Contrastive Dual Gating: Learning Sparse Features With Contrastive Learning

Jian Meng[*] Li Yang[*] Jinwoo Shin[†] Deliang Fan[*] Jae-sun Seo[*]
[*]Arizona State University, USA [†]KAIST, South Korea
[*]{jmeng15, lyang166, dfan, jaesun.seo}@asu.edu [†]{jinwoos}@kaist.ac.kr

Abstract

Contrastive learning (or its variants) has recently become a promising direction in the self-supervised learning domain, achieving similar performance as supervised learning with minimum fine-tuning. Despite the labeling efficiency, wide and large networks are required to achieve high accuracy, which incurs a high amount of computation and hinders the pragmatic merit of self-supervised learning. To effectively reduce the computation of insignificant features or channels, recent dynamic pruning algorithms for supervised learning employed auxiliary salience predictors. However, we found that such salience predictors cannot be easily trained when they are naïvely applied to contrastive learning from scratch. To address this issue, we propose contrastive dual gating (CDG), a novel dynamic pruning algorithm that skips the uninformative features during contrastive learning without hurting the trainability of the networks. We demonstrate the superiority of CDG with ResNet models for CIFAR-10, CIFAR-100, and ImageNet-100 datasets. Compared to our implementations of state-of-the-art dynamic pruning algorithms for self-supervised learning, CDG achieves up to 15% accuracy improvement for CIFAR-10 dataset with higher computation reduction.

1. Introduction

The success of the conventional *supervised learning* relies on the large-scale labeled dataset to minimize the loss and achieve high accuracy. However, manually annotating millions of data samples is labor-intensive and time-consuming. This promotes the *self-supervised learning* (SSL) to be an attractive solution, since artificial labels are used instead of human-annotated ones for training.

The state-of-the-art self-supervised learning frameworks, such as SimCLR [3] and MoCo [11], utilize the concept of contrastive learning (CL) [9] with wide and deep models to achieve comparable performance as the supervised training counterpart. Figure 1 shows the CIFAR-10 inference accuracy vs. the number of floating-point operations (FLOPs). By training from scratch, SimCLR [3] requires a model that is 4 times wider (ResNet-18 (4×)) to achieve similar accuracy as the baseline model trained with supervised learning (ResNet-18 (1×)). On the other hand, it is also difficult to achieve good accuracy with the compact model architecture (e.g., ResNet-20). The extraordinary computation cost necessitates efficient computation reduction techniques for self-supervised learning.

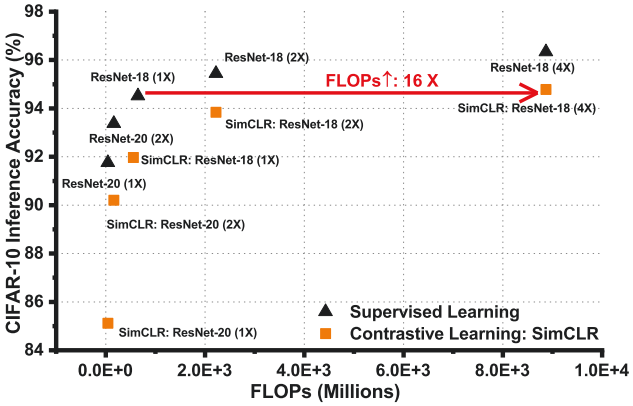

Figure 1. Inference accuracy of various ResNet models with supervised and self-supervised training [3] from scratch. After contrastive pre-training, models are fine-tuned on 50% of training set.

Under the context of supervised learning, network sparsification has been widely studied. Both static weight pruning [10, 21] and dynamic computation skipping [1, 8, 14, 16, 20] have achieved high accuracy with pruned architecture or sparse features. A recent work [2] reported the transferability of applying the lottery ticket hypothesis [7] to SSL for the downstream tasks. However, the requirements of self-supervised pretraining and iterative searching greatly limit the practicality of the algorithm. Sparsifying the SSL models that are trained from scratch is still largely unexplored, despite its importance.

To address this research gap, we investigate efficient dynamic sparse feature learning by training the model from scratch in a self-supervised fashion. Most of the prior works on dynamic computation reduction [1, 8, 16, 20] exploit the spatial sparsity by using an auxiliary *mini neu-*

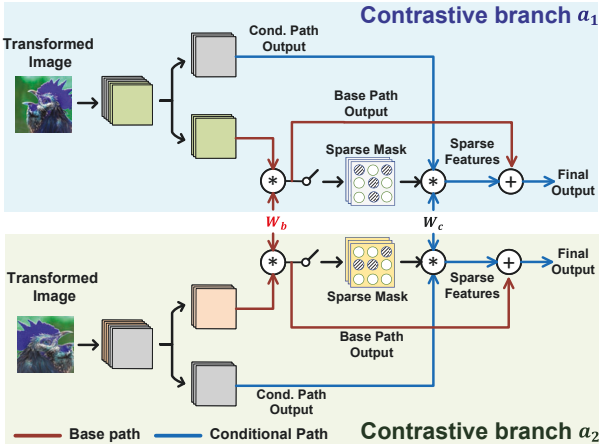

Figure 2. Overview of the proposed Contrastive Dual Gating (CDG) algorithm based on SimCLR [3] framework, which learns sparse feature in both constrative branches.

ral network (mini-NN) to determine the feature salience. Besides the extra computation cost of the mini-NN-based salience prediction, we found that it is problematic to use for contrastive learning due to significant accuracy degradation (see Section 5 for more details).

