Compress to One Point: Neural Collapse for Pre-Trained Model-Based Class-Incremental Learning

Kun Wei, Zhe Xu, Cheng Deng*

School of Electronic Engineering, Xidian University, Xian 710071, China
{weikunsk, zhexu.xd, chdeng.xd}@gmail.com

Abstract

Class-Incremental Learning (CIL) requires an artificial intelligence system to learn different tasks without class overlaps continually. To achieve CIL, some methods introduce the Pre-Trained Model (PTM) and leverage the generalized feature representation of PTM to learn downstream incremental tasks continually. However, the generalized feature representations of PTM are not adaptive and discriminative for these various incremental classes, which may be out of distribution for the pre-trained dataset. In addition, since the incremental classes cannot be learned at once, the class relationship cannot be constructed optimally, leading to undiscriminating feature representation for understream tasks. Thus, we propose a novel Pre-Trained Model-based Class-Incremental Learning (PTM-CIL) method to explore the potential of PTM and obtain optimal class relationships. Inspired by Neural Collapse theory, we introduce the frozen Equiangular Tight Frame classifier to construct optimal classifier structure for all seen classes, guiding the feature representation adaptation for downstream continual tasks. Specifically, Task-Related Adaptation is proposed to modulate the generalized feature representation to bridge the gap between the pre-trained dataset and various downstream datasets. Then, the Feature Compression Module is introduced to compress various features to the specific classifier weights, constructing the feature transfer pattern and satisfying the characteristic of Neural Collapse. Optimal Structural Alignment is designed to supervise the feature compression process, assisting in achieving optimal class relationships across different tasks. Sufficient experiments on seven datasets prove the effectiveness of our method.

Introduction

The complex streaming data (Guo et al. 2024)(Tao et al. 2021) in real-world situations requires artificial intelligence systems (Tao et al. 2022) to update themselves to continually adapt to novel tasks (Chen et al. 2021)(Dong et al. 2022)(Wei et al. 2022). Since the change of model parameters in the adaptation process, feature representation shifts will appear in the different incremental tasks and lead to the performance drop for previous tasks. In the incremental steps, the feature representation of the same data changes seriously, which is denoted as Catastrophic Forgetting (Robins 1995)(McCloskey and Cohen 1989) and limits artificial intelligence systems applied to real-world situations. Thus, Incremental Learning (Dong et al. 2024)(Kirkpatrick et al. 2017)(Wei et al. 2020) is a learning pattern to continually learn novel tasks without retraining previous tasks, improving model efficiency and performance. Benefitting from the generalization ability of Pre-Trained Models (PTM) trained on large-scale datasets, class-incremental learning methods can leverage the generalized feature representation to adapt to incremental tasks (Chen et al. 2022)(Wei et al. 2024)(Wang et al. 2022a), avoiding feature shifts and obtaining impressive performance compared with traditional class-incremental methods (Wei et al. 2021b)(Yan, Xie, and He 2021).

However, the generalized feature representation of PTM is indiscriminative for these continual downstream tasks, limiting incremental performance. Many PTM-CIL methods (Wang et al. 2022c)(Wang et al. 2022b) adapt the generalized feature representation to streaming out-of-distribution

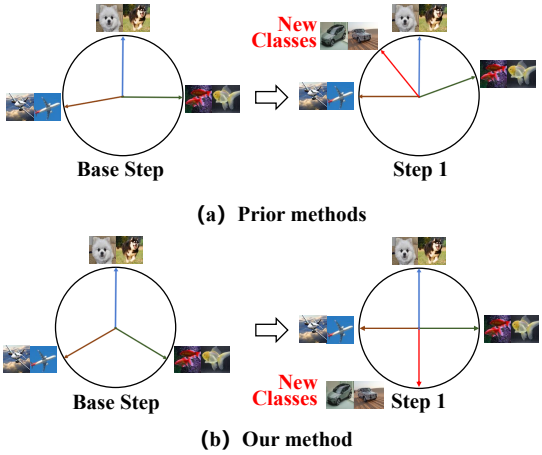


Figure 1: A sketch comparison between prior methods and our method. Since the classes of different steps cannot be learned simultaneously and the objective function may be unsuitable, the classifier structure is not optimal for all seen classes. Our method assigns the optimal classifier structure before the model optimization and freezes the classifier weight, guiding the feature representation adaptation.

*Corresponding Author

tasks (Chen et al. 2023). These methods can be divided into prompt-based methods, representation-based methods, and model-mixture methods. Prompt-based (Wang et al. 2022c)(Wang et al. 2022b) methods leverage the stored prompts to update the PTM lightweight, where limited learnable parameters of prompts constrain the performance of feature transfer. Representation-based methods (Zhou et al. 2023a)(Zhou et al. 2023a) employ the adapter to change the generalized representation, where the continual update of the adapter still faces Catastrophic Forgetting. Model-mixture methods (Zhou et al. 2024)(Wang et al. 2023) aim to merge the different task models to obtain the generalized feature representation for the whole incremental learning. In addition, since the classes of different tasks do not overlap, these methods cannot construct and optimize the optimal class relationships of different tasks, leading to undiscriminating feature representations for downstream incremental classes. Thus, the key to tackling the PTM-CIL problem is to leverage the generalized feature representation to adapt to downstream tasks, and construct discriminative feature representation.

