Sparse Tuning Enhances Plasticity in PTM-based Continual Learning

Huan Zhang[1], Shenghua Fan[1], Shuyu Dong[2], Yujin Zheng[1], Dingwen Wang[1*], Fan Lyu[3*]

[1]School of Computer Science, Wuhan University, Wuhan, China
[2]State Key Laboratory of Green Pesticide, Central China Normal University, Wuhan, China
[3]New Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{cszhanghuan, fanshenghua, zhengyujin, wangdw}@whu.edu.cn; dongshuyu@ccnu.edu.cn;fanlyu@ia.ac.cn

Abstract

Continual learning with Pre-trained Models (PTMs) holds great promise for efficient adaptation across sequential tasks. However, most existing approaches freeze PTMs and rely on auxiliary modules like prompts or adapters, limiting model plasticity and leading to suboptimal generalization when facing significant distribution shifts. While full fine-tuning can improve adaptability, it risks disrupting crucial pre-trained knowledge. In this paper, we propose Mutual Information-guided Sparse Tuning (MIST), a plug-and-play method that selectively updates a small subset of PTM parameters, less than 5%, based on sensitivity to mutual information objectives. MIST enables effective task-specific adaptation while preserving generalization. To further reduce interference, we introduce strong sparsity regularization by randomly dropping gradients during tuning, resulting in fewer than 0.5% of parameters being updated per step. Applied before standard freeze-based methods, MIST consistently boosts performance across diverse continual learning benchmarks. Experiments show that integrating our method into multiple baselines yields significant performance gains.

Code — https://github.com/zhwhu/MIST

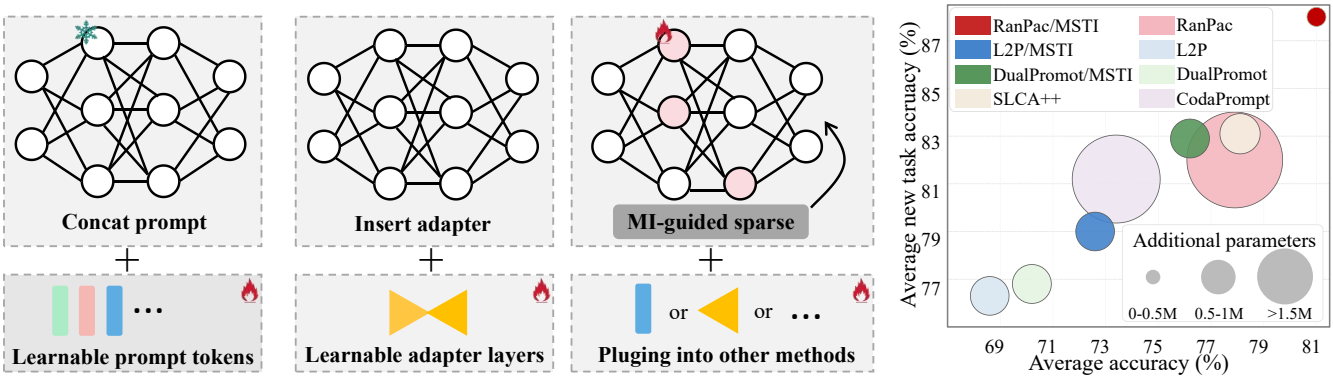
Introduction

Continual Learning (CL) is a paradigm in which tasks are learned sequentially, aiming to reduce forgetting of previously acquired knowledge while integrating new information. Recently, Pre-Trained Models (PTMs) (Han et al. 2021) have shown potential to enhance learning efficiency in CL tasks. By fine-tuning, PTMs can be easily adapted to various downstream tasks, enabling continual learners to acquire new task-specific knowledge more effectively and improving resilience to catastrophic forgetting (Sun et al. 2022; Lyu et al. 2021). One important challenge of PTMs in CL lies in how to effectively adapt to incremental tasks without harming the generalization ability of PTMs.

A common practice is to freeze the PTM and introduce additional learnable parameters to adapt the frozen PTM to new tasks. These methods can typically be categorized into two types: prompt-based methods and adapter-based methods. Prompt-based methods, such as L2P (Wang et al. 2022b) and DualPrompt (Wang et al. 2022a) introduce additional learnable prompt pools, which dynamically guide the frozen pre-trained layers to accommodate incremental tasks. Adapter-based methods, such as APER (Zhou et al. 2025) and RanPAC (McDonnell et al. 2023), adapt the frozen PTM by introducing additional adapters during the initial incremental stage to bridge the domain gap between pre-trained representations and incremental task distributions. In summary, most PTM-based CL methods typically freeze PTMs during incremental learning, relying heavily on the pure pre-trained knowledge for downstream adaptation. However, new task distributions may deviate from the encoded PTM knowledge, the freezing backbone struggles to generalize effectively across all tasks, that is, poor plasticity (Zhang et al. 2023). Since freezing PTMs can reduce model plasticity, it raises the question of why some methods that fine-tune PTMs still achieve suboptimal performance. A possible explanation lies in the fact that heuristic fine-tuning or fully updating all parameters can lead to the loss of crucial parameters, diminishing the effectiveness of PTMs themselves. To avoid this, effective sparse tuning is needed, which selectively updates only a subset of parameters, thus preserving key knowledge within PTMs. *The goal of this paper is to propose a sparse update method that strikes a balance between effective adaptation to new tasks and the preservation of generalization in PTM-based CL methods*.