To resolve the issue, we propose *Contrastive Dual Gating* (CDG), a dynamic sparse feature learning algorithm for contrastive self-supervised learning. As opposed to the mini-NN-based salience prediction, CDG exploits spatial redundancy by using a spatial gating function. Different from channel gating network (CGNet) [14] presented for supervised learning, the proposed CDG algorithm for self-supervised learning exploits the spatial redundancies with full awareness of the saliency difference between the contrastive branches. As illustrated in Figure 2, CDG learns the sparse features in both contrastive branches during the unsupervised learning process. Furthermore, CDG can exploit the sparse features in both structured and unstructured manner. Aided by the efficient and optimized sparsification, CDG achieves high FLOPs reduction and high inference accuracy, without any auxiliary predictors. Overall, the main contributions of this work are:

- Contrary to dynamic pruning for supervised learning where mini-NN-based saliency prediction improved the overall performance, we show that such auxiliary predictor scheme leads to inferior accuracy in dynamic pruning for self-supervised learning.
- We present CDG, a new dynamic pruning algorithm with dual gating strategy, designed for contrastive self-supervised training with multiple recent contrastive learning frameworks.
- We evaluate CDG for ResNet models across multiple datasets, where CDG achieved up to 2.25× and 1.65× computation reduction for CIFAR-10/-100 [15] and ImageNet-100 datasets, respectively.

2. Related Work

2.1. Dynamic computation reduction

Learnable salience prediction. The inflation of the model sizes produces the different channel importance with the changing inputs. Several recent works proposed to use an additional mini-NN to predict the uninformative features or channels. Given the high-dimensional input, the salience predictor generates the low-dimensional salience vector, which will be used to formulate the binary feature masks during supervised training.

FBS [8] estimates the input channel importance by using an additional fully-connected (FC) layer followed by the ReLU activation function. Dynamic group convolution (DGC) [20] extends the design of FBS with more FC layers while deploying separate salience predictors in different output channel groups. Dynamic dual gating (DDG) [16] utilizes both convolution and fully-connected layers to exploit spatial and channel feature sparsity. The complex salience predictor designs improve the computation reduction with the cost of deteriorating the trainability of the model. DDG [16] requires the pretrained static model for initialization, even for the CIFAR-10 [15] dataset. None of the salience predictor designs have been studied for self-supervised learning.

Channel gating-based dynamic pruning. Channel gating networks (CGNet) [14] first executes a subset of input channels in every layer W_b (base path), the resultant partial sum will be strategically gated to determine the remaining computation of the convolution layer W_c (conditional path). Strong correlations have been reported between the base path outcomes and the final sum output, which means the uninformative features of the base path computation are also highly likely to be unimportant for the conditional path. The salience of the computation is evaluated based on the normalized base path output, where the features with large magnitude are deemed important and selected. Specifically, the base path output is formulated as:

$$Y_{base} = X_{base} * W_b \tag{1}$$

Subsequently, the computation decision $M_c \in \{0, 1\}$ for the conditional path W_c can be computed as:

$$M_c = \sigma_s(\mathbf{normal}(Y_{base}) - \tau), \tag{2}$$

where τ represents the learnable gating threshold. For better gradient approximation, the non-linear function σ_s consists of a non-linear activation function and a unified step function. The features with small magnitude (less than the threshold) will be gated, and the binary decision mask M_c will be applied to the conditional path computation. The final output of the convolution layer combines the dense base

path and the sparse conditional path:

$$Y_{i,j,k} = \begin{cases} \{Y_{base}\}_{i,j,k} & \text{if } \{M_c\}_{i,j,k} = 0 \\ \{Y_{base}\}_{i,j,k} + \{Y_{cond}\}_{i,j,k} & \text{if } \{M_c\}_{i,j,k} = 1 \end{cases} \tag{3}$$

As orthogonal to other methods that exploit the structured channel sparsity, CGNet focuses on fine-grained sparsity along the spatial axes. However, employing the unstructured sparsity in hardware could be cumbersome due to the fine-grained sparse indexes. As a result, the structured feature sparsity should also be carefully investigated.

2.2. Contrastive self-supervised learning

In contrast to learning the representative features with the labeled data, contrastive learning (CL) trains the model based on the latent contrastiveness of the high-dimensional features [12, 13]. With the similarity-based contrastive loss function [18], CL maximizes the agreement between similar samples while repelling mismatched representations from each other. The success of the contrastive loss enables the state-of-the-art methods to optimize the model by using gradient-based learning.

As a representative work, SimCLR [3] encodes two sets of augmented inputs (e.g., color jitter, Gaussian blur) with one single base encoder. Such end-to-end training frameworks exhibit less complexity but perform better with large models. However, the impact of the salience difference between the augmented features is still not clearly understood, which could largely impact the dynamic pruning performance for contrastive learning.

3. Learning Sparse Features with Contrastive Training

In this section, we discuss the optimal dynamic gating strategy for self-supervised sparse feature learning. We use ResNet-18 architecture as the default base encoder of SimCLR [3] contrastive learning framework.

