# Dual-Generalization-Aware Minimization for Continual Fine-Tuning of Vision-Language Models

Yanan Chen[1,2], Tieliang Gong[1,2⋆], Yanle Lyu[4], Yuanhong Zhang[1,3], and Weizhan Zhang[1,3]

[1] School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, P.R. China
[2] Ministry of Education Key Laboratory of Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, P.R. China
[3] Shaanxi Province Key Laboratory of Big Data Knowledge Engineering, Xi'an Jiaotong University, Xi'an 710049, P.R. China
[4] Data Science Institute, Brown University, Providence, RI 02903, USA
{ynchen894, adidasgtl}@gmail.com

**Abstract.** Continual fine-tuning of large vision–language models (VLMs) such as CLIP is crucial for adapting foundation models to evolving real-world distributions. However, such adaptation often disrupts pretrained representations and severely harms zero-shot generalization. Despite recent progress, existing methods overlook the underlying optimization geometry, failing to anchor convergence within regions robust to distribution shift. While recent studies suggest that flatter loss minima improve generalization, we empirically demonstrate that seeking local flatness on a single task is insufficient to bridge the distributional gap between pre-trained and task-specific representations. Motivated by this observation, we propose Dual-Generalization-aware Minimization (DGM), a plug-and-play optimization framework for continual adaptation of VLMs. DGM extends flatness-aware training by redefining the perturbation objective to jointly promote pre-trained knowledge preservation and task-specific robustness. By incorporating these objectives into the inner maximization step, DGM steers optimization toward solutions that remain stable under both zero-shot evaluation and task adaptation. Extensive experiments demonstrate that DGM effectively maintains zero-shot generalization while achieving strong performance on sequential tasks.

**Keywords:** Continual Learning · VLMs · Sharpness-Aware Methods

## 1 Introduction

Continual Learning (CL) enables models to sequentially acquire new knowledge without discarding previously learned capabilities [9, 15, 33]. Conventional approaches primarily focus on training models from scratch [27, 35], aiming to

⋆ Corresponding author.

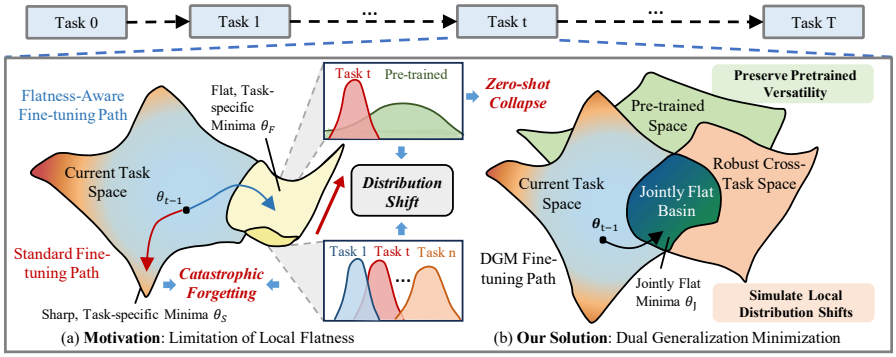



**Fig. 1:** Illustration of generalization degradation during continual fine-tuning and the proposed DGM framework. (a) Standard fine-tuning leads to sharp minima, and flatness-aware methods optimized for a single task distribution are vulnerable under sequential distribution shifts. (b) DGM incorporates perturbations from both pre-trained knowledge and task-specific variations to locate flatter solutions that remain stable across evolving tasks.

mitigate the catastrophic forgetting of previous task knowledge while maintaining sufficient plasticity for new data. Leveraging the remarkable zero-shot transfer and cross-modal alignment of large-scale pre-trained models, adapting vision-language models (VLMs) such as CLIP [34] has emerged as a promising direction that attracts increasing attention. However, sequential fine-tuning on these foundation models still faces a fundamental bottleneck between acquiring task-specific expertise and preserving zero-shot transferability [6, 26].

Recent research has explored diverse strategies for continual adaptation of VLMs, including structural isolation through modular routing or task-specific adapters [32,47,49] and representation-level alignment that enforces feature consistency across tasks [21,22,51,54]. While these approaches mitigate catastrophic forgetting through parameter or representation constraints, they rarely consider how the optimization process shapes the model's generalization behavior. Recent studies on foundation model adaptation suggest that generalization is closely related to the geometry of the loss landscape [5, 20, 41]. In particular, solutions located in flatter regions tend to exhibit stronger robustness to distribution shifts and improved transferability across tasks [41]. However, existing continual learning methods rarely incorporate such geometric considerations into their design. Consequently, continual fine-tuning often drives the model toward sharp, task-specific minima that degrade zero-shot generalization and cross-domain transfer (Fig. 1(a)).

While flatness-aware methods have been shown to improve generalization in static, single-task settings [23,50] and have proven beneficial for continual learning for models trained from scratch [3], they remain insufficient for the multi-distribution nature of continual fine-tuning in vision-language models. When large pre-trained VLMs are adapted to sequential tasks, the optimization process

is influenced not only by parameter sensitivity but also by distributional drift between the broad pre-training distribution and the narrow task-specific distributions encountered during adaptation. Such shifts alter both feature statistics and the curvature of the loss landscape over time (Fig. 1(a)). Perturbations derived solely from the current task loss thus capture only a partial view of this evolving landscape and fail to account for directions of instability introduced by distributional drift. This limitation makes single-task perturbation strategies insufficient to capture the dynamics that lead to catastrophic forgetting and degradation of zero-shot generalization.

To address this challenge, we propose Dual-Generalization-aware Minimization (DGM), a plug-and-play optimization framework that promotes flat solutions during continual fine-tuning of vision-language models. DGM explicitly targets the multi-distribution nature of continual adaptation by estimating perturbations from two complementary sources: (1) the pre-trained model's predictions, which preserve zero-shot generalization inherited from pre-training, and (2) task-specific visual perturbations, which improve robustness to the current task distribution. By jointly minimizing sensitivity to perturbations from these two sources, DGM encourages the model to converge to solutions that remain stable against both pre-trained knowledge erosion and task-specific overfitting, thereby mitigating catastrophic forgetting while preserving zero-shot generalization (Fig. 1(b)).

The key contributions of this work are summarized as follows:

- We empirically reveal that optimizing flatness exclusively on the current task leaves the model vulnerable under the pre-trained distribution, leading to both catastrophic forgetting and degradation of zero-shot generalization.
- We propose Dual-Generalization-aware Minimization (DGM), an optimization framework that integrates perturbations from both pre-trained knowledge and task-specific variations to locate solutions stable across evolving task distributions.
- DGM is a plug-and-play optimizer that consistently improves knowledge retention and zero-shot generalization across multiple continual learning benchmarks without additional data or parameters.

