DYNAMIC MODEL MERGING WITH MULTI-LAYER AGGREGATED ROUTING EVIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

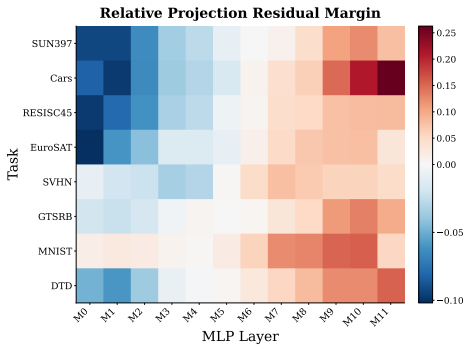
Dynamic model merging aims to retain the efficiency of a unified model while better preserving task-specific expertise through input-dependent routing at both the task and head levels. However, even the routing components of the state-of-the-art approach remain limited: the task routing is highly sensitive to the choice of routing layer, whereas the head routing relies on simple max-logit competition. To address these limitations, we propose multi-layer aggregated routing evidence (MATE) as a unified mechanism for improving both task and head routing. Specifically, we first introduce a metric to quantify task routing separability across different layers. Empirical analysis of this metric reveals that different layers provide complementary evidence for identifying the tasks relevant to an input. Motivated by this observation, MATE selects multiple informative layers according to their routing separability and aggregates their layer-wise routing scores to produce a more robust signal that can be directly used for task selection. Moreover, we observe that this signal also encodes useful information about the suitability of the corresponding classification heads. This motivates us to further combine the MATE signal with the max-logit score to guide classification head selection. Extensive experiments on vision benchmarks demonstrate that MATE consistently achieves state-of-the-art performance.

1 INTRODUCTION

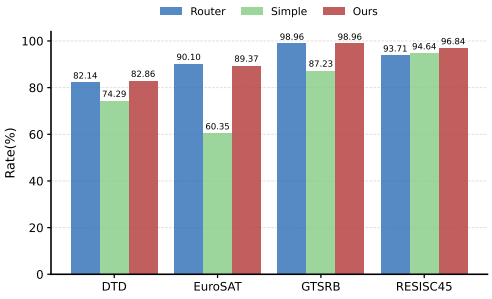
Over the past several years, fine-tuning large pretrained models has become a standard approach to obtaining specialized models for downstream tasks (Devlin et al., 2019; Neyshabur et al., 2020; Wortsman et al., 2022b; Ilharco et al., 2022). As the number of tasks grows, however, maintaining a separate fine-tuned model for each task becomes increasingly costly (Houlsby et al., 2019; Hu et al., 2022). Conventional multi-task learning can reduce this burden by consolidating these capabilities into a single model through joint training, but requires access to the original data and incurs additional training costs (Jin et al., 2022). By contrast, model merging offers an alternative route to consolidation by directly combining the parameters of existing fine-tuned models and reusing the task knowledge encoded in their weights (Matena & Raffel, 2022; Ilharco et al., 2023). Its reduced dependence on training and the original datasets has attracted growing research interest in recent years (Yang et al., 2024; Zheng et al., 2025; Yang et al., 2026).

Early research on model merging explored various forms of parameter averaging to consolidate separately trained models (Singh & Jaggi, 2020; Wortsman et al., 2022a; Matena & Raffel, 2022). In a seminal work, Ilharco et al. (2023) introduced Task Arithmetic, establishing an influential formulation for composing task knowledge through parameter updates rather than averaging complete model weights. The method represents task-specific parameter changes relative to a shared pretrained model as task vectors and combines them to compose task capabilities. Building on this formulation, subsequent methods reduce interference among task vectors by removing redundant updates and resolving conflicts in the original parameter space (Yadav et al., 2023; Yu et al., 2024). More recent work extends this perspective to the subspace structure of task vectors, using singular value decomposition to identify singular directions and their associated magnitudes. These efforts include exploiting low-rank structure (Choi et al., 2024; Gargiulo et al., 2025; Skorobogat et al., 2025) and improving subspace alignment and spectral balance during merging (Stoica et al., 2025; Marczak et al., 2025; Panariello et al., 2026; Li et al., 2026; Zhang et al., 2026; Chaves et al., 2026).

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



(a) Layer-wise task routing separability



(b) True-task selection rates among samples with multiple candidate tasks and the true task retained

Figure 1: Empirical motivation for our method. (a) Relative projection residual margins across MLP layers of ViT-B/32 with 8 tasks. (b) With the true task among multiple candidates, Router reports how often it has the highest routing score. Simple and Ours respectively report its head's win rate under simple and routing-guided max-logit competition. More details are provided in Appendix A.

Despite these advances, these methods share a static paradigm: a single merged model applies the same parameter combination to every input. This fixed combination can weaken task-specific directions and diminish the expertise needed for individual tasks. To better preserve task-specific expertise, dynamic model merging draws on the mixture-of-experts (MoE) idea, treating task vectors as experts to be selected and combined according to each input (Yadav et al., 2024). A straightforward solution retains complete task vectors and uses task identity to guide adaptive merging and prediction. While this preserves access to task-specific knowledge, its storage cost grows linearly with the number of tasks. More importantly, this solution relies on task identity at inference time, whereas the unknown-task scenario requires the model to infer task relevance and make predictions from the input alone. These considerations raise two challenges. First, how can task knowledge be retained at low cost? Second, how can input-dependent routing identify both the relevant task knowledge and the appropriate classification head without prior knowledge of the task identity?

WEMoE (Tang et al., 2024) addresses the first challenge by retaining task-specific MLP weights for dynamic merging while statically merging the remaining parameters. However, its input-dependent merging weights rely on parametric routers that require training through test-time adaptation. Several subsequent studies further explore expert sparsification, shared and task-exclusive components, and selective dynamic merging across layers, but still rely on parametric routers that require additional training (Lu et al., 2024; Ye et al., 2025; Shen et al., 2025). To eliminate this training requirement, SMILE (Tang et al., 2025) compresses task vectors into low-rank experts through singular value decomposition and introduces nonparametric routing based on projections of intermediate input representations onto task subspaces. However, it selects experts sequentially from lower to higher layers, allowing routing errors to propagate as later decisions depend on representations produced by earlier selections. More importantly, WEMoE and SMILE address only part of the second challenge: they select and combine task knowledge according to the input, but their evaluations still rely on oracle classification heads specified by the true task identity.

