Learning the Unlearned: Mitigating Feature Suppression in Contrastive Learning

Jihai Zhang*,[1], Xiang Lan*,[2], Xiaoye Qu[3], Yu Cheng[1],
Mengling Feng[2,†], Bryan Hooi[2,†]

[1]The Chinese University of Hong Kong
[2]National University of Singapore [3]Shanghai AI Laboratory

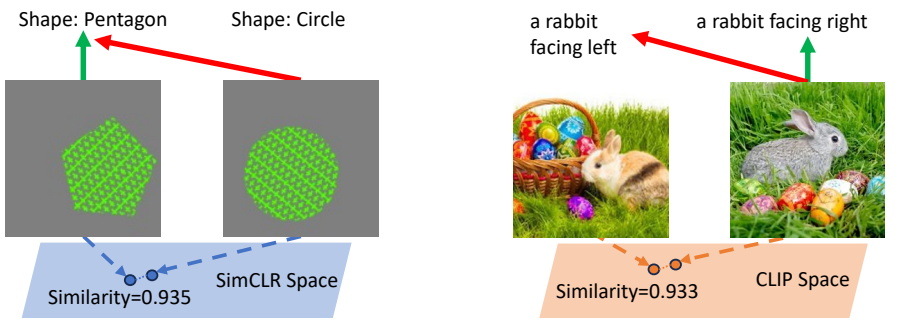
Abstract. Self-Supervised Contrastive Learning has proven effective in deriving high-quality representations from unlabeled data. However, a major challenge that hinders both unimodal and multimodal contrastive learning is feature suppression, a phenomenon where the trained model captures only a limited portion of the information from the input data while overlooking other potentially valuable content. This issue often leads to indistinguishable representations for visually similar but semantically different inputs, adversely affecting downstream task performance, particularly those requiring rigorous semantic comprehension. To address this challenge, we propose a novel model-agnostic **M**ultistage **C**ontrastive **L**earning (MCL) framework. Unlike standard contrastive learning which inherently captures one single biased feature distribution, MCL progressively learns previously unlearned features through *feature-aware negative sampling* at each stage, where the negative samples of an anchor are exclusively selected from the cluster it was assigned to in preceding stages. Meanwhile, MCL preserves the previously well-learned features by *cross-stage representation integration*, integrating features across all stages to form final representations. Our comprehensive evaluation demonstrates MCL's effectiveness and superiority across both **unimodal** and **multimodal** contrastive learning, spanning a range of model architectures from ResNet to Vision Transformers (ViT). Remarkably, in tasks where the original CLIP model has shown limitations, MCL dramatically enhances performance, with improvements up to threefold on specific attributes in the recently proposed MMVP benchmark. Codes are available at `https://github.com/MajorDavidZhang/MCL.git`.

Keywords: Self-Supervised Learning · Contrastive Learning · Feature Suppression

1 Introduction

Self-Supervised contrastive learning obtains high-quality representations by maximizing the similarity between an anchor and its associated positive samples,

* Equal Contribution † Correspondence to: Mengling Feng <ephfm@nus.edu.sg>, Bryan Hooi <bhooi@comp.nus.edu.sg>. Xiang Lan and Mengling Feng are affiliated with Saw Swee Hock School of Public Health and Institute of Data Science, NUS. Part of this work was done during Jihai Zhang's internship at Shanghai AI Laboratory.



(a) Images from Trifeature with different shapes and textures have high similarity in the SimCLR space.

(b) Images from MMVP with different orientations and directions have high similarity in the CLIP space.

Fig. 1: Demonstration of feature suppression in both unimodal (SimCLR) and multimodal (CLIP) settings. The green arrows refer to correct linear evaluation classification/ pairing; the red arrows refer to incorrect ones.

while concurrently increasing the separation among the dissimilar data samples in the embedding space [23]. Various contrastive learning models serve as fundamental pretrained backbones across different fields [6, 19, 37, 50]. However, recent studies [40, 45, 53] have shown that representations derived from standard contrastive learning often miss substantial portions of input information. This phenomenon is referred to as feature suppression. Such suppression can severely compromise the effectiveness of models in various downstream tasks, ranging from classification [40] to pattern recognition [47]. In addition, the feature suppression issues are also observed in multimodal contrastive learning [3,47], such as CLIP [37], which is predominantly adopted in current multimodal large language models(MLLMs) [32, 54, 58] as the vision encoder. Feature suppression in CLIP significantly impedes the capability of MLLMs to differentiate between images with varying semantics, resulting in severe hallucination problems [29–31, 47] within these models.

Referencing Fig. 1a, SimCLR trained on the Trifeature dataset fails to differentiate between a circle and a pentagon with the same shape and color. Similarly, the OpenAI pretrained CLIP model struggles to distinguish between a rabbit facing left and a rabbit facing right due to the high similarity of their representations in the embedding space, as illustrated in Fig. 1b. Consequently, MLLMs that utilize CLIP as their vision encoder experience systematic failures on tasks involving such distinctions. While feature suppression presents a critical challenge in contrastive learning, there are only a handful of methods proposed to address it. These approaches often come at the expense of compromising previously well-learned features [40, 45]. Alternatively, they necessitate an additional reconstruction loss [3], which is rendered impractical for large-scale applications such as CLIP due to the high computational demands. Furthermore, the scope of these existing methodologies is often restricted to either unimodal or multimodal contrastive learning, lacking universal applicability.

In this paper, we propose a novel model-agnostic framework: Multistage Contrastive Learning (MCL), designed to effectively tackle feature suppression in both **unimodal** and **multimodal** settings. Unlike standard single-stage contrastive learning that often collapses to certain features, MCL aims to progressively learn new features that have not been explored in the previous training stages, while retaining the well-learned features. Throughout the multistage training process, we implement a *feature-aware negative sampling* strategy designed to compel the model towards exploring unlearned features in earlier stages. Inspired by the observation that representations in contrastive learning tend to cluster according to the learned features [37, 40], at each stage, MCL selects negative samples for each anchor exclusively from the cluster it was assigned to in preceding stages. Because samples within the same cluster share similar learned features, these features cannot be reused to accomplish the contrastive learning objective: to discriminate the anchor from negative samples. Therefore, the model is necessitated to discover and incorporate previously unlearned features to fulfill the contrastive learning objective. After the multistage learning process, *cross-stage representation integration* is employed. Here the representations of data samples from all stages are concatenated to form the ultimate representations, ensuring that the well-learned features are retained.

In summary, the contributions of this work are three-fold. *First*, we propose a novel model-agnostic contrastive learning framework: Multistage Contrastive Learning that mitigates the severe issues of feature suppression commonly encountered in contrastive learning. *Second*, to the best of our knowledge, this is the first work to discuss and address the problem of feature suppression in both unimodal and multimodal contrastive learning. *Third*, we empirically demonstrate that the proposed MCL can be adapted to various contrastive learning settings and further boost their performance using different encoder backbones scaling from ResNet-18 to ViT-L-14. Notably, MCL demonstrates a significant improvement and boosts the average accuracy from 20.0 to 32.6 in the CLIP setting on the MMVP benchmark.