Therefore, how to selectively identify important parameters in PTM-based CL remains a key challenge. To address this, we investigate the underlying behavior of PTMs through a probabilistic analysis. We theoretically and empirically demonstrate that the parameters sensitive to the MI objective can effectively model task-specific knowledge while minimizing disruption to the original knowledge structure of the PTM. Motivated by this, we introduce a simple yet effective plug-and-play method named **Mutual Information-guided Sparse Tuning** (MIST). Specifically, before training each incremental task with other freeze-based methods, we first determine the sensitivity of each PTM parameter to the MI objective. We then select the top 5% most sensitive parameters for MI-guided tuning , which enables the model to fully adapt to the new task distribution while maximally preserving the structural knowledge encoded in the PTM. During this process, we apply strong

*Corresponding authors: Dingwen Wang and Fan Lyu.



(a) Prompt-based CL methods (b) Plasticity Comparison

Figure 1: (a) 🔥 indicates learnable parameters, while ❄ denotes frozen parameters. MIST leverages MI for pre-adaptation, enabling it to be plugged into other methods. (b) Comparison of different methods in terms of average accuracy, new task accuracy, and additional parameters. MIST achieves superior accuracy through MI-guided sparse tuning.

regularization by randomly dropping the gradients of 90% of the selected parameters in each mini-batch, thereby updating only 0.5% of the parameters per batch. After this tuning stage, we freeze the PTM and proceed with the original freeze-based method for continual learning. We insert our approach into six representative freeze-based methods and conduct experiments on serval datasets. Results show consistent performance improvements with MIST, particularly on datasets with large distribution shifts from the pretraining domain. For example, SimpleCIL with MIST achieves 17.9% and 15.7% improvements on Split-ImageNet-R and Split-Cars, respectively. The contributions of this paper are summarized as follows:

(1) We study PTM-based CL from a probabilistic and information-theoretic perspective, and theoretically and empirically demonstrate, through MI techniques, PTMs can effectively adapt to new tasks by updating only a small subset of parameters.

(2) We introduce a simple yet effective plug-and-play method named Mutual Information-guided Sparse Tuning (MIST), which can be integrated into freeze-based methods to provide significant performance improvements.

(3) We incorporate MIST into six representative PTM-based CL methods and evaluate them across five benchmark datasets. All methods achieve consistent performance gains after integrating MIST, highlighting its broad applicability and effectiveness. The empirical results clearly demonstrate the superiority of MIST.

Related Work

Continual Learning on Pre-trained Models. Recently, advancements in PTMs and their exceptional performance in adapting to downstream tasks have inspired researchers to investigate how PTMs can be adapted for continual learning across sequential tasks. Prompt-based methods learn continual prompts to provide fixed PTMs with additional instruction. DualPrompt (Wang et al. 2022a) combines task-shared and task-specific prompts to achieve an effective balance between adaptability and mitigating forgetting, while CODA-Prompt (Smith et al. 2023) leverages contrastive learning-based prompts to enhance the representation learning of PTMs for improved task adaptation. HiDe-Prompt(Wang et al. 2024) optimizes hierarchical components by combining task-specific prompts and representation statistics, enhanced with a contrastive regularization strategy. Adapter-based methods also freeze the PTM and introduce additional lightweight modules for task-specific adaptation. SLCA++(Zhang et al. 2024a) sequentially fine-tunes low-rank LoRA matrices with a small learning rate to avoid disrupting the pre-trained features. RanPAC(McDonnell et al. 2023) adapts the PTM during the first task to enhance downstream performance, while APER (Zhou et al. 2025) further combines the adapted PTM with the original frozen PTM to jointly extract features, aiming to balance generalization and task-specific learning.

Mutual Information in Machine Learning. With the advancement of deep learning(Lyu et al. 2024; Yin et al. 2025; Zhang et al. 2024b; Du et al. 2024), mutual information (MI) has become an important tool for capturing both linear and nonlinear dependencies between variables, supporting tasks such as feature selection, clustering, and model optimization (Zhang, Liu, and Song 2023; Vinh, Epps, and Bailey 2009; Tishby and Zaslavsky 2015). In particular, InfoNCE (Oord, Li, and Vinyals 2018) has emerged as a widely used lower-bound estimator of MI in representation learning. Building on this, recent works have applied InfoNCE-based MI objectives to continual learning. For example, Guo et al.(Guo, Liu, and Zhao 2022) used InfoNCE to measure MI between samples to mitigate catastrophic forgetting, while Li et al.(Li et al. 2023) maximized MI between outputs of current and previous models for knowledge distillation. In this work, we construct an MI-guided sparse tuning to identify important parameters during incremental fine-tuning in PTM-based CL, enabling more targeted and generalization-preserving updates.

Rethinking PTMs in Continual Learning

Continual Learning with PTMs and the Impact of Freezing PTMs

Preliminaries. Given a sequence of tasks with data $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^T\}$, where $\mathcal{D}^t = \{(x_i, y_i)\}_{i=1}^{n_t}$ with n_t input pair, sample x and its corresponding label y. Different tasks are with disjoint label spaces across tasks: $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$. At the training stage t, only the current task dataset $\mathcal{D}^t$ is available. The model is denoted as f_θ, where θ is parameter.

In PTM-based CL, θ is initialized from a PTM trained on a large-scale dataset and is typically frozen during adaptation. The model adapts to new tasks by introducing additional parameters, which can take the form of prompts or adapters, depending on the chosen tuning strategy. Despite their structural differences, both methods freeze the backbone and optimize lightweight parameters for efficient adaptation. Freezing the PTM parameters θ limits adaptability to new tasks, shifting the burden to auxiliary modules like prompts or adapters.