MASS (Crisostomi et al., 2026) takes a different approach to task routing, decoupling task selection from adaptive merging through two-pass inference. It first constructs a fixed merged model using TSV-M (Gargiulo et al., 2025) and uses projection residuals at a single preselected layer to select candidate task subspaces, which are then merged for a second forward pass. Crucially, it extends routing to classification heads through max-logit competition, enabling prediction without prior knowledge of the task. MASS achieves state-of-the-art performance among existing methods. However, it still has two key limitations in both task and head routing. Specifically, task routing is highly sensitive to layer selection, with different tasks favoring different layers. MASS reports cross-task accuracy variation of up to 40% at some layers (Crisostomi et al., 2026). Meanwhile, head routing relies on simple max-logit competition, which can select an incorrect head even when the true task's subspace is retained among the candidates. For example, among EuroSAT samples with multiple candidate tasks including EuroSAT, simple max-logit competition selects the EuroSAT classification head in only 60.35% of cases, as shown in Figure 1b.

In this paper, we propose multi-layer aggregated routing evidence (MATE) as a unified mechanism for improving both task and head routing. To improve task routing, we first investigate its sensitivity to layer selection, which calls for a quantitative measure of layer-wise task separability. The projection residuals used in MASS reflect input–task compatibility, but do not directly provide such a measure. Their magnitudes depend on representation scale, while a small residual for the true task does not ensure separation from competing tasks. We therefore normalize the residuals by the representation norm and measure the margin between the true task and its most competitive incorrect task. The resulting relative projection residual margin characterizes task separability across layers, with larger values indicating clearer separation. Based on this metric, our layer-wise analysis in Figure 1a reveals distinct separability patterns across tasks: layers that discriminate well for one task may be less informative for another. These patterns suggest that multiple layers provide complementary routing evidence. Motivated by this observation, MATE selects informative candidate layers according to their routing separability and aggregates their routing scores into a more robust signal for adaptive task subspace selection.

Moreover, since each classification head corresponds to a particular task, the routing signal identifying relevant task knowledge should also help distinguish suitable heads. We test this intuition on samples whose candidate sets contain multiple tasks, including the true task, as shown in Figure 1b. Notably, on EuroSAT, the true task receives the highest routing score in 90.10% of cases, whereas simple max-logit competition selects its head in only 60.35%. This gap shows that the true task's top routing rank does not imply that its classification head will win under simple max-logit competition. Interestingly, we further weight head logits by routing scores and find that this raises the correct-head selection rate to 89.37%, substantially narrowing the gap. These findings motivate us to incorporate the MATE signal into head routing, weighting head logits before max-logit selection so the same evidence supports both task subspace selection and classification head routing.

We evaluate MATE on visual model merging benchmarks comprising 8, 14, and 20 tasks using three CLIP (Radford et al., 2021) backbones: ViT-B/32, ViT-B/16, and ViT-L/14. MATE consistently outperforms the evaluated baselines across all nine configurations, supporting the effectiveness of multi-layer routing aggregation and routing-guided head selection.

To summarize, our key contributions are as follows:

- We introduce the relative projection residual margin to quantify layer-wise task routing separability, revealing task-dependent layer preferences and complementary routing evidence. We further show that routing scores provide useful information for classification head selection.
- We propose MATE, a unified mechanism for task and head routing in the unknown-task scenario. MATE selects informative layers and aggregates their routing scores for adaptive task subspace selection, then uses the same aggregated signal to guide classification head selection.
- We evaluate MATE across nine configurations spanning three CLIP ViT backbones and three task scales. MATE achieves the best model merging performance in all nine configurations, with maximum gains across the three task scales of 2.19, 1.84, and 0.69 percentage points over the strongest baseline on ViT-B/32, ViT-B/16, and ViT-L/14, respectively. Routing analyses and ablation studies further support the effectiveness of both designs.

2 PRELIMINARIES

In this section, we briefly review the basic ingredients underlying MATE: task vectors, TSV-M for subspace-based model merging, and the routing procedure used in MASS.

Task vectors. Let θ_{pre} denote the pretrained model parameters and $\{\theta_i\}_{i=1}^{T}$ the parameters of models independently fine-tuned from θ_{pre} on T downstream tasks. Following Task Arithmetic (Ilharco et al., 2023), the parameter update for task i, referred to as its task vector, is

$$\Delta_i = \theta_i - \theta_{\text{pre}}. \tag{1}$$

TSV-M. Let $\Delta_i^{(\ell)}$ denote the matrix corresponding to task vector Δ_i at layer ℓ. TSV-M (Gargiulo et al., 2025) compresses this matrix by retaining its leading singular components, yielding $\Delta_i^{(\ell)} \approx$

Algorithm 1: TSV-M (Gargiulo et al., 2025)

Input: Task vectors $\{\Delta_i^{(\ell)}\}_{i=1}^{T}$ at layer ℓ
Output: Merged update $\Delta_{\mathrm{MT}}^{(\ell)}$

for $i = 1, \ldots, T$ **do**
    Compute the reduced SVD $\Delta_i^{(\ell)} = U_i^{(\ell)} \Sigma_i^{(\ell)} V_i^{(\ell)\top}$
    Keep the leading $1/T$ fraction of singular components in $U_i^{(\ell)}$, $\Sigma_i^{(\ell)}$, and $V_i^{(\ell)}$
end
$U \leftarrow [U_1^{(\ell)}, \ldots, U_T^{(\ell)}], \quad V \leftarrow [V_1^{(\ell)}, \ldots, V_T^{(\ell)}]$
$\Sigma \leftarrow \text{block-diag}(\Sigma_1^{(\ell)}, \ldots, \Sigma_T^{(\ell)})$
Compute the SVDs $U = P_U D_U Q_U^\top$ and $V = P_V D_V Q_V^\top$
$U_\perp \leftarrow P_U Q_U^\top, \quad V_\perp \leftarrow P_V Q_V^\top$
$\Delta_{\mathrm{MT}}^{(\ell)} \leftarrow U_\perp \Sigma V_\perp^\top$
return $\Delta_{\mathrm{MT}}^{(\ell)}$

$U_i^{(\ell)} \Sigma_i^{(\ell)} V_i^{(\ell)\top}$, where $V_i^{(\ell)}$ contains the retained right singular vectors. In MASS and our method, $V_i^{(\ell)}$ is used as the task subspace basis for routing. At each layer, let U and V denote the concatenations of the retained left and right singular vectors across tasks, and let Σ be the block-diagonal matrix of their singular values. TSV-M orthogonalizes these bases to reduce cross-task interference and reconstructs the merged update:

$$\Delta_{\mathrm{MT}}^{(\ell)} = \mathrm{orth}(U)\Sigma\,\mathrm{orth}(V)^\top, \tag{2}$$

where $\mathrm{orth}(X) = PQ^\top$ for the reduced SVD $X = PDQ^\top$. The layer-wise updates form Δ_{MT}, yielding $\theta_{\mathrm{MT}} = \theta_{\mathrm{pre}} + \alpha\Delta_{\mathrm{MT}}$, where TSV-M sets $\alpha = 1$. Algorithm 1 summarizes the layer-wise merging procedure.