3.1. Non-transferability of dynamic sparse masks

The pruning decision of CGNet [14] is formulated by evaluating the feature salience of the base path outcome. With supervised learning, all the intermediate features maps are originated from the clean input image. However, in the contrastive supervised learning scheme, the inputs of the base encoder are the transformed images for different contrastive branches. For SimCLR [3], the two transformed inputs are generated by the separate transformation operators from the same augmentation family $\mathcal{T}$. Therefore, the question arises: *Given the unique encoder network, will the base path feature salience be similar between the two augmented paths?* In other words, *can the pruning decisions be transferred between the two augmented features?*

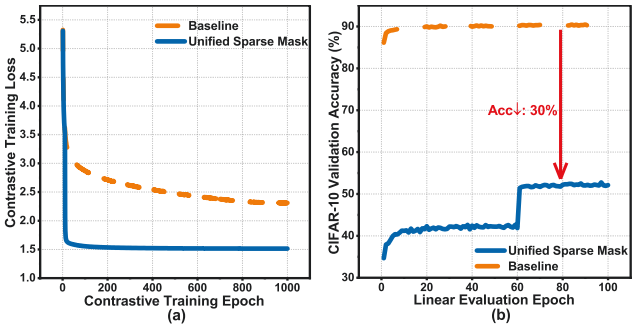

Figure 3. Broadcasting the computed sparse masks $M_c^{a_1}$ to both contrastive paths results in: (a) reduced contrastive training loss, and (b) defective generalizability with unsuccessful supervised linear evaluation.

To answer the above questions, we use CGNet [14] as the starting point but disable the channel shuffling to avoid the distortion of randomness. Given the two contrastive branches a_1 and a_2, we first compute $\mathbf{M_c^{a_1}}$ based on Eq. 2 with the base path input $X_{base}^{a_1}$, then broadcast $\mathbf{M_c^{a_1}}$ to the conditional path of both contrastive branches:

$$Y_{cond}^{a_1} = X_{cond}^{a_1} * W_c \cdot \mathbf{M_c^{a_1}}, \tag{4}$$

$$Y_{cond}^{a_2} = X_{cond}^{a_2} * W_c \cdot \mathbf{M_c^{a_1}}, \tag{5}$$

where

$$\mathbf{M_c^{a_1}} = \sigma_s(\mathbf{normal}(Y_{base}^{a_1}) - \tau) \tag{6}$$

We train a ResNet-18 encoder from scratch on the CIFAR-10 dataset. Due to the low resolution (32×32), the random Gaussian blur is excluded from the augmentation. Similar transformation methods have been verified in a previous implementation [6]. As shown in Figure 3(a), applying the identical dynamic pruning mask leads to a large reduction in contrastive loss from the baseline. However, the low contrastive pre-training loss cannot empower the subsequent supervised linear evaluation stage. The low accuracy is shown in Figure 3(b) implies that the feature extractor is defective due to unsuccessful contrastive learning.

With the absence of the geometric transformations, broadcasting the dynamic sparse masks across different contrastive paths can be considered as revealing similar spatial features during the conditional path convolution. After convolving with the shared conditional path W_c, the projected low-dimensional vectors tend to have high similarities, leading to decreased contrastive loss. Summarizing these empirical results, our main observations are:

A1: The unanimous data transformation operation $\mathcal{T}$ and the identical encoder f cannot guarantee the feature salience to be similar across different augmented branches. The observation of **A1** yields the following conclusion of dynamic pruning:

C1: Due to the distinct feature salience of contrastive learning, the pruning decision $\mathbf{M_c}$ is non-transferable between the contrastive branches.

Methods	Gating Groups	Cond. path Spars. (%)	Inference Acc. (%)
Baseline	-	-	89.16
Unified Gating	4	52.29	52.53
Dual Gating	4	71.88	87.67

Table 1. Comparison of different gating schemes for CIFAR-10 accuracy after contrastive pre-training and linear evaluation. Applying the discriminative dual gating during the contrastive learning significantly improves the model performance.

3.2. Dual gating for contrastive learning

Based on the conclusion **C1**, we employ separate pruning decisions for both contrastive branches. Specifically, given the base path outputs $Y_{base}^{a_1}$, $Y_{base}^{a_2}$, the dynamic sparse masks can be separately generated based on W_b:

$$\mathbf{M_c^{a_1}} = \sigma_s(\mathbf{normal}(X_{base}^{a_1} * W_b) - \tau) \quad (7)$$

$$\mathbf{M_c^{a_2}} = \sigma_s(\mathbf{normal}(X_{base}^{a_2} * W_b) - \tau) \quad (8)$$

Following the same training setup as Section 3.1, we apply separate sparse masks to both contrastive branches during training. During the subsequent linear evaluation, we only apply $\mathbf{M_c^{a_1}}$ to the frozen backbone model. As summarized in Table 1, the discriminative dual gating scheme improves both inference accuracy and conditional path sparsity by a significant margin. Conclusion **C1** confirms the necessity of applying distinct sparse masks to both contrastive branches whereas the salience difference between a_1 and a_2 requires a more quantitative investigation.

