DEBTMERGE: LONG-HORIZON EXPERT EVOLUTION FOR CONTINUAL MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual model merging (CMM) sequentially integrates task-specific experts without joint retraining. Existing methods mainly make local merge decisions based on the incoming task and current expert state, without accounting for the cumulative effects of past merges. Over long task streams, this can lead to persistent historical perturbation, insufficient integration of incoming knowledge, and growing expert-capacity pressure. Jointly optimizing expert selection and update parameters is also constrained by costly candidate evaluations. To address these, we introduce **DebtMerge**, a long-horizon optimization framework for expert-managed CMM that jointly addresses long-horizon control and finite-budget merge optimization. DebtMerge formulates CMM as an online problem with time-average targets and uses Lyapunov virtual queues to represent accumulated *stability*, *plasticity*, and *resource debts*. These debts induce a per-arrival objective, while stability and plasticity debts guide geometry initialization without future task information. To optimize this objective under a fixed evaluation budget, DebtMerge uses evolutionary transfer optimization over merge recipes specifying expert structure, knowledge composition, and integration geometry. Compatible historical recipe blocks are adapted to the current expert bank and debt state, with objective feedback guiding their reuse toward promising combinations. We characterize debt-guided directional control, incorporate finite-budget solver error into the long-horizon control bound, and derive sufficient conditions for preserving historical predictions under stated assumptions. Experiments across vision and language models show improved average performance over the compared merging baselines.

1 INTRODUCTION

Large pretrained models are commonly adapted independently to different tasks, producing task-specific models or parameter modules that we refer to as experts. Model merging combines knowledge from these experts without joint retraining (Matena & Raffel, 2022; Yu et al., 2024). Existing approaches (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Shen et al., 2026; Zhou et al., 2026) typically assume that all participating models are available simultaneously. In continual model merging (CMM), experts instead arrive sequentially, requiring the system to preserve historical knowledge while integrating each new expert. Projection-based methods such as OPCM (Tang et al., 2025) and DOP (Yang et al., 2025), together with null-space filtering in NUFILT (Qiu et al., 2026b), address interference between incoming and historical knowledge. Expert-managed CMM methods such as MINGLE (Qiu et al., 2025) and MADE-IT (Qiu et al., 2026a) maintain an expert bank of retained experts and design a routing mechanism that selects these experts during prediction.

Despite these advances, existing methods primarily regulate interference and expert allocation through local merge decisions based on the incoming task and current expert state. This state records what is retained, but does not explicitly quantify the cumulative effects of past merges. As illustrated in Fig. 1a, even when these merge decisions appear reasonable, historical perturbation, incoming-knowledge mismatch, and expert-capacity pressure can persist or accumulate (Yang et al., 2025; Shenaj et al., 2026). Quantifying these cumulative effects would allow subsequent merges to prioritize historical protection, incoming-knowledge integration, or expert capacity (Neely, 2010). This motivates the first challenge, **long-horizon online control**: tracking cumulative effects to guide subsequent decisions when future tasks are unknown. Additionally, this control requires jointly deciding which experts to retain and how to combine and filter their updates. These decisions couple

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.

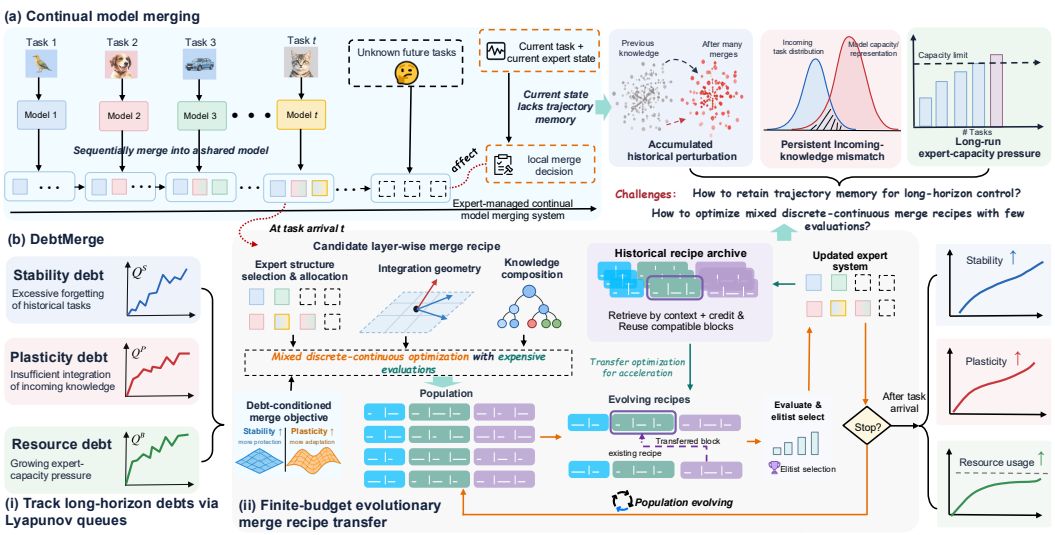


Fig. 1: Overview of DebtMerge. (a) Continual merging under unknown future tasks can accumulate historical perturbation, incoming-knowledge mismatch, and expert-capacity pressure. (b) DebtMerge couples two components: (i) Lyapunov virtual queues track stability, plasticity, and resource debts to construct a debt-conditioned merge objective; (ii) finite-budget evolutionary search optimizes layer-wise merge recipes over expert structure, knowledge composition, and integration geometry. Compatible historical recipe blocks are retrieved using context and transfer credit, reused in the evolving population, and re-evaluated under the current objective. The selected recipe updates the expert system, while the realized outcomes update the debts for subsequent arrivals.

discrete expert choices with continuous update parameters (Akiba et al., 2025), while their feasible combinations change as the expert bank evolves (Shenaj et al., 2026). Because evaluating candidate merges can be computationally expensive, only a limited number of combinations can be assessed at each arrival (Lee et al., 2025). This creates the second challenge, **finite-budget structured optimization**: efficiently searching these coupled decisions within a limited evaluation budget.

To address these challenges, we propose **DebtMerge**, a long-horizon optimization framework for expert-managed CMM, as illustrated in Fig. 1. For long-horizon online control, we formulate CMM as an online optimization problem with time-average targets on historical perturbation, incoming-knowledge mismatch, and expert occupancy. Using Lyapunov drift-plus-penalty optimization (Neely, 2006; 2010; Bi et al., 2021), DebtMerge maintains three virtual queues whose states are defined as *stability*, *plasticity*, and *resource debts*. The resulting debt-conditioned objective converts long-term constraints into per-arrival decisions, allowing accumulated deviations to guide merging without future task information. Stability and plasticity debts further initialize the integration geometry along historically important directions: for fixed historical sensitivities, larger stability debt favors protection, whereas larger plasticity debt favors incoming-knowledge transmission.

To solve the second challenge, DebtMerge employs evolutionary transfer optimization (Tan et al., 2021; Xue et al., 2025) to minimize the debt-conditioned objective under a fixed evaluation budget. Each candidate action is encoded as a *merge recipe* comprising three semantic blocks: expert structure, knowledge composition for combining updates, and integration geometry for filtering and scaling them. DebtMerge reuses compatible blocks from historical recipes and adapts them to the current expert bank and debt state, focusing limited evaluations on promising and feasible combinations. Feedback from the current objective further adjusts reuse according to its utility, encouraging beneficial transfer while suppressing harmful transfer. We quantify solver error as the objective gap between the selected and best feasible recipes at each arrival. Under explicit assumptions, reducing the accumulated solver error tightens the control bound for the same realized sequence of pre-merge states, motivating more effective finite-budget search.

We further derive sufficient conditions for preserving historical predictions under explicit assumptions on expert outputs and routing. Finally, experiments on vision and language models evaluate DebtMerge across varying task-stream lengths, expert capacities, and search budgets. Our main contributions are summarized as follows:

- **Debt-based long-horizon control for expert-managed CMM.** We formulate CMM as a long-horizon constrained online problem and derive a debt-conditioned objective for each task arrival. Stability, plasticity, and resource debts retain accumulated violations to regulate subsequent merge decisions.
- **Finite-budget evolutionary optimization with historical recipe transfer.** We develop an evolutionary solver that jointly optimizes expert structure, knowledge composition, and integration geometry under a fixed evaluation budget. Historical recipe blocks are adapted to the current expert bank and debt state, with objective feedback guiding their reuse toward promising combinations.
- **Theoretical characterization and empirical evaluation.** We establish the monotonic effects of stability and plasticity debts on debt-guided geometry initialization and explicitly incorporate finite-budget solver error into the control bound. Under stated assumptions, we also derive sufficient conditions for historical prediction preservation. Experiments evaluate these mechanisms across vision and language models, varying task-stream lengths, expert capacities, and search budgets.

2 Related Work

Model merging combines independently adapted models through parameter averaging, task-vector composition, or interference-aware integration (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023), typically assuming simultaneous access to the source models. CMM extends this setting to sequential arrivals, where historical knowledge must be retained while incoming knowledge is integrated under limited capacity. Existing approaches address this trade-off through interference suppression (Tang et al., 2025; Yang et al., 2025; Qiu et al., 2026b), temporal adaptation (Qiao & Mahdavi, 2026; Lin & Kang, 2026), and expert management (Qiu et al., 2025; Shenaj et al., 2026; Qiu et al., 2026a). These mechanisms capture aspects of merge history or expert state, but do not jointly track accumulated deviations from long-term targets for historical perturbation, incoming-knowledge mismatch, and expert occupancy. DebtMerge addresses this gap by formulating CMM as a constrained online problem and using these deviations to coordinate expert structure, knowledge composition, and integration geometry. Its control objective is derived from Lyapunov drift-plus-penalty optimization (Neely, 2010), which has also been applied to replay-based continual learning (Shah et al., 2026) and sequential knowledge editing in large language models (Wang et al., 2025). These applications regulate learning or editing within a model, whereas DebtMerge maintains stability, plasticity, and resource debts to regulate merging within an evolving expert bank. For the resulting finite-budget optimization, it adapts evolutionary transfer optimization (Tan et al., 2021; Huang et al., 2023; Feng et al., 2017; 2019) to reuse compatible historical recipe blocks under the current debt-conditioned objective and feasible expert configurations.

3 Problem Formulation

Expert-Managed CMM. Expert-managed CMM integrates sequentially arriving task-specialized models into a maintained expert system (Qiu et al., 2025; 2026a; Shenaj et al., 2026). At task arrival t, the incoming model θ_t^{task} is assumed to be parameter-compatible with a common pretrained model θ_0. Relative to this shared model, its parameter update is expressed as $d_t = \theta_t^{\text{task}} - \theta_0$ following task-vector arithmetic (Ilharco et al., 2023). Here, the maintained system consists of a bounded-capacity expert bank $\mathcal{E}_{t-1}$ and a routing mechanism $\mathcal{R}_{t-1}$, which together form the pre-merge state $X_t = (\mathcal{E}_{t-1}, \mathcal{R}_{t-1})$. Each merge uses the incoming update d_t and state X_t, without access to historical task training data or labels. The admissible expert-management and knowledge-integration decisions form a state-dependent feasible set $\mathcal{A}(X_t, d_t)$. A merge action a_t is selected from this set to update the expert system according to $X_{t+1} = \mathcal{M}(X_t, d_t, a_t)$, where $\mathcal{M}$ performs the corresponding expert-management and merging operations. The updated state X_{t+1} specifies the expert configuration available at the next arrival and therefore constrains subsequent merge decisions, making these decisions temporally coupled.

Long-Horizon Objective and Constraints. Motivated by the data-free interference signals studied in existing work (Tang et al., 2025; Yang et al., 2025; Qiu et al., 2026b), we introduce three data-free optimization proxies for historical perturbation, incoming-knowledge mismatch, and expert

occupancy, which are defined as

$$F_t(a) = \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{\|b_{t,i}(a) - b_{t-1,i}\|_2^2}{\|b_{t-1,i}\|_2^2 + \varepsilon}, \qquad P_t(a) = \frac{\|b_{t,t}(a) - d_t\|_2^2}{\|d_t\|_2^2 + \varepsilon}, \\ C_t(a) = \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\frac{m_{t,l}(a)}{M_l}, \tag{1}$$

where $\varepsilon > 0$ is a numerical constant preventing zero denominators, $\mathcal{L}$ is the set of managed layers, $m_{t,l}(a)$ is the number of active experts at layer l after applying a, M_l is the corresponding hard capacity limit, $\theta_{t,i}(a)$ is the effective parameter branch associated with task i after applying a, and $b_{t,i}(a) = \theta_{t,i}(a) - \theta_0$ is its update relative to the pretrained model. For a historical task $i < t$, $b_{t-1,i}$ is its corresponding update before the current merge. No historical tasks are present at the first arrival, so $F_1(a) = 0$. All three proxies are non-negative, with $C_t(a) \in [0, 1]$ under the capacity constraint. Because historical predictions depend on both expert outputs and routing, a small $F_t(a)$ alone does not guarantee their preservation. Under stated expert-mixture assumptions, **Appendix B.6** bounds prediction drift from changes in these two components and derives a sufficient condition for preserving previously correct predictions.

The proxies in Eq. 1 evaluate individual merges, but each action also alters the expert state and future feasible choices. We therefore impose time-average bounds on these proxies, allowing temporary deviations while constraining their long-run averages. The objective then minimizes the average incoming-knowledge mismatch and expert-occupancy costs.

$$\begin{aligned} \mathcal{P}1: \quad \min_{\{a_t\}_{t\geq 1}} \quad & \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\ell_t(a_t) = \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\big(P_t(a_t) + \omega C_t(a_t)\big) \\ \text{s.t.} \quad & \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}F_t(a_t) \leq \epsilon_S, \\ & \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}P_t(a_t) \leq \epsilon_P, \\ & \limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}C_t(a_t) \leq C_0, \\ & a_t \in \mathcal{A}(X_t, d_t), \qquad \forall t. \end{aligned} \tag{2}$$

where $\ell_t(a_t)$ is the nominal per-arrival cost, $\omega > 0$ weights expert occupancy relative to incoming-knowledge mismatch, and ϵ_S, ϵ_P, and C_0 bound the long-run averages of F_t, P_t, and C_t, respectively.

4 DEBTMERGE

Solving $\mathcal{P}1$ online is challenging because merge decisions must account for long-term constraints without future task information, while searching an evolving mixed discrete–continuous action space under limited evaluations. To address this problem, **DebtMerge** couples debt-based long-horizon control with evolutionary transfer optimization, as illustrated in Fig. 1. Lyapunov virtual queues first transform accumulated constraint deviations into a debt-conditioned per-arrival objective. Each merge is then represented as a layer-wise recipe over expert structure, knowledge composition, and integration geometry, with stability and plasticity debts guiding geometry initialization. DebtMerge searches this space under a fixed evaluation budget by combining evolutionary transfer optimization with compatible blocks from historical recipes. The selected recipe updates the expert system, and its realized outcomes update the debts and recipe archive for subsequent arrivals. The complete workflow is provided in **Appendix A**.

4.1 DEBT-BASED ONLINE CONTROL

To track long-run constraint violations, we define $u_t(a) = [F_t(a) - \epsilon_S,\ P_t(a) - \epsilon_P,\ C_t(a) - C_0]^\top$ and the merge-debt state $Q_t = [Q_t^S, Q_t^P, Q_t^B]^\top$, corresponding to accumulated stability, plasticity, and

resource violations. Following the standard virtual-queue construction for time-average constrained optimization (Neely, 2010),

$$Q_{t+1} = [Q_t + u_t(a_t)]_+ , \tag{3}$$

where $[\cdot]_+$ denotes element-wise projection onto the non-negative orthant. Unlike the expert state X_t, Q_t retains information about accumulated constraint violations.

Proposition 1 (Debt stability and long-run feasibility)**.** *If $\|Q_{T+1}\|_2/T \to 0$ as $T \to \infty$, then the time-average constraints in Eq. 2 are satisfied.*

The proof is given in **Appendix B.1**.

