,q=a,p\neq q}^{m,n} \langle \Delta\theta_t, u_p^{(t-1)} v_q^{(t-1)\top} \rangle_F u_p^{(t-1)} v_q^{(t-1)\top}, \quad (3)$$

where $u_p^{(t-1)}$ and $v_q^{(t-1)}$ are the p-th and q-th singular vectors from the singular value decomposition (SVD) of previous experts $\sum_{i=1}^{t-1} f_i(X)$, and α denotes the effective rank of previous experts. This projection removes previously learned directions to mitigate interference. We then apply a rank-r truncated SVD for $\mathcal{P}^{(t-1)}(\Delta\theta_t)$ to construct a low-rank expert [21]:

$$f_t = BA = (\tilde{U}\tilde{\Sigma})\tilde{V}^\top, \quad (4)$$

where $\tilde{U} \in \mathbb{R}^{d_1 \times r}$, $\tilde{\Sigma} \in \mathbb{R}^{r \times r}$, and $\tilde{V} \in \mathbb{R}^{d_2 \times r}$, retaining the top r singular components. The resulting expert captures the principal directions while significantly reducing parameter overhead.

Each gating function is implemented as a linear projection:

$$g_t(X) = W_t^{(g)\top} X + b_t^{(g)}, \quad (5)$$

where $W_t^{(g)} \in \mathbb{R}^{d \times 1}$ and $b_t^{(g)} \in \mathbb{R}$ are *learnable* parameters. The gating function is adapted at test time using a small number of unlabeled test data.

Test-Time Adaptation. To encourage the merged model to retain task-specific behavior, we minimize the Kullback–Leibler divergence between its prediction and that of the corresponding individual fine-tuned model θ_t. We define the adaptation objective as:

$$L_t = \mathbb{E}_{x \sim \mathcal{D}_t^{\text{seed}}} \left[\text{KL} \left(p(x;\ \theta_t^{\text{merged}}) \,\|\, p(x;\ \theta_t) \right) \right]. \quad (6)$$

where $p(x;\theta)$ denotes the predictive distribution

3.4 Adaptive Null-Space Constrained Gating for Interference Mitigation

When merging models continually, the primary challenge of gating is to integrate new experts without disturbing prior task predictions. Consider two experts f_1, f_2 and their corresponding gates g_1, g_2. When evaluating on the first task domain X_1, the interference from g_2 can be quantified as:

$$\xi(g_2) = \| g_1(X_1) \cdot f_1(X_1) + g_2(X_1) \cdot f_2(X_1) - f_1(X_1) \|^2. \quad (7)$$

This measures the deviation introduced by g_2 on the domain where f_1 originally dominates. A desirable gating function should suppress $g_2(X_1)$, ensuring predictions on X_1 remain unaffected. However, as X_1 becomes inaccessible after adaptation, this error becomes unobservable and cannot be minimized directly, resulting in prediction drift and catastrophic forgetting.

Hard Null-Space Projection. After completing task t, we cache the l-th layer inputs in the seed buffer $\mathcal{D}_t^{\text{seed}}$ and estimate their covariance $\text{Cov}_t^{(l)} \in \mathbb{R}^{d \times d}$. Applying truncated SVD yields the

[3]The layer index l is omitted hereafter whenever it does not cause ambiguity.

top-k dominant subspaces $\tilde{U}_t^{(l)} \in \mathbb{R}^{d\times k}$. We then concatenate these with the subspaces from all previous tasks and orthonormalize: $U_t^{(l)} = \mathrm{orthonorm}\big[U_{t-1}^{(l)} | \tilde{U}_t^{(l)}\big] \in \mathbb{R}^{d\times tk}$. The hard projector is $P_t = I - U_t^{(l)} U_t^{(l)\top} \in \mathbb{R}^{d\times d}$. To suppress interference from tasks $\leq t-1$, the gating update for task t is $W_t^{(g,l)} \leftarrow W_t^{(g,l)} - \eta \nabla L_t^{(l)} P_{t-1}^{(l)}$. However, this projection may also discard gradient components that are informative for task t whenever its feature support overlaps with $\mathrm{span}(U_{t-1})$.

Adaptive Null-Space Relaxation. To restore plasticity, we replace the *all-one* eigenvalues of P_{t-1} with *soft* coefficients learned online.

(i) Interference statistics. For each column $u_p^{(l)}$ of $U_{t-1}^{(l)}$ we measure instantaneous alignment:

$$r_p^{(l)} = \|(\nabla L_t)^\top u_p^{(l)}\|_2 \;/\; \|\nabla L_t\|_2. \quad (8)$$

We maintain per-direction interference scores $S^{(m,l)} \in \mathbb{R}^k$ (initialized to 0) by applying an exponential moving average at each iteration m:

$$S^{(m,l)} = \beta\, S^{(m-1,l)} + (1-\beta)\, r^{(l)}, \quad (9)$$

which suppresses stochastic gradient noise while preserving the dominant interference directions.

(ii) Adaptive shrinkage. Each direction is attenuated by $\lambda^{(m,l)} = \exp(-\gamma\, S^{(m,l)})$ ($\gamma > 0$, $\lambda^{(m,l)} \in (0,1]$). Let $\Lambda_{t-1}^{(m,l)} = \mathrm{diag}(\lambda_1^{(m,l)}, \ldots, \lambda_{(t-1)\cdot k}^{(m,l)})$. The *relaxed projector* becomes:

$$\widetilde{P}_{t-1}^{(m,l)} = U_{t-1}^{(m,l)}\, \Lambda_{t-1}^{(m,l)}\, U_{t-1}^{(m,l)\top}, \quad (10)$$

interpolating smoothly between no protection ($\Lambda = 0$) and the hard null projector ($\Lambda = I$).

(iii) Update rule. We finally update:

$$W_t^{(g,l)} \;\leftarrow\; W_t^{(g,l)} - \eta\, \nabla L_t^{(l)}\, \widetilde{P}_{t-1}^{(l)}. \quad (11)$$

Algorithm 1 MINGLE Procedure.

Input: pre-trained model θ_0 and fine-tuned models $\{\theta_t\}_{t=1}^T$; seed data $\{\mathcal{D}_t^{\text{seed}}\}_{t=1}^T$; hyperparameters k, β, γ; learning rate η
Output: Merged model θ_T^{merged}
Init: $\theta_0^{\text{merged}} \leftarrow \theta_0$; $U_0^{(l)} \leftarrow \emptyset$
for task $t = 1$ **to** T **do**
  ▷ `create low-rank experts (Eqs. 3 and 4)`
  $f_t = \mathrm{SVD}_{\text{TRUNC.}}\big(\underbrace{\mathcal{P}^{(t-1)}(\theta_t - \theta_0)}_{\text{use } \theta_1-\theta_0 \text{ when } t=1}\big) = B_t A_t$
  ▷ `add expert & initialize gate (Eq. 5)`
  $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + g_t \cdot f_t,\ \{W_t^{(g)}, b_t^{(g)}, S^{(0)}\} \leftarrow 0$
  for $m = 1$ **to** *total iterations* **do**
    $X \leftarrow \mathrm{batch}(\mathcal{D}_t^{\text{seed}})$
    $\{\nabla_{W_t^{(g)}} L, \nabla_{b_t^{(g)}} L\} \leftarrow \nabla L_t(X, \theta_t^{\text{merged}}, \theta_t)$
    if $t > 1$ **then**
      ▷ `project gradient onto null-space (Eqs. 8-11)`
      $S^{(m)} = \beta S^{(m-1)} + (1-\beta) r$
      $\nabla_{W_t^{(g)}} L_t \leftarrow \nabla_{W_t^{(g)}} L_t\, \widetilde{P}_{t-1}$
    end
    $W_t^{(g)} \leftarrow W_t^{(g)} - \eta\, \nabla_{W_t^{(g)}} L_t$
    $b_t^{(g)} \leftarrow b_t^{(g)} - \eta\, \nabla_{b_t^{(g)}} L_t$
  end
  ▷ `update dominant subspaces`
  $U_t \leftarrow \mathrm{orthonorm}\big[U_{t-1} \mid \tilde{U}_t\big]$
end

Relaxing the projector inevitably allows more residual interference than the hard null-space variant, yet empirically the increase is minor and is offset by markedly higher plasticity (Tab. 5), indicating a favorable *stability–plasticity balance*. The overall procedure is outlined in Algo. 1.

4 Experiments

We describe the experimental setup in Sec. 4.1, followed by the main results in Sec. 4.2 and further analysis and ablations in Sec. 4.3. Due to page limitations, detailed results are provided in the Appendix.

4.1 Experimental Setup

Datasets and Models. Following [70], we evaluate on image-classification tasks with CLIP-ViT backbones [54]. We consider 8, 14, and 20-task groups using ViT-B/32, ViT-B/16, and ViT-L/14 models, each fine-tuned on up to 20 downstream tasks, with checkpoints from FusionBench [67]. To assess order sensitivity, we repeat experiments over 10 random seeds (42–51). For comparison with conventional CL, we use the Multi-domain Task-Incremental Learning (MTIL) benchmark [97] with eleven vision tasks. Beyond vision, we evaluate on eight GLUE language tasks [74] with a Flan-T5-base backbone [6].

Table 1: Comparative results of continual merging methods, reporting average accuracy (ACC) and backward transfer (BWT) over ten task orders (mean±std). DM and DA denote method assumptions: dynamic merging or test data access. Best results are in bold; second-best are underlined. MINGLE* denotes a lightweight variant.

	Method	Assump.	ViT-B/32			ViT-B/16			ViT-L/14		
		DM / DA	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
	FINE-TUNED	– / –	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) [26]	✗ / ✗	66.3 ±0.0	65.4 ±0.0	61.1 ±0.0	72.3 ±0.0	69.7 ±0.0	64.8 ±0.0	80.0 ±0.0	77.5 ±0.0	71.1 ±0.0
	C. TASK ARITHMETIC [24]	✗ / ✗	67.5 ±0.0	66.5 ±0.0	60.0 ±0.0	77.1 ±0.0	70.9 ±0.6	64.2 ±0.0	82.1 ±0.0	77.9 ±0.0	70.3 ±0.0
	C. TIES-MERGING [86]	✗ / ✗	49.0 ±10.2	66.2 ±0.6	59.9 ±0.7	66.8 ±3.7	70.5 ±0.8	63.0 ±1.6	64.3 ±7.0	78.0 ±0.6	68.3 ±0.9
	MAGMAX-IND [41]	✗ / ✗	70.7 ±0.0	67.0 ±0.0	61.2 ±0.0	76.7 ±1.8	67.0 ±0.0	62.5 ±0.0	83.4 ±0.0	71.2 ±0.0	71.2 ±0.0
	CONSENSUS TA [77]	✗ / ✗	67.1 ±0.4	64.1 ±0.8	45.8 ±1.5	72.8 ±0.5	69.0 ±0.0	49.9 ±1.9	80.4 ±0.5	75.0 ±1.0	51.3 ±2.4
	OPCM [70]	✗ / ✗	75.5 ±0.5	71.9 ±0.3	65.7 ±0.2	81.8 ±0.3	77.1 ±0.5	70.3 ±0.2	87.0 ±0.4	83.5 ±0.2	76.0 ±0.2
	C. LW ADAMERGING [87]	✗ / ✓	53.4 ± 3.2	59.8 ± 1.6	59.7 ± 7.4	59.9 ±2.3	64.3 ± 1.2	61.5 ± 1.1	68.8 ± 2.9	73.1 ± 5.7	66.9 ± 1.1
	C. LORA-WEMOE [68]	✓ / ✓	68.8 ± 7.8	63.8 ± 3.4	49.6 ± 15.4	72.6 ± 3.7	67.9 ± 2.9	55.0 ± 7.0	75.6 ± 7.8	74.0 ± 5.0	56.9 ± 19.8
	MINGLE (Ours)	✓ / ✓	**85.8** ±0.8	81.6 ±1.4	**77.1** ±2.0	**88.3** ±0.6	**84.9** ±0.8	**81.9** ±0.9	**91.8** ±0.2	88.8 ±0.7	**85.5** ±1.3
	MINGLE* (Ours)	✓ / ✓	85.0 ±0.5	**81.7** ±1.0	**77.1** ±1.3	87.0 ±0.6	84.7 ±1.0	81.6 ±1.3	91.4 ±0.3	**89.2** ±0.1	83.6 ±0.6
BWT (%) ↑	AVERAGE (SWA) [26]	✗ / ✗	-11.5 ±2.2	-8.0 ±1.3	-7.1 ±2.1	-9.7 ±1.5	-7.1 ±1.4	-7.3 ±1.7	-7.3 ±1.4	-5.8 ±1.0	-6.4 ±1.5
	C. TASK ARITHMETIC [24]	✗ / ✗	-9.6 ±1.5	-1.3 ±1.6	-3.4 ±1.0	-4.2 ±1.0	-1.3 ±0.4	-3.6 ±0.4	-7.1 ±0.8	-1.8 ±0.3	-3.3 ±0.3
	C. TIES-MERGING [86]	✗ / ✗	-15.3 ±8.0	**1.9** ±0.6	-1.5 ±0.7	-5.5 ±0.4	**1.4** ±0.7	-1.5 ±1.2	-13.0 ±5.7	-1.1 ±0.4	-2.9 ±1.0
	MAGMAX-IND [41]	✗ / ✗	-8.3 ±1.3	-7.4 ±1.4	-7.2 ±1.6	-6.1 ±1.3	-7.4 ±2.0	-8.0 ±2.2	-5.0 ±0.8	-6.0 ±2.1	-6.5 ±2.1
	CONSENSUS TA [77]	✗ / ✗	**3.8** ±0.9	-1.3 ±0.9	-11.8 ±1.9	**3.5** ±0.6	-1.1 ±0.8	-11.6 ±1.3	**2.4** ±0.6	-2.5 ±0.8	-16.5 ±1.5
	OPCM [70]	✗ / ✗	-6.3 ±1.1	-6.0 ±1.0	-7.8 ±1.5	-4.8 ±0.7	-5.1 ±1.4	-6.3 ±2.2	-2.6 ±1.0	-4.3 ±0.7	-6.5 ±1.8
	C. LW ADAMERGING [87]	✗ / ✓	-32.5 ±3.6	-24.1 ±1.7	-22.7 ±4.3	-27.8 ±2.7	-22.1 ±1.4	-21.4 ±1.2	-24.3 ±3.3	-19.6 ±1.7	-21.7 ±1.1
	C. LORA-WEMOE [68]	✓ / ✓	-20.4 ±9.0	-20.2 ±3.9	-24.5 ±10.0	-18.0 ±6.2	-18.8 ±3.4	-25.8 ±7.9	-17.8 ±5.9	-16.8 ±5.3	-27.9 ±17.2
	MINGLE (Ours)	✓ / ✓	-0.6 ±0.4	-1.1 ±0.3	-2.2 ±0.8	-0.4 ±0.1	-0.9 ±0.1	-1.9 ±0.4	-0.6 ±0.1	-1.0 ±0.3	-2.6 ±0.9
	MINGLE* (Ours)	✓ / ✓	-0.1 ±0.1	-0.4 ±0.1	**-1.3** ±0.6	-0.1 ±0.1	-0.3 ±0.1	**-1.0** ±0.4	-0.2 ±0.0	**-0.4** ±0.2	**-1.5** ±0.6

