Aggregative Self-Supervised Feature Learning

Jiuwen Zhu
Institute of Computing Technology,
Chinese Academy of Sciences
zhujiuwen19g@ict.ac.cn

Yuexiang Li
Jarvis Lab, Tencent
vicyxli@tencent.com

S. Kevin Zhou
Institute of Computing Technology, Chinese Academy of Sciences
s.kevin.zhou@gmail.com

Abstract

Self-supervised learning (SSL) is an efficient approach that addresses the issue of annotation shortage. The key part in SSL is its proxy task that defines the supervisory signals and drives the learning toward effective feature representations. However, most SSL approaches usually focus on a single proxy task, which greatly limits the expressive power of the learned features and therefore deteriorates the network generalization capacity. In this regard, we hereby propose three strategies of ***aggregation*** *in terms of complementarity of various forms to boost the robustness of self-supervised learned features. In spatial context aggregative SSL, we contribute a heuristic SSL method that integrates two ad-hoc proxy tasks with spatial context complementarity, modeling global and local contextual features, respectively. We then propose a* ***principled framework*** *of multi-task aggregative self-supervised learning to form a unified representation, with an intent of exploiting feature complementarity among different tasks. Finally, in self-aggregative SSL, we propose to self-complement an existing proxy task with an auxiliary loss function based on a linear centered kernel alignment metric, which explicitly promotes the exploring of where are uncovered by the features learned from a proxy task at hand to further boost the modeling capability. Our extensive experiments on 2D natural image and 3D medical image classification tasks under limited annotation scenarios confirm that the proposed aggregation strategies successfully boost the classification accuracy.*

1. Introduction

Recently, self-supervised learning (SSL) [3, 11, 13, 10, 18, 22, 15] gains increasing attentions in the community as it attempts to loose the requirement of annotated data for neural networks by exploiting the rich information contained in unlabeled data. A conventional SSL approach starts with a formulated proxy task to encourage the learning of informative features from raw data. A multitude of proxy tasks, dealing with 2D natural images or 3D medical volumes, have been proposed, including grayscale image colorization [16], images rotation [8], Jigsaw puzzles [19, 23], BigBiGAN [7], SimCLR [2], Rubik's cube [28], Rubik's cube+ [27], Model Genesis [26], and D2D-CNNs [1].

Most SSL approaches usually focus on a single proxy task, which greatly limits the expressive power of the learned features and therefore decreases the network generalization capability. In the literature there are a few attempts of leveraging multiple tasks in SSL. Doersch and Zisserman [6] are the first to explore how to combine multiple self-supervised tasks. Chen et al. [4] introduce an adversarial training strategy into assembling self-supervised tasks. In medical imaging field, Rubik's cube+ [27] integrates three synergistic puzzles, including rearrangement, rotation, and partial masking, to enforce networks to learn a more robust representation. Model Genesis [25, 26] assembles several proxy tasks, including an array of image transformations, and designs a unitary self-supervised learning framework. Those approaches demonstrate that the integration of proxy tasks may strengthen the generalization of pre-trained networks and thus boost the performance of subsequent target tasks.

Albeit successful, such a task integration is usually derived from ad-hoc assumptions and *there is a lack of a principled way of aggregation*. In this paper, we attempt to bridge the gap by exploiting three different forms of complementarity. We first contribute a heuristic aggregation method by exploiting *spatial context complementarity*. It combines two ad-hoc proxy tasks that extracts global and local context information, respectively. Then, we systematically explore the *feature complementarity* between multiple SSL approaches and propose a greedy algorithm to

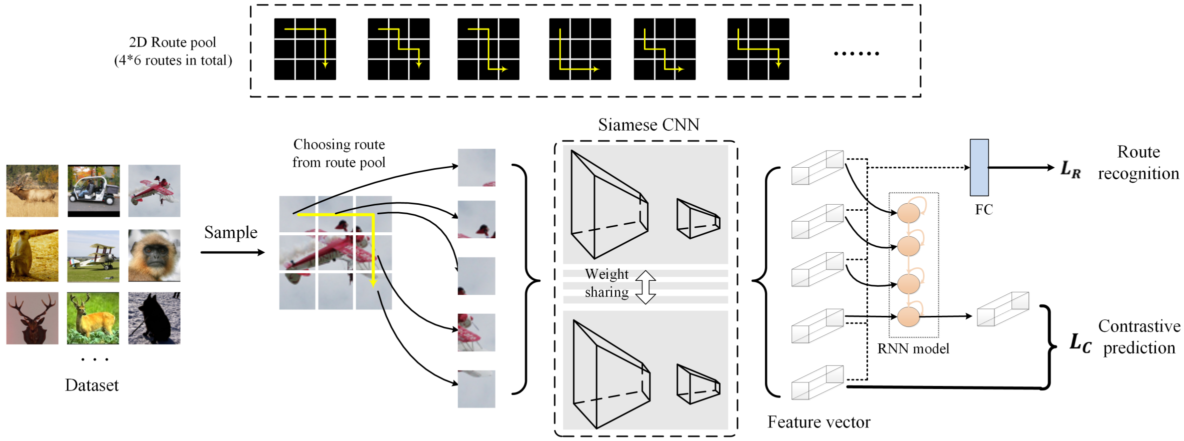

Figure 1. **The proposed spatial context aggregative self-supervised learning (SC-ASSL) framework.** It heuristically integrates global and local context information.

