SELECT, SHARE, PROTECT: MANOVA-GUIDED ADAPTATION FOR CONTINUAL MULTIMODAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning enables multimodal large language models (MLLMs) to adapt to evolving audio, visual, and joint audio–visual understanding tasks as new data arrive. Effective adaptation requires learning plasticity and memory stability, yet balancing them remains challenging. Updating shared parameters supports new learning but risks forgetting, while freezing them helps preserve earlier knowledge but restricts adaptation. These trade-offs motivate M**A**NOVA-guided **s**election and shared **a**daptation with selective **p**rotection (**ASAP**), a sparse fine-tuning method for continual multimodal learning. We *select* shared, audio, video, and interaction blocks of pretrained matrices using a multivariate analysis of variance (MANOVA) factorial decomposition of loss gradients. Tasks *share* updates at modality-relevant blocks to reuse learned knowledge. To balance memory stability and learning plasticity, we *protect* updates important to earlier tasks using loss-gradient importance scores while leaving other active updates trainable for new learning. Unselected weights remain frozen, and active updates merge into pretrained weights to avoid additional forward-pass latency. Under mild assumptions, our analysis bounds increases in earlier-task loss in terms of the importance left unprotected. Experiments on MUSIC-AVQA and CMU-MOSI across multiple task orders demonstrate superior new-task learning and final accuracy compared with the state-of-the-art baselines, alongside competitive retention. Efficiency comparisons show comparable memory and latency among selected methods. The implementation of ASAP is available at `https://anonymous.4open.science/r/ASAP-ICLR/`.

1 INTRODUCTION

Multimodal large language models (MLLMs) use a common language interface to perform tasks involving audio, images, and video (Lin et al., 2024; Tang et al., 2024; Zhang et al., 2023). A deployed music assistant, for example, may successively learn to recognize sounds from audio, locate instruments in video, and associate sounds with their visible sources (Figure 1). The corresponding datasets may arrive at different times and require different combinations of sensory inputs, yet users expect earlier capabilities to remain available after each update. This setting motivates continual multimodal learning under changing modality availability (Wang et al., 2024; Guo et al., 2025).

Such learning requires *learning plasticity* to learn the current task and *memory stability* to retain earlier-task performance after subsequent updates. In an MLLM, these objectives are coupled through shared parameters: an update that improves audio understanding may also affect joint audio–visual reasoning. This sharing allows knowledge reuse, but later updates can also disrupt earlier capabilities and cause catastrophic forgetting. Freezing shared parameters limits this interference while restricting later refinement. The central challenge is therefore to identify useful sharing across modality combinations and selectively preserve the updates on which earlier tasks depend, within a fixed adaptation budget.

Existing methods trade off sharing against protection. Continually training one LoRA adapter (Hu et al., 2022) supports new learning and knowledge reuse but exposes earlier knowledge to interference. Freezing earlier adapters or experts preserves their updates for reuse but prevents their direct

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.

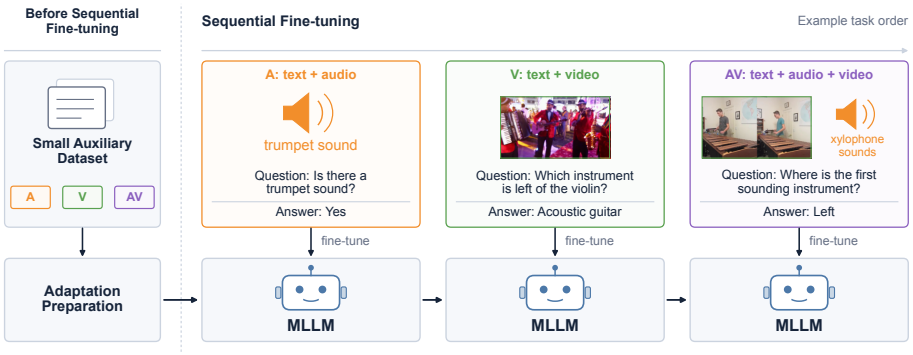


Figure 1: **Continual learning of audio, visual, and joint audio–visual understanding tasks.** An MLLM learns tasks pairing text with audio (A), video (V), or both (AV) in sequence. Representative samples come from distinct MUSIC-AVQA clips (Li et al., 2022).

refinement by later tasks (Yu et al., 2024; Ge et al., 2025). Adding adapters requires extra parameter storage, while compressing and integrating them adds training computation (Meng et al., 2026). OSFT and DGG limit interference by constraining or guiding updates (Nayak et al., 2026; Li et al., 2026), and SMT and SPU select parameters within pretrained weights (He et al., 2025; Zhang et al., 2024), but none of these four uses the modalities that tasks share to decide what to update or protect. These trade-offs motivate selecting modality-relevant parameters within pretrained weights and protecting only those on which earlier tasks depend, while keeping the remaining shared parameters trainable for subsequent learning.

We introduce MANOVA-guided **s**election and shared **a**daptation with selective **p**rotection (**ASAP**), a sparse fine-tuning method that separates *which learned updates a task uses* from *which it may modify*. Modality availability determines which updates participate in prediction, while earlier-task importance determines which active updates remain trainable. Before sequential fine-tuning, we evaluate loss gradients on a small labeled auxiliary dataset spanning the three task types, with aligned audio and video, using text alone and each sensory-input combination. A factorial decomposition, as in multivariate analysis of variance (MANOVA), separates the overall mean gradient, the audio and video main effects, and their interaction. We partition each pretrained weight matrix into nonoverlapping square submatrices, called blocks, and use standardized cross-sample effect scores to *select* disjoint shared, audio, video, and interaction block sets within an adaptation budget. Only these blocks receive updates, while all other pretrained weights remain frozen.

During sequential fine-tuning, each task activates the block sets associated with its available modalities and *shares* the updates accumulated in those blocks. To limit interference when later tasks update the same blocks, we use auxiliary samples after each stage to measure block importance through mean squared sample-loss gradients, and *protect* the highest-ranked blocks covering a prescribed fraction of the learned task's total importance during subsequent learning of other tasks. Protected updates are frozen but remain in the forward pass, so their learned contributions persist, while other active blocks stay trainable. The block sets are selected once and reused across tasks. At inference, available modalities determine which updates apply, so no task identifier is needed. These updates can be merged into pretrained weights. Figure 2 summarizes the design.

Our main contributions are:

- **A unified design for selective sharing.** ASAP separates which parameters a task uses from which it may update. MANOVA-guided scores select shared, audio, video, and interaction blocks in pretrained weight matrices that tasks reuse according to their available modalities. Importance-based protection keeps active updates important to other tasks frozen but available in the forward pass, while other active updates remain trainable, without adding task-specific adapters.
- **Theoretical foundations for selective protection.** Under the stated assumptions, we bound each task's auxiliary-sample loss increase from its last learning stage to the final

stage in terms of unprotected importance (Theorem 1). Under fixed comparison conditions, importance ranking minimizes this bound among equally sized protection sets and achieves prescribed coverage with the fewest blocks per task (Corollary 1). Empirical diagnostics confirm that a small fraction of blocks captures most importance and that ASAP has smaller earlier-task test-loss increases than its unprotected variant (Figure 4).

- **Plasticity, stability and efficiency.** With Gemma-4-E2B under a matched adaptation budget, ASAP outperforms all compared baselines in current-task and final test accuracy across all twelve dataset–order combinations on MUSIC-AVQA and CMU-MOSI. Its six-order mean forgetting is lowest on MUSIC-AVQA and second-lowest on CMU-MOSI. With Qwen2.5-Omni-3B on CMU-MOSI, it also leads all three metrics and A/V/AV final accuracy, averaged over six orders. MUSIC-AVQA ablations show that selection improves new-task learning, trainable sharing improves new-task and final accuracy, and selective protection improves both retention and new-task learning. Efficiency comparisons show comparable peak GPU memory and forward-pass latency among compared methods.

2 Related Work

Continual multimodal learning. PathWeave adds adapters for new modalities (Yu et al., 2024), while MERA merges backbones and realigns modality-specific components (Zhang et al., 2025). MoInCL addresses changing modalities and task types through pseudo targets and distillation (Pian et al., 2026). D-MoLE allocates and routes LoRA experts (Ge et al., 2025), whereas PCLR compresses and integrates LoRA components (Meng et al., 2026). When reused modules are frozen, later tasks cannot directly refine them. ASAP instead shares selected pretrained-weight updates across modality combinations and selectively refines them, without adding task-specific adapters.

Sparse and modality-aware adaptation. LoRA learns low-rank updates (Hu et al., 2022), whereas SMT selects submatrices using warm-up gradients (He et al., 2025) and SPU selects task-relevant parameters to preserve pretrained knowledge (Zhang et al., 2024). MokA models modality-specific adaptation and cross-modal interactions (Wei et al., 2025). IMoD decomposes LoRA parameters into text-specific, non-text-specific, and shared components through gradient constraints (Zhang et al., 2026a). Unlike task-gradient ranking or adapter decomposition, ASAP uses factorial modality effects to assign pretrained-weight blocks to shared, audio, video, and interaction roles. This assignment determines which updates tasks reuse according to their available modalities.

Statistical criteria for block selection. High-dimensional mean tests use cross-sample inner products to estimate squared mean effects and covariance trace products to estimate their uncertainty (Chen & Qin, 2010; Cao et al., 2019). Building on these estimators, ASAP standardizes factorial gradient effects to rank blocks for disjoint, budgeted assignment rather than make significance decisions. The criterion therefore accounts for both effect magnitude and estimation uncertainty when allocating adaptation capacity.

Knowledge preservation during continual learning. EWC penalizes changes to important parameters (Kirkpatrick et al., 2017), while PackNet and HAT reserve task-specific subnetworks through pruning or learned masks (Mallya & Lazebnik, 2018; Serra et al., 2018). OSFT restricts update subspaces (Nayak et al., 2026), and DGG guides updates using approximated earlier-task gradients (Li et al., 2026). ASAP instead couples modality-based activation with coverage-controlled protection. Rather than freezing every block a task uses, it reserves the fewest blocks capturing a prescribed fraction of that task's importance. As in PackNet and HAT, protected contributions remain available for prediction, while unprotected shared blocks can be refined.

3 Method

3.1 Problem Setting and Overview

We adapt a pretrained MLLM to K sequential datasets $\mathcal{D}_1, \ldots, \mathcal{D}_K$, one per fine-tuning stage. Dataset $\mathcal{D}_t$ belongs to task $m_t \in \mathcal{M} := \{\mathrm{A}, \mathrm{V}, \mathrm{AV}\}$: audio, visual, or joint audio–visual understanding, with text always present. Task identity corresponds to the modality combination. The same task may recur with a different dataset. Each sample has an input x containing text and sensory inputs and a target answer-token sequence y.

A small labeled auxiliary dataset containing samples of the three tasks is available before sequential fine-tuning. ASAP uses it for selection before the first stage and importance estimation after each stage, without task-training updates or test evaluation on these samples.

Within a fixed adaptation budget, ASAP combines three components: **Select**, **Share**, and **Protect**. We select parameter blocks in pretrained weight matrices once, share their learned updates across tasks according to modality availability, and protect updates important to earlier tasks while leaving other active blocks trainable. Figure 2(a) illustrates block selection, while panel (b) shows sharing and selective protection. The following subsections describe these three components in turn.

3.2 MANOVA-Guided Block Selection

Before sequential fine-tuning, we compare loss gradients under controlled modality conditions to select shared and modality-associated blocks. We therefore omit the stage index t.

We use equal-sized blocks as units for selection, sharing, and protection, giving each block the same parameter cost. We partition a pretrained weight matrix W_0 into B nonoverlapping square blocks of size $d \times d$, indexed by $b \in \{1, \dots, B\}$. The side length d controls the granularity of these operations.

Let $\boldsymbol{\omega}$ denote the model parameters, with pretrained value $\boldsymbol{\omega}_0$. For input x and answer y, the mean answer-token loss is $\ell(x, y; \boldsymbol{\omega}) := -|y|^{-1} \log p_{\boldsymbol{\omega}}(y \mid x)$, where $|y|$ counts answer tokens. Let $G_b(x, y; \boldsymbol{\omega}) \in \mathbb{R}^{d \times d}$ denote the gradient of $\ell(x, y; \boldsymbol{\omega})$ with respect to block b of the weight matrix used in the forward pass.

Separate modality effects. To identify how modality presence changes a block's learning signal, we compare gradients at the same pretrained model while holding each auxiliary sample's text and answer fixed. For each of R auxiliary samples, let $x_r(v, a)$ retain the text and include or omit source video and audio according to $v, a \in \{0, 1\}$, respectively, where one indicates presence. Using the same answer y_r, we compute $G^{(r)}_{va,b} := G_b(x_r(v, a), y_r; \boldsymbol{\omega}_0)$ under all four conditions.

The MANOVA factorial contrasts give shared, audio, video, and interaction effect matrices $\Gamma^{(r)}_{e,b}$, indexed by $e \in \mathcal{E} := \{\mathrm{S}, \mathrm{A}, \mathrm{V}, \mathrm{I}\}$:

$$\begin{bmatrix} \Gamma^{(r)}_{\mathrm{S},b} \\ \Gamma^{(r)}_{\mathrm{A},b} \\ \Gamma^{(r)}_{\mathrm{V},b} \\ \Gamma^{(r)}_{\mathrm{I},b} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} G^{(r)}_{00,b} \\ G^{(r)}_{01,b} \\ G^{(r)}_{10,b} \\ G^{(r)}_{11,b} \end{bmatrix}. \tag{1}$$

The transformation applies entrywise, with inputs ordered row by row as in Figure 2(a). The four rows use mutually orthogonal sign patterns to compute the overall mean, the audio and video main effects, and their interaction, respectively. The effects reconstruct $G^{(r)}_{11,b} = \sum_{e \in \mathcal{E}} \Gamma^{(r)}_{e,b}$ (Appendix A.1) and guide the allocation of blocks to four sets.

Compute block scores. We seek blocks whose modality effects are consistent across auxiliary samples. Cross-sample inner products capture directional agreement between effects from different samples and estimate each effect's squared population-mean magnitude (Chen & Qin, 2010). We normalize this estimate by $\widehat{\sigma}_{e,b} > 0$, the estimated standard deviation of $\widehat{\theta}_{e,b}$ under the zero-mean hypothesis, to account for differences in estimation variability across blocks:

$$\widehat{\theta}_{e,b} := \frac{2}{R(R-1)} \sum_{1 \le r < r' \le R} \langle \Gamma^{(r)}_{e,b}, \Gamma^{(r')}_{e,b} \rangle_F, \qquad \widehat{T}_{e,b} := \widehat{\theta}_{e,b} / \widehat{\sigma}_{e,b}. \tag{2}$$

Here $\langle \cdot, \cdot \rangle_F$ and $\| \cdot \|_F$ are the Frobenius inner product and norm, and r, r' index distinct samples. Larger scores receive higher priority in block assignment. Under independent sampling with a common mean and finite second moments, $\mathbb{E}[\widehat{\theta}_{e,b}] = \|\mathbb{E}[\Gamma^{(r)}_{e,b}]\|^2_F$, so cross-sample products remove the single-sample variance contribution. When selection samples are stratified by task and task means differ, Equation 18 gives the corresponding expectation. Appendices A.2.1–A.2.4 give the variance estimator (Cao et al., 2019) and the agreement and consistency results.

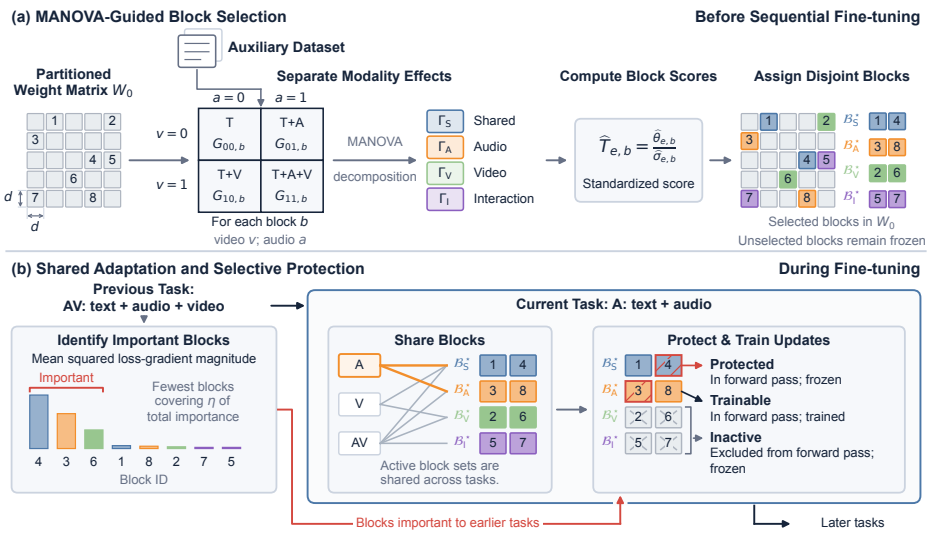


Figure 2: **Overview of ASAP.** (a) For each $d \times d$ block b of a pretrained matrix W_0, loss gradients $G_{va,b}$ on auxiliary samples, with video (v) and audio (a) present or absent, are decomposed into effects Γ_e. Scores $\widehat{T}_{e,b}$ divide each estimated squared mean effect $\widehat{\theta}_{e,b}$ by its standard deviation under the zero-mean hypothesis, $\widehat{\sigma}_{e,b}$, and determine disjoint block sets $\mathcal{B}_e^\star$. (b) A task's modalities determine its active block sets. Each learned task's important blocks, the fewest covering a fraction η of its importance, are protected wherever other tasks activate them.