Table 2: Results of continual merging Flan-T5-base models on 8 tasks, ordered alphabetically.

Method	DM / DA	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	ACC ↑	BWT ↑
PRE-TRAINED	– / –	69.1	56.5	76.2	88.4	82.1	80.1	91.2	62.2	75.7	-
INDIVIDUAL	– / –	75.0	83.4	87.5	91.5	85.4	85.9	93.6	88.7	86.4	-
TASK ARITHMETIC	✗ / ✗	69.1	58.1	77.9	88.9	83.1	79.1	90.7	74.0	77.6	-4.6
TIES-MERGING	✗ / ✗	39.3	70.0	82.4	88.8	81.8	75.8	89.7	76.8	75.6	-6.1
OPCM	✗ / ✗	69.9	72.9	78.7	90.3	83.8	**83.0**	92.2	73.7	80.6	-2.5
LW ADAMERGING	✗ / ✓	69.1	58.1	77.9	88.9	83.1	79.1	90.7	74.2	77.6	-4.7
LORA-WEMOE	✓ / ✓	71.5	**80.6**	78.2	90.3	82.7	80.5	91.3	76.2	81.4	**0.1**
MINGLE (Ours)	✓ / ✓	**75.0**	78.2	**86.0**	**90.9**	**84.2**	80.5	**92.5**	**78.8**	**83.3**	**0.1**

Implementation Details. We insert low-rank experts into the CLIP vision encoder. Two variants are used: a full setup modifying all attention and MLP layers, and a lightweight one on `attn.qkv` and `mlp.fc1`. All experiments share a single set of *global* hyper-parameters across models and task orders. Each expert has rank $r = 64$; the null-space constraint uses $k = 3$, $\gamma = 1$, and $\beta = 0.99$. Adaptation runs for 50 iterations with Adam (lr 1e-4, batch size 16). For vision tasks we use 5 unlabeled samples per class, and for NLP tasks 100 in total, all without access to prior-task data.

Evaluation Metrics. We evaluate using average accuracy (ACC) and backward transfer (BWT) [37]. ACC is the mean accuracy of the final merged model across all tasks: $\text{ACC} = \frac{1}{T}\sum_{i=1}^{T} a_i(\theta_t^{\text{merged}})$, where $a_i(\cdot)$ is accuracy on task i. BWT measures forgetting by comparing performance on earlier tasks before vs. after the final merge: $\text{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left[a_i(\theta_T^{\text{merged}}) - a_i(\theta_i^{\text{merged}})\right]$.

4.2 Main Results

Overall Performance As shown in Tab. 1, MINGLE substantially outperforms previous continual merging methods on all CLIP backbones and task counts. It achieves the highest accuracy with backward forgetting kept near zero, demonstrating both strong forward learning and long-term stability. The lightweight variant performs on par with the full version, further underscoring the robustness of our approach. On NLP benchmarks (Tab. 2), MINGLE likewise attains the best overall accuracy and non-negative BWT, improving on multiple GLUE tasks while maintaining balanced performance across the suite. Together, these results across vision and language confirm that MINGLE

Table 3: Comparison of last accuracy (%) with conventional CL approaches on MTIL benchmark.

Method	Aircraft	Caltech101	CIFAR100	DTD	EuroSAT	Flowers	Food101	MNIST	Pets	Cars	SUN397	Avg.
Conventional CL (*Sequential fine-tuned*)												
WISE-FT [81]	27.2	90.8	68.0	68.9	86.9	74.0	87.6	**99.6**	92.6	77.8	81.3	77.7
ZSCL [97]	40.6	92.2	81.3	70.5	94.8	90.5	91.9	98.7	93.9	85.3	80.2	83.6
MOE-ADAPTER [89]	49.8	92.2	86.1	78.1	95.7	94.3	89.5	98.1	89.9	81.6	80.0	85.0
DIKI [71]	45.2	95.7	86.3	72.9	**98.0**	97.0	89.2	99.4	**94.2**	81.6	76.6	85.1
AWOFORGET [96]	42.4	92.7	83.2	73.2	97.0	91.8	**92.2**	99.1	93.9	**87.4**	**82.6**	85.0
DUAL-RAIL [85]	52.5	96.8	83.3	**80.1**	96.4	**99.0**	89.9	98.8	93.5	85.5	79.2	**86.8**
MAGMAX [41]	40.2	96.1	81.1	72.0	97.8	76.3	88.4	99.2	93.0	70.5	68.9	80.3
MINGLE-SEQ	**58.7**	**97.5**	**87.2**	79.7	97.3	87.2	90.1	**99.6**	93.0	80.4	73.3	85.8
Continual Merging (*Independent fine-tuned*)												
AVERAGE (SWA) [26]	26.5	92.3	74.3	48.4	73.7	74.0	87.1	84.0	91.2	67.5	68.5	71.6
C. TA [24]	26.6	92.5	74.5	48.7	74.3	74.4	87.0	85.5	91.2	67.7	68.6	71.9
C. TIES [86]	30.5	94.0	74.8	49.8	71.7	73.8	87.3	81.5	90.6	67.0	67.9	71.7
MAGMAX-IND [41]	29.9	93.7	78.4	46.1	58.3	68.1	86.8	82.8	91.4	62.7	69.3	69.8
OPCM [70]	35.7	95.9	77.0	54.6	90.3	76.4	87.1	96.3	93.3	70.1	70.5	77.0
MINGLE	**54.2**	**97.3**	**79.7**	**72.3**	**96.0**	**86.7**	**88.7**	**99.3**	**93.9**	**73.1**	**71.6**	**83.0**

Table 4: Robustness results of ViT-B/32 continually merged across 4 tasks.

	Method	Clean	Motion	Impulse	Gaussian	Pixelate	Spatter	Contrast	JPEG	Avg.
ACC (%) ↑	C. LW ADAMERGING [87]	56.0 ±5.3	47.5 ±4.4	43.1 ±2.3	43.3 ±3.4	18.1 ±4.7	46.6 ±3.0	48.9 ±4.8	49.1 ±4.0	44.9
	C. WEMOE [68]	3.4 ±0.8	3.1 ±0.4	4.3 ±1.4	3.4 ±1.4	3.0 ±1.6	4.0 ±0.9	3.3 ±0.7	4.0 ±1.2	3.6
	C. LORA-WEMOE [68]	78.7 ±4.5	71.0 ±4.9	55.0 ±3.8	59.4 ±3.8	24.9 ±24.9	60.5 ±3.8	68.5 ±4.8	69.7 ±4.4	61.0
	C. TASK ARITHMETIC [24]	77.5 ±0.0	66.0 ±0.0	58.9 ±0.0	59.6 ±0.0	29.7 ±0.0	63.5 ±0.0	66.0 ±0.0	67.8 ±0.0	61.1
	MAGMAX-IND [41]	79.1 ±0.0	69.0 ±0.0	60.6 ±0.0	61.5 ±0.0	33.0 ±0.0	66.4 ±0.0	68.6 ±0.0	69.9 ±0.0	63.5
	OPCM [70]	83.6 ±0.5	72.5 ±0.6	64.7 ±1.2	65.2 ±1.2	35.2 ±0.6	70.5 ±0.5	72.5 ±0.6	74.4 ±0.3	67.3
	MINGLE (Ours)	**89.9** ±0.4	**82.8** ±0.8	**67.5** ±2.0	**70.7** ±1.2	**37.9** ±0.4	**77.0** ±0.7	**80.1** ±0.8	**82.9** ±0.9	**73.2**
BWT (%) ↑	C. LW ADAMERGING [87]	-38.0 ±7.1	-37.3 ±5.9	-22.2 ±3.0	-25.2 ±4.5	-20.8 ±6.3	-28.6 ±4.0	-34.7 ±6.5	-36.1 ±5.3	-29.5
	C. WEMOE [68]	-30.7 ±3.1	-28.7 ±3.8	-22.1 ±11.6	-25.5 ±9.8	-8.0 ±4.5	-23.4 ±11.0	-27.6 ±5.6	-28.6 ±5.2	-24.3
	C. LORA-WEMOE [68]	-13.6 ±6.9	-14.6 ±8.2	-11.3 ±4.7	-10.2 ±3.7	-15.6 ±8.8	-10.8 ±3.9	-11.2 ±8.8	-16.7 ±6.6	-13.0
	C. TASK ARITHMETIC [24]	-4.8 ±0.9	-6.1 ±1.2	-1.6 ±3.0	-1.6 ±1.7	-2.7 ±1.5	-3.1 ±2.5	-6.1 ±1.2	-5.1 ±0.8	-3.9
	MAGMAX-IND [41]	-7.7 ±0.8	-8.1 ±1.5	-6.1 ±4.9	-5.1 ±3.7	-3.5 ±3.0	-7.3 ±2.9	-8.4 ±1.6	-8.2 ±1.2	-6.8
	OPCM [70]	-4.3 ±1.8	-4.5 ±2.8	-6.4 ±7.1	-6.1 ±4.3	-2.9 ±0.9	-6.3 ±2.9	-4.5 ±2.8	-5.7 ±1.5	-5.1
	MINGLE (Ours)	**-0.2** ±0.2	**-0.1** ±0.4	**0.7** ±1.0	**0.6** ±1.1	**-0.2** ±1.1	**0.2** ±0.7	**0.0** ±0.5	**0.5** ±0.5	**-0.2**

consistently delivers state-of-the-art accuracy while nearly eliminating forgetting under diverse continual scenarios.

Comparison with Conventional CL. Tab. 3 evaluates two CL paradigms on the MTIL benchmark: conventional CL, where each task model is fine-tuned from its immediate predecessor; and continual merging, which fine-tunes each task model independently before fusion, eliminating inter-model dependencies and enabling flexible task ordering and model reuse. Within the merging family, MINGLE sets a new state-of-the-art, and when integrated into a sequential fine-tuning pipeline, it matches the performance of SOTA CL methods. This demonstrates both its strength as a fusion strategy and its versatility across different training regimes.

Robustness to Test-Time Distribution Shifts. Following prior work [87, 68], we evaluate MINGLE on seven corruptions (motion blur, impulse noise, Gaussian noise, pixelate, spatter, contrast, JPEG) and report results in Tab. 4. It preserves high accuracy and near-zero or even positive BWT, outperforming all baselines, whereas direct application of SOTA TTA-based merging (WEMOE, AdaMerging) in a continual setting fails without tailored designs to continual setup.

4.3 Ablation Results and Analysis

Ablation Study. We explore the contribution of each component in Tab. 5. Row 1 shows a fixed-weight merging of low-rank experts as our baseline. In Row 2, adding TTA boosts ACC substantially but at the cost of worsening BWT with more tasks. Row 3 demonstrates that freezing earlier gates curbs forgetting while retaining ACC gains. Row 4 then applies null-space constraints, yielding further BWT improvements. Finally, Row 5 presents the full method with adaptive relaxation, which best harmonizes accuracy and long-term stability.

Table 5: Ablation study of MINGLE with CLIP ViT-B/16 over 8, 14, and 20 tasks.

