Memory-Efficient Continual Learning with CLIP Models

Ryan C. King
Department of Computer Science
Texas A&M University
College Station, TX 77843
kingrc15@tamu.edu

Gang Li
Department of Computer Science
Texas A&M University
College Station, TX 77843
gang-li@tamu.edu

Bobak J. Mortazavi
Texas A&M University
College Station, TX 77843
bobakm@tamu.edu

Tianbao Yang
Texas A&M University
College Station, TX 77843
tianbao-yang@tamu.edu

Abstract

Contrastive Language-Image Pretraining (CLIP) models excel at understanding image-text relationships but struggle with adapting to new data without forgetting prior knowledge. To address this, models are typically fine-tuned using both new task data and a memory buffer of past tasks. However, CLIP's contrastive loss suffers when the memory buffer is small, leading to performance degradation on previous tasks. We propose a memory-efficient, distributionally robust method that dynamically reweights losses per class during training. Our approach, tested on class incremental settings (CIFAR-100, ImageNet1K) and a domain incremental setting (DomainNet) adapts CLIP models quickly while minimizing catastrophic forgetting, even with minimal memory usage.

1 Introduction

In dynamic environments, machine learning systems must continuously learn and adapt to new information. Continual learning (CL) allows models to acquire new skills while retaining knowledge from past tasks, which is essential as data evolves over time. While there is extensive research on addressing the challenge of catastrophic forgetting in traditional supervised models, most methods—such as parameter regularization, knowledge distillation, and dynamic architectures—have not been applied to models like CLIP, which excel at understanding image-text relationships.

CLIP models need CL to adapt to real-world data streams. However, CL with CLIP models is still under-explored. Recent works, such as those by [15] and [6], have shown promising results in mitigating forgetting through rehearsal-based approaches and memory buffers. Despite these advances, a key question remains: how can we efficiently leverage memory buffers in CLIP's CL to balance new and old task performance?

Our study addresses this by proposing two approaches: one treats old and new data equally during fine-tuning, while the other dynamically reweights class losses using Distributionally Robust Optimization (DRO). We evaluate these methods in class-incremental and domain-incremental settings, demonstrating improved retention of past knowledge and efficient adaptation to new tasks with minimal memory requirements.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related Works

Continual Learning There are many approaches to address catastrophic forgetting. One approach is through replay methods, which update models with a combination of new task data and examples from previous tasks stored in a memory buffer. While effective, maintaining these buffers increases computational costs and poses challenges under privacy constraints. Generative replay methods attempt to mitigate this by synthesizing prior task data, though their success depends on the quality of the generated examples. Dynamic model expansion is another technique, where architectures are extended after each task. For example, [17] trains a new model per task, which avoids forgetting but is impractical for large models. [16] reduces memory usage by retaining the previous model for distillation, while [19] only expands specific network blocks. Knowledge distillation (KD) is another approach, transferring knowledge from previous tasks to a target model. Methods like [14, 1, 5] utilize predictions from prior models as pseudo-labels for training on current tasks.

Contrastive Pretraining In the realm of self-supervised learning (SSL), contrastive learning has emerged as a key technique. Unimodal methods like [2, 18] create positive pairs from augmented input data, while bimodal methods such as CLIP [13, 18] treat different modalities (e.g., image and caption) as positive pairs. Unimodal methods like [1] adapt pretrained models using memory banks, while [5] uses SSL objectives for cross-task knowledge transfer. Bimodal methods, like CLIP, have shown strong performance in both zero-shot and fine-tuning settings [15, 7], and recent studies explore their potential in continual learning contexts [15, 6, 1, 4, 10].

3 Methods

Notation. Let E_1, E_2 denote the image encoder and text encoder respectively, parameterized by $\mathbf{w}$. A datasets $\mathcal{D}$ consists of $\mathcal{T}$ tasks where each task contains a subset of the dataset $\mathcal{D}^t$ where $\mathcal{D}^t \cap D^{t'} = \emptyset, \forall t' \neq t$ and $N_t = |\mathcal{D}^t|$.