Inspired by Neural Collapse (Papyan, Han, and Donoho 2020), we propose a novel PTM-CIL method, which assigns the optimal classifier structure to guide the feature representation adaptation for downstream tasks. Neural Collapse elucidates an elegant phenomenon about the last-layer features and the classifier, and the following characteristics can concisely summarize it:

- **Variability Collapse** ($\mathcal{NC}_1$): Within-class variability of features collapses to zero.
- **Convergence to Simplex ETF** ($\mathcal{NC}_2$): Class-mean features converge to a simplex Equiangular Tight Frame (ETF), achieving equal lengths, equal pair-wise angles, and maximal distance in the feature space.
- **Self-Duality** ($\mathcal{NC}_3$): Linear classifiers converge to class-mean features, up to a global rescaling.

The requirement for Neural Collapse is consistent with the PTM-CIL problem shown as Figure 1, establishing the robust relationship between the last-layer feature representation and optimal classifier structure. Specifically, benefitting from the feature replay strategy, we introduce the Equiangular Tight Frame classifier to construct the optimal structural classifier for all seen classes. First, we design Task-Related Adaptation to modulate the feature representation for each transformer block of PTM, bridging the domain gap between the pre-trained dataset and various downstream datasets, which are out of distribution for the pre-trained dataset. Then, the Feature Compression Module is introduced to compress various and undiscriminating features to the assigned classifier weights, which satisfies $\mathcal{NC}_1$ and $\mathcal{NC}_3$. Finally, Optimal Structural Alignment is designed to supervise the feature compression process to satisfy $\mathcal{NC}_2$, guiding the feature representation adaptation.

In summary, the contributions are as follows:

- To the best of our knowledge, we are the first to introduce Neural Collapse theory to Pre-Trained Model feature adaptation, providing a novel paradigm to leverage Pre-Trained Model to tackle downstream tasks[1].
- We construct an Optimal Structural Classifier and align the feature representation with the assigned optimal classifier structure, guaranteeing the generalized feature representation can be explored adequately for downstream tasks.
- Sufficient experiments on seven datasets in different class-incremental settings prove the effectiveness of our method.

Related Work

Pre-Trained Model-Based Class-Incremental Learning

Class-incremental learning (Ashok, Joseph, and Balasubramanian 2022)(Yu et al. 2020)(Kirkpatrick et al. 2017)(Wei et al. 2021a) is a learning pattern where the model learns different tasks with non-overlapped classes in a continual manner. In order to address the challenges of class-incremental learning, a significant number of researchers make use of the Pre-Trained Model (PTM) as a means to support and enhance the Class-Incremental Learning process. These methods can be divided into three parts: prompt-based methods (Wang et al. 2022c) (Wang et al. 2022b), representation-based methods (Zhou et al. 2023a) (Zhou et al. 2023a), and model mixture-based methods (Zhou et al. 2023a) (Zhou et al. 2023a). Prompt-based methods utilize prompt-tuning strategies to bring about a minor alteration in the feature representation. In such methods, limited learnable parameters restrict the model's capacity to adapt to long-sequence incremental tasks. Representation-based methods aim to utilize the generalized representation for constructing the classifier. However, they achieve unsatisfactory performance when dealing with out-of-distribution downstream datasets. Model mixture-based methods involve the design of a collection of models during the learning process. By employing model merging and model ensemble strategies, they seek to represent all tasks. Nevertheless, this approach leads to an increase in memory cost due to the growing number of model parameters. In addition, PTM-CIL methods can be developed in many other application fields, such as continual learning with pre-trained large language models (LLM) (Zhang et al. 2024), learning with restricted computational resources (Prabhu et al. 2023) and multi-modal recognition (Zhou et al. 2023b). How to integrate Incremental Learning and various Pre-Trained Models is the future research direction.

Neural Collapse

Neural Collapse (Papyan, Han, and Donoho 2020) explains an elegant phenomenon about the last-layer features and the classifier. Neural Collapse theory offers a mathematically sophisticated and refined characterization of the learned representations or features within deep learning-based classification models. It holds a remarkable degree of independence, being unaffected by various factors such as network architectures, dataset characteristics, and optimization algorithms.

[1]Code is available at https://github.com/wkkkk/NC-PTM.

This theory thus presents a highly generalized and fundamental understanding of the nature of learned features in the context of deep learning classification. Based on Neural Collapse, many methods (Fang et al. 2021; Graf et al. 2021) leverage the last-layer optimization to achieve balanced training. In addition, Neural Collapse theory is also employed to Transfer Learning (Galanti, György, and Hutter 2022, 2021), Incremental Learning (Yang et al. 2023), and architecture designs (Chan et al. 2022), bringing impressive performance improvement.

Unlike other PTM-CIL methods, we construct an optimal classifier structure for all seen classes to guide the feature adaptation for downstream tasks. In addition, the Feature Compress Module and Optimal Structural Alignment are designed to guarantee the feature adaptation satisfies the Neural Collapse theory, obtaining discriminative class relationships for downstream tasks. Sufficient experiments prove the effectiveness of our method for PTM-related adaptation tasks.

Preliminaries

In this section, we introduce the background of PTM-CIL and Neural Collapse, which is the basic theory for our method.

Pre-Trained Model-based Class-Incremental Learning

Class-incremental learning is the learning pattern where a model continually learns new tasks with novel classes. The whole incremental learning process can be divided into T tasks, and the datasets for these tasks are denoted as $D = \{D_1, D_2, \ldots, D_T\}$, where $D_t = \{(x_i, y_i)\}_{i=1}^{n_t}$ is the $t-$th training step with n_t samples. In addition, the class sets of different tasks $C = \{C_i, \ldots, C_T\}$ are not overlapped, where the number of all seen classes is K_t. The goal of CIL is to learn a unified classifier for all seen classes. Following traditional PTM-CIL methods, we denote a pre-trained model is available as the initialization for feature exactor $f(\cdot)$ and the linear classifier as $h(\cdot)$.