aggregate multiple proxy tasks. Based on the hypothesis: *a weaker correlation means a higher complementarity between two features*, we first calculate the correlation measure (i.e., linear centered kernel alignment (LCKA) [14]) between the features yielded by different proxy tasks, and then employ the proposed greedy algorithm to iteratively add a proxy task with the lowest LCKA to the proxy tasks in the current task pool, and finally form a multi-task SSL framework. Lastly but not least, we implement a self-aggregation method to enlarge the feature space explored by a proxy task in a mode of *self-complementarity*. To achieve this, an auxiliary loss function based on LCKA is proposed as an add-on to the existing loss function to promote the exploring of where a single proxy task fails to cover.

In summary, our paper contributes three SSL aggregation strategies to exploit the complementarity between global and local contexts, among different proxy tasks, and with respect to self, respectively. Such aggregative SSL methods boost the robustness of the learned feature representation as demonstrated by our extensive evaluations on a 2D natural image dataset and a 3D medical volume dataset.

2. Spatial Context Aggregative SSL

As the global and local context information plays an important role in image classification, in this regard, we first propose two novel proxy tasks, exploiting spatial context complementary information, i.e., global and local context information from the raw image, respectively, and show the effectiveness of spatial context complementarity by combining them. The whole pipeline of our approach, as presented in Figure 1, is named as SC-ASSL.

Pre-processing. For an input image, we first randomly crop a sub-area, and then divide the cropped area into 3×3 tiles, $\{x_{i,j};\ i,j = 1,2,3\}$, with an overlap of 50%. A pool of sampling route is pre-defined as shown in the top of Figure. 1. After dividing the input image into tiles, we sample the tiles along a route randomly selected from the route pool as specified in Section 2.1. The sampled tiles are fed to a Siamese convolutional neural network (CNN) for feature extraction. The backbone of Siamese CNN can be any network architecture (e.g., VGG and ResNet). Denoting the network function of Siamese CNN and input tile by $f(.)$ and x, respectively, the feature vector (z) of each input tile can be yielded via:

$$z_i = f(x_i), \quad i = 1, ..., m \tag{1}$$

where m is the number of sampling tiles; $m = 5$ for 2D images in our experiments, as illustrated in Figure 1.

2.1. Sampling route classification

To extract the global context information, we formulate a novel proxy task, namely sampling route classification (SRC). Particularly, taking the two endpoints of a diagonal line illustrated in Figure 1 as an example, the framework is enforced to sample tiles from top-left corner (origin) to its bottom-right corner (destination) via a sampling route randomly selected from a pre-defined pool, which contains a total of 6 routes. Since each vertex of the image can be chosen as the origin, we can extend the pool to $4 \times 6 = 24$ sampling routes, as presented in the top of Figure 1. The neural networks are trained to recognize the sampling route by observing the input tiles, which enforce them to construct the global structure information of objects. Hence, the proposed SRC can be formulated as a 1-of-K classification task. To achieve this, the feature vectors of the tiles extracted by the Siamese CNN are concatenated and fed to a fully-connected layer for route classification. The cross-entropy loss is adopted for optimization.