Class Incremental Learning In class incremental learning, new tasks come with new classes. The ultimate goal is to continually build a classification model for all classes. In other words, the model should not only acquire the knowledge from the current task $\mathcal{D}^t$ but also preserve the knowledge from former tasks. After each task, the trained model is evaluated over $\mathcal{D}^t_{test1} = \{(\mathbf{x}_i, y_i)\}$, $y_i \in \mathcal{Y}_t = Y_1 \cup ...Y_t$ and all the previously measured task $\mathcal{D}^b_{test} = \{(\mathbf{x}_i, y_i)\}$, for $b = 1, ..., t-1$

Domain Incremental Learning In domain incremental learning, the goal is to update a model given some new data from another domain with the same set of labels. After being trained on tasks t, the model is evaluated on $\mathcal{D}^b_{test} = \{(\mathbf{x}_i, y_i)\}$, $y_i \in Y_t$.

3.1 Bimodal Contrastive Continual Learning

CLIP models, as shown in recent studies [13, 18], possess the ability to process both image and text inputs by learning a joint embedding across modalities. Their impressive performance on image tasks without task-specific training is largely due to the contrastive learning objectives used during training. Moreover, encoding labels with the text encoder further boosts classification performance [7].

Building on CLIP models' ability to jointly encode labels and images improves resistance to catastrophic forgetting and enhances adaptability to new data. To extend CLIP for CL, we propose a bimodal contrastive learning objective tailored to the class-incremental setting. The contrastive objective during each task is defined as:

$$
\begin{aligned}
\mathcal{L}_{contrastive} = &- \frac{1}{N_t + |\mathcal{M}_t|} \sum_{\mathbf{x}_i \in \mathcal{D}^t \cup \mathcal{M}_t} \log \frac{\exp((E_1(\mathbf{w}_t, \mathbf{x}_i)^T E_2(\mathbf{w}_t, y_i))/\tau)}{\sum_{y_j \in \mathcal{D}^t \cup \mathcal{M}_t} \exp((E_1(\mathbf{w}_t, \mathbf{x}_i)^T E_2(\mathbf{w}_t, y_j))/\tau)} \\
&- \frac{1}{N_t + |\mathcal{M}_t|} \sum_{y_i \in \mathcal{D}^t \cup \mathcal{M}_t} \log \frac{\exp((E_2(\mathbf{w}_t, y_i)^T E_1(\mathbf{w}_t, \mathbf{x}_i))/\tau)}{\sum_{\mathbf{x}_j \in \mathcal{D}^t \cup \mathcal{M}_t} \exp((E_2(\mathbf{w}_t, y_i)^T E_1(\mathbf{w}_t, \mathbf{x}_j))/\tau)},
\end{aligned} \tag{1}
$$

Here, $\mathcal{D}_t$ represents data for the current task, and $\mathcal{M}_t$ is a memory bank storing past task samples. Labels y_i are encoded as text using E_2. To address computational constraints, we maintain a constant memory size, keeping an equal number of randomly sampled examples per class.

A key challenge in optimizing this objective arises from the summation over the entire dataset for contrastive terms:

$$g_I(\mathbf{w}, \mathbf{x}_i, \mathcal{D}^t \cup \mathcal{M}^t) = \sum_{y_j \in \mathcal{D}^t \cup \mathcal{M}_t} \exp((E_1(\mathbf{w}_t, \mathbf{x}_i)^T E_2(\mathbf{w}_t, y_j))/\tau) \tag{2}$$

$$g_T(\mathbf{w}, y_i, \mathcal{D}^t \cup \mathcal{M}^t) = \sum_{\mathbf{x}_j \in \mathcal{D}^t \cup \mathcal{M}_t} \exp((E_2(\mathbf{w}_t, y_i)^T E_1(\mathbf{w}_t, \mathbf{x}_j))/\tau) \tag{3}$$

To reduce the computational cost, we use moving average estimators u_i^I and u_i^T for g_I and g_T. The gradient estimator is then computed using a mini-batch $\mathcal{B}$ as:

$$\mathbf{m} = -\frac{1}{|\mathcal{B}|} \sum_{\mathbf{x}_i \in \mathcal{B}} \nabla(E_1(\mathbf{w}_t, \mathbf{x}_i)^T E_2(\mathbf{w}_t, y_i)) + \frac{\tau}{2|\mathcal{B}|u_i^I} \nabla g_I(\mathbf{w}, \mathbf{x}_i, \mathcal{B}) + \frac{\tau}{2|\mathcal{B}|u_i^T} \nabla g_T(\mathbf{w}, y_i, \mathcal{B}) \tag{4}$$

This method, which maintains a moving average across tasks, allows information from prior tasks to carry forward, enhancing CL. We call this approach the Global Contrastive Loss (GCL).