Freezing PTMs in prompt tuning. In prompt-based tuning (the left subfigure in Fig. 1(a)), learnable prompts ϕ are prepended or injected into the input embedding, yielding output $p(y \mid x; \theta, \phi) = f_\theta(P_\phi(x))$. The gradient of the log-likelihood with respect to ϕ follows the chain rule:

$$\frac{\partial \log p(y \mid x; \theta, \phi)}{\partial \phi} = \frac{1}{p(y \mid x; \theta, \phi)} \cdot \frac{\partial f_\theta(P_\phi(x))}{\partial x} \cdot \frac{\partial P_\phi(x)}{\partial \phi},$$

where $P_\phi(x)$ represents the modified input obtained by injecting the learnable prompt ϕ into the feature space of x. Since θ is frozen, the Jacobian term $\partial f_\theta / \partial x$ is fixed and reflects the model's sensitivity to input perturbations. When this Jacobian is close to zero in directions that encode task-specific features, the gradient signal received by ϕ is significantly diminished, regardless of its expressive capacity (Qiao et al. 2023; Fu et al. 2024; Gao et al. 2023). This severely restricts the effectiveness of prompt-based tuning, especially under distribution shifts where new tasks require directions outside the pre-trained manifold.

Freezing PTMs in adapter-based tuning. Adapter-based tuning (the center subfigure in Fig. 1(a)) inserts trainable adapters ψ into the intermediate layers, resulting in $p(y \mid x; \theta, \psi) = f_{\theta,\psi}(x)$. The gradient of ψ is:

$$\frac{\partial \log p(y \mid x; \theta, \psi)}{\partial \psi} = \frac{1}{p(y \mid x; \theta, \psi)} \cdot \frac{\partial f_{\theta,\psi}(x)}{\partial \psi} \quad (1)$$

where $f_{\theta,\psi}$ means the backbone modified by inserting adapter modules. From Eq. (1), adapter's influence must propagate through the remaining frozen layers to affect the output. If the PTM is not responsive to the features injected by adapters, particularly when such features lie outside the pre-trained distribution, then the resulting gradient with respect to ψ is similarly attenuated (Qiao et al. 2024; Son et al. 2024; Nowak et al. 2024).

In summary, despite using different mechanisms, both prompt- and adapter-based methods suffer from gradient suppression due to the fixed representational structure of the frozen PTM. The frozen PTM acts as a bottleneck that limits the flow of gradients to newly introduced parameters. This hampers the model's ability to adapt to novel tasks.

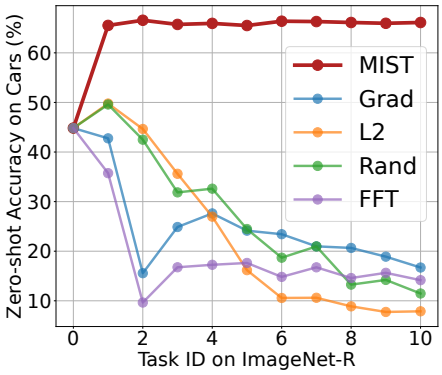


Figure 2: Zero-shot accuracy using different sparse update strategies. After training on each ImageNet-R task, we freeze the PTM and train only on a new classification head on the full Cars dataset. Higher accuracy reflects better generalization capability of the PTM learned from each task.

Continual Tuning on PTMs

To enhance plasticity in PTM-based CL, some works have explored direct fine-tuning. However, studies (Kingma and Ba 2014; Zhang et al. 2024a) show that this often results in significant performance drops, particularly under distribution shifts. To analyze this, we begin by examining the gradient of the log-likelihood:

$$\begin{aligned}\frac{\partial \log p(y \mid x; \theta)}{\partial \theta^i} = &- \frac{1}{p(x, y; \theta)} \cdot \frac{\partial p(x, y; \theta)}{\partial \theta^i} \\ &+ \frac{1}{p(x; \theta)} \cdot \frac{\partial p(x; \theta)}{\partial \theta^i},\end{aligned} \quad (2)$$

where $\theta^i \in \theta$ denotes an arbitrary parameter in θ. In Eq. (2) the term $\partial p(x, y; \theta)/\partial \theta^i$ encourages task-specific alignment through updates to $p(x, y; \theta)$, while the term $\partial p(x; \theta)/\partial \theta^i$ reflects how parameter changes disturb the pre-trained input distribution $p(x; \theta)$. Excessive increase or decrease in the second term can distort the underlying feature, resulting in poor generalization.

Existing tuning strategies, including full fine-tuning, naive partial fine-tuning, and Fisher-guided partial fine-tuning, share a common limitation: they inadvertently perturb the pre-trained structure by amplifying the term $\partial p(x; \theta)/\partial \theta^i$, thereby severely reducing the generalization of the PTM. As illustrated in Fig. 2, this drawback leads to a continual decline in zero-shot accuracy on the Cars dataset as tasks progress, clearly indicating progressive loss of generalization ability. Existing strategies either overfit to new tasks or disrupt pre-trained generalization due to their inability to disentangle task-relevant gradients from those that compromise structural stability. This motivates the need for a more principled tuning strategy that explicitly controls the influence on each gradient component in Eq. (2). This motivates the need for a more principled tuning strategy that explicitly controls the influence on each gradient component.