As shown in Figure 4, we compute the average shape-wise similarity $\mathbf{S_c}$ between $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$ along the channel dimension C. Since the sparse masks are binary, the element-wise similarity can only be "0" or "1". The global average mask similarity is computed by universally averaging the S_c of all the layers across all the training images of the CIFAR-10 dataset. Figure 4 shows the averaged similarity between the contrastive feature masks $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$ across the entire ResNet-18 model. At the start of training, the feature salience between the contrastive branches are similar ($\mathbf{S_c} > 0.6$). As the sparsity increases during training, the similarity reduces to 0.34. The magnification of the dissimilarity during contrastive training leads to the following conclusion:

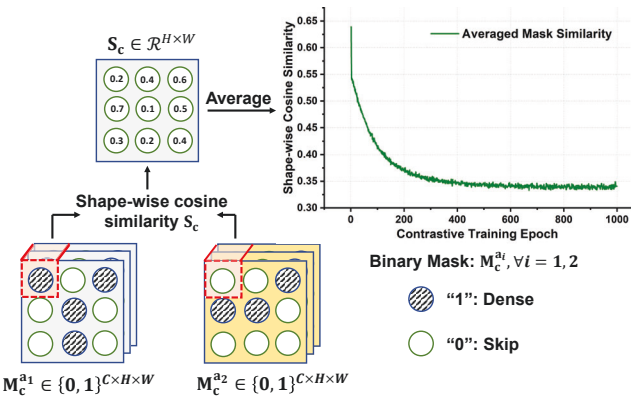

Figure 4. Shape-wise cosine similarity S_c between the contrastive masks $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$. With identical base path W_b of ResNet-18, $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$ become diverse from each other during training.

C2: Given the unanimous data transformation and identical base path selections W_b, contrastive training encourages the network f to highlight different contrastive features for better learning.

3.3. Unbiased contrastive grouping

To avoid the biased weight update, CGNet [14] diagonally selects the base path across the evenly-divided input/output gating groups. In the previous experiments of Section 3.1 and Section 3.2, we adopted the same computation strategy for contrastive learning. The conclusion **C2** suggests that the discriminative feature masks are beneficial for learning sparse features during contrastive training. The effectiveness of the distinct spatial feature selection motivates us to introduce separate base paths for different contrastive branches during training.

To that end, we investigate the impact of the overlapped base paths and different computation partitions between the two contrastive branches. With four gating groups ($G = 4$), Figure 5 depicts the different intersection percentages of the separate base paths, where $W_b^{a_1}$ and $W_b^{a_2}$ represents the base path weights of the two contrastive branches. We

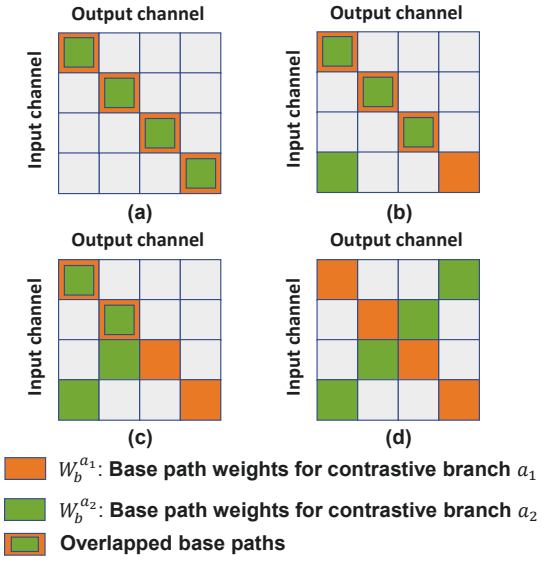

Figure 5. Dual gating with different overlapping percentages based on four gating groups: (a) Unified dual gating with 100% overlap, (b) 75% overlap, (b) 50% overlap, and (d) 0% with disjoint base paths.

Overlap	Gating Groups	Cond. Path Sparsity (%)	Inference Acc. (%)
Baseline	-	-	89.16
100%	4	71.88	87.67
75%	4	71.02	87.59
50%	4	70.60	87.12
Disjoint (0%)	4	**72.48**	**88.59**

Table 2. Comparison of different overlapping ratio between the contrastive base paths for CIFAR-10 accuracy after contrastive pre-training and linear evaluation.

first set $W_b^{a_1}$ along the diagonal, then vary the overlapping ratio with different selection of $W_b^{a_2}$. During the supervised linear evaluation, we only use $W_b^{a_1}$ as the base path.

Following the same contrastive training setup as Section 3.2, we train the ResNet-18 model for CIFAR-10 with different levels of overlapping, then evaluate the inference accuracy after the supervised linear evaluation. Table 2 summarizes the model performance that is trained by different base path selections. Noticeably, the pre-trained model reaches the lowest inference accuracy when the contrastive base paths are overlapped by 50% with each other. As illustrated in Figure 5(c), the first and second half of $W_b^{a_2}$ covers the same input channel groups while the remaining two output channel groups are ignored from the base path computation. Since the channel importance can be largely different, the inferior model performance with 50% channel overlapping signifies the importance of evenly distributing the computation to all the channel groups. Specifically, the repeated channels in base path makes the learning process tend to update the corresponding weights more frequently, and the inactive weights in the remaining channels will eventually cause the accuracy degradation. A similar discovery is also reported in [14].

On the contrary, when $W_b^{a_1}$ and $W_b^{a_2}$ are completely disjointed, the contrastively trained model achieves the best inference accuracy with only 0.5% degradation from the dense baseline. By selecting $W_b^{a_1}$, and $W_b^{a_2}$ along the disjoint diagonals, the base path computations are not subject any biased training, where different features among different channels are activated to enhance the contrastive learning. Based on these experiments and analysis, we have the following conclusion:

C3: Given the base encoder f, evenly activating the disjoint channels among the different contrastive paths will enhance the sparse feature learning during contrastive training.