Using the quadratic Lyapunov function $L(Q_t) = \frac{1}{2}\|Q_t\|_2^2$, bounded one-step violations yield $\Delta_t(a) \leq B + Q_t^\top u_t(a)$ for some finite constant B (Neely, 2010). Minimizing the resulting drift-plus-penalty bound gives

$$\begin{aligned} a_t^\star &\in \operatorname*{arg\,min}_{a \in \mathcal{A}(X_t, d_t)} \Psi_t(a), \\ \Psi_t(a) &= Q_t^\top u_t(a) + V\ell_t(a) \\ &\equiv Q_t^S F_t(a) + \left(Q_t^P + V\right) P_t(a) + \left(Q_t^B + V\omega\right) C_t(a), \end{aligned} \tag{4}$$

where terms independent of a are omitted and $V > 0$ balances nominal merge quality against long-run constraint enforcement. Throughout candidate scoring and transfer feedback, Ψ_t refers to the non-negative expression on the final line of Eq. 4. Hence, larger stability, plasticity, and resource debts place greater emphasis on reducing historical perturbation, incoming-knowledge mismatch, and expert occupancy, respectively. For a fixed candidate pool, Proposition 3 in **Appendix B.2** further establishes that, holding the other terms fixed, increasing any one debt cannot increase the selected value of its corresponding proxy under exact minimization.

4.2 Debt-Guided Merge Recipes

At each arrival, the pre-merge expert state X_t and debt state Q_t remain fixed during candidate search. A candidate action specifies a layer-wise merge recipe

$$a_t^{(l)} = \left(o_t^{(l)}, q_{t,\mathrm{tar}}^{(l)}, q_{t,\mathrm{src}}^{(l)}, \mu_{t,d}^{(l)}, \mu_{t,r}^{(l)}, r_t^{(l)}, \lambda_t^{(l)}, \alpha_t^{(l)}\right), \tag{5}$$

where the variables respectively encode expert management, relative expert selection, knowledge composition, protected-subspace rank, directional protection, and update absorption. The complete action is $a_t = \{a_t^{(l)}\}_{l \in \mathcal{L}}$, yielding a state-dependent mixed discrete–continuous search space. Relative expert positions are defined with respect to the current similarity ranking, allowing recipes to remain transferable as the expert bank evolves. The composition variables combine incoming and historical expert information into a composed update $\widetilde{d}_t^{(l)}$. Detailed variable domains, relative-selection mappings, and composition rules are provided in **Appendix A.2**.

We next use stability and plasticity debts to initialize how $\widetilde{d}_t^{(l)}$ interacts with historically important parameter directions. For each layer, DebtMerge maintains an incremental low-rank spectral sketch $U_t^{(l)}\Lambda_t^{(l)}U_t^{(l)\top}$ of normalized historical task updates. Let $v_{t,k}^{(l)}$ denote its k-th direction, $h_{t,k}^{(l)}$ its normalized historical sensitivity, and $e_{t,k}^{(l)}$ the fraction of composed-update energy aligned with this direction. For a directional pass rate $g \in [0,1]$, we use the local stability–plasticity surrogate

$$\begin{aligned} \Phi_{t,k}^{(l)}(g) &= A_t e_{t,k}^{(l)}(1-g)^2 + B_t h_{t,k}^{(l)} e_{t,k}^{(l)} g^2, \\ A_t &= V + Q_t^P, \qquad B_t = Q_t^S, \\ g_{t,k}^{\mathrm{geo}} &= \operatorname*{arg\,min}_{g \in [0,1]} \Phi_{t,k}^{(l)}(g) = \frac{A_t}{A_t + B_t h_{t,k}^{(l)}}. \end{aligned} \tag{6}$$

The first term discourages suppressing incoming information, whereas the second penalizes transmission along historically sensitive directions. Resource debt does not enter this directional gate because it acts through the structural occupancy term $C_t(a)$.

The resulting gate provides an anchor for the searchable geometry $(r_t^{(l)}, \lambda_t^{(l)}, \alpha_t^{(l)})$, whose detailed construction is given in **Appendix A.3**. The composed update is injected as

$$\delta_t^{(l)} = \alpha_t^{(l)} \left[I - \lambda_t^{(l)} U_{t,\mathcal{S}}^{(l)} U_{t,\mathcal{S}}^{(l)\top} \right] \widetilde{d}_t^{(l)}, \tag{7}$$

where $U_{t,\mathcal{S}}^{(l)}$ contains the historical directions selected by the candidate protected rank. The geometry variables remain searchable and may deviate from the debt-guided anchor to reduce the exact objective Ψ_t.

Proposition 2 (Debt-guided directional control)**.** *For any active historical direction with $e_{t,k}^{(l)} > 0$, the surrogate in Eq. 6 is strictly convex and admits the unique minimizer $g_{t,k}^{\mathrm{geo}}$. For fixed V and $h_{t,k}^{(l)}$, $g_{t,k}^{\mathrm{geo}}$ is non-increasing in Q_t^S and non-decreasing in Q_t^P.*

Thus, larger stability debt increases protection along historically sensitive directions, whereas larger plasticity debt allows greater transmission of incoming information. The proof and the corresponding protected-set monotonicity are given in **Appendix B.2**.

4.3 Finite-Budget Evolutionary Transfer

The structured merge recipes specify the decisions optimized by the debt-conditioned objective Ψ_t. However, their mixed discrete–continuous search space evolves with the expert bank, while only a limited number of complete recipes can be evaluated at each arrival. We therefore use evolutionary optimization under a fixed budget of K complete-action evaluations. The initial population combines simple structural configurations with the debt-guided geometry anchor derived in the preceding subsection; detailed initialization and mixed-variable operators are provided in **Appendix A**.

A local evolutionary search treats each arrival independently and therefore does not exploit useful search experience accumulated from previous tasks. Directly reusing complete historical recipes is also inappropriate because expert identities, bank sizes, and feasible operations change over time. To retain transferable decision patterns instead, we decompose each layer-wise recipe in Eq. 5 into three semantic blocks: $\mathcal{G}_1 = (o, q_{\mathrm{tar}})$ for expert structure, $\mathcal{G}_2 = (q_{\mathrm{src}}, \mu_d, \mu_r)$ for knowledge composition, and $\mathcal{G}_3 = (r, \lambda, \alpha)$ for integration geometry. These blocks encode relative decisions rather than fixed expert identities, allowing historical search experience to be adapted to the current expert configuration.

Not all historical recipes are equally relevant to the current arrival. DebtMerge therefore maintains a historical recipe archive $\mathcal{B}_t = \{(c_i, a_i, \gamma_i)\}_{i<t}$, where c_i records the optimization context of arrival i, a_i is its committed recipe, and γ_i summarizes its observed transfer utility. The current context $c_t = [\bar{Q}_t^S, \bar{Q}_t^P, \bar{Q}_t^B, \zeta_t]$ combines the normalized debt state with compact statistics describing the current expert and parameter geometry. To favor historical experience that is both contextually relevant and empirically useful, source arrival i is sampled with probability

$$p_{i\to t} = \frac{\exp\big(\kappa_\gamma \gamma_i - \kappa_c \|c_i - c_t\|_2^2\big)}{\sum_{j<t} \exp(\kappa_\gamma \gamma_j - \kappa_c \|c_j - c_t\|_2^2)}. \tag{8}$$

The first term favors historically useful sources, whereas the second penalizes contexts that differ from the current debt and expert state. The construction and normalization of c_t are detailed in **Appendix A.4**.

After selecting a historical source, mixed-variable crossover and mutation first produce a local proposal a_t^{loc}. With probability π_t, a compatible semantic block from the sampled historical recipe is inserted into this proposal and adapted to the current expert bank, producing a transferred challenger a_t^{tr}. Both proposals are evaluated under the same current objective Ψ_t. Their relative objective difference provides target-side feedback for updating the source credit and transfer probability, so useful transfer is reinforced while harmful reuse is gradually suppressed. All evaluated proposals then compete through the same elitist selection. The corresponding reward, credit, and transfer-probability updates are given in **Appendix A.4**.

The search ultimately returns the best feasible recipe evaluated within the budget. We use $\mathcal{C}_t^{(K)} \subseteq \mathcal{A}(X_t, d_t)$ to represent this evaluated candidate set, giving

$$\hat{a}_t = \arg\min_{a \in \mathcal{C}_t^{(K)}} \Psi_t(a). \tag{9}$$

Because $\mathcal{C}_t^{(K)}$ covers only a finite subset of the feasible action space, the practical optimizer may differ from the exact per-arrival solution. We quantify this approximation by

$$\xi_t = \Psi_t(\hat{a}_t) - \inf_{a \in \mathcal{A}(X_t, d_t)} \Psi_t(a) \geq 0, \tag{10}$$

where ξ_t separates finite-budget search error from the debt-conditioned control objective itself.

Theorem 1 (Finite-budget long-horizon guarantee)**.** *Assume that $\frac{1}{2}\|u_t(a)\|_2^2 \leq B$ for every feasible action and task arrival. If, at each arrival t, there exists a feasible reference action $a_t^\dagger \in \mathcal{A}(X_t, d_t)$, evaluated on the same pre-merge state, such that $u_t(a_t^\dagger) \preceq 0$, then the finite-budget actions $\{\hat{a}_t\}_{t=1}^T$ satisfy*

$$\frac{1}{T}\sum_{t=1}^{T}\Big[\ell_t(\hat{a}_t) - \ell_t(a_t^\dagger)\Big] \leq \frac{B}{V} + \frac{L(Q_1)}{VT} + \frac{1}{VT}\sum_{t=1}^{T}\xi_t. \tag{11}$$

Theorem 1 separates the standard drift-plus-penalty term from the additional error introduced by finite-budget optimization. For the same realized sequence of pre-merge states, reducing the accumulated solver error $\sum_{t=1}^T \xi_t$ directly tightens the bound in Eq. 11, motivating the transfer of useful search experience when the evaluation budget is limited. The reference actions are evaluated on these same pre-merge states and therefore do not assume knowledge of future task arrivals.

Under the same assumptions, Corollary 1 further bounds finite-horizon debt and time-average constraint violations. For fixed V, $\sum_{t=1}^T \xi_t = o(T^2)$ implies rate-stable debts, while a uniformly bounded average solver error yields $\|Q_{T+1}\|_2/T = O(T^{-1/2})$. The proof of Theorem 1, the corollary, a more general finite-horizon comparator result, and the relation between search coverage and expected solver error are provided in **Appendices B.3–B.5**.

After the search, only $\hat{a}_t$ is committed to the maintained expert system. Its realized merge outcomes update the three debts through Eq. 3, while the committed recipe and its pre-merge context are added to the historical archive for subsequent transfer. The historical spectral sketch is updated from the raw incoming task update, so it continues to summarize the original task sequence rather than the debt-filtered candidate trajectory.

5 Experiments

Benchmarks. We evaluate CLIP (Radford et al., 2021) with ViT-B/32, ViT-B/16, and ViT-L/14 on vision streams of $T \in \{8, 14, 20\}$ tasks (Tang et al., 2024), RoBERTa-base (Liu et al., 2019) on seven GLUE tasks (Wang et al., 2018), and Qwen2.5-1.5B (Qwen Team, 2024) on ten heterogeneous language tasks. All methods use the same task-specific checkpoints and arrival orders within each comparison.

Baselines and Protocol. We compare against conventional merging baselines, OPCM (Tang et al., 2025), and the expert-managed methods MINGLE (Qiu et al., 2025) and MADE-IT (Qiu et al., 2026a). Main expert-managed experiments use capacity $M = 8$, with $M \in \{2, 4, 8\}$ for capacity stress tests. Additionally, DebtMerge uses $N_p = 10$, $\pi_0 = 0.5$, and $K = 150$ complete-action evaluations per arrival, with equal evaluation budgets for compared search variants.

Table 1: Main vision results: average accuracy (%) / BWT (percentage points).

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 Tasks	14 Tasks	20 Tasks	8 Tasks	14 Tasks	20 Tasks	8 Tasks	14 Tasks	20 Tasks
Pre-Trained	48.10 / -	56.90 / -	55.60 / -	55.40 / -	62.00 / -	59.80 / -	64.90 / -	69.10 / -	65.60 / -
Fine-Tuned	90.40 / -	89.30 / -	89.80 / -	92.40 / -	91.30 / -	91.60 / -	94.30 / -	93.40 / -	93.50 / -
C. Fine-Tuned	79.80 / -	67.40 / -	62.60 / -	82.90 / -	72.20 / -	68.24 / -	90.00 / -	70.90 / -	77.74 / -
Average (SWA)	66.30 / -11.50	65.40 / -8.00	61.12 / -7.10	72.30 / -9.70	69.70 / -7.10	64.85 / -7.30	80.00 / -7.30	77.50 / -5.80	71.11 / -6.40
C. Task Arithmetic	67.50 / -9.60	66.60 / -1.30	60.62 / -3.40	77.10 / -4.20	70.90 / -1.30	64.20 / -3.60	82.10 / -7.10	77.90 / -1.80	70.27 / -3.30
C. TIES-Merging	49.00 / -15.30	66.20 / 1.90	59.93 / -1.50	66.80 / -5.50	70.50 / 1.40	62.90 / -1.50	64.30 / -13.00	78.00 / -1.10	68.32 / -2.90
C. LW AdaMerging	53.40 / -32.50	59.80 / -24.10	57.55 / -22.70	59.90 / -27.80	64.30 / -22.10	61.41 / -21.40	68.80 / -24.30	73.10 / -19.60	66.93 / -21.70
C. LoRA-WEMoE	68.80 / -20.40	63.80 / -20.20	49.62 / -24.50	72.60 / -18.00	67.90 / -18.80	55.00 / -25.80	75.60 / -17.80	74.00 / -16.80	56.94 / -27.90
OPCM	75.50 / -6.30	71.90 / -6.00	65.72 / -7.80	81.80 / -4.80	77.10 / -5.10	70.29 / -6.30	87.00 / -2.60	83.50 / -4.30	76.08 / -6.50
MINGLE	85.80 / -0.60	81.60 / -1.10	77.12 / -2.20	88.30 / -0.40	84.90 / -0.90	82.09 / -1.90	91.80 / -0.60	88.80 / -1.00	85.60 / -2.60
MADE-IT	87.10 / -2.20	84.30 / -2.60	81.41 / -3.80	90.40 / -1.20	87.70 / -1.90	83.04 / -3.00	93.10 / -0.60	91.70 / -1.00	89.64 / -1.70
DebtMerge	**89.99 / -0.07**	**88.37 / -0.54**	**88.93 / -0.56**	**91.79 / -0.35**	**90.50 / -0.45**	**90.93 / -0.43**	**93.95 / -0.21**	**92.97 / -0.20**	**93.20 / -0.19**

Metrics. We report final average performance $\frac{1}{T}\sum_{i=1}^{T} S_{T,i}$, denoted ACC for vision and Avg. for language, together with backward transfer $\mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}(S_{T,i} - S_{i,i})$, where $S_{t,i}$ denotes performance on task i after processing task t. Detailed datasets, baselines, protocols, metrics, and implementation are provided in **Appendices C.1–C.4**.

Main Results on Vision Tasks. Table 1 shows that DebtMerge achieves the highest average accuracy among the compared merging methods across all three CLIP backbones and task horizons. On the 20-task streams, it improves over MADE-IT (Qiu et al., 2026a) by 7.52, 7.89, and 3.56 percentage points on ViT-B/32, ViT-B/16, and ViT-L/14, respectively, while maintaining BWT values of -0.56, -0.43, and -0.19. Independently fine-tuned checkpoints remain slightly stronger in average accuracy, indicating that a residual gap to task-specific models persists. Detailed task-level results and remaining failure cases are provided in **Appendix D.1**.

Table 2: Main language results.

Method	RoBERTa / 7 Tasks		Qwen / 10 Tasks	
	Avg.	BWT	Avg.	BWT
Pre-Trained	36.7	-	39.5	-
Fine-Tuned	83.9	-	74.3	-
C. Fine-Tuned	67.5	-	57.8	-
Average (SWA)	57.1	-12.09	38.9	-3.86
C. Task Arithmetic	60.8	-5.39	42.1	-4.50
C. TIES-Merging	56.7	-3.74	40.0	-5.02
C. LW AdaMerging	48.6	-13.59	45.2	-13.87
C. LoRA-WEMoE	49.7	-11.83	53.8	-9.42
OPCM	61.9	-3.99	42.6	-6.25
MINGLE	68.4	-2.13	67.2	-0.81
MADE-IT	76.2	-1.80	71.1	-0.65
DebtMerge	**77.3**	**-1.62**	**72.0**	**-0.51**

Main Results on Language Tasks. Table 2 shows the same overall trend on language models: DebtMerge obtains the highest average score and least negative BWT among the compared merging methods for both RoBERTa-base and Qwen2.5-1.5B. Relative to MADE-IT, the average score improves from 76.2 to 77.3 on RoBERTa-base and from 71.1 to 72.0 on Qwen2.5-1.5B, with BWT improving from -1.80 to -1.62 and from -0.65 to -0.51, respectively. Task-level gains remain heterogeneous, and detailed comparisons are provided in **Appendix D.2**.