Test-Time Adaptation	Frozen Old Gate	Null-Space Constrained Gate	Adaptive Relaxation	ACC(%) ↑			BWT(%) ↑		
				8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
✗	–	–	–	78.7 ±0.1	76.4 ±1.0	70.6 ±0.4	-0.5 ±0.1	-1.0 ±0.3	-1.3 ±0.3
✓	✗	✗	✗	86.4 ±5.3	81.7 ±2.3	76.7 ±1.3	-6.0 ±5.2	-7.7 ±-3.4	-12.8 ±1.3
✓	✓	✗	✗	87.4 ±0.4	81.3 ±0.8	76.2 ±1.3	-2.3 ±0.5	-4.3 ±0.8	-6.8 ±0.9
✓	✓	✓	✗	86.0 ±1.5	83.5 ±0.9	78.3 ±1.7	-0.1 ±0.1	-0.1 ±0.1	-0.2 ±0.1
✓	✓	✓	✓	88.3 ±0.6	84.9 ±0.8	81.9 ±0.9	-0.4 ±0.1	-0.9 ±0.1	-1.9 ±0.4

Table 6: Ablation on number of adaptation steps for ViT-B/32 across 8, 14, and 20 tasks.

Steps	ACC (8-task)	BWT (8-task)	ACC (14-task)	BWT (14-task)	ACC (20-task)	BWT (20-task)
5	60.9±1.4	-0.1±0.2	62.4±1.7	-0.3±0.1	60.2±1.5	-0.1±0.2
10	68.6±1.6	-0.2±0.2	68.5±1.6	-0.1±0.2	63.5±1.0	-0.4±0.3
20	78.4±0.6	-0.2±0.1	75.5±1.3	-0.4±0.1	71.0±1.2	-0.4±0.4
50	85.8±0.8	-0.6±0.4	81.6±1.4	-1.1±0.3	77.1±2.0	-2.2±0.8

Table 7: Efficiency and sample analysis. (a) Expert insertion layers and rank sweep over 8 tasks on CLIP-ViT-B/32. (b) Wall-clock adaptation time across tasks on CLIP-ViT-B/32. (c) Accuracy (%) of CLIP-ViT-B/16 under varying numbers of test samples.

(a) Expert insertion layers and rank sweep (8 tasks)

Configuration	TTA Time	Train. Param	Full Param	ACC(%)
`attn.qkv_proj` ($r = 64$)	61 s	27.7 k	116.0 M	69.9
`attn.out_proj` ($r = 64$)	47 s	9.3 k	97.0 M	53.9
`mlp.fc1` ($r = 64$)	48 s	9.0 k	111.1 M	82.6
`mlp.fc2` ($r = 64$)	48 s	36.8 k	111.3 M	70.1
`attn & mlp` ($r = 64$)	78 s	83.0 k	173.1 M	85.8
`qkv & fc1` ($r = 32$)	65 s	36.9 k	113.7 M	84.5
`qkv & fc1` ($r = 128$)	65 s	36.9 k	191.6 M	85.1
`qkv & fc1` ($r = 768$)	70 s	36.9 k	710.3 M	83.7
`qkv & fc1` ($r = 64$)	65 s	36.9 k	139.7 M	85.0

(b) Wall-clock adaptation time across tasks

#Tasks	Adaptation steps	Total Time (s)	Avg./task (s)
8	50	78	9.8
14	50	138	9.9
20	50	211	10.6

(c) Results with varying samples/class

# Samples/class	8-task	14-task	20-task
0 (Static)	78.7±0.1	76.4±1.0	70.6±0.4
1	88.4±0.4	84.6±1.1	81.7±1.9
3	88.6±0.4	84.7±1.0	81.7±1.0
5	88.3±0.7	84.9±0.8	81.9±0.9
10	88.6±0.3	85.1±1.1	82.1±1.2

Ablation on TTA optimization steps. Tab. 6 reports accuracy and backward transfer of CLIP ViT-B/32 under different adaptation steps across 8, 14, and 20 tasks. Accuracy improves steadily with longer schedules, rising from 60–63% at 5 steps to 77–86% at 50 steps. Forgetting remains negligible, with BWT close to zero in all cases and only a minor drop of about 2% in the 20-task setting at 50 steps. Notably, as few as 20 steps are sufficient to surpass all baselines in Tab. 1, while 50 steps yield the best accuracy with only a modest increase in cost. These results confirm that MINGLE is both effective under tight compute budgets and scalable with additional adaptation.

Computation and Parameter Efficiency. Tab. 7 summarizes the efficiency analysis. In part (a), inserting experts into both `attn` and `mlp` layers yields the highest accuracy, 85.8%, but also the longest adaptation time of 78s and 83k trainable parameters. A lighter hybrid scheme `qkv & fc1` with rank 64 reaches 85.0% accuracy with 36.9k parameters and 65s, offering a better trade-off. The rank sweep shows that raising the rank from 32 to 64 improves accuracy from 84.5% to 85.0%, while larger ranks bring little or even negative gain, *e.g.*, rank 768 drops to 83.7%. Part (b) reports wall-clock adaptation time as tasks increase: with 20 tasks the total is 211s, averaging about 10s per task. After adaptation the router remains fixed and inference is purely feedforward without TTA, enabling low-latency deployment across all tasks. Overall, MINGLE achieves a strong balance of accuracy, parameter efficiency, and scalability under diverse resource budgets.

Number of Seed Samples. The number of seed samples per class is crucial for TTA reliability and efficiency. As shown in Tab. 7 (c), using no samples reduces to the *Static* baseline, where LoRA modules are merged with fixed coefficients (0.3) without adaptation, yielding 70–79% accuracy. Introducing a single sample lifts accuracy to 81–88%, and adding more samples offers only minor gains. Variance across task orders decreases with more samples, making five samples per class a balanced trade-off between performance and efficiency.

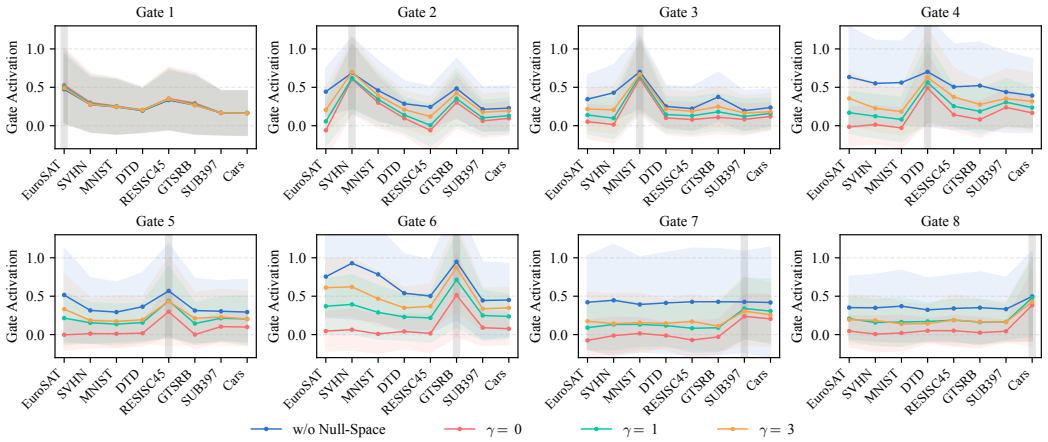
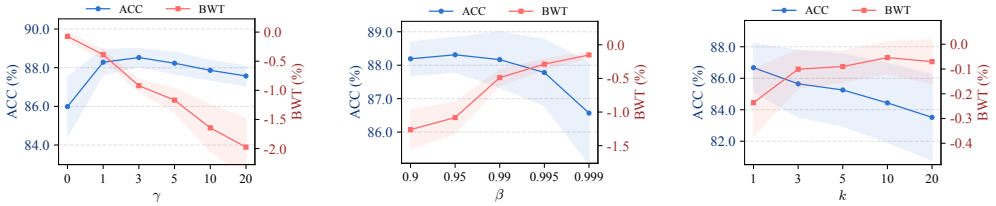


Figure 3: Gate activations across eight tasks under varying γ. Each subplot shows one gate; curves and shaded areas indicate mean and std across layers. Gray bars mark the gate's training task. Lower γ leads to stronger suppression on prior tasks.



(a) Vary γ, $\beta = 0.99$, $k = 3$ (b) Vary β, $\gamma = 1$, $k = 3$ (c) Vary k, $\gamma = 0$

Figure 4: Sensitivity analysis of the null-space constrained gating *w.r.t.* hyper-parameters β, γ, and k.

Visualization on Gate Activations. Fig. 3 shows that the null-space constraint suppresses gate responses on previously seen tasks, reducing forgetting, with smaller γ giving stronger attenuation. We also observe that gate activations remain below 1.0 across tasks, even in the w/o Null-Space variant (blue curve), showing that under-activation is not solely due to the constraint. Instead, it reflects the complementary nature of experts: multiple LoRA modules capture overlapping but distinct subspaces, and softly combining them often yields better performance, especially in settings like TTCMM where task boundaries are fuzzy.

Hyper-parameter of Gate. We study the effect of three key hyper-parameters: γ, β, and k. γ controls the strength of null-space suppression; smaller values lead to stronger attenuation of activations on prior tasks, reducing forgetting (Fig. 4a). As shown in Fig. 4b, β regulates the smoothness of the EMA used to accumulate interference signals. A moderate setting ($\beta = 0.99$) balances responsiveness and stability. Smaller β amplifies noise sensitivity, while larger β slows detection of interference. k determines the number of principal directions retained per task. The mitigation of forgetting saturates at $k = 3$ (Fig. 4c), indicating that a small number of task-specific directions suffices.

5 Conclusions

In this work, we introduced the task of test-time continual model merging (TTCMM) and proposed MINGLE, a novel framework for TTCMM that integrates a mixture-of-experts architecture with adaptive null-space constrained gating. Extensive empirical evaluations show that MINGLE substantially improves generalization and mitigates catastrophic forgetting, consistently outperforming prior state-of-the-art approaches. These results establish TTCMM as a principled paradigm for addressing both task interference and distribution shift, and highlight the practical potential of MINGLE for scalable and efficient continual learning in real-world applications.

Acknowledgments This work was supported in part by National Science and Technology Major Project (2021ZD0112001), National Natural Science Foundation of China (No.62271119, 08120002, 62071086, U23A20286), the Key Research and Development Project of Hainan Province (Grant No. ZDYF2024(LALH)003), the Fundamental Research Funds for the Central University of China (DUT No. 82232031), the Natural Science Foundation of Sichuan Province under Grant 2025ZNSFSC0475.

We thank all reviewers for taking the time to review our paper and give valuable suggestions.

References

[1] S. K. Ainsworth, J. Hayase, and S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

[2] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[3] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014.

[4] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[5] R. Chitale, A. Vaidya, A. Kane, and A. S. Ghotkar. Task arithmetic with loRA for continual learning. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@NeurIPS 2023)*, 2023.

[6] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[8] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. In *NeurIPS Workshop on Machine Learning for Creativity and Design*, 2018.

[9] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[10] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[11] M. Döbler, R. A. Marsden, and B. Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7704–7714, 2023.

[12] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. *European Conference on Computer Vision*, 2020.

[13] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

[14] C. Fang, A. Dziedzic, L. Zhang, L. Oliva, A. Verma, F. Razak, N. Papernot, and B. Wang. Decentralised, collaborative, and privacy-preserving machine learning for multi-hospital data. *EBioMedicine*, 101, 2024.

[15] C.-M. Feng, K. Yu, Y. Liu, S. Khan, and W. Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023.

[16] T. Fukuda, H. Kera, and K. Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. *arXiv preprint arXiv:2412.18219*, 2024.

[17] J. Gao, J. Zhang, X. Liu, T. Darrell, E. Shelhamer, and D. Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11786–11796, 2023.

[18] I. J. Goodfellow, D. Erhan, P.-L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *International Conference on Neural Information Processing*, pages 117–124. Springer, 2013.

[19] P. Helber, B. Bischke, A. Dengel, and D. 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.

[20] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Learning a unified classifier incrementally via rebalancing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[22] C. Huang, Q. Liu, B. Y. Lin, T. Pang, C. Du, and M. Lin. Lorahub: Efficient cross-task generalization via dynamic loRA composition. In *First Conference on Language Modeling*, 2024.

[23] L. Huang, X. Cao, H. Lu, and X. Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pages 214–231. Springer, 2024.

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

[25] Y. Iwasawa and Y. Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[26] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

[27] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

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

[29] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[30] S. Jung, H. Ahn, S. Cha, and T. Moon. Continual learning with node-importance based adaptive group sparse regularization. *Advances in neural information processing systems*, 33:3647–3658, 2020.

[31] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016.

[32] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561. IEEE, 2013.

[33] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems*, 30, 2017.

[36] Y. Lee, D. Kim, J. Kang, J. Bang, H. Song, and J.-G. Lee. RA-TTA: Retrieval-augmented test-time adaptation for vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[37] S. Lin, L. Yang, D. Fan, and J. Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177, 2022.

[38] T. Y. Liu and S. Soatto. Tangent model composition for ensembling and continual fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18676–18686, 2023.

[39] Y. Liu, B. Schiele, and Q. Sun. Rmm: Reinforced memory management for class-incremental learning. *Advances in Neural Information Processing Systems*, 34:3478–3490, 2021.

[40] Z. Lu, C. Fan, W. Wei, X. Qu, D. Chen, and Y. Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37: 78905–78935, 2024.

[41] D. Marczak, B. Twardowski, T. Trzciński, and S. Cygert. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision (ECCV)*, 2024.