MASS. MASS (Crisostomi et al., 2026) first constructs the fixed merged model θ_{MT} by setting $\theta_{\mathrm{MT}}^{(\ell)} = \theta_{\mathrm{pre}}^{(\ell)} + \text{TSV-M}(\{\Delta_i^{(\ell)}\}_{i=1}^{T})$ at each layer, then passes the input x through it. Let $z_\ell(x)$ denote the intermediate representation entering the weight matrix at layer ℓ of θ_{MT}. MASS then uses $z_\ell(x)$ at a single preselected routing layer to measure task relevance through the projection residual

$$r_{i,\ell}(x) = \left\| z_\ell(x) - V_i^{(\ell)} V_i^{(\ell)\top} z_\ell(x) \right\|_2, \tag{3}$$

where a smaller residual indicates a closer match to task i. The residuals are converted into a soft task distribution:

$$w_{\ell,i}(x) = \frac{\exp(-r_{i,\ell}(x))}{\sum_{j=1}^{T} \exp(-r_{j,\ell}(x))}. \tag{4}$$

MASS forms a candidate task set $\Omega(x)$ by thresholding these scores with η and retaining at most K_T top-scoring tasks. It then merges the selected task subspaces and performs a second forward pass to obtain the final representation $z^{\mathrm{ada}}(x)$. Let h_i denote the classification head for task i, with class set $\mathcal{C}_i$. MASS makes the final prediction through simple max-logit competition among candidate classification heads.

3 METHOD

In this section, we present multi-layer aggregated routing evidence (MATE) as a unified mechanism for task and head routing in the unknown-task scenario. We first quantify layer-wise task separability using the relative projection residual margin on a small set of proxy samples. We then select informative candidate routing layers and aggregate their routing scores into a task relevance signal for adaptive subspace selection. Finally, we use the same signal to guide the final decision among candidate classification heads.

3.1 LAYER-WISE ROUTING ANALYSIS

The projection residual in Equation (3) depends on both the alignment with a task subspace and the magnitude of the input representation. Differences in representation scale can therefore obscure

comparisons across layers. To assess layer-wise task separability on a comparable scale, we first normalize the residual by the representation norm:

$$\bar{r}_{i,\ell}(x) = \frac{r_{i,\ell}(x)}{\|z_\ell(x)\|_2 + \epsilon}. \tag{5}$$

Here, $r_{i,\ell}(x)$ is the projection residual between the representation $z_\ell(x)$ and task i's subspace at layer ℓ, and ϵ is a constant for numerical stability.

For a sample x from task i, the normalized residual of the true task alone still does not fully characterize the discriminative ability of a routing layer. Its separation from the most competitive incorrect task also matters. Therefore, we further define the sample-level relative projection residual margin as

$$m_{i,\ell}(x) = \min_{j \neq i} \bar{r}_{j,\ell}(x) - \bar{r}_{i,\ell}(x). \tag{6}$$

When $m_{i,\ell}(x) > 0$, the true task has the smallest relative projection residual. A larger margin indicates clearer separation between the true task and the most competitive incorrect task. To characterize layer-wise task routing separability, we use a small proxy dataset $\mathcal{D}_i^{\text{proxy}}$ for each task i and compute its task-specific mean margin at layer ℓ:

$$A_{i,\ell} = \frac{1}{|\mathcal{D}_i^{\text{proxy}}|} \sum_{x \in \mathcal{D}_i^{\text{proxy}}} m_{i,\ell}(x). \tag{7}$$

Figure 1a shows that layers with high routing separability for one task can be less informative for another. These task-dependent differences suggest complementary routing evidence across layers. Additional layer-wise analyses across backbones and task scales are provided in Appendix B.

3.2 Candidate Layer Selection and Multi-Layer Routing Aggregation

Building on the preceding analysis, we now introduce MATE's multi-layer routing aggregation mechanism. Following MASS, we first construct θ_{MT} using TSV-M (Gargiulo et al., 2025) for routing and cache the compressed task subspaces for efficient reuse during subsequent merging. Rather than relying on a single layer, MATE selects multiple informative layers according to their task separability and aggregates their routing scores for candidate task selection.

Candidate layer selection. However, aggregating routing scores from all layers would also include evidence from layers with weak task separability. MATE therefore computes the task-averaged margin $\bar{A}_\ell = \frac{1}{T}\sum_{i=1}^{T} A_{i,\ell}$ for each available MLP routing layer $\ell \in \mathcal{L}$ and selects the top K_L layers to form the candidate set:

$$\mathcal{L}_{\text{cand}} = \text{TopK}_{\ell \in \mathcal{L}} \left(\bar{A}_\ell, K_L \right). \tag{8}$$

Multi-layer routing aggregation. Given $\mathcal{L}_{\text{cand}}$, a single forward pass through θ_{MT} obtains the representations $\{z_\ell(x)\}_{\ell \in \mathcal{L}_{\text{cand}}}$, from which MATE computes the layer-wise routing scores using Equations (3) and (4). MATE then averages the routing scores over $\mathcal{L}_{\text{cand}}$ before candidate task selection, retaining evidence that individual-layer thresholding or Top-K selection would discard:

$$s_i(x) = \frac{1}{|\mathcal{L}_{\text{cand}}|} \sum_{\ell \in \mathcal{L}_{\text{cand}}} w_{\ell,i}(x). \tag{9}$$

The aggregated score $s_i(x)$ represents the relative support for task i across the candidate layers. Based on this score, we perform task routing by constructing the candidate task set $\Omega(x)$. To be precise, we first retain tasks whose scores meet the threshold η, forming $\Omega_\eta(x) = \{i \mid s_i(x) \geq \eta\}$. Furthermore, we restrict this set to at most K_T tasks with the highest aggregated scores:

$$\Omega(x) = \text{TopK}\left(\Omega_\eta(x), s(x), K_T\right), \tag{10}$$

where $s(x) = (s_1(x), \ldots, s_T(x))$. If no task passes the threshold, i.e., $\Omega_\eta(x) = \varnothing$, we retain the task with the highest aggregated score by setting $\Omega(x) = \{\arg\max_i s_i(x)\}$.

After task routing, we merge the selected task subspaces to construct the model for the current input. Note that this step is not intrinsically tied to our routing mechanism: once the candidate task set $\Omega(x)$ is determined, other compatible model merging methods could in principle be employed. To isolate the contribution of our routing design and ensure a fair comparison, we follow MASS in using TSV-M (Gargiulo et al., 2025) to merge the task subspaces associated with $\Omega(x)$ at each layer.

