SAMPLe: A Sharpness Aware Minimization based Optimizer for Prompt Learning in Vision-Language Models

Hossein Rajoli[1,2]*, Fatemeh Lotfi[1]*, Niloufar Alipour[1], Hossein Kashiani[1], Xiaolong Ma[3], and Fatemeh Afghah[1]

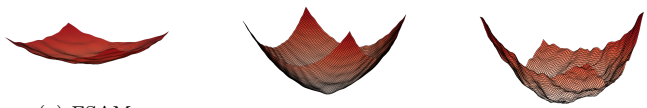
[1] Clemson University, Clemson SC 29634, USA
{hrajoli,flotfi, nalipou, hkashia, fafghah}@clemson.edu
[2] Siemens Energy (AI Lab), Orlando FL 32826, USA
hossein.rajoli.nowdeh@siemens-energy.com
[3] University of Arizona, Tucson AZ 85721, USA
xiaolongma@arizona.edu

Abstract. Pre-trained Vision-Language Models (VLMs) like CLIP have proven highly effective as foundation models for various downstream applications. However, prompt learning in VLMs encounters a performance-generalization dilemma: while prompts can be tuned to achieve high accuracy on seen distributions, this tuning process often undermines their generalizability to unseen data. The limited set of learnable prompts, which contextualize and condition the input to steer it toward the task within the pretrained VLM, tends to overfit the training data, leading to a trade-off between task-specific performance and preserving generalization. To address this dilemma, we introduce SAMPLe (Sharpness-Aware Minimization Prompt Learning), a plug-in sharpness-aware optimizer that enhances prompt generalizability by accounting for loss landscape sharpness. Unlike conventional methods, SAMPLe balances exploration and exploitation by satisfying objective function constraints at each step, dynamically adapting to the current optimization state based on the local curvature and gradient properties. This approach reduces overfitting on seen distributions and improves adaptability to unseen data, preserving the generalization potential of pre-trained VLM models. We integrate SAMPLe into multiple prompt learning frameworks, including CoOp, CoCoOp, MaPLe, TCP, and Co-Prompt, demonstrating its effectiveness across diverse methods. Experiments show that SAMPLe elevates prompt learning frameworks and consistently outperforms existing optimizers across diverse settings, establishing itself as a robust, model-agnostic solution for prompt learning.

1 Introduction

Vision-Language Models (VLMs) such as CLIP have become essential for a wide range of visual tasks due to their robust multimodal understanding, leveraging both vision and text to achieve notable zero-shot performance across domains [18,

* Hossein Rajoli and Fatemeh Lotfi contributed equally to this work.



(a) FSAM (b) SAGM (c) SAMPLe

Fig. 1: Normalized loss landscapes of CoOp [47] on ImageNet using F-SAM [25], SAGM [38], and the proposed SAMPLe, each scaled by the maximum absolute loss (among FSAM, SAGM, and SAMPLe) preserving relative depth and sharpness. The visualization demonstrates SAMPLe's effectiveness in achieving both flatter minima and lower empirical risk.

33]. For downstream applications, prompt learning has emerged as an efficient alternative to traditional fine-tuning by introducing learnable prompts that adapt VLMs to specific tasks while keeping their parameters fixed [47, 48]. However, a core challenge persists in balancing high task-specific accuracy while retaining generalization to unseen classes [8, 19] under the constraint of a limited number of learnable parameters.

This restriction amplifies sensitivity in optimization, since updates are confined to a narrow parameter subspace of the learnable prompts. Authors in [31] showed that smaller capacity models often converge to solutions that are less robust to perturbations, leading to weaker generalization. On the other hand, [19] further linked such generalization gaps to sharp minima characterized by large positive eigenvalues of the Hessian, while [16] emphasized that even with low training error, a large train–test gap persists under sharp solutions. These findings highlight that the limited learnable space in prompt learning makes models especially vulnerable to sharp, high-curvature minima in the loss landscape. Although various techniques [20, 40, 47, 48, 50] have been proposed to preserve the generalizability of VLMs while adapting them to downstream tasks, just [26] considered role of gradient manipulation with respect to the loss landscape.

Incorporating model-agnostic loss landscape awareness into VLM prompt learning can substantially enhance generalization. By guiding optimization toward flatter and more stable minima, the model can produce prompts that are both robust and transferable across tasks. However, because learnable prompts involve only a small number of tunable parameters, they are especially prone to converging to sharp, non-robust minima. While flatter landscapes are crucial for generalization, prior work shows they may come at the cost of reduced accuracy on seen distributions [38, 51]. Thus, achieving effective prompts requires carefully balancing the minimization of training loss (exploitation) with the search for sufficiently flat regions of the landscape (exploration), ensuring strong performance on both seen and unseen classes.

To address these challenges, we propose SAMPLe (Sharpness-Aware Minimization Prompt Learning), a framework specifically designed to incorporate sharpness-aware minimization (SAM) into prompt learning. SAMPLe directly targets the trade-off between minimizing loss values and maintaining a flat loss landscape, ensuring both accuracy and generalization. It introduces three es-

sential strategies to achieve this balance. First, SAMPLe optimizes for a loss minimum that is sufficiently low in training loss and flat in the loss landscape, providing both stability and robustness. Second, it ensures that SAM gradient updates, computed at the deviated point (see Sec.4), are coherently aligned with empirical risk minimization (ERM) gradients, as illustrated in Fig.2a and Fig.2b. This alignment addresses the instability caused by the limited parameter space of prompts. Finally, SAMPLe constrains the SAM gradient updates at the perturbed point to be orthogonal to the full-batch gradient (refer to Sec.3.1, Sec.4), effectively aligning it with a relaxed version of the ERM gradient that seeks the flattest minima. Satisfying both constraints of alignment to the ERM gradient and its relaxed version in each iteration is the central objective of the SAMPLe optimizer, ensuring a precise adaptive balance between exploration and exploitation.

These dual objectives are critical for effective prompt learning. However, sharpness-aware minimization optimizers, including SAM, F-SAM [25], and SAGM [38] (see Fig.2a and Fig.3), fail to balance both conditions. Beyond these optimizers, a recently published study [26] proposed GCSCoOP, a SAM-based method designed to address the generalization problem by considering both loss value and loss sharpness. However, it suffers from limited scalability, as it relies on a heuristic gradient computation with two hyperparameters (β_1, β_2). In contrast, SAMPLe updates gradients by automatically balancing exploitation and exploration constraints considering the current state of optimization. The key contributions of this work can be summarized as follows:

- We introduce SAMPLe, a SAM-based optimizer specifically designed for VLM prompt learning. It comprehensively defines a dual-objective optimization framework that adaptively balances exploiting low ERM loss and exploring the flattest minima at every iteration.
- We analyze the limitations of existing sharpness-aware optimization methods in VLM prompt learning and establish the necessity of an adaptive dual-objective approach, motivating the design of SAMPLe.
- We demonstrate the superior performance of SAMPLe over SOTA in VLM prompt learning and provide an ablation study comparing it with related sharpness-aware optimizers.

2 Related Work

Prompt learning is a powerful approach for adapting large pre-trained VLMs like CLIP [33], ALIGN [18], LiT [44], and FILIP [42] to downstream tasks. These models, trained on large image-text datasets using contrastive learning [3], excel at general representation learning [45], but adapting them to specific tasks, especially in low-data settings like few-shot learning [47], remains challenging. Traditional fine-tuning methods are computationally expensive and prone to overfitting [21]. Prompt learning addresses these issues by introducing learnable prompts, enabling efficient task adaptation without retraining the entire model. CoOp [48] fine-tunes continuous text prompts for few-shot recognition but struggles with

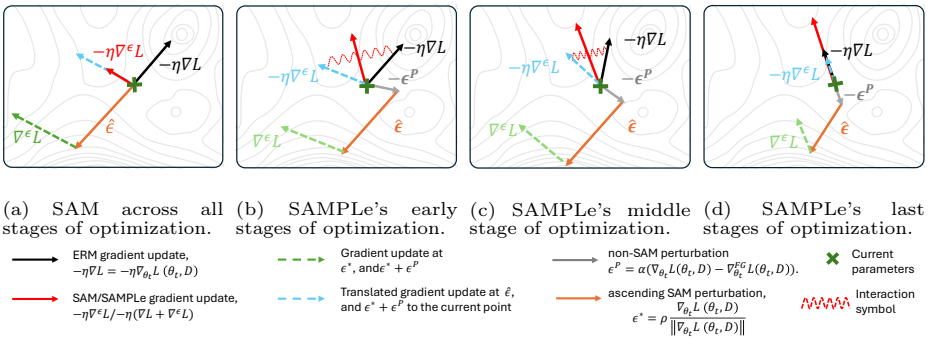


Fig. 2: SAMPLe vs. vanilla SAM: Unlike SAM, which applies uniform gradient updates, SAMPLe dynamically adjusts gradients across stages, promoting smoother optimization in early stages and robust convergence in later stages, resulting in improved generalization and performance.

unseen classes. CoCoOp [47] improves zero-shot generalization by conditioning prompts on image features, while MaPLe [21] optimizes prompts across both vision and language branches. Methods like PromptSRC [21] and ProDA [28] further enhance generalization using regularization and diverse task-related representations. Recent works extend this line with textual-based class-aware prompt tuning, TCP [41]), decoupled embedding parameterization, DEPT [15]), overfitting-aware prompt regularization, LOBG [7]), and diversity covariance-aware prompt learning [49].

SAM is an effective optimization framework designed to improve model generalization by finding flatter minima in the loss landscape [11]. SAM has been successfully applied in language modeling [1], fluid dynamics [17], medical imaging [43], multimodal learning [34], and Reinforcement Learning (RL) [27]. Building on the original framework, several extensions have refined SAM's adaptability. F-SAM [25] utelizes a less aggressive perturbation that leads to better generalization, while SAGM [38] enhances gradient alignment to balance loss minimization and flatness. GCSCoOP [26], a recent SAM-based approach, aims to address generalization by considering both loss sharpness and value but remains dependent on manually set hyperparameters. Adaptive SAM (ASAM) adjusts the sharpness radius dynamically [24], GSAM simplifies sharpness measurement using surrogate gap calculations [51], and Fisher SAM leverages Fisher information for sharper neighborhood estimations [22]. GAM further improves generalization by focusing on the maximal gradient norm [46]. In vision tasks, SAM has demonstrated notable success in models like Vision Transformers and MLP-Mixers [4], enhancing both accuracy and robustness. However, SAM's potential remains insufficiently explored in multi-modal learning.