Equiangular Tight Frame Classifier

Neural Collapse refers to a phenomenon on balanced data, which reveals a geometric structure formed by the last-layer feature and classifier. A simplex Equiangular Tight Frame (ETF), refers to a matrix composed of K vectors in $\mathcal{R}^d$ and satisfies:

$$E = \sqrt{\frac{K}{K-1}} U \left(I_K - \frac{1}{K} 1_K 1_K^T \right), \tag{1}$$

where $E = [e_1, \ldots, e_K] \in \mathbb{R}^{d \times k}$, $U \in \mathbb{R}^{d \times k}$ allows a rotation and satisfies $U^T U = I_K$, I_K is the identity matrix, and 1_K is an all-ones vectors.

All column vectors in E have the same l_2 norm and any pair has an inner produce of $-\frac{1}{K-1}$:

$$e_{k_1}^T e_{k_2} = \frac{K}{K-1} \delta_{k_1,k_2} - \frac{1}{K-1}, \forall k_1, k_2 \in [1, K], \tag{2}$$

where $\delta_{k_1,k_2} = 1$ when $k_1 = k_2$ and 0 otherwise.

Since Neural Collapse describes an optimal geometric structure of the last-layer feature and classifier, we pre-assign such optimality by fixing a learnable classifier as the structure. We adopt an ETF classifier that initializes a classifier as a simplex ETF and fixes it during training. Based on the total number of seen classes in different incremental steps, we reassign the classifier weights to achieve optimal structural alignment for the current step. We hope ETF can guide the generalized feature representation transfer to the optimal classifier structure.

Methodology

Observing Neural Collapse can construct optimal classifier structure and guide the feature representation transfer, we aim to tackle the PTM-CIL problem with ETF classifier. Hence, we first introduce a Task-Related adapter to bridge the domain gap between the pre-trained dataset and downstream tasks that are out of distribution. Then, we introduce the Feature Compression Module to compress the feature to the assigned classifier weights, decreasing within-class variability and aligning with the specific simplex Equiangular Tight Frame. Finally, Optimal Structural Alignment is proposed to supervise the feature compression process. The frame of our method is shown as Figure 2.

Task-Related Adaptation

Since the domain gap between the pre-trained dataset and the downstream task datasets, the generalized feature representation of the pre-trained model would be undiscriminating for downstream incremental tasks. Thus, we introduce lightweight adapter tuning in the base step to bridge the domain gap. Denote there are L transformer blocks in the Pre-Trained backbone $f(\cdot)$, each containing a self-attention module and an MLP (Multi-Layer Perceptron) layer. To preserve the generalization ability of the Pre-Trained model, we learn an adapter module as a side branch for the MLP to modulate the downstream task information into the generalized feature representation in this transformer block. Specifically, an adapter is a bottleneck module that contains a down-projection layer $W_{down} \in \mathbb{R}^{d \times r}$, a non-linear activation function σ, and an up-projection layer $W_{down} \in \mathbb{R}^{r \times d}$. It adjusts the outputs of the MLP as:

$$m_o = \sigma(m_i W_{down}) W_{up} + MLP(m_i), \tag{3}$$

where m_i and m_o represent the input and output of MLP, respectively. We designate the set of adapters among all L transformer blocks as A, and we refer to the adapted embedding function with adapter A as $f(x; A)$. During the Task-Related Adaptation process, we keep the pre-trained weights fixed and only focus on optimizing the parameters of the adapters. In the remaining learning steps, both the weights of the Pre-Trained Model and those of the adapters are frozen. This is done to maintain the generalized feature representation, which is crucial for representing the unseen classes.

Feature Compression Module

Since the ETF classifier is introduced as the classifier to guide feature representation optimization, we need to guide feature representation transfer to the specific classifier weight,

