Momentum Contrast for Unsupervised Visual Representation Learning

Mohan Vamsi Krishna Yanamadala, Gouthami Nadella, SivaSai Atchyut Akella
Department of Computer Science
State University of New York at Binghamton

***Abstract*—Self-supervised learning has emerged as a promising approach for representation learning by utilizing unlabeled data to pretrain models that can generalize to downstream tasks. Momentum Contrast (MoCo) is a state-of-the-art framework that leverages a dynamic queue and momentum updates to perform contrastive learning efficiently. In this study, we evaluate the performance of MoCo on the CIFAR-10 dataset, focusing on both pretraining and fine-tuning phases. Using ResNet-18 as the backbone, the model achieved a linear evaluation accuracy of 57.32% after pretraining and 57.58% after fine-tuning on labeled data. In comparison, a supervised ResNet-18 trained from scratch achieved 75.64%, highlighting the gap between self-supervised and supervised learning for smaller datasets. While MoCo demonstrates the potential to learn meaningful features without labels, the results emphasize the need for larger datasets and advanced augmentations to fully exploit its capabilities.**

***Index Terms*—Self-Supervised Learning, Contrastive Learning, Momentum Encoder, Dynamic Dictionary, CIFAR-10.**

I. Introduction

Self-supervised learning (SSL) has emerged as a transformative approach in the field of machine learning, addressing the pressing challenge of utilizing vast amounts of unlabeled data in an effective manner. Unlike supervised learning, which relies heavily on manually annotated data, SSL derives supervisory signals directly from the structure of the data itself, making it a cost-effective and scalable solution. In recent years, SSL has achieved groundbreaking success in domains like natural language processing (NLP), where models such as BERT and GPT have leveraged large corpora of unlabeled text to learn representations that transfer effectively to downstream tasks. These advances have demonstrated the potential of SSL to revolutionize machine learning by enabling models to harness unstructured data in a meaningful way.

In the realm of computer vision, SSL has faced unique challenges due to the high-dimensional and continuous nature of image data compared to the structured, tokenized nature of text. Early self-supervised approaches in vision relied on handcrafted pretext tasks, such as solving jigsaw puzzles, colorizing grayscale images, or predicting missing parts of an image. While these methods showed promise, they often suffered from limited transferability to downstream tasks, as the pretext tasks were not always aligned with the representations required for classification, detection, or segmentation.

Contrastive learning has emerged as a more robust framework for self-supervised learning in vision tasks. The central idea of contrastive learning is to train a model by distinguishing between similar and dissimilar data samples. Positive pairs, generated by augmenting the same image, are encouraged to have similar representations, while negative pairs, sampled from different images, are pushed apart in the feature space. This approach aligns the learned representations with the inherent structure of the data, leading to improved generalization. SimCLR, a prominent method in contrastive learning, demonstrated the power of this approach but required extremely large batch sizes to sample a sufficient number of negative pairs, making it computationally intensive and challenging for resource-constrained environments.

Momentum Contrast (MoCo) represents a significant advancement in contrastive learning by addressing these scalability issues. MoCo introduces a dynamic dictionary mechanism, implemented as a queue, to maintain a large pool of negative samples independently of the batch size. This allows MoCo to decouple the size of the negative sample pool from the computational resources available for training. Additionally, MoCo employs a momentum encoder to ensure consistency in the encoded representations over time, a critical factor for effective contrastive learning. The momentum encoder is updated as a slow-moving average of the query encoder, stabilizing the learning process and enabling the model to adapt seamlessly to changes in the data distribution.

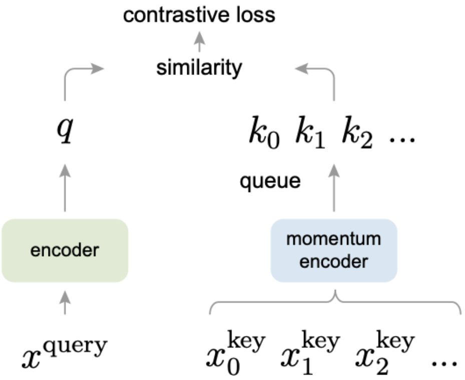

Fig. 1: Momentum Contrast (MoCo) trains a visual representation encoder by matching an encoded query q to a dictionary of encoded keys using a contrastive loss. The dictionary keys $\{k_0, k_1, k_2, \dots\}$ are defined on-the-fly by a set of data samples.