[42] I. E. Marouf, S. Roy, E. Tartaglione, and S. Lathuilière. Weighted ensemble models are strong continual learners, 2024.

[43] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[44] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[45] S. Mu and S. Lin. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*, 2025.

[46] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[47] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[48] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

[49] G. Ortiz-Jimenez, A. Favero, and P. Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023.

[50] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.

[51] A. Porrello, L. Bonicelli, P. Buzzega, M. Millunzi, S. Calderara, and R. Cucchiara. A second-order perspective on model compositionality and incremental learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] Z. Qiu, L. Xu, Z. Wang, Q. Wu, F. Meng, and H. Li. Ism-net: Mining incremental semantics for class incremental learning. *Neurocomputing*, 523:130–143, 2023.

[53] Z. Qiu, Y. Xu, F. Meng, H. Li, L. Xu, and Q. Wu. Dual-consistency model inversion for non-exemplar class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24025–24035, 2024.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[55] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2016.

[56] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020.

[57] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

[58] K. Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[59] M. Shu, W. Nie, D.-A. Huang, Z. Yu, T. Goldstein, A. Anandkumar, and C. Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

[60] C. Simon, P. Koniusz, and M. Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021.

[61] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

[62] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[63] J. Song, J. Lee, I. S. Kweon, and S. Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023.

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

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

[66] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[67] A. Tang, L. Shen, Y. Luo, H. Hu, B. Du, and D. Tao. FusionBench: A Comprehensive Benchmark of Deep Model Fusion, June 2024.

[68] A. Tang, L. Shen, Y. Luo, N. Yin, L. Zhang, and D. Tao. Merging multi-task models via weight-ensembling mixture of experts. In *International Conference on Machine Learning*, pages 47778–47799. PMLR, 2024.

[69] A. Tang, L. Shen, Y. Luo, Y. Zhan, H. Hu, B. Du, Y. Chen, and D. Tao. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*, 2024.

[70] A. Tang, E. Yang, L. Shen, Y. Luo, H. Hu, B. Du, and D. Tao. Merging models on the fly without retraining: A sequential approach to scalable continual model merging. *arXiv preprint arXiv:2501.09522*, 2025.

[71] L. Tang, Z. Tian, K. Li, C. He, H. Zhou, H. Zhao, X. Li, and J. Jia. Mind the interference: Retaining pre-trained knowledge in parameter efficient continual learning of vision-language models. In *European Conference on Computer Vision*, pages 346–365. Springer, 2024.

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

[73] B. S. Veeling, J. Linmans, J. Winkens, T. S. Cohen, and M. Welling. Rotation equivariant cnns for digital pathology. *arXiv preprint arXiv:1806.03962*, 2018.

[74] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[75] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[76] H. Wang, B. Ping, S. Wang, X. Han, Y. Chen, Z. Liu, and M. Sun. Lora-flow: Dynamic lora fusion for large language models in generative tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12871–12882, 2024.

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

[78] X. Wang, T. Chen, Q. Ge, H. Xia, R. Bao, R. Zheng, Q. Zhang, T. Gui, and X. Huang. Orthogonal subspace learning for language model continual learning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[79] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[80] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022.

[81] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[82] Y. Wu, L.-K. Huang, R. Wang, D. Meng, and Y. Wei. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth International Conference on Learning Representations*, 2024.

[83] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[84] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

[85] Y. Xu, Y. Chen, J. Nie, Y. Wang, H. Zhuang, and M. Okumura. Advancing cross-domain discriminability in continual learning of vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[86] P. Yadav, D. Tam, L. Choshen, C. A. Raffel, and M. Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36: 7093–7115, 2023.

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

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

[89] J. Yu, Y. Zhuge, L. Zhang, P. Hu, D. Wang, H. Lu, and Y. He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230, 2024.

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

[91] L. Yuan, B. Xie, and S. Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

[92] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. *International conference on machine learning*, pages 3987–3995, 2017.

[93] M. Zhang, S. Levine, and C. Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[94] Z. Zhao, L. Gan, G. Wang, W. Zhou, H. Yang, K. Kuang, and F. Wu. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4447–4462, 2024.

[95] Z. Zhao, T. Shen, D. Zhu, Z. Li, J. Su, X. Wang, and F. Wu. Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

[96] M. Zheng, Y. Tang, Z. Hao, K. Han, Y. Wang, and C. Xu. Adapt without forgetting: Distill proximity from dual teachers in vision-language models. In *European Conference on Computer Vision*, pages 109–125. Springer, 2024.

[97] Z. Zheng, M. Ma, K. Wang, Z. Qin, X. Yue, and Y. You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19125–19136, 2023.

[98] D.-W. Zhou, H.-L. Sun, H.-J. Ye, and D.-C. Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions, which are consistently supported by both theoretical insights and empirical evaluations presented throughout the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed approach are explicitly discussed in Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical claims are accompanied by clear assumptions and complete proofs in Appendix, with theorem formally stated and rigorously derived.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed descriptions of datasets, training protocols, and evaluation metrics are provided in Section 4.1 and Appendix, enabling faithful reproduction of the main results.

   Guidelines:

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

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

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

Answer: [No]

Justification:The code and data are not provided at submission time, but the authors state they will release them upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper includes detailed descriptions of datasets, data splits, training procedures, hyperparameters, and evaluation protocols in both the main text and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports mean and standard deviation across multiple runs with different random seeds and clearly states the sources of variability, ensuring the statistical reliability of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the type of GPUs used, training time per experiment, and overall compute requirements, providing sufficient details to assess reproducibility and resource demands.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics, with no identified ethical concerns related to data usage, human subjects, or societal impact.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a Broader Impacts section stating that the work aims to advance machine learning and does not raise specific societal concerns requiring detailed discussion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce models or datasets with a high risk of misuse, so safeguards are not applicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models used in the paper are properly cited, and their licenses and terms of use are respected and included where applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release any new datasets, models, or code assets, so this question is not applicable.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so this question is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core methods in this research do not involve LLMs in any important, original, or non-standard way, so this question is not applicable.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Theoretical Risk Comparison: Dynamic MoE vs. Static Averaging

Problem Setup and Definitions Consider T independent tasks, each associated with a data distribution D_t for $t = 1, \ldots, T$. For each task t, a pre-trained expert model $f_t(x)$ outputs a probability distribution over classes, trained specifically on D_t. The overall data distribution D is a mixture of these tasks, where an example (x, y) is drawn from task t with prior probability $P(t)$, and then $(x, y) \sim D_t$. The expected risk of a predictive model $h(x)$ is defined as:

$$R(h) = \mathbb{E}_{(x,y)\sim D}\big[\ell(h(x), y)\big] = \sum_{t=1}^{T} P(t)\, \mathbb{E}_{(x,y)\sim D_t}\big[\ell(h(x), y)\big], \tag{12}$$

where $\ell(h(x), y)$ is a loss function (*e.g.*, cross-entropy or 0-1 loss).

We compare two methods to combine the experts into a final prediction $h(x)$:

- **Static Averaging**: Defined as $h_{\text{static}}(x) = \sum_{i=1}^{T} \alpha_i f_i(x)$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_T)$ is a fixed weight vector independent of x, typically with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$ for probability outputs.
- **Dynamic Mixture-of-Experts (MoE)**: Defined as $h_{\text{MoE}}(x) = f_{i^*(x)}(x)$, where $i^*(x) = \arg\max_i g_i(x)$ and $g(x) = (g_1(x), \ldots, g_T(x))$ is a gating function that selects one expert per input (hard routing). The gating is subject to routing noise, modeled below.

Routing Noise Model For each input x drawn from D_t, the true task is t, and the ideal expert is f_t. The gating selects the correct expert $i^*(x) = t$ with probability $1 - \varepsilon_t$, and an incorrect expert $i^*(x) \neq t$ with probability $\varepsilon_t = P(i^*(x) \neq t \mid x \sim D_t)$, the task-specific routing error rate. On error, the gating selects a random expert from $\{1, \ldots, T\} \setminus \{t\}$ uniformly. Define:

- $R_t(i) = \mathbb{E}_{(x,y)\sim D_t}[\ell(f_i(x), y)]$, the risk of expert i on task t.
- $R_{\text{ideal}} = \sum_{t=1}^{T} P(t) R_t(t)$, the risk with perfect routing.
- $R_{\text{wrong},t} = \frac{1}{T-1} \sum_{i \neq t} R_t(i)$, the average risk of incorrect experts on task t.
- $\varepsilon = \sum_{t=1}^{T} P(t)\varepsilon_t$, the overall routing error rate.

Theorem A.1 (Dynamic MoE versus Static Averaging)**.** *Let $\{(D_t, f_t)\}_{t=1}^T$ be T independent tasks with priors $P(t)$ and per-task risks $R_t(i)$. For any static mixture $h_{\text{static}}(x) = \sum_{i=1}^T \alpha_i f_i(x)$ and any hard-routed MoE $h_{\text{MoE}}(x) = f_{i^*(x)}(x)$ with task-specific routing errors ε_t:*

$$R(h_{\text{MoE}}) \;=\; R_{\text{ideal}} + \sum_{t=1}^{T} P(t)\,\varepsilon_t\big(R_{\textit{wrong},t} - R_t(t)\big), \tag{13}$$

where $R_{\text{ideal}} = \sum_t P(t)R_t(t)$ and $R_{\textit{wrong},t} = \frac{1}{T-1}\sum_{i\neq t} R_t(i)$. Moreover,

1. (Perfect routing) *If $\varepsilon_t = 0$ for all t, then $\inf_g R(h_{\text{MoE}}) \;<\; \inf_{\boldsymbol{\alpha}} R(h_{\text{static}})$ whenever at least two tasks disagree on their best expert.*
2. (Noisy routing) *If $\sum_t P(t)\varepsilon_t\big(R_{\textit{wrong},t} - R_t(t)\big) \;<\; R^*_{\text{static}} - R_{\text{ideal}}$, where $R^*_{\textit{static}} = \inf_{\boldsymbol{\alpha}} R(h_{\textit{static}})$, then the MoE still attains lower risk than any static mixture.*

Proof. The proof proceeds in three parts: (1) deriving the MoE risk with routing noise, (2) proving the optimal gating case, and (3) establishing the condition for MoE superiority under routing noise.

Step 1: MoE Risk with Routing Noise

The MoE prediction is $h_{\text{MoE}}(x) = f_{i^*(x)}(x)$, where $i^*(x) = \arg\max_i g_i(x)$. The expected risk is:

$$R(h_{\text{MoE}}) = \mathbb{E}_{(x,y)\sim D}[\ell(f_{i^*(x)}(x), y)] = \sum_{t=1}^{T} P(t)\,\mathbb{E}_{(x,y)\sim D_t}[\ell(f_{i^*(x)}(x), y)]. \tag{14}$$

For task t, condition on routing correctness:

- **Correct routing** ($i^*(x) = t$): Probability $1 - \varepsilon_t$, risk $R_t(t)$.
- **Incorrect routing** ($i^*(x) \neq t$): Probability ε_t, selects a random expert from $\{1, \ldots, T\} \setminus \{t\}$, with average risk $R_{\text{wrong},t} = \frac{1}{T-1}\sum_{i\neq t} R_t(i)$.

The expected risk on task t is:

$$\mathbb{E}_{(x,y)\sim D_t}[\ell(f_{i^*(x)}(x), y)] = (1-\varepsilon_t)R_t(t) + \varepsilon_t R_{\text{wrong},t}. \tag{15}$$

Thus, the total risk is:

$$R(h_{\text{MoE}}) = \sum_{t=1}^{T} P(t)\big[(1-\varepsilon_t)R_t(t) + \varepsilon_t R_{\text{wrong},t}\big]. \tag{16}$$

Rewrite:

$$R(h_{\text{MoE}}) = \sum_{t=1}^{T} P(t)R_t(t) + \sum_{t=1}^{T} P(t)\varepsilon_t(R_{\text{wrong},t} - R_t(t)) = R_{\text{ideal}} + \sum_{t=1}^{T} P(t)\varepsilon_t\delta_t, \tag{17}$$

where $\delta_t = R_{\text{wrong},t} - R_t(t) > 0$ is the risk increase due to misrouting on task t.

