Concentrated Gradients Amplify Forgetting: Dominant-direction Projection for Continual Multimodal Learning

Chengxiang Huang[1*] **Haopeng Zhang**[1] **Yuzhe Han**[2] **Rui Dai**[1†]
Kaikui Liu[1] **Xiangxiang Chu**[1‡] **Piotr Koniusz**[3,4‡]
[1]Amap Alibaba [2]University of Tübingen
[3]University of New South Wales [4]CSIRO

Abstract

Multimodal Large Language Models (MLLMs) trained on sequential tasks suffer from catastrophic forgetting. We study this problem in LoRA-based multimodal continual instruction tuning. By analyzing the diagonal Fisher information of LoRA parameters, we find that forgetting is amplified not merely by shared high-sensitivity parameters across tasks, but by the concentration of updates on those shared parameters, *i.e.*, some tasks distribute their change uniformly across parameters, while others pack most of it into a narrow subset, causing disproportionate interference on the parameters that prior tasks depend on. We further observe that this concentration manifests in the dominant directions of recent gradients. Based on this insight, we propose **D**om**i**nant-direction **G**radient **Pro**jection (**DiGPro**), which maintains a short buffer of recent gradient directions, extracts their dominant subspace with adaptive rank selection, and attenuates the parallel component for each optimizer step. Crucially, DiGPro requires no prior-task data or stored subspaces, operating on the current gradient history. Experiments on cross-modal and vision-centric continual learning benchmarks show DiGPro consistently reduces forgetting and attains competitive final results. Project repository: https://github.com/AMAP-ML/DiGPro.

1 Introduction

Multimodal large language models (MLLMs) have been applied to a range of downstream tasks [1–6]. However, catastrophic forgetting remains a central challenge in continual learning, as sequentially acquiring new tasks can degrade performance on previously learned tasks [7, 8]. Parameter-efficient fine-tuning methods such as LoRA [9] reduce the number of trainable parameters and thereby limit the potential for interference [10], yet sequential LoRA fine-tuning still leads to significant forgetting in practice.

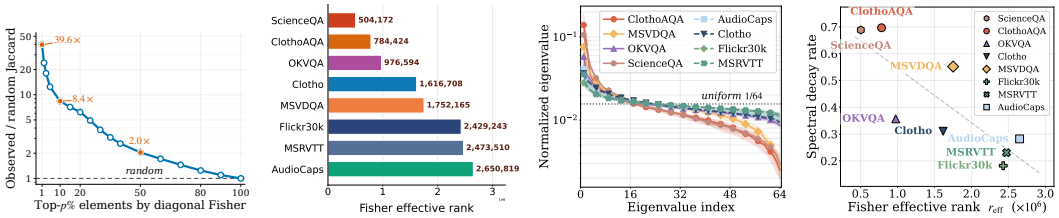
A standard second-order Taylor approximation expresses the increase in old-task loss around the parameter vector $\boldsymbol{\theta}^*$ attained after training the old task as [11, 12] (see Appendix A):

$$\Delta\mathcal{L}^{\text{old}} \approx \tfrac{1}{2}\sum_i \underbrace{F_i^{\text{old}}}_{\text{old-task sensitivity}} \cdot \underbrace{\Delta\theta_i^2}_{\text{new-task update}}, \tag{1}$$

where $F_i^{\text{old}} = \mathbb{E}\big[(\partial\mathcal{L}^{\text{old}}/\partial\theta_i)^2\big]$ is the *diagonal Fisher information* of the old task, measuring its sensitivity to perturbations at the i^{th} parameter θ_i [13], and $\Delta\theta_i$ is the parameter change induced by

*Work done as an intern at Alibaba.
†Project leader.
‡Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Parameter-level overlap. (b) Update concentration. (c) Gradient Gram spectrum. (d) Concentration *vs*. spectral decay.

Figure 1: **(a)** Ratio of pairwise Jaccard overlap to random expectation for top-$p\%$ highest Fisher parameters (8 tasks). **(b)** Fisher Effective rank r_{eff} per task. **(c)** Normalized Gram eigenvalue spectrum of recent gradient directions. **(d)** Fisher effective rank *vs*. spectral decay rate ξ. (See Appendix C.)

new-task training. Eq. (1) reveals that forgetting on each parameter is determined by the product of two factors: (i) the old task's sensitivity at the i^{th} parameter and (ii) the square of the new task's change at the i^{th} parameter. Thus, even if two tasks modify the same set of parameters, the severity of forgetting depends on *how* the new task distributes its updates – concentrating large changes on the parameters where the old task is sensitive amplifies the product $F_i^{\text{old}} \cdot \Delta\theta_i^2$, causing large interference.

To understand how these two factors interact in practice, we analyze the diagonal Fisher information of LoRA parameters at the individual parameter level across eight tasks spanning three modalities (audio, image, video). Two patterns emerge as detailed next.

Firstly, the individual high-sensitivity parameters are heavily shared by tasks. We quantify this by measuring, for each pair of tasks, the Jaccard overlap [14] of their top-$p\%$ highest Fisher parameters, and compare it to the overlap as if Fisher values were randomly distributed across parameters. Figure 1a shows the ratio of observed to random pairwise Jaccard overlap: from $40\times$ at $p{=}1\%$ down to $2\times$ at $p{=}50\%$, showing that the most sensitive parameters are shared far beyond chance (see Appendix B). **Secondly**, despite sharing the same high-sensitivity parameters, tasks differ sharply in how they distribute updates across those parameters. We quantify this via the *Fisher effective rank* [15]: lower values indicate stronger concentration on fewer parameters. Figure 1b shows that the effective rank varies by over $5\times$ across tasks: some spread their update energy uniformly across parameters, while others concentrate most of it in a narrow subset of the shared high-sensitivity parameters.

The above two patterns imply that when a new task concentrates its updates, those updates tend to land on the same high-sensitivity parameters that prior tasks depend on, amplifying the product $F_i^{\text{old}} \cdot \Delta\theta_i^2$ on the parameters most critical to earlier tasks.

Since the diagonal Fisher coeff. i is defined as $\mathbb{E}[g_i^2]$, its concentration directly shapes the geometry of gradient directions, a connection we formalize in Proposition 1. As a result, concentrated updates leave a detectable signature in the dominant directions of recent gradients, without requiring per-task Fisher computation. Figure 1c confirms this: tasks with concentrated updates show rapid spectral decay in their gradient Gram matrices, while tasks with more uniform updates show flatter spectra. Moreover, lower effective Fisher rank overall corresponds to steeper spectral decay (Figure 1d).

Leveraging the above points, we propose **D**omi**n**ant-direction **G**radient **Pro**jection (**DiGPro**) to detect and suppress concentrated update directions. DiGPro maintains a short ring buffer of recent gradient directions (gradient stream) and incrementally tracks their Gram matrix. Eigendecomposition of this small Gram matrix recovers the dominant subspace, whose rank is selected adaptively based on the cumulative eigenvalue energy. The current gradient is decomposed into components parallel and orthogonal to this subspace; the parallel component is attenuated before the optimizer step, while the orthogonal component passes through unchanged. Both the rank and attenuation strength adapt automatically, intervening strongly during concentrated phases and weakly during uniform phases. Unlike most multimodal continual learning methods that mitigate forgetting by retaining past-task data or stored subspaces, DiGPro uses no information from prior tasks. Our contributions are:

i. We identify through empirical analysis of diagonal Fisher information that, in LoRA-based multimodal continual instruction tuning, catastrophic forgetting is amplified by the combination of shared high-sensitivity parameters and concentrated new-task updates.

ii. We show that this concentration manifests in the dominant directions of recent gradients, making it detectable without per-task Fisher information.

iii. We propose DiGPro, a gradient projection method that detects and suppresses recurring dominant directions from the recent gradient stream, requiring no past-task storage, and demonstrate its effectiveness on cross-modal and `MLLM-DCL` benchmarks [16].

2 Related Work

Multimodal continual learning. Continual learning methods can be broadly categorized into three families [17]: (i) regularization-based methods that penalize changes to important parameters [11, 18–20], (ii) replay-based methods that store or generate past-task exemplars [21–23], and (iii) architecture-based methods that allocate separate capacity per task [24–26]. Early work on multimodal continual learning largely focused on CLIP-style [27] dual-tower architectures [28], while more recent work has explored continual instruction tuning [29–33] within the broader area of multimodal model adaptation [34, 35]. LoRA-specific methods impose structural constraints on adapter spaces to reduce forgetting [36–41], while Biderman et al. [10] show that LoRA forgets less than full fine-tuning. The diagonal of Fisher Information Matrix (FIM), central to continual learning since EWC [11, 13], is typically leveraged for protecting important parameters, and requires access to past data for FIM estimation. Our approach differs by intervening directly on gradient *directions* of the current buffer during training, without accessing past-task data.

Gradient projection methods. This line of work projects gradients to avoid interference with prior tasks, including OGD [42], GPM [43], FSDGPM [44], TRGP [45], SGP [46], PGP [47], and ZeroFlow [48]. PCGrad [49] and CAGrad [50] address gradient conflicts in multi-task settings. A common limitation of these methods is that they store and access *past-task gradient information*, which grows with the number of tasks. Our method instead captures dominant directions from the *current* gradient stream, requiring no past-task storage.

3 Preliminaries

Multimodal LLMs with LoRA. We consider Multimodal Large Language Models (MLLMs) that combine a frozen LLM backbone with modality-specific encoders (*e.g.*, `CLIP` [27] for vision, `BEATs` [51] for audio) and learnable projectors that map encoder outputs into the LLM's input space. During fine-tuning, the LLM backbone and encoders remain frozen; the projectors and LoRA [9] adapters attached to the LLM's attention layers are trained. Each LoRA adapter reparameterizes a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{m\times n}$ as

$$\mathbf{W} = \mathbf{W}_0 + \mathbf{B}\mathbf{A}, \tag{2}$$

where $\mathbf{B} \in \mathbb{R}^{m\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times n}$ are low-rank factors of rank $r \ll \min(m, n)$. We denote by $\boldsymbol{\theta} \in \mathbb{R}^d$ the concatenation of all trainable LoRA parameters (d in total).

Continual learning setup. The model is trained on a sequence of T tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$, where each task may involve a different modality (audio, image, or video) and a different task type (captioning or question answering). All tasks share the same set of LoRA parameters $\boldsymbol{\theta}$; at each task $\mathcal{T}_t$, the model is trained for one epoch on the task's training data. At training step t, we denote the mini-batch loss by $\mathcal{L}_t$ and the gradient by $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}}\mathcal{L}_t \in \mathbb{R}^d$.

4 Method

Section 1 showed that forgetting is amplified when updates concentrate on shared high-sensitivity parameters, and that this concentration leaves a detectable signature in the gradient stream. As shown in Figure 2, DiGPro exploits this signature to extract and suppress dominant gradient directions before each optimizer step, without requiring any past-task information.

The following proposition formalizes the link between diagonal Fisher concentration and concentration in normalized gradient directions, motivating why concentration can be detected from the gradient stream without directly computing Fisher information.

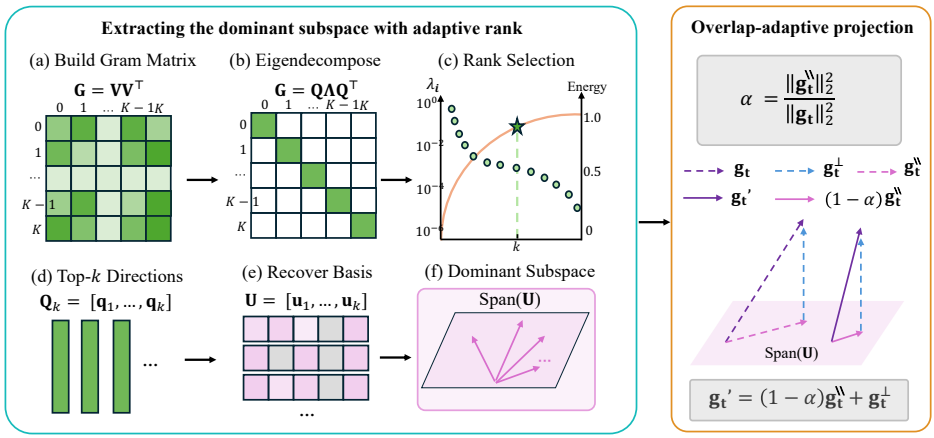


Figure 2: Overview of DiGPro. **Left**: the dominant subspace is extracted from a ring buffer of recent gradient directions via adaptive rank selection. **Right**: the current gradient is decomposed into parallel and perpendicular components; the parallel component is attenuated.