## 2 Related Work

**Class-Incremental Learning (CIL).** Traditional continual learning (CL) methods are typically categorized as follows [33, 42, 44]. 1) Replay-based methods [1, 35, 39] mitigate forgetting by storing or generating a subset of exemplars from previous tasks for joint training. However, they are sensitive to memory size and prone to overfitting. 2) Regularization-based methods constrain parameter updates important to prior tasks [28, 36, 38], their fixed capacity often limits plasticity as tasks accumulate. 3) Architecture-based methods [30, 53] dynamically adapt or expand the network to accommodate new knowledge while isolating interference across tasks, though they frequently rely on task identifiers during inference. Recently, several works have explored continual learning from

the perspective of optimization dynamics and loss landscape geometry [10, 37], demonstrating that sharpness-aware optimization significantly alleviate forgetting by encouraging flatter minima and improving generalization across tasks.

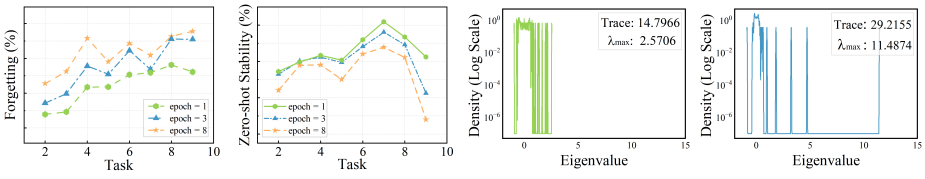
**CIL with Vision-Language Models.** The strong zero-shot and cross-domain capabilities of pretrained vision-language models (VLMs), such as CLIP, have motivated increasing efforts to adapt them to downstream tasks. While robust fine-tuning methods [14,45] partially preserve zero-shot capabilities, they are primarily designed for single-stage adaptation. When extended to class-incremental learning (CIL) settings [18,40], these models still suffer from catastrophic forgetting during sequential adaptation. Existing approaches attempt to mitigate this issue from several perspectives. Architecture-based methods introduce modular components, such as task-specific adapters, dynamic routing mechanisms, or mixture-of-experts modules, to decouple task representations and reduce inter-task interference [32, 47, 49]. Regularization-based methods aim to preserve pre-trained knowledge through various constraints, including representation-level alignment [18, 52], restricted feature updates [22, 51], and auxiliary alignment objectives [19, 21, 54]. While optimization-based strategies have also been explored, they are typically limited to heuristic adjustments such as learning rate scheduling [48]. Consequently, most existing methods focus on controlling parameter updates or representations, while the role of optimization dynamics in shaping generalization during continual fine-tuning remains underexplored.

**Flatness-Aware Optimization.** Recent research has explored optimization strategies that seek flat minima to improve model generalization [13, 16]. Representative approaches include Sharpness-Aware Minimization (SAM) [13], which approximates zeroth-order sharpness by maximizing the training loss within a local neighborhood, and Gradient-norm Aware Minimization (GAM) [50], which measures first-order sharpness to encourage feature stability and reuse. Subsequent studies have provided theoretical insights highlighting the importance of local curvature and loss landscape flatness for improving robustness and transferability [2, 7, 8]. Motivated by these findings, recent work has extended flatness-aware optimization to the fine-tuning of large pretrained models, particularly in parameter-efficient adaptation settings [12, 25, 29]. However, these methods are primarily designed for single-task fine-tuning and do not explicitly address the challenges of maintaining generalization during continual adaptation across evolving tasks.

## 3 Methodology

### 3.1 Preliminaries

**Class-Incremental Learning.** We adopt the Class-Incremental Learning (CIL) framework, where a pre-trained model $f_0$ is sequentially fine-tuned on a stream




**(a)** Forgetting **(b)** Zero-shot stability **(c)** Small epoch **(d)** Large epoch

**Fig. 2:** Forgetting and zero-shot stability under different fine-tuning budgets. In(a)(b), we show the cross-task accuracy matrices under small and large epoch. In(c)(d), we show the forgetting and zero-shot stability curves.

of $T$ classification tasks $\{(\mathcal{C}_t, \mathcal{D}_t)\}_{t=1}^{T}$. Each task $t \in [1, T]$ consists of a training dataset $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ with $N_t$ labeled samples and a unique set of $n_t$ classes $\mathcal{C}_t = \{c_t^1, c_t^2, \ldots, c_t^{n_t}\}$. Following standard CIL conventions, tasks have non-overlapping class spaces: $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $i \neq j$. During training on task $t$, the model $f_t$ has access solely to the current dataset $\mathcal{D}_t$ and optimizes the classification loss over $\mathcal{C}_t$, while all previous task data $\{\mathcal{D}_1, \ldots, \mathcal{D}_{t-1}\}$ are unavailable. After completing task $t$, the model is required to accurately classify samples across the cumulative class space $\mathcal{C}_{1:t} = \bigcup_{i=1}^{t} \mathcal{C}_i$ without reliance on task identifiers at inference time.

**Sharpness-Aware Optimization.** Sharpness aware minimization (SAM) [13] aims to reduce the sharpness of the loss surface by minimizing the maximal loss changes from a small perturbation $\epsilon$ on the model's learnable parameters $\theta$. The optimization objective is defined as:

$$\min_{\theta} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{train}(\theta + \epsilon), \tag{1}$$

where $\rho$ is the size of the neighborhood and $\mathcal{L}_{train}$ represents a general learning objective. To efficiently optimize the objection in Eq. (1), it is approximated via first order expansion and compute the adversarial perturbation $\epsilon$ as follows:

$$\epsilon = \rho \cdot \frac{\nabla L(\theta)}{\|\nabla L(\theta)\|_2}, \tag{2}$$

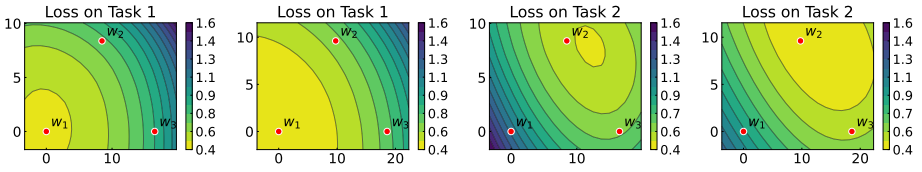
where $\nabla L(\theta)$ is the gradient of $L(\theta)$ with respect to $\theta$. Subsequently, one can compute the gradient at the perturbed point $\theta + \epsilon$, and then use the updating step of the base optimizer such as Adam to update

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta)|_{\theta_t + \epsilon_t}, \tag{3}$$

where $\eta$ denotes the learning rate.