Assign disjoint blocks. A block may score highly for several effects. To resolve this competition, we jointly assign blocks within each weight matrix under a per-matrix budget of N parameters:

$$\max_{\{\mathcal{B}_e\}_{e\in\mathcal{E}}} \sum_{e\in\mathcal{E}} \sum_{b\in\mathcal{B}_e} \widehat{T}_{e,b} \quad \text{s.t.} \quad \mathcal{B}_e \subseteq \{1,\dots,B\}, \quad |\mathcal{B}_e| = \tfrac{N}{4d^2}, \quad \mathcal{B}_e \cap \mathcal{B}_{e'} = \varnothing \quad (e \neq e'). \tag{3}$$

Equal quotas reserve capacity for each effect, while disjointness assigns each selected block one modality-based activation rule. After this matrixwise assignment, we rank assigned blocks across matrices within each effect and trim them to the final model-wide budget, producing the selected block sets $\mathcal{B}_e^\star$. These sets remain fixed across stages while their learned updates evolve. See Appendix A.3 for more details.

3.3 Shared Adaptation

The selected sets fix where updates are learned, while modality availability determines which updates each task uses, as illustrated by the "Share Blocks" step in Figure 2(b). For task $m \in \mathcal{M}$, let $a_m = 1$ for $m \in \{\mathrm{A}, \mathrm{AV}\}$ and $v_m = 1$ for $m \in \{\mathrm{V}, \mathrm{AV}\}$, each zero otherwise. Let $\Delta_{e,t}$ have W_0's shape, with updates accumulated through stage t at blocks in $\mathcal{B}_e^\star$, zeros elsewhere, and $\Delta_{e,0} = 0$. Task m uses the weight matrix

$$W_t(v_m, a_m) = W_0 + \Delta_{\mathrm{S},t} + a_m\Delta_{\mathrm{A},t} + v_m\Delta_{\mathrm{V},t} + v_m a_m \Delta_{\mathrm{I},t}. \tag{4}$$

Shared updates are always active. Audio and video updates are active when the respective modality is present, while interaction updates require both. The active block sets of the three tasks are

$$\mathcal{C}_{\mathrm{A}} = \mathcal{B}_{\mathrm{S}}^\star \cup \mathcal{B}_{\mathrm{A}}^\star, \quad \mathcal{C}_{\mathrm{V}} = \mathcal{B}_{\mathrm{S}}^\star \cup \mathcal{B}_{\mathrm{V}}^\star, \quad \mathcal{C}_{\mathrm{AV}} = \mathcal{B}_{\mathrm{S}}^\star \cup \mathcal{B}_{\mathrm{A}}^\star \cup \mathcal{B}_{\mathrm{V}}^\star \cup \mathcal{B}_{\mathrm{I}}^\star. \tag{5}$$

Tasks with overlapping active sets reuse the same learned updates. For example, A and AV reuse the shared and audio updates, allowing learning on either task to refine parameters used by the other. Inactive updates are excluded from the forward pass and frozen. Unselected parameters also remain fixed. At evaluation, updates corresponding to the available modalities are merged into W_0 as in Equation 4 to avoid extra forward-pass latency.

3.4 Selective Protection

Later tasks can modify blocks that earlier tasks also use. We protect updates important to earlier tasks while retaining their contribution to prediction and leaving other active blocks trainable.

Measure importance after learning. Selection identifies coherent mean effects before learning, whereas the "Identify Important Blocks" step in Figure 2(b) measures samplewise loss sensitivity after learning. Averaging squared per-sample gradient norms retains sensitivity even when gradients cancel in the mean. At the parameters $\boldsymbol{\omega}_t$ after stage t, we score each active block using the R_{m_t} auxiliary samples of task m_t:

$$s_{b,t} := \frac{1}{R_{m_t}} \sum_{r=1}^{R_{m_t}} \|G_b(x_r(v_{m_t}, a_{m_t}), y_r; \boldsymbol{\omega}_t)\|_F^2 , \qquad b \in \mathcal{C}_{m_t}. \tag{6}$$

Larger scores indicate greater samplewise loss sensitivity to block changes (Appendix B.2).

Select blocks by importance coverage. We choose how many blocks to protect by the fraction of total importance they cover. This lets each task reserve fewer blocks when its importance scores are concentrated. Task m stores its important-block set $\mathcal{F}_m^{(t)} \subseteq \mathcal{C}_m$ as a record after stage t, with $\mathcal{F}_m^{(0)} = \varnothing$. After learning m_t, we rank its active blocks by $s_{b_{(1)},t} \geq s_{b_{(2)},t} \geq \cdots$, where $b_{(i)}$ is the i-th ranked block and ties are resolved consistently. We replace its record with the fewest blocks covering a prescribed fraction $\eta \in (0, 1]$ of total importance:

$$\mathcal{F}_{m_t}^{(t)} = \{b_{(i)} : 1 \leq i \leq k^\star\}, \quad k^\star = \min\Big\{k \in \{0, \ldots, |\mathcal{C}_{m_t}|\} : \sum_{i=1}^{k} s_{b_{(i)},t} \geq \eta \sum_{b \in \mathcal{C}_{m_t}} s_{b,t}\Big\}. \tag{7}$$

Other tasks retain their records: $\mathcal{F}_m^{(t)} = \mathcal{F}_m^{(t-1)}$ for $m \neq m_t$. Larger η requires greater coverage and may leave fewer active blocks trainable later. For a fixed set size, importance ranking maximizes protected importance and hence minimizes the bound under Corollary 1's conditions.

Protect overlaps during subsequent learning. When dataset $\mathcal{D}_{t+1}$ arrives after stage $t < K$, we protect its active blocks that appear in any other task's importance record $\mathcal{F}_m^{(t)}$. The protected block set is

$$\mathcal{P}_{t+1} = \mathcal{C}_{m_{t+1}} \cap \bigcup_{m \in \mathcal{M} \setminus \{m_{t+1}\}} \mathcal{F}_m^{(t)}. \tag{8}$$

Initially, $\mathcal{P}_1 = \varnothing$. At stage $t+1$, only blocks in $\mathcal{W}_{t+1} := \mathcal{C}_{m_{t+1}} \setminus \mathcal{P}_{t+1}$ are trained. Protected updates remain in the forward pass but stay frozen, as illustrated in the "Protect & Train Updates" box of Figure 2(b). Excluding the current task's own record lets a recurring task refine its updates: on a later audio dataset, blocks important only to A become trainable, while active blocks important to V or AV remain protected. After training, we refresh the current task's record to reflect the updated model and retain the others. Appendix B.3 gives the full training procedure.

4 Theoretical Results

Protecting important blocks restricts where later updates can affect an earlier task. We bound each task's loss increase from its last learning or relearning stage t' to the final stage K. Let $m = m_{t'}$ and define $\mathcal{L}_m(\boldsymbol{\omega}) := R_m^{-1} \sum_{r=1}^{R_m} \ell(x_r(v_m, a_m), y_r; \boldsymbol{\omega})$ over the R_m auxiliary samples of task m used to measure importance after stage t', where ℓ is the per-sample loss defined in Section 3.2.

Theorem 1 (Final loss stability)**.** *Only learned updates in $\mathcal{C}_m$ enter this task's forward pass (Equations 4–5). Let $\mathcal{F} \subseteq \mathcal{C}_m$ be a set whose learned updates remain fixed after stage t' through stage K. Assume that $\mathcal{L}_m$ has an L-Lipschitz gradient along the segment from $\boldsymbol{\omega}_{t'}$ to $\boldsymbol{\omega}_K$, and that $\sum_{b \in \mathcal{C}_m} \|\Delta_{b,t} - \Delta_{b,t-1}\|_F^2 \leq \rho^2$ for $t = t' + 1, \ldots, K$, where $\Delta_{b,t} \in \mathbb{R}^{d \times d}$ denotes block b's accumulated update after stage t, and $\rho \geq 0$ bounds each stage's change.*

$$\mathcal{L}_m(\boldsymbol{\omega}_K) - \mathcal{L}_m(\boldsymbol{\omega}_{t'}) \leq (K - t')\rho \sqrt{\underbrace{\sum_{b \in \mathcal{C}_m} s_{b,t'}}_{\textit{total importance}} - \underbrace{\sum_{b \in \mathcal{F}} s_{b,t'}}_{\textit{protected importance}}} + \frac{L}{2}(K - t')^2 \rho^2. \tag{9}$$

The first term captures the contribution of unprotected importance, while the second accounts for loss curvature. Reducing the unprotected importance under the square root tightens the bound when other quantities are fixed. After task m's last learning stage t', updates in $\mathcal{F}_m^{(t')}$ remain protected or inactive under Equation 8, so Theorem 1 applies to $\mathcal{F}_m^{(t')}$. The next result connects this bound to the importance-ranking rule in Section 3.4:

Table 1: **Continual learning performance on MUSIC-AVQA and CMU-MOSI,** averaged over six orders per dataset. A, V, and AV report final accuracy. Final.ACC is their mean. Accuracy is in percent; forgetting is in percentage points. Bold and underline indicate best and second best.

	MUSIC-AVQA						CMU-MOSI					
Method	A	V	AV	Final.ACC↑	Avg.FOR↓	New.ACC↑	A	V	AV	Final.ACC↑	Avg.FOR↓	New.ACC↑
LoRA	57.38	74.47	61.90	64.59	5.12	<u>68.61</u>	81.50	79.80	78.22	79.84	0.42	79.93
PCLR	60.52	77.84	<u>64.03</u>	<u>67.46</u>	2.27	68.05	81.72	79.87	78.75	80.11	0.36	80.29
DGG	59.49	<u>77.97</u>	63.61	67.02	<u>2.17</u>	67.38	82.06	<u>80.24</u>	<u>78.83</u>	<u>80.37</u>	0.48	<u>80.39</u>
SPU	54.28	60.68	50.49	55.15	16.59	63.94	75.28	73.36	72.92	73.86	7.78	74.83
OSFT	<u>62.15</u>	76.51	63.31	67.32	2.18	68.22	<u>82.33</u>	79.87	78.51	80.24	**0.17**	79.68
ASAP	**63.14**	**79.51**	**65.73**	**69.46**	**1.07**	**69.47**	**83.99**	**82.97**	**81.49**	**82.82**	<u>0.25</u>	**82.72**

Corollary 1 (Optimality of importance-ranked protection)**.** *Under the assumptions of Theorem 1, with the importance scores, $K - t'$, ρ, and L fixed, ASAP's set $\mathcal{F}_m^{(t')}$ minimizes the bound in Equation 9 among equally sized protection subsets of $\mathcal{C}_m$ whose updates remain fixed from stage t' through stage K. It also achieves coverage η with the fewest blocks, leaving at most a fraction $1 - \eta$ of total importance unprotected.*

Remarks. Theorem 1 bounds earlier-task loss increases, supporting **memory stability**. Corollary 1 establishes minimum-size protection for the required importance coverage. When importance is concentrated, this protected set is small, leaving capacity for **learning plasticity**. Appendix C.1 provides proofs for both results. Section 5.4 shows that a small block subset captures most importance and that ASAP has smaller earlier-task test-loss increases than its unprotected variant. Appendices C.2–C.3 analyze current-task descent and the effects of shared updates on other tasks.

5 Experiments

We evaluate ASAP on MUSIC-AVQA question answering and CMU-MOSI sentiment classification under six task orders per dataset. We compare new-task learning, forgetting, and final performance against continual fine-tuning baselines, then examine selection, sharing, and protection through MUSIC-AVQA ablations. Efficiency comparisons assess peak GPU memory and forward-pass latency, and importance and loss analyses examine the protection mechanism characterized by Theorem 1 and Corollary 1. Experiments use NVIDIA H100, A100, and RTX 5090 GPUs.

5.1 Experimental Setup

Datasets. Both datasets provide A, V, and AV tasks. For MUSIC-AVQA (Li et al., 2022), original question-type annotations (Audio, Visual, Audio-Visual) define the three groups of question–answer pairs and their sensory inputs. For CMU-MOSI (Zadeh et al., 2016), we partition training clips into three groups and supply audio, video, or both alongside transcripts for sentiment classification, with Positive, Neutral, or Negative as the answer. Both datasets use answer-token loss and exact-match evaluation. Each task's training data form two subsets, $\mathrm{A}_1, \mathrm{A}_2$, $\mathrm{V}_1, \mathrm{V}_2$, and $\mathrm{AV}_1, \mathrm{AV}_2$, where subscripts distinguish datasets for the same task, not training stages. The six subsets share no source videos or audio recordings with one another or the test set.

Task sequences. For each dataset, we evaluate continual learning across A, V, and AV using six five-stage paths, shown on the x-axes of Figure 3. Each path begins with the union of one task's two subsets, followed by the four subsets of the other two tasks. The starting task varies across pairs. Within each pair, the remaining tasks alternate in opposite orders. Appendix D.1 gives details.

Baselines and implementation. We compare LoRA (Hu et al., 2022), PCLR (Meng et al., 2026), DGG (Li et al., 2026), SPU (Zhang et al., 2024), and OSFT (Nayak et al., 2026) using Gemma-4-E2B (Gemma Team, 2026). We retain each method's official implementation with protocol adaptations. All methods share task orders, evaluation schedules, and optimizer settings, with three epochs per stage and a total adaptation-parameter budget matched to rank-8 LoRA (approximately 12.1M). For each benchmark, ASAP reserves approximately 1% of the original training samples for block selection and importance estimation, without task-training updates. Baselines include these samples

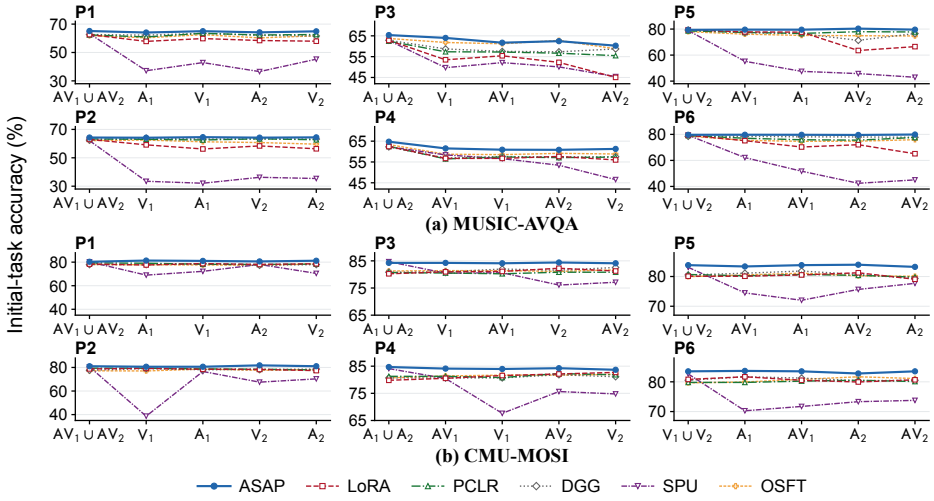


Figure 3: **Initial-task retention across six orders on each dataset.** (a) MUSIC-AVQA; (b) CMU-MOSI. Each point shows initial-task accuracy after training on the dataset labeled on the x-axis.

in training. All methods exclude them from test evaluation. Appendix D.2 details method-specific adaptations, target layers, and optimization settings.

Metrics. Let $A_{p,t,m}$ denote task m's exact-match test accuracy after stage t of path p, and $m_{p,t}$ the task trained at that stage. *Learning plasticity* is measured by $\mathrm{New.ACC}_p = K^{-1}\sum_{t=1}^{K} A_{p,t,m_{p,t}}$, with $K = 5$ (Meng et al., 2026). For *memory stability*, Avg.FOR averages peak-to-current accuracy drops over previously learned tasks except the current task, then over stages 2 through K, including forgetting before relearning. Final performance is measured by $\mathrm{Final.ACC}_p = \frac{1}{3}\sum_{m\in\mathcal{M}} A_{p,K,m}$, averaging final test accuracy equally across the three tasks. These metrics are averaged over six paths. We also report peak allocated GPU memory and forward-pass latency for computational efficiency comparisons. Appendix D.3 gives full definitions.