3 Preliminaries

Prompt Learning in VLM: Prompt learning aims to enhance the adaptability of VLMs by learning task-specific prompts. This approach involves introducing tunable token vectors $\mathbf{v} = \{v_i\}_{i=1}^{N}$, where the text inputs for each class are

represented as $t_m = v_1, \ldots, v_N, c_m$, with c_m denoting the class name and where $N+1$ defines number of tokens for text input. These token vectors are optimized using cross-entropy loss $\mathcal{L}_{CE}$, while keeping the parameters of the underlying VLM model, such as CLIP, frozen:

$$L_{CE}(\mathbf{v}) = -\sum_{m} y_m \log p(m|x), \tag{1}$$

$$p(m|x) = \frac{\exp(\cos(\mathcal{I}(x), \mathcal{T}(t_m))/\tau)}{\sum_{j=1}^{M} \exp(\cos(\mathcal{I}(x), \mathcal{T}(t_j))/\tau)}. \tag{2}$$

The learned token vectors enable the model to adapt to specific tasks by refining how class names are represented as input to the model, improving the prompt's alignment with task requirements.

Sharpness-aware Prompt Learning: SAM is a recently proposed optimization framework designed to enhance generalization by avoiding sharp minima in the loss landscape [10]. Consider a deep neural network $f(x;\theta)$ parameterized by $\theta \in \mathbb{R}^d$, with a loss function $\ell(f(x;\theta), y)$ that measures the discrepancy between the model's prediction $f(x;\theta)$ and the true label y. For a dataset $\mathcal{D} = (x_i, y_i)_{i=1}^{N}$, the empirical loss over the dataset is given by:

$$L(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i;\theta), y_i). \tag{3}$$

Optimization methods like stochastic gradient descent (SGD) often converge to sharp minima that generalize poorly. SAM mitigates this by solving the following min-max optimization problem:

$$\min_{\theta} \max_{\|\epsilon\|_2 \leq \rho} L(\theta + \epsilon; \mathcal{D}), \tag{4}$$

where $\epsilon \in \mathbb{R}^d$ is a perturbation constrained by $\|\epsilon\|_2 \leq \rho$, with ρ controlling the radius of the perturbation around the parameter θ. Because solving the inner maximization directly is computationally expensive, SAM uses a first-order Taylor expansion to approximate the perturbation $\epsilon^\star$:

$$\epsilon^\star \approx \rho \cdot \frac{\nabla L(\theta; \mathcal{D})}{\|\nabla L(\theta; \mathcal{D})\|_2}. \tag{5}$$

The network parameters are updated as follows:

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \nabla \mathcal{L}(\theta_t + \epsilon^\star; \mathcal{D}), \tag{6}$$

where η is the learning rate, and $\epsilon^\star$ is the perturbation that guides the model towards flatter minima in Fig. 2a.

The limited number of learnable parameters in prompt learning often makes traditional approaches susceptible to overfitting or getting trapped in sharp loss surfaces [50]. SAM addresses this challenge by guiding the optimization

process toward flatter minima in the loss landscape, enhancing the model's ability to generalize to unseen distributions and domains. By integrating SAM into the VLM prompt learning frameworks, we effectively improve the model's generalization across unseen domains while maintaining good performance on the training dataset.

3.1 Full-Batch Bias Hinders SAM

One of the core ideas behind SAM is its perturbation vector, ϵ, which plays a crucial role in guiding the model toward flatter minima, thereby enhancing generalization. When applied effectively, the perturbation vector steers the optimization process toward regions of the loss landscape that are less sensitive to sharp parameter changes. However, when the perturbation vector is computed as if all training samples were processed in a single batch (i.e., a full-batch perturbation), it can diminish the natural stochasticity of a mini batch based approach. This stochasticity is crucial for balancing exploration across the loss landscape. As noted by [25], this lack of stochastic variability in full-batch perturbation can limit generalization, even resulting in worse performance compared to SGD in some cases.

While the full gradient provides a global view of the entire dataset, this global smoothness often comes at the expense of exploration, particularly when the perturbation ϵ is excessively aligned with the full gradient. This alignment suppresses the stochastic variability introduced by mini-batch gradients, leading to over-smoothing during training. As a result, the model may converge to sharper minima, reducing generalization. This phenomenon can cause SAM to underperform even compared to traditional optimization methods like SGD [25]. The full-gradient perturbation in the SAM context captures the overall trend of the loss function across the entire dataset. The projection of the mini-batch gradient onto the full batch gradient demonstrated in Fig. 4 (in Appendix) can be formulated as:

$$\text{Proj}_{\nabla^{\mathcal{F}} L(\theta;\mathcal{D})} \nabla L(\theta;\mathcal{D}) = \frac{\|\nabla L(\theta;\mathcal{D})\|_2}{\|\nabla^{\mathcal{F}} L(\theta;\mathcal{D})\|_2} \cos\big(\nabla^{\mathcal{F}} L(\theta;\mathcal{D}), \nabla L(\theta;\mathcal{D})\big) \nabla L^{\mathcal{F}}(\theta;\mathcal{D}), \tag{7}$$

where $\cos(\cdot)$ represents the cosine similarity between the mini-batch gradient, $\nabla L(\theta;\mathcal{D})$, and the full batch gradient vector, $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$. The batch-specific gradient is then defined as:

$$\nabla^{\mathcal{B}} L(\theta;\mathcal{D}) = \nabla L(\theta;\mathcal{D}) - \sigma\ \xi\ \nabla^{\mathcal{F}} L(\theta;\mathcal{D}), \tag{8}$$

where ξ and $\sigma = \frac{\|\nabla L(\theta;\mathcal{D})\|_2}{\|\nabla^{\mathcal{F}} L(\theta;\mathcal{D})\|_2}$ are cosine similarity operator and normalization factor, respectively. This batch-specific gradient represents the component of the mini-batch gradient orthogonal to the full batch gradient. Due to the computational overhead of computing the full gradient over the entire dataset, it is typically approximated using an exponentially moving average (EMA) of the mini-batch gradients:

$$m_t = \lambda m_{t-1} + (1-\lambda)\nabla L(\theta;\mathcal{D}), \tag{9}$$

where m_t approximates $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ if t is large enough, and λ controls the influence of past gradients [25]. The full-gradient component biases the perturbation vector ϵ based on the entire dataset's loss landscape, leading to smoother updates but reducing the exploration necessary for finding flatter minima. This smoothing effect can hinder SAM's generalization performance by diminishing the stochastic variability introduced by mini-batch gradients [25].
Therefore, balancing the stability offered by full-batch gradient updates with the exploration provided by batch-specific gradients is essential for maintaining SAM's effectiveness.

4 SAMPLe

Given the challenges of VLM prompt learning, we introduce two pivotal challenges to ensure that the model performs robustly across both seen and unseen distributions: (i) ***Optimal minima*:** The learned prompts must not only identify a flat minimum in the loss surface but also ensure that this minimum corresponds to a sufficiently low loss value, guaranteeing well-optimized performance on the training samples. (ii) ***Generalization to unseen domains*:** While prompts may perform well on the training data, it is critical to ensure they generalize effectively to unseen distributions. This challenge is amplified by the significantly smaller number of parameters in prompts compared to overparameterized neural networks, making it more challenging to train models that generalize across diverse domains. To address this, the optimization strategy enforced by the objective function must inherently balance exploitation and exploration, dynamically adapting to the current state of the optimization. This ensures that the learning process acquires network parameters capable of achieving both a sufficiently low loss on the training data and robust generalization to unseen domains and distributions. Inspired by [38] and designed to satisfy these two conditions simultaneously, we propose the SAMPLe objective function to optimize the training of learnable prompts as follows:

$$\begin{aligned}\min_{\theta} \mathcal{L}(\theta;\mathcal{D}) \; &= \min_{\theta} \Big[L(\theta;\mathcal{D}) + L\left(\theta + \epsilon^{\star} - \alpha(\nabla^{\mathcal{B}}\mathcal{L}(\theta;\mathcal{D}))\right) \Big] = \\ &\min_{\theta} \Big[L(\theta;\mathcal{D}) + L\left(\theta + \epsilon^{\star} - \alpha(\nabla L(\theta;\mathcal{D}) - \xi\ \sigma\ \nabla^{\mathcal{F}} L(\theta;\mathcal{D})); \mathcal{D}\right) \Big],\end{aligned} \tag{10}$$

where the optimal perturbation is defined by Eq. 5 as $\epsilon^{\star} = \rho \cdot \frac{\nabla L(\theta;\mathcal{D})}{\|\nabla L(\theta;\mathcal{D})\|_2}$.

4.1 SAMPLe Analysis and Algorithm:

This section presents an analysis to clarify the proposed SAMPLE method. Applying the Taylor expansion reveals that using $\nabla^{\mathcal{B}} L(\theta;\mathcal{D})$ in the second term of the loss function implies two critical conditions that enhance the learning process:

Algorithm 1 SAMPLe algorithm

Require: Few-shot training set: $\mathcal{D} = \bigcup_{i=1}^{N}\{\bigcup_{m=1}^{M}(x_m^i, y^i)\}$, number of classes M, class name $\mathbf{c} = \{c_j\}_{j=1}^{M}$, pre-trained CLIP with image-encoder $\mathcal{I}$ and text encoder $\mathcal{T}$, prompt token length P, training epoch E, learning rate η, SAM perturbation radius ρ, α, weight decay coefficientλ, batch-size, b.

Ensure: trained learnable prompts.