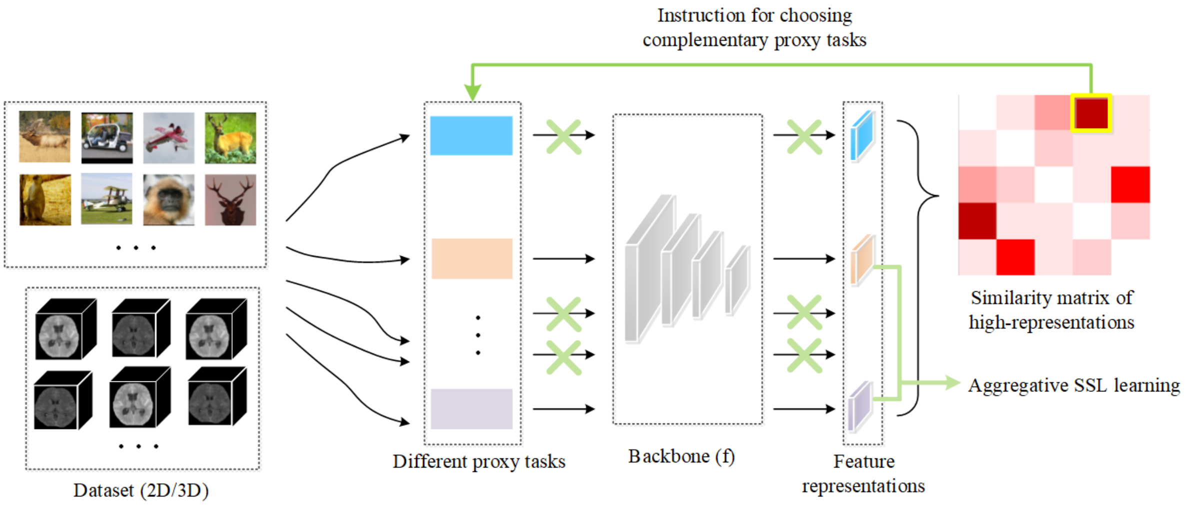

Figure 2. **The proposed multi-task aggregative self-supervised learning strategy (MT-ASSL) of each iteration.** We evaluate the similarity between feature representations learned by different proxy tasks using LCKA, and accordingly integrate the proxy task with the low similarity into the proposed MT-ASSL framework.

2.2. Modified contrastive predictive coding

To enrich information exploited by the Siamese CNN, we formulate another proxy task, namely modified contrastive predictive coding (MCPC) to extract local information, complementing global feature. The main difference between conventional CPC [9] and our MCPC lies in the route for feature extraction, where CPC extracts features from the tiles following a column-by-column manner and MCPC uses the same sampling route as in SRC. Assuming the feature extracted by the backbone network from a tile i as z_i, the support set containing $m-1$ features can be represented as $Z = \{z_i, \ i = 1, ..., m-1\}$ and the ground-truth for the predictive coding is z_m. A recurrent neural network (RNN) ψ is adopted to predict the feature vector (z'_m) of tile m based on the support features:

$$z'_m = \psi_{RNN}(Z). \qquad (2)$$

Contrastive loss. The purpose of MCPC is to recognize the unseen z_m from other representations $z_i, \ i \in \{1, ..., m-1\}$. To this end, the Siamese CNN is required to deeply exploit the local information of each input tile. A contrastive loss is adopted in our proxy task for network optimization, which is defined as:

$$L_C = -\sum_{i}^{z_i \neq z_m} \frac{z_m \log z'_m + (1-z_m) log(1-z'_m)}{z_i \log z'_m + (1-z_i) log(1-z'_m)} \qquad (3)$$

The $\{z_i | z_i \neq z_m\}$ denotes negative representations in the mini-batch. This loss is inspired by InfoNCE [5] and proved to have the ability to maximize the mutual information.

2.3. 3D extension of SC-ASSL

It is worthwhile to mention that our SC-ASSL can deal with not only 2D images but also 3D volumes. Our 3D extensions are as follows. A 3D volume is separated into $3 \times 3 \times 3$ cubes, $\{x_{i,j,k}, \ i,j,k = 1,2,3\}$, with a 50% overlap, as in Figure 3. The feature vector (z) of each cube is generated according to Eq. (1). Since each cuboid has 8 vertexes, we thereby can generate a route pool of $8 \times C_6^2 C_4^2 = 720$ alternative routes, which are too difficult for a network to classify. To reduce the computational complexity, we randomly select six alternative routes for each vertex, as shown in Figure 3. Consequently, the number of 3D sampling routes contained in the pool decreases to $8 \times 6 = 48$, i.e., the values of K and m are set to 48 and 7, respectively.

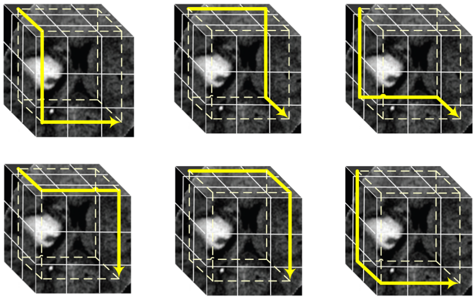

Figure 3. **Examples of 3D sampling routes.**

3. Multi-Task Aggregative SSL

Our SC-ASSL in terms of spatial context complementarity is simple yet effective to improve the robustness of learned feature representation. This observation drives us to explore deeper into the approach integrating the complementary features from multiple proxy tasks. In this section, we propose a novel multi-task aggregative self-supervised learning (MT-ASSL) strategy to iteratively integrate the features extracted by different proxy tasks and finally construct a robust feature representation.

The pipeline of our MT-ASSL is shown in Figure 2. First, we separately train the backbone network with each of the candidate proxy tasks and obtain a pool of feature representations $\{f_1, f_2, ..., f_k\}$, where k denotes the number of candidate proxy tasks. Then, using the linear centered kernel alignment technique [14] or other appropriate measures [24, 17], a similarity matrix S as in Figure 2 is obtained by calculating the correlation between the features yielded by each pair of proxy tasks,

$$S_{ij} = \|C(f_i^T f_i) * C(f_j^T f_j)\|_1 \tag{4}$$

where $f_i^T f_i$ and $*$ denote a dot product and element-wise multiplication, respectively. $C(x)$ denotes the centered alignment operation, which is defined as:

$$C(X) = X(I_n - \mathbf{1}\mathbf{1}^T/n), \tag{5}$$

where n denotes the dimension of X, I_n is an identity matrix of size $n \times n$, and $\mathbf{1}$ is an $n \times 1$ vector of ones.