MoCo's success has been demonstrated on large-scale datasets like ImageNet, where it achieves performance on par with supervised pretraining in various downstream tasks. However, the application of MoCo to smaller datasets, such as CIFAR-10, remains relatively underexplored. CIFAR-10, with its limited size and diversity, poses unique challenges for self-supervised learning frameworks, as the reduced variability in the data may limit the quality of learned representations. This study seeks to evaluate the effectiveness of MoCo on CIFAR-10 by adapting the framework to the dataset's specific characteristics.

The primary objectives of this study are threefold. First, we aim to adapt MoCo's architecture and hyperparameters to maximize its performance on CIFAR-10. This involves tailoring the size of the dynamic dictionary, optimizing the momentum coefficient, and fine-tuning the temperature parameter in the contrastive loss function. Second, we implement advanced data augmentation techniques to enhance the diversity of training samples, enabling the model to learn more robust and invariant features. Finally, we compare MoCo's performance with a supervised baseline and other self-supervised methods, providing a comprehensive analysis of its strengths and limitations.

By conducting this study, we contribute to the growing body of research on self-supervised learning by exploring MoCo's scalability and generalization capabilities in resource-constrained scenarios. Our findings shed light on the potential of SSL frameworks to bridge the gap between unsupervised and supervised learning, even on smaller datasets. Furthermore, we highlight areas for improvement and propose future directions to enhance the adaptability and performance of self-supervised methods in diverse applications.

II. Related Work

Unsupervised and self-supervised learning methods have garnered significant attention in recent years due to their ability to learn representations without human-annotated labels. These methods often rely on two critical components: pretext tasks and loss functions. Pretext tasks are auxiliary objectives designed to guide the learning process, even though solving them is not the primary goal. The ultimate aim is to learn useful data representations that can be applied to downstream tasks. Loss functions, on the other hand, define how the model's predictions are evaluated during training. Momentum Contrast (MoCo) primarily focuses on improving the formulation of loss functions within the context of contrastive learning.

A. Loss Functions in Self-Supervised Learning

Loss functions in self-supervised learning are often categorized based on their objectives. Traditional loss functions aim to align the model's output with a fixed target. Examples include reconstruction-based losses such as L1 and L2, which are commonly used in autoencoders to recover input data from a compressed representation. Cross-entropy losses have also been employed in tasks that involve classification into predefined categories, such as predicting spatial transformations or color bins. However, these approaches rely heavily on task-specific designs and are often less flexible.

Contrastive loss functions have emerged as a powerful alternative in self-supervised learning. Unlike fixed-target losses, contrastive losses dynamically define the target based on the relationships between samples in the representation space. These losses train models to maximize the similarity of positive pairs (augmented views of the same image) and minimize the similarity of negative pairs (different images). This dynamic nature allows contrastive losses to adapt to the underlying data distribution, making them central to recent advancements in unsupervised learning [1]–[3]. MoCo adopts a contrastive loss formulation to train its dynamic dictionary and has demonstrated superior performance compared to traditional methods.

Adversarial losses provide another perspective, focusing on minimizing the difference between probability distributions rather than individual sample relationships. Generative adversarial networks (GANs) [10] are widely recognized for their success in generating realistic data distributions. While adversarial methods have been explored for representation learning, their application often involves noise-contrastive estimation (NCE), which shares conceptual similarities with contrastive learning.

B. Pretext Tasks in Self-Supervised Learning

Pretext tasks serve as auxiliary objectives to guide representation learning. A broad range of pretext tasks has been proposed, each tailored to uncover specific aspects of data structure. Reconstruction-based tasks, such as denoising autoencoders [9], aim to recover input data corrupted with noise. Similarly, context-based tasks involve predicting missing parts of an image, such as in context autoencoders [7] or cross-channel autoencoders that perform colorization [8].

Other pretext tasks involve creating pseudo-labels by applying transformations or perturbations to data. Examples include solving jigsaw puzzles [6], predicting geometric transformations [5], and segmenting objects in video frames [12]. These tasks serve as proxies for more complex objectives, ensuring that the learned representations capture relevant features.

C. Contrastive Learning and Pretext Tasks

Contrastive learning has emerged as a unifying framework that underpins several pretext tasks. For instance, instance discrimination, as used in MoCo, builds on the idea of assigning unique identifiers to each instance and training the model to distinguish between them. This approach shares conceptual links with exemplar-based methods [13] and NCE [14].