Proposition 1 (Fisher concentration implies direction concentration)**.** *Let $F_i = \mathbb{E}[g_i^2]$ denote the diagonal Fisher coefficient $i = 1, \ldots, d$, and let $\nu = \|\mathbf{g}\|_2^2$ be the total gradient energy. Then*

$$\mathbb{E}[v_i^2] = \mathbb{E}\left[\frac{g_i^2}{\nu}\right], \tag{3}$$

where $\mathbf{v} = \mathbf{g}/\|\mathbf{g}\|_2$ is the normalized gradient direction vector. Note the lowest energy ν_{min} across all ν in a local gradient window meets $\nu \geq \nu_{min} > 0$. If the coefficient of variation of ν, given as $\mathrm{cv}(\nu) = \sqrt{\mathrm{Var}(\nu)}/\mathbb{E}[\nu]$, is small within the local gradient window, then

$$\mathbb{E}[v_i^2] \approx \frac{F_i}{\sum_j F_j} \quad \textit{because} \quad \epsilon = \left|\mathbb{E}[v_i^2] - \frac{F_i}{\sum_j F_j}\right| \leq \frac{\sqrt{\mathbb{E}[g_i^4]}}{\nu_{\min}} \cdot \mathrm{cv}(\nu). \tag{4}$$

The approximation error is bounded; see Appendix D for the proof and the ablation of $\mathrm{cv}(\nu)$.

Proposition 1 implies that Fisher concentration leaves a detectable signature in the gradient stream – when the diagonal Fisher is concentrated, normalized gradient directions concentrate on the same parameters. This motivates detecting concentration directly from recent gradients, without access to per-element Fisher information.

Online dominant-subspace estimation and projection. As parameters change slowly during training, successive gradients and the evolving Fisher diagonal within a local window both also change slowly, resulting in a common subspace of dominant parameters. Figure 1(c,d) verifies that this shared concentration produces a rapidly decaying spectrum in the Gram matrix of recent gradients, indicating that the recent direction history spans an effectively low-dimensional subspace. DiGPro monitors this signature by maintaining a ring buffer $\mathbf{V} \in \mathbb{R}^{K \times d}$ of the K most recent normalized gradient directions $\mathbf{v} = \mathbf{g}/\|\mathbf{g}\|_2$ and forming their Gram matrix $\mathbf{G} = \mathbf{V}\mathbf{V}^\top \in \mathbb{R}^{K \times K}$. The buffer stores the *original* (pre-projection) directions so that the estimated subspace reflects the task's intrinsic gradient structure rather than our method's own modifications, and the Gram matrix is maintained incrementally at $\mathcal{O}(Kd)$ cost per step. At each task boundary, the buffer and Gram matrix are reset, and a brief warmup of W steps allows the subspace estimate to stabilize before projection activates.

To extract the dominant subspace, we eigendecompose the small Gram matrix at $\mathcal{O}(K^{2.376})$ cost:

$$\mathbf{G} = \mathbf{Q}\,\mathbf{\Lambda}\,\mathbf{Q}^\top, \tag{5}$$

where $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_K]$ contains the eigenvectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$. The d-dimensional eigenvectors of $\mathbf{V}^\top\mathbf{V}$ are recovered via the classical kernel trick [52]:

$$\mathbf{u}_i = \frac{\mathbf{V}^\top \mathbf{q}_i}{\sqrt{\lambda_i}} \in \mathbb{R}^d, \quad \text{for } \lambda_i > 0, \tag{6}$$

Algorithm 1 DiGPro: Dominant-direction Gradient Projection.

Require: Buffer size K, energy threshold η, warmup steps W

On each task start: reset ring buffer $\mathbf{V} \leftarrow \mathbf{0}_{K\times d}$, Gram $\mathbf{G} \leftarrow \mathbf{0}_{K\times K}$, cursor $c \leftarrow 0$, count $n \leftarrow 0$
for each training step t **do**
  Compute loss $\mathcal{L}_t$ and backpropagate $\rightarrow$ gradient $\mathbf{g}_t$ over LoRA parameters
  if $n \geq 2$ **and** $t > W$ **then**
    $\{\lambda_i, \mathbf{q}_i\} \leftarrow$ symmetric eigendecomposition of $\mathbf{G}_{1:n,\,1:n}$; sort descending ▷ $K\times K$ Gram
    Select rank k: smallest k s.t. $\sum_{i=1}^{k} \lambda_i \,/\, \sum_j \lambda_j \geq \eta$
    $\mathbf{u}_i \leftarrow \mathbf{V}^\top \mathbf{q}_i / \sqrt{\lambda_i}$; $\quad \mathbf{U} \leftarrow [\mathbf{u}_1, \ldots, \mathbf{u}_k]^\top$ ▷ recover d-dim basis
    $\alpha \leftarrow \|\mathbf{U}\mathbf{g}_t\|_2^2 / \|\mathbf{g}_t\|_2^2$ ▷ overlap ratio
    $\mathbf{g}'_t \leftarrow \mathbf{g}_t - \alpha\, \mathbf{U}^\top(\mathbf{U}\,\mathbf{g}_t)$ ▷ project out dominant component
  else
    $\mathbf{g}'_t \leftarrow \mathbf{g}_t$ ▷ no projection during warmup
  end if
  $\mathbf{v} \leftarrow \mathbf{g}_t / \|\mathbf{g}_t\|_2$ ▷ store original (pre-projection) direction
  $\mathbf{V}[c] \leftarrow \mathbf{v}$; update row/col c of $\mathbf{G}$ ▷ $\mathcal{O}(Kd)$ incremental Gram update
  $c \leftarrow (c{+}1) \bmod K$; $\quad n \leftarrow \min(n{+}1,\, K)$
  Optimizer step with $\mathbf{g}'_t$
end for

which are orthonormal by construction. Rather than a fixed rank, we select k adaptively as the smallest value whose cumulative eigenvalue energy reaches a threshold η:

$$k = \min\Big(\Big\{j \in \mathbb{N}^+ : \sum_{i=1}^{j} \lambda_i \,\Big/\, \sum_i \lambda_i \;\geq\; \eta\Big\}\Big). \tag{7}$$

This makes the method self-calibrating: concentrated phases yield rapidly decaying spectra, so a small k suffices; dispersed phases require a larger k, producing a higher-rank more uniform spectra.

With the dominant subspace basis $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]^\top \in \mathbb{R}^{k\times d}$ (rows are orthonormal basis vectors, so $\mathbf{U}^\top\mathbf{U}$ is the projector onto the dominant subspace and $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_k$), we decompose the current gradient:

$$\mathbf{g}_t = \underbrace{\mathbf{U}^\top(\mathbf{U}\,\mathbf{g}_t)}_{\mathbf{g}_t^{\shortparallel}} + \underbrace{(\mathbf{I} - \mathbf{U}^\top\mathbf{U})\,\mathbf{g}_t}_{\mathbf{g}_t^{\perp}}, \tag{8}$$

where $\mathbf{g}_t^{\shortparallel} \in \mathrm{span}(\mathbf{U})$ and $\mathbf{g}_t^{\perp}$ is orthogonal to it through the virtue of subspace complement projection matrix $(\mathbf{I} - \mathbf{U}^\top\mathbf{U})$. Not every gradient necessarily has a large parallel component, so we measure the overlap ratio:

$$\alpha = \frac{\|\mathbf{g}_t^{\shortparallel}\|_2^2}{\|\mathbf{g}_t\|_2^2} = \frac{\|\mathbf{U}\,\mathbf{g}_t\|_2^2}{\|\mathbf{g}_t\|_2^2}, \tag{9}$$

where $0 \leq \alpha \leq 1$. The role of α is to perform attenuation as the observed overlap increases: gradients with larger parallel energy (*i.e.*, gradients residing closer to the subspace spanned by $\mathbf{U}$) receive stronger suppression, while gradients with smaller overlap with the subspace are less suppressed, thereby preserving useful gradient information in dispersed phases. One could also pass α through sigmoid to boost attenuation but we opt out for brevity. The projected gradient is given as

$$\mathbf{g}'_t = \mathbf{g}_t - \alpha\,\mathbf{g}_t^{\shortparallel} = (1-\alpha)\,\mathbf{g}_t^{\shortparallel} + \mathbf{g}_t^{\perp}, \tag{10}$$

which attenuates the parallel component by $(1-\alpha)$ while leaving the perpendicular component intact. In fact, the projection acts as a selective spectral filter, *i.e.*, Fisher eigenvalues along the dominant subspace are suppressed by $(1-\alpha)^2$, while those in the orthogonal complement are preserved exactly (Appendix E). The complete procedure is given in Algorithm 1.

The projection operator has the following deterministic safety properties, which hold for any orthonormal basis $\mathbf{U}$:

Table 1: Results on the 8-task multimodal continual learning benchmark under three task orders. For QA tasks, we report average accuracy (AvgAcc); for captioning tasks, we report average CIDEr (AvgCIDEr). All results are reported as mean $\pm$std over three runs. Best mean results are in bold and second-best mean results are underlined. Our method is shaded.

	Order I			Order II			Order III		
Method	AvgAcc ↑	AvgCIDEr ↑	AvgFgt ↓	AvgAcc ↑	AvgCIDEr ↑	AvgFgt ↓	AvgAcc ↑	AvgCIDEr ↑	AvgFgt ↓
LoRA-FT [9]	45.58 ±0.72	21.02 ±1.18	38.80 ±1.31	47.64 ±0.68	22.46 ±1.09	33.40 ±1.24	47.79 ±0.70	26.64 ±1.10	28.08 ±1.19
A-GEM [22]	46.58 ±0.61	<u>29.42</u> ±1.26	<u>28.82</u> ±1.05	50.40 ±0.57	30.29 ±1.13	24.24 ±0.91	48.03 ±0.62	<u>29.14</u> ±1.08	24.62 ±0.93
GPM [43]	**47.74** ±0.66	23.60 ±1.04	33.57 ±1.22	50.76 ±0.63	24.67 ±1.11	28.98 ±1.16	48.26 ±0.65	25.21 ±1.12	29.41 ±1.18
ONG [53]	46.83 ±0.70	23.50 ±0.98	34.58 ±1.28	50.14 ±0.59	24.15 ±1.07	30.57 ±1.19	47.88 ±0.68	26.73 ±1.05	27.98 ±1.21
SGP [46]	47.27 ±0.58	24.01 ±1.15	33.33 ±1.10	<u>51.29</u> ±0.52	27.43 ±1.02	26.03 ±0.96	48.44 ±0.55	27.96 ±1.00	26.31 ±0.98
TRGP [45]	41.71 ±0.84	26.94 ±1.39	34.39 ±1.46	30.34 ±0.93	<u>34.36</u> ±1.32	31.10 ±1.41	40.72 ±0.88	29.02 ±1.34	31.26 ±1.38
BiLoRA [54]	42.12 ±0.79	25.60 ±1.21	34.33 ±1.25	42.11 ±0.76	**34.42** ±1.35	<u>24.11</u> ±0.94	47.99 ±0.77	26.91 ±1.17	27.93 ±1.03
KeepLoRA [55]	45.56 ±0.64	27.48 ±1.12	29.74 ±1.08	47.64 ±0.60	26.81 ±1.06	28.65 ±1.13	47.81 ±0.63	27.76 ±1.03	26.98 ±0.97
MoInCL [30]	47.27 ±0.57	25.73 ±1.19	30.26 ±1.27	49.02 ±0.61	25.77 ±1.08	29.35 ±1.14	<u>49.05</u> ±0.58	28.02 ±1.04	<u>24.51</u> ±0.95
DiGPro	<u>47.37</u> ±0.49	**31.56** ±0.87	**22.78** ±0.82	**51.71** ±0.43	32.31 ±0.92	**19.83** ±0.76	**49.10** ±0.45	**30.58** ±0.84	**22.36** ±0.72

Proposition 2 (Projection conservatism). *Let* $\mathbf{U} \in \mathbb{R}^{k\times d}$ *with* $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_k$, *and let* $\mathbf{g} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$. *Let* $\alpha = \|\mathbf{U}\mathbf{g}\|_2^2/\|\mathbf{g}\|_2^2$ *be the overlap ratio. Then the projected gradient* $\mathbf{g}' = (1-\alpha)\mathbf{g}^{\shortparallel} + \mathbf{g}^{\perp}$ *satisfies*

$$\|\mathbf{g}'\|_2 \le \|\mathbf{g}\|_2 \quad \textit{and} \quad \langle \mathbf{g}', \mathbf{g}\rangle > 0. \tag{11}$$

The first inequality states that the projection never amplifies the gradient norm; the second states that $\mathbf{g}'$ always forms an acute angle with $\mathbf{g}$, ensuring the projected gradient never opposes the original descent direction. See Appendix D for the proof.