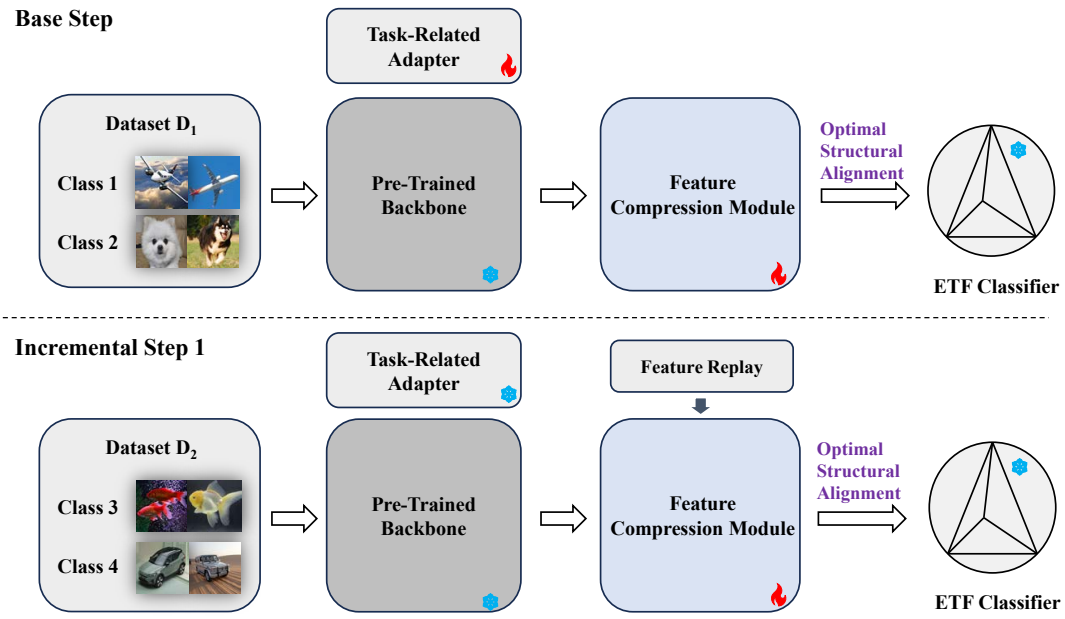


Figure 2: Our method's framework comprises a Pre-Trained Backbone, a Task-Related adapter, a Feature Compression Module, and an ETF classifier. The ETF classifier represents the optimal classifier structure for guiding feature representation adaptation. During all Incremental learning processes, the Pre-Trained Backbone is frozen to maintain the generalization of feature representation. In the base step, the Task-Related Adapter is utilized to adjust the feature representation of the Pre-Trained Backbone to suit downstream tasks. Thanks to the Feature Replay strategy, the Feature Compression Module can compress the features of all seen classes into the assigned classifier weights. Optimal Structural Alignment is incorporated to supervise the feature compression process, ensuring the Neural Collapse characteristic.

following the Neural Collapse characteristics. Thus, we introduce the Feature Compression Module (FCM) to compress the various features. After the feature extractor, we add Feature Compression Module $P(\cdot)$ to transform the generalized feature representation to adjust the pre-assign ETF class prototypes. Feature Compression Module $P(\cdot)$ is constructed by a two-layer MLP and BatchNorm layer. In the Incremental learning steps, $P(\cdot)$ will be fine-tuned to adapt to the pre-assign class prototype continually:

$$z_i = f(x_i), \quad \hat{z}_i = P(z_i), \tag{4}$$

where z_i and $\hat{z}_i$ are the extracted and compressed features respectively.

To alleviate representation overfitting for ETF classifier, we introduce the mixup strategy to increase the feature diversity, which can be denoted as:

$$\tilde{z}_i = \lambda \hat{z}_i + (1-\lambda)\,\hat{z}_j, \quad \tilde{y}_i = \lambda y_i + (1-\lambda)\,y_j, \tag{5}$$

where λ is the random hyper-parameter to balance the samples x_i and x_j. Then, we leverage the cross-entropy to optimize the learnable parameters $P(\cdot)$, denoted as:

$$\mathcal{L}_{ce}(\tilde{z}_i, \tilde{y}_i) = -log\frac{exp(\tilde{z}_i/\tau)}{\sum_{j=1}^{K} exp(\tilde{z}_j/\tau)}, \tag{6}$$

where τ is the temperature parameters.

Optimal Structural Alignment

Since both the Pre-Trained Model and the adapter are frozen in the incremental steps, we need to supervise the feature compression process, guaranteeing the feature transfer satisfies the Nerual Collapse characteristic. In addition, we also want to capture the class relationships among classes from different tasks. Thus, we design two feature replay strategies to introduce the features of previous tasks in the current training step.

- **Full Memory (FM)** : construct the memory M_{FM} to store the features z of all samples;
- **Class Memory(CM)**: construct the memory M_{CM} to collect mean μ and covariance Σ of all classes.

Owing to the utilization of the Pre-Trained Model, which yields well-distributed and generalized representations, each class exhibits a tendency to be single-peaked and can thus be intuitively modeled as a Gaussian $\mathcal{N}(\mu_c, \Sigma_c)$. We employ the re-parameter strategy to replay the features of prior tasks. By means of the features generated from the CM, we are able to acquire a substantial amount of simulation features for each class to achieve Optimal Structural Alignment. In all experiments, the replay number ρ is set to 256.

Benefitting from two feature replay strategies, we can re-assign the weight of the ETF classifier to preserve the optimal classifier structure following the seen class number. To compress the various features, we need to align the feature and

Method	CF B0 Inc5		CUB B0 Inc10		IN-R B0 Inc5		IN-A B0 Inc20		ON B0 Inc10		OB B0 Inc30		VB B0 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$	$\bar{\mathcal{A}}$	$\mathcal{A}_t$
Finetune	38.91	20.17	26.08	13.96	21.61	10.79	24.28	14.51	19.14	8.73	23.61	10.57	34.95	21.25
Finetune Adapter	60.51	49.32	66.84	52.99	47.59	40.28	45.41	41.10	50.22	35.95	62.32	50.53	48.91	45.12
LwF	46.29	41.07	48.97	32.03	39.93	26.47	37.75	26.84	33.01	20.65	47.14	33.95	40.48	27.54
SDC	68.21	63.05	70.62	66.37	52.17	49.20	29.11	26.63	39.04	29.06	60.94	50.28	45.06	22.50
L2P	85.94	79.93	67.05	56.25	66.53	59.22	49.39	41.71	63.78	52.19	73.36	64.69	77.11	77.10
DualPrompt	87.87	81.15	77.47	66.54	63.31	55.22	53.71	41.67	59.27	49.33	73.92	65.52	83.36	81.23
CODA-Prompt	89.11	81.96	84.00	73.37	64.42	55.08	53.54	42.73	66.07	53.29	77.03	68.09	83.90	83.02
SimpleCIL	87.57	81.26	92.20	86.73	62.58	54.55	59.77	48.91	65.45	53.59	79.34	73.15	85.99	84.38
ADAM+Finetune	87.67	81.27	91.82	86.39	70.51	62.42	61.01	49.57	61.41	48.34	73.02	65.03	87.47	80.44
ADAM+VPT-S	90.43	84.57	92.02	86.51	66.63	58.32	58.39	47.20	64.54	52.53	79.63	73.68	87.15	85.36
ADAM+VPT-D	88.46	82.17	91.02	84.99	68.79	60.48	58.48	48.52	67.83	54.65	81.05	74.47	86.59	83.06
ADAM + SSF	87.78	81.98	91.72	86.13	68.94	60.60	61.30	50.03	69.15	56.64	80.53	74.00	85.66	81.92
ADAM+Adapter	90.65	85.15	92.21	86.73	72.35	64.33	60.47	49.37	67.18	55.24	80.75	74.37	85.95	84.35
EASE	91.51	85.80	92.23	86.81	78.31	70.58	65.34	55.04	70.84	57.86	81.11	74.85	93.61	**93.55**
RanPAC	93.51	89.30	93.13	89.40	75.74	68.75	64.16	52.86	71.67	60.08	85.95	79.55	92.56	91.83
Ours/with FM	**94.41**	**90.49**	**93.52**	**89.14**	**81.64**	**76.22**	**67.43**	**59.51**	**75.23**	**63.94**	**87.23**	**81.67**	95.42	90.60
Ours/with CM	94.01	89.71	93.51	88.51	79.75	73.43	65.63	55.63	75.02	63.64	86.53	80.22	**96.02**	92.09