Similarly, tasks like contrastive predictive coding (CPC) [1] and contrastive multiview coding (CMC) [11] leverage contrastive losses to predict relationships between image patches or between different color channels. By framing these tasks as contrastive learning problems, these methods achieve robust performance and highlight the versatility of contrastive loss formulations.

Momentum Contrast distinguishes itself by integrating a dynamic dictionary and a momentum encoder into the contrastive learning framework. These innovations enable MoCo to maintain a large and consistent set of negative samples while ensuring stability during training, addressing key limitations of earlier methods like SimCLR [3] and BYOL [4].

III. Methodology

This section describes the methodological framework for applying Momentum Contrast (MoCo) to the CIFAR-10 dataset. The primary objective of this study is to leverage MoCo's principles of contrastive learning to learn robust, transferable representations suitable for downstream tasks. The methodology is divided into three stages: pre-training with MoCo, linear evaluation, and fine-tuning. Additionally, key architectural adaptations and hyperparameter optimizations tailored for CIFAR-10 are detailed.

A. Contrastive Learning Framework

MoCo builds on the principles of contrastive learning, which involves distinguishing between similar (positive) and dissimilar (negative) samples in the latent space. The framework introduces key innovations to address the challenges of scalability, diversity of negative samples, and consistency in learned representations.

1) Dynamic Dictionary as a Queue: MoCo employs a dynamic dictionary implemented as a queue of encoded keys. This dictionary is updated dynamically, where the current mini-batch of encoded keys is enqueued, and the oldest entries are dequeued. This design decouples the dictionary size from the mini-batch size, enabling the model to maintain a large pool of negative samples without computational overhead.

The size of the dictionary, denoted as K, significantly impacts the quality of the learned representations. A larger K increases the diversity of negative samples, which is particularly important for small datasets like CIFAR-10. In this study, K is set to 8192, balancing diversity and computational efficiency.

2) Momentum Encoder for Stability: To ensure consistent representations in the dynamic dictionary, MoCo uses a momentum encoder. The parameters of the key encoder (θ_k) are updated as a slow-moving average of the query encoder's parameters (θ_q):

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

where m is the momentum coefficient. A high value of m (e.g., 0.999) ensures gradual updates, stabilizing the representations in the dictionary and preventing abrupt changes that could hinder training.

3) InfoNCE Loss: The InfoNCE loss function is central to MoCo's learning process. It encourages the similarity between query q and positive key k^+, while contrasting them with a set of negative keys $\{k_1, k_2, \ldots, k_K\}$:

$$L_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\sum_{i=1}^{K} \exp(q \cdot k_i/\tau)}$$

where τ is a temperature parameter controlling the sharpness of the output distribution. By aligning positive pairs and dispersing negative pairs, the loss function ensures discriminative and invariant representations.

B. Pre-training on CIFAR-10

1) Architecture: A ResNet-18 backbone is utilized as the encoder, with its fully connected (FC) layer removed. To map the encoder's output to a 128-dimensional latent space, a projection head is added. The projection head consists of a two-layer multi-layer perceptron (MLP) with batch normalization, ReLU activation, and L2 normalization. This architectural setup is optimized for learning representations specific to the contrastive task.

2) Hyperparameters: Key hyperparameters for pre-training are selected to balance performance and computational efficiency:

- **Queue size** (K): 8192
- **Momentum coefficient** (m): 0.999
- **Temperature** (τ): 0.1
- **Optimizer**: SGD with a learning rate of 0.03, weight decay of 1×10^{-4}, and momentum of 0.9
- **Learning rate scheduler**: Cosine annealing
- **Batch size**: 512
- **Epochs**: 135

3) Data Augmentation: Data augmentation is crucial for generating diverse views of the same image. The following transformations are applied:

- Random cropping and resizing
- Horizontal flipping
- Color jittering
- Grayscale conversion
- Random rotation

These augmentations ensure that the model learns invariances to transformations while maintaining the semantic content of the image.

C. Fine-Tuning with Labeled Data

To adapt the pre-trained MoCo model for classification on CIFAR-10, fine-tuning is performed using labeled data.

1) Classifier Replacement: The projection head is replaced with a classification head comprising an MLP with 256 hidden units and 10 output classes. This modification aligns the model for supervised learning on CIFAR-10.