2 Related Works

2.1 Contrastive Learning

How to extract useful information from unlabeled data is an important question in machine learning [2]. Among all branches of methods, contrastive learning flourishes in recent years [23] and plays an important role in text-to-image generation [37–39] and multimodal large language models [32, 54, 58]. Contrastive learning aims to learn useful representations from unlabeled data by maximizing the agreement between different views of the data [14]. Recently, different variants of contrastive learning methods have been proposed [4, 6, 19, 36]. CPC [36] learns representations by predicting future samples in a sequence using an autoregressive model with a contrastive loss. MoCo [19] is designed to overcome the limitations of batch size in contrastive learning by introducing a dynamic dictionary with a queue and a moving-averaged encoder. SimCLR [6] is a simple

yet effective framework for contrastive learning of visual representations. SimCLR demonstrated that, with sufficiently large batch sizes, it is possible to learn powerful representations without needing specialized architectures or a memory bank. MoCo-v2 [9] builds upon the original MoCo by incorporating several improvements from SimCLR. Negative-free contrastive learning [10, 17] further simplifies contrastive learning by removing the requirement of explicit negative samples. Although contrastive learning achieves promising performance in many fields [26, 27, 44, 48, 50, 55], it cannot guarantee all semantically relevant features are learned when multiple features exist [3, 7, 47, 52, 53].

2.2 Feature Suppression in Contrastive Learning

The feature suppression phenomenon has been first empirically observed by Chen *et al*. [7]. Subsequently, Robinson *et al*. [40] formally brings up this problem as feature suppression, and shows that simply minimizing the InfoNCE loss cannot avoid feature suppression. Both Robinson *et al*. [40] and Kukleva *et al*. [25] observed that the temperature parameter affects the trade-off of which features are learned and which features are suppressed. Xiao *et al*. [52] have discovered that certain augmentations used to generate positive samples might destroy the feature information, hence hindering the learning process of corresponding features. Assran *et al*. [1] point out that feature suppression might be caused by the hidden prior distribution bias in contrastive learning. Xue *et al*. [53] demonstrate that the simplicity bias of stochastic gradient descent is one of the factors. Not only in the above unimodal setting, Bleeker *et al*. [3] first studies this problem in the multimodal setting. Tong *et al*. [47] have observed that images with different semantics have unreasonably high similarities in CLIP [37] embedding space, which is also highly related to feature suppression in multimodal contrastive learning.

Few methods specifically address the challenge of feature suppression in contrastive learning. Robinson *et al*. [40] proposed a technique aimed at eliminating whichever features distinguish the positive sample from negative samples in the embedding space. Similarly, Tamkin *et al*. [45] applied a comparable strategy but targeted the input space. Both approaches are based on adversarial training, which does not ensure the preservation of previously well-learned features. In contrast, our approach diverges by sequentially learning new features stage by stage without compromising the integrity of already learned features. Bleeker *et al*. [3] mitigate the feature suppression problem by introducing an additional reconstruction loss, which is not feasible for large-scale settings such as CLIP due to the high computational cost. In contrast, our approach does not necessitate modifications to the original loss function or alterations to the base model, conserving computational resources and ensuring compatibility across different models. It is worth noting that, unlike the aforementioned methods and several other related approaches [8, 12, 16, 35, 43] that are confined to either unimodal or multimodal settings, our work stands as the first attempt to tackle feature suppression across both unimodal and multimodal contrastive learning.

3 Preliminaries

3.1 Self-Supervised Contrastive Learning

In contrastive learning, the core objective is to minimize the distance between positive pairs while maximizing the distance between negative pairs within the representation space. This objective compels the model to effectively distinguish positive pairs from their negative counterparts. Without loss of generality, here we consider the basic yet effective Noise Contrastive Estimation (NCE) [18] based contrastive learning model [33] as our backbone contrastive learning model. Given an anchor $\mathbf{x}$, its positive sample $\mathbf{x}^+$ and m negative samples $\{\mathbf{x}_i^-\}_{i=1}^m$, the model required to minimize the InfoNCE loss defined below:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},\mathbf{x}^+,\{\mathbf{x}_i^-\}_{i=1}^m}\left[-\log \frac{e^{s(\mathbf{z},\mathbf{z}^+)/\tau}}{e^{s(\mathbf{z},\mathbf{z}^+)/\tau} + \sum_{i=1}^m e^{s(\mathbf{z},\mathbf{z}_i^-)/\tau}}\right], \tag{1}$$

where $s(\cdot,\cdot)$ denotes the cosine similarity, τ represents the temperature, and $\mathbf{z}$, $\mathbf{z}^+$, and $\mathbf{z}^-$ are the corresponding embeddings of $\mathbf{x}$, $\mathbf{x}^+$, and $\mathbf{x}^-$. In unimodal contrastive learning, typically the positive samples are augmentations of the anchor, while in multimodal contrastive learning, the positive samples are usually data pairs with similar semantics as the anchor. The negative samples are randomly sampled data.

Table 1: Demonstration of feature suppression in unimodal and multimodal settings.

(a) Linear evaluation results of SimCLR and MoCo-v2 trained on Trifeature and CIFAR-MNIST (C-M). The feature suppression problem in SimCLR is severe. In Trifeature, SimCLR significantly ignores the shape information, and in CIFAR-MNIST, it almost completely neglects the CIFAR information. With MoCo-v2, the issue of feature suppression is less pronounced but still exists, considering the two datasets are very simple.

Feature	SimCLR	MoCo-v2
C-M(CIFAR)	0.10	0.77
C-M(MNIST)	0.99	0.98
Trifeature(Shape)	0.44	0.85
Trifeature(Texture)	0.92	0.99
Trifeature(Color)	1.00	1.00

(b) Performance of CLIP on the MMVP benchmark. The performance is low on most of the attributes. **O&D**: Orientation and Direction, **PSF**: Presence of Specific Features, **S&C**: State and Condition, **Q&C**: Quantity and Count, **P&R**: Positional and Relational Context, **C&A**: Color and Appearance, **S&P**: Structural and Physical Characteristics, **Texts**: Texts, **V&P**: Viewpoint and Perspective.

Attribute	Accuracy	Attribute	Accuracy
O&D	26.7	**C&A**	40.0
PSF	13.3	**S&P**	26.7
S&C	26.7	**Texts**	13.3
Q&C	6.7	**V&P**	20.0
P&R	6.7	Average	20.0

3.2 Feature Suppression

This section empirically illustrates the feature suppression phenomenon across both unimodal and multimodal contrastive learning settings.

In the unimodal setting, the ability of the encoder to capture specific feature information can be assessed through the linear evaluation accuracy of discriminating that feature. A high linear evaluation accuracy for a given feature suggests the encoder has successfully captured substantial information regarding that feature, and vice versa. We train ResNet-18 encoders using SimCLR [6] and MoCo-v2 [9] on two datasets (CIFAR-MNIST and Trifeature)[1] following Robinson *et al*. [40] and Chen *et al*. [7], to demonstrate the feature suppression phenomenon. The linear evaluation results for each feature across both datasets are shown in Tab. 1a. In CIFAR-MNIST, the linear evaluation accuracy for MNIST features is high but both SimCLR and MoCo-v2 show comparatively lower accuracy for CIFAR features. This discrepancy indicates MNIST features' predominance, with CIFAR information being notably overlooked—SimCLR, in particular, demonstrates this trend more pronouncedly. In Trifeature, a similar pattern emerges: both contrastive learning methods achieve high linear evaluation accuracy for texture and color, yet falter when it comes to shape. This divergence suggests that the encoders while the encoders sufficiently capture texture and color, they neglect shape information. Consequently, as illustrated in Fig. 1a, images with different shapes in Trifeature have high similarity in the SimCLR embedding space. This overlap significantly hampers the model's capacity to discern shapes, adversely affecting performance on downstream tasks reliant on shape differentiation.