Table 1: Average and last performance comparison on seven datasets with ViT-B/16-IN21K as the backbone. 'CF' stands for 'CIFAR100,' 'IN-R/A' stands for 'ImageNet-R/A,' 'ON' stands for 'ObjectNet', 'OB' stands for 'OmniBenchmark' and 'VB' stands for 'VTAB'. The best performance is shown in bold. The second performance is shown with underline.

the assigned corresponding classifier weight e_i. Since the capacity gap between random re-assigned classifier weight and the feature representation is large, the optimal classifier alignment becomes challenging. The Batchnorm layer within the Feature Compression Module is capable of minimizing collapse (Tian, Krishnan, and Isola 2019). It causes far fewer singular values to shrink towards zero, thereby fulfilling the requirement of $\mathcal{NC}_1$.

Consequently, we investigate the application of a soft maximum loss with the aim of achieving Optimal Structural Alignment. In this way, the loss, denoted as Optimal Classifier Loss, can be adjusted to compensate for poorly aligned features. We use the simple LogSum function to achieve alignment between feature representation and the optimal classifier weight, denoted as:

$$\mathcal{L}_{ocl}\left(Z_i, Z_k; W_p\right) = log \sum_i \left|z_i W_p - e_k\right|_i^{\alpha}, \tag{7}$$

where α is a smoothing factor, e_k is the classifier weight of corresponding class and W_p is the weight of feature Compression module P.

The final objective function for the whole incremental model is denoted as:

$$\mathcal{L} = \xi * \mathcal{L}_{ce} + \mathcal{L}_{ocl}, \tag{8}$$

where ξ is the hyper-parameter and denoted as 10.

Experiments

This section tests our method on seven PTM-CIL benchmark datasets and compares it with state-of-the-art methods. We also perform qualitative and quantitative ablation studies to validate the effectiveness of our method.

Implementation Details

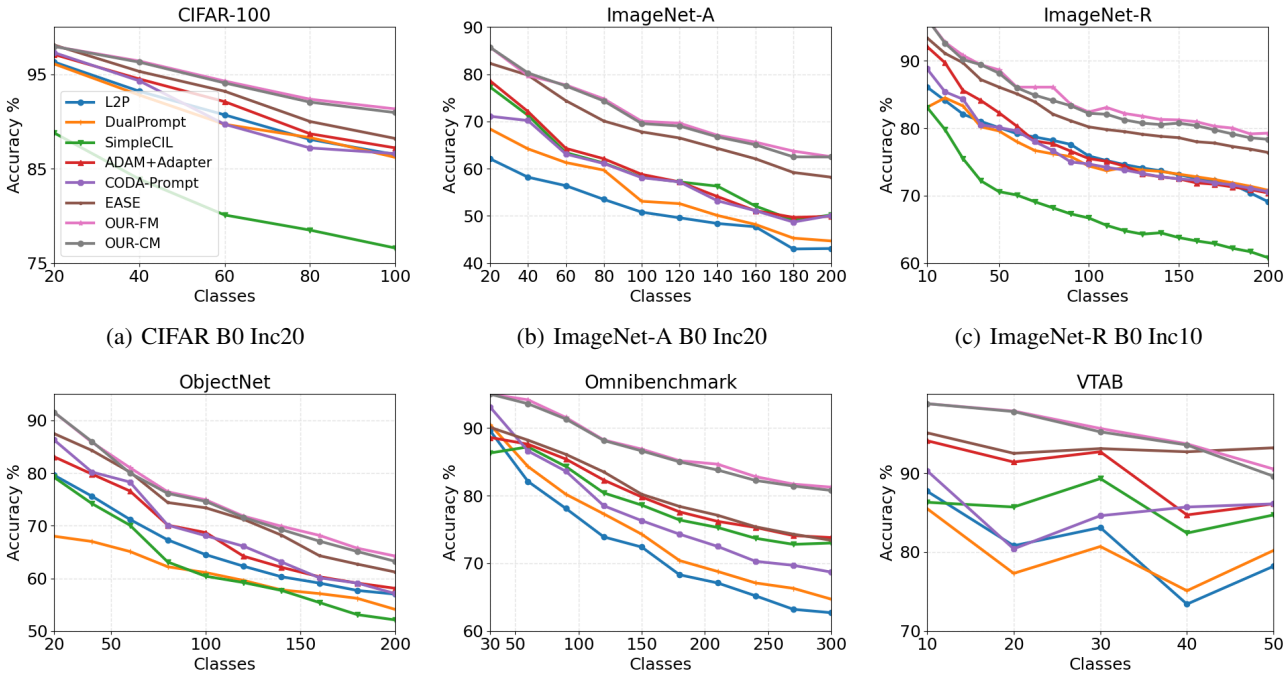
Dataset: Following the evaluation setting of EASE (Zhou et al. 2024), we select CIFAR100 (Krizhevsky, Hinton et al. 2009), CUB (Wah et al. 2011), ImageNet-R (Hendrycks et al. 2021a), ImageNet-A (Hendrycks et al. 2021b), ObjectNet (Barbu et al. 2019), Omnibenchmark (Zhang et al. 2022), and VTAB (Zhai et al. 2019) datasets to evaluate the performance. These datasets contain typical CIL benchmarks and out-of-distribution datasets, which have a large domain gap with ImageNet. There are 50 classes in VTAB, 100 classes in CIFAR100, 200 classes in CUB, ImageNet-R, ImageNet-A, ObjectNet, and 300 classes in OmniBenchmark.