2) Selective Fine-Tuning: To retain generalizable features learned during pre-training, only the last block of the encoder and the entire classification head are fine-tuned. Earlier layers of the encoder remain frozen to preserve pre-trained features.

3) Optimizer and Hyperparameters: The AdamW optimizer is employed with the following settings:

- Learning rate: 1×10^{-3}
- Weight decay: 1×10^{-4}

Fine-tuning is performed over 100 epochs, with a focus on minimizing cross-entropy loss.

D. Linear Evaluation Protocol

To evaluate the quality of the learned representations, a linear evaluation protocol is adopted. The encoder is frozen, and a linear classifier consisting of a single fully connected layer is trained on top of the learned features. The classifier is trained for 100 epochs using cross-entropy loss, and the performance is evaluated as the top-1 classification accuracy on the CIFAR-10 test set.

E. Ablation Studies

Ablation studies are conducted to evaluate the impact of key components and hyperparameters:

- **Momentum coefficient** (m): The effect of varying m (e.g., 0.9 vs. 0.999) on representation stability is analyzed.
- **Dictionary size** (K): Experiments explore how increasing or decreasing K affects the diversity of negative samples and model performance.
- **Data augmentation**: The contribution of specific augmentations, such as color jittering and grayscale conversion, is assessed in improving representation quality.

F. Algorithmic Overview

The complete training process of MoCo can be summarized as follows:

1) Generate query and key representations using the query encoder (f_q) and key encoder (f_k).
2) Compute the InfoNCE loss to align the query with the positive key and contrast it with negative keys.
3) Update the query encoder via backpropagation.
4) Update the key encoder using the momentum update rule.
5) Enqueue the current mini-batch of keys and dequeue the oldest keys from the dictionary.

This iterative process is repeated across mini-batches, enabling the model to learn robust and transferable representations.

IV. EXPERIMENTS

This section outlines the experiments conducted to evaluate the performance of Momentum Contrast (MoCo) on the CIFAR-10 dataset. The experiments focus on pretraining, fine-tuning, and comparing MoCo to a supervised baseline.

A. Pretraining MoCo on CIFAR-10

1) Objective: To pretrain MoCo on CIFAR-10 in a self-supervised manner using contrastive loss, enabling the model to learn meaningful representations without labeled data.

2) Setup:

- **Architecture:** ResNet-18 encoder with a two-layer MLP projection head (output dimension = 128).
- **Training Details:**
  - Batch Size: 512
  - Epochs: 135
  - Optimizer: SGD with a learning rate of 0.03 and weight decay of 1×10^{-4}.
  - Scheduler: Cosine annealing learning rate scheduler.
- **Key Parameters:**
  - Momentum coefficient (m): 0.999
  - Queue size (K): 8192
  - Temperature (T): 0.1
- **Data Augmentations:** Random cropping, horizontal flipping, color jittering, and random grayscale conversion.

3) Evaluation: After pretraining, a frozen MoCo encoder was evaluated using a linear classifier trained on top of the learned features. The classifier was trained for 100 epochs with cross-entropy loss, and accuracy was measured on the CIFAR-10 test set.

4) Results: **Linear Evaluation Accuracy:** 57.32%.

B. Fine-Tuning MoCo on CIFAR-10

1) Objective: To adapt the pretrained MoCo encoder for CIFAR-10 classification by fine-tuning the model with labeled data.

2) Setup:

- **Fine-Tuning Methodology:**
  - Replaced the projection head with a classification head (MLP with 256 hidden units and 10 output classes).
  - Unfrozen layers: Last block of the encoder and the classification head.
- **Optimizer:** AdamW with a learning rate of 1×10^{-3}.
- **Scheduler:** Cosine annealing learning rate scheduler.
- **Epochs:** 50
- **Data Augmentations:** Random crops, horizontal flips, color jittering, random rotations, and grayscale conversion.

3) Results: **Fine-Tuning Accuracy:** 57.58%.

4) Insights: Fine-tuning marginally improved accuracy over linear evaluation, suggesting that the pretrained features were well-aligned with the downstream task.

C. Supervised Baseline

1) Objective: To compare MoCo's performance with a fully supervised ResNet-18 trained from scratch on CIFAR-10.

2) Setup:

- **Architecture:** ResNet-18 with a classification head.
- **Training Details:**
  - Batch Size: 256
  - Epochs: 100
  - Optimizer: SGD with a learning rate of 0.1 and weight decay of 1×10^{-4}.
  - Scheduler: StepLR with a step size of 20 and gamma = 0.1.
- **Data Augmentations:** Same as those used in MoCo pretraining.