5.2 Continual Learning Performance

Plasticity and stability together. Under the same parameter budget, ASAP leads New.ACC, Final.ACC, and A/V/AV final accuracy on both datasets (Table 1). Its per-task accuracy gains over the strongest baselines are 0.99–1.70 points on MUSIC-AVQA and 1.66–2.73 on CMU-MOSI. It also achieves the lowest MUSIC-AVQA Avg.FOR and the second-lowest on CMU-MOSI. This advantage holds across learning orders: New.ACC and Final.ACC lead on all 12 path–dataset combinations, and Avg.FOR is lowest on all six MUSIC-AVQA paths (Appendix D.4).

Plasticity–stability trade-offs. On MUSIC-AVQA, without protection, LoRA ranks second in New.ACC but incurs 5.12 Avg.FOR. PCLR, DGG, and OSFT have less than half of its forgetting but lower New.ACC. On CMU-MOSI, OSFT's subspace-constrained adaptation achieves the lowest Avg.FOR (0.17), but lower New.ACC than LoRA, PCLR, and DGG. Nevertheless, ASAP achieves higher New.ACC and lower Avg.FOR than OSFT on MUSIC-AVQA. On CMU-MOSI, it exceeds OSFT's New.ACC by 3.04 points with slightly higher Avg.FOR (0.08). These results support the effectiveness of ASAP's selection, sharing, and protection in balancing plasticity and stability.

Retention across stages. Figure 3 shows ASAP generally retaining higher initial-task accuracy without relearning. Appendix D.4 quantifies its stability: the maximum within-path range on AV- and V-first MUSIC-AVQA paths is 1.06 percentage points versus 3.55 for PCLR, and the maximum single-stage drop across six paths is 3.23 versus 4.66 for OSFT, the respective best baselines. On CMU-MOSI, ASAP leads subsequent stages, while most baselines forget little, possibly due to the shared sentiment objective and consistently available transcripts.

A second backbone. With Qwen2.5-Omni-3B (Xu et al., 2025) on CMU-MOSI, ASAP achieves the highest mean New.ACC and Final.ACC and the lowest Avg.FOR. It leads all 30 initial-task evaluations and has the smallest maximum single-stage drop. Appendix D.5 provides detailed results.

Table 2: **Component ablations.** MUSIC-AVQA six-path means, with units as in Table 1.

Variant	Final.ACC↑	Avg.FOR↓	New.ACC↑
Without protection	66.45	2.69	68.24
Random selection	67.08	1.51	67.29
Frozen shared blocks	66.71	**0.84**	66.35
ASAP	**69.46**	1.07	**69.47**

Table 3: **Efficiency comparison:** peak GPU memory and forward-pass latency.

Method	Params.	Latency↓ (ms)	Memory↓ (GiB)
LoRA	12.1M	127.1	11.10
PCLR	12.1M	164.2	11.90
DGG	12.1M	126.4	11.10
ASAP	12.1M	126.4	11.10

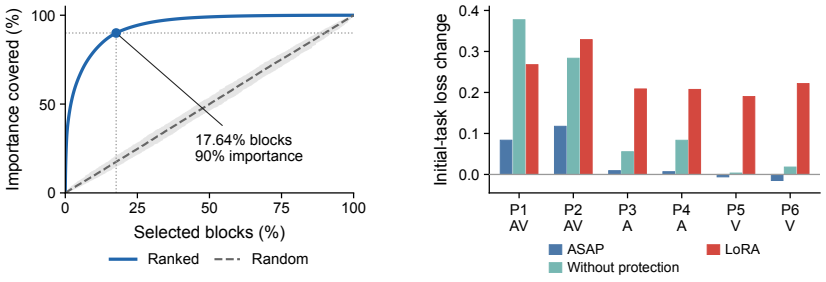
5.3 Ablations and Efficiency Comparison

ASAP component ablations. Table 2 varies one component at a time (Appendix D.6). Replacing MANOVA-guided selection with random blocks lowers New.ACC by 2.18 points, supporting effect-based selection. Freezing shared updates after the first stage prevents refinement and lowers New.ACC by 3.12 points despite reducing forgetting, demonstrating the benefit of trainable sharing. Removing protection leaves all active updates trainable, increasing Avg.FOR by 1.62 points and lowering New.ACC on all six paths, supporting protection for both stability and plasticity. All variants lower Final.ACC by 2.38–3.01 points, supporting the combined design of ASAP.

Efficiency comparison. Table 3 compares methods on MUSIC-AVQA at a matched budget on an H100 80GB GPU. Merged ASAP, LoRA, and DGG have comparable forward-pass latency (126.4–127.1 ms) and peak GPU memory, while native PCLR takes 164.2 ms (Appendix D.7).

5.4 Analysis of Selective Protection

On MUSIC-AVQA with Gemma-4-E2B, we examine retention and trainable capacity. After initial AV training on P1, the highest-ranked 17.64% of active blocks capture 90% of total importance, whereas random ordering requires 90% of blocks for the same expected coverage (Figure 4(a)), supporting Corollary 1's minimum-count coverage rule. Across subsequent stages, 76.7–94.8% of active blocks remain trainable (Figure 6, Appendix D.8). ASAP also has smaller initial-task acquisition-to-final test-loss changes than both comparison methods on all six paths, averaging 0.033 (Figure 4(b)). Together, these results show smaller initial-task test-loss changes while most active blocks remain trainable.



(a) Importance coverage (P1) **(b) Initial-task loss change (six paths)**

Figure 4: **Selective protection and loss retention on MUSIC-AVQA.** (a) P1 coverage, with random selection's 5th–95th percentiles shaded. (b) Final minus acquisition loss on the initial task.

6 Conclusion

ASAP combines MANOVA-guided selection, modality-based sharing, and selective protection for continual multimodal learning. Our analysis bounds earlier-task loss increases in terms of unprotected importance. Across six orders per dataset, it leads average current-task and final accuracy on MUSIC-AVQA and CMU-MOSI, with strong retention and comparable memory and latency. Ablations link selection and sharing to learning gains, and protection to learning and retention. These results support separating which updates tasks use from which they modify.

Reproducibility Statement

The main paper specifies the method equations, invariance conditions, evaluation metrics, and matching criteria for all comparisons. Appendix D documents data-split construction, adaptation-budget and optimization matching, and timing boundaries. The artifact will include the split manifest, task-sequence definitions, block assignments, model and training settings, and predictions after each training stage used to reproduce the tables. The implementation of ASAP is available at `https://anonymous.4open.science/r/ASAP-ICLR/`.

Ethics Statement

This work studies parameter-efficient adaptation of existing multimodal models and introduces no new data-collection protocol.

Use of Large Language Models

Generative AI tools assisted with writing only.

References

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173. URL `https://leon.bottou.org/publications/pdf/sirev-2018.pdf`.

Ming-Xiang Cao, Junyong Park, and Dao-Jiang He. A test for the k sample Behrens–Fisher problem in high dimensional data. *Journal of Statistical Planning and Inference*, 201:86–102, 2019. doi: 10.1016/j.jspi.2018.12.002. URL `https://arxiv.org/abs/1710.07878`.

Song Xi Chen and Ying-Li Qin. A two-sample test for high-dimensional data with applications to gene-set testing. *The Annals of Statistics*, 38(2):808–835, 2010. doi: 10.1214/09-AOS716. URL `https://arxiv.org/abs/1002.4547`.

Jianqing Fan and Yingying Fan. High-dimensional classification using features annealed independence rules. *The Annals of Statistics*, 36(6):2605–2637, 2008. doi: 10.1214/07-AOS504. URL `https://arxiv.org/abs/math/0701108`.

Chendi Ge, Xin Wang, Zeyang Zhang, Hong Chen, Jiapei Fan, Longtao Huang, Hui Xue, and Wenwu Zhu. Dynamic mixture of curriculum LoRA experts for continual multimodal instruction tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19011–19033, 2025.

Gemma Team. Gemma 4 technical report, 2026. URL `https://arxiv.org/abs/2607.02770`.

Zirun Guo, Shulei Wang, Wang Lin, Weicai Yan, Yangyang Wu, and Tao Jin. Efficient prompting for continual adaptation to missing modalities. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4317–4327, 2025. doi: 10.18653/v1/2025.naacl-long.219.

Haoze He, Juncheng B. Li, Xuan Jiang, and Heather Miller. SMT: Fine-tuning large language models with sparse matrices. In *International Conference on Learning Representations*, pp. 32369–32393, 2025. URL `https://proceedings.iclr.cc/paper_files/paper/2025/hash/5054c447afc6b0b855d6b0b2aad104d7-Abstract-Conference.html`.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.02340.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6265–6274, 2021. URL https://proceedings.mlr.press/v139/lewis21a.html.

Dongfang Li, Zetian Sun, Xinshuo Hu, Baotian Hu, and Min Zhang. CMT: A memory compression method for continual knowledge learning of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24413–24421, 2025. doi: 10.1609/aaai.v39i23.34619. URL https://ojs.aaai.org/index.php/AAAI/article/view/34619.

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19108–19118, 2022.

Songze Li, Mingyu Gao, Tonghua Su, Xu-Yao Zhang, and Zhongjie Wang. Multimodal continual instruction tuning with dynamic gradient guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Li_Multimodal_Continual_Instruction_Tuning_with_Dynamic_Gradient_Guidance_CVPR_2026_paper.pdf.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.342. URL https://aclanthology.org/2024.emnlp-main.342/.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Weicheng Meng, Jingyang Qiao, Zhizhong Zhang, Shaohui Liu, and Yuan Xie. PCLR: Progressively compressed LoRA for multimodal continual instruction tuning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5a5acfd0876c940d81619c1dc60e7748-Abstract-Conference.html.

Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 10 edition, 2019. ISBN 9781119492443.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Oleg Silkin, Mustafa Eyceoz, Hao Wang, Aldo Pareja, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/22620cccae29bbb3f18d226ea0320ff5-Abstract-Conference.html.

Weiguo Pian, Shijian Deng, Shentong Mo, Mingrui Liu, Yunhui Guo, and Yapeng Tian. Modality-inconsistent continual learning of multimodal large language models. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=FD8or43fBU.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4548–4557, 2018.

Jihoon Tack, Jaehyung Kim, Eric Mitchell, Jinwoo Shin, Yee Whye Teh, and Jonathan Richard Schwarz. Online adaptation of language models with a memory of amortized contexts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 130109–130135, 2024. doi: 10.52202/079017-4134. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/eaf956b52bae51fbf387b8be4cc3ce18-Paper-Conference.pdf.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards generic hearing abilities for large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/476ab8f369e489c04187ba84f68cfa68-Abstract-Conference.html.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. doi: 10.1109/TPAMI.2024.3367329.

Yake Wei, Yu Miao, Dongzhan Zhou, and Di Hu. MokA: Multimodal low-rank adaptation for MLLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Kaixuan Wu, Xinde Li, Xinling Li, Chuanfei Hu, and Guoliang Wu. AVQACL: A novel benchmark for audio-visual question answering continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3252–3261, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wu_AVQACL_A_Novel_Benchmark_for_Audio-Visual_Question_Answering_Continual_Learning_CVPR_2025_paper.html.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni technical report. *arXiv preprint arXiv:2503.20215*, 2025. URL https://arxiv.org/abs/2503.20215.

Jiazuo Yu, Haomiao Xiong, Lu Zhang, Haiwen Diao, Yunzhi Zhuge, Lanqing Hong, Dong Wang, Huchuan Lu, You He, and Long Chen. LLMs can evolve continually on modality for x-modal reasoning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. MOSI: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016. URL https://arxiv.org/abs/1606.06259.

Dingkun Zhang, Shuhan Qi, Xinyu Xiao, Kehai Chen, and Xuan Wang. Merge then realign: Simple and effective modality-incremental continual learning for multimodal LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13148–13164, 2025. doi: 10.18653/v1/2025.emnlp-main.665.

Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 543–553. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-demo.49. URL https://aclanthology.org/2023.emnlp-demo.49/.

Mingfang Zhang, Yunhong Wang, Lu Wang, and Jiaxin Chen. Parameter-efficient adaptation for MLLMs via implicit modality decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37745–37755, 2026a.

Mingxu Zhang, Yuhan Li, Lujundong Li, Dazhong Shen, Hui Xiong, and Ying Sun. SAE-FD: Sparse autoencoder feature distillation for continual learning of large language models. *arXiv preprint arXiv:2605.25525*, 2026b. URL https://arxiv.org/abs/2605.25525.

Wenxuan Zhang, Paul Janson, Rahaf Aljundi, and Mohamed Elhoseiny. Overcoming generic knowledge loss with selective parameter update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24046–24056, 2024.

APPENDIX

A STATISTICAL AND COMPUTATIONAL DETAILS OF BLOCK SELECTION

A.1 MANOVA DECOMPOSITION OF MODALITY EFFECTS

Controlled gradients. Use $G_b(x, y; \boldsymbol{\omega})$ from Section 3.2. The block count B and parameter budget N there refer to the current weight matrix. When considering blocks across matrices, each block index also identifies its containing matrix. At the pretrained model $\boldsymbol{\omega}_0$, the controlled gradient is

$$G^{(r)}_{va,b} := G_b(x_r(v, a), y_r; \boldsymbol{\omega}_0). \tag{10}$$

The subscript 0 denotes the pretrained state. W_0 is one pretrained matrix in this model, whereas $W_t(v_m, a_m)$ includes the updates accumulated through stage t and activated for task m. For $b \in \mathcal{B}^\star_e$, let Δ_b denote the learned additive block update. Its value after stage t is $\Delta_{b,t} = (\Delta_{e,t})_b$, where $(\Delta_{e,t})_b$ denotes block b of the update matrix in Section 3.3. For an active block, the loss gradient with respect to Δ_b equals $G_b(x, y; \boldsymbol{\omega})$. Each sample supplies aligned audio and video. All four conditions preserve its text, target, and pretrained model. An off modality is omitted from the processor input, so token lengths and attention masks can differ across conditions. The contrasts therefore capture the gradient effect of the full input-presence intervention, including its change to the input sequence. Suppressing (r, b), this defines the table $G : \{0, 1\}^2 \to \mathbb{R}^{d\times d}$ with $G(v, a) := G_{va}$.

Factorial decomposition. For a fixed sample and block, treat video presence v and audio presence a as two binary factors with equal weight on their four combinations. The cells $G_{00}, G_{01}, G_{10}, G_{11}$ correspond to text alone, text with audio, text with video, and text with both modalities, respectively. The decomposition applies entrywise to these gradient matrices. Define the marginal and overall means:

$$G_{v\cdot} := \frac{1}{2}\sum_{a\in\{0,1\}} G_{va}, \qquad G_{\cdot a} := \frac{1}{2}\sum_{v\in\{0,1\}} G_{va}, \qquad G_{\cdot\cdot} := \frac{1}{4}\sum_{v,a\in\{0,1\}} G_{va}. \tag{11}$$

A dot denotes averaging over that variable: $G_{v\cdot}$ averages audio states, $G_{\cdot a}$ averages video states, and $G_{\cdot\cdot}$ averages both. The overall mean, two main effects, and interaction are

$$\begin{aligned}
\Gamma_{\mathrm{S}} &:= G_{\cdot\cdot},\\
\Gamma_{\mathrm{A}}(a) &:= G_{\cdot a} - G_{\cdot\cdot},\\
\Gamma_{\mathrm{V}}(v) &:= G_{v\cdot} - G_{\cdot\cdot},\\
\Gamma_{\mathrm{I}}(v, a) &:= G_{va} - G_{v\cdot} - G_{\cdot a} + G_{\cdot\cdot}.
\end{aligned} \tag{12}$$

Thus $G_{va} = \Gamma_{\mathrm{S}} + \Gamma_{\mathrm{A}}(a) + \Gamma_{\mathrm{V}}(v) + \Gamma_{\mathrm{I}}(v, a)$. The audio and video main effects subtract the overall mean from their respective marginal means. The interaction is the residual after removing the overall mean and both main effects. These components satisfy the centering constraints

$$\sum_{a=0}^{1}\Gamma_{\mathrm{A}}(a) = 0, \quad \sum_{v=0}^{1}\Gamma_{\mathrm{V}}(v) = 0, \quad \sum_{v=0}^{1}\Gamma_{\mathrm{I}}(v, a) = 0, \quad \sum_{a=0}^{1}\Gamma_{\mathrm{I}}(v, a) = 0. \tag{13}$$

The last two equalities hold for each fixed a and v, respectively. Together with reconstruction, these constraints uniquely determine the balanced two-factor MANOVA decomposition (Montgomery, 2019).