3.2 Group Distributionally Robust Optimization

Due to the fixed memory size, after completing each task, we reduce the number of examples per class to accommodate new ones. This leads to an imbalance between previous and current task data distributions. While our Global Contrastive Loss (GCL) is effective for standard classification tasks, it doesn't handle these imbalances well. To address this, we introduce a group distributionally robust objective (DRO) that assigns greater weight to classes with higher losses during training.

We first define a contrastive loss for a specific class k as:

$$h_k = \frac{1}{2n_k} \sum_{i=1}^{n_k} (\tau \log g_1(\mathbf{w}, \mathbf{x}_i, \mathcal{D}^t \cup \mathcal{M}^t) + \tau \log g_2(\mathbf{w}, y_i, \mathcal{D}^t \cup \mathcal{M}^t)) \tag{5}$$

where g_1 and g_2 are computed for negative samples and are influenced by a pairwise squared hinge loss. This formulation improves learning, especially in the context of partial AUC loss [20].

The group DRO objective is then $\min_{\mathbf{w}} \max_{\mathbf{p} \in \Delta} \sum_{i=0}^{K_t} p_k h_k - \lambda \mathrm{KL}(\mathbf{p}, 1/K_t)$ or equivalently,

$$\min_{\mathbf{w}} \lambda \log \frac{1}{K_t} \sum_{k=1}^{K_t} \exp\left(\frac{h_k}{\lambda}\right) \tag{6}$$

This objective increases the weight for harder classes (those with higher losses) to reduce the imbalance.

In its compositional form, the DRO involves nested functions, making gradient estimation challenging. To address this, we apply a method from Stochastic Compositional Optimization, maintaining moving average estimators for the loss terms and contrastive components. These estimators allow us to efficiently compute the gradient using mini-batches:

$$\frac{1}{v|\mathcal{B}_c|} \sum_{c_k \in \mathcal{B}_c} \exp\left(\frac{u_{c_k}}{\lambda}\right) \frac{1}{2|\mathcal{B}_k|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}_k} \left(\tau \frac{1}{u_i^I} \nabla g_1 + \tau \frac{1}{u_i^T} \nabla g_2\right)$$

This approach ensures robust handling of imbalanced data distributions while efficiently optimizing the DRO objective.

4 Experiments

In this section, we evaluate the effectiveness of the two methods referred to as GCL and GDRO in the class and domain incremental learning setting. For each of our experiments, we begin with a pretrained CLIP model [3, 8]. Our experiments are written in PyTorch [11] and are run on 4 NVIDIA RTX A5000 GPUs.

Algorithm 1 The GDRO Method for Continual Learning of CLIP models

```
Set $\mathbf{u}^0 = 0, v^0 = 0$ and initialize $\mathbf{w}$
for $t = 1, \dots, T$ do
    Sample a batch $\mathcal{B}$
    For each class $c_k \in \mathcal{B}$, sample a minibatch of data points denoted by $\mathcal{B}_k$.
    For each $c_k \in \mathcal{B}_c$, update $u_k^{\mathbf{I}(j)} = (1-\gamma)u_k^{\mathbf{I}(j-1)} + \gamma g_1(\mathbf{w}, \mathbf{x}_i, \mathcal{D}^t \cup \mathcal{M}^t)$
    For each $c_k \in \mathcal{B}_c$, update $u_k^{\mathbf{T}(j)} = (1-\gamma)u_k^{\mathbf{T}(j-1)} + \gamma g_2(\mathbf{w}, y_i, \mathcal{D}^t \cup \mathcal{M}^t)$
    For each $c_k \in \mathcal{B}_c$, update $u_{c_k}^{(j)} = (1-\gamma)u_{c_k}^{(j-1)} + \gamma h_k$
    Let $v^{(j)} = (1-\gamma)v^{(j-1)} + \gamma \frac{1}{K}\sum_{i=1}^{K} \exp\left(\frac{u_k^t}{\lambda}\right)$
    Compute a gradient estimator $\nabla_j$ by
        $$\frac{1}{v|\mathcal{B}_c|}\sum_{c_k \in \mathcal{B}_c} \exp\left(\frac{u_{c_k}}{\lambda}\right) \frac{1}{2|\mathcal{B}_k|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{B}_k} \left(\tau \frac{1}{u_i^I}\nabla g_1 + \tau \frac{1}{u_i^T}\nabla g_2\right)$$
    Update $\mathbf{v}_j = (1-\beta_1)\mathbf{v}_{j-1} + \beta_1 \nabla_j$
    Update $\mathbf{w}_{j+1} = \mathbf{w}_j - \eta_1 \mathbf{v}_j$ (or Adam-style)
end for
```