### 3.2 Empirical Analysis

**Geometry of Generalization Decay.** To examine how generalization degrades under continual adaptation, we evaluate naive fine-tuning on CIFAR-100

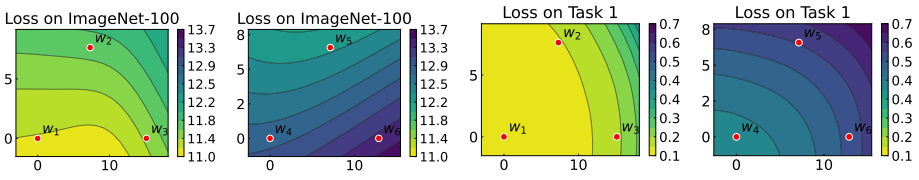



**(a)** Adam **(b)** SAM **(c)** Adam **(d)** SAM

**Fig. 3:** Loss Landscape Visualizations of Sequential Tasks. (a)(c) Loss surfaces for Task 1 and Task 2 optimized via Adam. (b)(d) Loss surfaces for Task 1 and Task 2 optimized via SAM. SAM identifies significantly broader basins for all tasks, mitigating inter-task interference.

with a ten-task incremental split and varying training budgets. We analyze the accuracy matrix $R_{i,j}$, where each entry denotes the accuracy on task $T_j$ after training on $T_i$, and two quantitative metrics: forgetting score $\frac{1}{T-1}\sum_{j<i}(R_{jj} - R_{ij})$ (past-task retention) and zero-shot stability $\frac{1}{T-1}\sum_{j>i} R_{ij}$ (future-task consistency). As shown in Fig. 2a and Fig. 2b, forgetting escalates while zero-shot stability monotonically declines as fine-tuning epochs increase. These results indicates that extended fine-tuning progressively overwrites prior knowledge while simultaneously eroding the pretrained model's generalization. Then we analyze the sharpness of the resulting solutions by comparing the Hessian eigenvalue distributions in Fig. 2c and Fig. 2d. Both the spectral norm ($\lambda_{max}$) and the trace increase substantially with longer fine-tuning, indicating a shift toward sharper, more task-specific regions of the loss landscape. This alignment between increased sharpness and performance decay suggests that standard fine-tuning progressively replaces broad pretrained representations with narrow, task-specific minima, highlighting the need for optimization strategies that explicitly preserve generalization during continual adaptation.

**Effect of Flatness-Aware Optimization.** Prior work [31, 37, 46] has shown that flatter minima mitigate forgetting in continual learning for small-scale models trained from scratch. To examine whether this property extends to pretrained models, we visualize the loss landscapes of sequential tasks using a two-dimensional projection around converged solutions. The converged parameter vector after training on task $T_i$ is denoted by $w_i$, with projection details provided in the Appendix. We conduct a controlled experiment on CIFAR-100, where three tasks are trained sequentially using either standard Adam or Sharpness-Aware Minimization (SAM). As illustrated in Fig. 3, optimization with standard Adam produces disjoint, sharp minima, where minimizing the loss of a new task leads to a significant increase in the loss of previous tasks, indicating severe inter-task interference. In contrast, models trained with SAM converge to significantly broader basins. The increased overlap between low-loss regions across tasks suggests that flatness provides a useful proxy for cross-task stability. However, pursuing parameter-level flatness for the current task alone remains in-




**(a)** First 3 Tasks **(b)** Second 3 Tasks **(c)** First 3 Tasks **(d)** Second 3 Tasks

**Fig. 4:** Evolution of loss landscapes under sequential distribution shifts. (a–b) Zero-shot landscape (ImageNet-100): the loss surface becomes increasingly irregular as tasks accumulate, indicating degradation of zero-shot generalization. (c–d) Task 1 landscape: the initial flat basin gradually shifts as the model adapts to later tasks, illustrating the instability of task-specific flatness under sequential adaptation.

sufficient under the evolving distributional conditions encountered in continual fine-tuning of VLMs.

**Fragility under Distribution Shifts.** While the previous analysis shows that flatness-aware optimization mitigates inter-task interference, an important question remains: does a flat solution for the current task remain stable under the evolving data distributions inherent in continual learning? To investigate this, we extend the task sequence to ten tasks and track the evolution of the loss landscape for both the initial task and an out-of-distribution proxy (ImageNet-100). As shown in Fig. 4, the flat region identified by SAM for Task 1 quickly drifts away from the low-loss basin as the model adapts to subsequent tasks. More critically, the zero-shot landscape (ImageNet-100) becomes increasingly irregular and exhibits persistent upward drift. These observations reveal a fundamental limitation of standard flatness-aware methods: it assumes a stationary data distribution and therefore cannot account for the distributional drift between broad pre-training knowledge and narrow task-specific objectives. This motivates the need for an optimization objective that explicitly considers both sources of distributional variation during continual adaptation.

### 3.3 Dual-Generalization-Aware Minimization

To address the limitation of single-task perturbation in sharpness-aware methods, we propose Dual-Generalization-aware Minimization (DGM), an optimization framework designed for continual fine-tuning of vision-language models. Instead of deriving perturbations solely from the current task loss, DGM incorporates two complementary objectives: (1) a zero-shot knowledge objective that preserves alignment with the pre-trained model, and (2) a task-specific robustness objective that captures variations within the current task distribution. By computing perturbations from the combined objective, DGM encourages the model to learn solutions that remain stable under both pre-trained knowledge preservation and newly observed task distributions.

**Dual-Generalization Perturbation** To obtain solutions that are robust to both pre-trained knowledge erosion and task-specific overfitting, we define a composite perturbation loss:

$$\mathcal{L}_{perturb}(\theta; \mathcal{D}_t) = \lambda_1 \mathcal{L}_{ZS}(\theta, \theta_0; \mathcal{D}_t) + \lambda_2 \mathcal{L}_{NV}(\theta; \hat{\mathcal{D}}_t), \tag{4}$$

where $\theta$ denotes the current model parameters, $\theta_0$ the frozen pre-trained model, $\mathcal{L}_{ZS}$ preserves alignment with the pre-trained predictions, and $\mathcal{L}_{NV}$ introduces robustness through noisy visual inputs $\hat{\mathcal{D}}_t$. We optimize the following minimax objective:

$$\min_{\theta} \max_{\|\epsilon\|_2 \leq \rho} \left[\mathcal{L}_{train}(\theta + \epsilon) + \mathcal{L}_{perturb}(\theta + \epsilon)\right]. \tag{5}$$

Following SAM, the adversarial perturbation is approximated by a first-order Taylor expansion:

$$\epsilon \approx \rho \frac{\nabla_\theta \left[\mathcal{L}_{train}(\theta) + \mathcal{L}_{perturb}(\theta)\right]}{\left\|\nabla_\theta \left[\mathcal{L}_{train}(\theta) + \mathcal{L}_{perturb}(\theta)\right]\right\|_2}. \tag{6}$$

The model parameters $\theta$ are then updated by minimizing the loss at the perturbed location $\theta + \epsilon$ in Eq. (3).