Ablation Studies. We next examine whether the gains arise from the proposed long-horizon control and finite-budget search mechanisms. Fig. 2 isolates the controller by disabling historical recipe transfer, showing the benefit of retaining accumulated debt information across task arrivals. Table 3 further shows that the three debts regulate complementary aspects of expert evolution: stability debt primarily affects historical retention, plasticity debt affects incoming-knowledge retention, and resource debt regulates expert occupancy. Debt-guided geometry initialization improves the stability–plasticity trade-off, while adaptive credit and context-conditioned block transfer progressively reduce the finite-budget solver gap and harmful transfer. **Appendix D.3** further compares accumulated debt with stateless, fixed-weight, and leaky control and separates debt-dependent objective reweighting from geometry initialization, while **Appendix D.4** analyzes finite-budget historical transfer in greater detail. Fig. 3 illustrates the resulting online behavior as the debt states, expert-management actions, and expert occupancy evolve along the task stream.

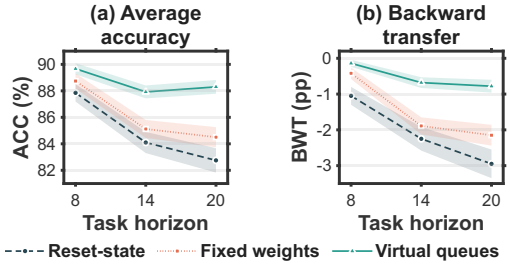


Fig. 2: Long-horizon control on ViT-B/32 with historical recipe transfer disabled for all three controllers.

Additional Analyses. Under matched evaluation budgets, DE performs best among the compared evolutionary operators, while historical transfer is most effective under tight search budgets. Context-conditioned block transfer improves search quality over local search and adapted full-recipe reuse, while reducing negative transfer relative to full-recipe reuse. Detailed results are provided in **Appendices D.4–D.5**. DebtMerge also remains effective under tighter expert capacities, with storage analysis clarifying that resource debt regulates expert occupancy rather than total memory. Further analyses of capacity and storage, prediction preservation, and sensitivity are provided in **Appendices D.6–D.8**.

Table 3: Component ablations on the 20-task ViT-B/32 stream. Current Ret. measures current-task retention; Hist. Deg. measures positive merge-induced historical degradation.

A. Debt Components

DebtMerge variant	ACC↑	BWT↑	Current Ret. (%)↑	Occupancy↓
w/o Stability Debt	87.42±0.61	−1.52±0.27	**94.25±0.45**	0.75±0.06
w/o Plasticity Debt	87.81±0.55	**−0.35±0.17**	91.84±0.52	**0.72±0.05**
w/o Resource Debt	**89.02±0.48**	−0.50±0.18	94.02±0.40	0.93±0.03
DebtMerge (full)	88.93±0.46	−0.56±0.18	94.10±0.40	0.74±0.07

B. Integration Geometry

DebtMerge variant	ACC↑	BWT↑	Hist. Deg.↓	Current Ret. (%)↑
w/o Geometry	86.95±0.72	−1.41±0.31	1.52±0.18	93.90±0.50
Global Scaling	87.34±0.64	−1.08±0.27	1.17±0.14	92.80±0.46
Fixed Geometry	88.01±0.56	−0.82±0.23	0.93±0.12	93.60±0.43
DebtMerge (debt-guided)	**88.93±0.46**	**−0.56±0.18**	**0.64±0.08**	**94.10±0.40**

C. Historical Recipe Transfer ($K = 150$)

DebtMerge variant	ACC↑	Norm. Gap↓	Pair NTR↓	Solver NTR↓
w/o Transfer (Local DE)	88.30±0.52	0.42±0.05	–	–
Fixed Transfer	88.45±0.49	0.31±0.04	24.6±4.0%	9.8±3.0%
+ Adaptive Credit	88.72±0.47	0.23±0.04	15.3±3.2%	4.6±2.2%
DebtMerge (+ Context Conditioning)	**88.93±0.46**	**0.17±0.03**	**9.5±2.7%**	**2.1±1.8%**

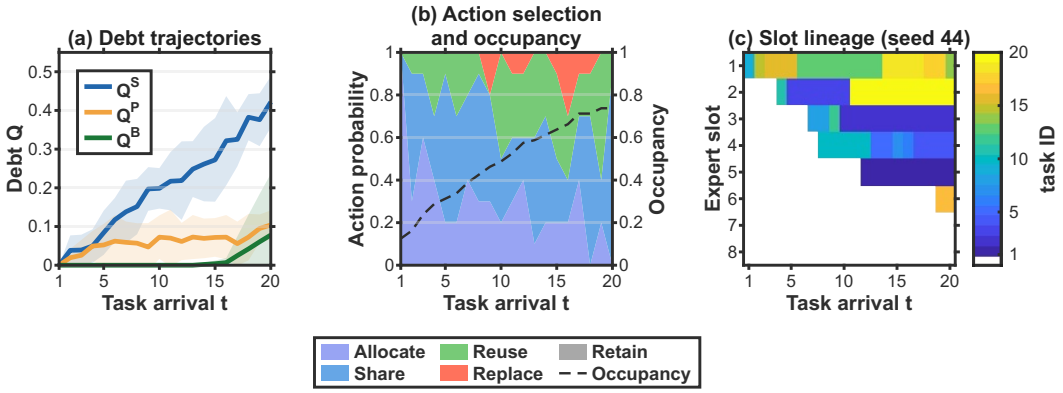


Fig. 3: Debt-driven expert evolution on the 20-task ViT-B/32 benchmark. (a) Stability, plasticity, and resource debt trajectories across task arrivals; (b) Expert-management action probabilities together with expert occupancy. (c) Representative expert-slot evolution under one task order.

6 Conclusion

This paper studies expert-managed continual model merging from a long-horizon perspective, where local merge decisions can accumulate historical perturbation, incoming-knowledge mismatch, and expert-capacity pressure under limited evaluation budgets. We proposed **DebtMerge**, which uses three merge debts to construct a debt-conditioned per-arrival objective, while stability and plasticity debts guide the initialization of integration geometry. Context-conditioned block transfer improves finite-budget optimization over the evolving mixed discrete–continuous merge space. Our analysis characterizes debt-guided directional control, incorporates finite-budget solver error into the long-horizon bound, and derives sufficient conditions for historical prediction preservation. Experiments across vision and language models show improved average performance over the compared merging baselines and support the proposed control, geometry, and historical-transfer mechanisms across varying task horizons, expert capacities, and search budgets.

Reproducibility Statement All algorithmic details and experimental settings are provided in the paper and appendix. The source code will be released publicly after the review process to ensure full reproducibility.

REFERENCES

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7:195–204, 2025. doi: 10.1038/s42256-024-00975-8.

Suzhi Bi, Liang Huang, Hui Wang, and Ying-Jun Angela Zhang. Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks. *IEEE Transactions on Wireless Communications*, 20(11):7519–7537, 2021. doi: 10.1109/TWC.2021.3085319.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, volume 8694 of *Lecture Notes in Computer Science*, pp. 446–461. Springer, 2014.

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

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

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks*, pp. 2921–2926. IEEE, 2017.

Liang Feng, Yew-Soon Ong, Siwei Jiang, and Abhishek Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 21(5):760–772, 2017. doi: 10.1109/TEVC.2017.2682274.

Liang Feng, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Yew-Soon Ong, Kay Chen Tan, and A. K. Qin. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 49 (9):3457–3470, 2019. doi: 10.1109/TCYB.2018.2845361.

Ian J. Goodfellow, Dumitru Erhan, Pierre-Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. *arXiv preprint arXiv:1307.0414*, 2013.

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

Pei-Qiu Huang, Yong Wang, Kezhi Wang, and Qingfu Zhang. Combining lyapunov optimization with evolutionary transfer optimization for long-term energy minimization in irs-aided communications. *IEEE Transactions on Cybernetics*, 53(4):2647–2657, 2023. doi: 10.1109/TCYB.2022.3168839.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, volume 80 of *Proceedings of Machine Learning Research*, pp. 876–885. PMLR, 2018.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561. IEEE, 2013.

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

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

Sanwoo Lee, Jiahao Liu, Qifan Wang, Jingang Wang, Xunliang Cai, and Yunfang Wu. Dynamic fisher-weighted model merging via bayesian optimization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4923–4935. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.254.

Lihong Lin and Haidong Kang. Unlocking the potential of continual model merging: An ode perspective. *arXiv preprint arXiv:2605.19409*, 2026.

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

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1287.

Michael J. Neely. Energy optimal control for time-varying wireless networks. *IEEE Transactions on Information Theory*, 52(7):2915–2934, 2006. doi: 10.1109/TIT.2006.876219.

Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2010. ISBN 978-1-60845-455-6. doi: 10.2200/S00271ED1V01Y201006CNT007.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin 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.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505. IEEE, 2012.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single lora continual learning via continual merging. In *International Conference on Learning Representations*, 2026.

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

Zihuan Qiu, Yi Xu, Chiyuan He, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. MINGLE: Mixture of null-space gated low-rank experts for test-time continual model merging. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4814.

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

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

Nazreen Shah, Govinda Arya, B. N. Bharath, and Ranjitha Prasad. Theoretical foundations of continual learning via drift-plus-penalty. *Transactions on Machine Learning Research*, 2026.

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*, 48(3):2329–2341, 2026. doi: 10.1109/TPAMI.2025.3629605.

Donald Shenaj, Ondrej Bohdal, Taha Ceritli, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. K-Merge: Online continual merging of adapters for on-device large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3013–3029, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.137.

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

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421.

Kay Chen Tan, Liang Feng, and Min Jiang. Evolutionary transfer optimization—a new frontier in evolutionary computation research. *IEEE Computational Intelligence Magazine*, 16(1):22–33, 2021. doi: 10.1109/MCI.2020.3039066.

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Fusionbench: A comprehensive benchmark of deep model fusion. *arXiv preprint arXiv:2406.03280*, 2024.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2747.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco S. Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. In *Medical Image Computing and Computer-Assisted Intervention*, volume 11071 of *Lecture Notes in Computer Science*, pp. 210–218. Springer, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Peng Wang, Biyu Zhou, Xuehai Tang, Jizhong Han, and Songlin Hu. Lyaplock: Bounded knowledge preservation in sequential large language model editing. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6434–6459. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.327.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022.

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

Jian Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492. IEEE, 2010.

Xiaoming Xue, Yao Hu, Liang Feng, Kai Zhang, Linqi Song, and Kay Chen Tan. Surrogate-assisted search with competitive knowledge transfer for expensive optimization. *IEEE Transactions on Evolutionary Computation*, 29(6):2416–2430, 2025. doi: 10.1109/TEVC.2024.3478732.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1311.

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*, pp. 57755–57775. PMLR, 2024.

Yu Zhou, Xingyu Wu, Jibin Wu, and Liang Feng. Hm3: Hierarchical multi-objective model merging for pretrained models. *Advances in Neural Information Processing Systems*, 38:114876–114910, 2026.

APPENDIX

CONTENTS

A Workflow of DebtMerge 14
A.1 Online Procedure 14
A.2 Merge-Recipe Construction 15
A.3 Debt-Guided Geometry Initialization 17
A.4 Context Construction and Target-Side Transfer Updates 17
B Additional Theoretical Analysis 17
B.1 Debt Stability and Long-Run Constraint Satisfaction 17
B.2 Debt-Guided Directional Control 18
B.3 Finite-Budget Optimization and Long-Horizon Control 19
B.4 General Finite-Horizon Comparator Bound 20
B.5 Search Coverage and Expected Solver Error 20
B.6 Historical Prediction Preservation 21
C Detailed Experimental Settings 22
C.1 Model and Dataset Details 22
C.2 Task Grouping and Sequence Protocols 22
C.3 Baseline Details 23
C.4 Evaluation Metrics and Search Settings 23
D Detailed Results 25
D.1 Detailed Per-Task Performance of Vision Tasks 25
D.2 Detailed Per-Task Performance of Language Tasks 25
D.3 Analysis of Debt-Based Control 28
D.4 Finite-Budget Historical Transfer 29
D.5 Historical Transfer Provenance 30
D.6 Expert Capacity and Storage Accounting 31
D.7 Prediction-Preservation Analysis 32
D.8 Hyperparameter Sensitivity 33
E Statement about usage of LLMs 33

Remark: The source code will be made publicly available after the review process is complete.

A WORKFLOW OF DEBTMERGE

This section first presents the online procedure, then details recipe construction in **Appendix A.2**, geometry initialization in **Appendix A.3**, and target-side transfer updates in **Appendix A.4**.

A.1 ONLINE PROCEDURE

Algorithm 1 summarizes the complete online merging procedure, comprising debt-conditioned initialization, finite-budget recipe search, and post-merge state updates. The current expert system and merge debts define a fixed optimization problem at each arrival; historical transfer changes the candidate proposals without changing this objective. Only the best feasible evaluated recipe is committed to the maintained expert system.

Debt-Conditioned Initialization. We denote the maintained historical spectral sketches by $\mathcal{H}_t = \{(U_t^{(l)}, \Lambda_t^{(l)})\}_{l\in\mathcal{L}}$. The procedure initializes the expert bank, routing state, sketches, merge debts, and historical recipe archive (lines 1–2). At each arrival, it computes the incoming task update, fixes the pre-merge state, constructs the normalized context, and defines Ψ_t (lines 4–7). Structural anchors and the debt-guided pass rate $g_{t,k}^{\mathrm{geo}} = (V + Q_t^P)/(V + Q_t^P + Q_t^S h_{t,k})$ initialize the population, whose evaluation consumes the first N_p units of the budget (lines 8–9). **Appendix A.3** specifies the anchor construction. The protected rank, protection strength, and absorption coefficient remain searchable rather than being fixed to these anchors.

Local Search and Historical Transfer. Mixed-variable crossover, mutation, and repair produce a feasible local proposal (lines 11–12). When the archive is nonempty and budget remains, transfer

is attempted with probability π_t. A historical source is sampled according to context similarity and transfer credit, and a compatible block from $G_1 = (o, q_{\mathrm{tar}})$, $G_2 = (q_{\mathrm{src}}, \mu_d, \mu_r)$, or $G_3 = (r, \lambda, \alpha)$ is inserted into the local proposal to form a transferred challenger (lines 14–18). We use $\mathrm{Evaluate}_t(a)$ for complete-recipe evaluation: it decodes relative expert selections using the current cosine-similarity rankings, constructs the donor and composed updates, and applies $\delta_t^{(l)} = \alpha_t^{(l)}[I - \lambda_t^{(l)} U_{t,S}^{(l)} U_{t,S}^{(l)\top}]\widetilde{d}_t^{(l)}$ within a temporary expert system, as detailed in **Appendix A.2**. The resulting $F_t(a)$, $P_t(a)$, $C_t(a)$, and $\Psi_t(a)$ are recorded without modifying $(X_t, Q_t, \mathcal{H}_t)$. All candidates use the same routing and materialization rule.

Adaptive Feedback and Budget Accounting. The objective difference between a local proposal and its transferred challenger determines the transfer reward, which updates the source credit and overall transfer probability using Eqs. 16 and 17 (lines 20–23). Both candidates enter the same elitist selection, and Best_{N_p} retains the N_p candidates with the smallest recorded objective values (line 26). Initial, local, and transferred evaluations all count toward K; a local–transfer pair therefore consumes two evaluations. Repair enforces the current capacity limits and variable domains, and an incompatible transfer is skipped. With an empty archive, the procedure reduces to local search.