Method

Mutual Information Analysis in PTM-based CL

MI (Kraskov, Stögbauer, and Grassberger 2004; Lei et al. 2023) is a fundamental concept in information theory and has been widely adopted in machine learning. By maximizing the MI $I(X;Y)$ between input X and output Y, MI explicitly quantifies the statistical dependency between features and labels (Guo, Liu, and Zhao 2022). Formally, the MI $I(X;Y)$ is defined as:

$$I(X;Y) = \mathbb{E}_{(x,y)\sim\mathcal{D}^t}\left[\log \frac{p(x,y;\theta)}{p(x;\theta)p(y;\theta)}\right], \quad (3)$$

where $p(y;\theta)$ denotes the prior probabilities of the target classes. Due to the normalization constraint of probability distributions, we have $\sum_x p(x;\theta) = 1$ and $\sum_y p(y;\theta) = 1$. Under this constraint, the gradient of the MI with respect to θ^i can be simplified as:

$$\frac{\partial I(X;Y)}{\partial \theta^i} = \mathbb{E}_{(x,y)\sim\mathcal{D}^t}\left[\frac{\partial p(x,y;\theta)}{\partial \theta^i}\log \frac{p(x,y;\theta)}{p(x;\theta)p(y;\theta)}\right]. \quad (4)$$

In this paper, we explore how MI contributes to the trade-off between plasticity and generalization in PTM-based CL, and make two observations.

(1) **Mutual Information Gradients: Stable Adaptation with Minimal Interference**. Compared with CE gradients, MI gradients induce less disruption to the pre-trained feature space. Both gradients, as shown in Eq. (2) and Eq. (4), include the term $\partial p(x,y;\theta)/\partial\theta^i$, which accounts for task-specific supervision. However, the CE gradient additionally involves the marginal term $\partial p(x;\theta)/\partial\theta^i$, which directly modifies the input distribution learned by the PTM. This term does not appear in MI gradients due to the probabilistic normalization constraint imposed by mutual information objectives, thereby naturally preserving the structural integrity of the input features.

(2) **Diverse Batches Improve Gradient Stability under MI Objectives**. The MI gradient formulation assumes a normalization condition $\sum_x p(x;\theta) = 1$, which holds exactly only when the full data distribution is observed. In practice, this assumption is better approximated when batches contain a diverse and representative set of samples. Consequently, using larger and more varied batches helps reduce gradient estimation bias and further mitigates unintended shifts in the pre-trained representation space during adaptation.

In summary, MI provides a more stable optimization objective than CE for CL with PTMs. Unlike CE gradients, which include the marginal term $\partial p(x;\theta)/\partial\theta^i$ and may disrupt the pre-trained input distribution, MI gradients inherently avoid this due to normalization constraints, preserving feature integrity. Additionally, MI benefits from diverse batches, which better approximate the underlying data distribution and reduce gradient bias. Together, these properties enable MI to strike a more effective balance between plasticity and stability during adaptation. *While MI enables a better plasticity–stability trade-off, directly replacing CE for full fine-tuning may still lead to information loss and high computational cost due to large-scale updates.* To address this, we next introduce a lightweight MI-based method that selectively tunes a small parameter subset and can be flexibly integrated as a plugin into existing PTM-based CL methods, including prompt-based and adapter-based approaches.

Mutual Information-Guided Sparse Tuning (MIST): A Plug-and-Play Solution

In this subsection, we introduce Mutual Information-guided Sparse Tuning (MIST), a plug-and-play pre-adaptation framework compatible with a wide range of PTM-based CL methods, including those based on prompt tuning and adapters. MIST acts as a pre-adaptation stage that sparsely fine-tunes the PTM before one freeze-based method. Specifically, it identifies the top-$k\%$ most MI-sensitive parameters through gradient-based sensitivity analysis, and selectively fine-tunes them using a mutual information objective. This pre-adaptation helps reshape the feature space with minimal interference to the pre-trained structure.

To efficiently estimate the sensitivity of each parameter $\theta^i \in \theta$ to the MI objective, we adopt the MI-based Fisher Information Matrix (Kirkpatrick et al. 2017) as an importance measure. While computing exact gradients over the entire task is computationally intensive, the sample distribution within a task is typically uniform in CL, enabling a batch-wise approximation:

$$F_{\mathrm{MI}} = \left(\frac{\partial \mathcal{L}_{\mathrm{MI}}^{\mathcal{D}t}}{\partial\theta}\right)^2 \approx F'_{\mathrm{MI}} = \left(\sum\nolimits_{j=1}^{B_j \leftarrow \mathcal{D}_t} \frac{\partial \mathcal{L}_{\mathrm{MI}}^{B_j}}{\partial\theta}\right)^2, \quad (5)$$

where $\mathcal{L}_{\mathrm{MI}}^{\mathcal{D}_t}$ is the MI loss computed over task $\mathcal{D}_t$, and B_j represents the j-th mini-batch sampled from $\mathcal{D}_t$. In practice, we identify the top $k\%$ of parameters with the highest F'_{MI} values as the most MI-sensitive parameters, denoted by $\mathcal{M}$:

$$\mathcal{M} = \left\{\theta^i \in \theta \mid \mathrm{rank}(F'_{\mathrm{MI}}(\theta^i)) \le \lfloor k\% \cdot |\theta| \rfloor\right\}. \quad (6)$$

where $\mathrm{rank}(\cdot)$ denotes the descending order index, and $\lfloor\cdot\rfloor$ denotes the floor function. That is, we select the top $k\%$ parameters with the highest MI-based importance scores. Given that only a small subset of parameters is updated in each batch and that PTMs are typically initialized near an optimal solution (Zhang et al. 2023; Zhou et al. 2025), we compute F'_{MI} once at the beginning of each task and reuse $\mathcal{M}$ throughout the pre-adaptation phase to ensure both efficiency and effectiveness. With the MI-sensitive parameter subset $\mathcal{M}$ identified, we proceed to the pre-adaptation stage using an MI-based objective to minimize disruption to the pre-trained feature structure. However, computing the exact MI loss is challenging in practice, as both joint and marginal distributions $p(x,y;\theta)$ and $p(x;\theta)$ are typically intractable. Inspired by OCM (Guo, Liu, and Zhao 2022), we adopt the supervised InfoNCE loss to construct the MI objective:

$$\mathcal{L}_{MI} = \sum_{i=1}^{|B|} \frac{A_i}{3|B|\sum_{s=1}^{|B|}\mathbf{1}(y_s = y_i)}, \quad (7)$$

where $X, Y \in \{x_i, y_i\}_{i=1}^{|B|}$. And A_i is given by:

$$-\sum_{y_k=y_i}\log\frac{g(x_i,x_k)\cdot g(x_i,x_k')\cdot g(x_i',x_k)}{\left(\sum_{j=1}^{|B|} g(x_i,x_j) + g(x_i,x_j') + g(x_i',x_j)\right)^3},$$

Algorithm 1: **MI-guided Sparse Tuning**

Require: Continual tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, pre-trained model f_θ, select rate $k\%$, dropout rate $d\%$

for task $t = 1$ to T **do**
  Compute the Fisher matrix F'_{MI} using Eq. (5)
  Generate parameters group $\mathcal{M}$ by selecting top $k\%$ parameters using Eq. (6)
  for each training mini-batch iteration **do**
    Compute MI loss using Eq. (7)
    Generate dropped parameters group $\mathcal{M}'$ by dropping $d\%$ parameters in $\mathcal{M}$
    Update the parameters in $\mathcal{M}'$
  end for
  Training other method's additional parameters (e.g. prompt, adapter and classifer) on $\mathcal{D}_t$
end for

where $g(x_i, x'_j) = e^{f_\theta(x_i)^T f_\theta(x'_j)/\tau}$ is the similarity of two samples, τ is temperature, x'_j is an augmentation view of sample x_j. By optimizing Eq. (7), we effectively maximize the MI I(X;Y), thereby modeling $p(x, y; \theta)$ in a task-discriminative manner.

To further reduce the number of parameters being updated, we introduce a lightweight regularization strategy called Gradient Dropout. During each batch of the pre-adaptation stage, we randomly drop $d\%$ of the MI-sensitive parameters in $\mathcal{M}$, resulting in only $k\% \times d\%$ of total parameters being updated per batch. In practice, we set $k\% = 5\%$ and $d\% = 90\%$, yielding updates to merely 0.5% of all parameters per batch. This stochastic suppression addresses a critical issue, i.e., repeatedly updating a fixed subset of parameters can constrain the model's exploration of the optimization landscape, leading to biased shifts in the feature space. By introducing randomness into the gradient flow, Gradient Dropout promotes more diverse and balanced parameter updates, reduces co-adaptation, and further stabilizes the pre-trained representation by mitigating local bias and limiting excessive perturbations.

Plugging MIST into PTM-based Continual Learning: The Algorithm

As shown in Algorithm 1, we begin by temporarily unfreezing the PTM f_θ and estimating the sensitivity of each parameter with respect to the MI objective. Based on this, we select the top $k\%$ most sensitive parameters to form the update set $\mathcal{M}$. During the MI-guided tuning phase, we apply Gradient Dropout. After a few epochs of such sparsified adaptation, the PTM is refrozen, and the standard freeze-based CL procedure resumes. This pre-adaptation phase introduces small computational overhead and is compatible with a wide range of PTM-based CL methods. For prompt-based approaches, MIST is applied to the PTM prior to prompt tuning. For adapter-based methods, we do not modify the PTM or perform any initial task-specific fine-tuning. Instead, MIST is used as a lightweight pre-adaptation step, after which classifier training proceeds as originally designed.

Experiment

Experimental Setups

Benchmark. We consider five representative benchmark datasets and randomly split each of them into 10 disjoint tasks. Including CIFAR-100 (Krizhevsky and Hinton 2009), ImageNet-R (Hendrycks et al. 2021a), ImageNet-A (Hendrycks et al. 2021b), CUB-200 (Wah et al. 2011) and Cars-196 (Krause et al. 2013). Performance is evaluated using the standard CL metric, *Average Accuracy* (Shim et al. 2021), defined as: $A_t = \frac{1}{t}\sum_{i=1}^{t} R_{t,i}$, where $R_{t,i}$ denotes the classification accuracy on the i-th task after training on the t-th task. We report both A_T and $\bar{A}$ in the main paper. Here, $\bar{A}$ denotes the mean of A_t over all tasks: $\bar{A} = \frac{1}{T}\sum_{t=1}^{T} A_t$. It reflects the average accuracy of all classes seen so far after each incremental task.

Implementation. Following previous works (Wang et al. 2022b,a), we adopt a pre-trained ViT-B/16 backbone (Dosovitskiy et al. 2020) for all baselines. Except for OCM (Guo, Liu, and Zhao 2022), no methods use a buffer; OCM is implemented in an offline setting with a buffer size of 1000. We follow the original implementations by employing the Adam optimizer for L2P, DualPrompt, and CoDA-Prompt, and the SGD optimizer for all other baselines. Our method, MIST, is inserted as a plug-in module before each selected baseline and is trained for 20 epochs using the SGD optimizer with a learning rate of 0.0001. In the MI-based selection stage, we select the top $k\% = 5\%$ most sensitive parameters. For each mini-batch, we further apply a dropout rate of $d\% = 90\%$ to the selected parameters, resulting in only 0.5% of total parameters being updated per batch. MIST solely optimizes the MI loss (Eq. 7), with the temperature τ set to 0.5. For each task, MIST first conducts this sparse fine-tuning, after which the corresponding baseline resumes training using its original configuration. We adopt this setting consistently across all datasets in our experiments.