Dataset split: Following EASE (Zhou et al. 2024), we denote the class split as 'B-m Inc-n', where m indicates the number of classes in the first stage and n represents the number of classes in every incremental learning step. For a fair comparison, we randomly shuffle class orders with random seed 1993 before the data split and keep the training and testing set as EASE (Zhou et al. 2024).

Comparison methods: We select state-of-the-art PTM-CIL methods for comparison, L2P (Wang et al. 2022c), DualPrompt (Wang et al. 2022b), CODA-Prompt (Smith et al. 2023), SimpleCIL (Zhou et al. 2023a), ADAM (Zhou et al. 2023a) and EASE (Zhou et al. 2024). In addition, we also compare our method with typical CIL methods with PTM, LwF (Li and Hoiem 2017), SDC (Yu et al. 2020), RanPAC (McDonnell et al. 2024).

Training details: We run experiments on NVIDIA RTX A6000 and report the results of other methods as EASE (Zhou et al. 2024). Following the previous setting, we consider two representative models, ViT-B/16-IN21K pre-trained on ImageNet21K and ViT-B/16-IN1K pre-trained on ImageNet1K. In our method, we train the model using an SGD optimizer for 35 epochs in the first learning step and an Adam optimizer for 100 epochs in other incremental steps, with a batch size of 256 and a 0.01 learning rate. The Pytorch deep learning structure supports all our experiments.

Evaluation metric: Following the benchmark (Zhai et al. 2019), we use $\mathcal{A}_t$ to evaluate the model's accuracy after



(a) CIFAR B0 Inc20 (b) ImageNet-A B0 Inc20 (c) ImageNet-R B0 Inc10

(d) ObjectNet B0 Inc20 (e) Omnibenchmark B0 Inc30 (f) VTAB B0 Inc10

Figure 3: Performance curve of different methods under different settings. All methods are initialized with ViT-B/16-IN1K.

FT	TRA	ETF	FC	OSA	CF B0 Inc5	CUB B0 Inc10	IN-R B0 Inc5	IN-A B0 Inc20	ON B0 Inc10	OB B0 Inc30	VB B0 Inc10
✓	×	×	×	×	38.91	26.08	21.61	24.28	19.14	23.61	34.95
×	×	×	×	×	90.22	93.42	74.43	61.21	69.40	85.81	94.81
×	✓	×	×	×	90.62	93.42	74.43	61.21	69.40	85.81	94.81
×	✓	✓	×	×	91.32	**93.63**	75.11	62.53	69.84	85.93	95.21
×	✓	✓	✓	×	93.43	93.42	77.73	64.64	73.53	86.61	94.93
×	✓	×	✓	✓	91.26	93.57	75.43	62.55	70.18	86.23	95.22
×	✓	✓	✓	✓	**94.41**	93.52	**81.64**	**67.43**	**75.23**	**87.23**	**95.42**

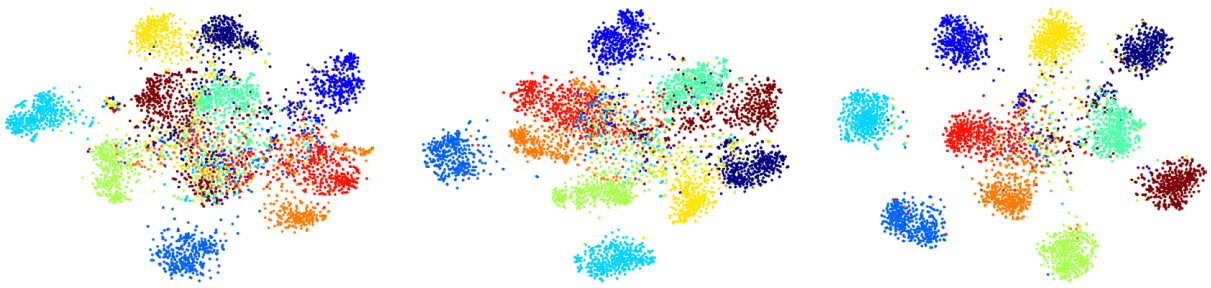
Table 2: The ablation study of different modules on seven datasets.

the b-th step. In addition, we introduce average incremental accuracy $\bar{\mathcal{A}} = \frac{1}{t}\sum_{i=1}^{t}\mathcal{A}_t$ to evaluate the whole incremental learning process. Through the two evaluation metrics, we can roundly evaluate the incremental performance of all PTM-CIL methods.