An effect with no modality argument denotes its value when the corresponding modality is present. Thus the block-selection effects are:

$$\begin{aligned}
\Gamma_{\mathrm{S}} &= G_{\cdot\cdot},\\
\Gamma_{\mathrm{A}} &:= \Gamma_{\mathrm{A}}(1) = G_{\cdot 1} - G_{\cdot\cdot},\\
\Gamma_{\mathrm{V}} &:= \Gamma_{\mathrm{V}}(1) = G_{1\cdot} - G_{\cdot\cdot},\\
\Gamma_{\mathrm{I}} &:= \Gamma_{\mathrm{I}}(1, 1) = G_{11} - G_{1\cdot} - G_{\cdot 1} + G_{\cdot\cdot}.
\end{aligned} \tag{14}$$

These are the four components in Equation 1. Substituting the means from Equation 11 gives their explicit forms:

$$\begin{aligned}
\Gamma_{\mathrm{S}} &= \tfrac{1}{4}(G_{00} + G_{10} + G_{01} + G_{11}), \qquad \Gamma_{\mathrm{A}} = \tfrac{1}{4}(-G_{00} - G_{10} + G_{01} + G_{11}),\\
\Gamma_{\mathrm{V}} &= \tfrac{1}{4}(-G_{00} + G_{10} - G_{01} + G_{11}), \quad \Gamma_{\mathrm{I}} = \tfrac{1}{4}(G_{00} - G_{10} - G_{01} + G_{11}).
\end{aligned}$$

In particular, $2\Gamma_{\mathrm{A}} = G_{\cdot 1} - G_{\cdot 0}$ and $2\Gamma_{\mathrm{V}} = G_{1\cdot} - G_{0\cdot}$ are the marginal on–off differences, while $4\Gamma_{\mathrm{I}} = (G_{11} - G_{10}) - (G_{01} - G_{00})$ is the change in the audio on–off difference when video is added. This explains the normalization by four in the explicit contrasts. As text is present in every condition, the shared component averages all text-containing conditions rather than isolating a text effect.

For any of the four inputs, reconstruction is

$$G_{va} = \Gamma_{\mathrm{S}} + (2a-1)\Gamma_{\mathrm{A}} + (2v-1)\Gamma_{\mathrm{V}} + (2v-1)(2a-1)\Gamma_{\mathrm{I}}. \tag{15}$$

The four sign functions $1, 2a-1, 2v-1, (2v-1)(2a-1)$ are orthogonal under uniform averaging over the four conditions. Consequently, $\frac{1}{4}\sum_{v=0}^{1}\sum_{a=0}^{1}\|G_{va}\|_F^2 = \sum_e \|\Gamma_e\|_F^2$. The contrast functions are orthogonal over the four input conditions, but the gradient matrices Γ_e need not be orthogonal in parameter space. At the complete input, $G_{11} = \Gamma_{\mathrm{S}} + \Gamma_{\mathrm{A}} + \Gamma_{\mathrm{V}} + \Gamma_{\mathrm{I}}$. These gradient effects guide block selection. Sequential fine-tuning learns the update matrices $\Delta_{e,t}$ using the modality-based activation rule in Equation 4.

The MANOVA contrasts above are linear identities and apply entrywise to matrix-valued responses. Vectorization preserves those contrasts and the sums of products of corresponding entries. Estimating each effect's squared population mean is a multivariate problem, addressed by the cross-sample construction of Chen & Qin (2010). For MUSIC-AVQA, a block has $32^2 = 1{,}024$ coordinates and $R = 51$ selection samples, so its sample covariance has rank at most 50. The trace-product construction avoids inversion of that singular covariance. Each sample supplies four paired input conditions, which together yield one observation of each effect.

Restoring the sample index r and block index b, the joint-input reconstruction implied by Equation 1 is

$$G_{11,b}^{(r)} = \underbrace{G_{\cdot\cdot,b}^{(r)}}_{\Gamma_{\mathrm{S},b}^{(r)}} + \underbrace{G_{\cdot 1,b}^{(r)} - G_{\cdot\cdot,b}^{(r)}}_{\Gamma_{\mathrm{A},b}^{(r)}} + \underbrace{G_{1\cdot,b}^{(r)} - G_{\cdot\cdot,b}^{(r)}}_{\Gamma_{\mathrm{V},b}^{(r)}} + \underbrace{G_{11,b}^{(r)} - G_{\cdot 1,b}^{(r)} - G_{1\cdot,b}^{(r)} + G_{\cdot\cdot,b}^{(r)}}_{\Gamma_{\mathrm{I},b}^{(r)}}.$$

A.2 Block-score computation

A.2.1 Sampling assumptions and unbiased effect estimation

Sampling model. For covariance calculations, write $\boldsymbol{\gamma}_{e,b}^{(r)} := \mathrm{vec}(\Gamma_{e,b}^{(r)}) \in \mathbb{R}^{d^2}$ for the same matrix effect with its columns stacked into a vector. Coordinate $c = i + (j-1)d$ is entry (i,j) of the matrix. Thus $\sum_{c=1}^{d^2}(\boldsymbol{\gamma}_{e,b}^{(r)})_c(\boldsymbol{\gamma}_{e,b}^{(r')})_c = \sum_{i=1}^{d}\sum_{j=1}^{d}(\Gamma_{e,b}^{(r)})_{ij}(\Gamma_{e,b}^{(r')})_{ij}$. Assume independent observations with population mean $\Gamma_{e,b}^{\mathrm{pop}} := \mathbb{E}[\Gamma_{e,b}^{(r)}]$, vectorized mean $\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}} := \mathrm{vec}(\Gamma_{e,b}^{\mathrm{pop}})$, covariance $\Sigma_{e,b} \in \mathbb{R}^{d^2\times d^2}$, and finite fourth moments. The superscript pop denotes a population mean, whereas (r) denotes a sample. Covariance entry (c,c') is $\mathbb{E}[((\boldsymbol{\gamma}_{e,b}^{(r)})_c - (\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_c)((\boldsymbol{\gamma}_{e,b}^{(r)})_{c'} - (\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_{c'})]$. Entries within a block can be correlated.

The hypotheses are $H_{0,e,b} : \Gamma_{e,b}^{\mathrm{pop}} = 0_{d\times d}$ and $H_{1,e,b} : \Gamma_{e,b}^{\mathrm{pop}} \neq 0_{d\times d}$, equivalently $\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 = 0$ versus $\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 > 0$. Here $0_{d\times d}$ is the zero matrix. Subscripts 0 and 1 identify the null and alternative hypotheses. For one sample, $\mathbb{E}\|\Gamma_{e,b}^{(r)}\|_F^2 = \|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 + \sum_{c=1}^{d^2}(\Sigma_{e,b})_{cc}$.

Unbiasedness of the cross-sample estimator. For the common population mean $\Gamma_{e,b}^{\mathrm{pop}}$ defined above, independence across samples and finite second moments suffice, even when entries within a

block are dependent. For distinct samples r, r' and entry indices i, j,

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{d}\sum_{j=1}^{d}(\Gamma_{e,b}^{(r)})_{ij}(\Gamma_{e,b}^{(r')})_{ij}\right] &= \sum_{i=1}^{d}\sum_{j=1}^{d}\mathbb{E}[(\Gamma_{e,b}^{(r)})_{ij}(\Gamma_{e,b}^{(r')})_{ij}] \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d}\mathbb{E}[(\Gamma_{e,b}^{(r)})_{ij}]\mathbb{E}[(\Gamma_{e,b}^{(r')})_{ij}] \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d}(\Gamma_{e,b}^{\mathrm{pop}})_{ij}^2 = \|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2. \end{aligned} \tag{16}$$

Linearity of expectation then gives

$$\mathbb{E}[\widehat{\theta}_{e,b}] = \frac{2}{R(R-1)}\sum_{r=1}^{R-1}\sum_{r'=r+1}^{R}\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 = \|\mathbb{E}[\Gamma_{e,b}^{(r)}]\|_F^2. \tag{17}$$

This is the one-sample cross-product identity underlying the construction of Chen & Qin (2010). It applies to general sample distributions satisfying the stated independence and moment conditions. A zero population mean describes cancellation in expectation and allows nonzero sample gradients.

Fixed stratified sampling. For MUSIC-AVQA, the selection subset contains a fixed number of auxiliary samples from each of the A, V, and AV tasks. For independent observations with different population means, write $\Gamma_{e,b}^{\mathrm{pop},r} := \mathbb{E}[\Gamma_{e,b}^{(r)}]$ and $\overline{\Gamma}_{e,b}^{\mathrm{pop}} := R^{-1}\sum_{r=1}^{R}\Gamma_{e,b}^{\mathrm{pop},r}$. The extra index r allows the population mean to vary by sample group, while the bar averages these means. Then

$$\begin{aligned}\mathbb{E}[\widehat{\theta}_{e,b}] &= \frac{\left\|\sum_{r=1}^{R}\Gamma_{e,b}^{\mathrm{pop},r}\right\|_F^2 - \sum_{r=1}^{R}\|\Gamma_{e,b}^{\mathrm{pop},r}\|_F^2}{R(R-1)} \\ &= \|\overline{\Gamma}_{e,b}^{\mathrm{pop}}\|_F^2 - \frac{\sum_{r=1}^{R}\|\Gamma_{e,b}^{\mathrm{pop},r} - \overline{\Gamma}_{e,b}^{\mathrm{pop}}\|_F^2}{R(R-1)}. \end{aligned} \tag{18}$$

When these means coincide, the estimator is unbiased for the squared pooled mean. When they differ, the second term accounts for their heterogeneity. Experiments use this pooled score for allocation, with the displayed between-group correction determining its expectation when the means differ.

A.2.2 Variance estimation under the null hypothesis

Variance of the cross-sample estimator. Under the independent common-mean, common-covariance model above, consider each distinct pair sum $\sum_{c=1}^{d^2}(\boldsymbol{\gamma}_{e,b}^{(r)})_c(\boldsymbol{\gamma}_{e,b}^{(r')})_c$, $1 \le r < r' \le R$. Under $H_{0,e,b}$, it has mean zero and variance $\tau_{e,b} := \sum_{c=1}^{d^2}\sum_{c'=1}^{d^2}(\Sigma_{e,b})_{cc'}^2$. Distinct pairs are uncorrelated, including pairs sharing one sample, because the other samples are independent and mean zero. The estimator averages $R(R-1)/2$ such pairs, so

$$\mathrm{Var}(\widehat{\theta}_{e,b}) = \frac{2\tau_{e,b}}{R(R-1)} =: \sigma_{e,b}^2 \qquad \text{under } H_{0,e,b}. \tag{19}$$

Under a nonzero population mean, the variance has the additional term $\frac{4}{R}\sum_{c=1}^{d^2}\sum_{c'=1}^{d^2}(\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_c(\Sigma_{e,b})_{cc'}(\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_{c'}$. The normalization $\widehat{\sigma}_{e,b}^2$ estimates the first term, which is the variance under $H_{0,e,b}$.

Split-half covariance estimator. Divide the auxiliary indices into disjoint halves $\mathcal{I}_1$ and $\mathcal{I}_2$ whose union is $\{1, \dots, R\}$. For half $h \in \{1, 2\}$, let $n_h := |\mathcal{I}_h| \ge 2$. Its mean vector and covariance entries are

$$\begin{aligned}\overline{\boldsymbol{\gamma}}_{e,b}^{(h)} &:= \frac{1}{n_h}\sum_{r\in\mathcal{I}_h}\boldsymbol{\gamma}_{e,b}^{(r)}, \\ (\widehat{\Sigma}_{e,b}^{(h)})_{cc'} &:= \frac{1}{n_h - 1}\sum_{r\in\mathcal{I}_h}\left((\boldsymbol{\gamma}_{e,b}^{(r)})_c - (\overline{\boldsymbol{\gamma}}_{e,b}^{(h)})_c\right)\left((\boldsymbol{\gamma}_{e,b}^{(r)})_{c'} - (\overline{\boldsymbol{\gamma}}_{e,b}^{(h)})_{c'}\right). \end{aligned} \tag{20}$$

The superscript (h) identifies the half used for estimation, and $c, c' \in \{1, \ldots, d^2\}$ index vector coordinates. Multiplying corresponding covariance entries gives

$$\widehat{\tau}_{e,b} := \sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\widehat{\Sigma}_{e,b}^{(1)})_{cc'} (\widehat{\Sigma}_{e,b}^{(2)})_{cc'}. \tag{21}$$

The variance and standard-deviation estimates used in Equation 2 are $\widehat{\sigma}_{e,b}^2 := 2\widehat{\tau}_{e,b}/[R(R-1)]$ and $\widehat{\sigma}_{e,b} := \sqrt{\widehat{\sigma}_{e,b}^2}$.

Split-half unbiasedness and consistency. With the split chosen independently of the observations, the two halves contain independent samples. Each sample covariance is unbiased, so

$$\begin{aligned}\mathbb{E}[\widehat{\tau}_{e,b}] &= \sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} \mathbb{E}[(\widehat{\Sigma}_{e,b}^{(1)})_{cc'}]\, \mathbb{E}[(\widehat{\Sigma}_{e,b}^{(2)})_{cc'}] \\ &= \sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\Sigma_{e,b})_{cc'}^2 = \tau_{e,b}.\end{aligned}$$

This is the entrywise form of the covariance trace-product construction in Cao et al. (2019, Lemma 3.1 and its appendix proof). Consequently, $\mathbb{E}[\widehat{\sigma}_{e,b}^2] = \sigma_{e,b}^2$. For $\tau_{e,b} > 0$, if $\mathrm{Var}(\widehat{\tau}_{e,b})/\tau_{e,b}^2 \to 0$ as R grows, Chebyshev's inequality gives $\widehat{\tau}_{e,b}/\tau_{e,b} \xrightarrow{\mathbb{P}} 1$. The arrow denotes convergence in probability. At fixed block dimension, i.i.d. samples with finite fourth moments and growing half-sample sizes provide consistency. The method uses the resulting scores for fixed-budget ranking.

A.2.3 Cross-sample agreement of gradient effects

We interpret the population score $T_{e,b}^{\mathrm{pop}} := \|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2/\sqrt{\tau_{e,b}}$ through the agreement of mean effects estimated from independent sample groups. Using the group means defined in Equation 20, define

$$Z_{e,b} := \sum_{c=1}^{d^2} (\overline{\gamma}_{e,b}^{(1)})_c (\overline{\gamma}_{e,b}^{(2)})_c. \tag{22}$$

This is the sum of products of corresponding entries in the two mean effect matrices. We call $Z_{e,b} > 0$ directional agreement: the two vectorized mean effects form an acute angle.

Proposition 1 (Cross-sample agreement of gradient effects)**.** *Fix an effect e and block b. Assume i.i.d. sample effects with finite second moments, nonzero population mean $\Gamma_{e,b}^{\mathrm{pop}}$, covariance $\Sigma_{e,b}$, and $\tau_{e,b} > 0$. For two independent sample groups of sizes $n_1, n_2 \geq 2$,*

$$\mathbb{P}(Z_{e,b} > 0) \geq \frac{n_1 T_{e,b}^{\mathrm{pop}}}{1 + n_1 T_{e,b}^{\mathrm{pop}}} \frac{n_2 T_{e,b}^{\mathrm{pop}}}{1 + n_2 T_{e,b}^{\mathrm{pop}}}. \tag{23}$$

For fixed group sizes, this lower bound increases with $T_{e,b}^{\mathrm{pop}}$.

Proof. Subtracting the population mean from each group mean gives independent, zero-mean deviations with covariances $\Sigma_{e,b}/n_1$ and $\Sigma_{e,b}/n_2$. Expanding Equation 22 and using independence gives

$$\begin{aligned}\mathbb{E}[Z_{e,b}] &= \|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2, \\ \mathrm{Var}(Z_{e,b}) &= \frac{\tau_{e,b}}{n_1 n_2} + \left(\frac{1}{n_1} + \frac{1}{n_2}\right) \sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\gamma_{e,b}^{\mathrm{pop}})_c (\Sigma_{e,b})_{cc'} (\gamma_{e,b}^{\mathrm{pop}})_{c'}.\end{aligned} \tag{24}$$

Cauchy–Schwarz applied to the covariance entries yields

$$\begin{aligned}&\sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\gamma_{e,b}^{\mathrm{pop}})_c (\Sigma_{e,b})_{cc'} (\gamma_{e,b}^{\mathrm{pop}})_{c'} \\ &\quad \leq \left(\sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\Sigma_{e,b})_{cc'}^2\right)^{1/2} \left(\sum_{c=1}^{d^2} \sum_{c'=1}^{d^2} (\gamma_{e,b}^{\mathrm{pop}})_c^2 (\gamma_{e,b}^{\mathrm{pop}})_{c'}^2\right)^{1/2} = \sqrt{\tau_{e,b}}\, \|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2.\end{aligned}$$

Since $\mathbb{E}[Z_{e,b}] > 0$, Cantelli's one-sided variance inequality gives

$$\mathbb{P}(Z_{e,b} > 0) \geq \frac{(\mathbb{E}[Z_{e,b}])^2}{(\mathbb{E}[Z_{e,b}])^2 + \mathrm{Var}(Z_{e,b})} \geq \frac{\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^4}{\big(\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 + \sqrt{\tau_{e,b}}/n_1\big)\big(\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2 + \sqrt{\tau_{e,b}}/n_2\big)}.$$

Dividing numerator and denominator by $\tau_{e,b}$ proves Equation 23. Each factor increases with $T_{e,b}^{\mathrm{pop}}$.