Step 2: Optimal Gating (No Routing Noise)

Assume an oracle gating function with $\varepsilon_t = 0$ for all t, so $i^*(x) = t$ for all $x \sim D_t$. Then:

$$R(h_{\text{MoE}}) = \sum_{t=1}^{T} P(t)R_t(t) = R_{\text{ideal}}. \tag{18}$$

Define hypothesis classes:

$$\mathcal{H}_{\text{static}} = \left\{ x \mapsto \sum_{i=1}^{T} \alpha_i f_i(x) \mid \boldsymbol{\alpha} \in \mathbb{R}^T \right\}, \tag{19}$$

$$\mathcal{H}_{\text{MoE}} = \left\{ x \mapsto f_{i^*(x)}(x) \mid i^*(x) = \arg\max_i g_i(x), g : \mathcal{X} \to \mathbb{R}^T \right\}. \tag{20}$$

Any static model $h_{\text{static}}(x) = \sum_{i=1}^{T} \alpha_i f_i(x)$ can be approximated by an MoE with $g(x)$ assigning constant weights, so $\mathcal{H}_{\text{static}} \subseteq \mathcal{H}_{\text{MoE}}$. Thus:

$$R^*_{\text{MoE}} = \inf_{g(\cdot)} R(h_{\text{MoE}}) \leq \inf_{\boldsymbol{\alpha}} R(h_{\text{static}}) = R^*_{\text{static}}. \tag{21}$$

Under task heterogeneity ($R_t(t) < R_t(s)$ and $R_s(s) < R_s(t)$ for some $t \neq s$), the ideal MoE routes each $x \sim D_t$ to f_t, achieving:

$$R_{\text{ideal}} = \sum_{t=1}^{T} P(t) R_t(t). \tag{22}$$

For static averaging:

$$R(h_{\text{static}}) = \sum_{t=1}^{T} P(t) \mathbb{E}_{(x,y)\sim D_t} \left[\ell \left(\sum_{i=1}^{T} \alpha_i f_i(x), y \right) \right]. \tag{23}$$

Since ℓ is convex (*e.g.*, cross-entropy), Jensen's inequality implies:

$$\mathbb{E}_{D_t} \left[\ell \left(\sum_{i} \alpha_i f_i(x), y \right) \right] \geq R_t(t), \tag{24}$$

with strict inequality unless $\alpha_t = 1$ and $\alpha_i = 0$ for $i \neq t$, which cannot hold for all tasks simultaneously under heterogeneity. Thus:

$$R^*_{\text{static}} > R_{\text{ideal}} = R^*_{\text{MoE}}. \tag{25}$$

Step 3: MoE Superiority with Routing Noise

Let $\gamma = R^*_{\text{static}} - R_{\text{ideal}} > 0$ under task heterogeneity. The MoE outperforms the static model if:

$$R(h_{\text{MoE}}) < R^*_{\text{static}}. \tag{26}$$

Substitute:

$$R_{\text{ideal}} + \sum_{t=1}^{T} P(t) \varepsilon_t \delta_t < R_{\text{ideal}} + \gamma. \tag{27}$$

Thus:

$$\sum_{t=1}^{T} P(t) \varepsilon_t \delta_t < \gamma = R^*_{\text{static}} - R_{\text{ideal}}. \tag{28}$$

Since $\delta_t = R_{\text{wrong},t} - R_t(t)$, the condition is:

$$\sum_{t=1}^{T} P(t) \varepsilon_t (R_{\text{wrong},t} - R_t(t)) < R^*_{\text{static}} - R_{\text{ideal}}. \tag{29}$$

If this holds, the MoE's risk, despite routing noise, remains below the best static risk. □

Conclusion: The MoE risk is $R_{\text{ideal}} + \sum_{t=1}^{T} P(t) \varepsilon_t (R_{\text{wrong},t} - R_t(t))$, and it outperforms static averaging when routing noise is sufficiently small relative to the static model's suboptimality. The optimal gating case confirms $R^*_{\text{MoE}} \leq R^*_{\text{static}}$, with strict inequality under task heterogeneity.

B Additional Descriptions

B.1 Details of Dataset and Task Settings

Dataset Details Following prior works [70], we evaluate continual model merging on twenty publicly available image classification datasets, including SUN397 [84], Stanford Cars [32], RESISC45 [4], EuroSAT [19], SVHN [46], GTSRB [64], MNIST [34], DTD [7], Flowers102 [47], PCAM [73], FER2013 [18], Oxford-IIIT Pet [50], STL-10 [9], CIFAR-100 and CIFAR-10 [33], Food-101 [3], Fashion-MNIST [83], EMNIST [10], KMNIST [8], and Rendered SST-2 [62].

Table 8: Extended downstream datasets used in our experiments.

Dataset	#Classes	#Train (k)	#Test (k)	Task
SUN397	287	19.9	19.9	Scene category
Stanford Cars	196	8.1	8.0	Car series
RESISC45	45	18.9	6.3	Remote–sensing scene
EuroSAT	10	21.6	2.7	Satellite land-use
SVHN	10	73.3	26.0	Digit recognition
GTSRB	43	39.2	12.6	Traffic sign
MNIST	10	60	10	Hand-written digit
DTD	47	3.8	1.9	Texture recognition
Flowers102	102	1.0	6.1	Flower species
PCAM	2	262	32.8	Tumour detection
FER2013	7	28.7	3.6	Facial emotion
Oxford IIIT Pet	37	3.7	3.7	Animal species
STL10	10	5	8	Object recognition
CIFAR-100	100	50	10	Natural object
CIFAR-10	10	50	10	Natural object
Food101	101	75.8	25.3	Food type
Fashion-MNIST	10	60	10	Fashion product
EMNIST (digits)	10	60	10	Hand-written digit
KMNIST	10	60	10	Kuzushiji character
Rendered SST-2	2	6.9	1.8	Rendered sentiment

Task Grouping We group the 20 datasets into three progressive task sets and evaluate the merged models using average accuracy (ACC) and backward transfer (BWT) metrics. For each task group, we perform 10 experiments using different task sequences (listed in Tab. 9), and report both the mean and standard deviation of the results to ensure robustness and consistency.

- **8-task group**: (1) SUN397, (2) Stanford Cars, (3) RESISC45, (4) EuroSAT, (5) SVHN, (6) GTSRB, (7) MNIST, (8) DTD.
- **14-task group**: (1) SUN397, (2) Stanford Cars, (3) RESISC45, (4) EuroSAT, (5) SVHN, (6) GTSRB, (7) MNIST, (8) DTD, (9) Flowers102, (10) PCAM, (11) FER2013, (12) OxfordIIITPet, (13) STL10, (14) CIFAR100.
- **20-task group**: (1) SUN397, (2) Stanford Cars, (3) RESISC45, (4) EuroSAT, (5) SVHN, (6) GTSRB, (7) MNIST, (8) DTD, (9) Flowers102, (10) PCAM, (11) FER2013, (12) OxfordIIITPet, (13) STL10, (14) CIFAR100, (15) CIFAR10, (16) Food101, (17) FashionMNIST, (18) EMNIST, (19) KMNIST, (20) RenderedSST2.

B.2 Details of Downstream Models

In this section, we present the evaluation setup for pre-trained and fine-tuned models. As shown in Tab. 10, we evaluate the zero-shot accuracy of the original CLIP-ViT models and the performance of fine-tuned models on the test sets of various downstream tasks. The fine-tuned checkpoints are obtained directly from Hugging Face (`https://huggingface.co/tanganke`), where each model has been fine-tuned on task-specific training data using a standard protocol. The visual encoder is updated during fine-tuning, while the classification head is fixed and initialized from the pre-trained text encoder. The fine-tuning setup follows a standard configuration: cross-entropy loss, Adam optimizer, cosine annealing learning rate schedule with a peak learning rate of 1e-5, batch size 128, and 4000 training steps.

B.3 Details of Baselines

Our experiments involve the following comparison methods and our method:

- **Stochastic Weight Averaging (SWA).** A simple model averaging technique to stabilize optimization and improve generalization [26]. At each step t, the model parameters are averaged across previous checkpoint: $\theta_t^{\text{SWA}} = \frac{1}{t}\left[\theta_{t-1}^{\text{SWA}}(t-1) + \theta_t^{\text{SWA}}\right]$. This approach can be interpreted

Table 9: Dataset orderings used for experiments in each task group.

Group	Order	Dataset Order (by ID)
8 tasks	1	(04 → 05 → 07 → 08 → 03 → 06 → 01 → 02)
	2	(07 → 08 → 05 → 04 → 02 → 06 → 03 → 01)
	3	(03 → 06 → 04 → 02 → 01 → 08 → 05 → 07)
	4	(06 → 08 → 02 → 01 → 03 → 07 → 04 → 05)
	5	(07 → 06 → 03 → 08 → 05 → 01 → 04 → 02)
	6	(07 → 02 → 03 → 08 → 05 → 04 → 01 → 06)
	7	(07 → 01 → 04 → 03 → 08 → 05 → 02 → 06)
	8	(08 → 05 → 06 → 07 → 01 → 04 → 03 → 02)
	9	(01 → 04 → 05 → 02 → 06 → 03 → 07 → 08)
	10	(08 → 03 → 01 → 02 → 06 → 05 → 07 → 04)
14 tasks	1	(09 → 13 → 08 → 07 → 14 → 12 → 06 → 03 → 10 → 04 → 05 → 01 → 02 → 11)
	2	(09 → 10 → 11 → 14 → 07 → 13 → 04 → 02 → 06 → 08 → 03 → 12 → 05 → 01)
	3	(05 → 08 → 12 → 06 → 11 → 01 → 10 → 04 → 14 → 03 → 02 → 13 → 09 → 07)
	4	(03 → 10 → 09 → 12 → 04 → 13 → 01 → 06 → 11 → 02 → 14 → 08 → 07 → 05)
	5	(08 → 14 → 09 → 06 → 12 → 13 → 05 → 03 → 04 → 11 → 10 → 01 → 07 → 02)
	6	(03 → 12 → 13 → 01 → 11 → 04 → 10 → 05 → 14 → 08 → 09 → 07 → 02 → 06)
	7	(07 → 01 → 12 → 10 → 02 → 08 → 13 → 04 → 05 → 11 → 14 → 03 → 06 → 09)
	8	(05 → 12 → 04 → 11 → 03 → 08 → 10 → 01 → 09 → 13 → 14 → 07 → 06 → 02)
	9	(10 → 07 → 09 → 02 → 03 → 13 → 01 → 12 → 14 → 04 → 11 → 06 → 05 → 08)
	10	(01 → 02 → 11 → 06 → 08 → 12 → 07 → 05 → 10 → 14 → 03 → 13 → 09 → 04)
20 tasks	1	(20 → 06 → 15 → 05 → 10 → 14 → 16 → 19 → 07 → 13 → 18 → 11 → 02 → 12 → 03 → 17 → 08 → 09 → 01 → 04)
	2	(09 → 14 → 06 → 03 → 07 → 04 → 18 → 01 → 17 → 19 → 08 → 20 → 13 → 16 → 11 → 12 → 15 → 05 → 10 → 02)
	3	(09 → 15 → 16 → 11 → 03 → 13 → 08 → 10 → 12 → 02 → 20 → 01 → 05 → 19 → 07 → 06 → 04 → 18 → 17 → 14)
	4	(17 → 04 → 11 → 19 → 18 → 10 → 07 → 15 → 12 → 13 → 08 → 02 → 01 → 06 → 05 → 03 → 20 → 16 → 14 → 09)
	5	(14 → 16 → 04 → 20 → 15 → 17 → 07 → 11 → 06 → 18 → 12 → 01 → 19 → 09 → 10 → 05 → 08 → 02 → 13 → 03)
	6	(02 → 06 → 17 → 04 → 19 → 18 → 08 → 16 → 20 → 01 → 10 → 13 → 07 → 09 → 05 → 11 → 15 → 14 → 03 → 12)
	7	(19 → 01 → 09 → 14 → 06 → 20 → 17 → 04 → 08 → 02 → 15 → 03 → 16 → 13 → 12 → 07 → 10 → 05 → 11 → 18)
	8	(15 → 07 → 08 → 02 → 10 → 06 → 17 → 20 → 05 → 19 → 16 → 01 → 18 → 09 → 13 → 11 → 04 → 14 → 12 → 03)
	9	(10 → 05 → 07 → 11 → 01 → 03 → 17 → 15 → 18 → 04 → 14 → 19 → 02 → 06 → 13 → 20 → 08 → 12 → 09 → 16)
	10	(01 → 11 → 02 → 15 → 03 → 10 → 12 → 19 → 16 → 13 → 07 → 05 → 09 → 04 → 14 → 20 → 06 → 18 → 17 → 08)

as a form of uniform model ensembling. While conceptually straightforward, SWA treats all checkpoints equally and does not account for inter-task conflicts.

- **Continual Task Arithmetic (C. TA).** A training-free merging strategy that linearly combines task-specific fine-tuned models with a shared pre-trained model [24]. It computes the merged parameters as $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + \lambda(\theta_t - \theta_0)$, where λ is a scaling factor. TA is computationally efficient and easy to apply, but sensitive to λ and prone to destructive interference when merging dissimilar tasks.

- **Continual Ties-Merging (C. Ties).** An extension of Task Arithmetic that reduces parameter-level redundancy and sign conflicts during model merging [86]. For task t, the difference vector $\Delta\theta_t = \theta_t - \theta_0$ is trimmed and sign-normalized to obtain $\Delta\theta_t^{\text{Ties}} = \text{Ties}\big(\Delta\theta_{t-1}^{\text{Ties}}, \Delta\theta_t\big)$, and the merged model is given by $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + \lambda\Delta\theta_t^{\text{Ties}}$.

- **Orthogonal Projection-based Continual Merging (OPCM).** A projection-based scheme to mitigate task interference by enforcing orthogonality between parameter updates [70]. Specifically, each $\Delta\theta_t$ is projected onto the orthogonal complement of the subspace spanned by previous updates: $\theta_t^{\text{merged}} = \theta_0 + \frac{1}{\lambda_t}\left[\lambda_{t-1}\Delta\theta_{t-1}^{\text{merged}} + \mathcal{P}^{(t-1)}(\Delta\theta_t)\right]$, where $\mathcal{P}^{(t-1)}$ denotes the orthogonal projection.