5 Experiments

5.1 Setup

We evaluate DiGPro under two multimodal continual learning settings.

Setting I: cross-modal continual learning benchmark. We first consider a benchmark that contains eight sequential tasks spanning three modalities following [30], *i.e.*, audio, image, and video. Following [56], in this setting, the base model consists of `Llama-3.1-8B` [57] as the language backbone, `CLIP` [27] as the vision encoder, `BEATs` [51] as the audio encoder, and `Q-Former` [1] projectors for modality bridging. To examine robustness to task order, we adopt three task sequences:

- **Order I:** `Clotho` [58] → `Flickr30k` [59] → `AudioCaps` [60] → `MSRVTT` [61] → `MSVDQA` [62] → `ScienceQA` [63] → `OKVQA` [64] → `ClothoAQA` [65];
- **Order II:** `MSRVTT` → `Flickr30k` → `Clotho` → `ScienceQA` → `AudioCaps` → `MSVDQA` → `OKVQA` → `ClothoAQA`;
- **Order III:** `Flickr30k` → `ClothoAQA` → `MSRVTT` → `OKVQA` → `AudioCaps` → `ScienceQA` → `Clotho` → `MSVDQA`.

Setting II: `MLLM-DCL` benchmark [16]. We further evaluate DiGPro on the `MLLM-DCL` benchmark, which focuses on continual learning over purely visual tasks. In this setting, we use `LLaVA-1.5-7B` [2] as the base model. Following [55], the task sequence is: `Sensing` → `Medical` → `Driving` → `Science` → `Finance`.

Evaluation metrics. For question answering tasks, we report the average accuracy over all QA tasks. For captioning tasks, we report the average CIDEr over all captioning tasks. To measure catastrophic forgetting, we also report average forgetting (AvgFgt). For a task i, let $m_{i,t}$ denote its evaluation score after learning task t, and let T be the final task in the sequence. The forgetting of task i is defined as

$$\mathcal{F}_i = \frac{m_{i,i} - m_{i,T}}{m_{i,i}}, \tag{12}$$

Table 2: Results on the MLLM-DCL benchmark. We report the final accuracy on each VQA task, the average accuracy across all tasks (AvgAcc), and the average forgetting (AvgFgt). All methods are run with the official KeepLoRA codebase [55] under the same protocol, and results are reported as mean $\pm$std over three runs (seeds 42, 123, 999); the values reported in the original papers are listed in Table 13. Best mean results are in bold and second-best mean results are underlined. Our method is shaded.

Method	Sensing ↑	Medical ↑	Driving ↑	Science ↑	Finance ↑	AvgAcc ↑	AvgFgt ↓
LoRA-FT [9]	69.46 $\pm$0.18	44.22 $\pm$0.22	29.27 $\pm$0.27	41.33 $\pm$0.21	88.49 $\pm$0.12	54.55 $\pm$0.16	14.06 $\pm$0.20
O-LoRA [36]	72.21 $\pm$0.16	47.03 $\pm$0.24	31.47 $\pm$0.25	41.58 $\pm$0.19	88.11 $\pm$0.13	56.08 $\pm$0.15	11.97 $\pm$0.18
CL-MoE [31]	71.90 $\pm$0.17	47.23 $\pm$0.23	29.65 $\pm$0.28	41.42 $\pm$0.20	89.18 $\pm$0.11	55.88 $\pm$0.16	13.31 $\pm$0.19
SEFE [66]	76.94 $\pm$0.14	50.95 $\pm$0.20	40.40 $\pm$0.24	42.91 $\pm$0.18	88.44 $\pm$0.12	59.93 $\pm$0.14	7.43 $\pm$0.15
KeepLoRA [55]	<u>78.84</u> $\pm$0.11	**54.22** $\pm$0.18	<u>50.33</u> $\pm$0.21	<u>49.54</u> $\pm$0.16	<u>89.25</u> $\pm$0.09	<u>64.44</u> $\pm$0.12	<u>2.69</u> $\pm$0.08
DiGPro	**78.91** $\pm$0.09	<u>53.08</u> $\pm$0.17	**51.47** $\pm$0.19	**52.09** $\pm$0.14	**91.17** $\pm$0.08	**65.34** $\pm$0.10	**2.27** $\pm$0.07

i.e., the relative drop from its initial performance right after training to its final performance after the full sequence. Average forgetting is then computed as

$$\text{AvgFgt} = \frac{1}{T-1}\sum_{i=1}^{T-1}\mathcal{F}_i. \tag{13}$$

Here, $m_{i,t}$ is instantiated as accuracy for QA tasks and CIDEr for captioning tasks. Full training hyperparameters for both benchmarks are provided in Appendix F.

5.2 Main Results

DiGPro reduces forgetting while maintaining final performance. We compare DiGPro against two families of continual learning methods. On both benchmarks, LoRA-FT [9] without any anti-forgetting mechanism serves as the lower bound. On the cross-modal benchmark, we additionally compare against gradient-projection methods A-GEM [22], GPM [43], ONG [53], SGP [46], TRGP [45], and LoRA-specific methods BiLoRA [54] and KeepLoRA [55]. On the MLLM-DCL benchmark, we compare against O-LoRA [36], CL-MoE [31], SEFE [66], and KeepLoRA [55].

On the cross-modal benchmark (Table 1), we have three observations. **Firstly**, almost all continual-learning methods reduce forgetting compared with the LoRA-FT baseline, indicating that explicit continual-learning constraints are important for multimodal task streams. **Secondly**, DiGPro shows strong robustness across different task orders. It substantially reduces AvgFgt from 38.80% to 22.78% on Order I and from 33.40% to 19.83% on Order II, and shows the same trend on Order III. **Thirdly**, although DiGPro does not always achieve the highest AvgAcc or AvgCIDEr under every order, it is the only method that consistently balances both metrics: methods that lead on one metric often degrade on the other, whereas DiGPro remains strong across accuracy, captioning quality, and forgetting. This suggests that suppressing dominant gradient directions improves robustness to task-order variation rather than overfitting to a particular modality or task sequence. On the MLLM-DCL benchmark (Table 2), DiGPro achieves the best AvgAcc of 65.34% and the lowest AvgFgt of 2.27%. Across both benchmarks, these results indicate that DiGPro reduces forgetting while maintaining final-task performance, without requiring past-task storage. Figure 3 illustrates how the first three principal vectors of the subspace and its complement evolve over epochs.

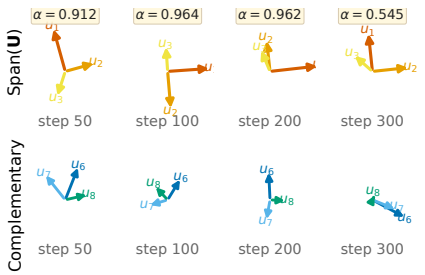
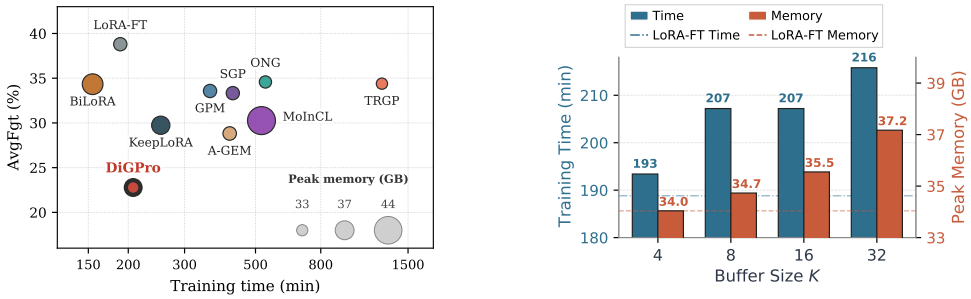


Figure 3: Dominant subspace Span(**U**) *vs*. its orthogonal complement w.r.t. step number.

5.3 Efficiency Analysis

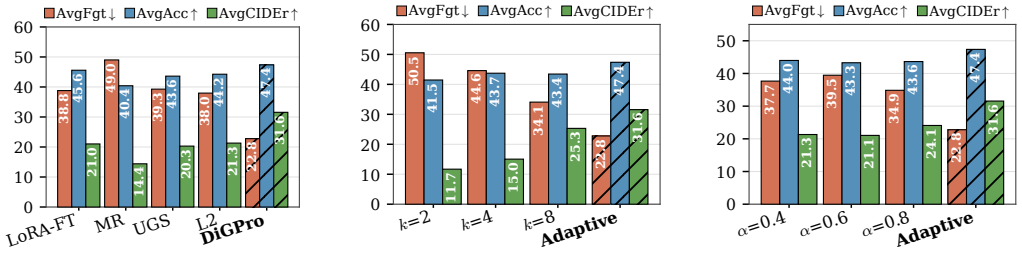
DiGPro adds limited computational overhead. We measure peak GPU memory and total training time on Order I with 4 GPUs. DiGPro adds 2.4% memory and 8.6% training time over LoRA-FT while achieving the lowest forgetting (Figure 4a). Gradient-projection baselines require 1.9–6.6× the



(a) Comparison with related methods.

(b) Effect of buffer size K.

Figure 4: **(a)** Forgetting *vs*. training time/peak GPU memory for DiGPro *vs*. competitors on Order I. **(b)** Training time/peak GPU memory for varying buffer size K. Dashed lines: LoRA-FT baseline.



(a) Directional projection.

(b) Rank selection.

(c) Projection strength.

Figure 5: Ablation on DiGPro design choices on Order I. **(a)** Directional projection *vs*. magnitude-only baselines. **(b)** Fixed *vs*. adaptive rank selection. **(c)** Fixed *vs*. overlap-adaptive proj. strength.

training time of LoRA-FT due to per-task subspace storage and full SVD; DiGPro's overhead comes only from the incremental Gram update and a small $K \times K$ eigendecomposition. Figure 4b shows that increasing K raises time and memory gently, providing a predictable quality–cost trade-off. At the larger Setting II configuration (r=64, K=48), DiGPro adds 22.9% time and 18.9% memory, which can be lowered by refreshing the subspace less often (Appendix I.2).

5.4 Scaling the LoRA Rank

Forgetting reduction persists as the LoRA rank grows. Keeping the model, benchmark (Order I) and protocol fixed, we sweep $r \in \{16, 32, 64, 128\}$ with K=$r/2$ (Table 3). Adding adapter capacity alone does not resolve interference: the AvgFgt of LoRA-FT stays at 38.8–40.4% and its AvgAcc drops from 45.58 to 42.99, whereas DiGPro reduces AvgFgt by 14.8–16.7 points at every rank. The adaptively selected rank k never reaches the cap K and covers a decreasing fraction of r (37.3% to 21.4%), while the selected subspace still captures 47–56% of the gradient energy, so the dominant-subspace assumption continues to hold. Training eight independent r=16 adapters, one per task, scores higher (AvgAcc 55.58, AvgCIDEr 45.05) but requires 8× storage and an oracle task identity at inference; DiGPro instead targets a fixed parameter budget with task-agnostic inference (Appendix I.1).

5.5 Ablation on Design Choices

We verify that each component of DiGPro contributes to its performance. All ablations use Order I.

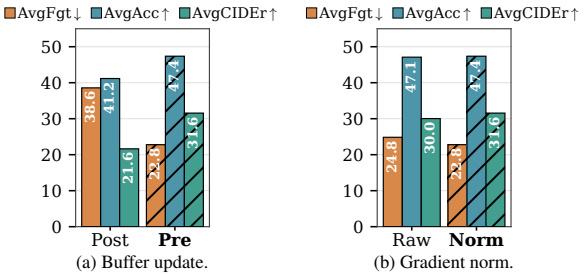
Directional projection is essential. We compare DiGPro against three magnitude-only baselines: uniform gradient scaling by a fixed scalar γ=0.6, an L2 gradient penalty $\lambda\|\mathbf{g}\|_2^2$ with λ=0.3, and magnitude reduction that rescales the gradient uniformly by the same average factor DiGPro applies directionally. All three perform poorly (Figure 5a), confirming that shrinking the gradient uniformly weakens useful updates in non-interfering directions.

Adaptive rank and projection strength. Fixed rank k substantially underperforms adaptive selection: k=2 collapses AvgFgt to 50.54%, and even k=8 reaches only 34.07%, versus 22.78% with the energy-threshold rule (Figure 5b). All fixed-α variants yield AvgFgt in the 35–40% range (Figure 5c), since a single strength cannot simultaneously handle concentrated and dispersed phases.