Initialize parameters θ_0, $t = 0$
while not converged **do**
  Sample mini-batch $\mathcal{B} \in \mathcal{D}$;
  Compute the training mini-batch loss gradient $\nabla L(\theta_t; \mathcal{D})$;
  update the full-gradient, $m_t \approx \nabla^{\mathcal{F}} L(\theta_t; \mathcal{D})$ using Eq. 9
  Compute SAM perturbation ϵ_t^*, according to Eq. 5:
  Compute $\xi_t = \cos\big(\nabla^{\mathcal{F}} L(\theta_t; \mathcal{D}), \nabla L(\theta_t; \mathcal{D})\big)$.
  Compute normalization factor, $\sigma_t = \frac{\|\nabla \mathcal{L}(\theta; \mathcal{D})\|_2}{\|\nabla^{\mathcal{F}} L(\theta; \mathcal{D})\|_2}$.
  Compute objective optimization, $\mathcal{L}(\theta_t; \mathcal{D})$, using Eq. 10
  Update weights: $\theta_t \leftarrow \theta_t - \eta_t \nabla \mathcal{L}(\theta_t; \mathcal{D})$
  $t \leftarrow t + 1$
end while

$$
\begin{aligned}
&\min_{\theta} L_p\left(\theta - \alpha \nabla^{\mathcal{B}} L(\theta; \mathcal{D})\right) = \\
&\min_{\theta} \Big[L_p(\theta; \mathcal{D}) - \alpha\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla L(\theta; \mathcal{D})\big) + \\
&\alpha \xi \sigma\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta; \mathcal{D})\big) + R(\nabla^{\mathcal{B}} L(\theta; \mathcal{D})^2) \Big] \approx \\
&\min_{\theta} \Big[L_p(\theta; \mathcal{D}) - \alpha \nabla L_p(\theta; \mathcal{D}) \cdot \nabla L(\theta; \mathcal{D}) + \alpha \xi \sigma \nabla L_p(\theta; \mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta; \mathcal{D}) \Big]
\end{aligned}
\tag{11}
$$

where L_P, represents the loss at the perturbed point in parameter space;

$$
\begin{aligned}
\min_{\theta} \Big[& L(\theta; \mathcal{D}) + L_p(\theta; \mathcal{D}) - \alpha\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla L(\theta; \mathcal{D})\big) + \\
& \alpha \xi \sigma\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta; \mathcal{D})\big) \Big].
\end{aligned}
\tag{12}
$$

The objective function can be interpreted as follows:
First, minimizing the loss at the current point (L) and the perturbed neighboring point (L_p). **Second**, considering the alignment between the gradient at the perturbed point and the gradient at the current point, represented as $-\alpha\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla L(\theta; \mathcal{D})\big)$, which we refer to as *exploitation*. **Third**, ensuring that the gradient at the perturbed point is orthogonal to the full-batch gradient or aligned with the batch-specific gradient (see Fig. 4 in Appendix and subsection 4.2), represented as $+\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta; \mathcal{D})\big)$, which we refer to as *exploration*.

4.2 Gradient Orthogonality Analysis:

The third term of Eq. 12 is designed to drive $\nabla L_p(\theta; \mathcal{D})$ orthogonal to $\nabla^{\mathcal{F}} L(\theta; \mathcal{D})$ at the original point θ_t, as we establish below. By Eq. 8, the mini-batch gradient decomposes as $\nabla L(\theta; \mathcal{D}) = \nabla^{\mathcal{B}} L(\theta; \mathcal{D}) + \xi\sigma \nabla^{\mathcal{F}} L(\theta; \mathcal{D})$, so the second term of

Eq. 12 alone would pull $\nabla L_p(\theta;\mathcal{D})$ toward both components. The third term introduces anti-alignment with $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ (Eq. 10): at every iteration, the second term drives $\nabla L_p(\theta;\mathcal{D})$ toward $\nabla L(\theta;\mathcal{D})$ for performance on seen distributions, while the third term resists the full-batch component $\xi\sigma\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ within that alignment. The equilibrium of these two forces steers $\nabla L_p(\theta;\mathcal{D})$ exclusively toward $\nabla^{\mathcal{B}} L(\theta;\mathcal{D})$. Since $\nabla^{\mathcal{B}} L(\theta;\mathcal{D}) \perp \nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ by construction (Eq. 8 and Fig. 4, Appendix), this equilibrium guarantees orthogonality of $\nabla L_p(\theta;\mathcal{D})$ to $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ at θ_t.
Substituting $\nabla^{\mathcal{B}} L(\theta;\mathcal{D}) = \nabla L(\theta;\mathcal{D}) - \xi\sigma\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ (Eq. 8) into the joint second and third terms of Eq. 12:

$$
\begin{aligned}
&- \alpha\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla L(\theta;\mathcal{D})\big) + \alpha\xi\sigma\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta;\mathcal{D})\big) \\
&= -\alpha\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{B}} L(\theta;\mathcal{D})\big) \qquad (13)\\
&\quad \underbrace{- \alpha\xi\sigma\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta;\mathcal{D})\big) + \alpha\xi\sigma\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{F}} L(\theta;\mathcal{D})\big)}_{=0} \\
&= -\alpha\big(\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{B}} L(\theta;\mathcal{D})\big). \qquad (14)
\end{aligned}
$$

The $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ component cancels, and the objective reduces to maximizing $\nabla L_p(\theta;\mathcal{D}) \cdot \nabla^{\mathcal{B}} L(\theta;\mathcal{D})$ exclusively. Since $\nabla^{\mathcal{B}} L(\theta;\mathcal{D}) \perp \nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ by construction (Eq. 8), $\nabla L_p(\theta;\mathcal{D})$ is driven toward the subspace orthogonal to $\nabla^{\mathcal{F}} L(\theta;\mathcal{D})$ at θ_t. To confirm this is preserved under $\epsilon^\star$, let $\hat{\theta}_t = \theta_t + \epsilon_t^\star - \alpha_t \nabla^{\mathcal{B}} L(\theta_t;\mathcal{D})$ denote the perturbed point. By the triangle inequality, $\|\epsilon_t^\star\| \le \rho_t$ (Eq. 5) and condition (i) in Sec. 4.3:

$$\|\hat{\theta}_t - \theta_t\| \le \rho_t + \alpha_t \nabla\mathcal{L}_{\max}. \qquad (15)$$

By K-Lipschitz gradient condition (ii) in Sec. 4.3 and Lemma 1 (Appendix, Proof of Theorem 1):

$$\big|(\nabla L_p(\theta;\mathcal{D}) - \nabla L(\theta;\mathcal{D})) \cdot \nabla^{\mathcal{F}} L(\theta;\mathcal{D})\big| \le K(\rho_t + \alpha_t \nabla\mathcal{L}_{\max}) \cdot \nabla L_{\max}(1-\lambda^T). \qquad (16)$$

As $\rho_t, \alpha_t = \mathcal{O}(1/\sqrt{t})$ and $\lambda^T \to 0$ as $t \to \infty$, the right-hand side vanishes, confirming orthogonality is preserved throughout training.

4.3 Convergence of SAMPLe:

Assuming that the objective function defined in Eq. 10 satisfies the following conditions;

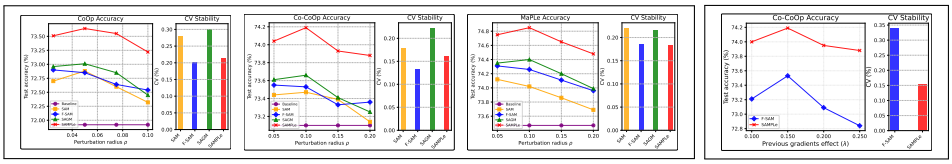
(i) The gradient of the loss function $\nabla\mathcal{L}(\theta_t;\mathcal{D})$ is bounded, i.e.,

$$\|\nabla\mathcal{L}(\theta_t;\mathcal{D})\| \le \nabla\mathcal{L}_{max}, \quad \forall t. \qquad (17)$$

(ii) The stochastic gradient is K-Lipschitz, i.e.,

$$\|\nabla\mathcal{L}(\theta_t;\mathcal{D}) - \nabla\mathcal{L}(\theta_t';\mathcal{D})\| \le K\|\theta_t - \theta_t'\|, \forall(\theta_t, \theta_t'). \qquad (18)$$

Let the learning rate and both perturbations radii be defined as $\eta_t = \frac{\eta_0}{\sqrt{t}}$, $\alpha_t = \frac{\alpha_0}{\sqrt{t}}$,



(a) Comparison of F-SAM, SAM, SAGM, and SAMPLe across ρ values, deployed on CoOp, CoCoOp, and MaPLe. **(b)** SAMPLe vs F-SAM across λ values.

Fig. 3: Accuracy and coefficient of variation of SAM, F-SAM, SAGM, and SAMPLe on ImageNet across different values of ρ and λ for various prompt learning methods, including CoOp, Co-CoOp, and MaPLe.

and $\rho_t = \frac{\rho_0}{\sqrt{t}}$, respectively. Theorem. 1 in Appendix. 8.1 proves that the objective function satisfies following inequality

$$\frac{1}{T}\sum_{t=1}^{T}\left[\|\nabla\mathcal{L}(\theta_t;\mathcal{D})\|^2\right] \leq \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right), \tag{19}$$

It means the objective function defined in Eq. 10 converges with a rate of $\mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$, which is comparable with optimization methods such as SGD and Adam.

5 Experiments

In this section, we elaborate on the datasets, baselines, and implementation details utilized in our study. Next, we evaluate the proposed method on base-to-new class generalization (Sec. 5.2), cross-dataset generalization (Sec. 5.3), and cross-domain generalization (Sec. 5.4).

5.1 Datasets

In this work, we utilize 11 publicly available image classification datasets as downstream tasks: ImageNet [6], Caltech101 [9], OxfordPets [32], StanfordCars [23], Flowers102 [30], Food101 [2], FGVCAircraft [29], SUN397 [39], DTD [5], EuroSAT [12], and UCF101 [36]. Also, four datasets of ImageNetV2 [35], ImageNet-Sketch [37], ImageNet-A [14], and ImageNet-R [13] serve exclusively as target domains for cross-domain generalization. Detailed dataset descriptions are provided in Supplementary 8.3. For each baseline we strictly adhere to the configurations reported in the original papers, including learning rate schedules, backbone architecture, prompt length, prompt initialization, few-shot setup, and three random seeds. Detailed of hyperparameter and implementation descriptions are provided in Supplementary 8.6.

5.2 Base-to-New Generalization Setting

To evaluate the generalization capability of our method, we train the model on base classes and test its performance on both base and novel classes. This

experimental setup is designed to evaluate the trade-off between maintaining strong performance on the base classes while adapting effectively to novel classes. The results are presented in Table 1, with the harmonic mean (HM) of base and novel accuracies serving as the primary metric for comparison. As shown in Table 1, integrating SAMPLe consistently improves the harmonic mean (HM) across different prompt-learning backbones and datasets, indicating a more balanced generalization between base and novel classes. On the averaged results (Table 1(a)), SAMPLe achieves the highest HM for all methods. For example, CoOp improves from 77.65 (+SAGM) to 78.88 with SAMPLe, CoCoOp from 78.28 to 79.51, MAPLe from 79.96 to 80.28, CoPrompt from 81.49 to 82.02, and TCP from 81.40 to 81.95. These improvements are mainly associated with higher accuracy on the new classes while maintaining competitive performance on the base classes. A similar trend is observed across individual datasets. On Flowers102, SAMPLe provides the best HM across all backbones, for instance improving CoCoOp from 85.26 (+SAGM) to 87.15 and TCP from 86.48 to 87.02. On DTD, SAMPLe increases the HM from 65.46 to 69.77 for CoOp and from 70.93 to 71.37 for MAPLe. On EuroSAT, SAMPLe again achieves the highest HM values, reaching 82.44 for CoOp and 84.19 for TCP. Overall, the consistent improvements across methods and datasets suggest that SAMPLe leads to more stable optimization behavior and improved generalization to novel classes without drastically degrading base-class performance. Another highlighted fact is that, superiority of SAMPLe is almost clear over all mentioned methods specifically in HM, which means SAMPLe keeps both Base and New performance and does not sacrify one of them in favor of the other.