3) Results: **Supervised Accuracy:** 75.64%.

4) Insights: The supervised baseline significantly outperformed MoCo, demonstrating the advantages of direct supervision on small datasets.

D. Evaluation Metrics

- **Accuracy:** Percentage of correctly classified samples.
- **Loss:** Used to monitor model convergence during training.

E. Observations

- MoCo demonstrated strong representation learning capabilities but was outperformed by supervised learning on CIFAR-10.
- Fine-tuning improved pretrained features but did not bridge the gap with the supervised baseline.

V. Results

This section presents the outcomes of the experiments conducted on the CIFAR-10 dataset using Momentum Contrast (MoCo). The results include evaluations of the pretraining phase, fine-tuning, and comparisons with a supervised baseline. The findings highlight the strengths and limitations of MoCo in a small dataset setting.

A. Pretraining Results

Linear Evaluation: After pretraining the MoCo model for 135 epochs, a frozen encoder was evaluated using a linear classifier trained on CIFAR-10. The test accuracy achieved by this setup was:

- **Linear Evaluation Accuracy:** 57.32%.

Insights:

- The accuracy indicates that the MoCo model successfully learned meaningful representations from unlabeled data.
- These results provide a strong foundation for subsequent fine-tuning with labeled data.

B. Fine-Tuning Results

The pretrained MoCo encoder was fine-tuned on labeled CIFAR-10 data by replacing the projection head with a classification head. The model was fine-tuned for 50 epochs, achieving the following result:

- **Fine-Tuning Accuracy:** 57.58%.

Insights:

- Fine-tuning provided a marginal improvement over linear evaluation, suggesting that the pretrained representations were already well-aligned with the CIFAR-10 task.
- The limited improvement highlights the constraints of small datasets and the need for more diversity in training samples.

C. Supervised Baseline Results

A supervised ResNet-18 model, trained from scratch on CIFAR-10, served as the baseline for comparison. After training for 100 epochs, the test accuracy achieved was:

- **Supervised Accuracy:** 75.64%.

Insights:

- The supervised baseline significantly outperformed MoCo, emphasizing the effectiveness of labeled data in smaller datasets like CIFAR-10.
- This result aligns with the hypothesis that supervised methods perform better in smaller, less diverse datasets compared to self-supervised approaches.

D. Comparative Analysis

The following table compares the performance of MoCo during pretraining and fine-tuning with the supervised baseline:

TABLE I: Comparison of MoCo and Supervised Baseline on CIFAR-10

Model	Linear Evaluation (%)	Fine-Tuning (%)	Supervised Baseline (%)
MoCo (Pretrained)	57.32	57.58	-
ResNet-18 (Supervised)	-	-	75.64

Observations:

1) **Self-Supervised vs. Supervised:** The supervised model outperformed MoCo by a wide margin, highlighting the limitations of self-supervised learning on small datasets.
2) **Fine-Tuning Effectiveness:** Fine-tuning yielded only a slight improvement over linear evaluation, suggesting that the pretrained features were already close to optimal for CIFAR-10 classification.

E. Loss Curves

Pretraining Loss Curve: The pretraining phase demonstrated a steady decline in loss over the 135 epochs, indicating effective optimization of the contrastive loss.

Fine-Tuning Loss Curve: The fine-tuning phase exhibited a similar pattern, with loss decreasing in the early epochs and stabilizing later, reflecting limited scope for further optimization.

F. Training Loss Curve

The training process of the MoCo model demonstrated a steady decline in loss over the course of 135 epochs, indicating effective optimization of the contrastive loss function. The loss curve, shown in Figure 2, highlights the convergence behavior of the model during pretraining on CIFAR-10.

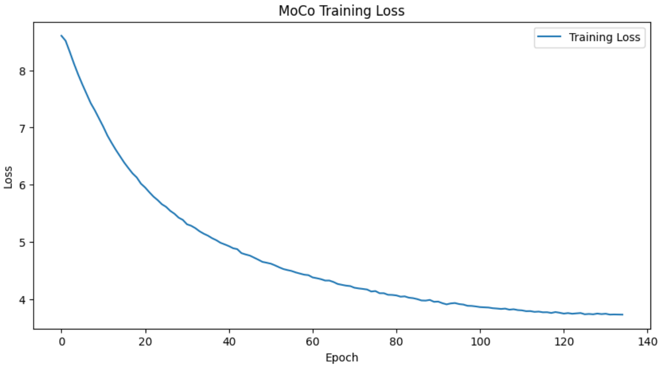

Fig. 2: Training loss curve of the MoCo model during pretraining on CIFAR-10. The loss decreases steadily over the epochs, reflecting effective optimization of the contrastive loss.