- **Maximum Magnitude Selection (MagMax).** An extension of Task Arithmetic that, for each parameter dimension, selects the update with the larger absolute value: $\Delta\theta_t^{\text{MagMax}} = \text{MagMax}\Big(\Delta\theta_{t-1}^{\text{MagMax}}, \Delta\theta_t\Big)$, and the merged model is given by $\theta_t^{\text{merged}} = \theta_{t-1}^{\text{merged}} + \lambda\Delta\theta_t^{\text{MagMax}}$.

Table 10: Test set accuracy of the pre-trained model and individual fine-tuned models on different downstream tasks.

Model	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers102	PCAM
CLIP-ViT-B/32										
Pre-trained	63.2	59.6	60.3	45.0	31.6	32.5	48.3	44.2	66.4	60.6
Fine-tuned	74.9	78.5	95.1	99.1	97.3	98.9	99.6	79.7	88.6	88.0
CLIP-ViT-B/16										
Pre-trained	65.5	64.7	66.4	54.1	52.0	43.5	51.7	45.0	71.3	54.0
Fine-tuned	78.9	85.9	96.6	99.0	97.6	99.0	99.7	82.3	94.9	90.6
CLIP-ViT-L/14										
Pre-trained	68.2	77.9	71.3	61.2	58.4	50.5	76.3	55.5	79.2	51.2
Fine-tuned	82.8	92.8	97.4	99.1	97.9	99.2	99.8	85.5	97.7	91.1

Model	FER2013	OxfordIIITPet	STL10	CIFAR100	CIFAR10	Food101	FashionMNIST	EMNIST	KMNIST	RenderedSST2
CLIP-ViT-B/32										
Pre-trained	41.3	83.3	97.1	63.7	89.8	82.4	63.0	12.0	10.0	58.6
Fine-tuned	71.6	92.5	97.5	88.4	97.6	88.4	94.7	95.6	98.2	71.3
CLIP-ViT-B/16										
Pre-trained	46.4	88.4	98.3	66.3	90.8	87.0	67.3	12.4	11.2	60.6
Fine-tuned	72.8	94.5	98.2	88.8	98.3	91.9	94.5	95.3	98.1	75.7
CLIP-ViT-L/14										
Pre-trained	50.0	93.2	99.4	75.1	95.6	91.2	67.0	12.3	9.7	68.9
Fine-tuned	75.9	95.7	99.2	93.0	99.1	94.8	95.3	95.4	98.3	80.5

B.4 Details of Baseline Hyper-parameters

Tab. 11 summarizes the hyper-parameters for all baseline methods under different task configurations (8, 14, 20 tasks). *Top-k* denotes the pruning ratio, *TALL* the TALL mask threshold, and *Cons.* the consensus mask threshold. The column *LR* is the learning rate, while *Steps* indicates the number of adaptation steps. r represents the LoRA rank, and the last column jointly reports the null dimension (k), EMA decay (β), and relaxation coefficient (γ).

Table 11: hyper-parameter settings for all baselines.

Method	Tasks	λ	Top-k	TALL	Cons.	LR	Steps	r	$k/\beta/\gamma$
TASK ARITHMETIC	8	0.3	-	-	-	-	-	-	-
	14/20	0.1	-	-	-	-	-	-	-
TIES-MERGING	8	0.3	20	-	-	-	-	-	-
	14/20	0.1	20	-	-	-	-	-	-
CONSENSUS TA	8/14/20	0.1	-	0.2	2	-	-	-	-
LW. ADAMERGING	8/14/20	0.3	-	-	-	1e-4	50	-	-
WEMOE	8/14/20	0.3	-	-	-	1e-4	50	-	-
LORA-WEMOE	8/14/20	0.3	-	-	-	1e-4	50	64	-
MINGLE-STATIC	8/14/20	0.3	-	-	-	-	-	-	-
MINGLE (Ours)	8/14/20	-	-	-	-	1e-4	50	64	3/0.99/1.0

B.5 Comparison of Assumptions and Requirements

Tab. 12 summarizes the assumptions and resource requirements of all baseline methods. We report whether each method requires storing intermediate activations, introduces additional parameters (for storage or inference), and incurs extra test-time computation. Our method only maintains a fixed-size covariance matrix instead of full activations, leading to constant memory regardless of test set size. Although LoRA experts are stored, the router merges them into a single model per input, so the effective inference cost matches that of a standard individual model.

Table 12: Comparison of baseline assumptions and requirements.

Method	Save Activations	Extra Parameters (Storage)	Extra Parameters (Inference)	Test-time Compute
TASK ARITHMETIC	No	No	No	No
TIES-MERGING	No	No	No	No
MAGMAX-IND	No	No	No	No
OPCM	No	No	No	No
CONSENSUS TA	No	Yes	No	No
LW. ADAMERGING	No	No	No	Yes
WEMOE	No	Yes	No	Yes
MINGLE-STATIC	No	No	No	No
MINGLE (Ours)	No[1]	Yes[2]	No[2]	Yes

[1] Only a fixed-size covariance matrix is maintained, resulting in constant memory regardless of test set size.
[2] LoRA experts are stored, but the router merges them into a single model per input, making the effective inference size equivalent to a standard individual model.

C Additional Results

In this section, we provide additional experimental results to support the findings reported in the main paper. Specifically, we include: (1) detailed overall performance results (C.1); (2) accuracy trends across sequential tasks (C.2); (3) detailed results under distribution shifts (C.3); and (4) extended visualizations of gate activations and hyper-parameter effects (C.6).

C.1 Detailed Overall Performance Results

Tab. 13 expands on the average results in Tab. 1 by reporting per-task average accuracy after continually merging 20 tasks. We compare six methods, SWA, Task Arithmetic, Ties-Merging, MagMax-IND, OPCM, and our proposed MINGLE across three CLIP-ViT backbones (B/32, B/16, L/14). MINGLE achieves the highest accuracy on most tasks. These fine-grained results reinforce the main paper's findings, highlighting MINGLE's ability to improve performance on continual model merging.

C.2 Accuracy Trends Across Sequential Tasks

Fig. 5 provides a detailed view of accuracy throughout the continual merging process across different settings, showing both the performance on the current task and on previously encountered ones. The progressive accuracy drop across columns illustrates the degree of forgetting over time. Notably, MINGLE consistently alleviates this degradation, demonstrating markedly reduced forgetting across the full task sequence. Fig. 6 further compares the average accuracy curves of MINGLE and baseline methods on previously seen tasks after each new model is merged, using the CLIP ViT-B/16 backbone. Results are averaged over 10 random task orderings. MINGLE consistently achieves the highest performance throughout the merging process, with its accuracy curve clearly dominating those of competing methods. Moreover, the narrower standard deviation bands indicate that MINGLE is more robust to the task orders.

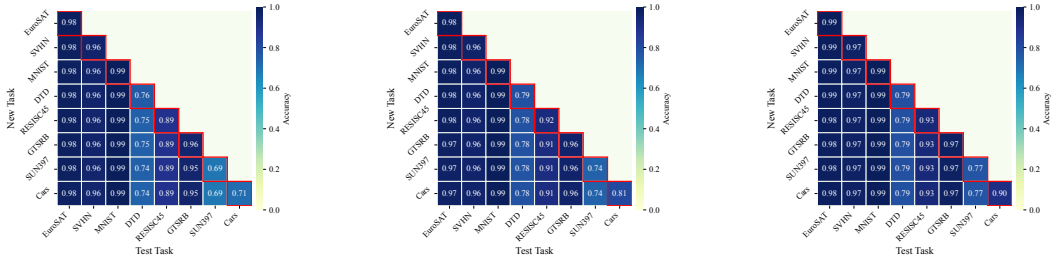
C.3 Detailed Results Under Distribution Shifts

Tab. 14 expands on Tab. 4 by reporting per-dataset accuracy under both clean test conditions and seven common corruption types: motion blur, impulse noise, Gaussian noise, pixelation, spatter, contrast shift, and JPEG compression. We evaluate six merging methods, across four downstream tasks: Cars, EuroSAT, RESISC45, and GTSRB. This detailed breakdown complements the average results in the main paper, providing a more comprehensive assessment of robustness under test-time distribution shifts.

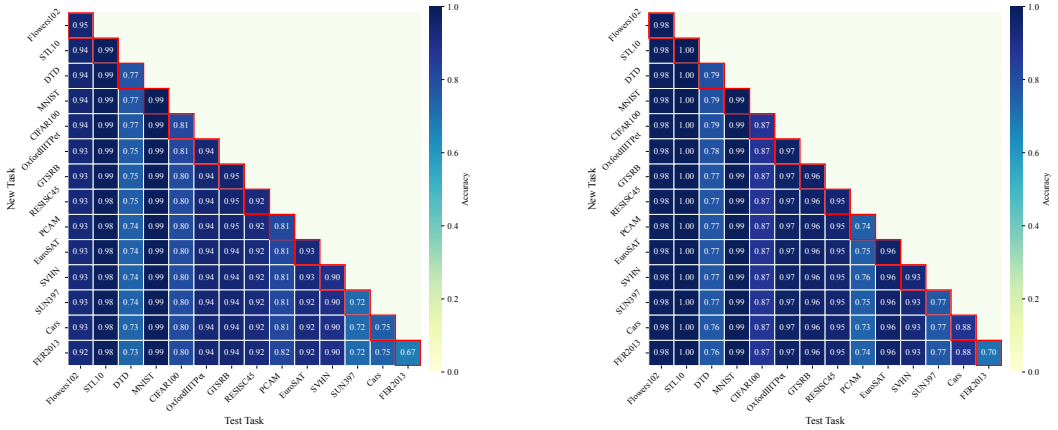
Table 13: Test set accuracy comparisons on different downstream tasks.

Model	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers102	PCAM
ViT-B/32										
C. Fine-Tuned	53.9	38.2	64.7	98.7	45.4	34.4	86.7	58.4	57.5	67.7
Average (SWA)	64.2	59.6	64.8	60.9	47.3	43.1	71.8	46.4	66.5	63.9
C.TA	62.0	53.7	60.9	58.1	48.5	48.9	79.4	46.1	61.1	73.4
C.TIES	62.5	49.1	55.8	50.9	54.6	49.3	82.0	46.7	58.5	69.9
MagMax-Ind	63.6	53.1	59.7	49.1	53.8	53.1	79.8	43.2	56.9	75.1
Consensus TA	37.0	25.2	35.2	36.7	37.3	44.1	80.6	30.3	33.5	59.2
C. LW AdaMerging	63.1	60.0	63.5	60.1	35.6	32.1	51.8	45.4	66.6	60.2
C. LoRA-WEMOE	51.4	45.8	63.3	43.5	42.9	34.6	58.9	46.5	47.5	60.1
OPCM	64.4	51.1	66.0	71.7	66.1	56.0	90.2	40.4	64.9	80.2
Mingle (Ours)	67.8	58.3	83.5	90.0	82.9	91.8	98.0	65.3	74.0	66.9
Mingle* (Ours)	68.8	64.2	83.8	91.1	82.4	89.0	96.9	62.8	76.7	72.8
ViT-B/16										
C. Fine-Tuned	62.7	58.0	67.6	99.1	46.0	29.2	93.9	61.9	64.1	75.2
Average (SWA)	67.1	64.6	69.3	63.4	62.4	52.7	80.7	46.6	71.8	63.1
C.TA	65.8	57.5	63.8	59.5	64.7	54.0	88.0	45.3	67.5	67.1
C.TIES	64.2	52.9	60.9	53.0	62.8	48.8	88.4	45.0	61.3	68.5
MagMax-Ind	65.8	51.8	57.8	42.6	54.4	43.7	83.0	42.8	60.4	69.8
Consensus TA	42.6	24.8	30.4	34.4	47.6	42.2	79.9	30.6	36.2	74.3
C. LW AdaMerging	65.5	65.7	69.8	59.4	50.1	44.2	61.1	47.1	71.8	57.9
C. LoRA-WEMOE	62.7	60.2	69.4	37.7	52.1	39.9	63.1	45.3	64.3	51.7
OPCM	67.9	55.9	73.7	77.5	74.4	63.2	94.1	49.2	72.3	79.6
Mingle (Ours)	71.5	64.9	85.3	90.0	87.5	90.1	97.1	62.7	82.6	80.6
Mingle* (Ours)	72.0	72.1	87.9	93.3	87.1	89.2	97.4	62.5	86.8	76.4
ViT-L/14										
C. Fine-Tuned	69.5	73.6	78.3	99.2	59.3	49.3	98.6	69.7	83.2	78.3
Average (SWA)	70.7	77.7	76.4	75.3	69.5	62.1	93.7	57.7	80.0	73.6
C.TA	70.4	74.1	73.9	66.3	69.9	65.6	95.1	56.6	78.6	70.4
C.TIES	69.7	70.3	65.3	47.9	76.1	63.6	94.7	54.4	77.9	72.3
MagMax-Ind	73.1	73.7	75.6	64.6	73.7	68.8	94.6	56.1	78.0	71.7
Consensus TA	50.7	39.1	31.7	36.4	39.4	44.9	88.5	33.8	45.7	62.5
C. LW AdaMerging	68.8	78.6	75.9	65.7	58.3	51.6	79.9	57.4	80.6	52.4
C. LoRA-WEMOE	62.1	68.1	68.7	53.2	47.5	49.4	69.8	49.1	66.2	54.2
OPCM	73.1	78.3	82.4	80.2	80.8	80.4	97.4	61.6	84.8	76.3
Mingle (Ours)	75.9	83.4	87.8	88.7	91.1	94.5	98.4	70.8	94.8	75.3
Mingle* (Ours)	74.5	85.9	90.5	92.5	90.1	92.7	98.1	69.2	95.7	74.0