**Zero-Shot Generalization Probe.** To preserve the model's zero-shot generalization behavior, we utilize the predictions of the frozen pre-trained model as a reference. Given a sample $x \in \mathcal{D}_t$, we align the prediction probabilities $P_t(x)$ of the evolving model $f_t$ with the zero-shot probabilities $P_{ZS}(x)$ generated by the frozen pre-trained model $f_0$. This consistency is enforced through the Zero-Shot Knowledge loss:

$$\mathcal{L}_{ZS} = \mathrm{KL}\big(P_t(x) \parallel P_{ZS}(x)\big), \tag{7}$$

where KL divergence is employed to quantify the distributional shift. By incorporating $\nabla_\theta \mathcal{L}_{ZS}$ into the perturbation computation, we ensure that the generated $\epsilon$ explores directions that would otherwise compromise the model's inherent versatile representations.

**Task-Specific Robustness Probe.** To improve robustness to variations within the current task distribution, we introduce a stochastic visual perturbation mechanism to probe the local curvature of the current task distribution. Following the intuition of visual uncertainty, we model each visual representation $x_i$, $i \in [1, N_t]$ as a Gaussian distribution to simulate potential distribution shifts within the task. We estimate the sample-wise mean $\mu$ and variance $\sigma^2$ by calculating the statistical moments of the latent visual features across their dimensions, and synthesize a noisy variant $\tilde{x}_i$:

$$\tilde{x}_i = \delta \mathcal{N}(\mu, \sigma^2) + (1 - \delta) x_i, \ i \in [1, N_t], \tag{8}$$

where $\delta$ is the noise ratio. The task-specific robustness loss is defined as:

$$\mathcal{L}_{NV} = \mathcal{L}_{train}\big(\theta; \hat{\mathcal{D}}_t\big), \quad \text{where } \hat{\mathcal{D}}_t = \{(\hat{x}_i, y_i)\}_{i=1}^{N_t}. \tag{9}$$

**Algorithm 1:** DGM for Continual Learning

**Inputs**: Training phase $T$, training data $\mathcal{D}^T$, model $f^{T-1}$ with parameter $\theta^{T-1}$ from last phase if $T > 1$, batchsize $b$, oracle loss function $\mathcal{L}_{train}$, learning rate $\eta > 0$, neighborhood size $\rho$, trade-off coefficient $\lambda_1, \lambda_2$, small constant $c$.
**Output**: Model trained at the current time $T$

**Initialize:**: **if** $T = 1$: Randomly Initialize parameter $\theta^{T=1}$
**while** not converged **do**
  Sample a batch $\{x, y\}$ from $X_t, Y_t$
  Compute $\mathcal{L}_{perturb}$ in Eq. (4)
  Compute gradient $\nabla_\theta \mathcal{L}_{train}(\theta)$ and $\nabla_\theta \mathcal{L}_{perturb}(\theta)$
  Compute $\epsilon(\theta)$ in Eq. (6)
  Approximate gradient $g = \nabla_\theta \mathcal{L}_{train}(\theta)|_{\theta+\epsilon(\theta)}$
  Update model parameter: $\theta = \theta - \eta g$;
**end while**
**return** $\theta^T$ and $f^T$

By training under the perturbation derived from $\nabla_\theta \mathcal{L}_{NV}$, the model is forced to identify regions that are not only optimal for the current samples but also flat with respect to visual variations, thereby enhancing resilience to the distribution shifts inherent in sequential learning.

**Integration and Optimization.** DGM jointly optimizes zero-shot and task-specific generalization objectives within each training step, ensuring stable adaptation while preserving pre-trained generality. As a plug-and-play optimization module, DGM can be seamlessly integrated into diverse continual learning or fine-tuning frameworks, ranging from full-parameter and parameter-efficient tuning (e.g., Adapter, LoRA) to constraint-based or mixture-of-experts architectures without additional parameters or architectural changes. In terms of computational efficiency, DGM maintains the streamlined nature of standard SAM, requiring 2 forward and 2 backward propagations per iteration to sense the dual-distribution landscape. This guides optimization toward flatter regions that better balance zero-shot transferability and task-specific robustness, without introducing prohibitive computational overhead. The full training procedure of DGM is summarized in Algorithm 1.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We evaluate DGM under the class-incremental learning (CIL) protocol on six widely used benchmarks: CIFAR-100 [24], Tiny-ImageNet [11], CUB-200 [43], ImageNet-R [17], and ImageNet-100/1000 [11]. To assess scalability and initialization effects, we adopt two configurations: B0_Inc$n$, which incrementally learns all classes from scratch, and B$m$_Inc$n$, which starts from $m$ base classes

**Table 1:** Main results across all benchmarks under the two incremental configurations. DGM consistently improves accuracy and reduces forgetting across all baselines and incremental settings.

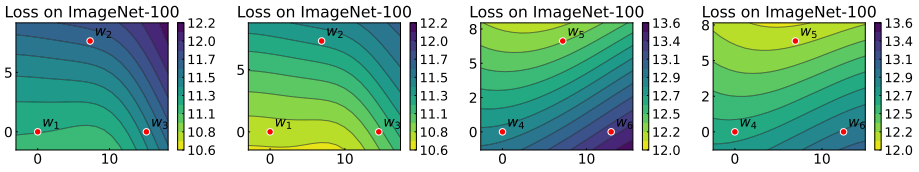
| Methods | CIFAR-100 | | | | ImageNet-R | | | | CUB-200 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B0_Inc10 | | B50_Inc10 | | B0_Inc20 | | B100_Inc20 | | B0_Inc20 | | B100_Inc20 | |
| | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. |
| Finetune | 79.30 | 86.88 | 81.33 | 86.08 | 81.63 | 87.15 | 82.38 | 84.98 | 58.97 | 70.31 | 63.26 | 70.79 |
| w/ours | 81.12 | 87.69 | 83.03 | 87.04 | 82.47 | 87.54 | 83.53 | 86.10 | 60.75 | 71.24 | 65.93 | 72.23 |
| RAPF [18] | 79.39 | 85.87 | 79.81 | 83.04 | 79.83 | 85.65 | 79.82 | 82.59 | 76.20 | 82.69 | 76.48 | 79.47 |
| w/ours | 80.11 | 86.15 | 80.13 | 83.39 | 80.38 | 86.15 | 80.35 | 83.11 | 76.34 | 82.64 | 76.56 | 79.34 |
| ZSCL [51] | 78.33 | 85.34 | 82.13 | 85.67 | 82.57 | 86.71 | 84.10 | 86.17 | 59.67 | 71.72 | 66.83 | 74.38 |
| w/ours | 80.45 | 86.81 | 83.42 | 87.05 | 83.17 | 87.18 | 84.35 | 86.46 | 61.06 | 72.48 | 68.02 | 74.88 |
| MoE4CL [47] | 78.95 | 86.05 | 81.60 | 85.66 | 81.88 | 87.32 | 83.27 | 85.73 | 55.95 | 69.76 | 59.80 | 69.58 |
| w/ours | 80.62 | 87.12 | 82.06 | 86.60 | 83.07 | 88.03 | 84.05 | 86.72 | 57.56 | 70.78 | 61.98 | 70.83 |
| MG-CLIP [19] | 79.56 | 86.99 | 80.56 | 84.64 | 82.62 | 87.71 | 82.78 | 84.88 | 65.72 | 73.48 | 64.29 | 69.28 |
| w/ours | 81.42 | 87.66 | 81.57 | 85.64 | 82.90 | 87.91 | 83.53 | 86.10 | 67.05 | 74.75 | 68.38 | 72.79 |
| DMNSP [22] | 79.57 | 87.23 | 82.39 | 86.43 | 81.58 | 87.05 | 83.20 | 85.39 | 56.96 | 69.61 | 60.39 | 69.27 |
| w/ours | 80.85 | 87.94 | 82.74 | 86.68 | 82.90 | 87.87 | 84.08 | 86.77 | 58.60 | 71.30 | 61.84 | 69.66 |