Algorithm 2: Dynamic Model Merging via MATE

Input: sample x; proxy sets $\{\mathcal{D}_i^{\text{proxy}}\}_{i=1}^T$; pretrained model θ_{pre}; task vectors $\{\Delta_i\}_{i=1}^T$; MLP routing layers $\mathcal{L}$ and candidate layer count K_L; routing threshold η and task limit K_T; task heads $\{h_i\}_{i=1}^T$
Output: Predicted class $c^\star$

Obtain the fixed merged model θ_{MT} and layer-wise subspace bases $\{V_i^{(\ell)}\}_{i=1,\dots,T;\,\ell\in\mathcal{L}}$ using TSV-M with θ_{pre} and $\{\Delta_i\}_{i=1}^T$ *// Cache compressed task subspaces*
// Offline: one-time candidate layer selection before inference
for $i = 1, \dots, T$ **do**
  for $x' \in \mathcal{D}_i^{\text{proxy}}$ **do**
    $\{z_\ell(x')\}_{\ell\in\mathcal{L}} \leftarrow \text{ForwardPass}(\theta_{\text{MT}}, x'; \mathcal{L})$
    for $\ell \in \mathcal{L}$ **do**
      Compute $\bar{r}_{j,\ell}(x')$ for $j = 1, \dots, T$ using Eq. (5)
      Compute $m_{i,\ell}(x')$ using Eq. (6)
    end
  end
  Compute $A_{i,\ell}$ for $\ell \in \mathcal{L}$ using Eq. (7)
end
$\bar{A}_\ell \leftarrow \frac{1}{T}\sum_{i=1}^T A_{i,\ell}, \quad \ell \in \mathcal{L}$
Select $\mathcal{L}_{\text{cand}}$ using Eq. (8)
// Online: multi-layer routing and prediction
$\{z_\ell(x)\}_{\ell\in\mathcal{L}_{\text{cand}}} \leftarrow \text{ForwardPass}(\theta_{\text{MT}}, x; \mathcal{L}_{\text{cand}})$
for $\ell \in \mathcal{L}_{\text{cand}}$ **do**
  Compute $r_{i,\ell}(x)$ for $i = 1, \dots, T$ using Eq. (3)
  Compute $w_{\ell,i}(x)$ for $i = 1, \dots, T$ using Eq. (4)
end
$s_i(x) \leftarrow \frac{1}{|\mathcal{L}_{\text{cand}}|}\sum_{\ell\in\mathcal{L}_{\text{cand}}} w_{\ell,i}(x), \quad i = 1, \dots, T$ *// Aggregate routing scores*
$\Omega_\eta(x) \leftarrow \{i \mid s_i(x) \geq \eta\}$
$\Omega(x) \leftarrow \text{TopK}\big(\Omega_\eta(x), s(x), K_T\big)$
if $\Omega_\eta(x) = \varnothing$ **then** $\Omega(x) \leftarrow \{\arg\max_i s_i(x)\}$
$\theta_{\text{adp}} \leftarrow \theta_{\text{pre}} + \text{TSV-M}\big(\{\Delta_i\}_{i\in\Omega(x)}\big)$ *// Apply to each layer using cached subspaces*
$z^{\text{ada}}(x) \leftarrow \text{ForwardPass}(\theta_{\text{adp}}, x)$
$\tilde{o}_i(x) \leftarrow s_i(x)\, h_i\big(z^{\text{ada}}(x)\big), \quad i \in \Omega(x)$ *// Routing-guided head decision*
Select $(i^\star, c^\star)$ using Eq. (12)
return $c^\star$

3.3 Routing-Guided Head Decision

Having selected and merged the candidate task subspaces, we now turn to head routing to determine which candidate classification head should make the final prediction. The second forward pass through the dynamic model yields $z^{\text{ada}}(x)$, from which each candidate head produces logits $o_i(x) = h_i(z^{\text{ada}}(x))$ for $i \in \Omega(x)$. A straightforward choice is to use simple max-logit competition following MASS.

However, Figure 1b shows that this rule can select an incorrect head even when the true task is retained, while routing scores remain informative for head selection. Motivated by these findings, we retain the aggregated task relevance scores after subspace selection and use them to guide head competition by weighting each candidate head's logits:

$$\tilde{o}_i(x) = s_i(x) o_i(x), \qquad i \in \Omega(x). \tag{11}$$

Finally, we select the task $i^\star$ and class $c^\star$ corresponding to the largest weighted logit across all candidate heads:

$$(i^\star, c^\star) = \underset{i\in\Omega(x),\, c\in\mathcal{C}_i}{\arg\max}\ \tilde{o}_i(x)[c]. \tag{12}$$

Algorithm 2 summarizes the complete procedure of dynamic model merging via MATE.

Table 1: Average classification accuracy (%) of model merging methods across three ViT backbones and different task scales. Bold indicates the best result among model merging methods.

	Method	ViT-B/32			ViT-B/16			ViT-L/14		
		8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
	Fine-tuned	90.31	89.07	89.51	92.42	91.37	91.92	94.23	93.43	94.00
Static	Task Arithmetic	68.82	64.62	64.03	72.96	70.64	69.01	84.39	80.37	76.85
Static	TSV-M	83.21	78.62	75.55	85.52	81.38	78.80	91.25	88.78	87.49
Static	Iso-C	82.85	78.44	73.23	87.47	79.82	75.29	92.58	89.64	86.82
Static	Iso-CTS	82.03	80.58	76.99	88.67	84.11	80.69	92.81	91.08	89.24
Dynamic	WEMoE	88.87	74.09	68.61	90.72	76.26	66.12	92.92	71.34	68.53
Dynamic	SMILE	83.27	77.29	72.87	87.89	81.78	79.51	91.31	86.61	84.93
Dynamic	MASS	87.08	82.90	81.17	90.60	87.83	81.11	92.97	90.95	90.84
Dynamic	**MATE**	**88.91**	**85.09**	**82.25**	**91.44**	**88.55**	**82.95**	**93.66**	**91.58**	**91.00**

4 EXPERIMENTS

In this section, we evaluate MATE against static and dynamic model merging baselines across three ViT backbones and three task scales. We further investigate its routing behavior and examine the contributions of multi-layer aggregation and routing-guided head decisions through routing analyses and ablation studies.