4. Contrastive Dual Gating

Based on the aforementioned analysis, we present the *Contrastive Dual Gating* (CDG) algorithm for efficient dynamic sparse feature learning during contrastive self-supervised training. We illustrate the details of CDG in Algorithm 1. In this work, we mainly focus on the SimCLR [3] framework with two contrastive branches, referred as a_1 and a_2. During the forward pass of the contrastive training, CDG selects the contrastive base paths $W_b^{a_1}$ and $W_b^{a_2}$ along the diagonal and inverse-diagonal of the channel groups. The pruning masks $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$ are generated separately based on the learnable salience thresholds $\tau \in \mathbb{R}^C$, along with the gating function:

$$\mathbf{M_c^{a_1}} = \sigma_s(\mathbf{normal}(X_{base}^{a_1} * W_b^{a_1}) - \tau) \quad (9)$$

$$\mathbf{M_c^{a_2}} = \sigma_s(\mathbf{normal}(X_{base}^{a_2} * W_b^{a_2}) - \tau) \quad (10)$$

The resultant element-wise binary sparse feature masks govern whether the corresponding 3×3 convolution of the conditional path computation is skipped or not. As illustrated in Figure 5, the disjoint base paths of CDG allow the model to exploit the feature redundancy in a symmetric manner. The unbiased contrastive learning strategy satisfies our observation in Section 3.3. After the forward pass computation, we optimize τ via L_2 regularization based on the target sparsity value s:

$$\tilde{\mathcal{L}} = \mathcal{L}_{\text{NT-Xent}} + \lambda \sum_{i=1}^{L} ||s - \tau||_2, \quad (11)$$

where L represents the number of layers of the encoder model. Tunable parameter λ controls the penalty level of the regularization. During the backward pass, we adopt the gradient smoothing technique [14] to approximate the gradient of the non-differentiable gating function σ_s.

Algorithm 1 The proposed contrastive dual gating (CDG)

Require: **Encoder** f, **projector** g, **target sparsity** s, **gating groups** G, **feature group size** $\mathcal{K}$

Initialize Learnable salience threshold τ
for sampled minibatch X_k **do**
  for contrastive branch $a_i \in \{1, n\}$ **do**
    Draw data augmentation $t_{a_i} \sim \mathcal{T}$
    $X_k^{a_i} = t_{a_i}(X_k)$
    Get base path output: $Y_{base}^{a_i} = X_{base}^{a_i} * W_b^{a_i}$
    Compute feature salience
    if $|\mathcal{K}| > 1$ **then**
      $\mathcal{S}_{base}^{a_i} = \text{AvgPool}_{\dim(\mathcal{K})}(Y_{base}^{a_i}, size(\mathcal{K}))$
      $\mathcal{S}_{base}^{a_i} = \text{Repeat-Extend}(\mathcal{S}_{base}^{a_i})$
    else
      $\mathcal{S}_{base}^{a_i} = Y_{base}^{a_i}$
    end if
    Sparse conditional path convolution:
    $\mathbf{M_c^{a_i}} = \sigma_s(\mathbf{normal}(\mathcal{S}_{base}^{a_i}) - \tau)$
    $Y_{cond}^{a_i} = (X_{cond}^{a_i} * W_c^{a_i}) \cdot \mathbf{M_c^{a_i}}$
    Get final output
    $Y_{total}^{a_i} = Y_{base}^{a_i} + Y_{cond}^{a_i}$
  end for
end for

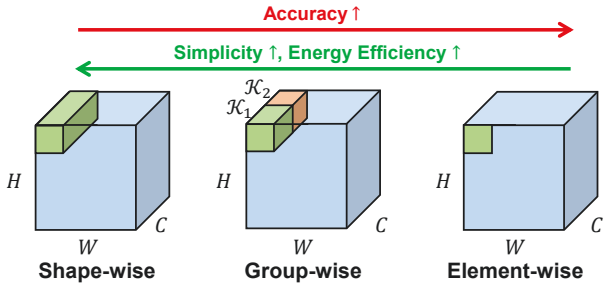

Figure 6. The granularity of structured CDG algorithm $\mathcal{K}_1$, $\mathcal{K}_2$ represents the two different groups with same size.

4.1. Structured contrastive dual gating

Compared to supervised training, the augmented contrastive inputs double the sparse indexes. Since both $\mathbf{M_c^{a_1}}$ and $\mathbf{M_c^{a_2}}$ have the same size as the output feature map, storing and processing such large fine-grained masks could introduce a large amount of memory and computation overhead in practice. Motivated by this, we introduce the coarse-grained sparsity on top of the CDG algorithm. Specifically, given the base path output $Y_{base}^{a_i}$, we first compute the average salience map $\mathcal{S}_{base}^{a_i}$ within each pre-defined group $\mathcal{K}$:

$$\mathcal{S}_{base}^{a_i} = \text{AvgPool}_{\dim(\mathcal{K})}(Y_{base}^{a_i}, size(\mathcal{K})) \tag{12}$$

The size of $\mathcal{K}$ can be either 2-D or 3-D, depending on the practical needs of the computation. Since the average pooling operation will cause the reduced size of $\mathcal{S}_{base}^{a_i}$, we duplicate each averaged value by $|\mathcal{K}|$ times to avoid the dimensionality mismatch. Compared to the fine-grained CDG, introducing the structured pruning strategy simplifies the sparse indexes by $|\mathcal{K}|$ times, leading to reduced computation complexity and memory cost. The performance of the sparsified contrastive learning model is highly dependent on the group size selection. The larger pruning granularity leads to the compendious sparse convolution, whereas the unitary features will also magnify the accuracy degradation [17]. To balance the model performance and inference efficiency on targeted hardware, we consider $\mathcal{K}$ as a tunable parameter and use the unified group size $|\mathcal{K}|$ across the entire network. In particular, given the base path output $Y_{base} \in \mathbb{R}^{C\times H\times W}$, we set the group size to $\mathcal{K} = C_g \times 1 \times 1$, where $1 < C_g < C$. Figure 6 depicts the group configuration of CDG.