Comparison with State-of-the-Art Methods: In addition to comparisons with SAM-based optimizers, we evaluate CoOp+SAMPLe against several representative prompt-learning approaches, including CoOp, CoCoOp, MAPLe, and CoPrompt. As shown in Table 1, CoOp+SAMPLe achieves competitive and often superior harmonic mean (HM) performance across several datasets when compared with the corresponding naive prompt-learning methods. For example, on OxfordPets, CoOp+SAMPLe achieves an HM of 97.08, surpassing the naive variants of CoCoOp (96.43), MAPLe (96.58), and CoPrompt (96.87). This result suggests that SAMPLe can effectively improve generalization while maintaining strong base-class performance. On more challenging datasets such as FGVCAircraft, CoOp+SAMPLe obtains an HM of 37.58, outperforming MAPLe (36.50) and approaching the performance of CoPrompt (39.76). These results indicate that the proposed optimizer remains effective even in fine-grained recognition scenarios where intra-class variations are large and training samples are limited. Similarly, on StanfordCars, CoCoOp+SAMPLe achieves an HM of 75.51, outperforming CoCoOp+FSAM (74.85) and CoCoOp+SAGM (74.75). While maintaining comparable base accuracy, SAMPLe improves performance on the novel classes (77.89), leading to a higher harmonic mean. More broadly, the averaged results in Table 1(a) reveal different behaviors among the optimizers. FSAM often improves generalization to novel classes but may reduce base-class accuracy. In contrast, SAMPLe maintains a stronger balance between base and

novel performance, leading to consistently higher HM across different prompt-learning backbones. These observations suggest that SAMPLe provides a more stable trade-off between specialization on the base classes and transfer to unseen classes.

Insights and Ablation Study: Table 1 and Fig 3 demonstrate that SAMPLe consistently outperforms existing sharpness-aware optimizers by maintaining both generalization and stability across varying ρ and λ. While F-SAM mitigates the excessive perturbation of traditional SAM by deviating from the mini-batch gradient, $\nabla^{\mathcal{F}}$, it still enforces a rigid batch-specific gradient, $\nabla^{\mathcal{B}}$, which remains inherently sensitive to the noisy approximation of $\nabla^{\mathcal{F}}$. This structural limitation is evident in Fig 3, where F-SAM and SAMPLe exhibit comparable robustness to variations in ρ, outperforming both SAM and SAGM. However, SAMPLe consistently achieves higher accuracy across all perturbation radii and prompt-learning methods, highlighting its optimization-aware perturbation alignment. Rather than rigidly enforcing batch-specific perturbations, SAMPLe dynamically aligns them with the current optimization state, striking an optimal balance between exploration and stability. This adaptability translates into significant performance gains, as confirmed by Figure 3.

5.3 Cross-Dataset Zero-Shot Generalization Setting

To evaluate cross-domain robustness, we train models on the ImageNet dataset containing 1000 classes and directly evaluate them on 10 unseen target datasets without any fine-tuning. This protocol measures the ability of prompt learning methods to transfer knowledge learned from a large-scale source domain to diverse downstream domains with varying visual characteristics.

Table 2 summarizes the cross-dataset performance of several prompt learning approaches with and without the proposed SAMPLe optimizer. Overall, incorporating SAMPLe consistently improves the transfer performance of prompt-based models across most datasets. For example, CoOp benefits significantly from SAMPLe, increasing its average accuracy from 63.88% to 65.85%. The improvement is particularly notable on datasets such as Cars (66.35% vs. 64.51%), Flowers (71.46% vs. 68.71%), and Aircraft (23.61% vs. 18.47%), which typically exhibit larger domain shifts from ImageNet.

A similar trend can be observed for CoCoOp. While the baseline CoCoOp achieves an average accuracy of 65.74%, integrating SAMPLe increases the average performance to 66.32%. The improvements are consistently observed across several datasets, including Cars (66.00% vs. 65.32%), Flowers (72.84% vs. 71.88%), Aircraft (23.40% vs. 22.94%), SUN397 (67.58% vs. 67.36%), and UCF101 (69.00% vs. 68.21%), indicating that SAMPLe improves the robustness of prompt learning under domain shift. For more advanced prompt learning architectures, SAMPLe continues to provide consistent gains. When applied to MaPLe, the average performance increases from 66.30% to 67.14%, with noticeable improvements on datasets such as Caltech (94.65% vs. 93.53%), Pets (91.80% vs. 90.49%), and EuroSAT (53.12% vs. 48.06%). Similarly, TCP benefits from SAMPLe with the average accuracy increasing from 66.29% to 67.53%,

Table 1: Comparison of different prompt learning methods on base and new classes. For each method, the last four rows show performance improvements when using the SAM, FSAM, SAGM, and the proposed SAMPLe optimizers.

(a) **Average**

Method	Base	New	HM
CoOp	**82.69**	63.22	71.66
+SAM	80.10	73.28	76.07
+FSAM	80.22	74.51	77.01
+SAGM	81.57	74.52	77.65
+SAMPLe	81.52	**76.76**	**78.88**
CoCoOp	80.47	71.69	75.83
+SAM	80.64	73.32	76.47
+FSAM	81.15	75.21	77.86
+SAGM	81.46	75.71	78.28
+SAMPLe	**82.11**	**77.39**	**79.51**
MAPLe	82.28	75.14	78.55
SAM	82.99	76.60	79.42
+FSAM	83.29	77.12	79.87
+SAGM	83.36	77.24	79.96
+SAMPLe	**83.51**	**77.68**	**80.28**
CoPrompt	84.00	77.23	80.48
+SAM	84.61	77.98	80.99
+FSAM	85.05	78.47	81.45
+SAGM	85.12	78.47	81.49
+SAMPLe	**85.62**	**79.03**	**82.02**
TCP	84.13	75.36	79.51
+SAM	84.17	75.58	79.64
+FSAM	84.02	76.11	79.83
+SAGM	86.97	76.50	81.40
+SAMPLe	**87.52**	**77.05**	**81.95**

(b) **ImageNet**

Method	Base	New	HM
CoOp	**76.47**	67.88	71.92
+SAM	76.05	69.85	72.88
+FSAM	69.79	70.27	70.03
+SAGM	76.40	69.91	73.01
+SAMPLe	76.36	**71.10**	**73.64**
CoCoOp	**75.98**	70.43	73.10
+SAM	75.93	71.16	73.47
+FSAM	75.59	71.62	73.55
+SAGM	75.69	71.77	73.68
+SAMPLe	75.95	**72.51**	**74.19**
MAPLe	76.66	70.54	73.47
+SAM	76.54	71.85	74.12
+FSAM	76.49	72.25	74.31
+SAGM	**76.69**	72.24	74.40
+SAMPLe	76.58	**73.20**	**74.85**
CoPrompt	**77.67**	71.27	74.33
+SAM	77.23	71.98	74.51
+FSAM	77.19	72.17	74.60
+SAGM	77.65	72.09	74.77
+SAMPLe	77.61	**72.38**	**74.90**
TCP	77.27	69.87	73.38
+SAM	77.34	69.96	73.46
+FSAM	77.11	70.13	73.47
+SAGM	77.21	70.61	73.76
+SAMPLe	**77.16**	**70.90**	**73.90**

(c) **Caltech101**

Method	Base	New	HM
CoOp	**98.00**	89.81	93.73
+SAM	97.05	**95.09**	96.06
+FSAM	96.57	93.62	95.07
+SAGM	97.16	93.78	95.44
+SAMPLe	97.49	94.98	**96.22**
CoCoOP	97.96	93.81	95.84
+SAM	97.63	94.85	96.22
+FSAM	97.54	94.90	96.20
+SAGM	97.81	95.33	96.55
+SAMPLe	**98.31**	**96.07**	**97.18**
MAPLe	97.74	94.36	96.02
+SAM	98.28	95.51	96.88
+FSAM	98.32	95.42	96.85
+SAGM	**98.37**	95.51	96.92
+SAMPLe	98.31	**95.74**	**97.01**
CoPrompt	98.27	94.90	96.55
+SAM	98.98	95.68	97.30
+FSAM	98.69	95.93	97.29
+SAGM	98.72	96.12	97.40
+SAMPLe	**99.03**	**96.66**	**97.83**
TCP	98.23	94.67	96.42
+SAM	98.31	94.82	96.55
+FSAM	98.02	94.97	96.49
+SAGM	98.62	95.83	97.20
+SAMPLe	**98.94**	**96.38**	**97.64**

(d) **OxfordPets**

Method	Base	New	HM
CoOp	93.67	95.29	94.47
+SAM	93.68	97.65	95.62
+FSAM	94.22	96.38	95.29
+SAGM	95.41	96.31	95.86
+SAMPLe	**95.92**	**98.27**	**97.08**
CoCoOp	95.20	97.69	96.43
+SAM	94.79	97.43	96.09
+FSAM	94.96	97.58	96.25
+SAGM	95.29	97.67	96.47
+SAMPLe	**95.98**	**98.59**	**97.27**
MAPLe	95.43	97.76	96.58
+SAM	96.23	98.45	97.33
+FSAM	96.41	**98.63**	97.51
+SAGM	96.45	98.52	97.47
+SAMPLe	**96.54**	98.61	**97.56**
CoPrompt	95.67	98.10	96.87
+SAM	96.52	98.99	97.74
+FSAM	96.86	99.35	98.09
+SAGM	97.00	99.25	98.11
+SAMPLe	**97.65**	**99.93**	**98.78**
TCP	94.67	97.20	95.92
+SAM	94.19	97.53	95.83
+FSAM	94.50	98.03	96.23
+SAGM	95.33	98.18	96.73
+SAMPLe	**96.58**	**98.96**	**97.76**