Table 3: LoRA-rank sweep on Order I ($K{=}r/2$). k/r: mean selected subspace rank relative to r; overhead: time / memory of DiGPro over LoRA-FT.

		LoRA-FT		DiGPro			
r	K	AvgFgt ↓	AvgAcc ↑	AvgFgt ↓	AvgAcc ↑	k/r	Overhead (time / memory)
16	8	38.80	45.58	**22.78**	**47.37**	37.3%	+8.6% / +2.4%
32	16	39.21	44.43	**23.12**	**45.90**	34.5%	+8.6% / +5.4%
64	32	40.01	44.44	**23.30**	**45.01**	33.0%	+20.6% / +17.9%
128	64	40.36	42.99	**25.61**	**43.80**	21.4%	+47.1% / +44.6%

Buffer update and gradient normalization. Storing post-projection instead of pre-projection gradient directions degrades AvgFgt from 22.78% to 38.57% (Figure 6a), as the buffer then captures the method's own modifications rather than the task's intrinsic structure. Normalizing gradients to unit length before storage improves AvgFgt from 24.84% to 22.78% (Figure 6b), by preventing high-norm gradients from dominating the subspace estimate.



(a) Buffer update. (b) Gradient norm.

Figure 6: **(a)** Pre- *vs.* post-projection buffer. **(b)** With *vs.* without gradient normalization.

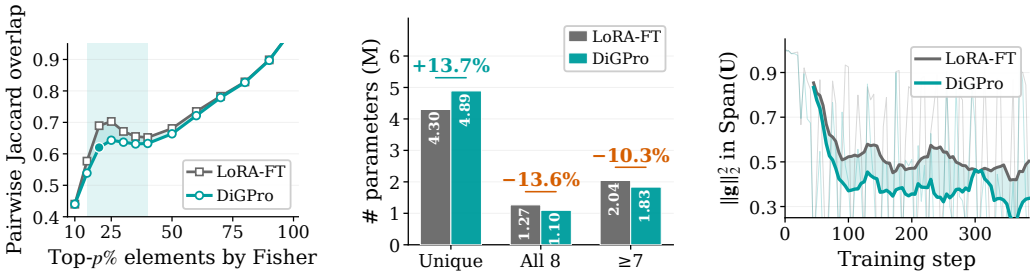
5.6 Mechanism Validation

We verify that DiGPro reduces forgetting through the intended mechanism, *i.e.*, directional projection prevents successive tasks from concentrating updates on the same parameters, and this dispersal originates directly from the gradient-level suppression of recurring dominant directions.

Parameter-level dispersal. If DiGPro works as designed, tasks trained under DiGPro should leave more distinct signatures on the parameter space rather than repeatedly overwriting the same high-sensitivity elements. We test this by computing, for each of the eight tasks, the set of top-p% parameters ranked by diagonal Fisher F_i, and measuring their cross-task Jaccard overlap (Appendix B). Figure 7a shows that DiGPro reduces pairwise overlap compared to LoRA-FT, with the separation concentrated at $p = 10$–40%. This is precisely where the method should matter: the extreme top-1% core is architecturally determined and shared by all tasks regardless, but the broader sensitive region reflects task-specific adaptation. DiGPro forces each task to recruit a more distinct set of parameters. Quantifying this at $p{=}20\%$, LoRA-FT concentrates updates on 4.30M unique parameters, of which 1.27M are shared by all eight tasks; DiGPro expands unique parameters to 4.89M (+13.7%) while reducing universally-shared ones to 1.10M (-13.6%, Figure 7b). Fewer parameters are reused by every task, and more are used by only a subset.

Gradient-level mechanism. The parameter-level dispersal should be a direct consequence of the gradient projection: if DiGPro attenuates the component of each gradient that lies within the recent dominant subspace Span($\mathbf{U}$), then the optimizer is forced to update parameters along more diverse directions. To verify this causal link, we record the dominant subspace from the incremental Gram tracker during single-task ScienceQA training and measure the fraction of each gradient's energy falling within Span($\mathbf{U}$), *i.e.*, $\alpha = \|\mathbf{U}\mathbf{g}\|_2^2 / \|\mathbf{g}\|_2^2$. Figure 7c shows that under LoRA-FT, α remains persistently high (mean 0.56). Over half of every gradient's energy stays trapped within the same small set of directions throughout training. DiGPro reduces mean α to 0.44, and the gap widens as training progresses – by the final phase, the overlap drops to 0.35, a 30% reduction from the baseline's 0.50, confirming DiGPro is responsible for the observed parameter-level dispersal: by continuously suppressing the dominant gradient components, optimizer is prevented from following the same parameter directions, forcing each task to spread its updates across a broader set of parameters.

Reduced forgetting is not reduced plasticity. Suppressing the current task's dominant directions could reduce forgetting merely by slowing learning. Yet, measured right after each task is learned, DiGPro retains 97.5% of the score of LoRA-FT on average (-0.33 CIDEr on captioning and -1.92 accuracy points on QA), whereas its final scores are higher by 10.54 AvgCIDEr and 1.79 AvgAcc. Moreover, shrinking the gradient uniformly by the same average factor does not reduce



(a) Cross-task Fisher overlap. (b) Sensitive parameter sharing. (c) Gradient overlap with Span($\mathbf{U}$).

Figure 7: Mechanism validation. **(a)** Pairwise Jaccard overlap of top-p% highest-Fisher elements across tasks on Order I. **(b)** Element-level sharing counts at p=20%: how many parameters are selected as top-20% by exactly m out of eight tasks. **(c)** Fraction of gradient energy within the dominant subspace Span($\mathbf{U}$) during ScienceQA training. See Appendix G for experimental setup.

forgetting (Figure 5a), so the gain stems from directional selectivity rather than from weaker updates (Appendix I.3).

Suppressed directions are shared with old tasks. We compare the rank-8 gradient subspace of an old task (Clotho) with the dominant subspace of each subsequent task. Under LoRA-FT, their normalized overlap is 10.0–35.4$\times$ the level expected for random subspaces for Flickr30k, MSVDQA and ScienceQA, confirming that new tasks reuse the directions of old ones; DiGPro reduces this overlap by 35.6–38.0% (Appendix I.3).

5.7 Parameter Sensitivity

Buffer size controls the quality–cost trade-off. The ring buffer is the main memory component of DiGPro, so its size K directly controls estimation quality. We vary $K \in \{8, 32, 48, 64\}$ on the `MLLM-DCL` benchmark. Table 5 (Appendix H) shows that performance improves with K, peaking at K=48 with 65.34% AvgAcc; K=64 performs comparably at 64.34%, indicating diminishing returns once the principal subspace is well estimated. Practitioners can choose a moderate K of 32–48 to balance estimation quality and overhead, without risk of catastrophic degradation at smaller values.

Adaptive rank threshold η. The energy threshold η determines how much of the recent gradient spectrum is treated as dominant. Table 6 (Appendix H) shows that η=0.95 yields the lowest AvgFgt at 22.39%, while smaller values such as η=0.80 miss repeated directions and degrade substantially. Performance is stable across η=0.85–0.95, suggesting that DiGPro does not need narrow tuning of η.

6 Conclusion

We have shown that catastrophic forgetting in LoRA-based multimodal continual instruction tuning is amplified not merely by parameter overlap, but by the combination of shared high-sensitivity parameters and concentrated new-task updates. DiGPro detects and suppresses repeated dominant directions directly from the recent gradient stream, without past-task data or stored subspaces. With a short buffer, a lightweight incremental update, and auto-adaptive rank and projection strength, DiGPro adds minimal overhead while remaining robust to varying gradient concentration. It consistently reduces forgetting while maintaining competitive performance across two benchmarks.

7 Limitations and Future Work

Our analysis has several limitations. (1) We evaluate DiGPro with 7B and 8B backbones. Experiments on larger models (*e.g.*, 70B) would test whether the dominant-subspace assumption holds as the effective gradient dimension grows. (2) The method operates on flattened LoRA gradients without exploiting the low-rank matrix structure of LoRA adapters; adapting DiGPro to work in the factor space ($\mathbf{A}$/$\mathbf{B}$ matrices separately) could improve efficiency for very high-dimensional settings. (3) Extension to non-LoRA parameter-efficient methods and to full fine-tuning remain future work.

Acknowledgments and Disclosure of Funding

This work received no third-party funding or in-kind support. The authors declare no competing interests.

References

[1] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[2] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[3] Yueyang Ding, HaoPeng Zhang, Rui Dai, Yi Wang, Tianyu Zong, Kaikui Liu, and Xiangxiang Chu. LLaTiSA: Towards difficulty-stratified time series reasoning from visual perception to semantics. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32677–32717, 2026. doi: 10.18653/v1/2026.findings-acl.1636. URL https://aclanthology.org/2026.findings-acl.1636/.

[4] Tianyu Zong, Rui Dai, Hongzhu Yi, Yuanxiang Wang, Zhenghao Zhang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Yueyang Ding, Xiangxiang Chu, Kaikui Liu, and Jungang Xu. L2Dir: Integrating L_2-norm and directional alignment for unsupervised contrastive representation learning in multimodal retrieval. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34717–34740, 2026. doi: 10.18653/v1/2026.acl-long.1604. URL https://aclanthology.org/2026.acl-long.1604/.

[5] Yu Wang, Yi Wang, Rui Dai, Yujie Wang, Kaikui Liu, Xiangxiang Chu, and Yansheng Li. Urban socio-semantic segmentation with vision-language reasoning. In *International Conference on Learning Representations*, volume 2026, pages 74495–74518, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/792845ddfe4047d7066348e52e46b74d-Paper-Conference.pdf.

[6] Yu Wang, Rui Dai, Wei Chen, Yi Wang, Bo Dang, Kaikui Liu, Xiangxiang Chu, and Yansheng Li. Urbex: Agentic spatial grounding for city-scale 3d scenes. arXiv:2610.18356, 2026. URL https://arxiv.org/abs/2610.18356.

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

[8] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[10] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[12] Frederik Benzing. Unifying importance based regularisation methods for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2372–2396. PMLR, 2022.

[13] James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[14] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[15] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[16] Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. MLLM-CL: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025.

[17] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385, 2021.

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

[19] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[20] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[21] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[22] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

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

[24] 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*, pages 7765–7773, 2018.

[25] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

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

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[28] Jinghan He, Haiyun Guo, Kuan Zhu, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *IEEE Transactions on Image Processing*, 2026.

[29] Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. *Advances in neural information processing systems*, 37:57817–57840, 2024.

[30] 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. ISSN 2835-8856. URL https://openreview.net/forum?id=FD8or43fBU.

[31] Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *Proceedings of the computer vision and pattern recognition conference*, pages 19608–19617, 2025.

[32] Yi-Kai Zhang, Shiyin Lu, Yang Li, Yanqing Ma, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Wings: Learning multimodal llms without text-only forgetting. *Advances in Neural Information Processing Systems*, 37:31828–31853, 2024.

[33] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning*, volume 234 of *Proceedings of Machine Learning Research*, pages 202–227. PMLR, 2024. URL https://proceedings.mlr.press/v234/zhai24a.html.

[34] Yanqi Dai, Yong Wang, Zebin You, Dong Jing, Xiangxiang Chu, and Zhiwu Lu. Adaptive task balancing for visual instruction tuning via inter-task contribution and intra-task difficulty. In *Proceedings of the ACM Web Conference 2026*, pages 3893–3904, 2026. doi: 10.1145/3774904.3792371.

[35] Haiwen Li, Jing Tang, Rui Chen, Lei Sun, Zining Chen, and Xiangxiang Chu. M^2Note: Continual evolution of vision language models via mistake notebook learning. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 16611–16629, 2026. doi: 10.18653/2026.emnlp-main.716. URL https://aclanthology.org/2026.emnlp-main.716/.

[36] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, 2023.

[37] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024.

[38] Yan-Shuo Liang, Jia-Rui Chen, and Wu-Jun Li. Gated integration of low-rank adaptation for continual learning of large language models. In *Advances in Neural Information Processing Systems*, volume 38, pages 68659–68689. Curran Associates, Inc., 2025. doi: 10.52202/085713-2310. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/63692d8d567db671c700df5df912204a-Abstract-Conference.html.

[39] Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5U1rlpX68A.

[40] Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6634–6645. IEEE, 2025.

[41] Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19165–19181, 2025.

[42] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pages 3762–3773. PMLR, 2020.

[43] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

[44] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. *Advances in Neural Information Processing Systems*, 34:18710–18721, 2021.

[45] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iEvAf8i6JjO.

[46] Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9677–9685, 2023.