Commitment and State Updates. Once the budget is exhausted, only the best evaluated recipe is committed (lines 28–29). Its realized proxies update the three merge debts (line 30), while the historical sketches are updated from the normalized raw incoming update rather than the debt-filtered update (line 31). Finally, the committed recipe and its pre-merge context are added to the archive, and the final transfer probability is carried to the next arrival (line 32).

Operational Conventions. For definiteness, the workflow retains sequential proposal-pair selection, zero initial credit for a newly archived recipe, and the running mean of transfer rewards within each arrival. When no source expert exists, source-dependent composition is bypassed; with no historical sketch, the geometry anchor is $(r, \lambda, \alpha) = (0, 0, 1)$. At the first task arrival, the stability proxy is $F_1 = 0$. Continuous variables use a DE-style operator, while categorical and integer variables use type-aware variation and repair. The returned recipe is the best evaluated feasible candidate, not a claimed global minimizer; the algorithm does not require the theoretical solver error ξ_t.

A.2 Merge-Recipe Construction

For each managed layer l, let $\mathcal{E}_{t-1}^{(l)} = \{e_{t-1,j}^{(l)}\}_{j=1}^{m_l}$ denote the current expert bank and let $d_t^{(l)}$ denote the effective incoming task update at that layer. When parameter-efficient experts such as LoRA are used, $d_t^{(l)}$ denotes the effective induced parameter update, e.g., $\Delta W_t^{(l)} = B_t^{(l)} A_t^{(l)}$, rather than a particular low-rank factorization. Accordingly, the protected-subspace rank $r_t^{(l)}$ in Eq. 5 is conceptually distinct from the rank used to parameterize a LoRA adapter. Because expert identities and bank sizes can change across arrivals, DebtMerge does not encode persistent expert indices. Instead, following affinity-based expert management in continual merging (Qiu et al., 2026a), the current experts are ranked by their cosine similarity $s_{t,j}^{(l)} = \langle d_t^{(l)}, e_{t-1,j}^{(l)} \rangle / (\|d_t^{(l)}\|_2 \|e_{t-1,j}^{(l)}\|_2 + \varepsilon)$, and the normalized variables $q_{t,\mathrm{tar}}^{(l)}, q_{t,\mathrm{src}}^{(l)} \in [0, 1]$ are mapped to relative positions in this ranking. This representation converts task-specific expert identities into relative selection policies that remain meaningful as the expert bank evolves. After selecting a source expert $e_{t,\mathrm{src}}^{(l)}$, the incoming update is decomposed into a source-aligned component and an orthogonal residual, from which DebtMerge constructs a controllable donor and composed update:

$$
\begin{aligned}
d_{t,\perp}^{(l)} &= d_t^{(l)} - \frac{\left\langle d_t^{(l)}, e_{t,\mathrm{src}}^{(l)} \right\rangle}{\left\| e_{t,\mathrm{src}}^{(l)} \right\|_2^2 + \varepsilon} e_{t,\mathrm{src}}^{(l)}, \\
d_{t,\mathrm{donor}}^{(l)} &= \left(1 - \mu_{t,r}^{(l)}\right) e_{t,\mathrm{src}}^{(l)} + \mu_{t,r}^{(l)} d_{t,\perp}^{(l)}, \\
\widetilde{d}_t^{(l)} &= \left(1 - \mu_{t,d}^{(l)}\right) d_t^{(l)} + \mu_{t,d}^{(l)} d_{t,\mathrm{donor}}^{(l)}.
\end{aligned}
\tag{12}
$$

Algorithm 1 DebtMerge: Online Debt-Based Evolutionary Recipe Transfer

Input: Pretrained model θ_0; task models $\{\theta_t^{\text{task}}\}_{t=1}^{T}$ arriving sequentially; managed layers $\mathcal{L}$ and capacities $\{M_l\}$.
Input: Controller parameters $(V, \omega, \epsilon_S, \epsilon_P, C_0)$; geometry threshold τ and rank limits $\{r_{\max}^{(l)}\}$; population size N_p and evaluation budget $K \geq N_p$.
Input: Transfer parameters $(\pi_0, \pi_{\min}, \pi_{\max}, \kappa_\gamma, \kappa_c, \eta_\gamma, \eta_\pi)$ and numerical constant $\epsilon > 0$.
Output: Final expert bank $\mathcal{E}_T$ and routing mechanism $\mathcal{R}_T$, used with θ_0 for inference.

Initialize $\mathcal{E}_0 \leftarrow \varnothing$, the base routing state $\mathcal{R}_0$, and empty sketches $\mathcal{H}_1$.
$Q_1 \leftarrow \mathbf{0}$, $\mathcal{B}_1 \leftarrow \varnothing$, $\pi_1 \leftarrow \pi_0$.
for $t = 1, \ldots, T$ **do**
    Debt-conditioned initialization
    Receive θ_t^{task} and compute $d_t \leftarrow \theta_t^{\text{task}} - \theta_0$.
    Set $X_t \leftarrow (\mathcal{E}_{t-1}, \mathcal{R}_{t-1})$; freeze $(X_t, Q_t, \mathcal{H}_t)$ throughout candidate search.
    Rank current experts per layer and construct the normalized context $c_t \leftarrow [\bar{Q}_t^S, \bar{Q}_t^P, \bar{Q}_t^B, \zeta_t]$.
    Define $\Psi_t(a) = Q_t^S F_t(a) + (Q_t^P + V)P_t(a) + (Q_t^B + V\omega)C_t(a)$.
    Initialize N_p feasible complete recipes $\mathcal{P}$ using structural and debt-guided geometry anchors.
    Evaluate all $a \in \mathcal{P}$; $\mathcal{C}_t \leftarrow \mathcal{P}$, $n \leftarrow N_p$, $\mathcal{V}_t \leftarrow \varnothing$.
    Finite-budget local and transferred proposals
    while $n < K$ **do**
        Generate and repair a local proposal a^{loc} from $\mathcal{P}$ using mixed-variable crossover and mutation.
        $\text{Evaluate}_t(a^{\text{loc}})$; $n \leftarrow n + 1$; $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{a^{\text{loc}}\}$.
        $\mathcal{O} \leftarrow \{a^{\text{loc}}\}$.
        if $\mathcal{B}_t \neq \varnothing$ and $n < K$ and $\text{Uniform}(0,1) < \pi_t$ **then**
            Sample source $i < t$ with $p_{i\to t} \propto \exp(\kappa_\gamma \gamma_i - \kappa_c \|c_i - c_t\|_2^2)$.
            Construct a^{tr} by compatible block insertion from a_i into a^{loc}, followed by repair.
            if a compatible feasible challenger is obtained **then**
                $\text{Evaluate}_t(a^{\text{tr}})$; $n \leftarrow n + 1$; $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{a^{\text{tr}}\}$.
                $\mathcal{O} \leftarrow \mathcal{O} \cup \{a^{\text{tr}}\}$.
                $R_{i\to t} \leftarrow \text{clip}\left(\dfrac{\Psi_t(a^{\text{loc}}) - \Psi_t(a^{\text{tr}})}{|\Psi_t(a^{\text{loc}})| + \epsilon}, -1, 1\right)$.
                $\gamma_i \leftarrow (1 - \eta_\gamma)\gamma_i + \eta_\gamma R_{i\to t}$.
                Append $R_{i\to t}$ to $\mathcal{V}_t$; $\bar{R}_t \leftarrow \text{mean}(\mathcal{V}_t)$.
                $\pi_t \leftarrow \text{clip}(\pi_t + \eta_\pi \bar{R}_t, \pi_{\min}, \pi_{\max})$.
            end if
        end if
        $\mathcal{P} \leftarrow \text{Best}_{N_p}(\mathcal{P} \cup \mathcal{O}; \Psi_t)$.
    end while
    Single commitment and state updates
    $\hat{a}_t \leftarrow \arg\min_{a \in \mathcal{C}_t} \Psi_t(a)$.
    Commit only $\hat{a}_t$: $X_{t+1} \leftarrow \mathcal{M}(X_t, d_t, \hat{a}_t) = (\mathcal{E}_t, \mathcal{R}_t)$.
    $Q_{t+1} \leftarrow \left[Q_t + \left(F_t(\hat{a}_t) - \epsilon_S,\ P_t(\hat{a}_t) - \epsilon_P,\ C_t(\hat{a}_t) - C_0\right)^\top\right]_+$.
    Update $\mathcal{H}_{t+1}$ incrementally from $\mathcal{H}_t$ using the normalized raw updates $\bar{d}_t^{(l)} = d_t^{(l)} / (\|d_t^{(l)}\|_2 + \epsilon)$.
    $\mathcal{B}_{t+1} \leftarrow \mathcal{B}_t \cup \{(c_t, \hat{a}_t, 0)\}$; $\pi_{t+1} \leftarrow \pi_t$.
end for
return $(\mathcal{E}_T, \mathcal{R}_T)$.

Here, $\mu_{t,r}^{(l)}$ determines whether the donor emphasizes source-aligned or complementary residual information, whereas $\mu_{t,d}^{(l)}$ controls how strongly historical expert knowledge is incorporated into the incoming task update.

The categorical variable $o_t^{(l)}$ determines how the current layer interacts with the expert bank. Its admissible operations include *retain*, *reuse*, *allocate*, *replace*, and *share*, depending on the current capacity and expert availability. *Retain* leaves the current expert configuration unchanged; *reuse* assigns the incoming task to an existing expert without modifying its parameters; *allocate* creates a new expert when capacity permits; *replace* uses an occupied slot for the incoming expert when capacity is saturated; and *share* injects the composed update into a selected existing expert. These variables jointly determine the structural and compositional components of a candidate merge recipe.

A.3 Debt-Guided Geometry Initialization

Given the composed update $\widetilde{d}_t^{(l)}$ from Eq. 12, the directional gate in Eq. 6 provides an analytical initialization for the geometric variables. We use the historical sensitivities $h_{t,k}^{(l)}$, aligned energy fractions $e_{t,k}^{(l)}$, and debt weights $A_t = V + Q_t^P$ and $B_t = Q_t^S$ defined in Sec. 4.2. Defining the protection score $\rho_{t,k}^{(l)} = 1 - g_{t,k}^{\mathrm{geo}}$ and the protected set $\mathcal{S}_t^{(l)} = \{k : \rho_{t,k}^{(l)} \geq \tau\}$ gives the anchor rank $r_{t,\mathrm{geo}}^{(l)} = |\mathcal{S}_t^{(l)}|$, while the corresponding protection and absorption anchors are

$$\begin{aligned} \lambda_{t,\mathrm{geo}}^{(l)} &= \frac{\sum_{k \in \mathcal{S}_t^{(l)}} e_{t,k}^{(l)} \rho_{t,k}^{(l)}}{\sum_{k \in \mathcal{S}_t^{(l)}} e_{t,k}^{(l)} + \varepsilon}, \\ \alpha_{t,\mathrm{geo}}^{(l)} &= \frac{A_t}{A_t + B_t h_{t,\perp}^{(l)}}, \end{aligned} \tag{13}$$

where $h_{t,\perp}^{(l)}$ denotes the residual historical sensitivity immediately outside the protected rank. These anchors seed the population; the optimizer can vary all three geometric variables when evaluating the controlled update in Eq. 7.

A.4 Context Construction and Target-Side Transfer Updates

Optimization Context. Historical transfer is conditioned on both the accumulated merge debts and the current expert geometry. At arrival t, we construct

$$c_t = [\bar{Q}_t^S, \bar{Q}_t^P, \bar{Q}_t^B, \zeta_t], \tag{14}$$

where $\bar{Q}_t^S$, $\bar{Q}_t^P$, and $\bar{Q}_t^B$ are normalized debt states, and ζ_t collects compact statistics characterizing the current expert bank and parameter geometry. Specifically, ζ_t summarizes expert similarity, spectral concentration of the historical sketch, overlap between the incoming update and the historical subspace, and residual update energy. All context dimensions are normalized before evaluating $\|c_i - c_t\|_2^2$ in Eq. 8. This relative representation avoids relying on persistent expert identities and allows contexts to remain comparable as the expert bank evolves.

Target-Side Transfer Feedback. For a local proposal a_t^{loc} and its transferred challenger a_t^{tr}, both evaluated on the same fixed pre-merge state (X_t, Q_t), the clipped relative improvement in Ψ_t defines the target-side transfer reward

$$R_{i \to t} = \mathrm{clip}\left(\frac{\Psi_t(a_t^{\mathrm{loc}}) - \Psi_t(a_t^{\mathrm{tr}})}{|\Psi_t(a_t^{\mathrm{loc}})| + \varepsilon}, -1, 1 \right). \tag{15}$$

The transfer credit of source i is updated by

$$\gamma_i \leftarrow (1 - \eta_\gamma)\gamma_i + \eta_\gamma R_{i \to t}, \tag{16}$$

and the transfer probability is adapted as

$$\pi_t \leftarrow \mathrm{clip}\left(\pi_t + \eta_\pi \bar{R}_t, \pi_{\min}, \pi_{\max}\right), \tag{17}$$

where $\bar{R}_t$ is the running mean of transfer rewards observed at the current arrival. Positive transfer therefore increases the future reuse probability of historical experience, whereas negative transfer reduces it.

All local and transferred candidates are evaluated under the same debt-conditioned objective Ψ_t and compete through the same elitist selection. Initial, local, and transferred evaluations all count toward the fixed budget K. If the archive is empty, historical transfer is disabled and the optimizer reduces to local evolutionary search.

B Additional Theoretical Analysis

B.1 Debt Stability and Long-Run Constraint Satisfaction

Proof of Proposition 1. From the component-wise debt recursion in Eq. 3, the projection onto the non-negative orthant gives

$$Q_{t+1,j} \geq Q_{t,j} + u_{t,j}(a_t), \tag{18}$$

for each debt component $j \in \{S, P, B\}$. Summing from $t = 1$ to T yields

$$\sum_{t=1}^{T} u_{t,j}(a_t) \le Q_{T+1,j} - Q_{1,j}. \tag{19}$$

Dividing by T and taking the limit superior gives

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} u_{t,j}(a_t) \le \limsup_{T\to\infty} \frac{Q_{T+1,j}}{T}, \tag{20}$$

where $Q_{1,j}/T \to 0$. Mean-rate stability implies $Q_{T+1,j}/T \to 0$, and therefore the time-average violation of every component is non-positive. Substituting the definitions of $u_{t,S}$, $u_{t,P}$, and $u_{t,B}$ gives

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} F_t(a_t) \le \epsilon_S, \tag{21}$$

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} P_t(a_t) \le \epsilon_P, \tag{22}$$

and

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} C_t(a_t) \le C_0. \tag{23}$$

Hence, mean-rate stability of the merge-debt process is sufficient for satisfaction of all time-average constraints in Eq. 2. □

B.2 Debt-Guided Directional Control

Proof of Proposition 2. For clarity, we omit the arrival and layer indices and write the local geometry surrogate as $\Phi(g) = Ae(1-g)^2 + Bheg^2$, where $A = V + Q_t^P > 0$, $B = Q_t^S \ge 0$, $h \ge 0$, and $e > 0$. Its first and second derivatives are

$$\Phi'(g) = 2e\left[(A + Bh)g - A\right], \qquad \Phi''(g) = 2e(A + Bh) > 0. \tag{24}$$

Hence, $\Phi(g)$ is strictly convex and has a unique stationary point,

$$g^\star = \frac{A}{A + Bh} = \frac{V + Q_t^P}{V + Q_t^P + Q_t^S h}, \tag{25}$$

which lies in $[0, 1]$ and is therefore the unique minimizer. Differentiating with respect to the two debt variables gives

$$\frac{\partial g^\star}{\partial Q_t^S} = -\frac{(V + Q_t^P)h}{\left(V + Q_t^P + Q_t^S h\right)^2} \le 0, \qquad \frac{\partial g^\star}{\partial Q_t^P} = \frac{Q_t^S h}{\left(V + Q_t^P + Q_t^S h\right)^2} \ge 0. \tag{26}$$

The first inequality is strict whenever $h > 0$, while the second is strict whenever $Q_t^S h > 0$. Therefore, increasing stability debt decreases the directional pass rate along historically sensitive directions, whereas increasing plasticity debt weakly increases it. □

The same monotonicity transfers to the protected-subspace construction. Since $\rho_{t,k} = 1 - g_{t,k}^{\text{geo}}$, increasing Q_t^S can only increase the protection score and therefore cannot remove an already protected direction under a fixed threshold τ. Conversely, increasing Q_t^P can only decrease the protection score and therefore cannot introduce a newly protected direction. Resource debt Q_t^B does not enter this fixed-structure result because it acts through the expert-occupancy term $C_t(a)$ rather than directional parameter flow.