The matrix S offers a simple yet concrete measurement of task correlation, which can be used as the guideline for aggregation of multiple proxy tasks. After the generation of matrix S, instead of using all candidate SSL methods, we form a pool of proxy tasks by selecting the ones with the smallest S values, since those proxy tasks collectively encourage the network to learn diverse representative features, thereby making them more amenable to be fine-tuned to the target task. Following this selection criterion, a greedy training strategy is proposed to aggregate different proxy tasks for feature learning in an iterative fashion. For each iteration, we first update the task pool by adding one more task that has the weakest correlation with the existing tasks in the pool. Then we conduct a multi-task learning similar to [6] to learn the feature, as it it a simple yet effective way of feature aggregation. Specifically, for each mini-batch, we randomly select a proxy task from the pool to optimize the neural network. Therefore, as the training iteration increases, the features exploited by different proxy tasks are gradually integrated, which yields a feature representation of better generalization. The process of the proposed aggregative training is described in Algorithm 1.

4. Self-Aggregative SSL

Since our MT-ASSL is in virtue of different SSL methods, we further propose a novel aggregation strategy based on self-complementarity, namely self-aggregative SSL (Self-ASSL), to boost the generalization of the feature, taking the advantages of a single SSL. Typically, the features learned from the same proxy task have high similarities to each others even in the experiments initialized with different random seeds, due to some trivial solutions to the proxy task. To alleviate the problem, our Self-ASSL encourages the network to explore a new latent space, which is complementary to the original one, for feature extraction with the same proxy task and thereby enriches the representation by aggregating newly explored space.

The pipeline of our Self-ASSL is shown in Figure 4, which includes three steps. First, for the unlabeled data D_U, we train the backbone network under the supervision of the proxy task p—using the loss function L_p. Thus, we obtain a feature representation denoted as f_p. Then in Step 2 of Figure 4, for the same unlabeled data D_U, apart from the conventional proxy task loss L_p, we propose an auxiliary loss L_{Com} to enforce the learned representation f'_p to be

Algorithm 1 Multi-task aggregative SSL

```
Input:
    Training dataset $D$ and backbone model $f$;
    Candidate proxy task list: $\mathcal{P} = \{p_1, p_2, ..., p_k\}$;
    Target task $p_t$ with ground truth $g$.
Function:
    $Train(f, p)$: Train network $f$ with proxy task $p$;
    $Eval(f, g)$: Evaluate $f$ using ground truth $g$;
Procedure:
Aggregated_Task_Pool ($\mathcal{A}$) = []
for $p_i$ in $\mathcal{P}$: do
    $f_i \leftarrow Train(f, p_i)$
    $f_t \leftarrow Train(f_i, p_t)$
    $ACC_i \leftarrow Eval(f_t, g)$
end for
$\mathcal{A}$ += $p_x$, where $p_x$ achieves the $Max(ACC)$
Best_Acc = $Max(ACC)$
$P \leftarrow P - p_x$
while Best_Acc is updated and $\mathcal{P} \neq []$ do
    $f_{\mathcal{A}} \leftarrow Train(f, \mathcal{A})$
    $f_t \leftarrow Train(f_{\mathcal{A}}, p_t)$
    $s_i \leftarrow S(f_i, f_{\mathcal{A}})$ defined in Eq. (4)
    $\mathcal{A}$ += $p_y$, where $p_y$ achieves the $Min(s)$
    $ACC_{\mathcal{A}} \leftarrow Eval(f_{\mathcal{A}}, g)$
    Best_ACC = $ACC_{\mathcal{A}}$ if $ACC_{\mathcal{A}} \geq$ Best_ACC
    $\mathcal{P} \leftarrow \mathcal{P} - p_y$
end while
Output: Trained backbone model $f_t$.
```

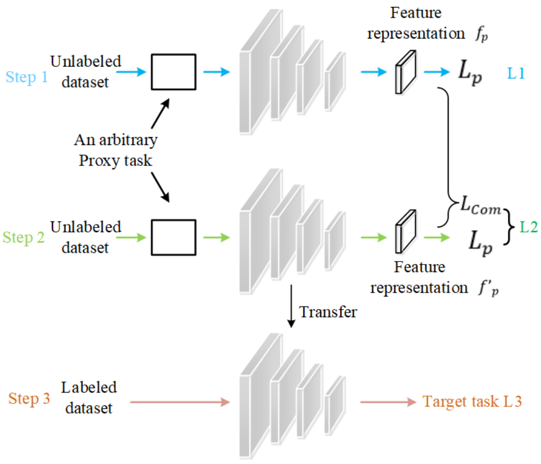

Figure 4. **The proposed self-aggregative SSL (Self-ASSL) learning pipeline.** It includes three different training steps. The feature representation f'_p gains a better generalization by aggregating a latent space complement to f_p.