Interpretation. The population score orders lower bounds on agreement, rather than exact agreement probabilities. Proposition 2 connects this population criterion to the sample-size-normalized estimated score under its consistency assumptions. Both results use the common-population sampling model. Equation 18 gives the estimator's expectation for the fixed stratified selection subset.

A.2.4 Population ranking target and allocation stability

The selector ranks estimated squared mean gradient-effect magnitude relative to the estimated standard deviation under the null hypothesis that the population mean gradient effect is zero. The following result identifies its population target and conditions for stable allocation.

Proposition 2 (Population ranking target and allocation stability)**.** *Fix the candidate matrices, block dimension, and quotas. At each effect–block pair, assume independent identically distributed observations with finite fourth moments, $\tau_{e,b} > 0$, and one predetermined split whose two half-sample sizes tend to infinity. For the population score $T_{e,b}^{\mathrm{pop}}$ defined in Appendix A.2.3, the sample-size-normalized score satisfies*

$$\frac{\widehat{T}_{e,b}}{\sqrt{R(R-1)/2}} = \frac{\widehat{\theta}_{e,b}}{\sqrt{\widehat{\tau}_{e,b}}} \xrightarrow{\mathbb{P}} T_{e,b}^{\mathrm{pop}}. \tag{25}$$

If each population-score matrixwise assignment has a unique optimum separated by a positive score gap, and each subsequent per-effect selection has a strict cutoff gap, the two-stage allocation using $\widehat{T}$ equals that using T^{pop} with probability tending to one.

Proof. The cross-sample statistic is unbiased for $\|\Gamma_{e,b}^{\mathrm{pop}}\|_F^2$, and

$$\mathrm{Var}(\widehat{\theta}_{e,b}) = \frac{2\tau_{e,b}}{R(R-1)} + \frac{4}{R}\sum_{c=1}^{d^2}\sum_{c'=1}^{d^2}(\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_c(\Sigma_{e,b})_{cc'}(\boldsymbol{\gamma}_{e,b}^{\mathrm{pop}})_{c'} \longrightarrow 0.$$

For fixed block dimension, each half-sample covariance converges in probability to $\Sigma_{e,b}$. The continuous-mapping theorem gives $\widehat{\tau}_{e,b} \to \tau_{e,b}$. Normalizing by its positive square root then proves Equation 25. The candidate set is finite, so convergence is uniform over its effect–block pairs by a union bound. Every feasible matrixwise assignment contains the same number of blocks. Uniform score error smaller than its assignment gap divided by twice that number preserves its maximizer. Once assignments agree, a score error below half each per-effect cutoff gap preserves final selection as well.

Ranking interpretation. At a common sample size, the score ranks squared mean factorial effects relative to the covariance trace-product scale. Cross-sample products remove the single-sample noise contribution from the numerator. Under the stated sampling assumptions and positive ranking gaps, the implemented two-stage allocation converges to its population-score allocation. In experiments, the score determines a fixed-budget ranking. We evaluate downstream utility through task accuracy and retention.

Comparability and distribution. At the same sample size, $T_{e,b}^{\mathrm{pop}}$ provides a common dimensionless ranking criterion while allowing different block covariances. Rescaling all observations of a block by any nonzero constant leaves $\widehat{T}_{e,b}$ unchanged: both numerator and denominator scale quadratically. Consequently, the score measures standardized effect strength. Statistical feature screening similarly ranks coordinates using standardized statistics (Fan & Fan, 2008). The allocation result above follows from score consistency and separated ranking gaps. A normal approximation under the zero-mean hypothesis requires additional high-dimensional covariance and moment conditions (Chen & Qin, 2010), whereas the allocation procedure uses scores directly with fixed quotas.

A.2.5 EFFICIENT SCORE COMPUTATION

Computation without covariance-matrix storage. The cross-sample estimate can be evaluated as

$$\widehat{\theta}_{e,b} = \frac{\left\|\sum_{r=1}^{R} \boldsymbol{\gamma}_{e,b}^{(r)}\right\|_2^2 - \sum_{r=1}^{R} \left\|\boldsymbol{\gamma}_{e,b}^{(r)}\right\|_2^2}{R(R-1)}.$$

Using the half-sample means from Equation 20,

$$\widehat{\tau}_{e,b} = \frac{1}{(n_1-1)(n_2-1)} \sum_{r\in\mathcal{I}_1} \sum_{r'\in\mathcal{I}_2} \left[\sum_{c=1}^{d^2} \big((\boldsymbol{\gamma}_{e,b}^{(r)})_c - (\overline{\boldsymbol{\gamma}}_{e,b}^{(1)})_c\big)\big((\boldsymbol{\gamma}_{e,b}^{(r')})_c - (\overline{\boldsymbol{\gamma}}_{e,b}^{(2)})_c\big)\right]^2.$$

This requires pairwise sums for $n_1 n_2$ sample pairs rather than storing $d^2 \times d^2$ covariance matrices. Statistics use double precision.

A.3 DISJOINT ASSIGNMENT AND FINAL SELECTION

For matrix ν, let B_ν be its candidate-block count, N_ν its parameter budget, and $\mathcal{B}_{\nu,e}$ its assigned set for effect e. These are the matrix-indexed versions of $B, N, \mathcal{B}_e$ in Section 3.2. Write $\widehat{T}_{e,(\nu,b)}$ for the score at local block b in matrix ν. The assignment is

$$\begin{aligned} &\max_{\{\mathcal{B}_{\nu,e}\}_e} \quad \sum_{e\in\{\mathrm{S,A,V,I}\}} \sum_{b\in\mathcal{B}_{\nu,e}} \widehat{T}_{e,(\nu,b)} \\ &\text{subject to} \quad \mathcal{B}_{\nu,e} \subseteq \{1,\dots,B_\nu\}, \quad |\mathcal{B}_{\nu,e}| = \frac{N_\nu}{4d^2}, \quad \mathcal{B}_{\nu,e} \cap \mathcal{B}_{\nu,e'} = \varnothing \quad (e \neq e'). \end{aligned} \tag{26}$$

We solve this exactly as a linear-sum assignment: each effect has $N_\nu/(4d^2)$ slots, each slot receives one distinct block, and its score is $\widehat{T}_{e,(\nu,b)}$. The implementation uses `scipy.optimize.linear_sum_assignment` with maximization enabled. For each effect, rank the assigned blocks from all matrices by $\widehat{T}_{e,(\nu,b)}$ and retain the highest-scoring $N_{\mathrm{total}}/(4d^2)$ blocks to form $\mathcal{B}_e^\star$, whose block identities include both ν and b. Here N_{total} is the final parameter budget across all matrices. It can be smaller than the sum of the per-matrix budgets before trimming. Thus $|\mathcal{B}_e^\star| = N_{\mathrm{total}}/(4d^2)$ and $N_{\mathrm{total}} = d^2 \sum_{e\in\{\mathrm{S,A,V,I}\}} |\mathcal{B}_e^\star|$. Ties use a fixed matrix/block order.

Meaning of summed-score allocation. The objective maximizes the total standardized effect score subject to equal quotas that reserve capacity for each effect and disjointness that resolves competition for blocks scoring highly for multiple effects. Score-based fixed-budget selection appears in SNIP (Lee et al., 2019), and summed-score balanced assignment appears in BASE Layers (Lewis et al., 2021). Here, standardized effect scores define the assignment utility. Their scale matters: a nonlinear monotone transformation can change the maximum summed-score assignment while preserving each effect's individual ranking. The later budget trim enforces the final budget after per-matrix assignment.

Selected block-set construction. Keep all pretrained model parameters fixed. For each auxiliary sample used for block selection, compute the four controlled gradients and the factorial contrasts in Section 3.2. After vectorizing each effect, calculate $\widehat{\theta}_{e,b}$ and the scalar covariance-product estimate $\widehat{\tau}_{e,b}$ from independently centered sample halves. Use these estimates to form $\widehat{T}_{e,b}$ using Equation 2. Within every weight matrix, solve the maximum-score disjoint assignment with $N_\nu/(4d^2)$ blocks per effect. Rank the assigned blocks across matrices within each effect and retain $|\mathcal{B}_e^\star| = N_{\mathrm{total}}/(4d^2)$, using a fixed ordering for ties. Initialize each selected block update to $\Delta_{b,0} = 0$. Appendix D.2 gives the numerical budgets and quotas.

B SHARED ADAPTATION AND SELECTIVE PROTECTION

B.1 SHARED ADAPTATION

The block states for each modality combination are summarized below.

m_t	Active blocks	Inactive blocks	Protected	Trainable
A	$\mathcal{B}^\star_{\mathrm{S}} \cup \mathcal{B}^\star_{\mathrm{A}}$	$\mathcal{B}^\star_{\mathrm{V}} \cup \mathcal{B}^\star_{\mathrm{I}}$	$\mathcal{P}_t$	$\mathcal{C}_{\mathrm{A}} \setminus \mathcal{P}_t$
V	$\mathcal{B}^\star_{\mathrm{S}} \cup \mathcal{B}^\star_{\mathrm{V}}$	$\mathcal{B}^\star_{\mathrm{A}} \cup \mathcal{B}^\star_{\mathrm{I}}$	$\mathcal{P}_t$	$\mathcal{C}_{\mathrm{V}} \setminus \mathcal{P}_t$
AV	$\mathcal{B}^\star_{\mathrm{S}} \cup \mathcal{B}^\star_{\mathrm{A}} \cup \mathcal{B}^\star_{\mathrm{V}} \cup \mathcal{B}^\star_{\mathrm{I}}$	$\varnothing$	$\mathcal{P}_t$	$\mathcal{C}_{\mathrm{AV}} \setminus \mathcal{P}_t$

Dense materialization. Updates at inactive blocks are frozen and omitted from the forward pass. For evaluation task m after stage t, materialize $W_t(v_m, a_m)$ using Equation 4 and apply the standard dense linear layer.

B.2 Block importance and protection

Squared-gradient loss sensitivity. Selection estimates the mean of a modality contrast using the pretrained MLLM. Protection instead measures how a learned task's individual losses respond to parameter changes. For an active block, the learned update Δ_b defined in Appendix A.1 enters the effective weights additively, so $\nabla_{\Delta_b} \ell(x, y; \boldsymbol{\omega}_t) = G_b(x, y; \boldsymbol{\omega}_t)$. At the model $\boldsymbol{\omega}_t$ learned from $\mathcal{D}_t$, the first-order change in auxiliary sample r's loss under a block displacement u_b is $\sum_{i=1}^{d} \sum_{j=1}^{d} (G_b(x_r(v_{m_t}, a_{m_t}), y_r; \boldsymbol{\omega}_t))_{ij} (u_b)_{ij}$, where i, j index matrix entries. Using the importance score from Equation 6, Cauchy–Schwarz gives

$$\frac{1}{R_{m_t}} \sum_{r=1}^{R_{m_t}} \left[\sum_{i=1}^{d} \sum_{j=1}^{d} (G_b(x_r(v_{m_t}, a_{m_t}), y_r; \boldsymbol{\omega}_t))_{ij} (u_b)_{ij} \right]^2 \leq s_{b,t} \|u_b\|_F^2. \tag{27}$$

For a fixed block displacement norm, the upper bound on mean squared first-order loss change is proportional to $s_{b,t}$. Squaring before averaging retains sensitivity even when sample gradients point in opposite directions and cancel in the mean. This makes the score suitable for identifying blocks whose updates can affect earlier-task losses.

Important-block records and optimizer state. When a task's total importance is zero, its importance set is empty. Otherwise, Equation 7 selects the fewest top-ranked blocks whose scores sum to at least η of that total. During subsequent training, protected blocks receive neither gradient nor weight-decay updates. Their delta values, Adam moments, and optimizer step counters also remain unchanged.

B.3 Continual training procedure

Initialize each task's importance record to the empty set. For $t = 1, \ldots, K$, use m_t to determine the active set $\mathcal{C}_{m_t}$ and Equation 8 to obtain $\mathcal{P}_t$, with $\mathcal{P}_1 = \varnothing$. Fine-tune on $\mathcal{D}_t$ using only the trainable blocks $\mathcal{W}_t$ defined in Section 3.4. Protected and inactive updates receive no parameter or optimizer-state changes, including weight decay. Afterward, compute $s_{b,t}$ on auxiliary samples for the task represented by $\mathcal{D}_t$ and refresh $\mathcal{F}_{m_t}^{(t)}$ using Equation 7. Carry all other task records forward unchanged. The candidate count is $k \in \{0, \ldots, |\mathcal{C}_{m_t}|\}$. An empty sum is zero, so zero total importance gives an empty important-block set. Retain the learned updates and optimizer state for the next stage, and evaluate all three tasks.

C Proofs and Supporting Analyses

We prove the protection result and analyze current-task descent and loss changes under shared updates. We use the stage datasets $\mathcal{D}_t$, task identities $m_t \in \mathcal{M}$, and trainable block sets $\mathcal{W}_t = \mathcal{C}_{m_t} \setminus \mathcal{P}_t$ from Section 3. At the start of stage t, Equation 8 determines $\mathcal{P}_t$ using the latest task-specific important sets $\mathcal{F}_m^{(t-1)}$ for $m \in \mathcal{M} \setminus \{m_t\}$. Each task's record starts at $\mathcal{F}_m^{(0)} = \varnothing$. After fine-tuning on $\mathcal{D}_t$, update $\mathcal{F}_{m_t}^{(t)}$ and carry the other task records forward unchanged. All losses use the sample loss ℓ from Section 3.2. The auxiliary-sample loss $\mathcal{L}_m$ is defined in Section 4. Superscripts train and test identify losses on training and evaluation data, respectively. A parameter displacement u changes the model from $\boldsymbol{\omega}$ to $\boldsymbol{\omega} + u$. Its block $u_b \in \mathbb{R}^{d \times d}$ contains the changes at block b. Entries outside the specified block set are zero. Thus $\|u\|_2^2 = \sum_b \|u_b\|_F^2$. Here $\|\cdot\|_2$ is the Euclidean vector norm, while $\|\cdot\|_F$ is the Frobenius norm defined in Section 3.2. Only selected weight updates change, while all other parameters affecting the analyzed loss and its modality-based activation stay fixed.

C.1 FINAL RETENTION AFTER THE LAST LEARNING STAGE

Proof of Theorem 1. Fix task $m = m_{t'}$, its last learning or relearning stage t' in the K-stage sequence, and the model $\boldsymbol{\omega}_{t'}$ after that stage. The mean loss from Section 4 is $\mathcal{L}_m(\boldsymbol{\omega}) := R_m^{-1} \sum_{r=1}^{R_m} \ell(x_r(v_m, a_m), y_r; \boldsymbol{\omega})$, where r indexes the R_m auxiliary samples for that task and v_m, a_m are its video and audio indicators. The active set $\mathcal{C}_m$ and importance scores $s_{b,t'}$ are defined in Equations 5 and 6, respectively. Since learned updates outside $\mathcal{C}_m$ are excluded from this task's forward pass and pretrained weights remain fixed, only updates in $\mathcal{C}_m$ can affect its loss. Write $u_b := \Delta_{b,K} - \Delta_{b,t'}$. Final evaluation uses $W_K(v_m, a_m)$, retaining the earlier task's modality indicators. Let $\mathcal{F} \subseteq \mathcal{C}_m$ be any protection set whose updates remain frozen through stage K, so $u_b = 0$ on $\mathcal{F}$. The per-stage change constraint in Theorem 1 and the triangle inequality give

$$\left(\sum_{b \in \mathcal{C}_m} \|u_b\|_F^2 \right)^{1/2} = \left(\sum_{b \in \mathcal{C}_m} \left\| \sum_{t=t'+1}^{K} (\Delta_{b,t} - \Delta_{b,t-1}) \right\|_F^2 \right)^{1/2} \leq \sum_{t=t'+1}^{K} \left(\sum_{b \in \mathcal{C}_m} \|\Delta_{b,t} - \Delta_{b,t-1}\|_F^2 \right)^{1/2} \leq (K - t')\rho.$$

The derivation conditions on the auxiliary samples and starting model, giving a deterministic bound on their mean loss.