Benchmark Comparison

In this section, we compare our method with other state-of-the-art methods on seven benchmark datasets and different backbone weights. Table 1 presents the comparison of different methods with the ViT-B/16-IN21K pre-trained model. We can note our method with FM obtains the best average incremental accuracy performance $\bar{\mathcal{A}}$ on seven popular benchmarks, except the $\mathcal{A}_t$ on VTAB dataset. Our method with FM obtains 0.90%, 0.39%, 3.33%, 2.09%, 2.68%, 1.28% and 1.81% average incremental accuracy performance improvement on CIFAR100, CUB, ImageNet-R, ImageNet-A, OmniBenchmark, ObjectNet and VTAB datasets, respectively. Our method with CM also obtains impressive performance with 0.50%, 1.27%, 1.44%, 0.29%, 2.47%, 0.58% and 2.41% performance improvement on CIFAR100, CUB, ImageNet-R, ImageNet-A, OmniBenchmark, ObjectNet and VTAB datasets, respectively. Our method with FM needs memory to store the extracted features of all samples, while our method with CM is without memory PTM-CIL methods and achieves the second-best performance. In addition, our methods also have the best performance on $\mathcal{A}_t$ evaluation metric, except the VTAB dataset, where EASE performance fluctuates in the learning steps and is unreasonable.

We also report the incremental performance trend of different methods in Figure 3 with Vit-B/16-IN1K. We can note that our methods still obtain the best performance on CIFAR100, ImageNet-A, ImageNet-R, ObjectNet, and Omnibenchmark datasets with impressive improvements in each learning step. In addition, we obtain the second best performance on the VTAB dataset, which is a small dataset with limited classes. The experiments indicate that our method can be applied to different Pre-Trained Models and experiment settings, further proving its effectiveness. Furthermore, our method obtains the best performance in the base step, surpass-



(a) ViT-B/16-IN21K pre-trained model (b) PTM with Task-Related adapter (c) Ours-FM

Figure 4: t-SNE visualizations of different modules, presenting the effectiveness of different modules.

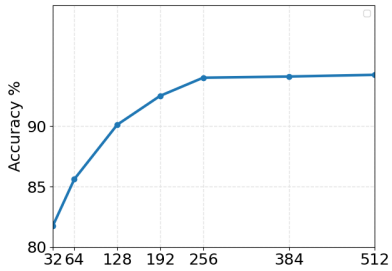


Figure 5: The parameter sensitivity analysis of the number of replayed feature.

ing other methods to a large extent. This phenomenon proves that our method has the potential to be a novel paradigm to leverage the Pre-Trained Model to tackle downstream tasks.

Ablation Study

To evaluate the performance of our different modules further, we introduce visual qualitative results to prove the feature representation transformation performance, shown in Figure 4. The samples are the first ten classes of the CIFAR100 dataset, where the pre-trained model is ViT-B/16-IN21K and the setting is B0 Inc5. We can note that different classes in t-SNE visualization results of the Pre-Trained Model are not separated clearly, proving the generalized feature representation of the Pre-Trained Model is not suitable and discriminative for downstream tasks. In addition, by gradually applying Task-Related adapters and Feature Compression Modules, different classes are clearly separated. Obviously, Task-Related Adapter can bridge the domain gap between pre-trained dataset and various downstream datasets. Based on the Task-Related adapter, our method is more compact and discriminative, proving that the Feature Compression Module can achieve Neural Collapse and feature representation adaptation.

To evaluate the different modules of our method quantitatively, we conduct several ablation experiments shown in Table 2, where pre-trained model ViT-B/16-IN21K is continually learning on seven different datasets. We denote Finutune Basebone, Task-Related Adaptation, ETF classifier, Feature Compression Module, and Optimal Structural Alignment as 'FT', TRA', 'ETF', 'FCM', and 'OSA'. The base model is the pre-trained model ViT-B/16-IN21K, which finetunes in a continual learning process. We observe that our method obtains better performance when applied gradually to different modules, which proves the effectiveness of different modules. We present the experimental results of our method with fixed random classifier weights. Without the guidance of the ETF, the performance of our method noticeably decreases, demonstrating that non-optimal structural alignment will lead to poor feature representation. Optimal Structural Alignment is essential for aligning the feature representation between pre-trained datasets and downstream datasets.

In addition, we also discuss the number of replayed features of our-CM on CIFAR100 dataset in B0 Inc5 setting shown as Figuer 5, where the number is set to 32, 64, 128, 192, 256, 384 and 512. The performance of our method improves with the increase of the number. When the number is bigger than 256, the performance of our method grows slowly, approaching the upper-performance limit. Hence, we set the number of replayed feature as 256 to obtain the trade-off between the performance and computing cost.

Conclusion

In this paper, we propose a novel Class-Incremental Pre-Trained method, which constructs an optimal classifier structure to guide the generalized feature adaptation. Task-Related Adaptation is introduced to bridge the domain gap between pre-trained datasets and downstream datasets. In addition, the Feature Compression Module is designed to compress various feature representations to corresponding class prototypes, satisfying $\mathcal{NC}_1$, $\mathcal{NC}_2$ and $\mathcal{NC}_3$ characteristic. Optimal Structural Alignment is proposed to supervise the Feature Compression process. Sufficient comparative experiments and ablation experiments prove the effectiveness of our method. In addition, our method has the potential to be a novel paradigm to leverage PTM to tackle downstream tasks.

Acknowledgments

Our work is supported in part by National Natural Science Foundation of China (62132016, 62406238), and Natural Science Basic Research Program of Shaanxi (2020JC-23).

References

Ashok, A.; Joseph, K.; and Balasubramanian, V. N. 2022. Class-Incremental Learning with Cross-Space Clustering and Controlled Transfer. In *European Conference on Computer Vision*, 105–122. Springer.