different from the previous one f_p, which can be formulated as:

$$L_{Com} = -S_{f'f} = -\|C({f'_p}^T f'_p) * C(f_p^T f_p)\|_1 \qquad (6)$$

Then the total loss of step two for network optimization is $L_2 = L_p + L_{Com}$. After several iterations, the feature representation f'_p, achieving the better generalization, can further boost the improvement of the subsequent target task.

5. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of the proposed self-supervised learning approaches (i.e., SC-ASSL, MT-ASSL and Self-ASSL) and present the results in the following.

5.1. Datasets

2D natural image dataset: STL10. The STL10 dataset is for image recognition. It contains images of 96 × 96 pixels. The dataset is partially annotated (10 classes, 100 per class) and provides a large amount of unlabeled images (100K) for the development of unsupervised learning approaches. The SSL methods are first trained on unlabeled data and then fine-tuned on 1K labeled data for classification task.

3D medical volume dataset: Brain hemorrhage. The brain hemorrhage dataset, containing 1,486 brain CT volumes, is constructed by our collaboration hospital with IRB approval. The dataset consists of four pathological causes of cerebral hemorrhage: aneurysm, arteriovenous malformation, moyamoya disease and hypertension. The CT volumes are of a uniform size (30 × 270 × 230 voxels). We separate the brain hemorrhage dataset into training and testing sets according to the ratio of 80:20.

5.2. Implementation Details

For 2D natural image classification, several state-of-the-art SSL approaches, including SRC, MCPC, 2D jigsaw puzzles (2D Jigsaw) [19], image rotation (2D Rot) [8], image inpainting (Inpaint) [20] and SimCLR [2], are involved to form a pool of proxy tasks for our MT-ASSL and Self-ASSL. The 2D ResNet-18 is adopted as backbone network for MT-ASSL and Self-ASSL. For 3D SSL method, several 3D-based SSL approaches are included, i.e., 3D SC-ASSL, 3D rotation (3D Rot) [8], Model genesis (MG) [26], 3D CPC (3D version of [9] and Rubik's cube (Cube) [28]). The 3D ResNet-18 is utilized as backbone for MT-ASSL and Self-ASSL.

All methods are implemented using PyTorch. The network is trained with a mini-batch size of 128 and 16 for STL10 and brain hemorrhage datasets, respectively. The initial learning rate for the proxy task and target task are set to $1e^{-3}$ and $1e^{-4}$ for STL10, $2e^{-5}$ and $15e^{-6}$ for brain hemorrhage dataset, respectively. The Adam solver [12] is used as the optimizer for network training. The average classification accuracy (ACC) is employed as metric for performance evaluation.

5.3. Evaluation of MT-ASSL

To demonstrate the effectiveness of integrating different proxy tasks and explore the performance improvement caused by embedding the complementary information, we conduct a simple experiment on the STL10 dataset — evaluating the integration performance of two proxy tasks (denoted as $A1$ and $A2$) randomly selected from the SSL pool. The evaluation results are presented in Table 1. We first pre-train and fine-tune the ResNet-18 using each of the two proxy tasks and record the accuracy, i.e., the average accuracy (Avg ACC) of the paired proxy tasks and the maximum accuracy (Max ACC) among them. Then, we pre-train and fine-tune another ResNet-18 simultaneously using the two proxy tasks. The aggregation is achieved by iteratively optimizing the loss functions of $\mathcal{L}_{A1}$ and $\mathcal{L}_{A2}$.

It can be easily observed from Table 1 that the feature learned by proxy task integration (Int ACC) yields consistent improvements to the classification accuracy, compared to the single one. Furthermore, we evaluate the similarity between the feature representations learned by different proxy tasks using LCKA, as presented in the 'Similarity' column of Table 1. The performance improvements are observed to decline as the similarity increases, which confirms our hypothesis: *a weaker correlation or similarity means a higher complementarity between two features*. Hence, the feature learned by the aggregation of proxy tasks with low similarity (high complementarity) can significantly boost

Table 1. **Evaluation of the effectiveness of proxy task integration that aggregates each pair of SSLs ('A1' and 'A2') on STL10 dataset.** Proxy task 'A2' is sorted by the similarity of the last layer of ResNet-18 to 'A1'. The 'Avg ACC' and 'Max ACC' are the average ACC and the maximum ACC calculated by separately fine-tuning 'A1' and 'A2' pre-trained weights on the target task. The accuracy of proxy task integration is denoted as 'Int ACC'. Avg (+/-) and Max (+/-) are the improvements comparing Int ACC with Avg ACC and Max ACC, respectively.