[47] Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[48] Tao Feng, Wei Li, Didi Zhu, Hangjie Yuan, Wendi Zheng, Dan Zhang, and Jie Tang. ZeroFlow: Overcoming catastrophic forgetting is easier than you think. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 16808–16824. PMLR, 2025. URL https://proceedings.mlr.press/v267/feng25j.html.

[49] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[50] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

[51] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: Audio pre-training with acoustic tokenizers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 5178–5193. PMLR, 2023. URL https://proceedings.mlr.press/v202/chen23ag.html.

[52] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[53] Yajat Yadav, Patrick Mendoza, and Jathin Korrapati. ONG: Orthogonal natural gradient descent. *arXiv preprint arXiv:2508.17169*, 2025.

[54] Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. Bilora: almost-orthogonal parameter spaces for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25613–25622, 2025.

[55] Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeepLoRA: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T3Vc5fkTzV.

[56] 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, pages 137470–137492. Curran Associates, Inc., 2025. doi: 10.52202/085713-4588. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c8d606e3fd965d80c1dbfde63744cd2b-Abstract-Conference.html.

[57] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[58] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE, 2020.

[59] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.

[60] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, 2019.

[61] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[62] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[63] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35: 2507–2521, 2022.

[64] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[65] Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. Clotho-aqa: A crowdsourced dataset for audio question answering. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1140–1144. IEEE, 2022.

[66] Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ip, and Sam Kwong. SEFE: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 7982–8001. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25n.html.

A Derivation of the Forgetting Approximation

The forgetting approximation in Eq. (1) follows from a standard second-order expansion of the old-task loss around its local optimum. Let $\boldsymbol{\theta}^*$ denote the parameter vector after training the old task, and let $\Delta\boldsymbol{\theta}$ denote the parameter change induced by subsequent training on a new task. Then

$$\Delta\mathcal{L}^{\text{old}} = \mathcal{L}^{\text{old}}(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) - \mathcal{L}^{\text{old}}(\boldsymbol{\theta}^*). \tag{14}$$

A second-order Taylor expansion around $\boldsymbol{\theta}^*$ gives

$$\Delta\mathcal{L}^{\text{old}} \approx \nabla\mathcal{L}^{\text{old}}(\boldsymbol{\theta}^*)^\top \Delta\boldsymbol{\theta} + \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbf{H}^{\text{old}} \Delta\boldsymbol{\theta}, \tag{15}$$

where $\mathbf{H}^{\text{old}}$ is the Hessian of the old-task loss at $\boldsymbol{\theta}^*$. Because $\boldsymbol{\theta}^*$ is a local optimum of the old task, the first-order term is small and can be neglected, yielding

$$\Delta\mathcal{L}^{\text{old}} \approx \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbf{H}^{\text{old}} \Delta\boldsymbol{\theta}. \tag{16}$$

Following common practice in continual learning, we replace the Hessian with the diagonal Fisher proxy and keep only diagonal terms:

$$\mathbf{H}^{\text{old}} \approx \mathbf{F}^{\text{old}} \approx \operatorname{diag}(F_1^{\text{old}}, \ldots, F_d^{\text{old}}). \tag{17}$$

Substituting this approximation gives

$$\Delta\mathcal{L}^{\text{old}} \approx \frac{1}{2}\sum_i F_i^{\text{old}}\, \Delta\theta_i^2, \tag{18}$$

which is the expression used in the main text. This approximation is used to explain the structure of forgetting, not to predict its exact numerical value. It shows that forgetting is amplified when the new task concentrates large updates on parameters for which the old task has high sensitivity.

B Element-Level Cross-Task Overlap Metric

This appendix defines the element-level cross-task overlap metric used in Figure 1a and Figure 7b.

Setup. For each task, we accumulate the element-wise squared gradient g_i^2 over all training steps via backward hooks, yielding the diagonal Fisher $F_i = \mathbb{E}[g_i^2]$ for each of the $d \approx 13.6 \times 10^6$ LoRA parameter elements. After training, we rank all d elements by F_i in descending order and extract the top-$p\%$ highest-Fisher elements for each task.

Pairwise Jaccard overlap. Let $\mathcal{T}_i(p)$ denote the set of top-$p\%$ elements for task i. The Jaccard overlap between tasks i and j is

$$J\big(\mathcal{T}_i(p), \mathcal{T}_j(p)\big) = \frac{|\mathcal{T}_i(p) \cap \mathcal{T}_j(p)|}{|\mathcal{T}_i(p) \cup \mathcal{T}_j(p)|}. \tag{19}$$

$J = 1$ means the two tasks share exactly the same top-$p\%$ elements; $J = 0$ means no overlap.

Average pairwise overlap. With $T = 8$ tasks, we compute the average over all $\binom{T}{2} = 28$ task pairs:

$$\bar{J}(p) = \frac{2}{T(T-1)} \sum_{1 \le i < j \le T} J\big(\mathcal{T}_i(p), \mathcal{T}_j(p)\big). \tag{20}$$

Random baseline. If Fisher values were randomly distributed across the d elements, two independent random samples of size $n = \lceil p \cdot d/100 \rceil$ would have an expected Jaccard overlap of

$$J_{\text{random}}(p) = \frac{n}{2d - n}. \tag{21}$$

This follows from the expected intersection size $\mathbb{E}[|\mathcal{T}_i \cap \mathcal{T}_j|] = n^2/d$ for two independent random subsets of size n. The ratio $\bar{J}(p)\,/\,J_{\text{random}}(p)$ measures how many times more overlap is observed than expected by chance; ratios above 1 indicate genuine element-level sharing driven by model architecture.

Table 4: Average pairwise Jaccard overlap of top-p% highest-Fisher elements across eight tasks, the random baseline, and their ratio.

p (%)	1	2	5	10	20	25	50	100
$\bar{J}(p)$	0.199	0.243	0.319	0.441	0.690	0.703	0.681	1.000
$J_{\text{random}}(p)$	0.005	0.010	0.026	0.053	0.111	0.143	0.333	1.000
Ratio	39.6×	24.0×	12.4×	8.4×	6.2×	4.9×	2.0×	1.0×

Absolute overlap values. Table 4 reports the average pairwise Jaccard overlap $\bar{J}(p)$, the random baseline $J_{\text{random}}(p)$, and their ratio for $p \in \{1, 2, 5, 10, 20, 25, 50, 100\}$%, computed from the baseline (LoRA-FT) run on Order I. The ratio decays smoothly from $40\times$ at p=1% to $2\times$ at p=50%, confirming that element-level sharing persists well beyond the immediate top-ranked elements.

Interpretation. At small p (core elements), the ratio is large because all tasks concentrate Fisher on a common set of architecturally important parameters (*e.g.*, certain layers' `v_proj.B` adapters). As p grows, the ratio decreases monotonically, approaching $1\times$ at p=100% (where all elements are included and overlap is trivially perfect). The key finding is that even at p=50%, overlap remains $2\times$ above random, indicating that a substantial fraction of elements are structurally predisposed to high Fisher across tasks. For the DiGPro mechanism validation (Figure 7b), we compare $\bar{J}(p)$ between DiGPro and the LoRA-FT baseline: DiGPro reduces cross-task overlap at intermediate p (10–40%), confirming that its directional projection disperses updates away from the shared high-Fisher core.

C Experimental Details for Teaser Figures

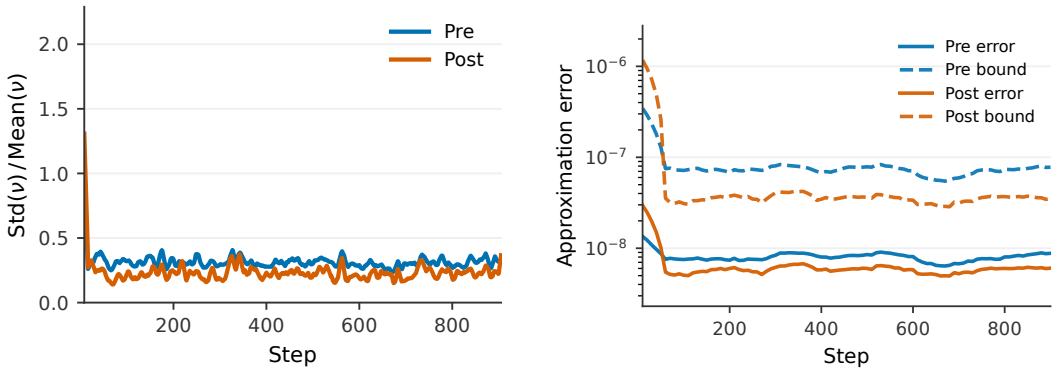
This appendix describes the experimental setup used to produce Figure 1. All statistics are collected during sequential continual learning runs under Order I: each task is trained for one epoch in sequence, and for each task we record the diagonal Fisher and gradient directions throughout its training epoch. Consequently, the statistics for later tasks reflect the presence of interference from previously trained tasks, matching the actual continual learning conditions under which DiGPro operates.

Model and training configuration. The base model consists of `Llama-3.1-8B` [57] as the language backbone, `CLIP` [27] as the vision encoder, `BEATs` [51] as the audio encoder, and `Q-Former` [1] projectors for modality bridging. During training, LoRA adapters attached to all four attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) across all 32 transformer layers of the LLM backbone are trained (giving a total of $d \approx 13.6\times10^6$ trainable LoRA parameters), along with the `Q-Former` projectors. The LLM backbone weights and vision/audio encoders are kept frozen during training. Each task is trained for one epoch using the same hyperparameters as the main continual learning experiments (Section 5).

Tasks. The eight tasks span three modalities and two task types:

- **Audio captioning:** `Clotho`, `AudioCaps`;
- **Audio QA:** `ClothoAQA`;
- **Image captioning:** `Flickr30k`;
- **Image QA:** `OKVQA`, `ScienceQA`;
- **Video captioning:** `MSRVTT`;
- **Video QA:** `MSVDQA`.

Figure 1a: Element-level cross-task overlap. For each task, the diagonal Fisher information $F_i = \mathbb{E}[g_i^2]$ is accumulated over all training steps at the element level (each of the $d \approx 13.6 \times 10^6$ LoRA parameters individually). After training, we rank all d elements by F_i in descending order and extract the top-p% highest-Fisher elements for each task. For each pair of tasks, we compute the Jaccard overlap $J(\mathcal{T}_i(p), \mathcal{T}_j(p))$ and average across all $\binom{8}{2} = 28$ task pairs to obtain $\bar{J}(p)$. The random baseline $J_{\text{random}}(p) = n/(2d - n)$ with $n = \lceil p \cdot d/100 \rceil$ gives the expected Jaccard if Fisher values were randomly distributed across elements. Figure 1a plots the ratio $\bar{J}(p)/J_{\text{random}}(p)$ against p; absolute values are tabulated in Appendix B. The ratio decays from $40\times$ at p=1% to $2\times$ at p=50%, confirming that individual high-sensitivity elements are shared far beyond chance.



(a) Relative variation of ν.

(b) Empirical error and bound.

Figure 8: Empirical validation of Proposition 1 on the task-local DiGPro buffer. **(a)** The coefficient of variation of $\nu = \|\mathbf{g}\|_2^2$. **(b)** The empirical discrepancy between normalized direction energy and the Fisher ratio.

Figure 1b: Effective Fisher rank. Using the same element-level Fisher values F_i from above, we compute the effective Fisher rank for each task as

$$r_{\text{eff}} = \exp\big(-\textstyle\sum_{i=1}^{d} p_i \log p_i\big), \quad p_i = F_i \,/\, \textstyle\sum_{j=1}^{d} F_j, \tag{22}$$

where the sum runs over all $d \approx 13.6\times10^6$ LoRA parameter elements. A value of $r_{\text{eff}} = d$ would indicate a perfectly uniform Fisher distribution; values much smaller than d indicate concentration. The observed values range from 5.0×10^5 (`ScienceQA`, most concentrated) to 2.7×10^6 (`AudioCaps`, most dispersed), varying by over $5\times$ despite all tasks sharing the same architecture and parameter count.