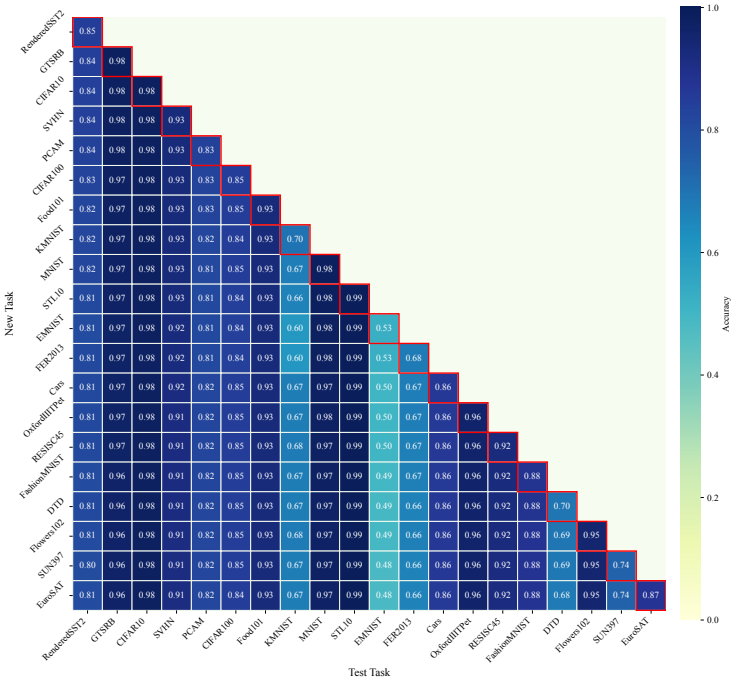
Model	FER2013	OxfordIIITPet	STL10	CIFAR100	CIFAR10	Food101	FashionMNIST	EMNIST	KMNIST	RenderedSST2
ViT-B/32										
C. Fine-Tuned	58.3	68.5	86.7	40.2	70.5	50.0	90.7	72.4	54.5	54.5
Average (SWA)	50.2	84.1	97.0	69.8	92.7	80.4	71.3	15.0	11.5	61.8
C.TA	51.4	82.3	94.9	64.6	91.4	71.9	73.9	17.8	12.2	59.9
C.TIES	49.5	81.3	95.2	63.7	91.2	70.2	73.7	17.8	16.9	59.8
MagMax-Ind	56.5	79.9	94.6	68.7	91.9	73.8	74.3	18.3	15.4	63.9
Consensus TA	41.7	58.8	81.8	41.5	78.1	29.8	72.6	17.4	18.5	54.1
C. LW AdaMerging	43.2	83.7	96.8	67.0	89.9	81.6	63.7	16.8	10.7	59.1
C. LoRA-WEMOE	44.6	72.5	86.1	40.1	63.8	63.8	48.1	10.3	12.8	55.7
OPCM	58.5	82.9	95.9	67.6	92.8	74.0	76.3	22.4	18.3	64.6
Mingle (Ours)	65.0	85.5	97.0	72.6	94.1	81.5	85.4	50.4	65.2	67.1
Mingle* (Ours)	65.3	88.5	97.7	73.9	94.7	83.7	86.4	39.3	56.1	68.7
ViT-B/16										
C. Fine-Tuned	60.5	84.5	90.5	38.8	73.6	61.9	89.7	83.3	51.5	72.8
Average (SWA)	50.9	89.6	98.0	72.9	94.2	85.9	73.3	15.6	12.4	62.5
C.TA	50.7	89.3	97.0	68.0	93.1	80.3	75.7	18.1	16.7	61.8
C.TIES	50.4	87.9	96.3	63.1	91.7	78.0	75.0	23.4	24.9	61.5
MagMax-Ind	57.7	88.8	97.5	71.5	94.4	81.3	77.2	24.5	25.0	59.4
Consensus TA	45.6	76.8	87.7	44.4	82.2	38.4	72.7	18.8	30.0	58.6
C. LW AdaMerging	46.8	88.9	98.1	69.2	91.4	86.6	67.2	17.2	11.0	59.2
C. LoRA-WEMOE	45.6	91.2	92.3	41.3	64.3	78.1	48.0	23.5	16.6	52.7
OPCM	59.5	91.8	97.7	73.2	94.7	83.1	81.3	26.5	23.4	66.8
Mingle (Ours)	67.6	92.7	97.4	74.0	95.3	87.7	87.4	73.5	79.9	74.0
Mingle* (Ours)	67.9	93.5	98.4	77.7	96.4	89.7	87.8	56.6	64.5	75.3
ViT-L/14										
C. Fine-Tuned	68.0	92.1	94.5	60.5	85.7	74.8	93.1	89.0	59.2	78.8
Average (SWA)	52.7	94.2	99.2	81.7	97.0	90.7	77.4	16.1	10.4	66.1
C.TA	55.7	94.2	98.6	79.1	96.6	87.6	80.8	17.6	10.6	63.6
C.TIES	57.6	93.5	97.8	74.0	95.6	84.7	79.7	20.2	12.6	58.4
MagMax-Ind	52.9	93.9	98.7	82.1	97.3	89.5	81.6	19.2	11.1	68.4
Consensus TA	50.3	82.2	89.7	47.5	86.2	43.5	75.3	14.5	10.4	53.4
C. LW AdaMerging	49.2	93.5	99.3	77.2	95.8	91.1	68.2	18.6	9.8	66.6
C. LoRA-WEMOE	46.3	84.5	87.6	52.1	70.5	73.3	50.0	18.7	10.9	56.5
OPCM	61.8	95.4	99.2	83.0	97.8	90.9	86.0	26.4	14.7	71.0
Mingle (Ours)	67.7	96.0	98.7	81.4	97.1	90.6	90.6	60.7	88.6	79.8
Mingle* (Ours)	67.9	96.0	99.4	84.7	97.8	92.4	88.8	53.0	57.1	75.5



(a) ViT-B/32 on 8 Tasks (b) ViT-B/16 on 8 Tasks (c) ViT-L/14 on 8 Tasks

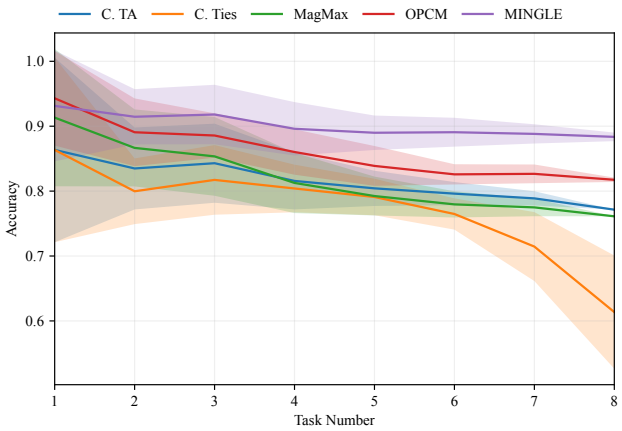


(d) ViT-B/16 on 14 Tasks (e) ViT-L/14 on 14 Tasks

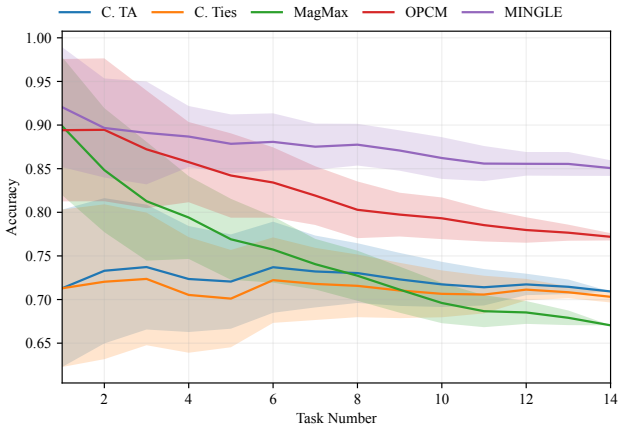


(f) ViT-L/14 on 20 Tasks

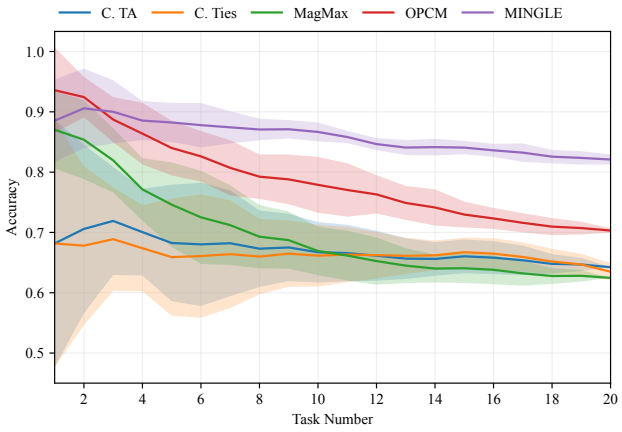
Figure 5: Accuracy matrices of MINGLE (ViT-B/32, ViT-B/16, and ViT-L/14) under different task settings.



(a) 8 tasks



(b) 14 tasks



(c) 20 tasks

Figure 6: Sequential test accuracy curves of MINGLE and baselines (C.TA, C.Ties, MagMax, OPCM) under different task settings. Shaded regions indicate standard deviation across 10 task orders.

Table 14: Robustness results when merging ViT-B/32 models on four tasks.

Method	Clean Test Set					Corruption: Motion Blur				
	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**
C. LW AdaMerging	65.3	49.7	65.4	43.8	56.0	64.2	25.6	62.5	37.5	47.5
C. WEMOE	0.5	8.1	2.6	2.5	3.4	0.5	8.0	1.8	2.3	3.1
C. LoRA-WEMOE	66.1	84.3	81.0	83.6	78.7	64.8	57.9	82.0	79.2	71.0
C. Task Arithmetic	64.6	90.4	80.2	74.8	77.5	62.3	59.4	78.5	63.3	65.9
MagMax-Ind	63.1	89.2	81.7	82.5	79.1	61.4	62.1	80.0	72.6	69.0
OPCM	65.7	92.3	85.7	90.5	83.6	62.8	62.5	83.7	82.2	72.8
Mingle (Ours)	74.4	96.5	91.5	97.3	89.9	73.2	70.5	91.9	95.8	82.9

Method	Corruption: Impulse Noise					Corruption: Gaussian Noise				
	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**
C. LW AdaMerging	60.5	30.1	56.3	25.5	43.1	62.3	25.6	59.7	25.6	43.3
C. WEMOE	0.5	11.2	2.3	3.2	4.3	0.5	8.1	2.4	2.8	3.4
C. LoRA-WEMOE	62.2	23.4	69.9	64.6	55.0	64.9	31.7	77.8	63.4	59.4
C. Task Arithmetic	59.9	57.7	72.9	45.0	58.9	61.8	51.4	75.1	50.1	59.6
MagMax-Ind	59.2	56.3	74.3	52.5	60.6	60.6	51.7	77.0	56.5	61.5
OPCM	61.1	57.1	78.5	62.0	64.7	63.0	52.4	80.7	64.9	65.2
Mingle (Ours)	69.6	28.0	86.1	86.1	67.5	72.0	38.5	89.4	82.9	70.7

Method	Corruption: Pixelate					Corruption: Spatter				
	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**
C. LW AdaMerging	3.4	16.5	13.5	39.2	18.1	61.3	34.1	58.2	32.8	46.6
C. WEMOE	0.5	6.3	2.5	2.5	3.0	0.5	10.1	2.7	2.6	4.0
C. LoRA-WEMOE	0.8	26.0	5.8	67.0	24.9	62.4	35.4	71.2	73.0	60.5
C. Task Arithmetic	2.5	31.7	19.1	65.6	29.7	61.2	63.1	72.7	57.0	63.5
MagMax-Ind	2.6	36.1	19.3	74.0	33.0	60.0	64.9	74.8	66.1	66.4
OPCM	2.1	34.3	19.5	84.9	35.2	61.5	64.7	78.8	76.8	70.5
Mingle (Ours)	2.3	35.6	18.5	95.1	37.9	70.1	57.8	86.2	93.9	77.0

Method	Corruption: Contrast					Corruption: JPEG Compression				
	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**	Cars	EuroSAT	RESISC45	GTSRB	**Avg ACC**
C. LW AdaMerging	61.8	26.0	63.1	44.8	48.9	65.1	29.6	65.4	36.4	49.1
C. WEMOE	0.5	7.5	2.3	3.0	3.3	0.5	10.5	2.4	2.7	4.0
C. LoRA-WEMOE	64.3	46.5	77.7	85.6	68.5	65.5	59.1	80.4	74.0	69.7
C. Task Arithmetic	62.5	55.2	75.3	70.8	66.0	64.1	66.2	80.0	61.0	67.8
MagMax-Ind	61.3	58.0	76.9	78.2	68.6	62.5	67.7	81.1	68.5	69.9
OPCM	63.8	57.5	81.3	87.4	72.5	65.0	68.0	85.4	79.3	74.4
Mingle (Ours)	72.4	60.1	90.4	97.3	80.1	73.7	73.5	92.0	92.4	82.9

C.4 Inference Efficiency and Parameter Overhead

Tab. 15 compares the inference efficiency and parameter overhead of all baselines on the CLIP ViT-B/32 model after merging eight tasks. We report the total number of parameters, additional storage and inference parameters, throughput (images per second), and accuracy. The results show that most static merging methods (*e.g.*, Task Arithmetic, Ties-Merging, MAGMAX-Ind, OPCM) incur no extra storage or inference overhead, but typically achieve limited accuracy. Consensus TA and WEMOE introduce significant storage overhead, while WEMOE also scales up inference parameters considerably. By contrast, Mingle achieves a favorable trade-off: although it introduces additional parameters for LoRA experts and the router, the effective inference overhead remains small, and throughput is only marginally reduced compared to static baselines. This efficiency advantage comes while delivering substantially higher accuracy.