4.1 EXPERIMENTAL SETUP

Following the experimental setup of MASS (Crisostomi et al., 2026), we evaluate model merging in the 8-task, 14-task, and 20-task visual classification settings using CLIP (Radford et al., 2021) with ViT-B/32, ViT-B/16, and ViT-L/14 backbones. More details on the datasets for each task are provided in Appendix C. We report the average test accuracy across tasks. Baselines include static merging methods: Task Arithmetic (TA) (Ilharco et al., 2023), TSV-M (Gargiulo et al., 2025), Iso-C and Iso-CTS (Marczak et al., 2025), and dynamic merging methods: WEMoE (Tang et al., 2024), SMILE (Tang et al., 2025), and MASS (Crisostomi et al., 2026). To evaluate WEMoE and SMILE without task identity, we also follow MASS in selecting classification heads by majority voting over their layer-wise routing decisions. Independently fine-tuned models serve as a reference. All merging methods use the same task models and evaluation configurations.

For MATE, we construct each proxy dataset $\mathcal{D}_i^{\text{proxy}}$ by randomly sampling 32 examples from task i's training set. We select $K_L = 5$ MLP routing layers with the highest relative projection residual margins averaged across tasks. Following MASS, we set the routing threshold to $\eta = 0.20$ and the maximum number of candidate tasks to $K_T = 3$ in all main experiments. Appendix D examines the sensitivity of MATE to these hyperparameters.

4.2 MULTI-TASK MODEL MERGING PERFORMANCE

Table 1 reports average classification accuracy across three ViT backbones in the 8-task, 14-task, and 20-task settings. MATE achieves the highest accuracy among model merging methods in all nine configurations, outperforming both static and dynamic baselines. Across the three task scales, its maximum gains over the strongest baseline are 2.19, 1.84, and 0.69 percentage points on ViT-B/32, ViT-B/16, and ViT-L/14, respectively. As the number of tasks increases, input-dependent merging requires selecting relevant knowledge from a broader set of task models. MATE retains its performance advantage in the 14-task and 20-task settings across all three backbones, supporting its effectiveness across the evaluated task scales.

4.3 ROUTING PERFORMANCE

To better understand MATE's performance gains, we also compare its multi-layer routing aggregation with MASS's single-layer routing using top-1 task accuracy and task survival. Top-1 task accuracy measures the proportion of samples for which the highest-scoring task matches the input's

Table 2: Routing performance of the single-layer baseline and MATE across ViT backbones and task scales. **(a)** Top-1 task accuracy; **(b)** Task survival. Bold indicates the better result.

(a) Top-1 Task Accuracy (%)

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Single-Layer (MASS)	97.34	94.25	84.08	99.11	96.86	83.08	99.46	97.99	90.17
MATE	**98.69**	**96.27**	**86.28**	**99.32**	**97.67**	**84.50**	**99.63**	**98.32**	**91.12**

(b) Task Survival (%)

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Single-Layer (MASS)	99.67	96.72	91.57	99.88	98.65	89.15	99.91	99.24	95.19
MATE	**99.73**	**98.00**	**92.66**	**99.93**	**98.90**	**91.14**	**99.95**	**99.47**	**96.14**

true task. Task survival measures the proportion of samples for which the true task is retained in the candidate set $\Omega(x)$, allowing its knowledge to participate in subsequent merging even when it is not ranked first.

Table 2 shows that MATE outperforms MASS in both top-1 task accuracy and task survival across all nine configurations. On ViT-B/32, top-1 task accuracy reaches 98.69%, 96.27%, and 86.28% in the 8-task, 14-task, and 20-task settings, respectively, with a gain of 2.20 percentage points in the 20-task setting. The improvements extend to ViT-B/16 and ViT-L/14. Task survival also improves consistently, rising from 89.15% to 91.14% on ViT-B/16 in the 20-task setting.

As the number of tasks increases, top-1 task accuracy declines more sharply than task survival. For example, in the ViT-B/16 20-task setting, MATE achieves a top-1 task accuracy of 84.50% and a task survival of 91.14%. This gap shows that the true task can remain in the candidate set $\Omega(x)$ without receiving the highest routing score. Once retained, its task subspace participates in adaptive merging and its classification head enters the final prediction competition, preserving the opportunity for a correct prediction despite a task-ranking error.

Together, these results indicate that multi-layer aggregation provides a more reliable signal for task selection than single-layer routing across all nine configurations. By incorporating task-relevant evidence that a single layer can overlook, it reduces reliance on any individual layer. The resulting improvements in task identification and true-task retention help explain MATE's gains in final classification accuracy.

4.4 Routing Separability of Attention and MLP Layers

Our main experiments use MLP layers for routing. To examine this choice, we additionally compare attention and MLP layers in the ViT-B/32 8-task setting using the relative projection residual margin averaged across tasks at each Transformer block. Figure 2 shows that attention layers have higher mean margins in the early blocks, but these remain negative or near zero, indicating limited task separability. In blocks 6–11, MLP layers instead achieve larger positive margins than attention layers within the same blocks. Since each MLP sublayer receives hidden states already updated by self-attention, it also provides a natural location for routing within the block. Motivated by these observations, we use MLP layers as the candidate pool and select informative layers, drawing routing evidence from distinct blocks while retaining a single representation source per block.

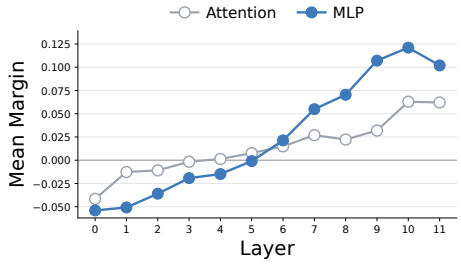


Figure 2: Task-averaged relative projection residual margins in the ViT-B/32 8-task setting.

Table 3: Average classification accuracy (%) with different combinations of multi-layer routing aggregation and routing-guided head decision. ✓ and × indicate enabled and disabled components. Bold indicates the best result in each configuration.

Multi-Layer Routing Aggregation	Routing-Guided Head Decision	ViT-B/32			ViT-B/16			ViT-L/14		
		8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
×	×	87.08	82.90	81.17	90.60	87.83	81.11	92.97	90.95	90.84
×	✓	88.52	83.77	81.43	91.31	88.17	80.95	93.55	91.39	90.55
✓	×	87.34	84.34	81.82	90.35	88.20	**83.36**	93.00	91.08	**91.11**
✓	✓	**88.91**	**85.09**	**82.25**	**91.44**	**88.55**	82.95	**93.66**	**91.58**	91.00

4.5 Ablation Study

Finally, we evaluate four combinations of multi-layer routing aggregation and routing-guided head decision. Disabling aggregation retains single-layer routing, while disabling head guidance uses simple max-logit competition among candidate heads. Disabling both recovers MASS, while enabling both yields the full MATE. As shown in Table 3, enabling multi-layer routing aggregation alone improves classification accuracy in eight of the nine configurations. In the ViT-B/16 20-task setting, accuracy rises from 81.11% to 83.36%, a gain of 2.25 percentage points. Complementing the direct evaluation of top-1 task accuracy and task survival in Section 4.3, this ablation shows that multi-layer aggregation also benefits final classification accuracy.