| Methods | Tiny-imagenet | | | | ImageNet-100 | | | | ImageNet-1K | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B0_Inc20 | | B100_Inc20 | | B0_Inc10 | | B50_Inc10 | | B0_Inc100 | | B0_Inc200 | |
| | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. | Last | Avg. |
| Finetune | 75.91 | 83.85 | 76.94 | 81.77 | 76.88 | 86.62 | 78.88 | 84.25 | 71.83 | 80.97 | 73.40 | 81.48 |
| w/ours | 76.62 | 84.35 | 77.80 | 82.35 | 77.58 | 87.18 | 80.28 | 84.93 | 73.40 | 82.08 | 74.95 | 82.49 |
| RAPF [18] | 73.73 | 81.51 | 73.45 | 77.83 | 79.48 | 88.42 | 79.68 | 84.60 | 72.80 | 81.67 | 73.09 | 81.41 |
| w/ours | 74.15 | 81.79 | 74.30 | 78.36 | 80.42 | 88.68 | 80.32 | 84.87 | 73.02 | 82.06 | 73.73 | 81.65 |
| ZSCL [51] | 72.78 | 81.48 | 75.64 | 81.51 | 64.32 | 78.51 | 72.70 | 81.42 | 71.49 | 80.61 | 73.88 | 81.84 |
| w/ours | 75.46 | 83.06 | 77.56 | 82.74 | 66.60 | 80.26 | 74.12 | 82.08 | 72.50 | 81.12 | 74.49 | 82.09 |
| MoE4CL [47] | 75.79 | 83.14 | 76.88 | 81.71 | 74.28 | 85.17 | 77.22 | 83.55 | 72.01 | 81.17 | 73.89 | 82.02 |
| w/ours | 76.93 | 84.15 | 78.15 | 82.77 | 75.32 | 85.86 | 78.80 | 84.34 | 72.94 | 81.76 | 74.78 | 82.54 |
| MG-CLIP [19] | 76.68 | 84.04 | 77.09 | 81.41 | 78.14 | 87.20 | 78.34 | 83.99 | 73.33 | 81.71 | 74.25 | 81.77 |
| w/ours | 76.72 | 84.40 | 77.85 | 81.94 | 78.96 | 87.62 | 79.25 | 84.45 | 74.11 | 82.04 | 74.75 | 81.99 |
| DMNSP [22] | 76.39 | 83.97 | 78.05 | 81.62 | 75.22 | 86.38 | 78.18 | 83.76 | 72.87 | 81.57 | 74.55 | 82.22 |
| w/ours | 76.82 | 84.46 | 78.72 | 82.68 | 76.26 | 86.60 | 79.16 | 84.51 | 72.93 | 81.99 | 74.90 | 82.76 |

and adds $n$ classes per step. For ImageNet-1000, we use two full incremental setup that learns all classes sequentially from scratch due to its scale.

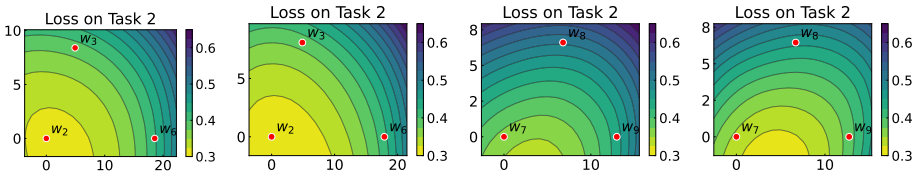
**Compared methods.** We integrate DGM into six representative continual fine-tuning paradigms: 1) full-parameter tuning: ZSCL [51]; 2) adapter-based methods: DMNSP [22], RAPF [18], MOE4CL [47]; and 3) LoRA-based methods: Finetune and MG-CLIP [19]. This integration allows a consistent and fair comparison across architectures.

**Implementation details.** All experiments are implemented in PyTorch and conducted on NVIDIA RTX-4090 GPUs. Each baseline uses the pre-trained CLIP ViT-B/16 weights [34]. Trainable adapter or LoRA modules are inserted into each transformer block following the original settings of their respective




**(a)** SAM **(b)** DGM **(c)** SAM **(d)** DGM

**Fig. 5:** Evolution of the zero-shot loss landscape on ImageNet-100 under sequential distribution shifts across $T_1 - T_6$. (a-b) Visualization of the loss surfaces for SAM, which exhibit sharper curvatures. (c-d) Visualization for DGM, where the loss landscapes are significantly flatter and smoother.




**(a)** SAM **(b)** DGM **(c)** SAM **(d)** DGM

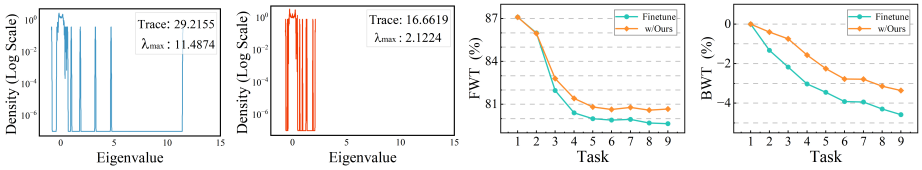
**Fig. 6:** Loss landscape evolution across downstream tasks ($T_2 - T_6$ and $T_7 - T_9$). (a–b) SAM produces sharper loss surfaces for later tasks, indicating reduced stability under sequential distribution shifts. (c–d) DGM maintains flatter and smoother loss basins across the same task sequences, demonstrating improved stability during continual adaptation.

baselines. When applying DGM, we keep the same optimizer, learning rate, batchsize, and number of epochs as in the corresponding baseline to ensure a strictly fair comparison.