Similarly, in the multimodal setting, our observations align with those as reported by Tong *et al*. [47]. Referencing Fig. 1b, we find that images, despite varying significantly in object orientation and direction, are represented with striking similarity in the CLIP embedding space. As a result, CLIP struggles to discern differences in object orientation and direction. This limitation can lead to orientation-based hallucinations in multimodal large language models that utilize CLIP as their vision encoder [47]. To evaluate this phenomenon, Tong *et al*. [47] introduce the MMVP benchmark. We evaluate the OpenAI ViT-L-14 CLIP model [37] with 224^2 resolution on the MMVP benchmark. The results, presented in Tab. 1b, show limited performance across a range of attributes, highlighting a severe feature suppression issue of the current CLIP model.

4 Multistage Contrastive Learning

4.1 Feature-aware Negative Sampling

In the initial phase of MCL, we train an encoder f_0 using the standard NCE objective as in Eq.1. For clarity, we specify that the subscript's first component refers to the sample index, while the latter signifies the stage index. Considering a dataset with M samples $\boldsymbol{X} = \{\mathbf{x}_i\}_{i=1}^{M}$, after training we obtain the dataset's encoded representations $\boldsymbol{Z}_0 = \{f_0(\mathbf{x}_i)\}_{i=1}^{M}$. Subsequently, we apply K-means clustering to $\boldsymbol{Z}_0$ and obtain the initial cluster assignments $\boldsymbol{Y}_0 = \{\mathbf{y}_{(i,0)}\}_{i=1}^{M}$. The MCL training process consists of N stages. At j^{th} stage, for a given sample $\mathbf{x}_i$,

[1] More detailed dataset description and settings can be found in Sec. 5.

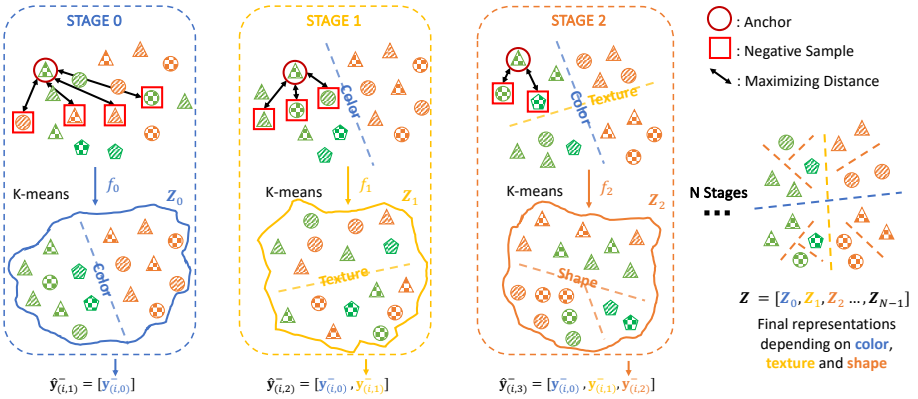


Fig. 2: Overview of the Multistage Contrastive Learning (MCL) Framework: Initially, the model is trained and its output representations are clustered. At each subsequent stage, cluster assignments from the previous stages are concatenated to derive a pseudo label. Throughout the training phase, negative samples are selected based on their matching pseudo label with the anchor. The final representations are the concatenation of representations in each stage. For simplicity, only three stages are shown here. The 'color', 'shape', and 'texture' here are used metaphorically to represent abstract features.

we define its pseudo label $\hat{\mathbf{y}}_{(i,j)}$ as the concatenation of its cluster assignments from all preceding stages, as follows:

$$\hat{\mathbf{y}}_{(i,j)} = [\mathbf{y}_{(i,0)}, \cdots, \mathbf{y}_{(i,j-1)}]. \tag{2}$$

When feature suppression happens, representations will be clustered by dominant features, since inputs with the same dominant features exhibit high similarity in representation space [40, 47]. Therefore, data samples with identical pseudo labels indicate they have similar dominant features learned in prior stages, as illustrated in Fig. 2. During the j^{th} stage, the encoder f_j is trained under a refined InfoNCE objective incorporating feature-aware negative sampling: the negative samples must have identical pseudo labels with the anchor. Formally, we define the optimization objective for f_j at j^{th} stage as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},\mathbf{x}^+,\{\mathbf{x}_i^-\}_{i=1}^m} \left[-\log \frac{e^{s(\mathbf{z},\mathbf{z}^+)/\tau}}{e^{s(\mathbf{z},\mathbf{z}^+)/\tau} + \sum_{i=1}^m \mathbb{1}_{\hat{\mathbf{y}}_j = \hat{\mathbf{y}}^-_{(i,j)}} e^{s(\mathbf{z},\mathbf{z}_i^-)/\tau}} \right], \tag{3}$$

where $\hat{\mathbf{y}}_j = [\mathbf{y}_0, \cdots, \mathbf{y}_{j-1}]$ represents the pseudo label of the anchor $\mathbf{x}$ at j^{th} stage; $\hat{\mathbf{y}}^-_{(i,j)} = [\mathbf{y}^-_{(i,0)}, \cdots, \mathbf{y}^-_{(i,j-1)}]$ denotes the pseudo label of the negative sample $\mathbf{x}_i^-$ at j^{th} stage; and $\mathbb{1}_{\hat{\mathbf{y}}_j = \hat{\mathbf{y}}^-_{(i,j)}} \in \{0, 1\}$ corresponds to an indicator function evaluating to 1 if and only if $\hat{\mathbf{y}}_j = \hat{\mathbf{y}}^-_{(i,j)}$. Feature-aware negative sampling ensures that the previously dominant features can not be re-utilized by the model to optimize the InfoNCE loss since the anchor and its negative samples share similar dominant

features learned in earlier stages. As such, the model has to identify and utilize features distinct from those previously dominant features. For example, if an encoder f initially learns color as a dominant feature in Trifeature, through feature-aware negative sampling, the subsequent encoder f' will be tasked with differentiating samples with the same color. This forces the contrastive model to explore alternative features such as texture or shape, rather than relying solely on color, as depicted in Fig. 2. Following the training of the j^{th} stage, we obtain the representations $\boldsymbol{Z}_j = \{f_j(\mathbf{x}_i)\}_{i=1}^M$ encoded by f_j, and the cluster assignments $\boldsymbol{Y}_j = \{\mathbf{y}_{(i,j)}\}_{i=1}^M$ for the feature-aware negative sampling in the next stage.

Notice that derived from Eq. 2, N stages of clustering with K clusters each lead to a potential total of K^N unique clusters. To ensure a meaningful clustering where clusters are approximately balanced and contain a sufficient number of samples, there's a mathematical constraint on the values of N and K:

$$K^N \leq \frac{M}{b}, \tag{4}$$

where M represents the total number of samples in the training dataset, and b denotes the batch size. This constraint guarantees that the resulting clusters each have a sufficient number of samples to form a batch, avoiding situations where a cluster contains fewer samples than the batch size, which would make it impractical for training purposes.