Smoothness, freezing, Cauchy–Schwarz, and Jensen's inequality give

$$\begin{aligned}
&\mathcal{L}_m(\boldsymbol{\omega}_K) - \mathcal{L}_m(\boldsymbol{\omega}_{t'}) \\
&\leq \sum_{b \in \mathcal{C}_m \setminus \mathcal{F}} \sum_{i=1}^{d} \sum_{j=1}^{d} \left(\nabla_{\Delta_b} \mathcal{L}_m(\boldsymbol{\omega}_{t'}) \right)_{ij} (u_b)_{ij} + \frac{L}{2}(K - t')^2 \rho^2 \\
&\leq (K - t')\rho \sqrt{\sum_{b \in \mathcal{C}_m \setminus \mathcal{F}} \left\| \frac{1}{R_m} \sum_{r=1}^{R_m} G_b(x_r(v_m, a_m), y_r; \boldsymbol{\omega}_{t'}) \right\|_F^2} + \frac{L}{2}(K - t')^2 \rho^2 \\
&\leq (K - t')\rho \sqrt{\underbrace{\sum_{b \in \mathcal{C}_m} s_{b,t'}}_{\text{total importance}} - \underbrace{\sum_{b \in \mathcal{F}} s_{b,t'}}_{\text{protected importance}}} + \frac{L}{2}(K - t')^2 \rho^2.
\end{aligned}$$

Here i, j index matrix entries and r indexes auxiliary samples. The first inequality is the descent lemma (Bottou et al., 2018, Equation (4.3)). The last follows because the squared norm of an average gradient is no larger than the average squared norm in Equation 6. This proves Equation 9 for any such protection set $\mathcal{F}$.

Since t' is the task's last learning stage, every subsequent dataset represents a different task from $\mathcal{D}_{t'}$. Its record remains $\mathcal{F}_m^{(t')}$ through stage K. Equation 8 therefore makes each block in $\mathcal{F}_m^{(t')}$ either inactive or protected. In both cases its learned update and optimizer state stay fixed, including weight decay. Consequently $u_b = 0$ on $\mathcal{F}_m^{(t')}$. Thus the theorem applies to ASAP's selected set through the final stage K.

Proof of Corollary 1. Fix the scores $s_{b,t'}$, and let $k^\star = |\mathcal{F}_m^{(t')}|$ be the count chosen by Equation 7 after stage t'. The scores are nonnegative and ordered as $s_{b_{(1)},t'} \geq s_{b_{(2)},t'} \geq \cdots$. For any $\mathcal{F} \subseteq \mathcal{C}_m$ with $|\mathcal{F}| = k^\star$,

$$\sum_{b \in \mathcal{F}} s_{b,t'} \leq \sum_{i=1}^{k^\star} s_{b_{(i)},t'} = \sum_{b \in \mathcal{F}_m^{(t')}} s_{b,t'}.$$

Indeed, replacing any selected lower-ranked block with an omitted higher-ranked block cannot decrease the sum. Repeating these exchanges gives the top-$k^\star$ set. Since total importance, the remaining sequence length $K - t'$, ρ, and L are fixed, maximizing protected importance minimizes the right-hand side of Equation 9 among equally sized sets held frozen through stage K. This proves optimality of the final loss-increase bound.

For any $k < k^\star$ and any $\mathcal{F} \subseteq \mathcal{C}_m$ with $|\mathcal{F}| = k$, the definition of $k^\star$ gives

$$\sum_{b\in\mathcal{F}} s_{b,t'} \le \sum_{i=1}^{k} s_{b_{(i)},t'} < \eta \sum_{b\in\mathcal{C}_m} s_{b,t'}.$$

No smaller set therefore meets the required coverage, whereas $\mathcal{F}_m^{(t')}$ does. This proves minimum cardinality. Ties can give multiple optimal sets. If total importance is zero, the empty set suffices. Finally, the coverage rule gives

$$\mathcal{L}_m(\boldsymbol{\omega}_K) - \mathcal{L}_m(\boldsymbol{\omega}_{t'}) \le (K-t')\rho\sqrt{(1-\eta)\sum_{b\in\mathcal{C}_m} s_{b,t'}} + \frac{L}{2}(K-t')^2\rho^2. \tag{28}$$

If the task recurs earlier in the sequence, the analysis uses the model and refreshed importance set from its last occurrence t'.

Role of selection and earlier training. Selection determines $\mathcal{C}_m$, earlier training determines $\boldsymbol{\omega}_{t'}$, and auxiliary samples determine $\mathcal{L}_m$ and its scores. The comparison fixes these quantities, $K-t'$, the per-stage limit ρ, and L. For these fixed quantities, ranking minimizes the final loss-increase bound among equally sized protection sets. The realized loss change also depends on the displacement produced by training. Taking unions preserves each task's bound, while minimum cardinality applies to each task's important set individually.

C.2 Plasticity: descent retained after protection

During fine-tuning on $\mathcal{D}_t$, the mean training loss is $\mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}) := |\mathcal{D}_t|^{-1}\sum_{(x,y)\in\mathcal{D}_t}\ell(x,y;\boldsymbol{\omega})$, and $G_{t,b}^{\text{train}} := \nabla_{\Delta_b}\mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}) \in \mathbb{R}^{d\times d}$ is its block gradient. The superscript identifies training data and the subscript t identifies the current task's stage. The fraction of squared gradient magnitude retained after protection is $\chi_t := \sum_{b\in\mathcal{W}_t}\|G_{t,b}^{\text{train}}\|_F^2 / \sum_{b\in\mathcal{C}_{m_t}}\|G_{t,b}^{\text{train}}\|_F^2$, with $\chi_t = 0$ if the denominator is zero.

Theorem 2 (Current-task descent capacity)**.** *Suppose $\mathcal{L}_t^{\text{train}}$ has smoothness constant L_t^{train} along a displacement u supported on $\mathcal{W}_t$. Then*

$$\mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}+u) - \mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}) \le \sum_{b\in\mathcal{W}_t}\sum_{i=1}^{d}\sum_{j=1}^{d}(G_{t,b}^{\text{train}})_{ij}(u_b)_{ij} + \frac{L_t^{\text{train}}}{2}\sum_{b\in\mathcal{W}_t}\|u_b\|_F^2. \tag{29}$$

A negative right-hand side is sufficient for loss reduction. For the block-gradient step $u_b = -\alpha G_{t,b}^{\text{train}}$ on $\mathcal{W}_t$ and $u_b = 0$ elsewhere, where α is the step size and $0 < \alpha \le 1/L_t^{\text{train}}$,

$$\mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}+u) - \mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}) \le -\frac{\alpha}{2}\sum_{b\in\mathcal{W}_t}\|G_{t,b}^{\text{train}}\|_F^2 = -\frac{\alpha}{2}\chi_t\sum_{b\in\mathcal{C}_{m_t}}\|G_{t,b}^{\text{train}}\|_F^2. \tag{30}$$

For a fixed update-norm limit $\rho_{\text{step}} \ge 0$ with $\sum_{b\in\mathcal{W}_t}\|u_b\|_F^2 \le \rho_{\text{step}}^2$, the smallest possible first-order loss change is $-\rho_{\text{step}}\sqrt{\sum_{b\in\mathcal{W}_t}\|G_{t,b}^{\text{train}}\|_F^2}$.

Remark (Learning plasticity). Theorem 2 describes learning capacity after protection. If a trainable block has a nonzero training-loss gradient, a sufficiently small step opposite the trainable gradients decreases current-task loss while preserving protected blocks. The summed squared trainable gradients determine the greatest first-order improvement for a fixed parameter-change limit. Equation 29 separately evaluates the sufficient condition for an actual AdamW update.

Proof. Applying the descent lemma to $\mathcal{L}_t^{\text{train}}$ gives Equation 29, because $u_b = 0$ outside $\mathcal{W}_t$. Substituting $u_b = -\alpha G_{t,b}^{\text{train}}$ yields

$$\mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}+u) - \mathcal{L}_t^{\text{train}}(\boldsymbol{\omega}) \le -\alpha\left(1 - \frac{L_t^{\text{train}}\alpha}{2}\right)\sum_{b\in\mathcal{W}_t}\|G_{t,b}^{\text{train}}\|_F^2,$$

which proves Equation 30. Cauchy–Schwarz bounds the first-order sum below by $-\rho_{\text{step}}\sqrt{\sum_{b\in\mathcal{W}_t}\|G_{t,b}^{\text{train}}\|_F^2}$. Equality is attained by $u_b = -\rho_{\text{step}}G_{t,b}^{\text{train}} / \sqrt{\sum_{b'\in\mathcal{W}_t}\|G_{t,b'}^{\text{train}}\|_F^2}$ for $b \in \mathcal{W}_t$, where b' is the summation index over trainable blocks. If every trainable gradient is zero, the first-order change is zero.

C.3 Effects of shared updates on other tasks

A trainable block used by multiple tasks can change their losses simultaneously. We characterize this coupling through the blocks shared with the current task. Let $\mathcal{L}_m^{\text{test}}(\boldsymbol{\omega})$ be the mean evaluation loss for task m, and let u be a displacement supported on $\mathcal{W}_t$ during fine-tuning on $\mathcal{D}_t$. Only blocks in $\mathcal{W}_t \cap \mathcal{C}_m$ can affect this evaluation loss. Define their evaluation gradients as $G_{m,b}^{\text{test}} := \nabla_{\Delta_b} \mathcal{L}_m^{\text{test}}(\boldsymbol{\omega})$, using the same block-gradient notation as for training loss.

Theorem 3 (Loss change under shared updates)**.** *If $\mathcal{L}_m^{\text{test}}$ has smoothness constant L_m^{test} along the change in its active blocks, then*

$$\begin{aligned}&\mathcal{L}_m^{\text{test}}(\boldsymbol{\omega} + u) - \mathcal{L}_m^{\text{test}}(\boldsymbol{\omega}) \\ &\quad \le \sum_{b \in \mathcal{W}_t \cap \mathcal{C}_m} \sum_{i=1}^{d} \sum_{j=1}^{d} (G_{m,b}^{\text{test}})_{ij} (u_b)_{ij} + \frac{L_m^{\text{test}}}{2} \sum_{b \in \mathcal{W}_t \cap \mathcal{C}_m} \|u_b\|_F^2.\end{aligned} \tag{31}$$

A negative right-hand side guarantees lower evaluation loss.

Remark (Implications for sharing). An update on trainable blocks used by both tasks improves the other task when its first-order loss reduction exceeds the curvature term.

Proof. The evaluation loss depends on changing weights only through $\mathcal{C}_m$. Applying the descent lemma to the overlapping blocks gives Equation 31. For the training-gradient step from Theorem 2,

$$\begin{aligned}&\mathcal{L}_m^{\text{test}}(\boldsymbol{\omega} + u) - \mathcal{L}_m^{\text{test}}(\boldsymbol{\omega}) \\ &\quad \le -\alpha \sum_{b \in \mathcal{W}_t \cap \mathcal{C}_m} \sum_{i=1}^{d} \sum_{j=1}^{d} (G_{m,b}^{\text{test}})_{ij} (G_{t,b}^{\text{train}})_{ij} + \frac{L_m^{\text{test}} \alpha^2}{2} \sum_{b \in \mathcal{W}_t \cap \mathcal{C}_m} \|G_{t,b}^{\text{train}}\|_F^2.\end{aligned}$$

Thus sufficiently aligned training and evaluation gradients permit beneficial transfer. For an actual optimizer update, the entrywise sum in Equation 31 gives the first-order change directly. If $\mathcal{W}_t \cap \mathcal{C}_m = \varnothing$, stage-$t$ training leaves task m's predictions and loss unchanged.

D Experimental Details

D.1 Datasets and Task Sequences

Question labels and task construction. We split MUSIC-AVQA into three groups using the modality component of the original question-type labels introduced by Li et al. (2022, Section 3.3 and supplementary Table 5). Each annotation records a modality category (Audio, Visual, or Audio-Visual) and a question type. Audio questions concern counting or comparing sounds. Visual questions concern counting or locating visible instruments. Audio–visual questions concern associations between instruments and their sounds through existential, counting, location, comparative, or temporal reasoning. These labels identify the information needed to answer each question. We retain each text question and answer and provide audio for A questions, video for V questions, and both for AV questions. These groups define three tasks with different input modalities and required reasoning, sharing the text input and answer-generation objective. Each task has two datasets with different source recordings and questions: A_1 and A_2 belong to the audio understanding task, V_1 and V_2 to the visual understanding task, and AV_1 and AV_2 to the joint audio–visual understanding task. At stage t, $\mathcal{D}_t$ is the designated dataset for one of these tasks. Thus, datasets with different samples can have the same task identity. In this protocol, task identities correspond one-to-one to modality combinations, so task-based protection has the same behavior as excluding the current combination's important set.

MUSIC-AVQA partitions and evaluation. The test set contains 9,129 unique questions: 1,611 A, 2,422 V, and 5,096 AV. All questions are evaluated before training and after every stage. Unparseable answers count as incorrect, and no question is dropped. Question categories are used only for offline physical-input construction. At runtime, modality-based activation uses the presence of audio and video. Every training, auxiliary, and test partition is source-video-disjoint. For ASAP, the six training subsets each contain 875 questions. Thus, the first-stage union has 1,750 samples and each later stage has 875, so each task contributes 1,750 samples per path.

Relation to AVQACL. AVQACL (Wu et al., 2025) constructs Split-AVQA and Split-MUSIC-AVQA by organizing tasks by question type and subtasks by object and sound categories, with audio, video, and question text supplied jointly. Our reconstruction instead varies the sensory inputs accompanying text across tasks and fine-tunes a pretrained MLLM. It therefore evaluates learning and retention under changing modality availability, complementing AVQACL's question-type and category-based protocol.

CMU-MOSI partitions and evaluation. CMU-MOSI (Zadeh et al., 2016) provides aligned transcripts, audio, and video for sentiment analysis. We partition training clips into three groups with no shared source videos or audio recordings and supply audio, video, or both alongside transcripts to form A, V, and AV tasks. Each training clip belongs to exactly one group, and clips from the same source video remain together. Each group's training data are split into two source-disjoint subsets, $\mathrm{A}_1, \mathrm{A}_2$, $\mathrm{V}_1, \mathrm{V}_2$, and $\mathrm{AV}_1, \mathrm{AV}_2$, following the six paths below. Within a path, each training clip appears in one stage and may be revisited across that stage's training epochs. Each clip has one transcript and one sentiment target. A fixed classification instruction requests Positive, Neutral, or Negative. Fine-tuning uses the answer-token loss in Section 3, and evaluation compares the generated label with the target label. During evaluation, the same test clips are presented under A, V, and AV input conditions, with transcripts, instructions, and targets unchanged. ASAP's auxiliary clips are excluded from its task training. Their selection and importance-estimation views are specified in Appendix D.2. Block selection is recomputed on these CMU-MOSI auxiliary clips, and the selected blocks are reused across stages. Thus, the sentiment experiment changes the prediction problem while retaining the selection, sharing, and protection procedure.

Task sequences. The two datasets of each task contain different samples. The initial task is trained on their union and is not subsequently trained again. Each path specifies $\mathcal{D}_1, \ldots, \mathcal{D}_K$ with $K = 5$, with the other two tasks recurring on their second datasets:

Path	Initial task	Training sequence
P_1	AV	$(\mathrm{AV}_1 \cup \mathrm{AV}_2) \to \mathrm{A}_1 \to \mathrm{V}_1 \to \mathrm{A}_2 \to \mathrm{V}_2$
P_2	AV	$(\mathrm{AV}_1 \cup \mathrm{AV}_2) \to \mathrm{V}_1 \to \mathrm{A}_1 \to \mathrm{V}_2 \to \mathrm{A}_2$
P_3	A	$(\mathrm{A}_1 \cup \mathrm{A}_2) \to \mathrm{V}_1 \to \mathrm{AV}_1 \to \mathrm{V}_2 \to \mathrm{AV}_2$
P_4	A	$(\mathrm{A}_1 \cup \mathrm{A}_2) \to \mathrm{AV}_1 \to \mathrm{V}_1 \to \mathrm{AV}_2 \to \mathrm{V}_2$
P_5	V	$(\mathrm{V}_1 \cup \mathrm{V}_2) \to \mathrm{AV}_1 \to \mathrm{A}_1 \to \mathrm{AV}_2 \to \mathrm{A}_2$
P_6	V	$(\mathrm{V}_1 \cup \mathrm{V}_2) \to \mathrm{A}_1 \to \mathrm{AV}_1 \to \mathrm{A}_2 \to \mathrm{AV}_2$

On recurrence, ASAP reuses the existing block sets and optimizer state without allocating new adaptation capacity. Previous task-training samples are not replayed.

D.2 Baselines and Implementation

Comparison methods. LoRA (Hu et al., 2022) represents weight updates as products of trainable low-rank matrices while freezing the pretrained weights. Our sequential baseline reuses one adapter across all tasks and continues optimizing it as new task data arrive.

PCLR (Meng et al., 2026) organizes LoRA rank vectors into a reusable rank pool. Its Compression–Integration–Learning pipeline compresses existing components, consolidates their knowledge through distillation, and allocates the released capacity to new tasks. Progressive compression balances parameter reuse with retention.