Experimental Results

Overall performance. To assess the versatility of MIST, we plug it into six representative freeze-based methods: L2P, DualPrompt, SLCA, EASE, RanPAC, and SimpleCIL. Among them, L2P and DualPrompt are prompt-based methods that freeze the PTM and learn token-like prompts for adaptation. SimpleCIL does not involve any parameter tuning and directly trains a prototype classifier on frozen representations. RanPAC and EASE are adapter-based methods that inserts and fine-tunes lightweight modules in the PTM. SLCA adopts a full fine-tuning strategy with a reduced learning rate. As shown in Table 1, incorporating MIST consistently improves all methods across all datasets. For example, DualPrompt/MIST achieves accuracy gains of +1.6%, +6.0%, +4.4%, +1.4%, and +11.2% on the five datasets, respectively. Furthermore, we observe that many methods perform poorly on the Cars196 dataset. For instance, the A_T of L2P and SimpleCIL are only 39.6% and 27.8%, respectively. This is mainly because pretraining knowledge offers

Method	CIFAR100		ImageNet-R		ImageNet-A		CUB200		Cars196	
	$\bar{A}$	A_T	$\bar{A}$	A_T	$\bar{A}$	A_T	$\bar{A}$	A_T	$\bar{A}$	A_T
EWC (Kirkpatrick et al. 2017)	53.2	41.2	42.3	26.2	22.0	7.9	57.6	37.8	46.7	24.3
OCM (Guo, Liu, and Zhao 2022)	63.6	37.0	77.0	66.7	58.1	44.4	85.1	71.7	65.3	50.5
CODA-Prompt (Smith et al. 2023)	91.3	86.9	78.5	73.4	63.9	52.7	84.1	79.3	52.1	45.4
SLCA++ (Zhang et al. 2024a)	94.1	91.5	83.0	77.5	67.1	58.7	91.0	86.7	79.2	73.8
APER(Adapter) (Zhou et al. 2025)	83.9	85.9	74.2	66.9	62.4	52.1	90.5	85.6	52.8	40.5
L2P (Wang et al. 2022b)	86.7	83.3	74.5	68.6	53.9	44.9	81.7	67.4	53.9	39.6
+MIST	89.1	86.1	77.5	72.6	56.9	51.2	82.3	71.8	63.4	52.7
DualPrompt (Wang et al. 2022a)	87.4	84.0	75.2	70.2	55.7	47.7	82.3	68.8	53.2	41.6
+MIST	89.0	86.2	80.1	76.2	60.1	53.3	83.1	70.2	62.4	52.8
SLCA (Zhang et al. 2023)	94.1	91.5	81.7	77.0	67.9	59.3	90.9	84.7	76.9	67.7
+MIST	94.8	92.2	83.6	80.0	69.9	61.0	92.0	87.3	80.7	74.6
EASE (Zhou et al. 2024)	91.8	87.4	81.1	75.1	65.3	54.9	90.9	85.8	38.4	27.3
+MIST	92.2	88.2	81.8	76.3	66.0	56.9	91.8	87.0	57.0	47.2
SimpleCIL (Zhou et al. 2025)	87.1	81.3	61.1	54.3	59.8	48.5	90.9	85.6	38.8	27.8
+MIST	87.9	82.1	79.5	72.2	65.5	55.3	91.6	86.8	57.0	43.5
RanPAC (McDonnell et al. 2023)	94.0	90.8	83.2	77.9	70.1	61.4	92.6	88.9	82.8	74.6
+MIST	95.3	92.4	84.9	81.0	72.5	62.5	93.6	90.4	83.0	76.4

Table 1: Performance comparison on various datasets.

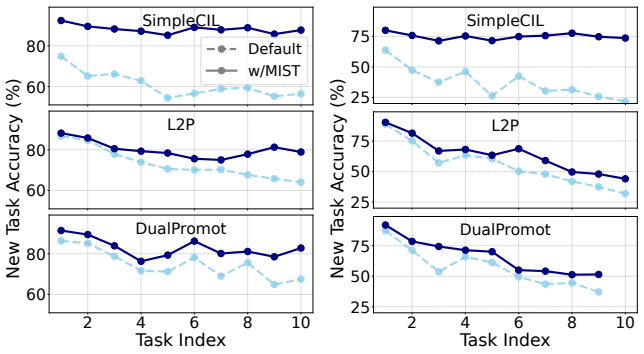
limited utility for complex fine-grained vehicle classification, making it particularly challenging for models to adapt to such new domains. After inserting MIST, the final accuracies of L2P and SimpleCIL increase by +13.1% and +15.7% respectively, indicating that MIST effectively enhances the model's ability to align with domain-specific structures before classifier training. Among all methods, RanPAC/MIST achieves the best overall performance, indicating that even well-designed adapter-based methods benefit from the MI-guided tuning stage.

Effect of MIST. To better understand the effect of MIST on PTM-based CL performance, we visualize both the new task accuracy (Figure 3) and the incremental accuracy (Figure 4). New task accuracy refers to the accuracy on the newly learned task, while incremental accuracy denotes the average accuracy over all tasks learned so far. As shown, MIST improves the learning effectiveness across all inserted methods. Specifically, in Figure 3, the new task accuracy increases significantly for all methods after integrating MIST. This improvement indicates that the pre-adaptation phase provided by MIST helps the PTM align more effectively with task-specific distributions. Correspondingly, Figure 4 shows that MIST also leads to notable gains in incremental accuracy across all tasks. This is attributed to the improved learning efficiency on new tasks, which in turn contributes to higher cumulative accuracy when evaluated on all seen classes. Taken together, these results highlight the effectiveness of MIST as a plug-in component that improves the task-specific learning capacity of freeze-based methods while preserving their stability, ultimately leading to consistent performance gains in CL scenarios.