A1	A2	Similarity	Avg ACC	Max ACC	Int ACC	Avg (+/-)	Max (+/-)
2D Jigsaw	2D Rot	0.1543	69.18	70.07	76.06	+6.88	+5.99
	SRC	0.1891	68.33	68.36	75.57	***+7.25***	***+7.21***
	SimCLR	0.2409	*70.67*	***73.05***	74.69	+4.02	+1.64
	Inpaint	0.4132	66.39	68.29	70.77	+4.38	+2.48
	MCPC	0.4202	67.62	68.29	68.81	+1.20	+0.52
2D Rot	SRC	***0.0695***	69.22	70.07	76.70	***+7.49***	***+6.63***
	SimCLR	0.0866	***71.56***	***73.05***	***78.21***	+6.65	+5.16
	Inpaint	0.1524	67.28	70.07	73.44	+6.16	+3.37
	2D Jigsaw	0.1543	69.18	70.07	76.06	+6.88	+5.99
	MCPC	0.1557	68.51	70.07	74.70	+6.20	+4.63
SimCLR	2D Rot	0.0866	***71.56***	***73.05***	***78.21***	***+6.65***	***+5.16***
	SRC	0.1085	70.71	***73.05***	77.25	+6.55	+4.20
	Inpaint	0.2373	68.77	***73.05***	74.08	+5.31	+1.03
	MCPC	0.2388	70.00	***73.05***	73.29	+3.30	+0.24
	2D Jigsaw	0.2409	70.67	***73.05***	74.69	+4.02	+1.64
Inpaint	2D Rot	0.1524	67.28	70.07	73.44	***+6.16***	+3.37
	SRC	0.1852	66.43	68.36	72.47	+6.04	***+4.11***
	SimCLR	0.2373	68.77	***73.05***	74.08	+5.31	+1.03
	MCPC	0.4125	65.72	66.94	65.09	-0.63	-1.85
	2D Jigsaw	0.4132	66.39	68.29	70.77	+4.38	+2.48
SRC	2D Rot	***0.0695***	69.22	70.07	76.70	***+7.49***	+6.63
	SimCLR	0.1085	70.71	***73.05***	77.25	+6.55	+4.20
	Inpaint	0.1852	66.43	68.36	72.47	+6.04	+4.11
	2D Jigsaw	0.1891	68.33	68.36	75.05	+6.73	***+6.69***
	MCPC	0.2573	67.65	68.36	72.03	+4.38	+3.67
MCPC	2D Rot	0.1557	68.51	70.07	74.70	***+6.20***	***+4.63***
	SimCLR	0.2388	70.00	***73.05***	73.29	+3.30	+0.24
	SRC	0.2573	67.65	68.36	72.03	+4.38	+3.67
	Inpaint	0.4125	65.72	66.94	65.09	-0.63	-1.85
	2D Jigsaw	0.4202	67.62	68.29	68.81	+1.20	+0.52

the target classification accuracy. Here we further validate the effectiveness of the proposed MT-ASSL strategy that integrates multiple proxy tasks using both 2D and 3D datasets.

STL10. We first evaluate the proposed MT-ASSL on STL10 dataset. The six proxy tasks are used to train ResNet-18 individually to obtain the corresponding self-supervised feature representations. Then, we calculate the similarity metric of the last layer of ResNet-18 by Eq. (4). The proxy task with the lowest calculated similarity is integrated to our aggregated task pool. We adopt the integrated proxy tasks to train a new ResNet-18 and repeat the process of similarity calculation and proxy task aggregation, following the greedy algorithm as presented in Algorithm 1. The results are shown in Table 2.

For the first iteration, SimCLR, which achieves the best performance of 73.05%, is added to the aggregation task pool. The 2D Rot and SRC proxy tasks with a lower similarity to SimCLR are involved during iteration two and three, respectively. The MT-ASSL is completed after four iterations since no further performance improvement is observed as the similarities of the rest three proxy tasks are nearly the same. Therefore, our MT-ASSL obtains the best combination of proxy tasks (i.e., SimCLR + 2D Rot + SRC) for the image classification on STL10 dataset, which results in a final target accuracy of 79.43%.