4.2 Cross-stage Representation Integration

Upon completing the training across all stages, we employ cross-stage representation integration to derive the final representation, which aims to preserve the information of well-learned features from each stage. Specifically, we element-wise concatenate the representations encoded by each trained encoder, resulting in comprehensive final representations for downstream tasks. The cross-stage representation integration is defined as:

$$\boldsymbol{Z} = \{[f_0(\mathbf{x}_i), \cdots, f_{N-1}(\mathbf{x}_i)]\}_{i=1}^M. \tag{5}$$

It is worth noting that more cross-stage representation integration methods can be further tailored to accommodate different downstream tasks in the future. In this work, we use a simple concatenation as a preliminary baseline to demonstrate MCL's main concept. In addition, since our framework does not change the backbone model, it can be seamlessly integrated with any NCE-based contrastive learning method.

5 Experiments

5.1 Datasets

Trifeature. Trifeature [21] is an image dataset, where each image has three independent features: color, texture and shape each taking 10 values. For each

combination of the three features there are 100 samples, in which the position and rotation of the object are random. The three downstream tasks are to classify the color ($C = 10$ classes), texture ($C = 10$ classes), and shape ($C = 10$ classes).
CIFAR-MNIST (C-M). CIFAR-MNIST consists of channel-wise concatenation of the CIFAR-10 [24] image and the MNIST [28] image, following Chen *et al.* [7]. Each image has four channels: three from the CIFAR-10 and one from the MNIST. As the images are randomly sampled from the two datasets in concatenation, the CIFAR-10 class and the MNIST class can be considered as two independent features. The two downstream tasks are to predict the CIFAR-10 class ($C = 10$ classes) and the MNIST class ($C = 10$ classes).
CelebA. CelebFaces Attributes Dataset (CelebA) [34] is a large-scale face attributes dataset, where each image has 40 attribute annotations. We take three attributes: black hair, male, and smiling. We resampled the dataset to make these three attributes independent of each other, which can be considered as three independent features. After resampling the dataset has more than 40k images in total. The three downstream tasks are to predict whether the celebrity has black hair ($C = 2$ classes), whether the celebrity is smiling ($C = 2$ classes), and whether the celebrity is male ($C = 2$ classes).

5.2 Baselines

IFM. Implicit Feature Modification (IFM) [40] aims to mitigate feature suppression by adaptive modifying samples to remove whichever features are used to discriminate a particular positive pair from negatives in feature space.
FD. Feature Dropout (FD) [45] mitigates feature suppression by adversarial perturbation on input space to break the features already used to discriminate a particular positive pair from negatives, forcing the model to learn new features.
TS. Temperature Schedules (TS) [25] aims to improve the contrastive learning performance on long-tail data by dynamically scheduling the temperature parameter along the training process. We consider it as a baseline here since temperature has a strong impact on which features are learned.

5.3 Evaluation

Linear Evaluation Protocol. The well-accepted linear evaluation protocol [6] is used to evaluate the quality of learned representations in the unimodal contrastive learning setting. Specifically, after the contrastive training process, the trained encoder is fixed and the projection head is discarded. A linear softmax classifier is trained on top of the trained encoder for each label. If the classification accuracy on the label related to one feature is high, it means the encoder encodes sufficient information about that feature, and vice versa.
MMVP. The Multimodal Visual Patterns (MMVP) benchmark [47] is used for the multimodal contrastive learning setting. It challenges the CLIP [37] model to accurately match images with the corresponding text statements (*e.g.*, "a rabbit facing left" and "a rabbit facing right") using the image-text similarity

computed on learned representations and evaluates the pairing accuracy. Sourced from ImageNet [41] and LAION-Aesthetics [42], the MMVP dataset comprises image pairs that exhibit high similarity in the CLIP embedding space but possess distinctly different semantic features in nine attributes (*e.g.*, orientation and direction, color and appearance). Lower performance on the MMVP benchmark indicates a more severe issue of feature suppression.

5.4 Mitigating Feature Suppression in the Unimodal Setting

First, we compare our proposed MCL with IFM, TS, FD, and vanilla SimCLR on Trifeature, CelebA, and CIFAR-MNIST datasets.

Experiment Setting. We use SimCLR as the backbone contrastive learning method for MCL and all the baselines. For CIFAR-MNIST, we adapt ResNet-18 to accommodate smaller input sizes, following the CIFAR-10 configuration described by He *et al*. [20]. We choose the best temperature value from $\{0.1, 0.25, 0.5\}$ for MCL, IFM, FD, and vanilla SimCLR, and set the temperature schedule range for TS to $[0.1, 1]$ as its default setting in comparison. For MCL, we train for 3 stages, and 200 epochs in each stage. The number of clusters for K-means is set to 5. For IFM and FD, we train for 200 epochs. For TS, we train for 600 epochs since it requires a long training time in adherence to the requirements of the original setting. We use the default settings for all the baselines and backbone models unless otherwise specified.

Experimental Results. As shown in Tab. 2, MCL achieves non-trivial improvements compared with the baselines in almost all the tasks. The performance gain is especially prominent in CIFAR-MNIST and Trifeature, where the feature suppression is severe. In CIFAR-MNIST, the CIFAR feature is entirely dominated by the MNIST feature. Whereas, with our MCL framework, the linear evaluation accuracy of the MNIST is largely improved. Meanwhile, the already well-learned features in vanilla SimCLR are maintained in MCL in all the settings, which indicates MCL can learn the previously ignored features without forgetting the already well-learned features. Though the baselines may show improvements on certain tasks, they experience performance declines on others, leading to an overall performance that is inferior to the vanilla SimCLR. To illustrate MCL's ability to prioritize different key features across stages, we present the three samples most similar to the anchor for each stage. The common characteristics of these top 3 samples highlight the model's shifting focus throughout the learning process. For example, the similarity in shape between the anchor and the top 3 samples in Stage 2 indicates the model's concentration on shape at this stage. From Fig. 3, we can observe the model evolves through stages, with a prioritization on color in Stage 0, texture in Stage 1, and shape in Stage 2.

Unlike CIFAR-MNIST and Trifeature, STL-10 [13] does not have explicitly identified semantic features. We incorporate MCL with both SimCLR and MoCo-v2 [9], use ResNet-50 [20] as the encoder, train on STL-10 dataset for 3 stages, 400 epochs for each stage. The results are shown in Fig. 4. MCL boosts the MoCo-v2 performance on STL-10 by more than two percent. This further validates the effectiveness of our proposed MCL framework.

Table 2: Linear evaluation accuracy of MCL and the baselines. Bold indicates the best performance.

	Trifeature			CelebA			CIFAR-MNIST		
	Shape	Texture	Color	Hair	Smiling	Gender	CIFAR	MNIST	Average
IFM	0.99	0.99	**1.00**	0.82	0.70	0.93	0.11	**0.99**	0.82
TS	0.93	**1.00**	**1.00**	0.62	0.65	0.87	0.09	**0.99**	0.77
FD	0.75	0.73	0.78	0.81	**0.89**	0.94	0.78	0.86	0.82
SimCLR	0.81	0.99	**1.00**	0.84	0.75	0.94	0.29	0.98	0.83
MCL	**1.00**	**1.00**	**1.00**	**0.85**	0.79	**0.95**	**0.87**	**0.99**	**0.93**

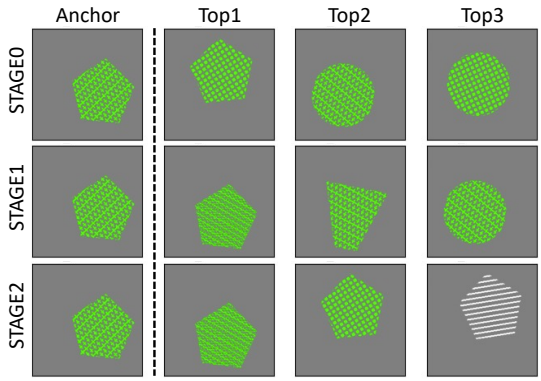


Fig. 3: In each stage, the top 3 most similar samples to the anchor, which demonstrates the model's shifting focus from color, texture, to shape across different stages of MCL.