4.1 Datasets

We consider two class incremental datasets, namely CIFAR-100 and ImageNet. For domain incremental learning we evaluate our methods on DomainNet [12] which consists nearly 0.6 million images from 6 domains with 345 imbalanced classes.

4.2 Evaluation

We measure the performance of a model by its ability to perform on the current task and the previous tasks that it has been previously trained on. In this section, we describe the metrics used throughout our experiments to evaluate our method. To show the learning process, after each stage, the trained model is evaluated over all classes that have already been trained, i.e., the t-th test set $\mathcal{D}^t_{test} = \{(\mathbf{x}_i, y_i)\}$, $y_i \in \mathcal{Y}_t = Y_1 \cup ...Y_t$. Denoted by A_t the accuracy evaluated on $\mathcal{D}^t_{test}$ after stage t.

4.3 Baselines

In our image experiments, we evaluate various baseline methods using a pretrained CLIP model with a VIT-B/16 vision encoder, starting with a zero-shot performance assessment to gauge prior knowledge. Our goal is to outperform this baseline with methods compared against benchmarks such as EWC [9], DER [19], iCaRL [14], Co2L [1], and FOSTER [16], all utilizing the same pretrained encoder. Some dynamic expansion methods were omitted due to computational constraints.

For domain-incremental experiments on image datasets, we compare our approach with CaSSLe [5], focusing on the supervised contrastive objective for optimal results.

We ensure fair comparison by using consistent weight decay, batch size, and optimizer settings across methods, while fine-tuning the learning rate and number of epochs. For our DRO method, we additionally tune hyperparameters such as γ, λ, and the margin. We also vary memory sizes to test the effectiveness of our methods under different conditions.

4.4 Class Incremental Learning

ImageNet1k Data We further test our approach on the ImageNet1K dataset, splitting it into 10 tasks with 100 classes each. Due to the larger number of classes, we evaluate the methods with larger memory sizes. A finetuning baseline is also included, where the model is trained on all available data to establish an upper performance bound. Results are illustrated in Figure 1.

Our methods significantly outperform others across all memory sizes. Notably, unimodal contrastive approaches like Co2L [1] experience a sharp performance drop as memory size decreases. This is