Enabling routing-guided head decision alone improves performance in seven of the nine configurations. For example, accuracy reaches 88.52% and 93.55% in the ViT-B/32 and ViT-L/14 8-task settings. These gains indicate that routing scores retain useful task relevance information after candidate selection and can guide the final competition among classification heads. Nevertheless, small decreases of 0.16 and 0.29 percentage points occur in the ViT-B/16 and ViT-L/14 20-task settings. Compared with the 14-task setting, this larger task set introduces additional visual overlap, with CIFAR10 added alongside CIFAR100 and STL10, and EMNIST and KMNIST alongside MNIST. Such overlap can reduce the discriminative power of routing scores within the candidate set. When task rankings become less reliable, routing-score weighting may shift the balance among candidate heads and slightly reduce accuracy.

With both components enabled, MATE outperforms MASS in all nine configurations and achieves the highest accuracy among the ablation variants in seven, indicating complementary benefits in most configurations. In the ViT-B/16 and ViT-L/14 20-task settings, however, adding routing-guided head decision to multi-layer routing aggregation slightly reduces accuracy. This is consistent with the preceding analysis: less discriminative routing scores among visually related candidate tasks can limit the benefit of head weighting, even after multi-layer aggregation.

5 Conclusion and Future Work

In this paper, we present MATE, a unified and improved mechanism for task and head routing in dynamic model merging without prior task information. Our analysis reveals that different layers provide complementary evidence for task identification and that routing scores also inform classification head selection. We first introduce the relative projection residual margin to quantify layer-wise task separability. Based on this measure, MATE selects informative layers and aggregates their routing scores into a task relevance signal for adaptive task subspace selection and routing-guided head decisions. Experiments across three CLIP ViT backbones and three task scales show that MATE achieves the best model merging accuracy among the evaluated methods in all nine configurations, alongside consistent improvements in task identification and true-task retention.

There are still some directions for future work. First, MATE relies on a small proxy dataset for layer selection. It is interesting to explore fully data-free methods for estimating layer-wise separability. Second, our ablation study shows that the gains from head weighting vary across task configurations. Thus, exploring alternative or complementary evidence for head routing may lead to more consistent improvements in final performance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with drafting and polishing the manuscript. This assistance included improving grammar and word choice and refining sentence structure and paragraph transitions. We have not used generative AI tools to generate the core research ideas and conclusions of this paper. We have reviewed all AI-assisted work to ensure that it accurately reflected the intended technical meaning and supporting evidence. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the method, experimental settings, and hyperparameters needed to reproduce our results in the main text and appendices. Our code is included in the supplementary materials, and a link to the public code repository will be provided upon acceptance.

REFERENCES

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pp. 446–461. Springer, 2014. doi: 10.1007/978-3-319-10599-4_29. 16

Levy Chaves, Chao Zhou, Rebekka Burkholz, Eduardo Valle, and Sandra Avila. Bridging domains through subspace-aware model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7990–7999, 2026. 1

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. doi: 10.1109/JPROC.2017.2675998. 16

Jiho Choi, Donggyun Kim, Chanhyuk Lee, and Seunghoon Hong. Revisiting weight averaging for model merging. *arXiv preprint arXiv:2412.12153*, 2024. 1

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014. doi: 10.1109/CVPR.2014.461. 16

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical Japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. 16

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223. PMLR, 2011. 16

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 2921–2926, 2017. doi: 10.1109/IJCNN.2017.7966217. 16

Donato Crisostomi, Alessandro Zirilli, Antonio Andrea Gargiulo, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, Iacopo Masi, and Emanuele Rodolà. MASS: MoErging through adaptive subspace selection. In *International Conference on Learning Representations*, volume 2026, pp. 19982–20008, 2026. 2, 4, 7, 16

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. 1

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, 2025. 1, 2, 3, 4, 5, 7

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing*, volume 8228 of *Lecture Notes in Computer Science*, pp. 117–124. Springer, 2013. doi: 10.1007/978-3-642-42051-1_16. 16

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242. 16

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019. 1

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, volume 35, 2022. 1

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with Task Arithmetic. In *International Conference on Learning Representations*, 2023. 1, 3, 7, 16

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022. 1

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77. 16

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 16

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 16

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Model merging in the essential subspace. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 1

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-Merging: Dynamic integration of modular expertise in model merging. In *Advances in Neural Information Processing Systems*, volume 37, 2024. 2

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Proceedings of the International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43177–43199, 2025. 1, 7

Michael S Matena and Colin Raffel. Merging models with Fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022. 1

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 16

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, 2020. 1

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47. 16

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. *Advances in Neural Information Processing Systems*, 38:61793–61825, 2026. 1

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092. 16

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021. 3, 7

Li Shen, Anke Tang, Enneng Yang, Guibing Guo, Yong Luo, Lefei Zhang, Xiaochun Cao, Bo Du, and Dacheng Tao. Efficient and effective weight-ensembling mixture of experts for multi-task model merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020. 1

Ronald Skorobogat, Karsten Roth, and Mariana-Iuliana Georgescu. Subspace-boosted model merging. *arXiv preprint arXiv:2506.16506*, 2025. 1

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013. 16

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German traffic sign recognition benchmark: A multi-class classification competition. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1453–1460, 2011. doi: 10.1109/IJCNN.2011.6033395. 16

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the Knots. In *International Conference on Learning Representations*, volume 2025, pp. 4501–4519, 2025. 1

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *Proceedings of the International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47778–47799, 2024. 2, 7

Anke Tang, Li Shen, Yong Luo, Shuai Xie, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Zero-shot sparse mixture of low-rank experts construction from pre-trained foundation models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2, 7

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, volume 11071 of *Lecture Notes in Computer Science*, pp. 210–218. Springer, 2018. doi: 10.1007/978-3-030-00934-2_24. 16

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022a. 1

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022b. 1

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 16

Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119 (1):3–22, 2016. doi: 10.1007/s11263-014-0748-y. 16

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023. 1

Prateek Yadav, Colin Raffel, Mohammed Muqeeth, Lucas Caccia, Haokun Liu, Tianlong Chen, Mohit Bansal, Leshem Choshen, and Alessandro Sordoni. A survey on model MoErging: Recycling and routing among specialized experts for collaborative learning. *arXiv preprint arXiv:2408.07057*, 2024. 2

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024. 1

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

Peng Ye, Chenyu Huang, Mingzhu Shen, Tao Chen, Yongqi Huang, and Wanli Ouyang. Dynamic model merging with mixture of weights. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. doi: 10.1109/TCSVT.2025.3544680. 2

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024. 1

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. DC-Merge: Improving model merging with directional consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. 1

Hongling Zheng, Li Shen, Anke Tang, Yong Luo, Han Hu, Bo Du, Yonggang Wen, and Dacheng Tao. Learning from models beyond fine-tuning. *Nature Machine Intelligence*, 7(1):6–17, 2025. 1

A Metric Definitions for Routing and Head Selection

We conduct the analysis in Figure 1b using ViT-B/32 in the 8-task setting. For each task i, all three metrics are computed over $\mathcal{S}_i = \{x \in \mathcal{D}_i^{\text{test}} \mid |\Omega(x)| > 1,\ i \in \Omega(x)\}$, where $\mathcal{D}_i^{\text{test}}$ denotes its test set.