5. Experimental Results

We present the experimental results of the proposed CDG algorithm for CIFAR-10, CIFAR-100, and ImageNet-100 datasets. We used 50% labeled data for supervised fine-tuning. Similar to prior works [6], all experiments are conducted by training the SimCLR-ResNet-18 [3] model from scratch. Additional results with larger models (e.g., ResNet-50) are reported in the supplementary materials.

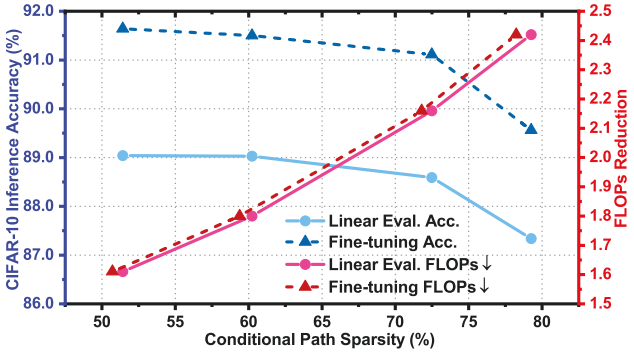

Figure 7. Unstructured conditional path sparsity vs. CIFAR-10 inference accuracy of ResNet-18 with 4 gating groups.

5.1. The impact of gating groups and model widths

The effectiveness of CDG is built upon the high correlation between the base path output and the final convolution results. Increasing the number of gating groups G reduces the amount of dense computation, whereas the insufficient base path partial sums will degrade the model performance. We evaluated model performance by changing the number of gating groups during the contrastive training. Given the number of gating groups G and conditional path sparsity η, the inference FLOPs reduction D_{FLOPs} is computed as:

$$D_{FLOPs} = \frac{1}{1/G + (1-\eta) \times (1-1/G)} \tag{13}$$

Table 3 summarizes the CIFAR-10 accuracy and unstructured conditional path sparsity after post-training linear evaluation. With only 0.5% accuracy degradation, the proposed CDG algorithm achieves 2.19× FLOPs reduction by only using 1/4 dense convolution as the base path computation. On the other hand, keeping 7/8 ($G = 8$) of the convolution operation sparse has conservative computation reduction to maintain the accuracy. Therefore, we use 4 gating groups for the ensuing experiments. Figure 7 illustrates the CIFAR-10 accuracy and computation reduction with different target s values and conditional path sparsity.

We also evaluated the proposed CDG algorithm based on ResNet-18 models with different widths. Table 4 summarizes the inference accuracy by training the model with CIFAR-100 and ImageNet-100 datasets from scratch. The first and last layer of the ResNet-18 model are adjusted accordingly for different input image sizes. After the contrastive pre-training, the resulting sparse models are fine-tuned with 50% labeled training set. Compared to the ResNet-18 baseline (1×) model, increasing the model width by 2× largely alleviates the accuracy degradation from the respective baseline model.

Following Algorithm 1, we exploit the structured feature sparsity based on the designed sparse group selections. Table 5 reports the inference accuracy by exploiting the structured spatial-wise sparsity with group size of $\mathcal{K} = 8 \times 1 \times 1$.

# of Gating Groups	Conditional Path Sparsity (%)	Inference Accuarcy (%)	Top-1 Accuracy Drop (%)	FLOPs Reduction
2	75.15	88.67	-0.42	1.60 ×
4	72.48	88.59	-0.50	2.19 ×
8	60.29	88.03	-1.06	1.83 ×

Table 3. Accuracy and FLOPs reduction of CDG with ResNet-18 (1×) on CIFAR-10 dataset with different number of gating groups.

Model	# of Gating Groups	Dataset	Conditional Path Sparsity (%)	Inference Acc. (%)	Top-1 Acc. Drop (%)	FLOPs Reduction
ResNet-18 (1×)	4	CIFAR-100	70.10	66.04	-1.74	2.11×
		ImageNet-100	50.05	76.82	-2.05	1.60×
ResNet-18 (2×)	4	CIFAR-100	73.32	67.62	-1.04	2.25×
		ImageNet-100	51.57	80.06	-1.14	1.65×

Table 4. Accuracy and FLOPs reduction of CDG on CIFAR-100 and ImageNet-100 datasets with different ResNet-18 widths.

Model	# of Gating Groups	Dataset	Conditional Path Sparsity (%)	Inference Acc. (%)	Top-1 Acc. Drop	FLOPs Reduction	Index Reduction
ResNet-18 (1×)	4	CIFAR-10	71.64	90.37	-0.89	2.16×	8×
		CIFAR-100	66.24	65.94	-1.84	1.98×	8×
		ImageNet-100	45.52	76.63	-2.24	1.53×	8×

Table 5. Structured contrastive dual gating for different datasets with the spatial group size $\mathcal{K} = 8 \times 1 \times 1$. After the sparse contrastive pre-training, the model is fine-tuned on 50% of the training labels.