Brain hemorrhage. We also evaluate our MT-ASSL with five 3D-based SSL methods on brain hemorrhage dataset. The evaluation results are in Table 3. The best model is the combination of SC-ASSL and Cube. The aggregation of

Table 2. **Results of our multi-task aggregative SSL (MT-ASSL) on STL10 dataset.** The 'A1' and 'A2' indicate two single SSL methods for aggregation. 'A2' are sorted by the similarity of the fourth layer to proxy task 'A1'. The 'Avg ACC' and 'Max ACC' are the average ACC and the maximum ACC of 'A1' and 'A2'. Avg (+/-) and Max (+/-) are the improvements comparing MT-ASSL ACC with Avg ACC and Max ACC, respectively. (Iter.–Iteration)

Iter.	A1	A2	Similarity	Avg ACC	Max ACC	MT-ASSL ACC	Avg (+/-)	Max (+/-)
1	SRC	-				68.36		
	2D Jigsaw	-				68.29		
	Inpaint	-				64.49		
	SimCLR	-				***73.05***		
	2D Rot	-				70.07		
	MCPC	-				66.94		
2	SimCLR	2D Rot	***0.0866***	***71.56***	***73.05***	***78.21***	***+6.65***	***+5.16***
		SRC	0.1085	70.71	***73.05***	77.25	+6.55	+4.20
		Inpaint	0.2373	68.77	***73.05***	74.08	+5.31	+1.03
		MCPC	0.2388	70.00	***73.05***	73.29	+3.30	+0.24
		2D Jigsaw	0.2409	70.67	***73.05***	74.69	+4.02	+1.64
3	SimCLR + 2D Rot	SRC	***0.0911***	73.29	78.21	***79.43***	+6.15	***+1.22***
		Inpaint	0.1973	71.35	78.21	78.16	***+6.81***	-0.05
		MCPC	0.1985	72.58	78.21	77.97	+5.40	-0.24
		2D Jigsaw	0.1986	73.25	78.21	77.89	+4.64	-0.32
4	SimCLR + 2D Rot + SRC	Inpaint	***0.2251***	71.96	***79.43***	76.01	+4.05	-3.42
		MCPC	0.2283	73.19	***79.43***	75.78	+2.60	-3.65
		2D Jigsaw	0.2284	73.86	***79.43***	76.25	+2.39	-3.18

Table 3. **Results of our multi-task aggregative SSL (MT-ASSL) on brain hemorrhage dataset.** The 'A1' and 'A2' indicate two single SSL methods for aggregation. 'A2' are sorted by the similarity of the fourth layer to proxy task 'A1'. The 'Avg ACC' and 'Max ACC' are the average ACC and the maximum ACC of 'A1' and 'A2'. Avg (+/-) and Max (+/-) are the improvements comparing MT-ASSL ACC with Avg ACC and Max ACC, respectively. (Iter.–Iteration)

Iter.	A1	A2	Similarity	Avg ACC	Max ACC	MT-ASSL ACC	Avg (+/-)	MAX (+/-)
1	SC-ASSL	-				***89.53***		
	Cube	-				87.50		
	3D CPC	-				83.79		
	3D Rot	-				87.16		
	MG	-				87.50		
2	SC-ASSL	Cube	***0.0612***	***88.52***	***89.53***	***90.20***	**+1.69**	**+0.67**
		3D CPC	0.1722	86.66	***89.53***	87.50	+0.84	-2.03
		3D Rot	0.2413	88.35	***89.53***	88.17	-0.17	-1.36
		MG	0.2415	***88.52***	***89.53***	87.83	-0.69	-1.70
3	SC-ASSL + Cube	3D Rot	***0.0403***	88.68	***90.20***	89.52	+0.84	-0.68
		MG	0.1054	88.85	***90.20***	88.51	-0.34	-1.69
		3D CPC	0.1059	87.00	***90.20***	87.50	+0.50	-2.70

3D CPC is observed to degrade the performance by a large margin of -2.70%. The underlying reason is that our SC-ASSL also contains a modified CPC branch; therefore, the information exploited by 3D CPC may be redundant to the integration of SC-ASSL and Cube, which deteriorates the pre-training.

5.4. Evaluation of SC-ASSL and Self-ASSL

For the evaluation of SC-ASSL and Self-ASSL, apart from the ResNet-18 model, we also use the VGG model as a backbone to validate the generalization of the proposed approaches. The evaluation on the STL10 and brain hemorrhage datasets are presented in Table 4.

For STL10, it is observed that the Self-ASSL strategy consistently boosts the accuracy of proxy tasks, e.g., $+2.73\%$ for SimCLR with VGG and $+2.51\%$ for 2D Jigsaw with ResNet-18, with the only exception of 2D Rot with VGG. The exact reason of such an exception is unclear and worthy of further investigation. Specifically, the Self-ASSL-trained SC-ASSL with ResNet-18 achieves the best ACC of 74.28%, which demonstrates the merit of this heuristic aggregation method. A similar trend of improve-

Table 4. **Accuracy (ACC %) of different proxy tasks tested on STL10 and Brain hemorrhage datasets.** ACC (+/-) lists the improvements of ACC comparing Self-ASSL-trained SSLs to the original ones. (T. f. s.–Train from scratch)