Barbu, A.; Mayo, D.; Alverio, J.; Luo, W.; Wang, C.; Gutfreund, D.; Tenenbaum, J.; and Katz, B. 2019. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32.

Chan, K. H. R.; Yu, Y.; You, C.; Qi, H.; Wright, J.; and Ma, Y. 2022. Redunet: A white-box deep network from the principle of maximizing rate reduction. *Journal of machine learning research*, 23(114): 1–103.

Chen, Z.; Luo, Y.; Qiu, R.; Wang, S.; Huang, Z.; Li, J.; and Zhang, Z. 2021. Semantics Disentangling for Generalized Zero-Shot Learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Chen, Z.; Luo, Y.; Wang, S.; Li, J.; and Huang, Z. 2022. GSMFlow: Generation Shifts Mitigating Flow for Generalized Zero-Shot Learning. *IEEE Transactions on Multimedia*.

Chen, Z.; Zhang, P.; Li, J.; Wang, S.; and Huang, Z. 2023. Zero-Shot Learning by Harnessing Adversarial Samples. In *Proceedings of the 31th ACM International Conference on Multimedia 2023*.

Dong, J.; Li, H.; Cong, Y.; Sun, G.; Zhang, Y.; and Van Gool, L. 2024. No One Left Behind: Real-World Federated Class-Incremental Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4): 2054–2070.

Dong, J.; Wang, L.; Fang, Z.; Sun, G.; Xu, S.; Wang, X.; and Zhu, Q. 2022. Federated Class-Incremental Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fang, C.; He, H.; Long, Q.; and Su, W. J. 2021. Exploring deep neural networks via layer-peeled model: Minority collapse in imbalanced training. *Proceedings of the National Academy of Sciences*, 118(43): e2103091118.

Galanti, T.; György, A.; and Hutter, M. 2021. On the role of neural collapse in transfer learning. *arXiv preprint arXiv:2112.15121*.

Galanti, T.; György, A.; and Hutter, M. 2022. Generalization bounds for transfer learning with pretrained classifiers. *arXiv preprint arXiv:2212.12532*.

Graf, F.; Hofer, C.; Niethammer, M.; and Kwitt, R. 2021. Dissecting supervised contrastive learning. In *International Conference on Machine Learning*, 3821–3830. PMLR.

Guo, D.; Li, K.; Hu, B.; Zhang, Y.; and Wang, M. 2024. Benchmarking Micro-action Recognition: Dataset, Method, and Application. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7): 6238–6252.

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.

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.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and van den Hengel, A. 2024. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Prabhu, A.; Al Kader Hammoud, H. A.; Dokania, P. K.; Torr, P. H.; Lim, S.-N.; Ghanem, B.; and Bibi, A. 2023. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3698–3707.

Robins, A. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146.

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.

Tao, R.; Li, H.; Wang, T.; Wei, Y.; Ding, Y.; Jin, B.; Zhi, H.; Liu, X.; and Liu, A. 2022. Exploring endogenous shift for cross-domain detection: A large-scale benchmark and perturbation suppression network. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 21157–21167. IEEE.

Tao, R.; Wei, Y.; Jiang, X.; Li, H.; Qin, H.; Wang, J.; Ma, Y.; Zhang, L.; and Liu, X. 2021. Towards real-world X-ray security inspection: A high-quality benchmark and lateral inhibition module for prohibited items detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10923–10932.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*.

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

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, 398–414. Springer.

Wang, X.; Chen, T.; Ge, Q.; Xia, H.; Bao, R.; Zheng, R.; Zhang, Q.; Gui, T.; and Huang, X. 2023. Orthogonal subspace learning for language model continual learning. *arXiv preprint arXiv:2310.14152*.

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

Wei, K.; Chen, D.; Li, Y.; Yang, X.; Deng, C.; and Tao, D. 2022. Incremental embedding learning with disentangled representation translation. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3): 3821–3833.

Wei, K.; Deng, C.; Yang, X.; and Li, M. 2021a. Incremental embedding learning via zero-shot translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10254–10262.

Wei, K.; Deng, C.; Yang, X.; and Tao, D. 2021b. Incremental zero-shot learning. *IEEE Transactions on Cybernetics*, 52(12): 13788–13799.

Wei, K.; Deng, C.; Yang, X.; et al. 2020. Lifelong Zero-Shot Learning. In *IJCAI*, 551–557.

Wei, K.; Yang, X.; Xu, Z.; and Deng, C. 2024. Class-Incremental Unsupervised Domain Adaptation via Pseudo-Label Distillation. *IEEE Transactions on Image Processing*.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3014–3023.

Yang, Y.; Yuan, H.; Li, X.; Lin, Z.; Torr, P.; and Tao, D. 2023. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6982–6991.

Zhai, X.; Puigcerver, J.; Kolesnikov, A.; Ruyssen, P.; Riquelme, C.; Lucic, M.; Djolonga, J.; Pinto, A. S.; Neumann, M.; Dosovitskiy, A.; et al. 2019. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*.

Zhang, N.; Yao, Y.; Tian, B.; Wang, P.; Deng, S.; Wang, M.; Xi, Z.; Mao, S.; Zhang, J.; Ni, Y.; et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zhang, Y.; Yin, Z.; Shao, J.; and Liu, Z. 2022. Benchmarking omni-vision representation through the lens of visual realms. In *European Conference on Computer Vision*, 594–611. Springer.

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. *arXiv preprint arXiv:2403.12030*.

Zhou, D.-W.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2023a. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*.

Zhou, D.-W.; Zhang, Y.; Ning, J.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2023b. Learning without forgetting for vision-language models. *arXiv preprint arXiv:2305.19270*.