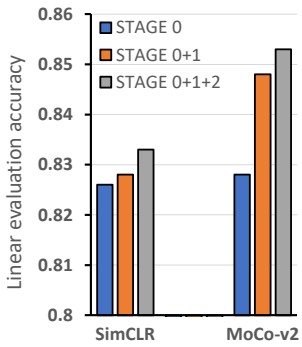


Fig. 4: Linear evaluation accuracy on STL-10 dataset by incorporating MCL with SimCLR and MoCo-v2.

5.5 Mitigating Feature Suppression in the Multimodal Setting

Experiment Setting. In this setting, we mainly adopt CLIP [37] to learn image representations on image-text pair datasets using contrastive learning. Specifically, we train two CLIP models in the MCL framework with different scales: one uses ResNet-50 as the image encoder, while the other utilizes ViT-L-14 [37]. Since the CLIP model requires training on hundreds of GPUs for a few days, instead of training from scratch, we tune the original CLIP model on Conceptual 12M (CC12M) [5], a large image-text pair dataset specifically designed for vision-and-language pre-training. We use the original OpenAI pre-trained weights *et al*. [37] as the initialization for each stage. Our implementation is based on OpenCLIP [11]. Since we mainly focus on image representation, we fix the text encoder in contrast to the approach in Zhai *et al*. [56]. Considering the features of the last few blocks in CLIP are more dominant [15] and avoid overfitting, we only leave the last 6 blocks trainable, with the other part of the image encoder fixed. For both two versions, we tune the models on CC12M for additional 3 stages, 10 epochs in each stage for the ResNet version, and 20 epochs for the ViT

Table 3: Accuracy of the MCL tuned CLIP models on the MMVP benchmark. Bold indicates the best performance across four stages. The STAGE0 model is the original OpenAI pre-trained model. The final MCL model result is obtained by using the concatenation of the representations across all four stages as the final representation.

	O&D	**PSF**	**S&C**	**Q&C**	**P&R**	**C&A**	**S&P**	**Texts**	**V&P**	Average
ResNet STAGE0	6.7	6.7	**46.7**	0.0	**13.3**	46.7	**33.3**	6.7	13.3	19.3
ResNet STAGE1	**13.3**	**13.3**	40.0	**20.0**	0.0	33.3	33.3	6.7	26.7	20.7
ResNet STAGE2	6.7	6.7	33.3	6.7	6.7	**73.3**	13.3	13.3	**33.3**	21.5
ResNet STAGE3	0.0	13.3	33.3	20.0	0.0	60.0	33.3	**26.7**	20.0	23.0
ResNet MCL	0.0	13.3	40.0	6.67	6.67	73.3	40.0	6.7	33.3	<u>24.4</u>
ViT STAGE0	**26.7**	13.3	26.7	6.7	6.7	40.0	26.7	**13.3**	20.0	20.0
ViT STAGE1	13.3	**33.3**	**66.7**	26.7	**13.3**	53.3	20.0	13.3	**26.7**	29.6
ViT STAGE2	0.0	13.3	46.7	**40.0**	6.7	53.3	20.0	13.3	20.0	23.7
ViT STAGE3	13.3	6.67	46.7	13.3	13.3	**66.7**	**40.0**	13.3	20.0	25.9
ViT MCL	6.67	20.0	73.3	13.3	13.3	80.0	46.7	13.3	26.7	<u>32.6</u>

version. We do K-means clustering on the image representation between each stage, and the number of clusters is set to 10. We use a batch size of 8192 and a warmup of 10% of the total steps. It is worth noting that although multiple stages are trained in our framework, the computation parameters will not increase much. For example, the total number of parameters in our four-stage ViT CLIP image encoder will only increase by 75% compared to a single vanilla CLIP. The parameters can be further reduced by training fewer layers or introducing parameter-efficient tuning, such as LoRA [22]. We leave this as future work.

Experimental Results. Table 3 presents the performance of the tuned CLIP models across different stages. For both the ResNet and ViT architectures, we observe improvements in later stages over earlier ones on tasks where initial performances were suboptimal. For example, ViT at Stage 1 outperforms its Stage 0 counterpart in **PSF**, ViT at Stage 2 exceeds the performance of both Stage 0 and Stage 1 in **Q&C**, and ViT at Stage 3 surpasses the earlier stages in **C&A**. These improvements underscore the models' evolving expertise in distinct features, enhancing their competence with specific attributes. Notice that each stage of the ViT model specializes in different attributes: ViT-STAGE0 in **O&D**, ViT-STAGE1 in **PSF**, **S&C**, and **V&P**, ViT-STAGE2 in **Q&C**, and ViT-STAGE3 in **C&A** and **S&P**.

Upon integrating representations from all stages, the ensemble model's average performance on the MMVP benchmark increases from 19.3 to 24.4 for the ResNet version, and from 20.0 to 32.6 for the ViT version. Notably, the final MCL model surpasses the peak performance seen in individual stages in certain attributes (*e.g.*, **S&C** and **C&A** for ViT), suggesting that it captures complementary features across stages. Interestingly, in certain attributes (*e.g.*, **O&D**), the MCL model does not achieve the best results compared to individual stages, suggesting room for improvement in our approach of cross-stage representation integration. This finding directs future research toward optimizing the process of combining

stage-specific insights, aiming to harness the full potential of the model's learned features.

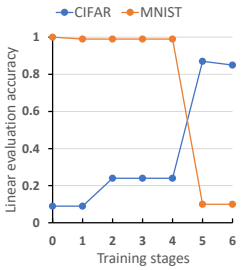
5.6 Discussion

In this section, we study the impact of training stage (N), number of clusters (K) in K-means, and temperature τ in our MCL framework. Unless noted otherwise, the experimental configurations adhere to those outlined in Sec. 5.4.

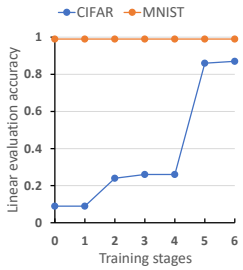
Learning Dynamics Across Stages. We explore the learning dynamics by increasing the number of training stages from one to seven. Given the constraint in Eq. (4), a larger number of stages (N) requires a smaller number of clusters (K). Therefore, we set $K = 2$ and the temperature $\tau = 0.25$. Initially, we assess the performance of the encoder at each stage independently, without cross-stage representation integration. As depicted in Fig. 5a, up to Stage 4, CIFAR features remain predominantly suppressed by MNIST features. Interestingly, at Stage 5, the model abruptly shifts to prioritize CIFAR features over MNIST, resulting in a sudden change rather than a gradual improvement in linear evaluation accuracy. Subsequently, we analyze the performance with cross-stage representation integration across N stages, illustrated in Fig. 5b. This process effectively preserves the initially learned MNIST features. Without cross-stage integration, there is a noticeable drop in MNIST feature performance from Stage 4 to 5. However, with integration, the performance on MNIST remains stable. From Stage 5 onwards, the model exhibits minimal performance variation indicating convergence.