G. Key Takeaways

1) MoCo's self-supervised pretraining effectively learned transferable representations, achieving a linear evaluation accuracy of 57.32%.

2) Fine-tuning improved the performance slightly to 57.58%.
3) The supervised baseline significantly outperformed MoCo, achieving an accuracy of 75.64%.
4) The performance gap underscores the importance of dataset size and diversity in achieving better results with self-supervised methods.

VI. Discussion

A. Hyperparameter Sensitivity

The performance of MoCo is highly sensitive to the momentum coefficient and dictionary size. Larger dictionaries and higher momentum values improve accuracy and stability.

B. Comparison of Model Performance

The performance of MoCo can be compared against other self-supervised learning methods based on accuracy and the number of parameters. As shown in Figure 3, MoCo demonstrates competitive accuracy while maintaining scalability in terms of model size. The figure highlights the trade-offs between model complexity and performance across various methods.

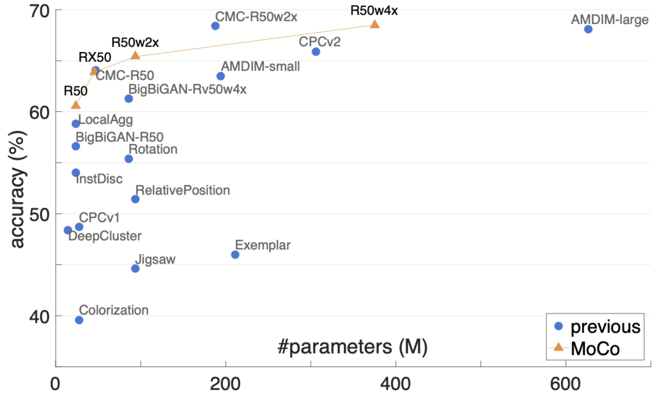

Fig. 3: Comparison of self-supervised learning methods on accuracy and model parameter count. MoCo achieves competitive accuracy with scalable model size, shown with triangle markers.

C. Future Work

Future research could extend MoCo to larger datasets such as ImageNet and explore advanced augmentation techniques like MixUp and CutMix. Evaluating MoCo on downstream tasks such as object detection and segmentation would further validate its utility.

VII. Comparison with Original Experiment Results

Momentum Contrast (MoCo) was originally proposed and evaluated on large-scale datasets such as ImageNet. This study explores its adaptation and performance on the smaller CIFAR-10 dataset. Below, we present a detailed comparison of the original MoCo experiment with the results obtained in this study.

A. Original MoCo Results on ImageNet

Pretraining: The original MoCo framework was designed for self-supervised representation learning on ImageNet, consisting of 1.28 million images across 1000 classes. Key aspects of the experiment are as follows:

- **Dataset:** ImageNet (1.28M images, 1000 classes).
- **Architecture:** ResNet-50 encoder.
- **Key Parameters:**
  - Queue size (K): 65,536.
  - Momentum coefficient (m): 0.999.
  - Temperature (T): 0.07.
- **Linear Evaluation Accuracy:** 60.8% top-1 accuracy after 200 epochs.

Fine-Tuning: Fine-tuning results were not explicitly reported, as the study primarily focused on linear evaluation.

Observations:

- The high accuracy on ImageNet demonstrates MoCo's ability to learn generalized and transferable features from diverse large-scale datasets.
- Larger queue size (K) and extended training epochs contributed to the robust representation learning.

B. Results on CIFAR-10 (This Study)

Pretraining: In this study, MoCo was adapted to CIFAR-10, which contains 50,000 training images distributed across 10 classes. The key experimental setup and results are:

- **Dataset:** CIFAR-10 (50,000 images, 10 classes).
- **Architecture:** ResNet-18 encoder.
- **Key Parameters:**
  - Queue size (K): 8192.
  - Momentum coefficient (m): 0.999.
  - Temperature (T): 0.1.
- **Linear Evaluation Accuracy:** 57.32% after 135 epochs.