Method	# of Gating Groups	Linear Eval. Inference Accuracy (%)	Fine-tuning Inference Accuracy (%)	FLOPs Reduction
This work (CDG_SimCLR)	4	**88.84**	**90.74**	**2.12**×
FBS_SimCLR	-	86.91	88.89	2.00×
DGC_SimCLR	4	73.10	81.77	2.11×
CGNet_SimCLR	4	87.40	89.26	2.09×

Table 6. With ResNet-18 (1×) for CIFAR-10 dataset, CDG outperforms our re-implementaiton of FBS [8], DGC [20], and CGNet [14] for SimCLR [3] (referred to as FBS_SimCLR, DGC_SimCLR, and CGNet_SimCLR, respectively) in both accuracy and FLOPs reduction.

Compared to unstructured pruning, the structured CDG algorithm achieves similar accuracy and computation reduction with $8\times$ index reduction.

5.2. Performance comparison

As discussed in Section 2, the typical feature salience predictors can be fully-connected layers [8, 20] or convolution layers [16, 19]. The increased complexity of the CNN-based salience prediction usually needs the pre-trained model as the starting point [16], which is not suitable for our case. Therefore, we mainly aim to evaluate CDG with the methods that can train the models from scratch, e.g., FBS [20], DGC [8] and CGNet [14]. Note that these works only reported the performance with supervised training. To evaluate the performance of the prior works' methods for self-supervised learning, we transferred the open-sourced dynamic pruning frameworks of [8,14,20] and re-implemented them with our self-supervised learning setup. As part of the model architecture, the auxiliary salience predictors will be shared between the contrastive paths then get updated in an end-to-end manner.

We evaluate the performance of the selected algorithms by training the ResNet-18 encoder on CIFAR-10 dataset from scratch, using multiple SSL frameworks including SimCLR [3], MoCoV2 [5], and SimSiam [4]. For the algorithms with group-wise computation [14, 20], we strictly follow the reported pruning strategy (e.g., sparsity schedule, number of output groups) during the self-supervised training. The pre-trained sparse encoder will be fine-tuned under both supervised linear evaluation and fine-tuning process. During the supervised fine-tuning phase, we use the

Method	# of Gating Groups	Linear Eval. Inference Acc.	Top-1 Acc. Drop (%)	FLOPS Reduction
This work (CDG_MoCo)	4	**90.58%**	**-0.86%**	2.00×
FBS_MoCo	-	88.29%	-3.15%	2.00×
DGC_MoCo	4	85.42%	-4.20%	2.11×
CGNet_MoCo	4	90.24%	-1.20%	2.04×
This work (CDG_SimSiam)	4	**89.04%**	**-0.32%**	**2.12**×
FBS_SimSiam	-	88.21%	-1.15%	2.00×
DGC_SimSiam	4	82.24%	-7.12%	2.11×
CGNet_SimSiam	4	88.65%	-0.71%	2.03×

Table 7. With ResNet-18 (1×) for CIFAR-10 dataset, CDG outperforms our re-implementaiton of FBS [20], DGC [8], and CGNet [14] with MoCoV2 [5] and SimSiam [4] SSL framework.

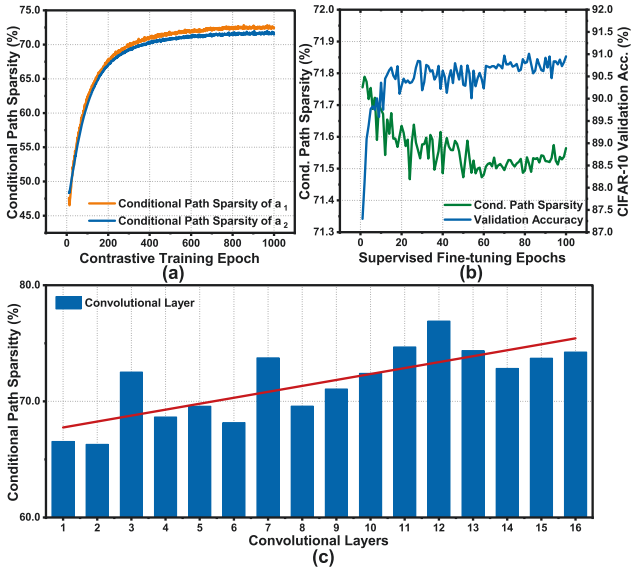

Figure 8. Conditional path sparsity during (a) sparse contrastive training and (b) supervised fine-tuning based on CIFAR-10 dataset. (c) The layer-wise sparsity of ResNet-18 after fine-tuning.

target (final) sparsity value to avoid the duplicate pruning. The model performance of methods that we implemented are summarized in Table 6 (SimCLR [3]) and Table 7 (MoCoV2 [5], and SimSiam [4]). With different SSL training schemes, the proposed CDG algorithm outperforms all implementations of prior dynamic pruning methods in both inference accuracy and computation reduction. Specifically, the proposed CDG algorithm outperforms FBS [20] and DGC [8] by up to 15.7% (SimCLR), 2.3% (MoCoV2), and 7.8% (SimSiam) CIFAR-10 accuracy.