DGG (Li et al., 2026) approximates historical-task gradients using the displacement between current parameters and previously learned parameters. It combines this guidance with current-task gradients and uses Bernoulli sampling to regulate their contribution. We disable its replay component to follow our protocol without revisiting previous task-training samples.

SPU (Zhang et al., 2024) localizes adaptation to selected feed-forward layers and ranks individual weights by accumulated current-task gradient magnitude. It repeats selection at each stage using that stage's training data, without an earlier-task importance constraint. It updates the selected weights while freezing the remainder. SPU applies this selective-update principle to multimodal language-model fine-tuning.

OSFT (Nayak et al., 2026) decomposes weight matrices using singular value decomposition, preserves dominant singular components, and constrains new-task updates to the complementary subspace. It reconstructs the dense weights from the preserved and updated components without allocating a separate adapter for each task.

Training settings. The main experiments use Gemma-4-E2B (Gemma Team, 2026). The target total adaptation-parameter budget is approximately 12.1M parameters per method. ASAP, LoRA, PCLR, DGG, and OSFT target the query, key, value, and output projections in self-attention and the gate, up, and down projections in the language model's feed-forward networks. SPU localizes updates to the first MLP layer in each transformer block to refine input-pattern detectors (Zhang et al., 2024). We map this input-layer restriction to Gemma's gate and up projections, keeping the down and attention projections frozen. LoRA and ASAP each store 12,079,104 adaptation scalars across 205 language-model linear maps. All methods train for three epochs per stage using AdamW with a learning rate of 5×10^{-5}, BF16, microbatch size one, and no gradient accumulation. A cosine schedule with 3% warmup restarts at each stage, while optimizer state is preserved. The loss is computed on answer tokens, with logits materialized only at those positions. Task order and evaluation stages are shared across methods. The matched total adaptation budget is distinct from the active and trainable subsets at each stage.

Auxiliary data. For each benchmark, ASAP reserves approximately 1% of the original task-training samples as an auxiliary dataset, available before sequential fine-tuning and between stages. ASAP uses these samples for block selection and importance estimation without task-training updates. These samples were included in the training data of all comparison baselines and excluded from test evaluation for all methods. Prior LLM methods also use auxiliary data to prepare adaptation components (Tack et al., 2024; Li et al., 2025; Zhang et al., 2026b).

For MUSIC-AVQA, selection uses $R = 51$ samples, with 17 from each of the A, V, and AV tasks, and one predeclared covariance split of sizes 25 and 26. For CMU-MOSI, the auxiliary dataset contains 13 unique clips shared across all three tasks. Selection evaluates four controlled input conditions per clip, yielding 52 views and $R = 13$ observations per effect. The covariance split partitions the clips into groups of six and seven, keeping each clip's four views together. For importance estimation, the same clips provide 13 views for each of A, V, and AV, totaling 39 task-specific views, with $R_m = 13$ for every $m \in \mathcal{M}$. After each stage, we estimate importance on auxiliary samples for the learned task and refresh its important-block set. Selection and protection settings are summarized below.

Selection and protection settings. The following settings apply to the Gemma-4-E2B experiments.

Setting	MUSIC-AVQA	CMU-MOSI
Selection samples R	51 (17 per task)	13 unique clips
Covariance split	One, predeclared	One, predeclared
Half-sample sizes (n_1, n_2)	$(25, 26)$	$(6, 7)$
Block side length d	32	32
Eligible weight matrices	205	205
Per-matrix budget N_ν	61,440 parameters	61,440 parameters
Per-effect quota $N_\nu/(4d^2)$	15 blocks per matrix	15 blocks per matrix
Final block count $\lvert\mathcal{B}_e^\star\rvert$	2,949 per effect	2,949 per effect
Protection coverage η	0.9	0.9
Protection samples	Learned-task samples	13 views for task m_t

This selects $\sum_e |\mathcal{B}_e^\star| = 11{,}796$ blocks, containing $d^2 \sum_e |\mathcal{B}_e^\star| = 12{,}079{,}104$ adaptation parameters.

D.3 Evaluation Metrics

Continual learning metrics. Let $A_{p,t,m}$ be test accuracy on task m after stage t of path p, and let $m_{p,t}$ be the task trained at that stage. The three tasks are A, V, and AV on both MUSIC-AVQA and CMU-MOSI, with text always present. For *learning plasticity*, $\mathrm{New.ACC}_p = K^{-1} \sum_{t=1}^{K} A_{p,t,m_{p,t}}$ averages performance immediately after learning from each stage dataset, with $K = 5$. It covers both initial task learning and learning fresh samples when a task recurs, measuring attained current-task performance at stage completion.

For *memory stability*, let $t^{\text{first}}_{p,m}$ be the first stage that trains task m, and let $\mathcal{H}_{p,t}$ contain the tasks learned before stage t, excluding $m_{p,t}$. The average stage-wise forgetting is

$$\text{Avg.FOR}_p = \frac{1}{K-1}\sum_{t=2}^{K}\frac{1}{|\mathcal{H}_{p,t}|}\sum_{m\in\mathcal{H}_{p,t}}\left(\max_{t^{\text{first}}_{p,m}\le t''\le t} A_{p,t'',m} - A_{p,t,m}\right). \tag{32}$$

Here t'' indexes evaluation stages since the task was first learned. The maximum includes stage t, so an improvement beyond the previous best gives zero forgetting. Tasks are averaged within each stage before stages are averaged. This adapts peak-to-current forgetting to recurring tasks and records deterioration before later relearning can repair it. For example, in P_1, the evaluation after V_1 includes forgetting on both AV and A, even though A_2 will subsequently train A again.

For final performance, $\text{Final.ACC}_p = \frac{1}{3}\sum_m A_{p,K,m}$ averages final test accuracy across all three conditions. Equal task weights prevent the larger MUSIC-AVQA AV group from dominating. For CMU-MOSI, the average is over the three input conditions. Evaluation uses audio and video sources absent from training.

Aggregation. The headline tables report $\frac{1}{6}\sum_{p=1}^{6} M_p$ for each path-level metric M_p, calculated independently for each dataset. Per-path results appear in Appendix D.4. Paired-order diagnostics average P_1 and P_2, P_3 and P_4, or P_5 and P_6. We compute forgetting from each path's accuracy trajectory before averaging. Accuracy values are percentages, forgetting is in percentage points, and latency is in milliseconds per sample.

Efficiency metrics. For computational efficiency comparisons, we report peak allocated GPU memory and forward-pass latency. Peak allocated memory measures live GPU tensor storage rather than allocator-reserved memory. Forward-pass latency is measured per sample under the protocol in Appendix D.7.

Initial-task diagnostics. The initial-task analyses retain three narrower quantities: $\text{Acq.}_p = A_{p,1,m_{p,1}}$, $\text{Final}_p = A_{p,K,m_{p,1}}$, and $\text{FOR}_p = \max_{1\le t\le K} A_{p,t,m_{p,1}} - A_{p,K,m_{p,1}}$. They describe acquisition and peak-to-final retention of the task that is not revisited. We label the last quantity Initial-task FOR to distinguish it from the all-task, stage-wise Avg.FOR in the headline and component tables.

D.4 CONTINUAL LEARNING RESULTS WITH GEMMA-4-E2B

Tables 4 and 5 report the Gemma-4-E2B results underlying Table 1. Each column corresponds to one complete task order.

Table 4: **MUSIC-AVQA results with Gemma-4-E2B by task order.** Accuracy: %; Avg.FOR: percentage points.

Method	Metric	P_1	P_2	P_3	P_4	P_5	P_6
LoRA	New.ACC	69.21	69.34	70.09	69.09	67.25	66.70
	Avg.FOR	3.69	3.54	7.52	3.82	5.43	6.69
	Final.ACC	65.44	66.23	62.83	66.09	64.37	62.56
PCLR	New.ACC	68.26	68.87	68.73	68.34	66.89	67.20
	Avg.FOR	1.47	1.25	3.72	3.60	1.57	2.00
	Final.ACC	67.02	67.55	66.36	66.28	68.82	68.75
DGG	New.ACC	67.53	68.06	68.43	67.80	66.58	65.90
	Avg.FOR	1.36	0.65	2.99	3.25	3.33	1.44
	Final.ACC	66.24	68.21	67.12	66.05	68.09	66.44
SPU	New.ACC	63.37	61.59	66.29	66.40	63.05	62.94
	Avg.FOR	17.10	22.96	11.53	6.50	22.08	19.39
	Final.ACC	55.88	51.89	58.89	59.18	51.33	53.71
OSFT	New.ACC	68.40	69.31	68.64	69.36	66.87	66.73
	Avg.FOR	1.27	1.42	2.00	2.97	2.57	2.83
	Final.ACC	67.65	66.76	66.20	67.93	67.92	67.49
ASAP	New.ACC	69.96	70.46	70.50	70.12	67.81	67.98
	Avg.FOR	0.88	0.18	1.82	2.24	0.61	0.72
	Final.ACC	69.54	69.70	68.79	68.73	70.03	69.96

Table 5: **CMU-MOSI results with Gemma-4-E2B by task order.** Accuracy: %; Avg.FOR: percentage points.

Method	Metric	P_1	P_2	P_3	P_4	P_5	P_6
LoRA	New.ACC	80.35	80.41	79.19	79.33	79.83	80.50
	Avg.FOR	0.33	0.67	0.18	0.13	0.53	0.69
	Final.ACC	80.28	79.22	79.70	80.24	78.73	80.87
PCLR	New.ACC	80.47	80.21	79.89	79.83	80.70	80.67
	Avg.FOR	0.56	0.56	0.09	0.44	0.33	0.16
	Final.ACC	80.09	79.22	80.19	80.14	80.53	80.53
DGG	New.ACC	80.79	80.15	79.62	80.15	80.88	80.76
	Avg.FOR	0.55	0.42	0.20	0.18	0.78	0.73
	Final.ACC	80.24	80.53	80.58	80.48	79.75	80.67
SPU	New.ACC	75.08	67.77	77.99	75.60	77.99	74.58
	Avg.FOR	5.85	15.94	4.90	7.18	6.09	6.74
	Final.ACC	72.49	70.30	74.49	74.54	76.38	74.93
OSFT	New.ACC	80.00	79.71	78.87	79.36	80.15	80.00
	Avg.FOR	0.27	0.29	0.04	0.00	0.31	0.11
	Final.ACC	80.38	79.85	79.90	80.28	80.48	80.53
ASAP	New.ACC	82.42	82.86	82.83	82.77	82.83	82.62
	Avg.FOR	0.18	0.31	0.05	0.51	0.27	0.16
	Final.ACC	82.46	82.56	83.43	82.75	82.90	82.80

Initial-task retention across stages. Tables 6 and 7 give the accuracies plotted in Figure 3. Each path's initial task is evaluated after every stage without relearning. The within-path accuracy range is its maximum minus minimum accuracy over stages 1–5. A single-stage drop is the decrease between consecutive stages, or zero if accuracy improves. Maximum drops are taken over all four transitions and all six paths for each method. Changes are computed before rounding.

Table 6: **Initial-task accuracy across stages on MUSIC-AVQA with Gemma-4-E2B.** Values plotted in Figure 3. Initial tasks: AV (P_1–P_2), A (P_3–P_4), and V (P_5–P_6). Accuracy: %. Bold: best within each path and stage.

Method	Path	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
LoRA	P_1	62.62	57.83	59.81	58.44	57.95
	P_2	62.97	59.13	56.22	58.36	56.36
	P_3	62.80	53.49	55.42	52.25	45.05
	P_4	62.43	56.72	56.72	57.65	55.98
	P_5	**79.46**	77.81	77.64	63.52	66.50
	P_6	79.34	75.17	70.30	72.07	65.13
PCLR	P_1	62.38	60.97	63.50	62.32	62.66
	P_2	63.31	63.03	62.56	63.38	62.72
	P_3	62.43	57.34	57.22	56.66	55.48
	P_4	62.49	56.47	57.40	57.09	57.34
	P_5	78.59	77.44	76.65	77.98	77.77
	P_6	79.13	77.03	75.87	75.58	77.60
DGG	P_1	62.66	62.23	63.52	62.09	61.64
	P_2	62.48	63.84	63.25	63.33	63.21
	P_3	62.87	58.71	57.53	57.34	58.71
	P_4	62.25	56.85	57.59	57.40	57.53
	P_5	79.26	77.31	76.49	71.33	76.57
	P_6	78.51	78.47	78.10	78.02	77.77
SPU	P_1	63.27	37.15	42.88	36.54	45.29
	P_2	62.09	33.46	32.09	36.27	35.42
	P_3	62.74	49.71	52.13	50.08	45.42
	P_4	62.31	58.40	56.60	53.43	46.54
	P_5	78.88	55.23	47.50	45.77	42.96
	P_6	79.34	62.16	51.72	42.43	45.03
OSFT	P_1	61.89	60.36	62.54	60.28	61.74
	P_2	62.13	62.50	61.28	60.76	59.62
	P_3	63.74	61.81	61.19	**62.80**	58.77
	P_4	63.36	58.71	58.52	59.14	58.83
	P_5	78.02	76.08	75.13	74.88	74.71
	P_6	78.72	75.42	74.67	74.80	75.91
ASAP	P_1	**65.13**	**64.07**	**65.01**	**64.19**	**64.91**
	P_2	**64.29**	**64.15**	**64.54**	**64.21**	**64.40**
	P_3	**65.36**	**64.00**	**61.70**	62.45	**60.27**
	P_4	**64.74**	**61.51**	**60.89**	**60.83**	**61.27**
	P_5	79.40	**79.56**	**79.52**	**80.35**	**79.56**
	P_6	**79.69**	**79.73**	**79.69**	**79.56**	**79.89**

Table 7: **Initial-task accuracy across stages on CMU-MOSI with Gemma-4-E2B.** Values plotted in Figure 3. Initial tasks: AV (P_1–P_2), A (P_3–P_4), and V (P_5–P_6). Accuracy: %. Bold: best within each path and stage.

Method	Path	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
LoRA	P_1	78.34	77.61	78.78	78.05	78.63
	P_2	78.78	79.22	78.49	78.34	77.32
	P_3	80.24	80.97	81.11	82.28	81.26
	P_4	79.80	80.53	81.55	82.13	82.71
	P_5	80.09	80.09	80.53	81.26	79.07
	P_6	80.82	81.69	80.53	79.94	80.67
PCLR	P_1	79.36	79.07	78.20	78.05	78.34
	P_2	79.36	79.36	78.78	78.49	77.90
	P_3	80.67	80.53	80.24	80.82	80.82
	P_4	81.26	81.26	80.82	82.28	81.84
	P_5	80.67	80.24	80.67	80.38	79.80
	P_6	79.80	79.80	80.24	80.53	80.09
DGG	P_1	77.90	77.90	78.78	77.32	77.90
	P_2	77.61	79.07	78.05	78.20	78.63
	P_3	80.82	80.97	81.99	81.55	82.57
	P_4	80.67	80.67	80.67	81.84	81.11
	P_5	80.53	81.11	81.84	80.67	79.36
	P_6	80.53	81.69	81.26	80.38	80.53
SPU	P_1	80.22	68.99	72.20	78.03	70.45
	P_2	80.51	38.67	76.57	67.53	70.30
	P_3	**84.69**	80.47	80.61	76.09	77.11
	P_4	84.11	80.47	67.64	75.66	74.78
	P_5	83.09	74.49	72.01	75.66	77.70
	P_6	82.65	70.26	71.72	73.32	73.76
OSFT	P_1	78.05	77.76	77.61	77.32	78.49
	P_2	77.17	77.03	78.63	77.61	78.20
	P_3	81.26	81.40	81.40	81.11	81.69
	P_4	80.67	81.40	81.69	81.69	81.99
	P_5	79.94	80.53	81.11	80.24	80.09
	P_6	79.65	79.94	80.82	81.69	81.11
ASAP	P_1	**80.32**	**81.34**	**81.05**	**80.61**	**81.20**
	P_2	**81.05**	**80.61**	**80.61**	**81.78**	**81.05**
	P_3	84.26	**84.26**	**84.11**	**84.40**	**84.11**
	P_4	**84.69**	**84.11**	**83.97**	**84.26**	**83.67**
	P_5	**83.82**	**83.38**	**83.82**	**83.97**	**83.24**
	P_6	**83.53**	**83.67**	**83.53**	**82.80**	**83.53**

D.5 CMU-MOSI WITH QWEN2.5-OMNI-3B

Tables 8 and 9 report results with Qwen2.5-Omni-3B (Xu et al., 2025) across the six task orders. ASAP achieves 85.00% Final.ACC, 85.12% New.ACC, and 0.28 Avg.FOR, leading the compared methods on all three six-path means.