(e) **StanfordCars**

Method	Base	New	HM
CoOp	78.12	60.40	68.13
+SAM	72.84	73.11	72.97
+FSAM	**76.13**	73.92	**74.99**
+SAGM	74.96	73.86	74.41
+SAMPLe	72.21	**75.10**	73.63
CoCoOp	70.49	73.59	72.01
+SAM	71.00	75.22	73.05
+FSAM	**73.92**	75.80	74.85
+SAGM	73.41	76.13	74.75
+SAMPLe	73.27	**77.89**	**75.51**
MAPLe	72.94	74.00	73.47
+SAM	74.21	75.83	75.01
+FSAM	75.75	76.41	76.08
+SAGM	75.40	76.68	76.03
+SAMPLe	**76.32**	**77.44**	**76.88**
CoPrompt	76.97	74.40	75.66
+SAM	77.64	75.00	76.30
+FSAM	78.38	75.49	76.91
+SAGM	78.39	75.56	76.95
+SAMPLe	**79.14**	**76.03**	**77.55**
TCP	80.80	74.13	77.32
+SAM	81.03	74.13	77.43
+FSAM	80.76	74.51	77.50
+SAGM	82.23	75.25	78.59
+SAMPLe	**83.03**	**75.70**	**79.19**

(f) **Flowers102**

Method	Base	New	HM
CoOp	97.60	59.67	74.06
+SAM	94.97	72.65	82.32
+FSAM	94.91	73.85	83.07
+SAGM	95.25	74.53	83.63
+SAMPLe	**96.73**	**77.52**	**86.07**
CoCoOp	94.87	71.75	81.71
+SAM	95.02	74.40	83.46
+FSAM	95.17	76.43	84.78
+SAGM	95.50	77.01	85.26
+SAMPLe	**96.47**	**79.48**	**87.15**
MAPLe	95.92	72.46	82.56
+SAM	96.53	75.03	84.43
+FSAM	96.69	75.91	85.05
+SAGM	96.77	76.07	85.18
+SAMPLe	**96.82**	**76.98**	**85.77**
CoPrompt	97.27	76.60	85.71
+SAM	97.79	77.55	86.50
+FSAM	98.61	78.10	87.16
+SAGM	98.36	78.02	87.02
+SAMPLe	**99.08**	**78.47**	**87.58**
TCP	97.73	75.57	85.23
+SAM	97.99	75.37	85.21
+FSAM	97.44	76.30	85.62
+SAGM	98.76	76.91	86.48
+SAMPLe	**99.49**	**77.36**	**87.02**

(g) **Food101**

Method	Base	New	HM
CoOp	88.33	82.26	85.19
+SAM	88.19	90.12	89.14
+FSAM	89.74	90.40	90.07
+SAGM	90.48	91.54	91.00
+SAMPLe	**90.75**	**92.01**	**91.38**
CoCoOp	90.70	91.29	90.99
+SAM	90.83	91.70	91.26
+FSAM	91.21	92.11	91.66
+SAGM	91.36	92.35	91.85
+SAMPLe	**91.84**	**93.08**	**92.46**
MAPLe	90.71	92.05	91.38
+SAM	91.55	92.81	92.18
+FSAM	91.88	93.20	92.54
+SAGM	**91.95**	**93.35**	**92.64**
+SAMPLe	91.94	93.31	92.62
CoPrompt	90.73	92.07	91.40
+SAM	91.61	92.81	92.21
+FSAM	92.10	93.47	92.78
+SAGM	92.21	93.42	92.81
+SAMPLe	**92.70**	**94.21**	**93.45**
TCP	90.57	91.37	90.97
+SAM	90.72	91.78	91.25
+FSAM	90.55	91.92	91.23
+SAGM	92.01	92.66	92.33
+SAMPLe	**92.49**	**93.44**	**92.96**

(h) **FGVCAircrafts**

Method	Base	New	HM
CoOp	**40.44**	22.30	28.75
+SAM	34.33	35.87	35.08
+FSAM	35.92	**39.12**	37.45
+SAGM	36.73	35.75	36.23
+SAMPLe	36.43	38.81	**37.58**
CoCoOp	33.41	23.71	27.74
+SAM	34.62	29.53	31.87
+FSAM	35.45	33.67	34.54
+SAGM	35.79	34.18	34.97
+SAMPLe	**37.14**	**36.59**	**36.86**
MAPLe	37.44	35.61	36.50
+SAM	38.55	37.99	38.27
+FSAM	39.08	38.79	38.93
+SAGM	39.11	38.37	38.74
+SAMPLe	**39.47**	**39.01**	**39.24**
CoPrompt	40.20	39.33	39.76
+SAM	41.08	39.88	40.47
+FSAM	41.28	40.68	40.98
+SAGM	41.45	40.68	41.06
+SAMPLe	**42.38**	**40.95**	**41.65**
TCP	41.97	34.43	37.83
+SAM	42.02	34.49	37.89
+FSAM	42.05	35.02	38.21
+SAGM	43.23	35.55	39.02
+SAMPLe	**44.19**	**35.79**	**39.60**

(i) **SUN397**

Method	Base	New	HM
CoOp	**80.60**	65.89	72.51
+SAM	77.84	78.82	78.33
+FSAM	78.42	78.98	78.70
+SAGM	78.50	79.03	78.76
+SAMPLe	78.40	**79.22**	**78.81**
CoCoOp	79.74	76.86	78.27
+SAM	79.48	77.44	78.45
+FSAM	79.73	78.35	79.03
+SAGM	80.03	78.83	79.43
+SAMPLe	**80.58**	**79.65**	**80.11**
MAPLe	80.82	78.70	79.75
+SAM	81.42	79.86	80.63
+FSAM	81.68	80.06	80.86
+SAGM	81.74	80.26	80.99
MAPLe+S	**81.83**	**80.48**	**81.15**
CoPrompt	82.63	80.03	81.31
+SAM	83.27	80.73	81.98
+FSAM	84.09	81.19	82.61
+SAGM	83.96	81.34	82.63
+SAMPLe	**84.17**	**82.08**	**83.11**
TCP	82.63	78.20	80.35
+SAM	82.83	78.66	80.70
+FSAM	82.57	78.99	80.74
+SAGM	83.92	79.43	81.61
+SAMPLe	**84.11**	**80.16**	**82.08**

(j) **DTD**

Method	Base	New	HM
CoOp	**79.44**	41.18	54.24
+SAM	76.97	56.52	65.18
+FSAM	73.93	54.78	62.93
+SAGM	77.31	56.76	65.46
+SAMPLe	78.12	**63.04**	**69.77**
CoCoOp	77.01	56.00	64.85
+SAM	77.05	57.13	65.61
+FSAM	76.60	57.92	65.96
+SAGM	77.47	59.15	67.08
+SAMPLe	**78.14**	**63.05**	**69.79**
MAPLe	80.36	59.18	68.16
+SAM	80.99	61.43	69.87
+FSAM	80.88	62.55	70.54
+SAGM	81.15	63.00	70.93
+SAMPLe	**81.23**	**63.65**	**71.37**
CoPrompt	83.13	64.73	72.79
+SAM	83.67	65.51	73.48
+FSAM	84.49	65.74	73.94
+SAGM	84.31	66.21	74.17
+SAMPLe	**84.72**	**66.57**	**74.56**
TCP	82.77	58.07	68.25
+SAM	82.73	58.39	68.47
+FSAM	82.84	59.41	69.17
+SAGM	83.89	59.34	69.51
+SAMPLe	**84.30**	**59.67**	**69.88**

(k) **EuroSAT**

Method	Base	New	HM
CoOp	**92.19**	54.74	68.69
+SAM	87.00	61.23	71.87
+FSAM	89.60	73.41	80.70
+SAGM	91.07	72.31	80.61
+SAMPLe	90.88	**75.43**	**82.44**
CoCoOp	87.49	60.04	71.21
+SAM	88.02	62.00	72.75
+FSAM	89.19	72.15	79.77
+SAGM	90.05	73.00	80.63
+SAMPLe	**90.76**	**75.32**	**82.32**
MAPLe	94.07	73.23	82.35
+SAM	94.55	74.25	83.18
+FSAM	94.89	75.12	83.86
+SAGM	94.92	75.38	84.03
+SAMPLe	**94.96**	**75.66**	**84.22**
CoPrompt	94.60	78.57	85.84
+SAM	95.27	79.56	86.71
+FSAM	95.89	80.05	87.26
+SAGM	96.03	79.85	87.20
+SAMPLe	**96.75**	**80.45**	**87.85**
TCP	91.63	74.73	82.32
+SAM	91.54	75.13	82.55
+FSAM	91.92	75.87	83.24
+SAGM	92.96	75.90	83.57
+SAMPLe	**93.66**	**76.46**	**84.19**

(l) **UCF101**

Method	Base	New	HM
CoOp	**84.69**	56.05	67.46
+SAM	82.16	75.13	78.49
+FSAM	83.14	74.88	78.79
+SAGM	83.97	75.96	79.76
+SAMPLe	83.40	**78.85**	**81.06**
CoCoOp	82.33	73.45	77.64
+SAM	82.64	75.62	78.97
+FSAM	83.26	76.77	79.88
+SAGM	83.61	77.44	80.41
+SAMPLe	**84.73**	**79.01**	**81.77**
MAPLe	83.00	78.66	80.77
+SAM	84.05	79.59	81.76
+FSAM	84.14	80.02	82.03
+SAGM	84.36	80.24	82.25
+SAMPLe	**84.62**	**80.39**	**82.45**
CoPrompt	86.90	79.57	83.07
+SAM	87.65	80.14	83.73
+FSAM	88.00	81.04	84.38
+SAGM	88.27	80.67	84.30
+SAMPLe	**88.55**	**81.59**	**84.93**
TCP	87.13	80.77	83.83
+SAM	87.22	81.17	84.08
+FSAM	87.50	82.04	84.69
+SAGM	88.45	81.83	85.01
+SAMPLe	**88.73**	**82.76**	**85.64**

Table 2: Performance improvement by SAMPLe optimizer in zero-shot cross-dataset.

	Source	Target										
	ImNet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN	DTD	EuSAT	UCF	Avg
CoOp	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
+SAMPLe	70.60	94.02	89.90	66.35	71.46	86.32	23.61	67.61	44.86	46.74	67.64	65.85
CoCoOp	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
+SAMPLe	71.03	94.52	90.30	66.00	72.84	86.40	23.40	67.58	46.75	46.41	69.00	66.32
MaPLe	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
+SAMPLe	70.69	94.65	91.80	66.75	72.59	87.17	23.32	67.83	44.55	53.12	69.58	67.14
CoPrompt	70.80	94.50	90.73	65.67	72.30	86.43	24.00	67.57	47.07	51.90	69.73	67.00
+SAMPLe	70.86	96.24	93.19	67.65	73.59	88.60	21.86	67.75	42.74	47.13	69.71	67.95
TCP	71.40	93.97	91.25	64.69	71.21	86.69	23.45	67.15	44.35	51.45	68.73	66.29
+SAMPLe	71.17	96.82	92.61	67.43	72.17	88.02	23.20	68.20	47.79	47.46	70.61	67.53

while achieving higher scores on datasets including Caltech (96.82% vs. 93.97%), Cars (67.43% vs. 64.69%), SUN397 (68.20% vs. 67.15%), and DTD (47.79% vs. 44.35%).