One important observation from the results in Table 6 and Table 7 is the opposite trend on the effectiveness of complex salience predictors between supervised vs. self-supervised learning. DGC [20] employed salience predictors for different output groups with 2× deeper mini-NNs than FBS [8], which improved the overall performance beyond FBS and CGNet [8, 14] for supervised training. For self-supervised training, however, such intricate salience predictors are difficult to train from scratch, resulting in degraded inference accuracy.

5.3. Sparsity variation during contrastive learning

Given the shared regularization target s, the conditional path sparsity between two contrastive branches has minimum difference, as shown in Figure 8(a). The balanced sparsity exploitation represents successful unbiased training and sparsification. With an inherited base path $W_b^{a_1}$ and the learnable threshold τ, the subsequent fine-tuning process optimizes the model with the retained sparsity level, as shown in Figure 8(b). As shown in Figure 8(c), the latter layers of the model tend to achieve higher spatial sparsity, since the increase of the channel depth generates more redundant features.

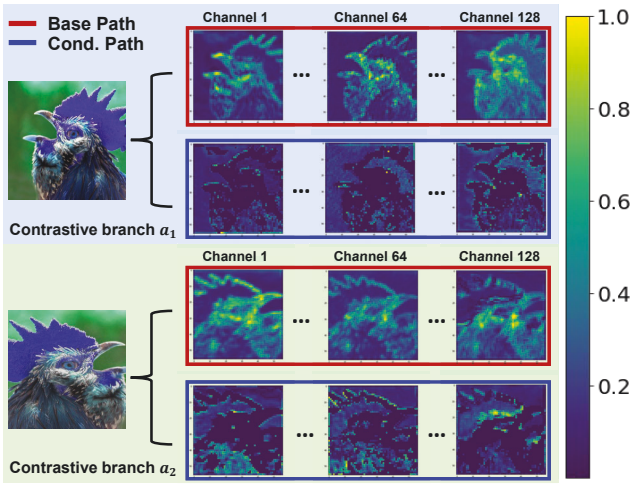

Figure 9. Feature map visualization of base path and conditional path along two different contrastive branches.

5.4. Sparse feature visualization

To validate the effectiveness of the proposed CDG algorithm, we visualize the second convolutional layer of the ResNet-18 (2×) model with ImageNet-100 input. As shown in Figure 9, for both contrastive branches a_1 and a_2, the base path (red rectangle) preserves the details with the dense computation while the sparse conditional path only keeps the important edges (e.g., the contour of the rooster's crest). As a result, the combined final output saves most of the information with considerable computation reduction.

6. Conclusion

In this work, we propose contrastive dual gating (CDG), a simple and novel dynamic pruning algorithm designed for contrastive self-supervised learning. As one of the first studies in this area, we analyze different sparse gating strategies with rigorous experiments. Based on the well-knit conclusions, we present the detailed algorithm design to exploit the feature redundancy in both fine-grained and structured manner. The proposed algorithms have been verified on multiple benchmark datasets and various SSL frameworks. Without any auxiliary salience predictors, the proposed CDG algorithm achieves up to 2.25× computation reduction for CIFAR-10 dataset, and outperforms our implementations of recent dynamic pruning algorithms. In addition, pruning the model in a structured manner elevates the practicality in terms of efficient hardware computing.

7. Acknowledgements

This work was in part supported by NSF grant 1652866, and the Center for Brain-inspired Computing (C-BRIC) in JUMP, an SRC program sponsored by DARPA.

References

[1] Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. Batch-shaping for learning conditional channel gated networks. *arXiv preprint arXiv:1907.06627*, 2019. 1

[2] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16306–16316, 2021. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607, 2020. 1, 2, 3, 5, 6, 7, 8, 10

[4] X. Chen et al. Exploring simple Siamese representation learning. In *IEEE/CVF CVPR*, 2021. 7, 8

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 7, 8

[6] Victor G. Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning. *arXiv preprint arXiv:2108.01775*, 2021. 3, 6

[7] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 1

[8] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic channel pruning: Feature boosting and suppression. *arXiv preprint arXiv:1810.05331*, 2018. 1, 2, 7, 8

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742, 2006. 1

[10] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015. 1

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020. 1

[12] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning (ICML)*, pages 4182–4192, 2020. 3

[13] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 3

[14] Weizhe Hua, Yuan Zhou, Christopher De Sa, Zhiru Zhang, and G Edward Suh. Channel gating neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 1, 2, 3, 4, 5, 7, 8

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[16] Fanrong Li, Gang Li, Xiangyu He, and Jian Cheng. Dynamic dual gating neural networks. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 5330–5339, 2021. 1, 2, 7

[17] Huizi Mao, Song Han, Jeff Pool, Wenshuo Li, Xingyu Liu, Yu Wang, and William J. Dally. Exploring the granularity of sparsity in convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. 6

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[19] Gil Shomron, Ron Banner, Moran Shkolnik, and Uri Weiser. Thanks for nothing: Predicting zero-valued activations with lightweight convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 234–250, 2020. 7

[20] Zhuo Su, Linpu Fang, Wenxiong Kang, Dewen Hu, Matti Pietikäinen, and Li Liu. Dynamic group convolution for accelerating convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 138–155, 2020. 1, 2, 7, 8

[21] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 2074–2082, 2016. 1