Debt-dependent selection on a common candidate pool. The initialization analysis can be complemented by a direct property of the selected proxies when candidate generation is held fixed.

Proposition 3 (Common-pool proxy monotonicity)**.** *Fix a nonempty finite candidate set $\mathcal{C}$ and its proxy values. For $q_2 > q_1$, let $a_k \in \arg\min_{a\in\mathcal{C}}\{q_k f(a) + h(a)\}$, where h is independent of q. Then $f(a_2) \le f(a_1)$.*

Proof. Optimality gives $q_1 f(a_1)+h(a_1) \le q_1 f(a_2)+h(a_2)$ and $q_2 f(a_2)+h(a_2) \le q_2 f(a_1)+h(a_1)$. Adding yields $(q_2 - q_1)[f(a_2) - f(a_1)] \le 0$, proving the claim. □

Applied to Ψ_t, with the candidate pool, proxy values, and all other debt components fixed, increasing only Q_t^S, Q_t^P, or Q_t^B cannot increase the selected value of F_t, P_t, or C_t, respectively, under exact minimization over the same pool. This property isolates the effect of objective reweighting and does not imply that the other proxies improve simultaneously. Debt-dependent initialization or rerunning the solver changes the candidate pool, so monotonicity of the complete search trajectory is a separate question.

B.3 Finite-Budget Optimization and Long-Horizon Control

Proof of Theorem 1. Using the quadratic Lyapunov function $L(Q_t) = \frac{1}{2}\|Q_t\|_2^2$ and the debt recursion in Eq. 3, the bounded-violation assumption gives

$$L(Q_{t+1}) - L(Q_t) \le B + Q_t^\top u_t(\hat{a}_t). \tag{27}$$

Adding $V\ell_t(\hat{a}_t)$ to both sides yields

$$L(Q_{t+1}) - L(Q_t) + V\ell_t(\hat{a}_t) \le B + Q_t^\top u_t(\hat{a}_t) + V\ell_t(\hat{a}_t). \tag{28}$$

From the solver-error definition in Eq. 10 and the feasibility of $a_t^\dagger$,

$$Q_t^\top u_t(\hat{a}_t) + V\ell_t(\hat{a}_t) \le Q_t^\top u_t(a_t^\dagger) + V\ell_t(a_t^\dagger) + \xi_t. \tag{29}$$

Because $Q_t \succeq 0$ and $u_t(a_t^\dagger) \preceq 0$, we have $Q_t^\top u_t(a_t^\dagger) \le 0$. Combining Eqs. 28 and 29 therefore gives

$$L(Q_{t+1}) - L(Q_t) + V\left[\ell_t(\hat{a}_t) - \ell_t(a_t^\dagger)\right] \le B + \xi_t. \tag{30}$$

Summing Eq. 30 over $t = 1, \dots, T$ yields

$$L(Q_{T+1}) - L(Q_1) + V\sum_{t=1}^{T}\left[\ell_t(\hat{a}_t) - \ell_t(a_t^\dagger)\right] \le BT + \sum_{t=1}^{T}\xi_t. \tag{31}$$

Using $L(Q_{T+1}) \ge 0$ and dividing by VT gives

$$\frac{1}{T}\sum_{t=1}^{T}\left[\ell_t(\hat{a}_t) - \ell_t(a_t^\dagger)\right] \le \frac{B}{V} + \frac{L(Q_1)}{VT} + \frac{1}{VT}\sum_{t=1}^{T}\xi_t, \tag{32}$$

which proves Eq. 11. □

Corollary 1 (Finite-horizon debt and time-average constraint bounds)**.** *Under the assumptions of Theorem 1, define $\ell_\dagger = \epsilon_P + \omega C_0$. Then*

$$\frac{\|Q_{T+1}\|_2}{T} \le \frac{\sqrt{\|Q_1\|_2^2 + 2T(B + V\ell_\dagger) + 2\sum_{t=1}^{T}\xi_t}}{T}. \tag{33}$$

For fixed V, $\sum_{t=1}^{T}\xi_t = o(T^2)$ implies rate stability. If the average solver error is uniformly bounded, the normalized debt is $O(T^{-1/2})$. With the algorithm's initialization $Q_1 = 0$, each component satisfies

$$\mathrm{CV}_j(T) := \left[\frac{1}{T}\sum_{t=1}^{T}u_{t,j}(\hat{a}_t)\right]_+ \le \frac{Q_{T+1,j}}{T} \le \frac{\|Q_{T+1}\|_2}{T}. \tag{34}$$

Proof. The soft-feasible reference satisfies $P_t(a_t^\dagger) \le \epsilon_P$ and $C_t(a_t^\dagger) \le C_0$, hence $\ell_t(a_t^\dagger) \le \ell_\dagger$. Since $\ell_t(\hat{a}_t) \ge 0$, Eq. 30 yields $L(Q_{t+1}) - L(Q_t) \le B + V\ell_\dagger + \xi_t$. Summing and substituting $L(Q) = \|Q\|_2^2/2$ proves Eq. 33. Dividing its square by T^2 gives rate stability when the accumulated error is $o(T^2)$. If $T^{-1}\sum_t \xi_t \le \bar{\xi}$, the bound is at most $\sqrt{\|Q_1\|_2^2/T^2 + 2(B + V\ell_\dagger + \bar{\xi})/T}$. Finally, summing the queue recursion gives $\sum_t u_{t,j}(\hat{a}_t) \le Q_{T+1,j} - Q_{1,j}$; with $Q_1 = 0$, taking the positive part proves Eq. 34. □

This result connects finite-budget error to finite-stream constraint violations under the theorem's same-state feasibility and uniform bounded-violation assumptions. A fixed evaluation budget alone does not establish these assumptions or the required error growth. The denominator constant ε prevents undefined proxies but does not itself ensure uniformly bounded violations.

B.4 General Finite-Horizon Comparator Bound

The per-arrival soft-feasibility condition $u_t(a_t^\dagger) \preceq 0$ in Theorem 1 provides a concise long-horizon interpretation but is not required for a finite-horizon comparison. Consider any sequence of feasible comparator actions $\{a_t^\dagger\}_{t=1}^T$ evaluated on the same realized pre-merge states as DebtMerge. Repeating the argument in **Appendix B.3** without dropping the term $Q_t^\top u_t(a_t^\dagger)$ gives

$$\frac{1}{T}\sum_{t=1}^{T}\Big[\ell_t(\hat{a}_t) - \ell_t(a_t^\dagger)\Big] \le \frac{B}{V} + \frac{L(Q_1)}{VT} + \frac{1}{VT}\sum_{t=1}^{T}\xi_t + \frac{1}{VT}\sum_{t=1}^{T} Q_t^\top u_t(a_t^\dagger). \tag{35}$$

The final term explicitly quantifies how the comparator interacts with the accumulated debt state. The bound in Theorem 1 is recovered when $u_t(a_t^\dagger) \preceq 0$ at every task arrival. As in the main theorem, the comparator is evaluated on the same realized sequence of pre-merge states and therefore does not represent a clairvoyant offline policy that induces a different expert-system trajectory.

B.5 Search Coverage and Expected Solver Error

The long-horizon guarantee depends on the finite-budget optimizer only through the solver error ξ_t. We next relate this error to the probability that the search visits a near-optimal region of the current action space. Let $\Psi_t^\star = \inf_{a\in\mathcal{A}(X_t,d_t)} \Psi_t(a)$ and define the ϵ-optimal set as $\mathcal{A}_{t,\epsilon} = \{a \in \mathcal{A}(X_t, d_t) : \Psi_t(a) - \Psi_t^\star \le \epsilon\}$.

Proposition 4 (Search coverage and expected solver error)**.** *Assume that $\Psi_t(a) - \Psi_t^\star \le \Delta_t^{\max}$ for all feasible actions. Let $p_t^{(K)}(\epsilon)$ denote the probability that at least one of the candidates evaluated within budget K belongs to $\mathcal{A}_{t,\epsilon}$. Since DebtMerge returns the best evaluated candidate,*

$$\mathbb{E}[\xi_t] \le p_t^{(K)}(\epsilon)\epsilon + \Big[1 - p_t^{(K)}(\epsilon)\Big]\Delta_t^{\max}. \tag{36}$$

Proof. Let $\mathcal{E}_{t,\epsilon}$ denote the event that at least one evaluated candidate lies in $\mathcal{A}_{t,\epsilon}$. On $\mathcal{E}_{t,\epsilon}$, the best evaluated candidate satisfies $\xi_t \le \epsilon$. On the complementary event, the assumed global gap bound gives $\xi_t \le \Delta_t^{\max}$. Taking expectation over the two events immediately yields Eq. 36. □

Proposition 4 does not depend on a particular evolutionary operator. It shows that, under the same complete-action evaluation budget, a search strategy that assigns greater probability to near-optimal regions produces a smaller upper bound on the expected solver error. Context-conditioned block transfer is designed to improve this finite-budget coverage by concentrating candidate generation around previously useful regions while retaining target-side evaluation to suppress harmful transfer.

Layer-wise structure of the proxy objective. A useful special case arises when each effective branch is a concatenation of layer-wise updates, each action block $a^{(l)}$ affects only layer l, and the feasible action set factorizes across layers. Under these assumptions, the fixed-state objective is separable:

$$\Psi_t(a) = \sum_{l\in\mathcal{L}} \psi_{t,l}(a^{(l)}).$$

Indeed, the denominators of F_t and P_t are fixed during candidate search, their squared-norm numerators decompose across layers, and C_t is additive over managed layers. Consequently,

$$\inf_{a\in\mathcal{A}(X_t,d_t)} \Psi_t(a) = \sum_{l\in\mathcal{L}} \inf_{a^{(l)}\in\mathcal{A}_t^{(l)}} \psi_{t,l}(a^{(l)})$$

whenever the feasible set admits the corresponding Cartesian-product factorization.

This separability is a special case rather than an assumption required by DebtMerge. Shared structural constraints, routing or materialization rules, and other cross-layer feasibility requirements can invalidate the factorization. When the separable conditions approximately hold, layer-wise or coordinate optimization provides a computationally efficient alternative; we explicitly evaluate this case in Appendix D.4. DebtMerge instead operates on the general state-dependent recipe space and does not rely on layer-wise separability.

B.6 Historical Prediction Preservation

The online stability proxy F_t measures parameter-branch perturbation, whereas forgetting is ultimately expressed through changes in model predictions. We therefore do not assume an unconditional equivalence between F_t and historical prediction loss. Instead, we characterize prediction drift by separating changes in expert functions from changes in routing.

Consider either hard routing or a convex output-level mixture over a fixed set of aligned expert slots $\mathcal{J}$. For a historical input x, let the pre- and post-merge logits be $z^{\pm}(x) = \sum_{e\in\mathcal{J}} w_e^{\pm}(x) h_e^{\pm}(x)$, where $w_e^{\pm}(x) \geq 0$ and $\sum_e w_e^{\pm}(x) = 1$. Inactive expert slots are assigned zero routing weight so that expert allocation or replacement can be represented in the same aligned slot space. Define the expert-function and routing-drift terms as

$$\begin{aligned} B_E(x) &= \sum_{e\in\mathcal{J}} w_e^{+}(x) \left\| h_e^{+}(x) - h_e^{-}(x) \right\|_\infty , \\ B_R(x) &= \frac{1}{2} D^{-}(x) \left\| w^{+}(x) - w^{-}(x) \right\|_1 , \\ D^{-}(x) &= \max_c \left[\max_{e\in\mathcal{J}} h_{e,c}^{-}(x) - \min_{e\in\mathcal{J}} h_{e,c}^{-}(x) \right] . \end{aligned} \tag{37}$$

Theorem 2 (Expert-and-routing prediction drift)**.** *For the expert mixture defined above,*

$$\left\| z^{+}(x) - z^{-}(x) \right\|_\infty \leq B_E(x) + B_R(x). \tag{38}$$

Proof. Adding and subtracting $\sum_e w_e^{+}(x) h_e^{-}(x)$ gives

$$z^{+}(x) - z^{-}(x) = \sum_e w_e^{+}(x) \left[h_e^{+}(x) - h_e^{-}(x) \right] + \sum_e \left[w_e^{+}(x) - w_e^{-}(x) \right] h_e^{-}(x). \tag{39}$$

The infinity norm of the first term is bounded by $B_E(x)$. For the second term, let $\Delta w_e = w_e^{+}(x) - w_e^{-}(x)$. Since both routing vectors sum to one, $\sum_e \Delta w_e = 0$. For every logit coordinate c, the range bound for a zero-sum weighted combination gives

$$\left| \sum_e \Delta w_e h_{e,c}^{-}(x) \right| \leq \frac{1}{2} \left[\max_e h_{e,c}^{-}(x) - \min_e h_{e,c}^{-}(x) \right] \|\Delta w\|_1. \tag{40}$$

Taking the maximum over c yields $B_R(x)$, and applying the triangle inequality to Eq. 39 proves Eq. 38. □

For a historical example (x, y) that is correctly classified before the current merge, let $m^{-}(x, y) = z_y^{-}(x) - \max_{c\neq y} z_c^{-}(x)$ denote its pre-merge classification margin.

Corollary 2 (Historical prediction preservation)**.** *If*

$$m^{-}(x, y) > 2 \left[B_E(x) + B_R(x) \right], \tag{41}$$

then the post-merge prediction for x remains y.

Proof. Let $\delta(x) = B_E(x) + B_R(x)$. From Theorem 2, every logit coordinate changes by at most $\delta(x)$. Therefore,

$$z_y^+(x) - \max_{c \neq y} z_c^+(x) \geq m^-(x, y) - 2\delta(x). \tag{42}$$

The condition in Eq. 41 makes the right-hand side strictly positive, so the predicted class remains y. □

The certificate also clarifies the role and limitation of the online stability proxy F_t. A small F_t is not claimed to provide an unconditional upper bound on either B_E or B_R; rather, F_t serves as a data-free control signal during merging, whereas $B_E + B_R$ directly characterizes prediction-level drift under the expert-mixture assumptions above. The experiments therefore evaluate the empirical association among F_t, expert-function drift, routing drift, and observed historical degradation.

C Detailed Experimental Settings

C.1 Model and Dataset Details

For image experiments, we use the CLIP (Radford et al., 2021) base configurations with patch sizes 32 and 16, together with the large configuration at patch size 14 (ViT-B/32, ViT-B/16, and ViT-L/14). The continual-merging protocol is aligned with OPCM (Tang et al., 2025), MINGLE (Qiu et al., 2025), and MADE-IT (Qiu et al., 2026a). The stream draws on 20 public image-classification datasets: SUN397 (Xiao et al., 2010), Stanford Cars (Krause et al., 2013), RESISC45 (Cheng et al., 2017), EuroSAT (Helber et al., 2019), SVHN (Netzer et al., 2011), GTSRB (Stallkamp et al., 2012), MNIST (LeCun et al., 1998), DTD (Cimpoi et al., 2014), Flowers102 (Nilsback & Zisserman, 2008), and PCAM (Veeling et al., 2018). The remaining tasks are FER2013 (Goodfellow et al., 2013), Oxford-IIIT Pet (Parkhi et al., 2012), STL-10 (Coates et al., 2011), CIFAR-100 and CIFAR-10 (Krizhevsky & Hinton, 2009), Food-101 (Bossard et al., 2014), Fashion-MNIST (Xiao et al., 2017), EMNIST (Cohen et al., 2017), KMNIST (Clanuwat et al., 2018), and Rendered SST-2, the image-rendered version of SST-2 (Socher et al., 2013) introduced for CLIP (Radford et al., 2021). Task-specific CLIP checkpoints come from the public FusionBench release (Tang et al., 2024). During adaptation, we update CLIP's vision branch and freeze its text branch. Each checkpoint is trained for $4{,}000$ steps with batches of 128, minimizing cross-entropy with Adam; the learning rate follows a cosine schedule from 1×10^{-5}.