Figure 1c: Gram eigenvalue spectrum. During the training epoch of each task (within the sequential run), we maintain a ring buffer of the $K{=}64$ most recent normalized gradient directions $\mathbf{v}_t = \mathbf{g}_t/\|\mathbf{g}_t\|_2$, where $\mathbf{g}_t \in \mathbb{R}^d$ is the gradient over all LoRA parameters at step t. Every 50 training steps (after the buffer has been filled), we compute the $K\times K$ Gram matrix $\mathbf{G} = \mathbf{V}\mathbf{V}^\top$ and record its eigenvalue spectrum $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$, normalized so that $\sum_i \lambda_i = 1$. This yields approximately 90 spectrum snapshots per task (varying slightly with task size). Figure 1c plots, for each task, the median normalized eigenvalue at each index position $i \in \{1, \ldots, 64\}$, with shaded bands showing the interquartile range (25th–75th percentile) across snapshots. A horizontal dashed line at $1/K = 1/64$ marks the uniform baseline: eigenvalues above this line carry more energy than a random direction would. Tasks with concentrated diagonal Fisher (*e.g.*, `ScienceQA`, `ClothoAQA`) show rapid spectral decay, with a few eigenvalues dominating, while tasks with dispersed Fisher (*e.g.*, `AudioCaps`, `Flickr30k`) show a flatter spectrum closer to the uniform baseline.

Figure 1d: Fisher effective rank *vs.* spectral decay rate. This scatter plot relates the two preceding analyses. The x-axis is the Fisher effective rank from Figure 1b. The y-axis is the spectral decay rate ξ, obtained by fitting a power law $\lambda_i = c \cdot i^{-\xi}$ to each task's median Gram eigenvalue spectrum via ordinary least-squares regression in log-log space:

$$\xi = -\frac{\sum_{i=1}^{K}(\log i - \overline{\log i})(\log \lambda_i - \overline{\log \lambda})}{\sum_{i=1}^{K}(\log i - \overline{\log i})^2}, \tag{23}$$

where $\overline{\log i}$ and $\overline{\log \lambda}$ denote the respective means. A larger ξ indicates faster spectral decay and thus stronger gradient concentration. The dashed trend line is a linear fit to the eight data points. The negative correlation confirms the theoretical prediction (Eq. (4)): tasks whose diagonal Fisher is more concentrated produce gradient histories with more rapidly decaying Gram spectra, indicating that gradient directions are confined to a lower-dimensional subspace.

D Proofs of Propositions

D.1 Proof of Proposition 1

Let $F_i = \mathbb{E}[g_i^2]$ denote the diagonal Fisher information at coordinate i, and let $\nu = \|\mathbf{g}\|_2^2 = \sum_{j=1}^d g_j^2$ be the total gradient energy. The normalized gradient direction is $\mathbf{v} = \mathbf{g}/\|\mathbf{g}\|_2$, so $v_i = g_i/\sqrt{\nu}$ and

$$\mathbb{E}[v_i^2] = \mathbb{E}\left[\frac{g_i^2}{\nu}\right]. \tag{24}$$

To relate this to $F_i/\sum_j F_j$, we write

$$\mathbb{E}\left[\frac{g_i^2}{\nu}\right] - \frac{\mathbb{E}[g_i^2]}{\mathbb{E}[\nu]} = \mathbb{E}\left[\frac{g_i^2(\mathbb{E}[\nu]-\nu)}{\nu\cdot\mathbb{E}[\nu]}\right] = -\frac{1}{\mathbb{E}[\nu]}\mathbb{E}\left[\frac{g_i^2(\nu-\mathbb{E}[\nu])}{\nu}\right]. \tag{25}$$

Let $\nu_{\min}$ be the smallest ν in the buffer and assume $\nu_{\min} > 0$, *i.e.*, it is unlikely the gradient energy is ever exactly equal zero. Since $\nu \geq \nu_{\min} > 0$ almost surely, we have $1/\nu \leq 1/\nu_{\min}$. Applying the Cauchy–Schwarz inequality to the expectation on the right-hand side:

$$\left|\mathbb{E}\left[\frac{g_i^2(\nu-\mathbb{E}[\nu])}{\nu}\right]\right| \leq \frac{1}{\nu_{\min}}\mathbb{E}\left[\,|g_i^2(\nu-\mathbb{E}[\nu])|\,\right] \leq \frac{1}{\nu_{\min}}\sqrt{\mathbb{E}[g_i^4]\,\mathrm{Var}(\nu)}. \tag{26}$$

Therefore,

$$\left|\mathbb{E}[v_i^2] - \frac{F_i}{\sum_j F_j}\right| = \left|\mathbb{E}[v_i^2] - \frac{\mathbb{E}[g_i^2]}{\mathbb{E}[\nu]}\right| \leq \frac{\sqrt{\mathbb{E}[g_i^4]\,\mathrm{Var}(\nu)}}{\mathbb{E}[\nu]\cdot\nu_{\min}}. \tag{27}$$

When the coefficient of variation $\mathrm{cv}(\nu) = \sqrt{\mathrm{Var}(\nu)}/\mathbb{E}[\nu]$ is small and $\nu_{\min}$ is bounded away from zero, the right-hand side is small. In practice, the fourth moment $\mathbb{E}[g_i^4]$ does not dominate because training gradients are not heavy-tailed, and the total gradient energy ν has low relative variance within a local training window.

Empirical validation of the approximation. We validate the assumptions behind this bound using the same task-local gradient buffer used by DiGPro's online auto-rank tracker. For each training step, we record the eight most recent LoRA gradients before projection and after projection, compute $\nu_t = \|\mathbf{g}_t\|_2^2$ within the buffer, and compare the empirical discrepancy $\left|\mathbb{E}_B[v_i^2] - \mathbb{E}_B[g_i^2]/\mathbb{E}_B[\nu]\right|$ with the bound in Eq. (27). Figure 8a shows that the relative variation of ν remains controlled after the buffer is filled: the pre-projection ratio $\sqrt{\mathrm{Var}_B(\nu)}/\mathbb{E}_B[\nu]$ is typically around 0.25–0.40, while projection further reduces it to roughly 0.15–0.30. Thus, ν is not assumed to be constant, but its task-local variation is moderate rather than comparable to its mean. Figure 8b shows that the resulting approximation error is extremely small throughout training and remains consistently below the theoretical bound. This confirms that the finite-buffer correction term in Proposition 1 is numerically small in the actual gradient stream used by DiGPro, supporting the use of normalized gradient directions as a practical proxy for diagonal Fisher concentration.

□

D.2 Proof of Proposition 2

Let $\mathbf{g}^{\shortparallel} = \mathbf{U}^\top(\mathbf{U}\mathbf{g})$ and $\mathbf{g}^{\perp} = \mathbf{g} - \mathbf{g}^{\shortparallel}$. Since $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_k$, the parallel projector $\mathbf{\Pi}_{\shortparallel} = \mathbf{U}^\top\mathbf{U}$ is symmetric and idempotent, so $\mathbf{g}^{\shortparallel}$ and $\mathbf{g}^{\perp}$ are orthogonal:

$$\langle \mathbf{g}^{\shortparallel}, \mathbf{g}^{\perp}\rangle = \mathbf{g}^\top\mathbf{\Pi}_{\shortparallel}^\top(\mathbf{I}-\mathbf{\Pi}_{\shortparallel})\mathbf{g} = \mathbf{g}^\top\mathbf{\Pi}_{\shortparallel}(\mathbf{I}-\mathbf{\Pi}_{\shortparallel})\mathbf{g} = \mathbf{g}^\top(\mathbf{\Pi}_{\shortparallel}-\mathbf{\Pi}_{\shortparallel}^2)\mathbf{g} = 0. \tag{28}$$

The projected gradient is $\mathbf{g}' = (1-\alpha)\mathbf{g}^{\shortparallel} + \mathbf{g}^{\perp}$.

(i) Norm non-amplification. Because $\mathbf{g}^{\shortparallel} \perp \mathbf{g}^{\perp}$,

$$\|\mathbf{g}'\|_2^2 = (1-\alpha)^2\|\mathbf{g}^{\shortparallel}\|_2^2 + \|\mathbf{g}^{\perp}\|_2^2. \tag{29}$$

Since $\alpha \in (0,1)$, we have $(1-\alpha)^2 \leq 1$, and therefore

$$\|\mathbf{g}'\|_2^2 \leq \|\mathbf{g}^{\shortparallel}\|_2^2 + \|\mathbf{g}^{\perp}\|_2^2 = \|\mathbf{g}\|_2^2, \tag{30}$$

which yields $\|\mathbf{g}'\|_2 \leq \|\mathbf{g}\|_2$.

(ii) Direction non-reversal. Using orthogonality again,

$$\langle \mathbf{g}', \mathbf{g} \rangle = \langle (1-\alpha)\mathbf{g}^{\shortparallel} + \mathbf{g}^{\perp},\ \mathbf{g}^{\shortparallel} + \mathbf{g}^{\perp} \rangle = (1-\alpha)\|\mathbf{g}^{\shortparallel}\|_2^2 + \|\mathbf{g}^{\perp}\|_2^2. \tag{31}$$

Since $\|\mathbf{g}\|_2^2 = \|\mathbf{g}^{\shortparallel}\|_2^2 + \|\mathbf{g}^{\perp}\|_2^2 = \alpha\|\mathbf{g}\|_2^2 + \|\mathbf{g}^{\perp}\|_2^2$, we have $\|\mathbf{g}^{\perp}\|_2^2 = (1-\alpha)\|\mathbf{g}\|_2^2$. Substituting $\|\mathbf{g}^{\shortparallel}\|_2^2 = \alpha\|\mathbf{g}\|_2^2$ gives

$$\langle \mathbf{g}', \mathbf{g} \rangle = (1-\alpha)\alpha\|\mathbf{g}\|_2^2 + (1-\alpha)\|\mathbf{g}\|_2^2 = \|\mathbf{g}\|_2^2(1-\alpha^2). \tag{32}$$

Because $\alpha \in [0, 1)$, we have $\alpha^2 \in [0, 1)$, so $1 - \alpha^2 > 0$. Hence $\langle \mathbf{g}', \mathbf{g} \rangle > 0$ strictly, which means the angle between $\mathbf{g}$ and $\mathbf{g}'$ is always acute (never obtuse or right). □

D.3 Supplementary Discussion for Proposition 3

Proposition 3 states that when $\mathbf{F}$ and $\mathbf{P}$ commute (which occurs when $\mathbf{F}$'s eigenvectors are either in $\mathrm{span}(\mathbf{U})$ or orthogonal to it), the projected covariance $\mathbf{F}' = \mathbf{P}\mathbf{F}\mathbf{P}^\top$ has eigenvalues scaled by $(1-\alpha)^2$ or 1 accordingly.

For a general eigenvector $\mathbf{v}$ of $\mathbf{F}$ that is neither parallel nor orthogonal to $\mathrm{span}(\mathbf{U})$, $\mathbf{P}$ and $\mathbf{F}$ do not commute, so $\mathbf{v}$ need not remain an eigenvector of $\mathbf{F}'$. However, the spectral effect can still be understood through the quadratic form. Decompose any direction $\mathbf{v} = \mathbf{v}_{\shortparallel} + \mathbf{v}_{\perp}$ with $\mathbf{v}_{\shortparallel} = \mathbf{\Pi}_{\shortparallel}\mathbf{v}$ and $\mathbf{v}_{\perp} = \mathbf{\Pi}_{\perp}\mathbf{v}$. Then

$$\mathbf{v}^\top\mathbf{F}'\mathbf{v} = \mathbf{v}^\top\mathbf{P}\mathbf{F}\mathbf{P}^\top\mathbf{v} = (1-\alpha)^2\mathbf{v}_{\shortparallel}^\top\mathbf{F}\mathbf{v}_{\shortparallel} + \mathbf{v}_{\perp}^\top\mathbf{F}\mathbf{v}_{\perp} + (1-\alpha)\mathbf{v}_{\shortparallel}^\top\mathbf{F}\mathbf{v}_{\perp} + (1-\alpha)\mathbf{v}_{\perp}^\top\mathbf{F}\mathbf{v}_{\shortparallel}. \tag{33}$$

If $\mathbf{v}$ is an eigenvector of $\mathbf{F}$ with eigenvalue λ, then $\mathbf{v}^\top\mathbf{F}\mathbf{v} = \lambda$ and the cross terms vanish only when $\mathbf{v}_{\shortparallel}$ and $\mathbf{v}_{\perp}$ are both eigenvectors of $\mathbf{F}$ with the same eigenvalue (which is the commuting case). In the non-commuting case, the energy in $\mathbf{v}$ is redistributed between the parallel and perpendicular components, but the overall effect remains: the parallel component is suppressed by a factor of $(1-\alpha)^2$ relative to the perpendicular component. This justifies the heuristic that the projection acts as a selective spectral filter along the dominant subspace.