Even for strong baselines such as CoPrompt, which already achieves a relatively high average accuracy of 67.00%, SAMPLe further improves the overall performance to 67.95%. Improvements can be observed across multiple datasets including Caltech (96.24% vs. 94.50%), Pets (93.19% vs. 90.73%), and Flowers (73.59% vs. 72.30%), suggesting that the proposed optimizer provides benefits that are complementary to architectural advances in prompt learning. Across all evaluated methods, SAMPLe consistently improves the average performance while maintaining stable accuracy on the source domain (ImageNet). This behavior indicates that the proposed optimization strategy improves the generalization capability of learned prompts without sacrificing source-domain alignment. We attribute these gains to SAMPLe's ability to explore flatter regions of the loss landscape through its gradient component neutralization mechanism, which helps avoid overfitting to the source distribution and enables more robust transfer to unseen datasets.

5.4 Cross-Domain Zero-Shot generalization setting

Table 3 presents the results for domain generalization. The original ImageNet dataset trains learnable prompts to contextualize the model input in this setup. The evaluation is performed on four diverse ImageNet variants (*-V2*, *-Sketch*, *-Adversarial*, and *-Rendition*), each representing different types of distribution shifts. This evaluation tests how well the model generalizes to unseen domains. Integrating the proposed SAMPLe optimizer with CoOp and CoCoOp leads to consistent improvements in generalization performance. For example, CoOp+SAMPLe raises the average accuracy from 59.28% (CoOp) to 60.41%, with notable gains on datasets such as ImageNet-S (1.32%), ImageNet-A (1.26%), and ImageNet-R (1.34%). Similarly, CoCoOp+SAMPLe increases the average accuracy of CoCoOp from 59.91% to 60.47%, showing significant improvements on ImageNet-A (0.48%) and ImageNet-R (0.56%). Compared to other methods

like ProGrad, KgCoOp, and MAPLe, CoCoOp+SAMPLe achieves the highest average accuracy of 60.47%, surpassing MAPLe (60.27%). This demonstrates SAMPLe's ability to handle domain shifts more effectively.

The results show that SAMPLe enhances the model's exploration during optimization, allowing it to adapt better to challenging datasets such as ImageNet-A and ImageNet-R.

Table 3: Performance improvement by SAMPLe optimizer in zero-shot domain generalization.

	Source	**Target**				
	ImaNet	ImgNet-V2	ImgNet-S	ImgNet-A	ImgNet-R	Avg
CoOp	71.51	64.20	47.99	49.71	75.21	59.28
+SAMPLe	70.60	64.43	49.31	50.97	77.34	60.41
CoCoOp	71.02	64.07	48.75	50.63	76.18	59.90
+SAMPLe	71,03	64.31	49.00	51.05	77.52	60.47
MaPLe	70.72	64.07	49.15	50.90	76.98	60.28
+SAMPLe	70.69	65.78	50.51	52.11	77.84	61.56
CoPrompt	70.80	64.25	49.43	50.50	77.51	60.42
+SAMPLe	70.86	66.12	50.28	51.61	78.50	61.63
TCP	71.20	64.60	49.50	51.20	76.73	60.51
+SAMPLe	71.17	66.00	50.42	52.19	78.98	61.90

6 Conclusion

This paper introduces SAMPLe, a novel, model-agnostic optimizer that enhances prompt learning across all modalities by integrating sharpness-aware objectives into the optimization process. Inspired by recent advancements in sharpness-aware minimization (SAM), SAMPLe balances exploitation and exploration by considering the optimization state, achieving a harmonious trade-off between accuracy and generalization. A key innovation of SAMPLe lies in its ability to dynamically adapt gradients for prompt learning tasks, ensuring flatter minima that promote robust generalization while maintaining strong performance on seen distributions. Through extensive analysis and rigorous experiments, we demonstrated that SAMPLe significantly improves the generalization capabilities of prompt-based vision-language models (VLMs) and broadly applies to modality-specific prompt-learning tasks. It consistently outperforms counterpart methods across various benchmark datasets, underscoring its effectiveness and robustness in addressing the challenges of prompt learning. Further details are provided in the supplementary material.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Number CNS-2232048, CNS-2204445, and CCF-2553684.

References

1. Bahri, D., Mobahi, H., Tay, Y.: Sharpness-aware minimization improves language model generalization. arXiv preprint arXiv:2110.08529 (2021)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13. pp. 446–461. Springer (2014)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
4. Chen, X., Hsieh, C.J., Gong, B.: When vision transformers outperform resnets without pre-training or strong data augmentations. ICLR (2022)
5. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3606–3613 (2014)
6. 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/CVF Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)
7. Ding, C., Gao, X., Dong, S., He, Y., Wang, Q., Kot, A., Gong, Y.: Lobg: less overfitting for better generalization in vision-language model. arXiv preprint arXiv:2410.10247 (2024)
8. Dziugaite, G.K., Roy, D.M.: Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008 (2017)
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop. pp. 178–178 (2004). https://doi.org/10.1109/CVPR.2004.383
10. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. arXiv preprint arXiv:2010.01412 (2020)
11. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. In: International Conference on Learning Representations (2021)
12. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
13. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8340–8349 (2021)
14. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15262–15271 (2021)
15. Iacob, A., Sani, L., Kurmanji, M., Shen, W.F., Qiu, X., Cai, D., Gao, Y., Lane, N.D.: DEPT: Decoupled embeddings for pre-training language models. In: ICLR (2025), https://openreview.net/forum?id=vf5aUZT0Fz
16. Ishida, T., Yamane, I., Sakai, T., Niu, G., Sugiyama, M.: Do we need zero training loss after achieving zero training error? ICML'20, JMLR.org (2020)

17. Jetly, V., Ibayashi, H.: Splash in a flash: Sharpness-aware minimization for efficient liquid splash simulation. In: Annual Conference of the European Association for Computer Graphics, Eurographics (2022)
18. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
19. 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. arXiv preprint arXiv:1609.04836 (2016)
20. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19113–19122 (2023)
21. Khattak, M.U., Wasim, S.T., Naseer, M., Khan, S., Yang, M.H., Khan, F.S.: Self-regulating prompts: Foundational model adaptation without forgetting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15190–15200 (2023)
22. Kim, M., Li, D., Hu, S.X., Hospedales, T.: Fisher sam: Information geometry and sharpness aware minimisation. In: International Conference on Machine Learning. pp. 11148–11161. PMLR (2022)
23. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 554–561 (2013)
24. Kwon, J., Kim, J., Park, H., Choi, I.K.: Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In: International Conference on Machine Learning. pp. 5905–5914. PMLR (2021)
25. Li, T., Zhou, P., He, Z., Cheng, X., Huang, X.: Friendly sharpness-aware minimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5631–5640 (2024)
26. Liu, L., Wang, N., Zhou, D., Liu, D., Yang, X., Gao, X., Liu, T.: Generalizable prompt learning via gradient constrained sharpness-aware minimization. IEEE Transactions on Multimedia pp. 1–14 (2024). https://doi.org/10.1109/TMM.2024.3521702
27. Lotfi, F., Rajoli, H., Afghah, F.: Task-specific sharpness-aware o-ran resource management using multi-agent reinforcement learning. IEEE Transactions on Machine Learning in Communications and Networking **4**, 98–114 (2025)
28. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt distribution learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5206–5215 (2022)
29. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
30. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008)
31. Novak, R., Bahri, Y., Abolafia, D.A., Pennington, J., Sohl-Dickstein, J.: Sensitivity and generalization in neural networks: an empirical study. arXiv preprint arXiv:1802.08760 (2018)
32. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3498–3505. IEEE (2012)

33. 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. PMLR (2021)
34. Rajoli Nowdeh, H., Ji, J., Ma, X., Afghah, F.: Modality-aware sam: Sharpness-aware-minimization driven gradient modulation for harmonized multimodal learning. Advances in Neural Information Processing Systems **38**, 168772–168794 (2026)
35. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: International Conference on Machine Learning. pp. 5389–5400. PMLR (2019)
36. Soomro, K.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
37. Wang, H., Ge, S., Lipton, Z., Xing, E.P.: Learning robust global representations by penalizing local predictive power. Advances in Neural Information Processing Systems **32** (2019)
38. Wang, P., Zhang, Z., Lei, Z., Zhang, L.: Sharpness-aware gradient matching for domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3769–3778 (2023)
39. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3485–3492. IEEE (2010)
40. Yao, H., Zhang, R., Xu, C.: Visual-language prompt tuning with knowledge-guided context optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6757–6767 (2023)
41. Yao, H., Zhang, R., Xu, C.: Tcp: Textual-based class-aware prompt tuning for visual-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23438–23448 (2024)
42. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: Filip: Fine-grained interactive language-image pre-training. arXiv preprint arXiv:2111.07783 (2021)
43. Yeo, B.T., Sabuncu, M.R., Desikan, R., Fischl, B., Golland, P.: Effects of registration regularization and atlas sharpness on segmentation accuracy. Medical image analysis **12**(5), 603–615 (2008)
44. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18123–18133 (2022)
45. Zhang, J., Huang, J., Jin, S., Lu, S.: Vision-language models for vision tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
46. 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)
47. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16795–16804 (2022). https://doi.org/10.1109/CVPR52688.2022.01631
48. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022)
49. Zhou, Z., Dong, S., Ding, C., Gao, X., He, Y., Gong, Y.: Diversity covariance-aware prompt learning for vision-language models. Pattern Recognition p. 112806 (2025)

50. Zhu, B., Niu, Y., Han, Y., Wu, Y., Zhang, H.: Prompt-aligned gradient for prompt tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15659–15669 (2023)
51. Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N., Tatikonda, S., Duncan, J., Liu, T.: Surrogate gap minimization improves sharpness-aware training. ICLR (2021)

8 Appendix

In this supplementary material, we begin with the proof of Theorem 1, followed by a comprehensive comparison of various optimizers, including SAM, FSAM, and SAGM, against the proposed SAMPLe across all 11 datasets using CoOp, CoCoOp, MaPLe, CoPrompt, and TCP. We then present the ablation study and elaborate on deployment details in the subsequent sections.