Impact of Number of Clusters. We investigate how the number of clusters K in k-means clustering affects the learning process within the MCL framework. We train SimCLR in the MCL framework on CIFAR-MNIST for three stages, varying K among $\{2, 5, 10\}$ respectively and assess the linear evaluation accuracy on CIFAR after cross-stage representation integration of $N = \{0, 1, 2\}$ stages, given CIFAR features were notably suppressed in the vanilla SimCLR (STAGE0). The findings, depicted in Fig. 6, indicate that a higher K value enables the model to uncover previously suppressed features more rapidly, reducing the number of stages needed to achieve substantial performance gains. Conversely, a lower K value ($K = 2$) necessitates additional stages for similar outcomes, as evidenced in Fig. 5b. Nonetheless, increasing cluster count from 5 to 10 does not markedly improve performance beyond the third stage, suggesting a performance plateau. Further increasing K beyond this point does not yield additional benefits.

Robustness to Diverse Temperature Settings. Following [40], which highlight the pivotal role of temperature (τ) in influencing feature suppression within contrastive learning frameworks, we evaluate MCL's adaptability across varying temperature settings $\tau = \{0.1, 0.25, 0.5\}$. Results, as detailed in Tab. 4, indicate that MCL consistently enhances performance across all datasets, irrespective of the temperature setting employed. Notably, the most significant performance uplifts are observed under temperature conditions where the baseline SimCLR model exhibits pronounced feature suppression, such as $\tau = \{0.1, 0.5\}$ in Trifeature. This trend underscores MCL's capability to not only mitigate feature suppression but also fortify the robustness of SimCLR against temperature variations.



(a) Linear evaluation accuracy for each stage individually before cross-stage representation integration.



(b) Linear evaluation accuracy for training N stages after cross-stage representation integration.

Fig. 5: Linear evaluation results of MCL on CIFAR-MNIST for different training stages.

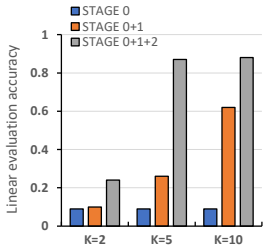


Fig. 6: Linear evaluation results of MCL on CIFAR using $K = \{2, 5, 10\}$, trained on CIFAR-MNIST for three stages respectively.

Table 4: The performance improvements by incorporating MCL with SimCLR under different temperature settings. Improvements are observed across all datasets, with more significant gains where feature suppression is more pronounced.

	$\tau = 0.1$		$\tau = 0.25$		$\tau = 0.5$	
	SimCLR	MCL	SimCLR	MCL	SimCLR	MCL
Trifeature(Shape)	0.66	1.00 (↑.34)	0.81	0.92 (↑.11)	0.44	0.75 (↑.31)
Trifeature(Texture)	0.91	1.00 (↑.09)	0.99	1.00 (↑.01)	0.92	0.99 (↑.07)
Trifeature(Color)	1.00	1.00 (↑.00)	1.00	1.00 (↑.00)	1.00	1.00 (↑.00)
CelebA(Hair)	0.84	0.85 (↑.01)	0.58	0.73 (↑.15)	0.57	0.69 (↑.12)
CelebA(Smiling)	0.75	0.79 (↑.04)	0.63	0.65 (↑.02)	0.63	0.66 (↑.03)
CelebA(Gender)	0.94	0.95 (↑.01)	0.72	0.87 (↑.15)	0.71	0.86 (↑.15)
C-M(CIFAR)	0.29	0.83 (↑.54)	0.10	0.87 (↑.77)	0.10	0.87 (↑.77)
C-M(MNIST)	0.98	0.98 (↑.00)	0.99	0.99 (↑.00)	0.99	0.99 (↑.00)

6 Conclusion

In this paper, we investigated the critical feature suppression in contrastive learning. Specifically, we introduced the Multistage Contrastive Learning (MCL) framework, a novel, model-agnostic framework. MCL employs a cross-stage negative sampling strategy that effectively promotes the learning of previously unlearned information at each stage. Meanwhile, MCL efficiently preserves well-learned features and mitigates degradation observed in prior works. The effectiveness of our approach is demonstrated through comprehensive analyses with commonly used baseline models on various datasets and settings, highlighting MCL's effectiveness and adaptability in both unimodal and multimodal contrastive learning.

Acknowledgement

This project was funded by the National Research Foundation Singapore under AI Singapore Programme (Award Number: AISG-GC-2019-001-2B and AISG2-TC-2022-004).

References

1. Assran, M., Balestriero, R., Duval, Q., Bordes, F., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., Ballas, N.: The hidden uniform cluster prior in self-supervised learning. arXiv preprint arXiv:2210.07277 (2022)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
3. Bleeker, M., Yates, A., de Rijke, M.: Reducing predictive feature suppression in resource-constrained contrastive image-caption retrieval. Transactions on Machine Learning Research (2023)
4. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
5. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3558–3568 (2021)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
7. Chen, T., Luo, C., Li, L.: Intriguing properties of contrastive losses. Advances in Neural Information Processing Systems **34**, 11834–11845 (2021)
8. Chen, T.S., Hung, W.C., Tseng, H.Y., Chien, S.Y., Yang, M.H.: Incremental false negative detection for contrastive learning. arXiv preprint arXiv:2106.03719 (2021)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning (2020)
10. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
11. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2818–2829 (2023)
12. Chu, T., Tong, S., Ding, T., Dai, X., Haeffele, B.D., Vidal, R., Ma, Y.: Image clustering via the principle of rate reduction in the age of pretrained models. arXiv preprint arXiv:2306.05272 (2023)
13. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
14. Federici, M., Dutta, A., Forré, P., Kushman, N., Akata, Z.: Learning robust representations via multi-view information bottleneck. arXiv preprint arXiv:2002.07017 (2020)

15. Gandelsman, Y., Efros, A.A., Steinhardt, J.: Interpreting clip's image representation via text-based decomposition. arXiv preprint arXiv:2310.05916 (2023)
16. Ge, S., Mishra, S., Li, C.L., Wang, H., Jacobs, D.: Robust contrastive learning using negative samples with diminished semantics. Advances in Neural Information Processing Systems **34**, 27356–27368 (2021)
17. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
18. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 297–304. JMLR Workshop and Conference Proceedings (2010)
19. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
21. Hermann, K., Lampinen, A.: What shapes feature representations? exploring datasets, architectures, and training. Advances in Neural Information Processing Systems **33**, 9995–10006 (2020)
22. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
23. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
24. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Toronto, ON, Canada (2009)
25. Kukleva, A., Böhle, M., Schiele, B., Kuehne, H., Rupprecht, C.: Temperature schedules for self-supervised contrastive methods on long-tail data. arXiv preprint arXiv:2303.13664 (2023)
26. Lan, X., Ng, D., Hong, S., Feng, M.: Intra-inter subject self-supervised learning for multivariate cardiac signals. Proceedings of the AAAI Conference on Artificial Intelligence **36**(4), 4532–4540 (Jun 2022). https://doi.org/10.1609/aaai.v36i4.20376, https://ojs.aaai.org/index.php/AAAI/article/view/20376
27. Lan, X., Yan, H., Hong, S., Feng, M.: Towards enhancing time series contrastive learning: A dynamic bad pair mining approach. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=K2c04ulKXn
28. LeCun, Y., Cortes, C., Burges, C.: Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010)
29. Leng, S., Zhang, H., Chen, G., Li, X., Lu, S., Miao, C., Bing, L.: Mitigating object hallucinations in large vision-language models through visual contrastive decoding. arXiv preprint arXiv:2311.16922 (2023)
30. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
31. Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Mitigating hallucination in large multi-modal models via robust instruction tuning. In: The Twelfth International Conference on Learning Representations (2023)

32. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
33. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE transactions on knowledge and data engineering **35**(1), 857–876 (2021)
34. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset. Retrieved August **15**(2018), 11 (2018)
35. Mishra, S., Shah, A., Bansal, A., Anjaria, J., Choi, J., Shrivastava, A., Sharma, A., Jacobs, D.: Learning visual representations for transfer learning by suppressing texture. arXiv preprint arXiv:2011.01901 (2020)
36. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
38. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
39. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021)
40. Robinson, J., Sun, L., Yu, K., Batmanghelich, K., Jegelka, S., Sra, S.: Can contrastive learning avoid shortcut solutions? Advances in neural information processing systems **34**, 4974–4986 (2021)
41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)
42. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)
43. Shah, A., Sra, S., Chellappa, R., Cherian, A.: Max-margin contrastive learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8220–8230 (2022)
44. Shvetsova, N., Chen, B., Rouditchenko, A., Thomas, S., Kingsbury, B., Feris, R.S., Harwath, D., Glass, J., Kuehne, H.: Everything at once-multi-modal fusion transformer for video retrieval. In: Proceedings of the ieee/cvf conference on computer vision and pattern recognition. pp. 20020–20029 (2022)
45. Tamkin, A., Glasgow, M., He, X., Goodman, N.: Feature dropout: Revisiting the role of augmentations in contrastive learning. arXiv preprint arXiv:2212.08378 (2022)
46. Tian, Y., Henaff, O.J., Van den Oord, A.: Divide and contrast: Self-supervised learning from uncurated data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10063–10074 (2021)
47. Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., Xie, S.: Eyes wide shut? exploring the visual shortcomings of multimodal llms. arXiv preprint arXiv:2401.06209 (2024)
48. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Van Gool, L.: Unsupervised semantic segmentation by contrasting object mask proposals. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10052–10062 (2021)

49. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: Proceedings of the 26th annual international conference on machine learning. pp. 1073–1080 (2009)
50. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. arXiv preprint arXiv:2202.01575 (2022)
51. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International conference on machine learning. pp. 23965–23998. PMLR (2022)
52. Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What should not be contrastive in contrastive learning. arXiv preprint arXiv:2008.05659 (2020)
53. Xue, Y., Joshi, S., Gan, E., Chen, P.Y., Mirzasoleiman, B.: Which features are learnt by contrastive learning? on the role of simplicity bias in class collapse and feature suppression. arXiv preprint arXiv:2305.16536 (2023)
54. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. arXiv preprint arXiv:2306.13549 (2023)
55. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8980–8987 (2022)
56. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: Lit: Zero-shot transfer with locked-image text tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18123–18133 (2022)
57. Zhang, B., Zhang, P., Dong, X., Zang, Y., Wang, J.: Long-clip: Unlocking the long-text capability of clip. arXiv preprint arXiv:2403.15378 (2024)
58. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)

A Appendix

In the appendix, we provide further details and additional experimental results that complement the main text. The contents include:

Appendix A.1 . Implementation details of *feature-aware negative sampling*.

Appendix A.2 . Learned features hinder the model from acquiring new features.

Appendix A.3 . Train SimCLR using MCL on ImageNet.

Appendix A.4 . Ensemble CLIP models trained using MCL with Model Soup.

Appendix A.5 . The computational cost of MCL.

Appendix A.6 . Ablation study for the number of trainable blocks when training ViT CLIP models using MCL.

Appendix A.7 . Analysis of the clustering results across different stages of MCL.

A.1 Details of Feature-Aware Negative Sampling

The *feature-aware negative sampling* strategy necessitates that an anchor and its negative samples belong to the same cluster, as determined in the previous stages. To effectively organize training samples according to their pseudo labels obtained between stages, we utilize a custom batch sampler. The implementation details of the batch sampler are shown in Algorithm 1.

Algorithm 1 Feature-Aware Negative Sampling for MCL

Input: Dataset D with pairs of data and pseudo labels $\{(x_i, y_i)\}_{i=1}^{M}$, number of unique pseudo labels $C = |\{y_i\}_{i=1}^{M}|$
Group D into a two-dimensional array `G` based on pseudo labels y, such that $\texttt{G}[c] = \{(x_i, y_i) \in D | y_i = c\}$
for batch index $j \leftarrow 0$ **to** ∞ **do**
    Select group index $k \leftarrow j \mod C$
    if all samples in $\texttt{G}[k]$ are traversed **then**
        continue
    end if
    Create batch from untraversed samples in $\texttt{G}[k]$
    if all samples in `G` are traversed **then**
        break
    end if
end for

A.2 Learned Features Hinder the Learning Process

We conduct experiments to demonstrate that the learned features can impede the model's ability to acquire new features. Specifically, at each stage, we initialize the model with parameters from the previous stage instead of training from scratch.

We train SimCLR with ResNet18 as the encoder on the Trifeature dataset. We use the $\tau = 0.5$ setting as described in Section 5.6 in the main paper. As shown in Tab. 5, this model makes almost no progress from Stage 1 to Stage 2, and its final performance (0.42/0.97/1.00) is markedly inferior to that of our proposed method (0.75/0.99/1.00). This can be attributed to the design of MCL, which does not require the retention of previously learned properties at each stage, thus allowing it to freely learn new features without constraints.

Table 5: Linear evaluation results on shape, texture, and color respectively using naive inheritance and MCL.

	Stage0	Stage1	Stage2
Inheritance	0.44/0.92/1.00	0.40/0.97/1.00	0.42/0.97/1.00
MCL	0.44/0.92/1.00	0.72/0.97/1.00	0.75/0.99/1.00

A.3 Train SimCLR Using MCL on ImageNet

To further validate the effectiveness of MCL on the large unimodal dataset, we train SimCLR with ResNet34 as the encoder on ImageNet. We use a batch size of 8192 and train for 100 epochs at each stage for 3 stages. As shown in Tab. 6, the experiment results demonstrate a notable improvement.

Table 6: Linear evaluation accuracy of incorporating MCL with SimCLR on ImageNet.

	Stage0	Stage0+1	Stage0+1+2
MCL	0.456	0.477	0.482

A.4 Ensemble ViT CLIP Models with Model Soup

We conduct experiments to compare MCL fine-tuned models with the standard OpenAI pretrained model, maintaining identical parameter sizes. Specifically, we utilize Model Soup [51], an effective and efficient ensemble approach that averages the weights of multiple fine-tuned models. We applied Model Soup to ensemble the fine-tuned ViT CLIP models across four stages, as described in Section 5.4 of the main text. As illustrated in Tab. 7, the MCL Model Soup achieves an average score of 29.6. Although this represents a slight decrease in performance compared to the simple concatenation method discussed in our main text, it still substantially surpasses the baseline average score of 20.0, with the same parameter size. We also test the MCL Model Soup on text-to-image and

image-to-text retrieval tasks, following the settings in Zhang *et al*. [57]. As shown in Tab. 8, the MCL Model Soup achieves a notable performance improvement compared to the OpenAI baseline. The results underscore the robustness and effectiveness of the MCL framework.