Router. Given routing scores $s_j(x)$, Router reports the proportion of samples for which the true task ranks highest:

$$R_i^{\text{Router}} = \frac{1}{|\mathcal{S}_i|} \sum_{x \in \mathcal{S}_i} \mathbf{1}\left[i = \arg\max_{j \in \Omega(x)} s_j(x) \right]. \tag{13}$$

Simple and Ours. Let $o_j(x)[c]$ denote the logit for class $c \in \mathcal{C}_j$ from head j. Simple and Ours respectively select heads through simple and routing-guided max-logit competition:

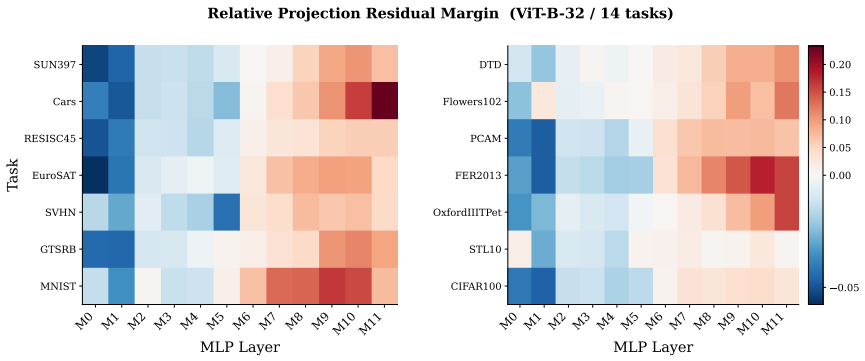
$$\hat{i}_{\text{Simple}}(x) = \arg\max_{j \in \Omega(x)} \max_{c \in \mathcal{C}_j} o_j(x)[c], \tag{14}$$

$$\hat{i}_{\text{Ours}}(x) = \arg\max_{j \in \Omega(x)} \max_{c \in \mathcal{C}_j} s_j(x) o_j(x)[c]. \tag{15}$$

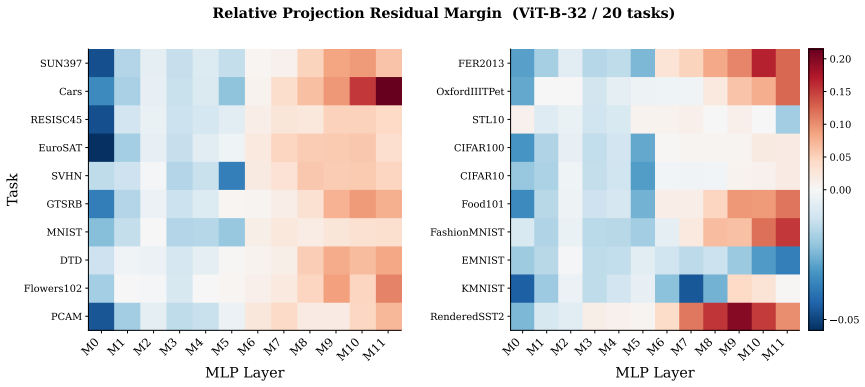
For both, we report the true-task head's win rate:

$$R_i^a = \frac{1}{|\mathcal{S}_i|} \sum_{x \in \mathcal{S}_i} \mathbf{1}\left[\hat{i}_a(x) = i \right], \qquad a \in \{\text{Simple}, \text{Ours}\}. \tag{16}$$

B Additional Layer-wise Routing Analysis

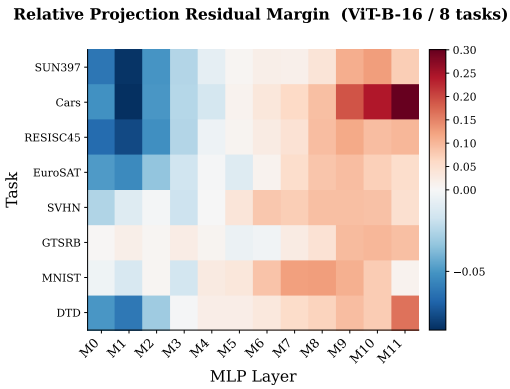


(a) 14-task

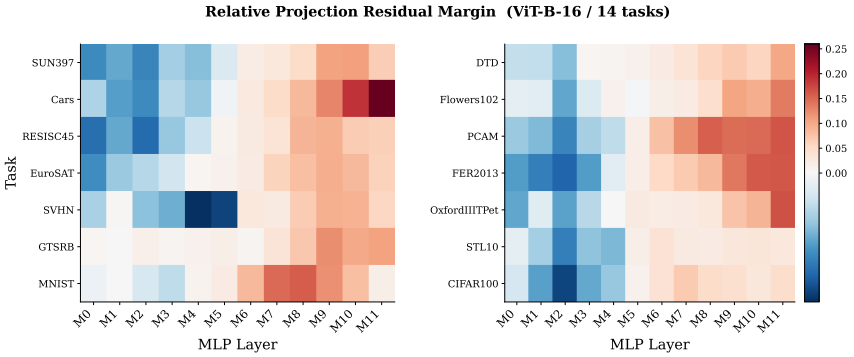


(b) 20-task

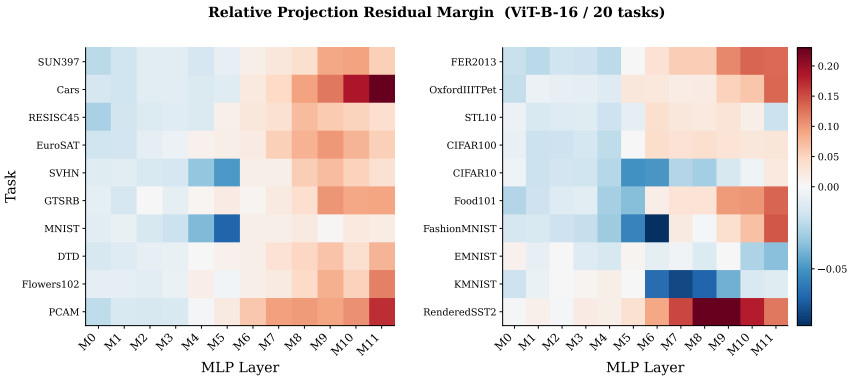
Figure 3: Relative projection residual margins across MLP layers for ViT-B/32.