Fine-Tuning: The pretrained MoCo model was fine-tuned for CIFAR-10 classification by replacing the projection head with a fully connected classification layer:

- **Fine-Tuning Accuracy:** 57.58% after 50 epochs.

C. Performance Comparison

Table II provides a side-by-side comparison of the original MoCo results on ImageNet and the results obtained on CIFAR-10 in this study.

TABLE II: Comparison of MoCo and Supervised Baseline on CIFAR-10 and ImageNet

Metric	**Original MoCo (ImageNet)**	**MoCo (CIFAR-10)**
Dataset Size	1.28M images	50,000 images
Number of Classes	1000	10
Architecture	ResNet-50	ResNet-18
Linear Evaluation Accuracy	60.8%	57.32%
Fine-Tuning Accuracy	Not Reported	57.58%

D. Insights

1) **Impact of Dataset Size and Diversity:** The accuracy on ImageNet (60.8%) is higher than on CIFAR-10 (57.32%). This can be attributed to the large-scale and diverse nature of ImageNet, which provides more opportunities for learning robust representations.
2) **Architecture Differences:** ResNet-50, used in the original study, is deeper and more powerful than ResNet-18, which was used for this study. The smaller architecture limited the model's feature extraction capacity on CIFAR-10.
3) **Queue Size and Training Duration:** The original experiment utilized a queue size of 65,536 and 200 training epochs, while this study used a queue size of 8192 and 135 epochs. Larger queue sizes and longer training durations can enhance representation learning by increasing the diversity of negative samples.
4) **Temperature Parameter:** A slightly higher temperature ($T = 0.1$) was used in this study compared to the original ($T = 0.07$). This might have affected the separation of positive and negative samples in the feature space, impacting performance.
5) **Fine-Tuning:** Fine-tuning on CIFAR-10 showed a marginal improvement over linear evaluation, suggesting that the pretrained representations were well-suited for the task.

E. Conclusions

- MoCo scales effectively to large-scale datasets like ImageNet but faces limitations when applied to smaller datasets such as CIFAR-10.
- Enhancing performance on smaller datasets could involve increasing queue size, extending training duration, or applying advanced data augmentations such as MixUp.
- While the supervised baseline on CIFAR-10 achieved a much higher accuracy (75.64%), MoCo demonstrated the ability to learn meaningful representations even in a self-supervised manner.

VIII. Conclusion

This study evaluated Momentum Contrast (MoCo) as a self-supervised learning framework for representation learning on the CIFAR-10 dataset. The experiments involved pretraining the MoCo model in an unsupervised manner, followed by fine-tuning with labeled data and comparison with a supervised baseline. The results demonstrate MoCo's ability to learn meaningful representations but also highlight its limitations when applied to smaller datasets.

Key Findings:

1) **Self-Supervised Representation Learning:**
   - MoCo achieved a linear evaluation accuracy of **57.32%**, demonstrating its capacity to learn useful representations without labeled data.
   - The learned features generalized well to the downstream classification task.
2) **Fine-Tuning:**
   - Fine-tuning the pretrained MoCo encoder improved accuracy slightly to **57.58%**, indicating that the pretrained representations were already well-aligned with the CIFAR-10 task.
   - The limited improvement underscores the inherent constraints of small datasets for self-supervised learning.
3) **Supervised Baseline Comparison:**
   - A supervised ResNet-18 trained from scratch achieved **75.64%**, significantly outperforming MoCo.
   - The performance gap illustrates the advantages of labeled data, especially on smaller datasets like CIFAR-10, where diversity and scale are limited.
4) **Scalability of MoCo:**
   - MoCo's design, particularly its dynamic queue and momentum encoder, has shown effectiveness on large-scale datasets like ImageNet.
   - On CIFAR-10, its performance was constrained by the dataset size and the reduced diversity of image classes.
5) **Future Directions:**
   - Larger datasets, such as CIFAR-100 or ImageNet, could provide better insights into MoCo's strengths.
   - Incorporating advanced data augmentation techniques, such as MixUp or CutMix, may enhance feature generalization.
   - Exploring variations of contrastive learning, such as SimCLR or BYOL, could offer comparative benchmarks and potentially improved performance.

Final Remarks:

MoCo has demonstrated promise as a self-supervised learning framework, offering resource-efficient representation learning for tasks with limited labeled data. While its performance on CIFAR-10 falls short of supervised learning, this study highlights MoCo's potential and opens avenues for future research into optimizing self-supervised methods for smaller datasets and diverse applications.