The language experiments use RoBERTa-base (Liu et al., 2019) on seven GLUE tasks (Wang et al., 2018) (CoLA, SST-2, MRPC, QQP, MNLI, QNLI, and RTE). For Qwen2.5-1.5B (Qwen Team, 2024), the ten tasks are PIQA (Bisk et al., 2020), the Winograd Schema Challenge (WSC) (Wang et al., 2019), RTE (Wang et al., 2018), ARC-E (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), CommitmentBank (CB) and WiC (Wang et al., 2019), CoLA and MRPC (Wang et al., 2018), and CommonsenseQA (CSQA) (Talmor et al., 2019). Within each model comparison, the task-specific checkpoints and arrival order are shared across methods.

C.2 Task Grouping and Sequence Protocols

We evaluate three nested vision streams to measure performance as the task horizon grows. The short-, medium-, and long-range benchmarks contain the first 8, 14, and all 20 datasets, respectively, in the order listed below; this setup follows OPCM (Tang et al., 2025), MINGLE (Qiu et al., 2025), and MADE-IT (Qiu et al., 2026a):

- Short-Range Benchmark (8-Task Group): (1) SUN397, (2) Stanford Cars, (3) RESISC45, (4) EuroSAT, (5) SVHN, (6) GTSRB, (7) MNIST, and (8) DTD.
- Medium-Range Benchmark (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) Oxford-IIIT Pet, (13) STL-10, and (14) CIFAR-100.
- Long-Range Benchmark (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) Oxford-IIIT Pet, (13) STL-10, (14) CIFAR-100, (15) CIFAR-10, (16) Food-101, (17) Fashion-MNIST, (18) EMNIST, (19) KMNIST, and (20) Rendered SST-2.

Table 4: Randomized Task Orders for the 8-, 14-, and 20-Task Vision Streams

Group	Order	Dataset Order (by Task 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

For each stream length, we evaluate ten independently randomized task orders using seeds 42 through 51. Table 4 records each permutation with dataset IDs defined by the inventory above. For the main vision comparison, Table 1 reports the mean over ten randomized task orders. Controlled analyses additionally report mean ± SD where shown.

C.3 Baseline Details

We compare against reference models, conventional merging rules, continual merging methods, and expert-managed approaches. The pre-trained CLIP checkpoint provides a zero-shot reference, while the independently fine-tuned checkpoints indicate task-specific performance; C. Fine-Tuned denotes one model fine-tuned sequentially on the arriving tasks.

Among conventional merging baselines, Average (SWA) (Izmailov et al., 2018) updates the model with a running parameter average. C. Task Arithmetic (Ilharco et al., 2023) adds a scaled task update to the current merged parameters, whereas C. TIES-Merging (Yadav et al., 2023) trims redundant update coordinates and resolves sign conflicts before integration. C. LW AdaMerging (Yang et al., 2024) learns layer-wise merging weights, and C. LoRA-WEMoE uses a low-rank version of WEMoE (Shen et al., 2026) with a learned gate, also evaluated as a baseline in MINGLE (Qiu et al., 2025).

OPCM (Tang et al., 2025) projects incoming updates relative to historical directions. The expert-managed baselines are MINGLE (Qiu et al., 2025) and MADE-IT (Qiu et al., 2026a). Together with the reference models and conventional merging baselines described above, these form the comparisons reported in Tables 1 and 2.

C.4 Evaluation Metrics and Search Settings

Metrics. Final average performance is computed as $\frac{1}{T}\sum_{i=1}^{T} S_{T,i}$, which we denote by ACC for vision accuracy and Avg. for heterogeneous language scores. Here, $S_{t,i}$ denotes the performance on task i after processing task t. Backward transfer is computed as $\mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(S_{T,i} - S_{i,i}\right)$. For the main comparisons, ACC or Avg. is computed from the displayed per-task scores and rounded once (half up) to the reported precision. We additionally report expert occupancy, storage overhead, merge-time cost, and the online proxies F_t, P_t, and C_t. Long-horizon constraint behavior is quantified by $\mathrm{CV}_j(T) = \left[\frac{1}{T}\sum_{t=1}^{T} u_{t,j}\right]_+$.

Current-task retention. Current Ret. measures the average fraction of incoming-task performance retained immediately after merging:

$$\mathrm{CurrentRet} = \frac{1}{T}\sum_{t=1}^{T}\frac{S_{t,t}}{S_t^{\mathrm{task}}+\epsilon}\times 100\%, \tag{43}$$

where S_t^{task} is the performance of the incoming task-specific checkpoint on task t, $S_{t,t}$ is the system's performance on that task immediately after the t-th merge, and $\epsilon > 0$ prevents zero denominators.

Immediate historical degradation. Hist. Deg. measures the average positive performance drop on previously learned tasks induced by each merge:

$$D_t^{\mathrm{hist}} = \frac{1}{t-1}\sum_{i=1}^{t-1}[S_{t-1,i}-S_{t,i}]_+, \qquad \mathrm{HistDeg} = \frac{1}{T-1}\sum_{t=2}^{T}D_t^{\mathrm{hist}}, \tag{44}$$

where $[x]_+ = \max(x, 0)$. Unlike BWT, which measures the final change relative to performance at task arrival, HistDeg measures immediate positive historical degradation; improvements on one task do not offset drops on another. On vision tasks, HistDeg is reported in percentage points. Both metrics are computed within each task order and summarized as mean ± SD over the ten randomized task orders.

Finite-budget search metrics. To evaluate search quality under a common scale, we use the normalized best-so-far solver gap. For solver variant m at arrival t, the best objective found within budget K is $\Psi_{t,m}^{(K)} = \min_{a\in\mathcal{C}_{t,m}^{(K)}}\Psi_t(a)$. We use Ψ_t^{ref} to represent the best objective found for the same arrival and fixed pre-merge state over the common reference pool formed by all compared search variants and budgets, and Ψ_t^{init} as the best objective in the shared initial population. The normalized gap is

$$G_{t,m}^{(K)} = \frac{\Psi_{t,m}^{(K)}-\Psi_t^{\mathrm{ref}}}{\Psi_t^{\mathrm{init}}-\Psi_t^{\mathrm{ref}}+\varepsilon}. \tag{45}$$

We average $G_{t,m}^{(K)}$ over arrivals within each task order. This empirical metric is distinct from the exact solver error ξ_t in Eq. 10, which uses the infimum over all feasible actions. Lower values indicate that the finite-budget search approaches the empirical reference more closely.

We report two negative-transfer rates (NTRs). For each evaluated local–transfer pair, pair-level NTR measures whether the transferred challenger is worse than its matched local proposal:

$$\mathrm{NTR}_{\mathrm{pair}} = \frac{1}{|\mathcal{V}|}\sum_{(t,j)\in\mathcal{V}}\mathbb{I}\left[\Psi_t(a_{t,j}^{\mathrm{tr}}) > \Psi_t(a_{t,j}^{\mathrm{loc}})+\delta\right], \tag{46}$$

where $\mathcal{V}$ contains all valid local–transfer pairs and δ is a small tolerance for numerical ties. Solver-level NTR instead measures whether historical transfer degrades the final finite-budget solution relative to the matched no-transfer solver:

$$\mathrm{NTR}_{\mathrm{solver}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\left[\Psi_t(\hat{a}_t^{\mathrm{tr}}) > \Psi_t(\hat{a}_t^{\mathrm{base}})+\delta\right]. \tag{47}$$

The matched solvers use the same pre-merge state, task order, initial population, evaluation budget, and random seed. Each metric is first aggregated within one task order and then reported as mean ± sample standard deviation over the ten randomized task orders.

Expert Capacity. All managed layers use the same expert capacity, i.e., $M_l = M$ for all $l \in \mathcal{L}$. The main expert-managed experiments use a maximum capacity of $M = 8$, while the capacity-stress experiments additionally consider $M \in \{2, 4, 8\}$.

Search Protocol. Unless otherwise specified, DebtMerge uses a population size of 10, an initial historical-transfer probability of 0.5, and $K = 150$ complete-action evaluations per expert arrival. Search-budget experiments consider $K \in \{50, 100, 150, 200, 250, 300\}$. All compared search variants receive the same number of complete-action evaluations.

D Detailed Results

D.1 Detailed Per-Task Performance of Vision Tasks

Tables 5, 6, and 7 report final accuracy on each of the 20 vision tasks for ViT-B/32, ViT-B/16, and ViT-L/14, respectively. These results show how the aggregate performance in Table 1 is distributed across individual tasks. All task-wise rankings and differences below are computed from the displayed scores, with accuracy differences expressed in percentage points.

Performance across Backbones. DebtMerge achieves the strictly highest accuracy among the listed methods on 15, 16, and 13 of the 20 tasks for ViT-B/32, ViT-B/16, and ViT-L/14, respectively. It also shares the highest score on Food-101 with ViT-B/32, where both methods reach 88.3%, and on STL-10 with ViT-L/14, where the highest score is 99.3%. The corresponding 20-task average accuracies reported in Table 1 are 88.93%, 90.93%, and 93.20%, showing that the aggregate improvements are accompanied by leading performance on a majority of tasks across all three backbones.

Where the Gains Occur. Several of the largest improvements over MADE-IT (Qiu et al., 2026a) occur on character-recognition tasks. On ViT-B/32, DebtMerge reaches 95.6% on EMNIST and 98.1% on KMNIST, exceeding MADE-IT by 24.6 and 26.7 points, respectively. On ViT-B/16, the corresponding improvements are 33.2 points on EMNIST and 12.8 points on KMNIST, while on ViT-L/14 they are 17.0 and 17.5 points. The gains also extend beyond character recognition. For CIFAR-100, DebtMerge improves over MADE-IT by 16.2, 10.6, and 6.0 points on ViT-B/32, ViT-B/16, and ViT-L/14, respectively. For RESISC45, the corresponding improvements are 7.1, 3.4, and 6.2 points. These results indicate that the aggregate improvement arises from multiple task families rather than from a single subset of the benchmark.

Remaining Task-Level Gaps. MADE-IT (Qiu et al., 2026a) remains stronger on SUN397, Cars, and FER2013 for all three backbones. On ViT-B/32, DebtMerge trails MADE-IT by 4.5 points on SUN397, 2.8 points on Cars, and 2.3 points on FER2013. The corresponding gaps on ViT-B/16 are 1.9, 2.0, and 3.2 points, while those on ViT-L/14 are 2.7, 1.0, and 0.4 points. Other methods also retain isolated advantages. Sequential fine-tuning performs better on EuroSAT with ViT-B/16 and ViT-L/14, while MINGLE (Qiu et al., 2025) exceeds DebtMerge by 0.1 point on Oxford-IIIT Pet with ViT-L/14. These results indicate that the overall gains coexist with identifiable task-specific trade-offs.

Relation to the Main Results. The corresponding 20-task BWT values in Table 1 are -0.56, -0.43, and -0.19 for ViT-B/32, ViT-B/16, and ViT-L/14, respectively. Together with the final task accuracies, these results indicate improved aggregate performance with limited average historical degradation under the evaluated protocol. Final task scores alone do not quantify how much each task changes after its arrival, while average BWT may conceal task-specific differences. The contributions of individual control and search components are therefore examined separately in the ablation analysis.

D.2 Detailed Per-Task Performance of Language Tasks

Tables 8 and 9 provide the task-level results underlying Table 2. We examine both the distribution of final task scores and the reported BWT. Differences below are expressed in points on the reported task-score scale.

RoBERTa-base: Gains Concentrated in a Subset of Tasks. DebtMerge obtains the highest scores among the listed methods on MRPC, MNLI, and QNLI, reaching 88.3, 83.9, and 83.6, respectively. Compared with MADE-IT (Qiu et al., 2026a), these results correspond to improvements of 1.3, 15.9, and 6.3 points. Its RTE score also increases slightly from 71.6 to 71.8. In contrast, DebtMerge scores 47.8 on CoLA, 91.9 on SST-2, and 73.6 on QQP, compared with 52.5, 93.1, and 84.2 for MADE-IT, respectively. Sequential fine-tuning retains the highest RTE score of 75.9. The resulting aggregate performance therefore reflects substantial gains on MNLI and QNLI together with regressions on several other tasks. DebtMerge achieves a BWT of -1.62, compared with -1.80 for MADE-IT (Qiu

Table 5: Final per-task accuracy (%) on the 20-task vision benchmark with ViT-B/32. Bold values indicate the highest displayed accuracy in each column, including ties.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers102	PCAM
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. Task Arithmetic	62.0	53.7	60.9	58.1	48.5	48.9	79.4	46.1	61.1	73.4
C. TIES-Merging	62.5	49.1	55.8	50.9	54.6	49.3	82.0	46.7	58.5	69.9
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	61.1	66.0	71.7	66.1	56.0	90.2	40.4	64.9	80.2
MINGLE	67.8	58.3	83.5	90.0	82.9	91.8	98.0	65.3	74.0	66.9
MADE-IT	**75.5**	**77.6**	86.6	86.5	78.0	96.3	80.8	75.1	84.5	84.9
DebtMerge	71.0	74.8	**93.7**	**99.0**	**96.6**	**98.3**	**99.5**	**78.3**	**88.0**	**88.0**

Method	FER2013	Oxford-IIIT Pet	STL-10	CIFAR-100	CIFAR-10	Food-101	Fashion-MNIST	EMNIST	KMNIST	Rendered SST-2
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. Task Arithmetic	51.4	82.3	94.9	64.6	91.4	71.9	73.9	17.8	12.2	59.9
C. TIES-Merging	49.5	81.3	95.2	63.7	91.2	70.2	73.7	17.8	16.9	59.8
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	65.0	85.5	97.0	72.6	94.1	81.5	85.4	50.4	65.2	67.1
MADE-IT	**70.3**	91.8	91.8	71.9	89.7	**88.3**	84.8	71.0	71.4	**71.3**
DebtMerge	68.0	**92.3**	**97.8**	**88.1**	**97.6**	**88.3**	**94.5**	**95.6**	**98.1**	71.1

Table 6: Final per-task accuracy (%) on the 20-task vision benchmark with ViT-B/16. Bold values indicate the highest displayed accuracy in each column, including ties.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers102	PCAM
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. Task Arithmetic	65.8	57.5	63.8	59.5	64.7	54.0	88.0	45.3	67.5	67.1
C. TIES-Merging	64.2	52.9	60.9	53.0	62.8	48.8	88.4	45.0	61.3	68.5
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	71.5	64.9	85.3	90.0	87.5	90.1	97.1	62.7	82.6	80.6
MADE-IT	**77.4**	**85.2**	91.8	95.2	67.8	88.0	89.6	76.1	93.5	88.6
DebtMerge	75.5	83.2	**95.2**	98.1	**97.2**	**98.4**	**99.6**	**82.0**	**94.1**	**90.3**

Method	FER2013	Oxford-IIIT Pet	STL-10	CIFAR-100	CIFAR-10	Food-101	Fashion-MNIST	EMNIST	KMNIST	Rendered SST-2
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. Task Arithmetic	50.7	89.3	97.0	68.0	93.1	80.3	75.7	18.1	16.7	61.8
C. TIES-Merging	50.4	87.9	96.3	63.1	91.7	78.0	75.0	23.4	24.9	61.5
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	67.6	92.7	97.4	74.0	95.3	87.7	87.4	73.5	79.9	74.0
MADE-IT	**72.3**	94.7	92.7	78.1	89.3	91.6	84.4	62.1	85.2	57.2
DebtMerge	69.1	**94.9**	**98.3**	**88.7**	**98.3**	**91.9**	**94.4**	**95.3**	**98.0**	**76.0**

Table 7: Final per-task accuracy (%) on the 20-task vision benchmark with ViT-L/14. Bold values indicate the highest displayed accuracy in each column, including ties.