C.5 Forward Transfer Analysis

Forward transfer (FWT) is an important metric in continual learning, as it quantifies how effectively prior knowledge facilitates the learning of future tasks. We adopt the standard definition:

$$\text{FWT} = \frac{1}{T-1}\sum_{t=2}^{T}\left[a_t(\theta_t^{\text{merged}}) - \bar{a}_t\right], \tag{30}$$

where $a_t(\theta_t^{\text{merged}})$ denotes the test accuracy on task t using the merged model after task t, and $\bar{a}_t$ is the accuracy of the individually fine-tuned model for task t. Positive FWT indicates beneficial transfer, while negative values suggest interference.

Table 15: Comparison of inference efficiency and parameter overhead on CLIP ViT-B/32 model after eight tasks merging.

Method	Total Params (M)	Extra Storage (M)	Extra Inference (M)	Throughput (img/s)	ACC (%)
TASK ARITHMETIC	87.5	0.0	0.0	∼910	67.5
TIES-MERGING	87.5	0.0	0.0	∼910	49.0
MAGMAX-IND	87.5	0.0	0.0	∼910	70.7
OPCM	87.5	0.0	0.0	∼910	75.5
CONSENSUS TA	87.5	87.5	0.0	∼910	69.0
LW. ADAMERGING	87.5	0.0	0.0	∼910	52.9
WEMOE	540.9	453.4	0.07	∼858	4.9
WEMOE-LORA	103.7	16.2	0.07	∼848	66.6
MINGLE	173.1	85.6	0.6	∼841	85.8
MINGLE*	113.7	26.2	0.3	∼862	85.0

Table 16: Forward transfer (FWT) results on 8-task continual merging with CLIP ViT-B/16.

Method	ACC (%)	BWT (%)	FWT (%)
TASK ARITHMETIC	77.1 $\pm$ 0.0	-4.2 $\pm$ 1.0	-13.4 $\pm$ 0.0
TIES-MERGING	66.8 $\pm$ 3.7	-5.5 $\pm$ 0.4	-30.7 $\pm$ 9.9
OPCM	81.8 $\pm$ 0.3	-4.8 $\pm$ 0.7	-9.0 $\pm$ 0.4
MINGLE (Ours)	88.3 $\pm$ 0.6	-0.4 $\pm$ 0.1	-3.8 $\pm$ 0.8

Tab. 16 reports the results for the 8-task continual merging setup on CLIP ViT-B/16. The results demonstrate that MINGLE achieves nearly zero forgetting (BWT ≈ 0) while obtaining the highest forward transfer among all baselines, showing that our adaptive gating and merging strategy not only preserves past knowledge but also enhances feature utility for future tasks.

C.6 Additional Visualizations of Gate Activations and the Relaxation Effect

We provide an extended ablation study on gate hyper-parameters, including visualizations of gate activations under 14-task (Fig. 7) and 20-task (Fig. 8) configurations, complementing the 8-task results presented in the main paper. The visualizations demonstrate that the null-space constraint remains effective as the number of tasks increases, consistently suppressing gate responses to inputs from previously seen tasks and thereby mitigating forgetting.

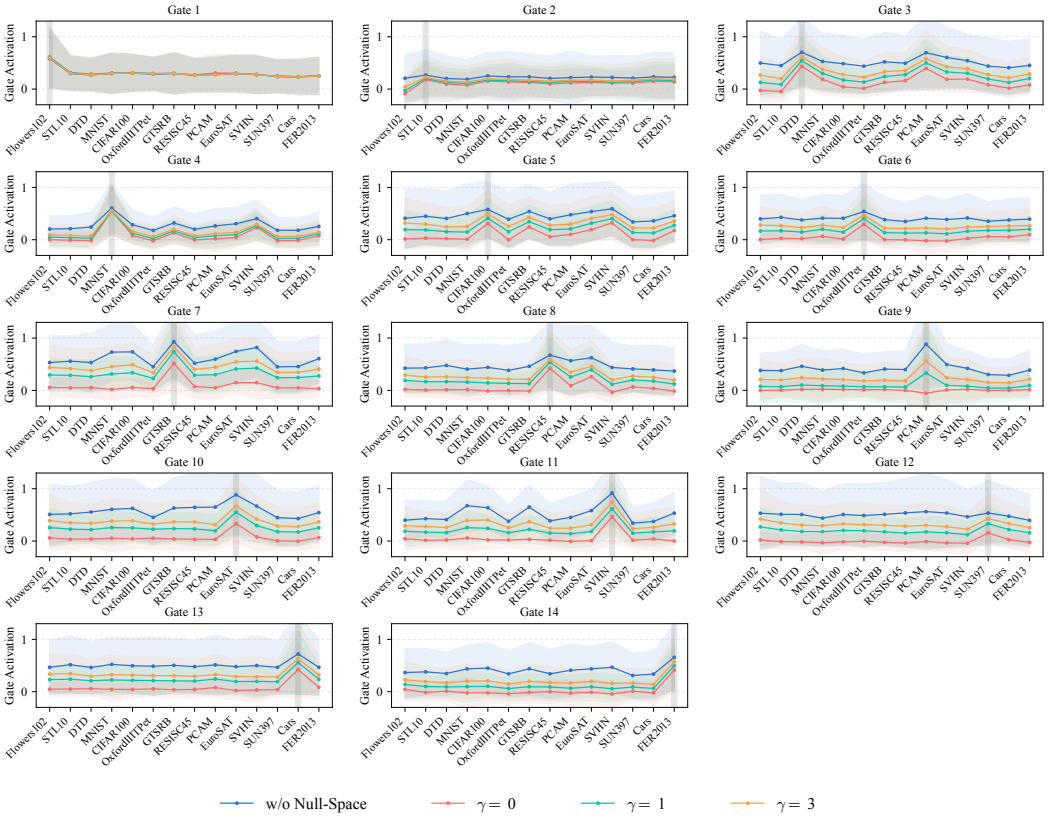


Figure 7: Visualization of gate activations across 14 tasks under varying γ values. Each subplot corresponds to a gate, with curves and shaded regions denoting the mean and standard deviation of activations across layers. Gray bars mark the training dataset for each gate. Smaller γ values result in stronger suppression of activations on previously learned tasks.

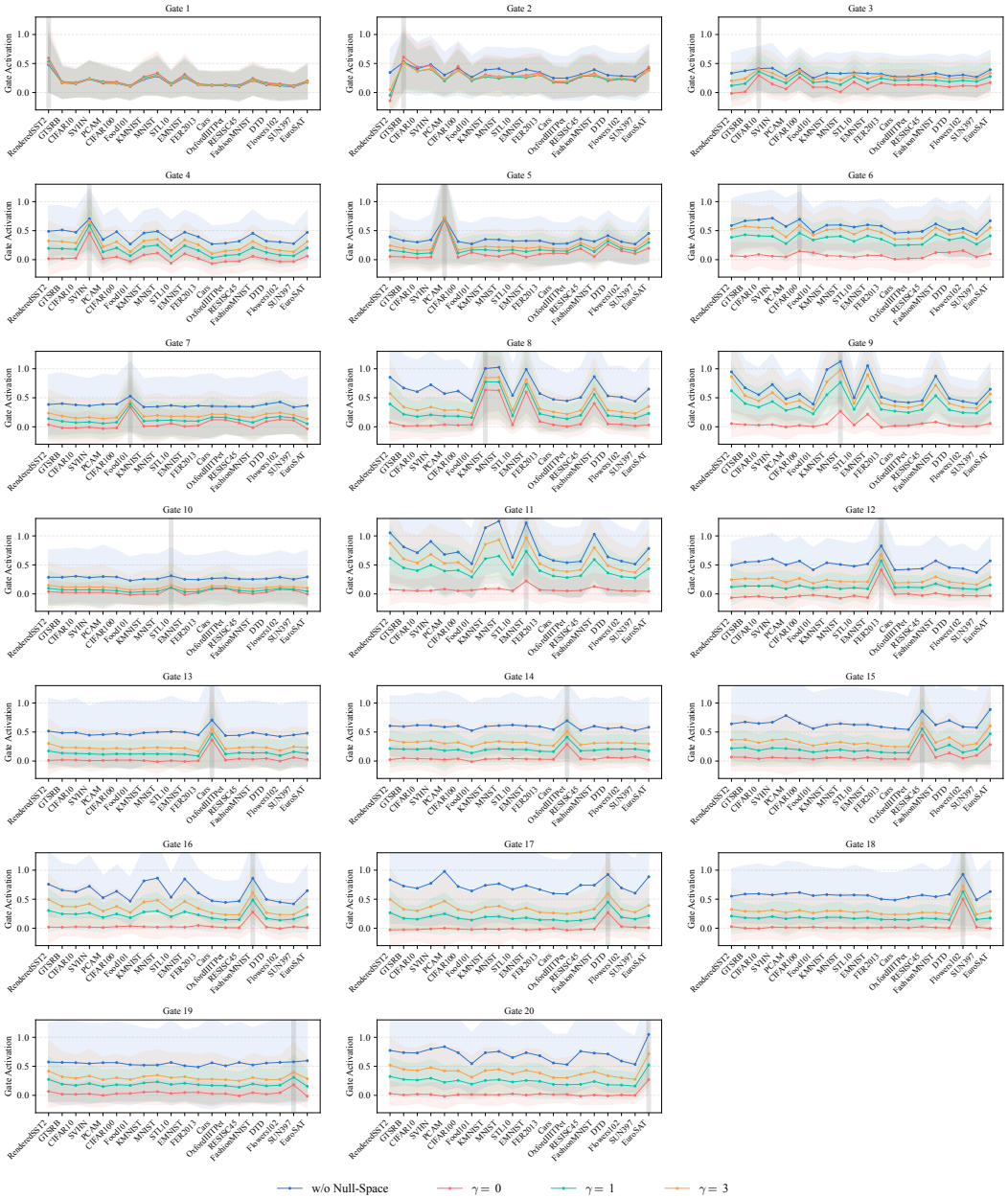


Figure 8: Visualization of gate activations across 20 tasks under varying γ values.

D Discussions

D.1 Use of Unlabeled Adaptation Samples

In our experiments, we simulate a realistic deployment setting by randomly sampling 5 unlabeled examples per class from the test split, which serve as adaptation samples for model merging. Such small unlabeled buffers are practical in real-world applications and can be obtained from various sources, including (i) incoming test-time data such as recent user queries or model inputs, (ii) held-out validation inputs or small training subsets (if available), (iii) user-provided samples (*e.g.*, few-shot examples) that do not raise privacy concerns, (iv) synthetically generated data, or (v) manually curated public data. Importantly, our method does not depend on precise sample selection, and the buffer size remains fixed and small, ensuring feasibility and robustness in deployment scenarios.

D.2 Relation to Rehearsal-Free Continual Learning

Test-time continual model merging (TTCMM) is closely related to the paradigm of rehearsal-free continual learning (RFCL), as both approaches share two fundamental constraints: (i) they do not retain past training data, and (ii) they avoid storing previous task models. The key distinction lies in the information available at each stage. RFCL assumes access to the training data of the current task and incrementally fine-tunes a single model over time. In contrast, TTCMM assumes access to independently fine-tuned models for each new task and focuses on merging these expert models rather than training them from scratch. Additionally, TTCMM relies on a small unlabeled buffer at test time (*e.g.*, 5 samples per class) to guide the merging process.

From a privacy perspective, TTCMM provides stronger guarantees. Since it does not require access to full training sets, it only depends on a small set of unlabeled samples, which can be user-provided without risk, synthetically generated, or curated from public data. By comparison, RFCL requires access to large-scale labeled datasets for every task, raising more significant concerns regarding privacy, storage, and legal constraints (*e.g.*, medical images, personal data, or copyrighted corpora). The reliance on a tiny unlabeled buffer makes TTCMM more practical in scenarios where data privacy is a primary consideration.

D.3 Limitations

As with many model merging methods, our approach assumes that all independently fine-tuned models originate from a shared pretrained initialization. The extent to which this assumption influences merging performance remains unclear and warrants further investigation. In addition, our current experiments focus on merging models with identical backbone architectures (*e.g.*, CLIP ViT-B/16). Although our use of LoRA-based expert offers some structural uniformity, which could potentially accommodate heterogeneous backbones, we have not yet explored this setting. Extending our framework to support diverse initialization points or architectural variants remains an open direction for future work.

D.4 Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.