(a) 8-task



(b) 14-task



(c) 20-task

Figure 4: Relative projection residual margins across MLP layers for ViT-B/16.

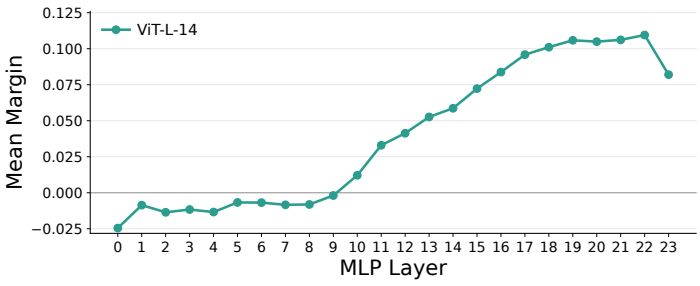


Figure 5: Relative projection residual margins averaged across tasks at each MLP layer of ViT-L/14 in the 20-task setting.

We extend the analysis in Figure 1a to additional task scales and backbones. Figures 3 and 4 show heatmaps for ViT-B/32 with 14 and 20 tasks and ViT-B/16 with 8, 14, and 20 tasks. Each cell reports $A_{i,\ell}$ from Equation (7), with larger values indicating clearer task separation on average. The heatmaps further illustrate task-dependent layer preferences. For example, in both 20-task settings, Cars exhibits large positive margins in the final MLP layers, whereas EMNIST has lower margins in these layers. These contrasting patterns indicate that layers informative for one task may be less useful for another, further motivating the aggregation of routing evidence from multiple informative layers.

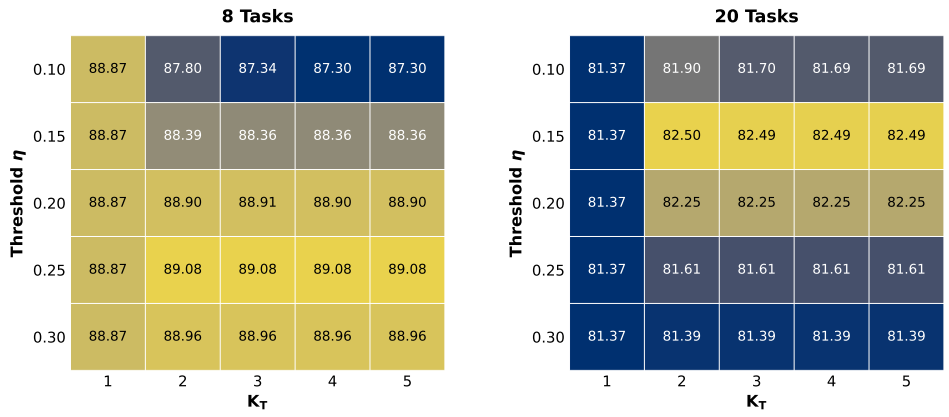
Additionally, we examine ViT-L/14 in the 20-task setting. For a compact view across its 24 MLP layers, Figure 5 plots the task-averaged margin $\bar{A}_\ell$, which is negative in early layers and generally increases in later layers.

C Benchmarks and Datasets

Table 4: Datasets used in the three model merging benchmarks. The final column lists the task scales at which each dataset is included.

Dataset	Task scales
Cars (Krause et al., 2013)	8, 14, 20
DTD (Cimpoi et al., 2014)	
EuroSAT (Helber et al., 2019)	
GTSRB (Stallkamp et al., 2011)	
MNIST (LeCun et al., 1998)	
RESISC45 (Cheng et al., 2017)	
SUN397 (Xiao et al., 2016)	
SVHN (Netzer et al., 2011)	
CIFAR100 (Krizhevsky, 2009)	14, 20
STL10 (Coates et al., 2011)	
Flowers102 (Nilsback & Zisserman, 2008)	
OxfordIIITPet (Parkhi et al., 2012)	
PCAM (Veeling et al., 2018)	
FER2013 (Goodfellow et al., 2013)	
CIFAR10 (Krizhevsky, 2009)	20
Food101 (Bossard et al., 2014)	
FashionMNIST (Xiao et al., 2017)	
EMNIST (Cohen et al., 2017)	
KMNIST (Clanuwat et al., 2018)	
RenderedSST2 (Socher et al., 2013)	

We evaluate MATE on three visual classification benchmarks containing 8, 14, and 20 tasks, following MASS (Crisostomi et al., 2026). The 8-task benchmark follows the task collection used in Task Arithmetic (Ilharco et al., 2023). Each larger benchmark retains all tasks from the preceding setting and adds six datasets. Table 4 lists all 20 datasets and their benchmark membership. The benchmarks cover natural images, remote sensing imagery, handwritten symbols, and medical images. Alongside this diversity, the larger task collections include related recognition problems, such as object recognition across CIFAR10, CIFAR100, and STL10, and handwritten symbol recognition across MNIST, EMNIST, and KMNIST. These settings therefore evaluate model merging across both visually distinct and related tasks.



(a) 8-task (b) 20-task

Figure 6: Sensitivity to the routing threshold η and the maximum candidate task count K_T on ViT-B/32.

D Hyperparameter Sensitivity

We further analyze the effects of the routing threshold η and the maximum number of candidate tasks K_T in the ViT-B/32 8-task and 20-task settings. As shown in Figure 6, MATE maintains stable performance across a broad parameter range, with favorable results near $\eta = 0.25$ and $\eta = 0.15$ in the two settings, indicating that the preferred routing threshold varies with task scale.

K_T controls the maximum size of the candidate task set $\Omega(x)$. When the additional tasks have routing scores below η, increasing K_T does not enlarge the actual candidate set, so the subsequent adaptive merging and predictions remain unchanged. For example, for all $K_T \in \{2, 3, 4, 5\}$, the 8-task setting achieves 89.08% at $\eta = 0.25$, while the 20-task setting achieves 82.25% at $\eta = 0.20$.

Our default setting, $\eta = 0.20$ and $K_T = 3$, lies in a high-performing region for both task scales, with accuracy only 0.17 and 0.25 percentage points below the best results in the 8-task and 20-task settings.