E Fisher Reshaping via Subspace Projection

This appendix gives the formal statement of how DiGPro's projection reshapes the local Fisher structure. Let $\mathbf{U} \in \mathbb{R}^{k\times d}$ be the dominant subspace basis with $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_k$, and let $\alpha \in [0, 1]$ be the overlap ratio defined in Eq. (9). Define the parallel and perpendicular projectors $\mathbf{\Pi}_{\shortparallel} = \mathbf{U}^\top\mathbf{U}$ and $\mathbf{\Pi}_{\perp} = \mathbf{I} - \mathbf{U}^\top\mathbf{U}$, and the projection matrix $\mathbf{P} = \mathbf{I} - \alpha\,\mathbf{\Pi}_{\shortparallel} = (1-\alpha)\,\mathbf{\Pi}_{\shortparallel} + \mathbf{\Pi}_{\perp}$.

Proposition 3 (Subspace projection reshapes Fisher)**.** *The projected gradient is* $\mathbf{g}' = \mathbf{P}\mathbf{g}$. *Denoting the local gradient covariance by* $\mathbf{F} = \mathbb{E}[\mathbf{g}\mathbf{g}^\top]$, *the projected covariance is:*

$$\mathbf{F}' = \mathbb{E}[\mathbf{g}'\mathbf{g}'^\top] = \mathbf{P}\,\mathbf{F}\,\mathbf{P}^\top. \tag{34}$$

In particular, for any eigenpair $(\lambda, \mathbf{v})$ *of* $\mathbf{F}$ *that is either aligned with or orthogonal to the dominant subspace:*

- *if* $\mathbf{v} \in \mathrm{span}(\mathbf{U})$, i.e. $\mathbf{\Pi}_{\shortparallel}\mathbf{v} = \mathbf{v}$, *then* $\mathbf{P}\mathbf{v} = (1-\alpha)\mathbf{v}$, *so the corresponding eigenvalue in* $\mathbf{F}'$ *is* $(1-\alpha)^2\lambda$;
- *if* $\mathbf{v} \perp \mathrm{span}(\mathbf{U})$, i.e. $\mathbf{\Pi}_{\perp}\mathbf{v} = \mathbf{v}$, *then* $\mathbf{P}\mathbf{v} = \mathbf{v}$, *so the corresponding eigenvalue in* $\mathbf{F}'$ *remains* λ.

(For eigenvectors that are neither parallel nor orthogonal to $\mathrm{span}(\mathbf{U})$, $\mathbf{P}$ *and* $\mathbf{F}$ *need not commute, so* $\mathbf{F}'$ *may not share the same eigenbasis as* $\mathbf{F}$*; see Appendix D.)*

This confirms that DiGPro acts as a selective spectral filter: Fisher eigenvalues along the dominant subspace are suppressed by $(1-\alpha)^2$, while eigenvalues in the orthogonal complement are preserved exactly.

F Full Benchmark Training Details

This appendix provides the complete training configuration for both benchmarks, reproducibility details, and baseline implementation notes.

Reproducibility and statistical reliability. All main results (Tables 1 and 2) are averaged over 3 independent runs with different random seeds $(42, 123, 999)$. The seed controls dataset shuffling, model initialization, and all stochastic operations (dropout, data augmentation). We use PyTorch's deterministic mode (`cudnn.deterministic=True, cudnn.benchmark=False`). The forgetting metric $\mathcal{F}_i$ and AvgFgt are computed for each run separately and then averaged. We report the mean and standard deviation across runs.

Baseline implementation and hyperparameter selection. All methods are evaluated under identical hardware (4×NVIDIA H100, 80 GB each), identical base models, identical LoRA configuration, and identical task sequences. For each baseline, we use the hyperparameters reported in the respective original papers or official code repositories: GPM [43], SGP [46], TRGP [45], and ONG [53] share the same subspace budget (max_rank $= 64$, threshold $= 0.95$, grad_collect_samples $= 256$) as specified in their official implementations. A-GEM [22] uses memory size 256 and reference batch size 64 following the original paper. BiLoRA [54] uses the official `bilora` parameterization with $n_{\text{frequency}} = 3000$ and spectral_scale $= 300$. KeepLoRA [55] uses threshold_w $= 0.6$, threshold_f $= 0.99$, and grad_data_ratio $= 0.2$ as in its public code. DiGPro's hyperparameters were chosen as follows and kept fixed across all settings without per-setting tuning. K is set to balance estimation quality against memory overhead; the sensitivity analysis (Appendix H) shows that performance is stable across a wide range. $\eta = 0.95$ is a standard energy retention threshold in PCA-based analysis, capturing the dominant spectral modes while dropping noise components. $W = 10$ provides enough steps at each task boundary to fill the buffer and let the subspace estimate stabilize before projection activates. The overlap ratio α (Eq. (9)) is used directly as the attenuation coefficient: when gradient directions are dispersed (α small), the parallel component is naturally small and the projection is weak, preserving useful learning signals; when gradients concentrate on recurring directions (α large), the parallel component dominates and receives proportionally stronger suppression. This linear mapping from overlap to attenuation is self-calibrating—it requires no additional hyperparameters because α already ranges in $[0, 1]$ and directly quantifies how much of the current gradient lies within the repeating subspace. For the magnitude-only baselines in the directional projection ablation (Section 5.5), UGS and L2 penalty hyperparameters were selected from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ by choosing the best-performing value on Order I. MR uses the same average reduction factor as DiGPro at each step, applied uniformly to the full gradient without directional projection.

Setting I: cross-modal continual learning benchmark. All experiments in this setting use 4 NVIDIA H100 GPUs with DeepSpeed ZeRO-2. The base model is `Llama-3.1-8B` [57] with `CLIP` [27] (vision), `BEATs` [51] (audio), and `Q-Former` [1] projectors. During continual learning, LoRA adapters attached to the attention projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and the `Q-Former` projectors are trainable; the LLM backbone and vision/audio encoders remain frozen. LoRA configuration: rank r=16, alpha =32, dropout 0.05. Each task is trained for one epoch. The model uses two `Q-Former` projectors: one for image and video (which share the same encoder `CLIP`) and one for audio. Both projectors are trainable and updated via the standard optimizer during each task. At the end of each task, the projector corresponding to that task's modality is averaged with the original pretrained projector using equal weights (coefficient 0.5); the averaged projector then serves as the initialization for the next task of the same modality. This projector averaging is independent of the gradient projection mechanism and applies to all methods (including baselines) evaluated on this benchmark.

Task order rationale. The three task orders are designed to test robustness under varying degrees of task-type interleaving. The eight tasks consist of four captioning tasks (`Clotho`, `Flickr30k`, `AudioCaps`, `MSRVTT`) and four QA tasks (`MSVDQA`, `ScienceQA`, `OKVQA`, `ClothoAQA`). Order I places all captioning tasks before all QA tasks, representing maximal type clustering: the model first encounters only open-ended generation tasks, then switches entirely to close-ended tasks, which creates the strongest asymmetry between early and late phases. Order II begins with captioning tasks but introduces QA tasks mid-sequence, producing moderate interleaving. Order III alternates captioning and QA tasks from the start, testing the method under frequent task-type switches. Together, these three orders span the spectrum from fully clustered to heavily interleaved, ensuring that reported results are not artifacts of a particular ordering.

Optimization settings: AdamW optimizer, learning rate $1e{-4}$, weight decay 0.05, cosine learning-rate decay with warmup ratio 0.03. Training is performed with bfloat16 mixed precision via DeepSpeed ZeRO-2. The per-device batch size is 4 with gradient accumulation of 4 steps, yielding an effective batch size of $4 \times 4 \times 4 = 64$.

DiGPro hyperparameters (Setting I). For the cross-modal benchmark, DiGPro uses a ring buffer of size $K{=}8$, energy threshold $\eta{=}0.95$, and warmup $W{=}10$ steps. The small buffer is sufficient because the LoRA rank is $r{=}16$, so the effective gradient-space dimensionality is low.

Setting II: `MLLM-DCL` benchmark. All experiments in this setting use 4 NVIDIA H100 GPUs with DeepSpeed ZeRO-2. The base model is `LLaVA-1.5-7B` [2]. LoRA configuration: rank $r{=}64$, alpha $=128$, dropout 0.05, targeting all linear layers in the vision-language projector and the LLM's attention modules. Each task is trained for at most three epochs.

Optimization settings: AdamW optimizer, learning rate $1e{-5}$, weight decay 0.05, cosine learning-rate decay with warmup ratio 0.03. Training is performed with bfloat16 mixed precision via DeepSpeed ZeRO-2. The per-device batch size is 4 with gradient accumulation of 4 steps, yielding an effective batch size of 64.

DiGPro hyperparameters (Setting II). For the `MLLM-DCL` benchmark, DiGPro uses a larger buffer $K{=}48$ (vs. $K{=}8$ in Setting I), with the same $\eta{=}0.95$ and $W{=}10$. The larger buffer accommodates the higher effective gradient-space dimensionality introduced by the larger LoRA rank ($r{=}64$).

G Mechanism Validation: Gradient Subspace Experiment Details

This appendix describes the experimental setup used for the gradient-level mechanism validation in Figure 7c.

Task and model configuration. The experiment uses `ScienceQA` [63], an image-based question answering task. ScienceQA is chosen because QA tasks exhibit stronger Fisher concentration than captioning tasks (Figure 1b), making the gradient collapse phenomenon more pronounced and the contrast between DiGPro and LoRA-FT clearer. The base model, LoRA configuration, optimizer, and training hyperparameters are identical to those used in the cross-modal benchmark (Appendix F): `Llama-3.1-8B` backbone, LoRA rank $r{=}16$, AdamW with learning rate $1e{-4}$, weight decay 0.05, and cosine learning-rate decay with warmup ratio 0.03. The task is trained for one epoch using bfloat16 mixed precision with DeepSpeed ZeRO-2 on 4 NVIDIA H100 GPUs.

Subspace dynamics recording. Both DiGPro and the LoRA-FT baseline use identical configurations except that the baseline disables gradient projection ($\alpha{=}0$). In both runs, a ring buffer of the $K{=}8$ most recent normalized gradient directions $\mathbf{v}_t = \mathbf{g}_t / \|\mathbf{g}_t\|_2$ is maintained and their $K \times K$ Gram matrix is updated incrementally at each training step. Every 5 steps, we eigendecompose the Gram matrix, recover the d-dimensional dominant subspace basis using Eq. (6), and record the fraction of current gradient energy in Span($\mathbf{U}$): $\alpha = \|\mathbf{U}\mathbf{g}\|_2^2 / \|\mathbf{g}\|_2^2$. All metrics are recorded to JSONL for post-hoc analysis.

Visualization. Figure 7c plots α against training step for both DiGPro and LoRA-FT, with trend lines obtained via a moving average of window size 12. The shaded region highlights the gap between the two trend lines where DiGPro yields lower subspace overlap.

H Parameter Sensitivity Tables

Buffer size.

Energy threshold.

I Additional Experiments

I.1 LoRA Rank and Adapter Capacity

Table 7 complements Table 3 with statistics of the adaptively selected subspace, averaged over the full Order I sequence. The energy threshold $\eta{=}0.95$ selects k from the Gram spectrum of the buffer,

Table 5: Sensitivity analysis on buffer size K on the MLLM-DCL benchmark (all variants use DiGPro with LoRA rank r=64). We report final accuracy on each task after training all five tasks. Best results per column are in **bold** and second-best are underlined.

K	Sensing ↑	Medical ↑	Driving ↑	Science ↑	Finance ↑	AvgAcc ↑
8	76.22	52.73	44.25	44.56	90.61	61.67
32	76.73	50.70	47.62	44.45	90.57	62.01
48	**78.91**	**53.08**	**51.47**	52.09	**91.17**	**65.34**
64	77.21	50.55	50.67	**52.14**	91.14	64.34

Table 6: Sensitivity analysis on energy threshold η (all variants use DiGPro). For captioning tasks we report CIDEr; for QA tasks we report accuracy. Best results per column are in **bold** and second-best are underlined.

η	Clotho	Flickr30k	AudioCaps	MSRVTT	MSVDQA	ScienceQA	OKVQA	ClothoAQA	AvgCIDEr	AvgAcc	AvgFgt
0.80	9.72	11.58	19.74	17.14	24.52	61.78	19.08	54.33	14.54	39.93	48.25
0.85	19.51	28.92	31.85	36.31	33.46	56.19	24.43	56.04	29.15	42.53	27.42
0.90	14.05	33.17	27.25	37.50	29.81	**71.04**	**28.78**	**57.25**	27.99	**46.72**	27.39
0.95	**19.86**	**35.75**	**34.29**	**37.92**	**34.39**	67.28	28.05	56.07	**31.95**	46.45	**22.39**

whereas $\alpha = \|\mathbf{U}\mathbf{g}\|_2^2/\|\mathbf{g}\|_2^2$ measures the energy of the current gradient captured by the selected subspace, so the two quantities need not coincide. The cap-hit fraction is the share of projection calls for which the selected rank reaches k=K; it is zero at every rank, so the selected rank is never constrained by the buffer size. The rank sweep sets K=$r/2$ to avoid tuning K separately at each rank. This rule differs from Setting II, where K=48 was selected at r=64 (Table 5); at that configuration DiGPro adds 22.9% training time and 18.9% peak memory relative to LoRA-FT.