Table 8: **CMU-MOSI results with Qwen2.5-Omni-3B.** Results are averaged over all six task orders. A, V, and AV report final task accuracy, and Final.ACC is their mean. Accuracy: %; Avg.FOR: percentage points. Bold: best; underline: second best.

Method	A	V	AV	Final.ACC↑	Avg.FOR↓	New.ACC↑
LoRA	83.87	83.83	<u>83.92</u>	83.87	0.65	83.69
PCLR	<u>84.48</u>	83.95	<u>83.92</u>	<u>84.12</u>	<u>0.39</u>	83.84
DGG	84.02	<u>84.04</u>	83.68	83.91	0.52	<u>83.98</u>
SPU	72.57	61.37	73.57	69.17	8.33	73.75
OSFT	83.56	83.07	83.68	83.44	0.45	83.56
ASAP	**84.52**	**85.06**	**85.42**	**85.00**	**0.28**	**85.12**

Table 9: **Qwen2.5-Omni-3B results by task order on CMU-MOSI.** Units and highlighting follow Table 8. Comparisons are across methods within each metric and path. Tied results share the same highlighting.

Method	Metric	P_1	P_2	P_3	P_4	P_5	P_6
LoRA	Final.ACC	83.73	83.68	83.97	<u>83.87</u>	83.97	84.02
	Avg.FOR	0.56	1.00	0.64	<u>0.27</u>	0.56	<u>0.84</u>
	New.ACC	83.60	83.57	83.75	83.69	83.75	83.81
PCLR	Final.ACC	<u>84.41</u>	<u>83.87</u>	<u>84.07</u>	<u>83.87</u>	<u>84.07</u>	<u>84.41</u>
	Avg.FOR	0.56	0.53	<u>0.22</u>	**0.02**	0.60	**0.42**
	New.ACC	84.01	83.69	83.84	83.37	84.04	<u>84.07</u>
DGG	Final.ACC	<u>84.41</u>	<u>83.87</u>	83.83	83.34	<u>84.07</u>	83.97
	Avg.FOR	0.49	0.93	0.51	0.29	<u>0.46</u>	**0.42**
	New.ACC	<u>84.13</u>	<u>83.86</u>	<u>84.07</u>	<u>83.75</u>	<u>84.19</u>	83.89
SPU	Final.ACC	65.60	72.35	43.25	72.64	79.20	81.97
	Avg.FOR	11.52	9.64	20.92	2.44	4.14	1.35
	New.ACC	67.11	74.17	64.66	74.08	80.79	81.69
OSFT	Final.ACC	83.63	83.29	83.58	83.24	82.76	84.12
	Avg.FOR	**0.38**	<u>0.49</u>	0.55	0.40	<u>0.46</u>	**0.42**
	New.ACC	83.75	83.28	83.98	83.40	83.49	83.46
ASAP	Final.ACC	**84.94**	**84.84**	**85.13**	**85.18**	**84.89**	**85.03**
	Avg.FOR	<u>0.40</u>	**0.47**	**0.13**	**0.02**	**0.24**	**0.42**
	New.ACC	**85.04**	**85.01**	**85.31**	**85.04**	**85.19**	**85.13**

Initial-task retention across stages. Figure 5 and Table 10 follow the same initial-task evaluation as Figure 3. ASAP leads at all five stages on every path. Its maximum single-stage drop is 1.02 percentage points, compared with 1.31 for PCLR, the strongest baseline on this measure.

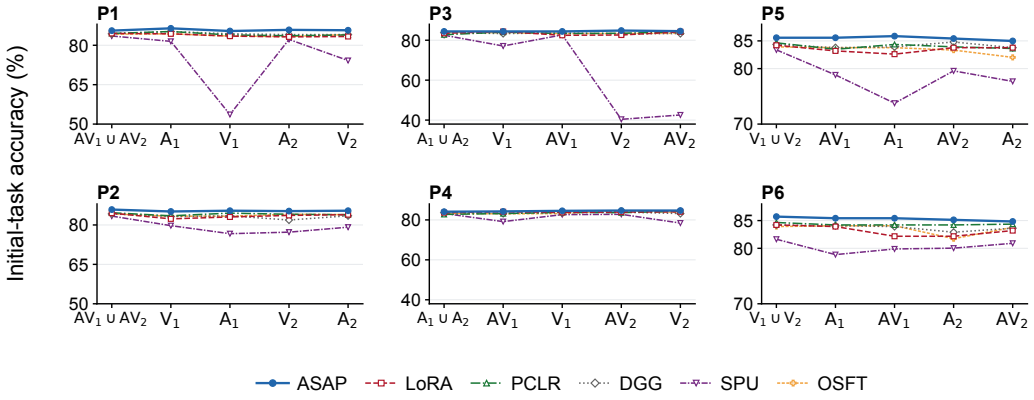


Figure 5: **Initial-task retention on CMU-MOSI with Qwen2.5-Omni-3B.** Each point reports initial-task accuracy after training on the dataset labeled on the x-axis, following Figure 3. Table 10 gives the plotted values.

Table 10: **Initial-task accuracy across stages on CMU-MOSI with Qwen2.5-Omni-3B.** Values plotted in Figure 5. Initial tasks: AV (P_1–P_2), A (P_3–P_4), and V (P_5–P_6). Accuracy: %. Bold: best within each path and stage.

Method	Path	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
LoRA	P_1	84.51	84.36	83.49	83.19	83.34
	P_2	84.51	82.32	83.05	83.63	83.92
	P_3	83.49	84.36	82.47	82.61	84.21
	P_4	83.63	84.07	83.78	83.92	84.07
	P_5	84.21	83.19	82.61	83.78	83.78
	P_6	84.21	83.92	82.17	82.17	83.19
PCLR	P_1	84.51	85.23	83.92	83.63	83.92
	P_2	84.65	83.49	84.51	84.07	83.92
	P_3	82.76	84.07	83.34	83.49	84.21
	P_4	82.76	83.19	83.78	83.63	84.36
	P_5	84.65	83.49	84.36	83.92	83.63
	P_6	84.65	84.21	84.21	84.21	84.36
DGG	P_1	84.36	85.23	84.36	84.07	84.07
	P_2	84.21	83.19	83.49	81.88	83.49
	P_3	83.78	83.34	83.63	83.19	83.34
	P_4	83.63	83.19	83.78	83.92	83.19
	P_5	84.36	83.78	83.92	84.80	83.78
	P_6	84.21	84.07	83.92	82.90	83.63
SPU	P_1	83.53	81.49	53.64	82.22	74.20
	P_2	83.38	79.74	76.68	77.26	79.15
	P_3	82.51	77.11	82.65	40.38	42.57
	P_4	83.24	79.15	82.65	82.80	78.43
	P_5	83.38	78.86	73.76	79.59	77.70
	P_6	81.63	78.86	79.88	80.03	80.90
OSFT	P_1	84.21	84.36	83.49	83.34	84.07
	P_2	84.21	83.34	83.34	84.07	83.92
	P_3	83.92	83.78	83.34	83.34	83.49
	P_4	83.34	83.05	83.19	84.65	83.78
	P_5	83.92	83.78	83.78	83.34	82.03
	P_6	83.92	84.07	84.07	81.74	83.78
ASAP	P_1	**85.57**	**86.44**	**85.42**	**85.86**	**85.71**
	P_2	**85.86**	**85.13**	**85.42**	**85.28**	**85.42**
	P_3	**84.40**	**84.40**	**84.40**	**84.84**	**84.55**
	P_4	**84.11**	**84.26**	**84.55**	**84.69**	**84.69**
	P_5	**85.57**	**85.57**	**85.86**	**85.42**	**84.99**
	P_6	**85.71**	**85.42**	**85.42**	**85.13**	**84.84**

D.6 ASAP Component Ablations

Variant definitions. Without protection removes the stagewise freezing of blocks important to earlier tasks. Random selection replaces standardized block ranking with random selection. The Frozen shared blocks variant freezes the shared block set after initial training and keeps it frozen throughout the remaining stages. The variants retain modality-based activation and the target parameter budget.

Initial-task comparisons. Table 11 expands the six-path summary in Table 2 with initial-task diagnostics. Each initial-task group averages its two task orders, whereas the overall row averages all six paths. Compared with the Without protection variant, ASAP improves New.ACC by 0.52, 1.67, 2.07, 0.93, 0.84, and 1.36 percentage points on P_1–P_6, respectively. The full method improves final initial-task accuracy over all three variants for each initial task. On V-first paths, ASAP has the highest acquisition (79.54%) and final accuracy (79.73%). Its 0.39-point Initial-task FOR measures peak-to-final declines despite an acquisition-to-final gain on average.

Table 11: **MUSIC-AVQA component ablations by initial task.** Each initial-task group averages its two orders, whereas Overall averages six paths. Accuracy: %; Initial-task FOR: percentage points. Bold: best; underline: second best within each path group.

Paths	Variant	Final.ACC ↑	New.ACC ↑	Acq. ↑	Final ↑	Initial-task FOR ↓
AV-first	Without protection	66.08	<u>69.12</u>	<u>62.96</u>	57.03	5.94
	Random selection	<u>67.39</u>	67.79	62.55	62.00	0.72
	Frozen shared blocks	66.73	66.78	62.58	<u>62.19</u>	<u>0.52</u>
	ASAP	**69.62**	**70.21**	**64.71**	**64.66**	**0.18**
A-first	Without protection	65.70	<u>68.81</u>	62.53	54.02	8.50
	Random selection	<u>66.15</u>	68.43	<u>62.90</u>	55.98	6.92
	Frozen shared blocks	65.94	66.92	62.84	<u>59.55</u>	**3.29**
	ASAP	**68.76**	**70.31**	**65.05**	**60.77**	<u>4.28</u>
V-first	Without protection	67.58	<u>66.80</u>	<u>79.32</u>	77.29	2.02
	Random selection	<u>67.70</u>	65.64	77.58	<u>78.41</u>	<u>0.10</u>
	Frozen shared blocks	67.46	65.35	76.82	77.44	**0.00**
	ASAP	**69.99**	**67.90**	**79.54**	**79.73**	0.39
Overall	Without protection	66.45	<u>68.24</u>	<u>68.27</u>	62.78	5.49
	Random selection	<u>67.08</u>	67.29	67.68	65.46	2.58
	Frozen shared blocks	66.71	66.35	67.41	<u>66.39</u>	**1.27**
	ASAP	**69.46**	**69.47**	**69.77**	**68.39**	<u>1.62</u>

D.7 Efficiency Comparison

The efficiency comparison in Table 3 covers LoRA, PCLR, DGG, and ASAP. Memory is peak allocated GPU tensor storage in GiB, excluding unused memory reserved by the allocator.

Table 3 reports forward-pass latency with answer-token loss evaluation on an NVIDIA H100 80GB GPU in BF16, with batch size one. The aggregate averages the A, V, and AV median latencies equally. Inputs are prepared on the GPU. Preprocessing, device transfer, and weight merging are excluded. This benchmark does not time autoregressive generation. LoRA, DGG, and ASAP are timed after merging their updates into pretrained weights, whereas PCLR retains its native forward implementation. These microbenchmarks are separate from the accuracy evaluations.

Adaptation-state accounting. With the selected block sets fixed, ASAP stores N_{total} learned update values and at most $2N_{\text{total}}$ Adam moment scalars, reusing them across stages. These counts are independent of the sequence length K. Protection maintains one important-block record per distinct task, with three records in our protocol. Base-model weights, activations, and temporary buffers contribute separately to peak GPU memory. Table 3 reports measured peak GPU memory and forward-pass latency.

D.8 Analysis of Selective Protection

Importance coverage. Figure 4(a) is a post-training analysis of the model after initial AV training on P1. Using saved per-block mean squared loss gradients from AV auxiliary samples, we rank all 11,796 active blocks by Equation 6 and compute the fraction of total importance captured at each block count. The top 2,081 blocks (17.64%) capture 90.0041% of total importance, compared with 89.9971% for the top 2,080. Thus, 2,081 is the smallest count reaching 90% coverage for these scores. Random curves use 200 uniformly permuted orderings of the same blocks. The dashed curve is their mean coverage, and shading shows the pointwise 5th–95th percentiles. These percentiles describe variation across random block selections. The analysis quantifies importance concentration and the coverage achieved by ranking, complementing the loss-retention measurements.

Trainable and protected capacity. Figure 6 reports all thirty stage-start block states. Learned updates at trainable blocks participate in the forward pass and may change. Protected updates also participate but remain frozen. Inactive updates are excluded and frozen. Each bar uses all selected blocks as its denominator. Among active blocks at later stages, protection freezes 5.2–23.3%, leaving 76.7–94.8% trainable. These proportions quantify the block capacity available for subsequent learning.

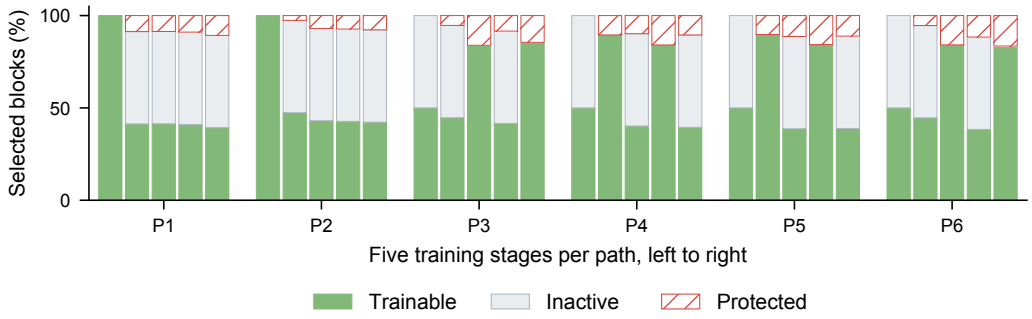


Figure 6: **Block states across all six paths.** Five bars per path show successive stages. Fractions use all selected blocks.

Full-test-set initial-task loss evaluation. Figure 7 compares ASAP and LoRA after each of the five training stages on all six paths, using each method's model after initial-task training as the reference. Each path evaluates the same full initial-task test set throughout. Across both methods and six paths this gives 60 model evaluations and 182,580 sample evaluations with zero drops. Answer loss includes the end-of-sequence token, averages over valid answer tokens within each question, and then weights questions equally. The paired curves use each method's own acquisition loss. No model parameters change during evaluation. The bars in Figure 4(b) compare final-minus-acquisition loss for ASAP, the Without protection variant, and LoRA using the final endpoint summaries. The equal-path mean gives each of the six final-minus-acquisition differences the same weight.

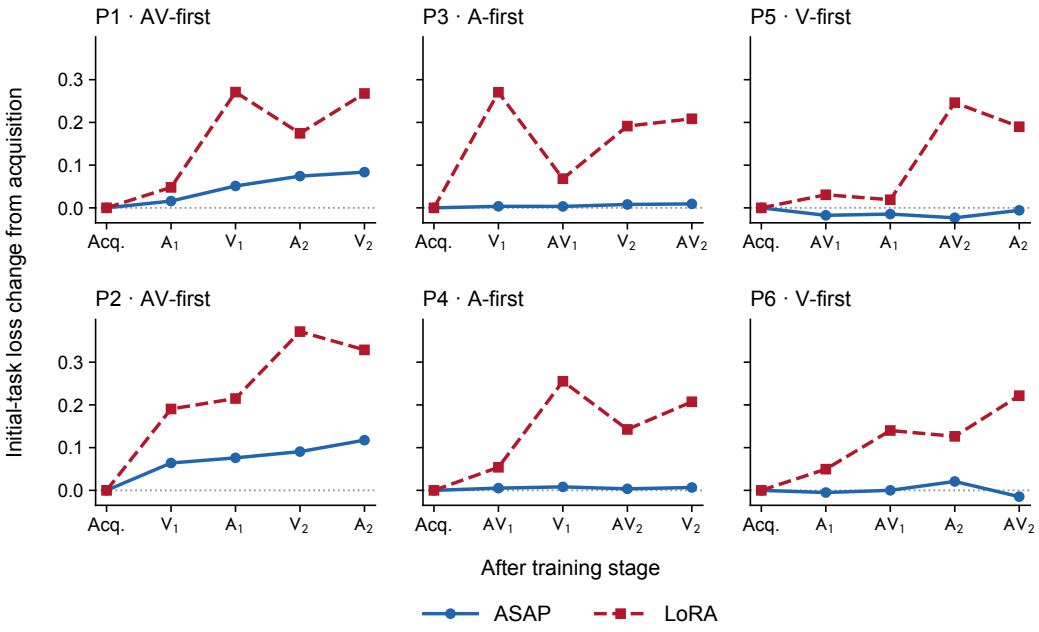


Figure 7: **Initial-task loss trajectories on the full test set.** Loss changes are relative to each method's model after initial-task training. Columns pair paths with the same initial task, and all panels share a vertical scale. Loss is computed using natural logarithms.