Interactive Continual Learning: Fast and Slow Thinking

Supplementary Material

6. Algorithms of ICL

We formalize the algorithms for the training and inference stages of ICL, as shown in Algorithm 1 and 2. Here, we set the detection threshold α as the upper 20th percentile of the standard normal distribution, which is -0.842.

Algorithm 1 Training Stage of ICL

```
Require: The parameters of the image feature extractor of the pre-trained ViT $\boldsymbol{\varphi}$, the memory buffer $\mathcal{M}$ with size $|\mathcal{M}|$, the parameters of the query memory $\boldsymbol{\theta}$, the CL training dataset $\mathcal{D}_t, 1 \leq t \leq |T|$ with $n^t$ batches
Initialize value memory parameters $\mathcal{Z} = \emptyset$ and memory buffer $\mathcal{M} = \emptyset$.
for $t = 1 \leftarrow |T|$ do
    for $i = 1 \leftarrow n^t$ do
        if $t > 1$ then
            Randomly sample a batch $\tilde{\mathcal{B}}_i^t$ from $\mathcal{M}$
            $\mathcal{B}_i^t = \mathcal{B}_i^t \cup \tilde{\mathcal{B}}_i^t$
        end if
        if $\exists\,(\boldsymbol{x}^t, y^t) \in \mathcal{B}_i^t$ s.t. no $\boldsymbol{z} \in \mathcal{Z}$ matches $y^t$ then
            Add $\boldsymbol{z}^{y^t} = \text{Concat}[\boldsymbol{z}_t, \boldsymbol{z}_{y^t}]$ into $\mathcal{Z}$.
        end if
        E-step: Update value memory parameters $\mathcal{Z}$ on $\mathcal{L}(\mathcal{B}_i^t)$
        M-step: Update query memory parameters $\boldsymbol{\theta}$ on $\mathcal{L}(\mathcal{B}_i^t)$
        Update memory buffer $\mathcal{M}$
    end for
    Freeze the memory parameters of classes in task $t$
end for
```

Algorithm 2 Inference Stage of ICL

```
Require: The image feature extractor in pre-trained ViT $f_{\boldsymbol{\varphi}}$, the trained query and value memory $f_{\boldsymbol{\theta}}$, $\mathcal{Z}$, the test dataset $\mathcal{D}$ with $n$ batches.
for $i = 1 \leftarrow n$ do
    for $j \leftarrow |\mathcal{B}_i|$ do
        $\hat{y}_j = \arg\max_y p^{\mathcal{B}_i^t}_{\boldsymbol{\theta},\boldsymbol{\varphi}}(\boldsymbol{z}^{y_i}|\boldsymbol{x})$
    end for
    $\tilde{\boldsymbol{X}}_i = \{(\tilde{\boldsymbol{x}}, \tilde{y}) \in \mathcal{B}_i \mid (\nu - \bar{\nu}_{\mathcal{B}_i})/\sigma_{\mathcal{B}_i} < \alpha\}$
    if $\tilde{\boldsymbol{X}}_i = \emptyset$ then
        Return the prediction results of System 1
    else
        Using System 2, perform inference on $\boldsymbol{x} \in \tilde{\boldsymbol{X}}_i$ by combining the top-$K$ output of System 1. Retrieve the result of the exact answer and combine it with the remaining predictions from System 1 before returning.
    end if
end for
```

7. Datasets Settings

CIFAR-10 comprises 10 classes, each with 50,000 training and 10,000 test color images. CIFAR-100 includes 100 classes, offering 500 training and 100 testing images per class. ImageNet-R, an extension of the ImageNet dataset, possess 200 classes. It contains a total of 30,000 images, of which 20% were allocated as the test set.

CIFAR-10 was divided into five tasks, two classes allocated to each task. CIFAR-100 was divided into ten tasks, each task with ten classes. Similarly, ImageNet-R was organized into ten tasks, with each task containing 20 classes. Input images were resized to 224 × 224 and normalized to the range $[0, 1]$. ICL was compared against both representative baselines and state-of-the-art methods across diverse buffer sizes and datasets.

8. Implementation Details

To ensure a fair comparison between methods, we carried out uniform resizing of the images to dimensions of 224 × 224 and applied image normalization. Following the settings of [3, 4, 32, 44], we adopted 10 batch size and 1 epoch for all methods during training, utilizing cross-entropy as the classification loss. For the L2P[46], DualPrompt[45] approaches, we followed the implementation details of the original paper and employed ViT as the backbone network, while ResNet18 served as the backbone network for the remaining methods. We meticulously reproduced the outcomes by adhering to the original implementation and settings. We have set up separate Adam optimizers with a constant learning rate of $1e-4$ for the query and value memory parameters.

9. Inference with System 1

We conducted a comparison by directly applying rehearsal-based fine-tuning with a same-sized buffer, using only the pretrained ViT with a trainable classification head on each dataset. The results, as shown in Tab. 3, were significantly lower than those obtained using only System 1. This stark contrast serves as strong evidence that both ViT and MiniGPT-4 have not undergone pretraining on the three datasets and highlights the effectiveness of our proposed method.

Memory Buffer	Method	**CIFAR10**		**CIFAR100**		**ImageNet-R**	
		Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
200	ViT Finetune	33.15	96.00	32.60	91.50	20.88	64.45
	ICL w/o System2	94.60	99.43	77.34	94.81	49.87	68.62
500/600	ViT Finetune	62.65	97.15	45.30	92.80	33.26	75.50
	ICL w/o System2	95.54	99.52	80.67	95.24	54.65	76.02

Table 3. Comparison of incremental accuracy (%). Vit Finetune represents the basic rehearsal method using Vit as the backbone.

10. Inference with System 2

In order to validate the mutually beneficial interaction between systems, we conduct experiments using the pretrained MiniGPT4 [53] to perform inference on the test sets of CIFAR-10, CIFAR-100, and ImageNet-R. MiniGPT4 loads the official 7B pre-trained parameters, and the prompt used by MiniGPT4 is the same as System2. Since System 1 does not provide a topk option, we provided all categories to MiniGPT4, allowing it to select a category for image classification based on the image description. Tab. 4 presents the accuracy of reasoning, error rate, and proportion of no exact response (i.e. there is not only one class in the given classes is returned or no response).

Dataset	Accuracy	Error	No Response	Total
CIFAR-10	9.53	15.04	75.43	10000
CIFAR-100	2.45	14.53	83.02	10000
ImageNet-R	2.67	10.33	87.00	6000

Table 4. MiniGPT4's inference accuracy, error rate, and proportion of no exact response on the CIFAR-10, CIFAR-100, and ImageNet-R test sets. The number of responses for each test set are reported in the last column.

The results presented in the table indicate that over 75% of the images fed in MiniGPT4, when applied to the CIFAR-10 dataset, fail to return a specific class to which the image belongs. And when faced with the CIFAR-100 and ImageNet-R datasets, MiniGPT4 with prompt that includes a larger number of classes, encounters increased difficulty in making accurate selections. Among the images that were returned with specific class information, over two-thirds were misclassified. These experimental results demonstrate that relying solely on MiniGPT4 for image classification tasks yields poor performance. Nevertheless, when System 1 offers the top-K option, incorporating MiniGPT4 as System2 enhances the image classification task and improves the final accuracy. This finding demonstrates that the interactive inference between System1 and System2 enables mutual promotion and improvement.

The limitations of MiniGPT-4 restricted the performance enhancement of System 2. To address these concerns, we chose more advanced MLLMs as System 2. As depicted in Tab. 1, there was a notable 3-4% improvement, especially on the challenging ImageNet-R dataset.