Table 7: Adaptively selected subspace across LoRA ranks (Order I, K=$r/2$).

r	K	Mean k	Mean k/K	Mean k/r	Cap-hit fraction	Captured energy α
16	8	5.96	74.45%	37.25%	0%	46.97%
32	16	11.04	69.00%	34.50%	0%	48.13%
64	32	21.09	65.90%	32.95%	0%	48.35%
128	64	27.42	42.85%	21.42%	0%	55.60%

Table 8 compares DiGPro with enlarging a shared adapter and with training one independent adapter per task. A higher rank alone does not resolve interference: from r=16 to r=128, the AvgFgt of shared LoRA-FT remains high while its AvgAcc drops from 45.58 to 42.99 and its AvgCIDEr from 21.02 to 9.46. Independent adapters perform best because they amount to eight separate models, which requires 8× storage and an oracle task identity to select the adapter at inference, so they do not fall under the task-agnostic incremental setting considered here. DiGPro is therefore preferable under a fixed parameter budget and task-agnostic inference.

I.2 Basis Refresh Interval

By default (q=1), DiGPro recomputes the dominant subspace at every projection call; the ring buffer and the Gram matrix are always updated. Table 9 recomputes the subspace only every q calls (Order I, r=16, K=8). Refreshing every 2 or 4 calls preserves performance and saves 3.1–4.0% training time, whereas $q \geq 8$ relies on a stale subspace and increases forgetting. We keep q=1 as the default.

I.3 Plasticity and Alignment with Old-Task Directions

Table 10 compares each task's score immediately after it is learned with its score at the end of the Order I sequence. DiGPro nearly preserves the immediate scores of LoRA-FT while substantially improving the final ones, so its forgetting reduction cannot be explained by reduced plasticity.

To test whether the attenuated directions are those that old tasks depend on, we construct rank-8 gradient bases for Clotho (the old task) and for each of three subsequent tasks on the same LoRA coordinates, and compute their normalized subspace overlap $\|\mathbf{U}_{\text{old}}\mathbf{U}_{\text{cur}}^{\top}\|_F/\sqrt{8}$, reported relative to

Table 8: Fixed-budget interference mitigation *vs*. capacity expansion on Order I.

Method	Capacity	AvgAcc ↑	AvgCIDEr ↑	AvgFgt ↓	Stored task states	Oracle task ID
Shared LoRA-FT, r=16	1×	45.58	21.02	38.80	1	No
Shared LoRA-FT, r=32	2×	44.43	18.57	39.21	1	No
Shared LoRA-FT, r=64	4×	44.44	13.26	40.01	1	No
Shared LoRA-FT, r=128	8×	42.99	9.46	40.36	1	No
Shared DiGPro, r=16	1×	47.37	31.56	22.78	1	No
8 independent r=16 adapters	8×	55.58	45.05	N/A	8	Yes

Table 9: Effect of the basis refresh interval q (Order I, r=16, K=8).

q	AvgAcc ↑	AvgCIDEr ↑	AvgFgt ↓	Time saved
1	47.37	31.56	22.78	–
2	46.58	33.48	22.17	3.1%
4	48.49	32.65	21.83	4.0%
8	46.92	28.52	29.20	5.0%
16	46.49	31.24	26.37	5.5%

the expected overlap of two random rank-8 subspaces (Table 11). Under LoRA-FT, the current-task dominant subspaces overlap the Clotho subspace far beyond chance; DiGPro consistently lowers this overlap, which shows that it suppresses current-task directions shared with the old task.

I.4 Update Concentration and Forgetting

Task characteristics. Update concentration is associated with the output format of a task: captioning tasks (mean median target length of 11.00 tokens) have a mean Fisher effective rank of 2.293M, whereas QA tasks (1.25 tokens) have 1.004M. Across the eight tasks, target length is positively associated with the Fisher effective rank (Spearman ρ=0.690, p=0.069). Longer free-form targets may impose more diverse token-level constraints and involve a broader set of parameters, whereas short answers may produce repeatedly aligned updates.

Controlled transitions. To relate update concentration directly to forgetting, we branch from the same post-Clotho checkpoint and fine-tune on three targets of increasing Fisher effective rank for 768 micro-steps (192 optimizer updates) each (Table 12). Relative Clotho forgetting decreases monotonically from 13.1% to 6.4% and 2.3% as the Fisher effective rank of the target grows, while the target scores recover to a similar extent; ScienceQA, the most concentrated target, causes the most forgetting despite the smallest parameter displacement. The single-task references use each task's standard schedule, so the recovery ratio only serves as a sanity check of target learning.

I.5 Reruns of the MLLM-DCL Baselines

All MLLM-DCL experiments use the official KeepLoRA codebase [55]. Table 2 reports the mean and standard deviation of our reruns of every method with seeds 42, 123 and 999. Table 13 compares them with the values reported in the original papers, whose AvgFgt we compute from the reported task-level results with the same definition. The reruns agree with the reported values within 0.2 points on every metric.

J Broader Impact

DiGPro is a general-purpose optimization technique for mitigating catastrophic forgetting in continual learning. On the positive side, it reduces the computational and data burden of retraining models from scratch when new tasks or modalities are added, which can lower the energy and carbon footprint of maintaining deployed multimodal AI systems. By not requiring storage of past-task data, it also reduces privacy risks associated with retaining user data from earlier tasks. On the negative side, continual learning methods that preserve performance across tasks may inadvertently preserve biases or harmful capabilities learned from earlier training phases; however, this risk is not specific to DiGPro and applies to continual learning methods broadly. We do not foresee specific harmful

Table 10: Immediate (right after learning the task) and final (after the full sequence) scores on Order I.

Task group (metric)	Immediate score			Final score		
	LoRA-FT	DiGPro	Δ	LoRA-FT	DiGPro	Δ
Captioning (CIDEr)	45.52	45.19	-0.33	21.02	31.56	$+10.54$
QA (accuracy)	57.21	55.29	-1.92	45.58	47.37	$+1.79$

Table 11: Overlap between the Clotho gradient subspace and the dominant subspace of subsequent tasks, relative to random subspaces.

Transition	LoRA-FT overlap / random	DiGPro overlap / random	Relative reduction
Clotho $\rightarrow$ Flickr30k	10.0$\times$	6.3$\times$	37.0%
Clotho $\rightarrow$ MSVDQA	27.9$\times$	17.3$\times$	38.0%
Clotho $\rightarrow$ ScienceQA	35.4$\times$	22.8$\times$	35.6%

use cases uniquely enabled by our method beyond those already possible with existing fine-tuning techniques.

Table 12: Forgetting of Clotho after fine-tuning the same post-Clotho checkpoint on targets with increasing Fisher effective rank (768 micro-steps each).

Target	Fisher effective rank	$\|\Delta\boldsymbol{\theta}\|_2$	Target score / single-task reference	Relative Clotho drop ↓
ScienceQA	0.504M	4.635	75.89 / 80.20 (94.6%)	13.1%
OK-VQA	0.977M	7.078	36.49 / 38.91 (93.8%)	6.4%
MSVDQA	1.752M	5.651	47.51 / 49.86 (95.3%)	2.3%

Table 13: MLLM-DCL baselines: our reruns (mean ± std over three seeds) *vs*. the values reported in the original papers.

Method	Source	`Sensing`	`Medical`	`Driving`	`Science`	`Finance`	AvgAcc	AvgFgt
LoRA-FT	rerun	69.46±0.18	44.22±0.22	29.27±0.27	41.33±0.21	88.49±0.12	54.55±0.16	14.06±0.20
	reported	69.34	44.30	29.10	41.44	88.43	54.52	13.93
O-LoRA	rerun	72.21±0.16	47.03±0.24	31.47±0.25	41.58±0.19	88.11±0.13	56.08±0.15	11.97±0.18
	reported	72.30	46.89	31.59	41.50	88.06	56.07	12.08
CL-MoE	rerun	71.90±0.17	47.23±0.23	29.65±0.28	41.42±0.20	89.18±0.11	55.88±0.16	13.31±0.19
	reported	71.83	47.36	29.49	41.48	89.16	55.86	13.19
SEFE	rerun	76.94±0.14	50.95±0.20	40.40±0.24	42.91±0.18	88.44±0.12	59.93±0.14	7.43±0.15
	reported	77.05	50.86	40.27	42.98	88.40	59.91	7.51
KeepLoRA	rerun	78.84±0.11	54.22±0.18	50.33±0.21	49.54±0.16	89.25±0.09	64.44±0.12	2.69±0.08
	reported	78.76	54.34	50.19	49.48	89.30	64.41	2.63
DiGPro	ours	78.91±0.09	53.08±0.17	51.47±0.19	52.09±0.14	91.17±0.08	65.34±0.10	2.27±0.07

NeurIPS Paper Checklist

i. **Claims**

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

Answer: [Yes]

Justification: The abstract and introduction (Section 1) state the key findings (shared high-sensitivity parameters, concentrated updates amplifying forgetting, gradient direction signature) and the proposed method (DiGPro), all of which are supported by the theoretical results (Propositions 1–3) and experimental results (Section 5). The contributions listed at the end of the introduction match the paper's actual content.

Guidelines:

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

ii. **Limitations**

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

Answer: [Yes]

Justification: Section 7 discusses the restriction to 7B and 8B backbones, the use of flattened LoRA gradients, and the open question of how DiGPro extends beyond LoRA.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

iii. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Propositions 1 and 2 state their assumptions in the main text and are proved in Appendix D. Proposition 3 and its assumptions are stated in Appendix E, with the spectral derivation in Appendix D.3.

Guidelines:

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

iv. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Appendix F provides complete training configuration including model architecture, LoRA settings, optimizer, learning rate, batch size, random seeds, hardware, and all DiGPro hyperparameters with rationale. All datasets are publicly available and cited.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: The benchmark datasets are publicly available. Before camera-ready submission, we will make the DiGPro code and instructions for installation, data preparation, and training public at https://github.com/AMAP-ML/DiGPro.

Guidelines:

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

vi. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 5 and Appendix F specify the complete experimental setup including model configuration, LoRA settings, optimizer (AdamW), learning rate, weight decay, batch size, number of epochs, hardware, random seeds, and all hyperparameters for both DiGPro and baselines.

Guidelines:

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

vii. **Experiment statistical significance**

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

Answer: [Yes]

Justification: All main results (Tables 1 and 2) report mean $\pm$ standard deviation over 3 independent runs with different random seeds (42, 123, 999), as stated in Appendix F.

Guidelines:

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

viii. **Experiments compute resources**

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

Answer: [Yes]

Justification: Hardware (4$\times$NVIDIA H100, 80 GB), training time, and peak GPU memory are reported in Section 5 (Figure 4a) and Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

ix. **Code of ethics**

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

Answer: [Yes]

Justification: This work uses publicly available datasets and models, does not involve human subjects, and does not release high-risk assets. All existing assets are properly cited.

Guidelines:

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

x. **Broader impacts**

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

Answer: [Yes]

Justification: A dedicated Broader Impact section (Appendix J) discusses both positive impacts (reduced computational and data burden, lower privacy risk from not storing past-task data) and potential negative impacts (preservation of biases or harmful capabilities from earlier training phases).

Guidelines:

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

xi. **Safeguards**

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

Answer: [N/A]

Justification: The DiGPro release consists of training code. We use existing benchmarks and pretrained models and do not release new datasets or model weights.

Guidelines:

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

xii. **Licenses for existing assets**

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

Answer: [No]

Justification: We credit the creators of the models, datasets, and baseline implementations in Section 5 and Appendix F. The manuscript does not list the license terms for every third-party asset.

Guidelines:

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

xiii. **New assets**

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

Answer: [Yes]

Justification: The DiGPro implementation and its README will be public at `https://github.com/AMAP-ML/DiGPro` before camera-ready submission. The README covers installation, data preparation, and training.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects. All experiments use existing publicly available benchmark datasets.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

xvi. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The proposed method (DiGPro) is a gradient projection algorithm designed entirely by the authors without the use of LLMs. LLMs appear in this work only as the base models being fine-tuned in the experimental evaluation, which is standard practice.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.