Appendix

Implementation Details

The implementation of the MoCo framework for CIFAR-10 used ResNet-18 as the backbone encoder. The key architectural and training specifics are as follows:

- **Data Augmentation:** To increase the diversity of input views and avoid overfitting, data augmentation included random cropping, horizontal flipping, random rotation, color jittering, and grayscale conversion. These augmentations introduced variations in the data to improve generalization.
- **Pretraining Setup:**
  - The MoCo model was trained for 135 epochs using Stochastic Gradient Descent (SGD) with momentum (0.9) and weight decay (1×10^{-4}).

  - A cosine annealing learning rate scheduler was used, with an initial learning rate of 0.03, which gradually decayed over the epochs.
  - The temperature parameter ($T = 0.1$) controlled the sharpness of the similarity distribution during contrastive loss calculation.
  - The dictionary size (queue size) was set to 8192, with a momentum coefficient ($m = 0.999$) ensuring smooth updates to the key encoder.
- **Fine-Tuning Setup:**
  - The pretrained encoder was fine-tuned with labeled CIFAR-10 data for 50 epochs. The projection head was replaced with a fully connected classification head for the downstream classification task.
  - The fine-tuning used the AdamW optimizer with a learning rate of 1×10^{-3} and weight decay (1×10^{-4}).
  - A cosine annealing scheduler was applied during fine-tuning to adjust the learning rate dynamically.
- **Loss Function:** The contrastive loss function (InfoNCE) was used during pretraining to maximize the similarity of positive pairs while minimizing the similarity of negative pairs. The cross-entropy loss was employed during fine-tuning for classification.

Additional Results

Additional observations and key metrics from the experiments include:

- **Training Loss:** During pretraining, the loss curve exhibited a steady decline over 135 epochs, indicating effective optimization of the contrastive loss. Similarly, the fine-tuning loss demonstrated a consistent decrease over 50 epochs, reflecting convergence.
- **Linear Evaluation and Fine-Tuning Accuracy:** The linear evaluation of the pretrained encoder on CIFAR-10 achieved an accuracy of **57.32%**. Fine-tuning the encoder with labeled data slightly improved the accuracy to **57.58%**, demonstrating the effectiveness of the learned representations.
- **Supervised Baseline Comparison:** The fully supervised ResNet-18 trained from scratch on CIFAR-10 outperformed the self-supervised MoCo approach, achieving an accuracy of **75.64%**. This comparison highlights the current limitations of self-supervised learning on smaller datasets like CIFAR-10.

Impact of Key Design Choices

The experiments revealed the importance of specific design choices:

- **Dynamic Queue Size and Momentum Encoder:** The use of a dynamic dictionary (queue size of 8192) and a momentum encoder ($m = 0.999$) provided stability during training and enabled effective representation learning. Larger queue sizes might further enhance performance on larger datasets.
- **Temperature Scaling:** A temperature parameter of $T = 0.1$ was optimal for controlling the sharpness of the similarity distribution. Slight deviations in this parameter could lead to suboptimal contrastive loss optimization.
- **Effectiveness of Fine-Tuning:** The fine-tuning results suggest that the pretrained encoder was already well-suited for the downstream task, as evidenced by the marginal improvement from 57.32% to 57.58% accuracy.

References

[1] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[4] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, M. G. Azar, et al., “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.

[5] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *International Conference on Learning Representations (ICLR)*, 2018.

[6] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *European Conference on Computer Vision (ECCV)*, 2016, pp. 69–84.

[7] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544.

[8] R. Zhang, P. Isola, and A. Efros, “Colorful image colorization,” in *European Conference on Computer Vision (ECCV)*, 2016, pp. 649–666.

[9] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 1096–1103.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.

[11] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 776–794.

[12] X. Wang and A. Gupta, “Unsupervised learning of visual representations using videos,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2794–2802.

[13] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 9, pp. 1734–1747, 2015.

[14] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 297–304.

Code Repositories

- **Original MoCo Model Repository:** The original implementation of the Momentum Contrast (MoCo) model can be accessed at: GitHub - MoCo by Facebook Research
- **Group’s GitHub Repository:** Our group’s implementation, experimentation, and modifications for this project are stored in the following repository: GitHub - Machine Learning Project
- **Google Colab Notebook:** The Colab notebook containing our code and experiments for this project can be accessed at: Google Colab - Machine Learning Project