Comparison with naive sparse tuning strategies. We integrate several pre-adaptation strategies into RanPAC, including full fine-tuning (FFT) of all parameters, top 5% selection based on gradient magnitude (Grad) or parameter norm (L2), and random 5% selection (Rand). As shown in Table 2(a) and (b), all these alternatives perform significantly worse than MIST. This is because MIST leverages mutual information (MI) to assess parameter sensitivity, enabling effective task adaptation while minimizing disruptions to the pre-trained representations. In contrast, the alternative methods ignore the importance of preserving pre-trained knowledge during parameter selection, making them vulnerable to catastrophic forgetting and performance collapse. Overall,

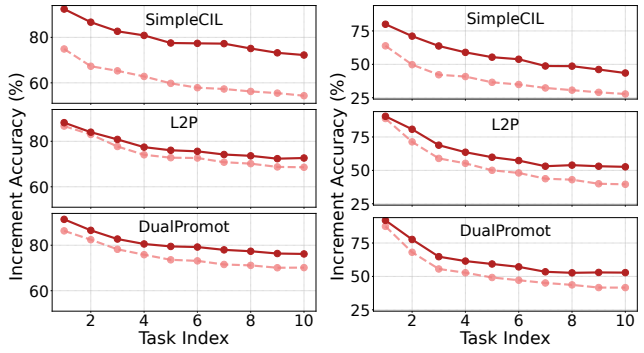
	Method	ImageNet-R		Cars196	
		$\bar{A}$	A_T	$\bar{A}$	A_T
	RanPAC	83.2	77.9	82.8	74.6
(a)	+FFT	57.6	36.7	31.3	11.2
	+Rand	52.4	11.8	33.7	10.9
	+L2	39.0	14.4	29.6	10.4
(b)	+Grad	54.8	33.9	29.9	10.6
	+Grad+ML	76.7	67.1	64.2	40.3
	+Grad+ML+Drop	77.5	68.3	64.9	42.0
(c)	+MS	75.5	66.8	76.0	66.2
	+MS+ML	82.2	76.7	81.3	72.1
	+MS+ML+Drop	**84.9**	**81.0**	**83.0**	**76.4**

Table 2: Comparison of tuning strategies used for pre-adaptation in RanPAC. "MS" refers to MI-based sparse selection, "ML" to MI loss, and "Drop" to dropout. "MS+ML+Drop" corresponds to our MIST implementation. (a) shows naive selection strategies, (b) adds MI loss and dropout to a naive strategy, and (c) uses MI-based sparse selection combined with MI loss and dropout.



(a) Imagenet-R (b) Cars-196

Figure 3: New task accuracy.



(a) Imagenet-R (b) Cars-196

Figure 4: Incremental accuracy.

MIST offers a more balanced adaptation path by jointly preserving plasticity and stability, demonstrating its superiority as a general plug-in pre-adaptation module.

Ablation study. Table 2(c) presents the ablation study of MIST. When applying only MI-based sparse selection ("MS"), the model achieves the lowest accuracy among the three. Although sparse selection reduces parameter interference, the optimization remains guided by the cross-entropy loss, which fails to explicitly preserve the pre-trained feature distribution. Nevertheless, this approach still outperforms alternative selection strategies. Upon introducing the MI loss ("MS+ML"), performance improves, indicating that the MI objective effectively guides the model toward downstream distributions while retaining generalization. Finally, adding gradient dropout ("MS+ML+Drop") yields the highest accuracy across all settings, as it regularizes the update path and mitigates overfitting to static parameter importance. To further validate the effectiveness of MI-based sparse selection, we also apply MI loss and dropout to the Grad-based strategy in Table 2(b). Although this leads to some performance gains, its results remain clearly inferior to MIST. This highlights the unique advantage of MI-guided selection and the complementary role of the three components in MIST. These findings confirm that MI sparsity, MI loss, and dropout contribute synergistically to overall performance gains.

Method	Δ P (M)	FLOPs (M)	Time (ms)
SLCA	85.40	171.6	12.3
RanPac	4.53	8.6	9.1
L2P	0.48	1.0	12.4
MIST	0.43	0.8	9.7

Table 3: Efficiency analysis.

Parameter efficiency analysis. Table 3 compares the efficiency of different methods in terms of (1) ΔP: the number of parameters updated per mini-batch (in millions), (2) FLOPs: flops for updating the selected parameters per batch, and (3) Time: time required to train a batch on an NVIDIA RTX 4090 GPU. Among the methods, SLCA performs full fine-tuning and thus has the highest update cost—both in terms of parameters (85.40M) and time (12.3ms). L2P only updates prompt tokens, but incurs additional overhead (12.4ms) likely due to its key-query matching mechanism during token routing. MIST, while updating only 0.43M parameters per task, incurs slightly higher computation time (9.7ms) compared to RanPAC. This is because computing the MI loss requires augmented views of each sample. In summary, MIST achieves the lowest update cost without introducing any additional parameters, making it easily pluggable into other methods.