**Evaluation metrics.** Let $A_t$ denote the average accuracy over all learned tasks after completing task $T_t$. Average Accuracy (Avg.) defined as $\frac{1}{T}\sum_{t=1}^{T} A_t$, Last Accuracy (Last) represents the accuracy after the final task, $A_T$. To further analyze transfer and forgetting behaviors, we adopt Forward Transfer (FWT) and Backward Transfer (BWT) [40] to quantify the positive influence on future tasks and the retention of previously learned knowledge. Let $R_{i,j}$ denote the accuracy on task $T_j$ after learning task $T_i$, and $b_j$ denote the initial (zero-shot) accuracy on $T_j$ before any training. They are defined respectively as FWT $= \frac{1}{T-1}\sum_{j=2}^{T}(R_{j-1,j} - b_j)$ and BWT $= \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,j} - R_{j,j})$.

### 4.2 Main Results

Tab. 1 reports the main results across all benchmarks under two incremental settings. Integrating DGM into six representative continual fine-tuning methods consistently improves accuracy and reduces forgetting across datasets and




**(a)** Finetune **(b)** w/Ours **(c)** FWT **(d)** BWT

**Fig. 7:** Comparison of Generalization Capability. In (a)(b), We report Hessian eigenvalue distributions and the trace for Finetune w/o and w/ DGM on CIFAR-100. In (c)(d) we show the Forward Transfer (FWT) and Backward Transfer (BWT) across tasks, where DGM indicating better preservation of prior knowledge and stronger transfer to future tasks.

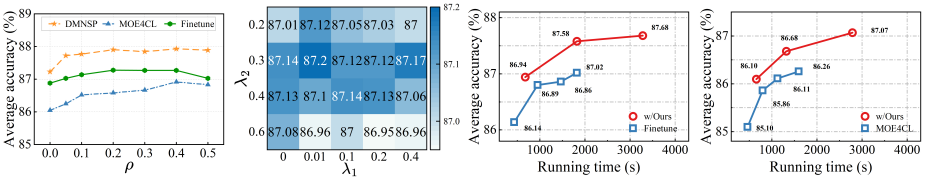
**Table 2:** Comparison of zero-shot ability among three methods on three datasets, *CLIP* is the pretrained model without training.

| Methods | Datasets | | | Avg. | Average Return |
|---|---|---|---|---|---|
| | Food101 | ImageNet-100 | ImageNet-1K | | |
| CLIP | 86.28±0.06 | 73.78±0.14 | 66.01±0.05 | 75.35±0.08 | |
| ZSCL | 82.38±0.10 | 67.50±0.12 | 62.43±0.10 | 70.77±0.10 | |
| w/Ours | **86.69**±0.08 | **74.28**±0.10 | **66.74**±0.05 | **75.90**±0.08 | +5.13 |
| RAPF | 78.78±0.10 | 62.20±0.09 | 56.39±0.12 | 65.79±0.10 | |
| w/Ours | **80.55**±0.10 | **66.70**±0.09 | **57.49**±0.12 | **68.25**±0.10 | +2.46 |
| DMNSP | 83.71±0.06 | 63.26±0.08 | 69.64±0.29 | 72.20±0.14 | |
| w/Ours | **83.84**±0.07 | **65.25**±0.05 | **71.28**±0.14 | **73.46**±0.09 | +1.26 |

architectures. The gains remain stable across datasets with different scales and domain characteristics, as well as under different task partition settings, indicating improved robustness to distribution shift. These results demonstrate that DGM can be effectively integrated into existing continual adaptation pipelines without introducing additional parameters or architectural changes.

### 4.3 Visualizations

To empirically validate that DGM effectively reshapes the optimization landscape, we conduct a comparative visualization of the loss surfaces for both the current task and zero-shot generalization. Specifically, we compare the landscapes generated by standard SAM against those optimized by DGM (incorporating $\mathcal{L}_{ZS}$ or $\mathcal{L}_{NV}$). As illustrated in Fig. 5, DGM successfully flattens the geometry of the zero-shot landscape (evaluated on ImageNet-100) across different task stages. This demonstrates that incorporating $\mathcal{L}_{ZS}$ into the perturbation sensing process encourages the model to preserve the zero-shot behavior of the frozen foundation model. Similarly, the integration of $\mathcal{L}_{NV}$ provides a robust stability margin for the current task. Under the task-specific landscape, DGM yields a significantly wider and smoother minimum compared to standard SAM. This increased flatness suggests that DGM is less sensitive to visual perturbations and domain noise, explaining its superior performance in preventing overfitting. Col-




**(a)** Ablation on $\rho$ **(b)** Ablation on $\lambda_1, \lambda_2$ **(c)** Finetune **(d)** MOE4CL

**Fig. 8:** Ablation experiments. In (a)(b), we conduct ablation experiments on parameter $\rho$, $\lambda_1$ and $\lambda_2$, respectively. Computation overhead. We compare accuracy and training time of Finetune and MOE4CL w/ and w/o DGM.

**Table 3:** Ablation on different combine of perturbation.

| Perturbation | | | CIFAR-100 | | CUB-200 | | Tinyimagenet | |
|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{train}$ | $\mathcal{L}_{NV}$ | $\mathcal{L}_{ZS}$ | Last | Avg. | Last | Avg. | Last | Avg. |
| ✓ | | | 80.42 | 87.38 | 60.40 | 71.18 | 76.20 | 84.20 |
| ✓ | ✓ | | 80.92 | 87.66 | 61.43 | 71.28 | 76.91 | 84.30 |
| ✓ | | ✓ | 81.20 | 87.90 | 61.35 | 71.27 | 76.85 | 84.22 |
| ✓ | ✓ | ✓ | **81.48** | **87.99** | **61.80** | **71.32** | **77.05** | **84.32** |

lectively, these visualizations confirm that DGM achieves a dual-distributional consensus, providing a superior trade-off by ensuring the model maintains a smoother landscape across both pre-trained and task-specific distributions.

### 4.4 Comparison of Generalization Capability

This section evaluates the ability of DGM to preserve generalization during continual adaptation. Our analysis focuses on two aspects: 1)Task transferability: how effectively the model retains and transfers knowledge across sequential tasks, and 2) Zero-shot generalization: whether the zero-shot performance on unseen datasets remains stable after continual fine-tuning. Beyond these behavioral metrics, we also examine the sharpness of the learned solutions by comparing the Hessian eigenvalue distributions and trace in Fig. 7. As can be seen, both the maximum eigenvalue and the trace are substantially smaller when DGM is applied, indicating convergence to flatter and less task-specific regions of the loss landscape. Forward Transfer (FWT) and Backward Transfer (BWT) results, shown in Fig. 7c and Fig. 7d, further corroborate this finding: DGM consistently yields higher FWT and less negative BWT, demonstrating enhanced ability to preserve prior knowledge while facilitating transfer to future tasks.