8.1 Proof of Theorem 1

Lemma 1. *Let $\nabla L(\theta_t; \mathcal{D})$ be the mini-batch gradient at iteration t and assume that $\nabla L(\theta_t; \mathcal{D})$ is bounded by L_{max}, i.e., $\nabla L(\theta_t; \mathcal{D}) \leq L_{max}$. Suppose the exponentially moving average (EMA) of the gradients is given by:*

$$m_t = \lambda m_{t-1} + (1 - \lambda)\nabla L(\theta_t; \mathcal{D}), \tag{20}$$

where $\lambda \in (0, 1)$. Then the EMA approximation m_T at iteration T is bounded by:

$$m_T \leq \nabla L_{max}(1 - \lambda^T). \tag{21}$$

Proof: We proceed by induction and using properties of the geometric series.

At $t = 1$, we have:

$$m_1 = (1 - \lambda)\nabla L(\theta_1; \mathcal{D}), \tag{22}$$

so,

$$m_1 \leq (1 - \lambda)\nabla L_{max}. \tag{23}$$

For $t \geq 2$, the EMA update is given by:

$$m_t = \lambda m_{t-1} + (1 - \lambda)\nabla L(\theta_{t-1}; \mathcal{D}), \tag{24}$$

and since $\nabla L(\theta_t; \mathcal{D}) \leq \nabla L_{max}$, we can bound:

$$m_t \leq (1 - \lambda) \sum_{i=1}^{t} \lambda^{t-i} \nabla L_{max}. \tag{25}$$

The sum $\sum_{i=1}^{t} \lambda^{t-i}$ is a geometric series and simplifies to:

$$\sum_{i=1}^{t} \lambda^{t-i} = \frac{1 - \lambda^t}{1 - \lambda}. \tag{26}$$

Thus, the upper bound becomes:

$$m_T \leq (1 - \lambda)\nabla L_{max} \frac{1 - \lambda^T}{1 - \lambda}. \tag{27}$$

Simplifying:

$$m_T \leq \nabla L_{max}(1 - \lambda^T). \tag{28}$$

Hence, as $T \to \infty$, m_T asymptotically approaches L_{max}.

Theorem 1 (definition). *Assuming the loss function defined as Eq. 10, we have:*

$$\begin{aligned}\mathcal{L}(\theta_t;\mathcal{D}) =& L(\theta_t;\mathcal{D}) + L\big(\theta_t + \epsilon_t^{\star} - \alpha_t \nabla L(\theta_t;\mathcal{D}) + \\ & \alpha_t\ \xi_t\ \sigma_t\ \nabla^{\mathcal{F}} L(\theta;\mathcal{D})\big)\end{aligned} \tag{29}$$

and $\mathcal{L}(\theta_t;\mathcal{D})$ satisfies the following assumptions:

(i) its gradient $\nabla\mathcal{L}(\theta_t;\mathcal{D})$ is bounded, i.e., $\|\nabla\mathcal{L}(\theta_t;\mathcal{D})\| \leq \nabla\mathcal{L}_{max}$, $\forall t$,

(ii) The stochastic gradient is K-Lipschitz gradient, i.e., $\|\nabla\mathcal{L}(\theta_t;\mathcal{D}) - \nabla\mathcal{L}(\theta_t^{'};\mathcal{D})\| \leq K\|\theta_t - \theta_t^{'}\|$, $\forall\theta_t, \theta_t^{'}$.

Let the learning rate η_t be $\eta_t = \frac{\eta_0}{\sqrt{t}}$, and let the perturbations decrease with the same rate as the learning rate, i.e., $\rho_t = \frac{\rho_0}{\sqrt{t}}$ and $\alpha_t = \frac{\alpha_0}{\sqrt{t}}$. If $\hat{\theta}_t$ be defined as follows:

$$\begin{aligned}\hat{\theta}_t =&\ \theta_t + \epsilon^* - \alpha_t \sum_{(x,y)\in\mathcal{D}} \nabla L(\theta_t,\mathcal{D}) + \\ & \alpha_t \sum_{(x,y)\in\mathcal{D}} \xi_t\ \sigma_t\ \nabla^{\mathcal{F}} L(\theta_t;\mathcal{D}) \\ \approx&\ \theta_t + \epsilon^* - \alpha_t \sum_{(x,y)\in\mathcal{D}} \nabla L(\theta_t,\mathcal{D})\ + \alpha_t \sum_{(x,y)\in\mathcal{D}} \xi_t\ \sigma_t\ m_t\end{aligned} \tag{30}$$

where m_t approximates full-gradient as it is defined in Eq. 9.

$$\frac{1}{T}\sum_{t=1}^{T}\left[\|\nabla\mathcal{L}(\theta_t;\mathcal{D})\|^2\right] \leq \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right), \tag{31}$$

Proof: For simplicity of equations, we define d_t as:

$$d_t = \theta_{t+1} - \theta_t = -\eta_t\nabla\mathcal{L}(\theta_t) - \eta_t\nabla\mathcal{L}(\hat{\theta}_t) \tag{32}$$

where η_t represents the learning rate. By K-Lipschitz gradient of $\mathcal{L}$, the definition of d_t in Eq. 32, and the inequality of $\|\nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t)\|^2 \leq 2(\|\nabla\mathcal{L}(\theta_t)\|^2 + \|\nabla\mathcal{L}(\hat{\theta}_t)\|^2)$ we need to define an upper bound for loss update difference between every consequent iteration as follows:

$$
\begin{aligned}
&\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\theta_t) \leq \langle \nabla\mathcal{L}(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{K}{2}\|\theta_{t+1} - \theta_t\|^2 = \\
&\langle \nabla\mathcal{L}(\theta_t), d_t \rangle + \frac{K}{2}\|d_t\|^2 = -\eta_t \langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t)) \rangle \\
&\quad + \frac{K\eta_t^2}{2}\|\nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t)\|^2 = \\
&- \eta_t \langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t) \rangle \\
&\quad + \frac{K\eta_t^2}{2}\|\nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t)\|^2 = \\
&- \eta_t \|\nabla\mathcal{L}(\theta_t)\|^2 - \eta_t \langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t) \rangle \\
&\quad + \frac{K\eta_t^2}{2}\|\nabla\mathcal{L}(\theta_t) + \nabla\mathcal{L}(\hat{\theta}_t)\|^2 = \\
&- 2\eta_t \|\nabla\mathcal{L}(\theta_t)\|^2 - \eta_t \langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\hat{\theta}_t) - \nabla\mathcal{L}(\theta_t) \rangle \\
&\quad + K\eta_t^2 (\|\nabla\mathcal{L}(\theta_t)\|^2 + \|\nabla\mathcal{L}(\hat{\theta}_t)\|^2) \\
&\leq -2\eta_t \|\nabla\mathcal{L}(\theta_t)\|^2 + \eta_t \langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\hat{\theta}_t) \rangle \\
&\quad + K\eta_t^2 \nabla\mathcal{L}_{max}^2
\end{aligned}
\tag{33}
$$

in this step, we should show that there is an upper bound for the expression $\langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\hat{\theta}_t) - \nabla\mathcal{L}(\theta_t) \rangle$, it is done as follows;

$$
\begin{aligned}
&\langle \nabla\mathcal{L}(\theta_t), \nabla\mathcal{L}(\hat{\theta}_t) - \nabla\mathcal{L}(\theta_t) \rangle = \\
&\leq \|\nabla\mathcal{L}(\theta_t)\| \; \|\nabla\mathcal{L}(\hat{\theta}_t) - \nabla\mathcal{L}(\theta_t)\| \\
&\leq K \; \|\nabla\mathcal{L}(\theta_t)\| \; \|\hat{\theta}_t - \theta_t\| \quad \textit{(K-Lipschitz condition.)} \\
&= K \; \|\nabla\mathcal{L}(\theta_t)\| \cdot \|\epsilon_t^* - \alpha_t \sum_{(x,y)\in\mathcal{D}} \nabla L(\theta_t, \mathcal{D}) + \\
&\qquad \alpha_t \sum_{(x,y)\in\mathcal{D}} \xi_t \; \sigma_t \; m_t\| \quad \textit{(using Eq. 30)} \\
&\leq K \; \|\nabla\mathcal{L}(\theta_t)\| \cdot \|\epsilon_t^*\| + \\
&\quad K\alpha_t \; \|\nabla\mathcal{L}(\theta_t)\| \cdot \| \sum_{(x,y)\in\mathcal{D}} (\nabla L(\theta_t, \mathcal{D}) - \xi_t \; \sigma_t \; m_t)\| \\
&\textit{(Lemma. 1 and the fact that } \nabla L_{max} \leq \nabla\mathcal{L}_{max}\textit{, given} \\
&\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \textit{ together would result in:)} \\
&\leq K \; \rho_t \; \nabla L_{max} \; + K \; N \; \alpha_t \; \nabla L_{max} \; \nabla L_{max} \; - \\
&K\alpha_t \; \nabla\mathcal{L}_{max} \; \xi_t \leq K(\rho_t + \alpha_t)\nabla\mathcal{L}_{max} + \; K \; N \; \alpha_t \; \nabla\mathcal{L}_{max}^2
\end{aligned}
\tag{34}
$$

by replacing the upper bound in Eq. 34 in Eq. 33 we have:

$$
\begin{aligned}
\mathcal{L}(\theta_{t+1}) - \mathcal{L}(\theta_t) \leq &-2\eta_t \|\nabla\mathcal{L}(\theta_t)\|^2 + \\
&\eta_t K(\rho_t + \alpha_t)\nabla\mathcal{L}_{max} + \; K \; N \; \eta_t \; \alpha_t \; \nabla\mathcal{L}_{max}^2 + \\
&K\eta_t^2 \nabla\mathcal{L}_{max}^2
\end{aligned}
\tag{35}
$$

by rearranging the inequality, it will be as:

$$2\sum_{t=1}^{T}\eta_t\|\nabla\mathcal{L}(\theta_t)\|^2 \leq \mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1}) + K\eta_t^2\nabla\mathcal{L}_{max}^2 + \eta_t K(\rho_t+\alpha_t)\nabla\mathcal{L}_{max} + \ K\ N\ \eta_t\ \alpha_t\ \nabla\mathcal{L}_{max}^2 \tag{36}$$

considering definition of $\eta_t = \frac{\eta_0}{\sqrt{t}}$ and taking summation over all iterations on the left side of Eq. 36, we can define a lower bound as follows:

$$2\frac{\eta_0}{\sqrt{T}}\sum_{t=1}^{T}\|\nabla\mathcal{L}(\theta_t)\|^2 \leq 2\sum_{t=1}^{T}\eta_t\|\nabla\mathcal{L}(\theta_t)\|^2 \tag{37}$$

using telescope sum properties and considering $0 \leq \mathcal{L}_t \leq \mathcal{L}_{max}\ \forall t$:

$$\sum_{t=1}^{T}\big(\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})\big) = \mathcal{L}_1 - \mathcal{L}_T \leq \mathcal{L}_1 \tag{38}$$

by taking summation over all iterations on the right side of Eq. 36 and using Eq. 38 we will have:

$$\begin{aligned}\sum_{t=1}^{T}\big(\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})\big) + K\sum_{t=1}^{T}\big(\eta_t(\rho_t+\alpha_t)\big)\mathcal{L}_{max} + \\ K\ N\ \sum_{t=1}^{T}(\eta_t\ \alpha_t)\ \mathcal{L}_{max}^2 + K\sum_{t=1}^{T}(\eta_t^2)\mathcal{L}_{max}^2\ \leq \mathcal{L}_1 + \\ K\sum_{t=1}^{T}\big(\eta_t(\rho_t+\alpha_t)\big)\nabla\mathcal{L}_{max} + KN\sum_{t=1}^{T}(\eta_t\ \alpha_t)\ \nabla\mathcal{L}_{max}^2 + \\ K\sum_{t=1}^{T}(\eta_t^2)\nabla\mathcal{L}_{max}^2\end{aligned} \tag{39}$$

using these lower and upper bounds, we can write the following inequality:

$$\begin{aligned}2\frac{\eta_0}{\sqrt{T}}\sum_{t=1}^{T}\|\nabla\mathcal{L}(\theta_t)\|^2 \leq \mathcal{L}_1 + K\sum_{t=1}^{T}\big(\eta_t(\rho_t+\alpha_t)\big)\nabla\mathcal{L}_{max} + \\ KN\sum_{t=1}^{T}(\eta_t\ \alpha_t)\ \nabla\mathcal{L}_{max}^2 + K\sum_{t=1}^{T}(\eta_t^2)\nabla\mathcal{L}_{max}^2\ = \\ \mathcal{L}_1 + K\eta_0(\rho_0+\alpha_0)\nabla\mathcal{L}_{max}\sum_{t=1}^{T}(\frac{1}{t}) + \\ KN\eta_0\ \alpha_0\sum_{t=1}^{T}(\frac{1}{t})\ \nabla\mathcal{L}_{max}^2 + K\eta_0^2\sum_{t=1}^{T}(\frac{1}{t})\ \nabla\mathcal{L}_{max}^2\end{aligned} \tag{40}$$

considering $\sum_{t=1}^{T}(\frac{1}{t}) \leq 1 + \log(T)$:

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}(\theta_t)\|^2 \leq \frac{\mathcal{L}_1}{2\eta_0} + \frac{K\mathcal{L}_{max}\big(\rho_0 + \eta_0\mathcal{L}_{max} + (1+N\mathcal{L}_{max})\alpha_0\big)}{2}\left(\frac{1+\log(T)}{\sqrt{T}}\right) \tag{41}$$

which means;

$$\frac{1}{T}\sum_{t=1}^{T}\left[\|\nabla\mathcal{L}(\theta_t;\mathcal{D})\|^2\right] \leq \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right), \tag{42}$$

Hence, we conclude that the proposed loss function $\mathcal{L}(\theta_t;\mathcal{D})$ converges with a rate comparable to first-order gradient-based optimization methods such as Adam and RMSProp, provided that the model is trained for sufficiently large iterations T.

Table 4: Details of Datasets.

Dataset	Number of Class	Train Samples	Test Samples
ImageNet	1000	1.28 M	50000
Caltech101	100	4128	2465
OxfordPets	37	2944	3669
StanfordCars	196	6509	8041
Flower102	102	4093	2463
Food101	101	50500	30300
FGVCAircraft	100	3334	3333
SUN397	397	15880	19850
DTD	47	2820	1692
EuroSSAT	10	13500	8100
UCF101	101	7639	3783
ImageNet-V2	1000	N/A	10000
ImageNet-Sketch	1000	N/A	50889
ImageNet-A	200	N/A	7500
ImageNet-R	200	N/A	30000

8.2 Optimizer Impact on CoOp, CoCoOp, MaPLe, and CoPrompt

Table 1 provides a comprehensive comparison of different prompt learning methods enhanced with various optimizers, including SAM, FSAM, SAGM, and the proposed SAMPLe, evaluated across all 11 datasets.

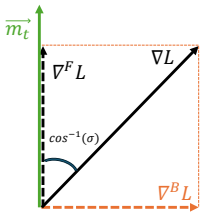


Fig. 4: Gradient decomposition .

8.3 Datasets Details

We describe the 11 datasets and 4 variations of ImageNet used for evaluation, providing details on the number of classes, along with the training and testing sample sizes, as summarized in Table. 4.

8.4 Robustness to Perturbation Radius

Unlike SAM and SAGM, which maximize the worst-case perturbation per mini-batch, SAMPLe and F-SAM mitigate instability by constraining perturbations to align with the mini-batch gradient rather than strictly enforcing adversarial maximization. While F-SAM explicitly computes a batch-specific gradient perturbation, SAMPLe encourages alignment with the mini-batch gradient direction without rigidly restricting updates. This distinction allows SAMPLe to balance batch-specific adaptation and generalization, enhancing stability across perturbation scales while achieving superior accuracy without excessive sensitivity to ρ in Fig. 3.

8.5 Robustness to Whole-Batch Gradient Approximation

Figure 3 demonstrates that SAMPLe is significantly more robust to variations in λ, which controls the effect of previous gradients in approximating the full-batch gradient. Unlike F-SAM, which strictly enforces the batch-specific gradient in its perturbation update, SAMPLe only encourages alignment, allowing for greater adaptability. This flexibility enables SAMPLe to dynamically compensate for changes in λ based on the optimization state, leading to consistently higher accuracy and lower variance.

8.6 Deployment Details

Hyperparameter selection Table 5 presents the tuned hyperparameters for SAM, SAGM, and SAMPLe applied to CoOp, CoCoOp, MaPLe, Co-Prompt, and TCP across 11 datasets, focusing on the perturbation radius, ρ, alignment parameter, α, and previous gradient effective factor, λ. The results in Table 1 are well aligned with the hyperparameter choices in Table 5. In particular, SAMPLe

Table 5: Hyperparameter settings for SAM, SAGM, and SAMPLe applied to CoOp, CoCoOp, MaPLe, Co-Prompt, and TCP during training on all 11 datasets (in Table. 1).

	SAM	**SAGM**		**F-SAM**		**SAMPLe**		
	ρ	ρ	α	ρ	λ	ρ	α	λ
CoOp	0.05	0.05	0.0010	0.05	0.15	0.05	0.0015	0.15
CoCoOp	0.10	0.10	0.0010	0.05	0.15	0.10	0.0015	0.15
MaPLe	0.05	0.10	0.0010	0.05	0.15	0.10	0.0015	0.15
CoPrompt	0.05	0.05	0.0010	0.05	0.15	0.05	0.0015	0.15
TCP	0.05	0.10	0.0010	0.05	0.15	0.10	0.0015	0.15

preserves the same perturbation radius used by the sharpness-aware baselines ($\rho = 0.05$ or 0.10, depending on the prompt learner), retains the regularization strength of F-SAM ($\lambda = 0.15$), and adopts a slightly stronger correction factor than SAGM ($\alpha = 0.0015$ vs. 0.0010). This combination yields a more favorable balance between optimization stability and generalization, which is reflected in the results: across all five prompt learning methods, SAMPLe achieves the best average harmonic mean and, in most cases, the strongest new-class accuracy, while maintaining competitive or improved base-class performance. The gains are especially pronounced on datasets with larger base-to-new generalization gaps, such as Flowers102, DTD, EuroSAT, StanfordCars, UCF101, and FGVCAircraft, where SAMPLe consistently improves the balance between memorization of base classes and transfer to unseen categories. Importantly, these improvements are obtained with fixed hyperparameter settings for each method across all 11 datasets, indicating that the performance gains are not due to dataset-specific tuning, but rather to the optimizer itself. Overall, the evidence suggests that SAMPLe more effectively controls loss sharpness and reduces over-specialization, leading to stronger and more consistent base-to-new generalization than SAM, F-SAM, and SAGM.

MaPLe We train MaPLe using a staged training strategy to improve generalization and stability. The training begins with optimizing the learnable prompts and the projection layers using standard optimization methods such as SGD or Adam. Once the initial training phase converges, we freeze the projection layers and continue training only the learnable prompts using SAMPLe, which is designed for prompt learning. This step ensures that the prompts adapt to both flatter minima and low values of loss, which enhances generalization while preserving the alignment between vision and text representations. By decoupling the optimization of prompts and projections, our method allows each component to specialize without disrupting the overall model convergence. The results in Table. 1 demonstrates that the SAMPLe optimizer performs well in staged strategy, leading to significantly better performance than the benchmark and other optimization methods, including SAM, F-SAM, and SAGM. Specifically, our approach achieves higher accuracy across the HM value,

Table 6: term-wise ablation study of objective function (Eq. (12)) average performance over all 11 datasets using CoPrompt method.

	Base	**New**	**HM**
L	84.00	77.23	80.48
L_p	84.61	77.98	80.99
$L + L_p - \alpha(\nabla L \cdot \nabla L_p)$	85.12	78.47	81.49
$L + L_p - \alpha(\nabla L \cdot \nabla L_p) + \alpha\xi\sigma(\nabla L_p \cdot \nabla^{\mathcal{F}} L)$	85.62	79.03	82.02

Co-Prompt Following the staged training strategy introduced in MaPLe, we apply SAMPLe to CoPrompt by first training the model using standard optimization and jointly updating the learnable prompts, projection layers, and adapter networks. After convergence, we freeze the adapters and projection layers and then optimize only the learnable prompts using SAMPLe. This allows them to refine their representations while preserving the alignment learned in earlier stages. This staged optimization ensures that the learned prompts remain discriminative and generalize well across unseen class distributions.

9 Term-wise ablation study

To assess the effectiveness of the objective function presented in Eq. 12, we perform an ablation study against each term. In particular, we compare ERM, $L(\theta, \mathcal{D})$, SAM, $L_p(\theta; \mathcal{D})$, SAGM, $L(\theta; \mathcal{D}) + L_p(\theta; \mathcal{D}) - \alpha\big(\nabla L_p(\theta; \mathcal{D}) \cdot \nabla L(\theta; \mathcal{D})\big)$, and SAMPLe, that consists of all terms. All items outperform ERM in the HM column, underscoring the effectiveness of SAM-based methods for prompt learning. Moreover, the exploration term in SAMPLe consistently improves performance across Base, New, and HM, highlighting the importance of balanced learning that accounts for both exploitation and exploration in prompt learning.