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Flowers102	PCAM
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. Task Arithmetic	70.4	74.1	73.9	66.3	69.9	65.6	95.1	56.6	78.6	70.4
C. TIES-Merging	69.7	70.3	65.3	47.9	76.1	63.6	94.7	54.4	77.9	72.3
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	75.9	83.4	87.8	88.7	91.1	94.5	98.4	70.8	94.8	75.3
MADE-IT	**82.6**	**92.7**	90.6	90.2	96.3	98.5	95.4	82.2	**98.2**	86.4
DebtMerge	79.9	91.7	**96.8**	98.8	**97.8**	**99.3**	**99.7**	**84.8**	97.9	**91.4**

Method	FER2013	Oxford-IIIT Pet	STL-10	CIFAR-100	CIFAR-10	Food-101	Fashion-MNIST	EMNIST	KMNIST	Rendered SST-2
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. Task Arithmetic	55.7	94.2	98.6	79.1	96.6	87.6	80.8	17.6	10.6	63.6
C. TIES-Merging	57.6	93.5	97.8	74.0	95.6	84.7	79.7	20.2	12.6	58.4
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	67.7	**96.0**	98.7	81.4	97.1	90.6	90.6	60.7	88.6	79.8
MADE-IT	**74.7**	95.6	98.4	86.9	97.4	94.7	92.9	78.4	80.8	79.9
DebtMerge	74.3	95.9	**99.3**	**92.9**	**99.1**	**94.8**	**95.1**	**95.4**	**98.3**	**80.8**

Table 8: Final per-task scores on the seven-task language benchmark with RoBERTa-base. Avg. is calculated from the displayed per-task scores; BWT is the reported backward transfer in score points. The same aggregates appear in Table 2. Bold values indicate the highest displayed score in each column.

Method	CoLA	SST2	MRPC	QQP	MNLI	QNLI	RTE	Avg.	BWT
C. Fine-Tuned	21.4	84.8	76.9	63.8	67.2	82.6	**75.9**	67.5	-
Average (SWA)	19.6	83.5	77.3	41.0	53.2	66.9	58.5	57.1	-12.09
C. Task Arithmetic	19.9	83.1	78.5	31.6	71.5	74.7	66.4	60.8	-5.39
C. TIES-Merging	32.7	79.9	75.7	57.0	49.2	49.3	53.1	56.7	-3.74
C. LW AdaMerging	1.3	73.5	76.4	78.0	27.0	36.1	47.7	48.6	-13.59
C. LoRA-WEMoE	5.1	74.9	75.0	75.3	31.9	40.0	45.8	49.7	-11.83
OPCM	28.2	87.4	76.2	70.4	62.0	55.9	53.4	61.9	-3.99
MINGLE	52.1	91.5	65.3	78.5	71.6	52.0	67.5	68.4	-2.13
MADE-IT	**52.5**	**93.1**	87.0	**84.2**	68.0	77.3	71.6	76.2	-1.80
DebtMerge	47.8	91.9	**88.3**	73.6	**83.9**	**83.6**	71.8	**77.3**	**-1.62**

et al., 2026a) and -2.13 for MINGLE (Qiu et al., 2025), indicating a smaller average decline from the scores measured at task arrival.

Qwen2.5-1.5B: Gains across Most Tasks. DebtMerge achieves the highest score on seven of the ten tasks and exceeds MADE-IT (Qiu et al., 2026a) on eight. The largest gains occur on CB and ARC-E, where DebtMerge reaches 53.1 and 85.3, compared with 50.0 and 82.3 for MADE-IT, respectively. Additional improvements are observed on PIQA, WSC, RTE, WinoGrande, WiC, and CSQA. The two regressions occur on CoLA and MRPC: DebtMerge obtains 51.9 and 82.4, whereas MADE-IT achieves 52.8 and 83.2, respectively. Although DebtMerge also improves over MADE-IT on WiC, MINGLE (Qiu et al., 2025) retains the highest WiC score of 73.8, compared with 72.8 for DebtMerge. The resulting average score is 72.0 for DebtMerge, compared with 71.1 for MADE-IT and 67.2 for MINGLE. The corresponding BWT values are -0.51, -0.65, and -0.81, respectively.

Scope of the Language Results. Both language settings support the main-text observation of higher reported average performance and less negative BWT among the compared merging methods. The task-level pattern differs: RoBERTa-base benefits most from a few large improvements, whereas Qwen2.5-1.5B improves over the strongest aggregate merging baseline on most tasks. Negative BWT remains in both settings, and several tasks still favor a baseline. These results support the method's empirical applicability to the two evaluated language models, with task-dependent gains and residual

Table 9: Final per-task scores on the ten-task language benchmark with Qwen2.5-1.5B. Avg. is calculated from the displayed per-task scores; BWT is the reported backward transfer in score points.

Method	PIQA	WSC	RTE	ARC-E	Winogrande	CB	WiC	CoLA	MRPC	CSQA	Avg.	BWT
C. Fine-Tuned	64.7	55.2	58.5	54.4	60.8	45.6	61.3	27.9	71.5	77.8	57.8	-
Average (SWA)	49.8	42.3	47.7	24.9	51.5	33.7	50.0	-4.0	74.6	18.9	38.9	-3.86
C. Task Arithmetic	59.5	57.7	52.7	29.3	49.6	38.5	50.8	19.1	41.9	21.8	42.1	-4.50
C. TIES-Merging	66.3	63.5	52.7	27.2	53.7	30.2	49.8	12.2	15.8	28.2	40.0	-5.02
C. LW AdaMerging	61.4	54.8	55.2	34.6	56.7	38.6	55.1	13.9	49.7	31.7	45.2	-13.87
C. LoRA-WEMoE	66.8	58.6	59.4	48.1	62.6	42.4	60.9	24.9	64.2	49.7	53.8	-9.42
OPCM	53.6	56.7	48.4	28.4	54.7	35.0	55.5	-2.4	74.5	22.0	42.6	-6.25
MINGLE	80.1	61.5	67.5	83.3	73.9	33.9	**73.8**	47.9	74.3	75.8	67.2	-0.81
MADE-IT	81.6	63.5	73.2	82.3	74.8	50.0	72.2	**52.8**	**83.2**	77.2	71.1	-0.65
DebtMerge	**82.5**	**64.2**	**74.6**	**85.3**	**75.4**	**53.1**	72.8	51.9	82.4	**78.2**	**72.0**	**-0.51**

historical degradation. The comparisons describe point estimates; the tables do not establish statistical significance for the observed differences.

D.3 Analysis of Debt-Based Control

We first examine the long-horizon control mechanism beyond the component ablations reported in Table 3. The main-text comparison isolates the effect of retaining accumulated debt information, while the experiments below further compare alternative state designs and separate the two roles of debt in candidate scoring and geometry initialization. Unless otherwise specified, historical recipe transfer is disabled so that the controller can be studied independently of the transfer mechanism.

Complementary Roles of the Three Debts. Removing one debt at a time produces distinct failure modes in Table 3: removing stability debt primarily increases historical degradation, removing plasticity debt weakens incoming-knowledge retention, and removing resource debt increases expert occupancy. These results indicate that the three debts regulate complementary aspects of continual expert evolution rather than acting as interchangeable regularizers.

Accumulated Debt versus Stateless and Smoothed Control. Table 10 uses a controlled 20-task ViT-B/32 setting with the same expert representation, routing mechanism, action executor, searchable geometry domain, population size $N_p = 10$, and evaluation budget $K = 150$. Reset-state control discards previous constraint deviations; fixed weights use tuned, time-invariant proxy coefficients; the leaky controller uses $Q_{t+1} = [\beta Q_t + u_t]_+$; and virtual queues use the undiscounted update in Eq. 3. Fixed-weight and leaky controllers use matched hyperparameter-tuning budgets.

Virtual queues achieve the strongest ACC and BWT while producing the smallest time-average stability and plasticity violations. Compared with the leaky controller, the predictive improvement is moderate, whereas the reductions in constraint violations and terminal resource debt are more pronounced. This supports the role of persistent virtual queues as a mechanism for retaining and regulating accumulated deviations rather than merely providing another set of adaptive weights.

Table 10: Controlled comparison of debt-state designs on the 20-task ViT-B/32 stream. Historical transfer is disabled and all variants use the same geometry search space. CV_S and CV_P denote time-average constraint violations. Values are mean ± SD.

Metric	Reset-state	Fixed weights	Leaky queue	DebtMerge (virtual queues)
ACC (%)↑	86.84 ± 0.73	87.32 ± 0.60	87.79 ± 0.54	$\mathbf{88.05 \pm 0.50}$
BWT (pp)↑	-1.56 ± 0.35	-1.03 ± 0.27	-0.78 ± 0.23	$\mathbf{-0.61 \pm 0.20}$
$\mathrm{CV}_S \downarrow$	0.0161 ± 0.0008	0.0114 ± 0.0007	0.0089 ± 0.0011	$\mathbf{0.0054 \pm 0.0010}$
$\mathrm{CV}_P \downarrow$	0.0221 ± 0.0016	0.0159 ± 0.0013	0.0107 ± 0.0015	$\mathbf{0.0062 \pm 0.0013}$
$Q^B_{T+1}/T \downarrow$	0.0369 ± 0.0023	0.0107 ± 0.0017	0.0048 ± 0.0010	$\mathbf{0.0007 \pm 0.0006}$
Final occupancy $C_T \downarrow$	0.866 ± 0.021	0.744 ± 0.019	0.709 ± 0.020	$\mathbf{0.658 \pm 0.022}$
Mean occupancy $\bar{C} \downarrow$	0.497 ± 0.008	0.435 ± 0.007	0.416 ± 0.006	$\mathbf{0.388 \pm 0.006}$

Separating Objective Reweighting from Geometry Initialization. Fig. 4 uses a 2×2 design that independently varies fixed versus debt-dependent objective weights and fixed versus debt-guided

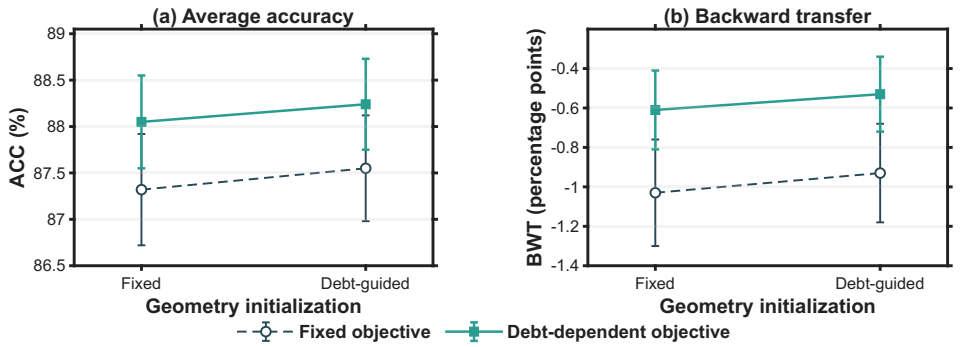


Fig. 4: Objective weighting and geometry initialization on the 20-task ViT-B/32 stream, with historical transfer disabled. Markers and error bars show mean ± SD.

geometry initialization. Historical transfer is disabled in all four variants, and the searchable (r, λ, α) domain is identical. For the fixed-objective/debt-anchor variant, a shadow debt state is maintained only to construct the geometry anchor and does not affect candidate scoring. Debt-dependent objective weighting provides the larger contribution: with fixed initialization, ACC increases from 87.32% to 88.05%; with the debt anchor, it increases from 87.55% to 88.24%. Debt-guided geometry initialization provides an additional but smaller improvement under both objective choices, with BWT following the same pattern. The results therefore separate the effect of accumulated debt in the per-arrival objective from its role in initializing the integration geometry.

D.4 Finite-Budget Historical Transfer

We next examine the finite-budget optimization component from three perspectives: how historical transfer behaves as the evaluation budget changes, whether the results depend strongly on the underlying evolutionary operator, and how the proposed transfer strategy compares with alternative finite-budget search and recipe-reuse schemes. Unless otherwise specified, the comparisons use the 20-task ViT-B/32 stream with population size $N_p = 10$ and count complete-action evaluations toward the same search budget.

Search Efficiency under Different Evaluation Budgets. We vary $K \in \{50, 100, 150, 200, 250, 300\}$ while fixing the population size to 10. The initial population is included in the total budget, and subsequent local and transferred proposals are evaluated sequentially according to Algorithm 1. Fig. 5 shows that historical transfer provides the largest advantage when the evaluation budget is tight. Context-conditioned block transfer reaches lower normalized solver gaps and higher final accuracy with fewer candidate evaluations, whereas Local DE gradually narrows the gap as K increases. The advantage therefore becomes smaller when sufficient evaluations are available, supporting the interpretation of historical transfer as a finite-budget search-efficiency mechanism rather than a change to the underlying debt-conditioned objective.

Choice of Evolutionary Search Operator. We additionally compare GA-, ES-, and DE-based local search while holding the DebtMerge controller, structured action space, expert capacity, historical transfer mechanism, population size, and candidate-evaluation budget fixed. All variants use $K = 150$ complete-action evaluations. Fig. 6 shows that the DE-based variant achieves the highest ACC and least negative BWT across the three CLIP backbones, with ES generally providing intermediate performance and GA producing the weakest results. Since the evolutionary operator itself is not a contribution of DebtMerge, this comparison is used only to justify the DE-style perturbation and crossover adopted in the default implementation.

Alternative Search and Historical Recipe Reuse. We compare five search strategies on the 20-task ViT-B/32 stream under the same task orders, evaluation protocol, and complete-action evaluation budget $K = 150$. Random seeds are matched for stochastic solvers, and population-based solvers use $N_p = 10$. The solver-gap panel evaluates the normalized best-so-far gap G on common frozen pre-merge states using the shared reference pool in Appendix C.4. The ACC panel instead evaluates

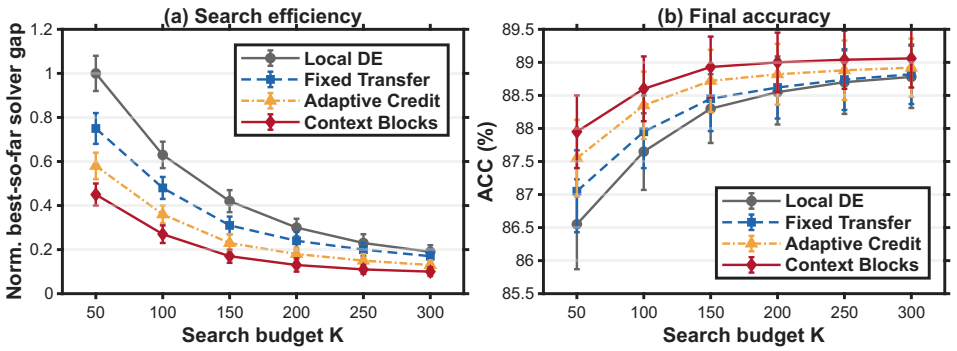


Fig. 5: Finite-budget search on the 20-task ViT-B/32 stream. Error bars show SD over ten task orders. Results at $K = 150$ match Table 3C.

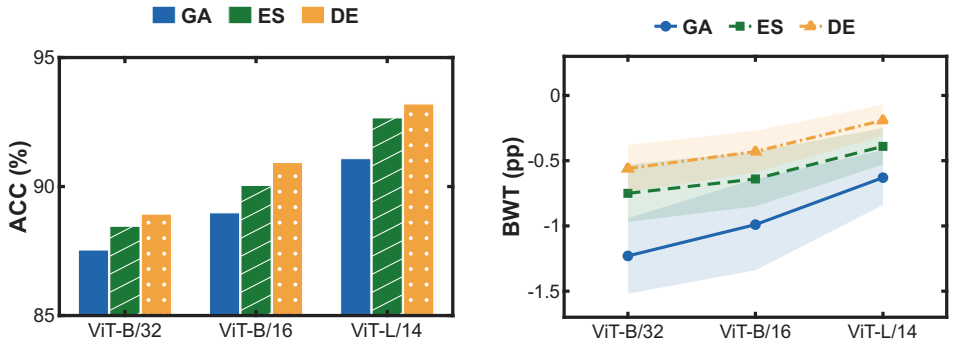


Fig. 6: Evolutionary search operators across three CLIP backbones under the same controller and evaluation budget $K = 150$. Bars and curves show means; shaded regions in the right panel show SD across task orders.

each method through an independent closed-loop rollout under the shared protocol, so its pre-merge states evolve separately. Relative merge time includes optimization at each arrival and is normalized to context-conditioned block transfer. The Local DE and context-conditioned block transfer results coincide with the corresponding configurations in Table 3C.