To assess whether DGM enhances zero-shot generalization during continual fine-tuning, we test models trained on CIFAR-100 against three unseen datasets: Food-101 [4], ImageNet-100, and ImageNet-1K [11]. As shown in Tab. 2, all DGM variants outperform their original versions, with an average gain of +4.45%. This consistent boost that DGM preserves and even strengthens zero-shot generalization under continual updates. Overall, these results show that DGM improves

generalization under sequential perturbations within an unified optimization framework.

### 4.5 Ablation and Sensitivity

To verify the role of each perturbation term, we compare four variants: (1) applying perturbation only to the base task loss, which is equivalent to standard SAM, (2) applying our zero-shot regularization loss $\mathcal{L}_{ZS}$ or noisy regularization loss $\mathcal{L}_{NV}$ individually, and (3) combining all three on Finetune method with three datasets. As summarized in Tab. 3, both $\mathcal{L}_{ZS}$ and $\mathcal{L}_{NV}$ individually improve continual performance, while their combination yields the highest accuracy and lowest forgetting. This confirms that the two objectives are complementary: $\mathcal{L}_{ZS}$ preserves pre-trained generality, and $\mathcal{L}_{NV}$ enhances task-specific robustness, jointly leading to the best overall stability.

We further analyze the sensitivity of DGM to its key hyperparameters: $\rho$, which controls the step length of the inner gradient-ascent perturbation; and $\lambda_1$ and $\lambda_2$, which weight the zero-shot regularization loss $\mathcal{L}_{ZS}$ and the noisy regularization loss $\mathcal{L}_{NV}$, respectively. As shown in Fig. 8, Performance improves for all non-zero values of $\rho$, validating the benefit of controlled perturbation. Both $\lambda_1$ and $\lambda_2$ yield stable improvements within a wide range, demonstrating that DGM maintains strong performance without fine-grained tuning. This robustness to hyperparameter variation further confirms the general applicability of our framework across diverse continual adaptation settings.

### 4.6 Efficiency Analysis

Since DGM introduces additional forward and backward passes for perturbation-based optimization, we evaluate its computational overhead empirically. Benefiting from the share feature extraction between the original and perturbed branches, the practical runtime overhead is substantially reduced. To quantify this, we benchmark both training time and memory usage on identical hardware with CIFAR-100.

Fig. 8 compares runtime accuracy trade-offs across training schedules. DGM incurs only a moderate increase in training time and a negligible memory consumption, while consistently improving average accuracy and reducing forgetting. Notably, DGM achieves comparable or even higher accuracy after only a single epoch whereas the baselines require 3-4 epochs to attain similar performance. These observations indicate that DGM accelerates convergence by steering optimization toward flatter, more stable minima, thereby providing a favorable trade-off between modest computational cost and substantially improved continual-learning performance.

## 5 Conclusion

In this paper, We study the degradation of generalization in vision-language models during continual fine-tuning, where adapting to narrow downstream

tasks gradually erodes pre-trained capabilities. Our analysis shows that although flatness-aware optimization mitigates inter-task interference, pursuing flatness for a single task distribution is insufficient under sequential distribution shifts. To address this limitation, we propose Dual-Generalization-aware Minimization (DGM), which incorporates perturbations derived from both pre-trained knowledge and task-specific variations during optimization. Extensive experiments across multiple benchmarks demonstrate that DGM consistently improves knowledge retention and zero-shot generalization when integrated into diverse continual learning frameworks.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62576268, the Key Research and Development Project in Shaanxi Province No. 2023GXLH-024, and the Project of China Knowledge Centre for Engineering Science and Technology.

## References

1. Experience Replay for Continual Learning. In: Advances in Neural Information Processing Systems. vol. 32, pp. 350–360 (2019)
2. Andriushchenko, M., Flammarion, N.: Towards Understanding Sharpness-Aware Minimization. In: Proceedings of the 39th International Conference on Machine Learning. pp. 639–668 (2022)
3. Bian, A., Li, W., Yuan, H., Yu, C., Wang, M., Zhao, Z., Lu, A., Ji, P., Feng, T.: Make Continual Learning Stronger via C-Flat. In: Advances in Neural Information Processing Systems. vol. 37, pp. 7608–7630 (2024)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – Mining Discriminative Components with Random Forests. In: European Conference on Computer Vision. pp. 446–461 (2014)
5. Budnikov, M., Bykova, A., Yamshchikov, I.P.: Generalization Potential of Large Language Models. Neural Computing and Applications **37**(4), 1973–1997 (2025)
6. Chen, J., Tang, Q., Lu, Q., Fang, S.: MoL for LLMs: Dual-Loss Optimization to Enhance Domain Expertise While Preserving General Capabilities. arXiv preprint arXiv:2505.12043 (2025)
7. Chen, R., Jing, X.Y., Wu, F., Chen, H.: Sharpness-Aware Gradient Guidance for Few-shot Class-incremental Learning. Knowledge-Based Systems **299**, 112030 (2024)
8. Chen, Z., Zhang, J., Kou, Y., Chen, X., Hsieh, C.J., Gu, Q.: Why Does Sharpness-Aware Minimization Generalize Better Than SGD? In: Advances in Neural Information Processing Systems. vol. 36, pp. 72325–72376 (2023)
9. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A Continual Learning Survey: Defying Forgetting in Classification Tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3366–3385 (2022)
10. Deng, D., Chen, G., Hao, J., Wang, Q., Heng, P.A.: Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. In: Advances in Neural Information Processing Systems. vol. 34, pp. 18710–18721 (2021)