Conclusion

In this paper, we investigate the fundamental challenge of balancing plasticity and generalization in PTM-based CL. We reveal that direct fine-tuning often compromises the pre-trained feature distribution, while existing freeze-based methods suffer from limited adaptability to new tasks. Through a theoretical lens grounded in MI, we analyze how gradients derived from MI objectives offer a more stable optimization path by avoiding unnecessary perturbations to the PTM. Motivated by this, we propose Mutual Information-guided Sparse Tuning, a lightweight and plug-and-play pre-adaptation strategy that selectively updates only the most informative parameters before each incremental task. By computing an MI-based Fisher Information Matrix, MIST identifies sensitive parameters, then applies strong gradient dropout to regularize the update path, enabling the PTM to better align with task-specific distributions while maintaining generalizable representations. Extensive experiments demonstrate that MIST can be seamlessly integrated into various freeze-based CL frameworks, consistently boosting performance across diverse datasets, especially under large distribution shifts. The limitation of MIST lies in its reliance on efficient approximation of the Fisher matrix. When the data within a task exhibits significant distributional variation, this approximation may become inaccurate, potentially compromising the effectiveness of MIST. In the future, we plan to explore more robust Fisher estimation techniques that can adapt to intra-task variation.

Acknowledgments

This paper was supported by the National Science Foundation of China (No. 62406323), China Postdoctoral Science Foundation (No. 2024M753496), and Postdoctoral Fellowship Program of CPSF (No. GZC20232993).

References

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, K.; Zhou, Y.; Lyu, F.; Li, Y.; Lu, C.; and Liu, G. 2024. Confidence self-calibration for multi-label class-incremental learning. In *European Conference on Computer Vision*, 234–252. Springer Nature Switzerland Cham.

Fu, D.; Vo, T. V.; Ma, H.; and Leong, T.-Y. 2024. Decoupled Prompt-Adapter Tuning for Continual Activity Recognition. *arXiv preprint arXiv:2407.14811*.

Gao, Q.; Zhao, C.; Sun, Y.; Xi, T.; Zhang, G.; Ghanem, B.; and Zhang, J. 2023. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11483–11493.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, 8109–8126. PMLR.

Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L.; et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2: 225–250.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8340–8349.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15262–15271.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Kraskov, A.; Stögbauer, H.; and Grassberger, P. 2004. Estimating mutual information. *Physical review E*, 69(6): 066138.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

Lei, X.; Xia, Y.; Wang, A.; Jian, X.; Zhong, H.; and Sun, L. 2023. Mutual information based anomaly detection of monitoring data with attention mechanism and residual learning. *Mechanical Systems and Signal Processing*, 182: 109607.

Li, X.; Wang, S.; Sun, J.; and Xu, Z. 2023. Variational data-free knowledge distillation for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Lyu, F.; Feng, W.; Li, Y.; Sun, Q.; Shang, F.; Wan, L.; and Wang, L. 2024. Elastic multi-gradient descent for parallel continual learning. *arXiv preprint arXiv:2401.01054*.

Lyu, F.; Wang, S.; Feng, W.; Ye, Z.; Hu, F.; and Wang, S. 2021. Multi-Domain Multi-Task Rehearsal for Lifelong Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 8819–8827.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and Van den Hengel, A. 2023. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36: 12022–12053.

Nowak, A. I.; Mercea, O.-B.; Arnab, A.; Pfeiffer, J.; Dauphin, Y.; and Evci, U. 2024. Towards Optimal Adapter Placement for Efficient Transfer Learning. *arXiv preprint arXiv:2410.15858*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Qiao, J.; Tan, X.; Chen, C.; Qu, Y.; Peng, Y.; Xie, Y.; et al. 2023. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*.

Qiao, J.; Zhang, Z.; Tan, X.; Qu, Y.; Zhang, W.; and Xie, Y. 2024. Gradient projection for parameter-efficient continual learning. *arXiv e-prints*, arXiv–2405.

Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J. 2021. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9630–9638.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11909–11919.

Son, H.; Son, Y.; Kim, C.; and Kim, Y. G. 2024. Not All Adapters Matter: Selective Adapter Freezing for Memory-Efficient Fine-Tuning of Language Models. *arXiv preprint arXiv:2412.03587*.

Sun, Q.; Lyu, F.; Shang, F.; Feng, W.; and Wan, L. 2022. Exploring Example Influence in Continual Learning. *Advances in Neural Information Processing Systems*.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, 1–5. IEEE.

Vinh, N.; Epps, J.; and Bailey, J. 2009. Information Theoretic Measures for Clusterings Comparison: Variants. *Properties, Normalization and Correction for Chance*, 18.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset.

Wang, L.; Xie, J.; Zhang, X.; Huang, M.; Su, H.; and Zhu, J. 2024. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 36.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022a. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 631–648. Springer.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Yin, H.; Feng, T.; Lyu, F.; Shang, F.; Liu, H.; Feng, W.; and Wan, L. 2025. Beyond Background Shift: Rethinking Instance Replay in Continual Semantic Segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 9839–9848.

Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2023. 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*, 19148–19158.

Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2024a. SLCA++: Unleash the Power of Sequential Fine-tuning for Continual Learning with Pre-training. *arXiv preprint arXiv:2408.08295*.

Zhang, H.; Lyu, F.; Fan, S.; Zheng, Y.; and Wang, D. 2024b. Constructing Enhanced Mutual Information for Online Class-Incremental Learning. *arXiv preprint arXiv:2407.18526*.

Zhang, P.; Liu, G.; and Song, J. 2023. MFSJMI: Multi-label feature selection considering join mutual information and interaction weight. *Pattern Recognition*, 138: 109378.

Zhou, D.-W.; Cai, Z.-W.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2025. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3): 1012–1032.

Zhou, D.-W.; Sun, H.-L.; Ye, H.-J.; and Zhan, D.-C. 2024. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23554–23564.