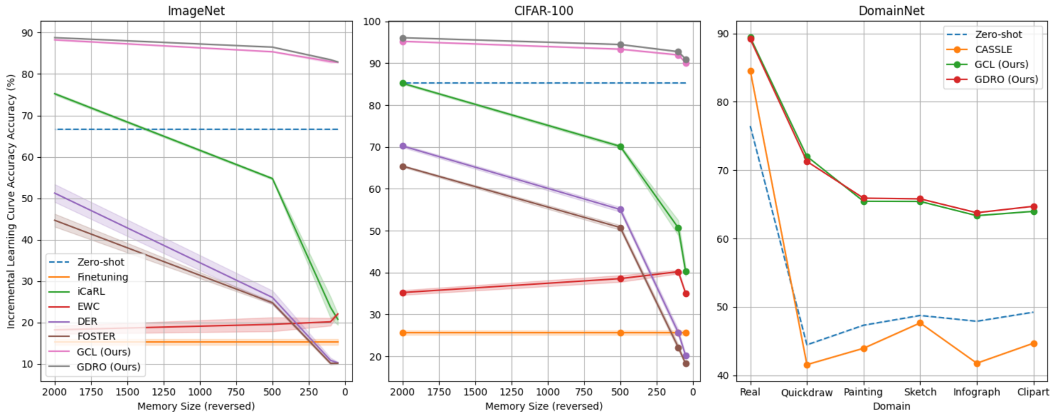

Figure 1: We report the mean and standard deviation of incremental learning curve accuracy over 3 runs on ImageNet1k at different memory sizes.

because Co2L relies on a self-supervised contrastive objective and requires labeled data from the memory bank for downstream tasks, which is limited when memory size is small.

CIFAR-100 Data We evaluate our method in a class-incremental learning (CIL) setting on the CIFAR-100 dataset, which is split into 10 tasks of 10 classes each. We assess performance across various memory sizes and report the accuracy after the final task. In addition, we compare our GDRO method with a baseline where finetuning is done solely with cross-entropy loss at each new task. Results are shown in Figure 1.

As memory size decreases, our method performs comparably to zero-shot evaluation, indicating that while some forgetting occurs, our method maintains solid performance as it progresses through tasks. When comparing the contrastive method with the DRO method, we observe that the contrastive method performs better with larger memory sizes, but its performance drops significantly when no memory is available. In contrast, the DRO method maintains more stable performance under memory constraints.

4.5 Domain Incremental Learning

We evaluate our methods in the domain incremental learning (DIL) setting, beginning with the image-based DomainNet dataset. Accuracy is assessed after each task as performance on all prior tasks, and we also report the model's zero-shot performance before any training. Results are shown in Figure 1.

Both of our methods outperform the baseline zero-shot results. As seen in our CIL experiments, contrastive CL methods like CaSSLe [5] struggle to retain knowledge from previous tasks due to the absence of a memory bank, as it relies on a self-supervised objective at each step. In contrast, our DRO objective outperforms the GCL method after completing all tasks, demonstrating better retention and adaptability.

5 Conclusion and Discussion

We propose two methods using bimodal contrastive learning to jointly embed labels and input data for CL. The first incorporates label embeddings with a memory buffer to retain past task knowledge, while the second dynamically reweights harder examples to address class imbalance in the buffer.

Using a pretrained CLIP vision encoder, we evaluate these methods in class-incremental and domain-incremental learning on image datasets. Our contrastive method excels with a larger memory buffer, while dynamic reweighting proves most effective with a smaller buffer.

The results show that embedding both input data and labels reduces forgetting more effectively than linear classifiers. Reweighting classes enhances retention, especially with limited memory, highlighting the benefits of multimodal learning and adaptive weighting for CL in dynamic environments.

References

[1] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525, 2021.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[3] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[4] Yawen Cui, Zitong Yu, Rizhao Cai, Xun Wang, Alex C. Kot, and Li Liu. Generalized few-shot continual learning with contrastive mixture of adapters, 2023.

[5] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022.

[6] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models, 2024.

[7] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023.

[8] Ryan King, Tianbao Yang, and Bobak J Mortazavi. Multimodal pretraining of medical time series and notes. In *Machine Learning for Health (ML4H)*, pages 244–255. PMLR, 2023.

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[10] Jiyong Li, Dilshod Azizov, Yang Li, and Shangsong Liang. Contrastive continual learning with importance sampling and prototype-instance relation distillation, 2024.

[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[14] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[15] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner, 2022.

[16] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414. Springer, 2022.

[17] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[18] Zhuoning Yuan, Yuexin Wu, Zi-Hao Qiu, Xianzhi Du, Lijun Zhang, Denny Zhou, and Tianbao Yang. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR, 2022.

[19] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. *arXiv preprint arXiv:2205.13218*, 2022.

[20] Dixian Zhu, Gang Li, Bokun Wang, Xiaodong Wu, and Tianbao Yang. When AUC meets DRO: optimizing partial AUC for deep learning with non-convex convergence guarantee. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27548–27573. PMLR, 2022.