11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-scale Hierarchical Image Database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
12. Deng, J., Zhu, Q., Pang, J., Yang, L., Fu, Z., Zhang, B.: Efficiently Seeking Flat Minima for Better Generalization in Fine-Tuning Large Language Models and Beyond. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 20728–20736 (2026)
13. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-Aware Minimization for Efficiently Improving Generalization. In: International Conference on Learning Representations (2021)
14. Goyal, S., Kumar, A., Garg, S., Kolter, Z., Raghunathan, A.: Finetune Like You Pretrain: Improved Finetuning of Zero-Shot Vision Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19338–19347 (2023)
15. Hadsell, R., Rao, D., Rusu, A.A., Pascanu, R.: Embracing Change: Continual Learning in Deep Neural Networks. Trends in Cognitive Sciences **24**(12), 1028–1040 (2020)
16. He, H., Huang, G., Yuan, Y.: Asymmetric Valleys: Beyond Sharp and Flat Local Minima. In: Advances in Neural Information Processing Systems. vol. 32, pp. 2553–2564 (2019)
17. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., others: The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8340–8349 (2021)
18. Huang, L., Cao, X., Lu, H., Liu, X.: Class-Incremental Learning with CLIP: Adaptive Representation Adjustment and Parameter Fusion. In: European Conference on Computer Vision. pp. 214–231 (2025)
19. Huang, L., Cao, X., Lu, H., Meng, Y., Yang, F., Liu, X.: Mind the Gap: Preserving and Compensating for the Modality Gap in CLIP-Based Continual Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3777–3786 (2025)
20. Imanov, O.Y.L.: Mechanistic Analysis of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning. arXiv preprint arXiv:2601.18699 (2026)
21. Jha, S., Gong, D., Yao, L.: CLAP4CLIP: Continual Learning with Probabilistic Finetuning for Vision-Language Models. In: Advances in Neural Information Processing Systems. vol. 37, pp. 129146–129186 (2024)
22. Kang, B., Wang, L., Wu, Z., Feng, T., Li, Y., Gao, Y., Li, W.: Dynamic Multi-Layer Null Space Projection for Vision-Language Continual Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2077–2086 (2025)
23. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In: International Conference on Learning Representations (2017)
24. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
25. Li, T., He, Z., Li, Y., Wang, Y., Shang, L., Huang, X.: Flat-LoRA: Low-Rank Adaptation over a Flat Loss Landscape. In: International Conference on Machine Learning. pp. 34549–34563 (2025)

26. Li, Y., Li, Z.Y., Zeng, Q., Hou, Q., Cheng, M.M.: Cascade-CLIP: Cascaded Vision-Language Embeddings Alignment for Zero-Shot Semantic Segmentation. In: International Conference on Machine Learning. pp. 28243–28258 (2024)
27. Li, Z., Hoiem, D.: Learning without Forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2018)
28. Lin, H., Zhang, B., Feng, S., Li, X., Ye, Y.: PCR: Proxy-Based Contrastive Replay for Online Class-Incremental Continual Learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24246–24255 (2023)
29. Liu, Y., Li, T., Huang, Z., Yang, Z., Huang, X.: Bi-LoRA: Efficient Sharpness-Aware Minimization for Fine-Tuning Large-Scale Models. arXiv preprint arXiv:2508.19564 (2025)
30. Lu, A., Feng, T., Yuan, H., Song, X., Sun, Y.: Revisiting Neural Networks for Continual Learning: An Architectural Perspective. In: International Joint Conference on Artificial Intelligence. pp. 4651–4659 (2024)
31. Mehta, S.V., Patil, D., Chandar, S., Strubell, E.: An Empirical Investigation of the Role of Pre-training in Lifelong Learning. Journal of Machine Learning Research **24**(214), 1–50 (2023)
32. Ni, Z., Wei, L., Tang, S., Zhuang, Y., Tian, Q.: Continual Vision-Language Representation Learning with Off-Diagonal Information. In: International Conference on Machine Learning. pp. 26129–26149 (2023)
33. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual Lifelong Learning with Neural Networks: A Review. Neural Networks **113**, 54–71 (2019)
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models From Natural Language Supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
35. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: Incremental Classifier and Representation Learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
36. Saha, G., Garg, I., Roy, K.: Gradient Projection Memory for Continual Learning. In: International Conference on Learning Representations (2021)
37. Shi, G., CHEN, J., Zhang, W., Zhan, L.M., Wu, X.M.: Overcoming Catastrophic Forgetting in Incremental Few-Shot Learning by Finding Flat Minima. In: Advances in Neural Information Processing Systems. vol. 34, pp. 6747–6761 (2021)
38. Sun, W., Li, Q., Zhang, J., Wang, W., Geng, Y.A.: Decoupling Learning and Remembering: a Bilevel Memory Framework with Knowledge Projection for Task-Incremental Learning. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20186–20195 (2023)
39. Sun, Z., Mu, Y., Hua, G.: Regularizing Second-Order Influences for Continual Learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20166–20175 (2023)
40. Thengane, V., Khan, S., Hayat, M., Khan, F.: CLIP Model is an Efficient Continual Learner. arXiv preprint arXiv:2210.03114 (2022)
41. Uppal, K., Tankala, P.K.: Foundational Models Must Be Designed To Yield Safer Loss Landscapes That Resist Harmful Fine-Tuning. In: ICML 2025 Workshop on Reliable and Responsible Foundation Models (2025)
42. van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three Types of Incremental Learning. Nature Machine Intelligence **4**(12), 1185–1197 (2022)
43. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-Ucsd Birds-200-2011 Dataset (2011)

44. Wang, L., Zhang, X., Su, H., Zhu, J.: A Comprehensive Survey of Continual Learning: Theory, Method and Application. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(8), 5362–5383 (2024)
45. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust Fine-Tuning of Zero-Shot Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7959–7971 (2022)
46. Yang, E., Shen, L., Wang, Z., Liu, S., Guo, G., Wang, X., Tao, D.: Revisiting Flatness-Aware Optimization in Continual Learning With Orthogonal Gradient Projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(5), 3895–3907 (2025)
47. Yu, J., Zhuge, Y., Zhang, L., Hu, P., Wang, D., Lu, H., He, Y.: Boosting Continual Learning of Vision-Language Models via Mixture-of-Experts Adapters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23219–23230 (2024)
48. Zhang, G., Wang, L., Kang, G., Chen, L., Wei, Y.: SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19148–19158 (2023)
49. Zhang, W., Huang, Y., Zhang, W., Zhang, T., Lao, Q., Yu, Y., Zheng, W.S., Wang, R.: Continual Learning of Image Classes With Language Guidance From a Vision-Language Model. IEEE Transactions on Circuits and Systems for Video Technology **34**(12), 13152–13163 (2024)
50. Zhang, X., Xu, R., Yu, H., Zou, H., Cui, P.: Gradient Norm Aware Minimization Seeks First-Order Flatness and Improves Generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20247–20257 (2023)
51. Zheng, Z., Ma, M., Wang, K., Qin, Z., Yue, X., You, Y.: Preventing Zero-shot Transfer Degradation in Continual Learning of Vision-Language Models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19125–19136 (2023)
52. Zhou, D.W., Li, K.W., Ning, J., Ye, H.J., Zhang, L., Zhan, D.C.: External Knowledge Injection for CLIP-Based Class-Incremental Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3314–3325 (2025)
53. Zhou, D.W., Wang, Q.W., Ye, H.J., Zhan, D.C.: A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning. In: International Conference on Learning Representations (2023)
54. Zhou, D.W., Zhang, Y., Wang, Y., Ning, J., Ye, H.J., Zhan, D.C., Liu, Z.: Learning Without Forgetting for Vision-Language Models. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(6), 4489–4504 (2025)