Fig. 7 shows a quality–cost trade-off. Layer-wise coordinate search provides the lowest merge-time cost among the structured alternatives, but its restricted local updates produce a larger solver gap and lower final accuracy than historical transfer. Local DE improves over random search but cannot reuse optimization experience from previous arrivals. Adapting a complete historical recipe improves search quality and ACC, while context-conditioned block transfer further reduces the solver gap and negative-transfer rate. Solver NTR on the frozen states decreases from $3.2 \pm 2.3\%$ for adapted full-recipe transfer to $2.1 \pm 1.8\%$ for context-conditioned block transfer. The improvement from semantic decomposition is modest in final ACC but more apparent in finite-budget search quality and transfer reliability. The coordinate-search comparison further clarifies the role of the joint evolutionary solver. When layer-wise decisions can be optimized approximately in isolation, coordinate updates provide an efficient alternative. DebtMerge instead searches complete structured recipes and supports historical transfer across expert structure, knowledge composition, and integration geometry. The comparison therefore exposes a computational trade-off rather than assuming that joint evolutionary search is uniformly preferable.

D.5 Historical Transfer Provenance

We further examine which historical optimization episodes contribute to later merge decisions. Fig. 8 visualizes the average historical retrieval probability $p_{i \to t}$ together with the corresponding target-side transfer reward $R_{i \to t}$ over the ten task orders. The retrieval distribution is selective rather than uniform, while transfer rewards vary across historical sources and target arrivals. This behavior is

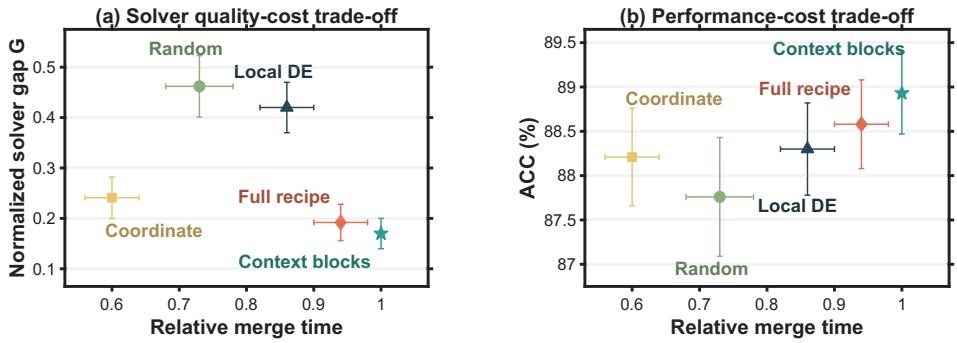


Fig. 7: Search quality–cost trade-offs on the 20-task ViT-B/32 stream at $K = 150$. Time is relative to context-conditioned block transfer; markers and error bars show mean ± SD.

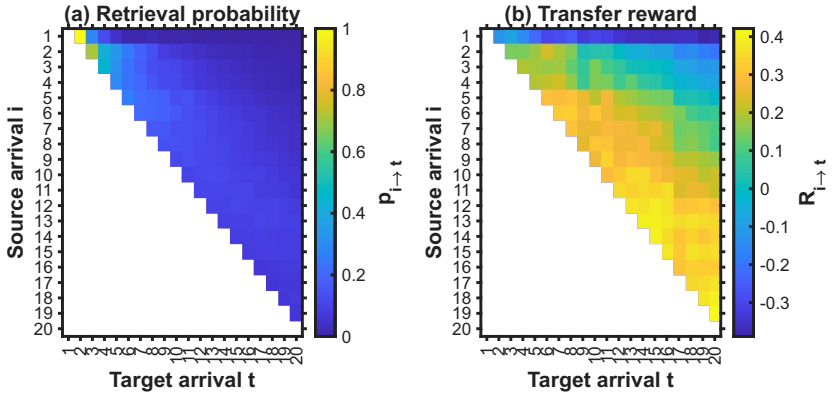


Fig. 8: Historical transfer provenance on the 20-task ViT-B/32 benchmark. The two heatmaps show the mean historical recipe retrieval probability $p_{i\to t}$ and the corresponding target-side transfer reward $R_{i\to t}$ over ten task orders. Only historical arrivals $i < t$ are eligible for transfer, producing the upper-triangular structure.

consistent with the intended role of context-and-credit conditioning: historical experience is reused selectively and re-evaluated under the current debt-conditioned objective rather than transferred indiscriminately.

D.6 Expert Capacity and Storage Accounting

We next examine the resource side of DebtMerge from two complementary perspectives. The capacity-stress experiment varies the number of available expert slots, whereas the controlled storage comparison distinguishes expert occupancy from total implementation memory.

Expert-Capacity Stress. Fig. 9 reports results under $M \in \{2, 4, 8\}$ on the 20-task vision stream. Reducing the capacity increases normalized expert occupancy for all three backbones, indicating that a larger fraction of the available expert bank must be reused or modified as allocation becomes more restrictive. Reducing M from 8 to 2 decreases ACC by 4.03, 3.68, and 2.88 points and degrades BWT by 1.16, 1.03, and 0.89 points on ViT-B/32, ViT-B/16, and ViT-L/14, respectively. The smaller degradation on larger backbones is consistent with greater robustness to restricted expert capacity. For $M = 2, 4, 8$, respectively, storage relative to the base model is 1.31 ± 0.03, 1.54 ± 0.04, and 1.89 ± 0.05 for ViT-B/32; 1.30 ± 0.03, 1.52 ± 0.04, and 1.86 ± 0.05 for ViT-B/16; and 1.28 ± 0.03, 1.49 ± 0.04, and 1.82 ± 0.05 for ViT-L/14. These storage increases accompany the ACC and BWT improvements in Fig. 9.

Storage Accounting under Common Caps. The resource debt regulates expert occupancy through C_t rather than total implementation memory. We therefore separately report deployed storage

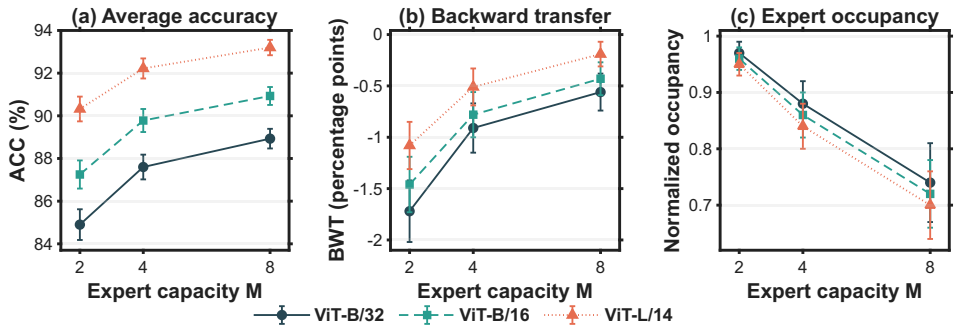


Fig. 9: Expert-capacity stress across three CLIP backbones on the 20-task vision benchmark. Markers and error bars show mean ± SD over ten randomized task orders at each capacity.

Table 11: Controlled policies under common deployed/online storage caps of $2.0\times$ /$2.5\times$ base-model storage on the 20-task ViT-B/32 stream. The MADE-IT-style policy is a controlled analogue implemented in the shared host. Values are mean ± SD.

Controlled policy	ACC↑	BWT↑	Deployed / base	Online / base
MADE-IT-style policy	87.69 ± 0.58	-0.87 ± 0.25	1.770 ± 0.012	2.110 ± 0.024
Tuned fixed weights	88.03 ± 0.53	-0.76 ± 0.23	1.790 ± 0.012	2.200 ± 0.023
DebtMerge	$\mathbf{88.72 \pm 0.47}$	$\mathbf{-0.49 \pm 0.19}$	1.800 ± 0.011	2.230 ± 0.022

and online retained storage. Deployed storage contains the components required for inference, whereas online retained storage additionally includes the historical spectral sketch, recipe archive, and controller state. Table 11 compares policies under common deployed and online storage caps of $2.0\times$ and $2.5\times$ the base-model storage, respectively. To isolate policy differences, all variants use the same expert factorization, managed modules, router, task heads, compression rule, and action executor. The MADE-IT-style variant is a controlled analogue implemented in this shared setting and should not be interpreted as the native MADE-IT implementation reported in Table 1. DebtMerge achieves the strongest ACC and BWT while remaining within both caps. Its online retained storage is moderately larger because the controller additionally maintains the debt state, spectral sketch, and recipe archive. The resource mechanism should therefore be interpreted as regulating expert-bank pressure rather than minimizing total implementation memory.

D.7 Prediction-Preservation Analysis

The online stability proxy F_t measures parameter-branch perturbation, whereas the theoretical prediction-preservation analysis characterizes prediction-level changes through expert-function and routing drift. We therefore examine how these signals relate to observed historical degradation. At each arrival $t \geq 2$, we first average $B_E(x)$ and $B_R(x)$ within each historical task, then average equally over the $t-1$ historical tasks to obtain task-balanced signals $\bar{B}_{E,t}$ and $\bar{B}_{R,t}$. Within each task order, we compute Spearman correlations across arrivals between D_t^{hist} and each of F_t, $\bar{B}_{E,t}$, and $\bar{B}_{E,t} + \bar{B}_{R,t}$. Table 12 reports mean ± SD of these correlations over ten task orders.

Table 12 shows that task-balanced expert-function drift $\bar{B}_{E,t}$ correlates more strongly with historical degradation than the parameter-level proxy F_t, while jointly accounting for expert and routing changes through $\bar{B}_{E,t} + \bar{B}_{R,t}$ yields the strongest association across all three backbones. This supports the distinction between the data-free proxy used for online control and the prediction-level quantities used in the preservation analysis.

Table 13 evaluates the sufficient prediction-preservation condition by comparing the pre-merge classification margin with the combined expert-function and routing-drift bound. The certificate covers a substantial fraction of historical predictions, and no certified sample violates the sufficient condition in the evaluated runs. Coverage decreases as the task stream grows because expert functions and routing decisions accumulate larger changes, making the certificate progressively more conservative.

Table 12: Spearman correlation between different model-change signals and observed historical degradation on the 20-task vision benchmark. Correlations are computed across arrivals within each task order and reported as mean ± SD over ten orders.

Signal	ViT-B/32	ViT-B/16	ViT-L/14
Parameter perturbation (F_t)	0.72 ± 0.05	0.74 ± 0.04	0.70 ± 0.05
Expert drift ($\bar{B}_{E,t}$)	0.80 ± 0.04	0.82 ± 0.04	0.81 ± 0.03
Expert + routing drift ($\bar{B}_{E,t} + \bar{B}_{R,t}$)	$\mathbf{0.86\pm0.03}$	$\mathbf{0.88\pm0.03}$	$\mathbf{0.87\pm0.03}$

Table 13: Historical prediction-preservation certificate for the indicated task horizons. Coverage is the percentage of previously correct historical samples satisfying the sufficient margin condition; Certified Viol. counts post-merge prediction changes among certified samples. Expert Drift and Routing Drift report the means of $B_E(x)$ and $B_R(x)$, respectively, over the evaluated historical samples, as defined in Appendix B.6.

Tasks	Coverage↑	Certified Viol.	Expert Drift B_E	Routing Drift B_R
8	$84.1\pm1.5\%$	0	0.04 ± 0.01	0.02 ± 0.01
14	$78.5\pm1.8\%$	0	0.31 ± 0.04	0.17 ± 0.03
20	$72.3\pm2.1\%$	0	0.34 ± 0.05	0.18 ± 0.03

D.8 Hyperparameter Sensitivity

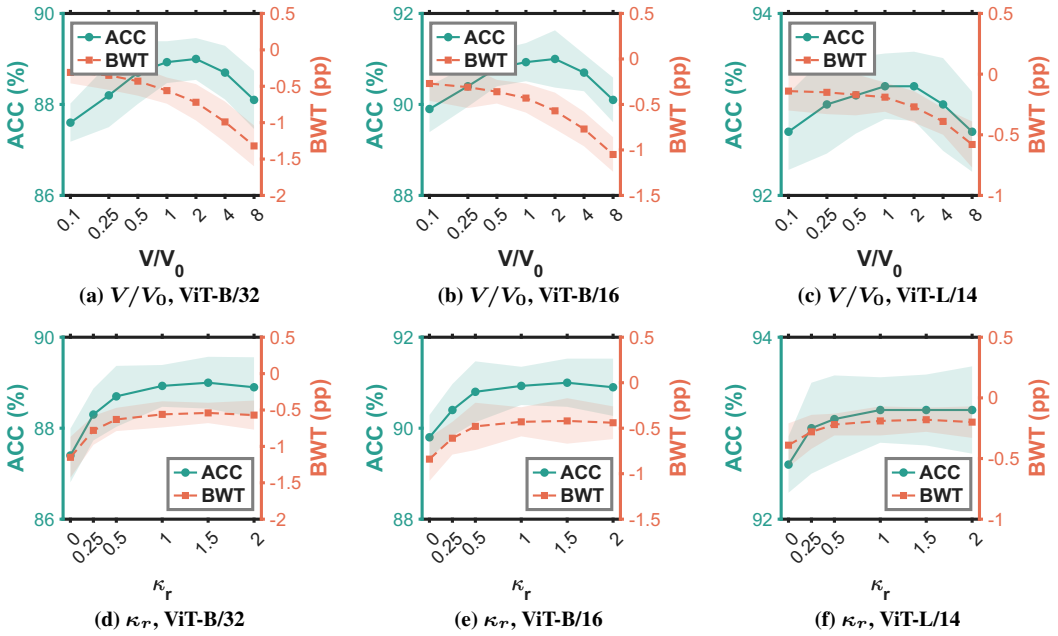
We examine two hyperparameters intrinsic to the long-horizon controller and debt-guided geometry initialization: the normalized control weight $\nu_V = V/V_0$ and the normalized protected-rank limit κ_r, where $r^{(l)}_{\max}(\kappa_r) = \lfloor \kappa_r r^{(l)}_{\max,0} \rfloor$. We evaluate $\nu_V \in \{0.1, 0.25, 0.5, 1, 2, 4, 8\}$ and $\kappa_r \in \{0, 0.25, 0.5, 1, 1.5, 2\}$ on ViT-B/32, ViT-B/16, and ViT-L/14 under the 20-task protocol, with all other settings fixed. The default configuration uses $\nu_V = 1$ and $\kappa_r = 1$.

Control Weight. Fig. 10 shows a consistent stability–plasticity trade-off as ν_V varies. Across all three backbones, moderate values $\nu_V \in [0.5, 2]$ maintain similar ACC, while excessively large V reduces the relative influence of the accumulated debts and produces more negative BWT. For example, ViT-B/32 improves from 87.60% ACC at $\nu_V = 0.1$ to 88.93% at the default setting and remains close at 89.00% for $\nu_V = 2$, whereas BWT becomes increasingly negative as V grows further. The broad plateau around the default value indicates that performance does not depend on a sharply tuned control weight.

Protected-Rank Limit. The effect of κ_r is strongest when the admissible historical subspace is restrictive. On ViT-B/32, increasing κ_r from 0 to 1 improves ACC from 87.40% to 88.93% and BWT from -1.15 to -0.56; further expansion changes performance only marginally. The same saturation pattern is observed on ViT-B/16 and ViT-L/14. Because the actual protected rank remains searchable, enlarging the upper limit beyond a moderate value expands the feasible geometry without forcing the optimizer to protect additional directions.

E Statement about usage of LLMs

In this work, we did not use generative AI tools for tasks requiring disclosure, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for language editing, LaTeX and scientific-figure formatting, checking the arithmetic consistency of reported results, formatting references, and sourcing or searching for information. We reviewed all AI-assisted work. In particular, information and references suggested by AI tools were verified against the original sources, and arithmetic checks were independently verified against the reported experimental results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.



(a) V/V_0, ViT-B/32 (b) V/V_0, ViT-B/16 (c) V/V_0, ViT-L/14

(d) κ_r, ViT-B/32 (e) κ_r, ViT-B/16 (f) κ_r, ViT-L/14

Fig. 10: Detailed sensitivity to the normalized control weight V/V_0 and protected-rank limit κ_r across ViT-B/32, ViT-B/16, and ViT-L/14 on the 20-task vision stream. Curves and shaded regions show mean $\pm$ SD over ten randomized task orders.