Table 7: Comparative analysis of ViT CLIP models fine-tuned with MCL on the MMVP benchmark. "OpenAI" denotes the baseline performance using the OpenAI pretrained model. "MCL Concat" represents the performance after applying MCL and using concatenated representations from all stages. MCL Model Soup (denoted as "MCL MS") illustrates the performance of the ensemble model created by Model Soup.

	O&D	**PSF**	**S&C**	**Q&C**	**P&R**	**C&A**	**S&P**	**Texts**	**V&P**	Average
OpenAI	26.7	13.3	26.7	6.7	6.7	40.0	26.7	13.3	20.0	20.0
MCL Concat	6.7	20.0	73.3	13.3	13.3	80.0	46.7	13.3	26.7	32.6
MCL MS	13.3	26.7	46.7	20.0	6.7	60.0	26.7	33.3	33.3	29.6

Table 8: Performance of MCL Model Soup. Tasks include text-to-image (T2I) retrieval and image-to-text (I2T) retrieval on 5k COCO validation set and 30k Flickr30k dataset. We use top-1, top-5, and top-10 Recall (R@1, R@5, R@10) as the evaluation metrics.

	COCO						Flickr30k					
	Image-to-Text			Text-to-Image			Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
OpenAI	56.1	79.5	86.8	35.4	60.1	70.2	48.5	72.6	80.8	28.0	49.3	58.7
MCL MS	60.4	82.4	89.5	42.2	66.9	76.2	52.7	76.6	84.4	36.5	59.0	68.1

A.5 The Computational Cost of MCL

The computational cost of MCL is detailed in Tab. 9, using our largest model in unimodal (as described in Appendix A.3) and multimodal settings for illustration. Notably, the fine-tuning cost of the CLIP model is approximately 3% of the OpenAI pre-training cost in TFLOPs, and the inference cost increases by only 75%.

A.6 Exploring the Number of Trainable Blocks

To assess the impact of varying the number of trainable blocks in the ViT architecture, we conduct experiments with the last 4, 6 (as detailed in the main text), and 8 transformer blocks set as trainable. Due to computational constraints, we focus on the first three stages of MCL. All other settings remained

Table 9: Computational complexity of MCL models. The training cost is evaluated by TFLOPs, Wall Clock Time (WCT), and TFLOPS. Inference cost is evaluated by relative complexity ratio (R) compared to backbone models (*i.e.* SimCLR ResNet34 and CLIP ViT). WCT is measured on 8 A100 GPUs.

	TFLOPs	WCT	TFLOPS	R
MCL SimCLR ResNet34	1.3×10^7	30h	1.6×10^2	300%
MCL CLIP ViT	1.8×10^8	32h	1.6×10^3	175%

consistent with those described in the main text. As summarized in Tab. 10, configuring 4 blocks as trainable results in an average score of 28.1, surpassing the 6-block configuration, which achieves a 27.4 average score. However, extending to 8 trainable blocks decreases the average score to 25.9. Despite these variations, MCL consistently improves performance compared to the vanilla OpenAI pretrained ViT CLIP, which scores an average of 20.0.

Table 10: Performance comparison of MCL-tuned ViT CLIP models with different numbers of trainable blocks on the MMVP benchmark. "S0" denotes the original OpenAI pre-trained ViT CLIP model. "S1" and "S2" denote Stage 1 and Stage 2. "MCL" represents the results derived from concatenating representations across the three stages.

Trainable Blocks		**O&D**	**PSF**	**S&C**	**Q&C**	**P&R**	**C&A**	**S&P**	**Texts**	**V&P**	Average
-	S0	26.7	13.3	26.7	6.7	6.7	40.0	26.7	13.3	20.0	20.0
4	S1	0.0	20.0	40.0	20.0	6.7	73.3	26.7	6.7	13.3	23.0
	S2	20.0	20.0	40.0	13.3	20.0	73.3	26.7	0.0	20.0	25.9
	MCL	13.3	33.3	60.0	20.0	13.3	73.3	13.3	6.7	20.0	28.1
6	S1	13.3	33.3	66.7	26.7	13.3	53.3	20.0	13.3	26.7	29.6
	S2	0.0	13.3	46.7	40.0	6.7	53.3	20.0	13.3	20.0	23.7
	MCL	6.7	13.3	53.3	26.7	13.3	66.7	40.0	0.0	26.7	27.4
8	S1	13.3	20.0	46.7	6.7	13.3	66.7	20.0	20.0	20.0	25.2
	S2	6.7	33.3	40.0	0.0	13.3	53.3	6.7	13.3	20.0	20.7
	MCL	13.3	33.3	33.3	13.3	6.7	60.0	26.7	20.0	26.7	25.9

A.7 Clustering Analysis

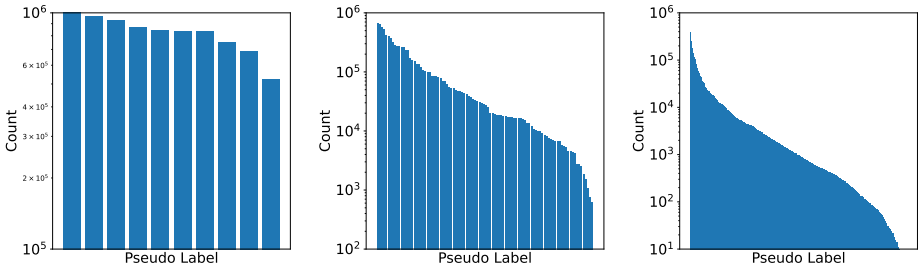
In this section, we explore the clustering dynamics observed during the training of ViT CLIP models using the MCL approach, as detailed in Section 5.4 of the main text. We begin by examining the Adjusted Mutual Information (AMI) [49] scores computed between the clustering outcomes of subsequent stages in the experiment. AMI scores range from 0, indicating no mutual information (independent clusterings), to 1, denoting identical clustering results.

Table 11: AMI scores between K-means clustering outcomes across different MCL stages. Low AMI scores between distinct stages suggest substantially different clustering results.

AMI	Stage 0	Stage 1	Stage 2
Stage 0	1.00	0.33	0.25
Stage 1	0.33	1.00	0.24
Stage 2	0.25	0.24	1.00

As illustrated in Tab. 11, the low AMI scores between distinct stages highlight the divergent clustering outcomes, underscoring that each stage learns a unique feature distribution. This divergence indicates the MCL framework's effectiveness in guiding the model to capture distinct, non-redundant features across stages.

Additionally, the distribution of pseudo labels for each stage, depicted in Fig. 7 (log scale), exhibits a long-tail distribution. This pattern aligns with characteristic distributions observed in large-scale datasets [1, 46], highlighting the natural variety of data captured at different stages of MCL.



(a) Stage 0 **(b)** Stage 1 **(c)** Stage 2

Fig. 7: Distribution of pseudo labels across stages 0, 1, and 2, demonstrating the variation in learned features.