STL10					Brain hemorrhage				
Backbone	Method	ACC	w. Self-ASSL	ACC (+/-)	Backbone	Method	ACC	w. Self-ASSL	ACC (+/-)
2D VGG	T. f. s.	63.90	-	-	3D VGG	T. f. s.	72.30	-	-
	2D Jigsaw	62.84	63.82	+0.98		3D CPC	77.02	83.44	**+6.42**
	Inpaint	64.76	65.25	+0.49		3D Rot	76.68	79.05	+2.37
	2D Rot	69.33	69.09	-0.24		Cube	77.36	81.08	+3.72
	SimCLR	**70.31**	**73.04**	+2.73		MG	85.81	86.15	+0.34
	SRC	62.90	63.21	+0.31		SRC	85.81	87.50	+1.69
	MCPC	65.25	68.42	**+3.17**		MCPC	87.50	**89.52**	+2.02
	2D SC-ASSL	65.69	66.45	+0.76		3D SC-ASSL	**87.83**	88.17	+0.34
2D ResNet-18	T. f. s.	63.19	-	-	3D ResNet-18	T. f. s.	81.08	-	-
	2D Jigsaw	68.29	70.80	**+2.51**		3D CPC	83.79	88.17	**+4.38**
	Inpaint	64.49	65.67	+1.18		3D Rot	85.81	85.47	-0.34
	2D Rot	70.07	72.41	+2.34		Cube	87.50	88.85	+1.35
	SimCLR	**73.05**	73.38	+0.33		MG	87.50	88.17	+0.67
	SRC	68.36	69.33	+0.97		SRC	87.16	88.51	+1.35
	MCPC	66.94	67.18	+0.24		MCPC	88.51	88.85	+0.34
	2D SC-ASSL	72.03	**74.28**	+2.25		3D SC-ASSL	**89.53**	**89.53**	+0.00

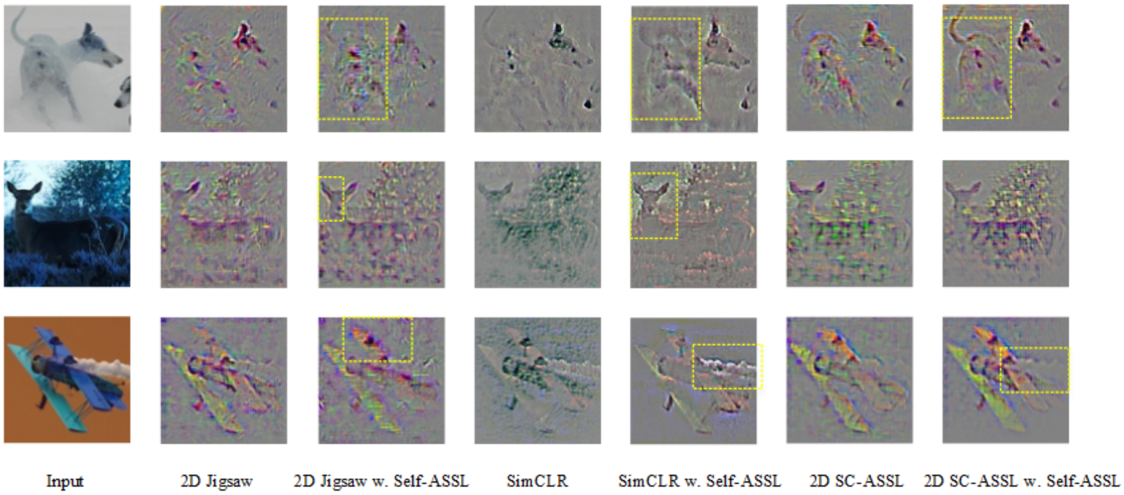

Figure 5. **Grad-CAM visualization of SSL and Self-ASSL.** The yellow rectangles indicate the pronounced differences.

ment is observed on the brain hemorrhage dataset. Our Self-ASSL training strategy boosts the 3D CPC and Cube with VGG by large margins of $+6.42\%$ and $+3.72\%$, respectively. Also, the only exception happens to 3D Rot with ResNet-18. Our Self-ASSL-trained SC-ASSL outperforms the benchmarking algorithms on brain hemorrhage dataset as well, i.e., an ACC of 89.53% is achieved using ResNet-18 as backbone.

Visualization. To further demonstrate the effectiveness of our Self-ASSL, we employ Guided Grad-CAM [21] to visualize the feature learned by the last convolution layer of ResNet-18. Three examples are presented in Figure 5. The Self-ASSL brings more attention to detailed information (marked using yellow rectangles), which is ignored by the conventional SSL method. For example, the Self-ASSL-trained Jigsaw captures the information of the dog body, which is omitted by the original Jigsaw. Overall, the feature visualization further validates the effectiveness of our Self-ASSL in helping the proxy task to capture more detailed information by aggregating self-complementary features from a raw image, thereby leading to improved classification accuracy.

6. Conclusion

We propose three approaches for SSL aggregation by exploiting complementarity of spatial context, multiple proxy tasks, and single proxy task itself, respectively. We first construct a heuristic SSL method that simultaneously exploits spatial context complementarity from raw data. Then, we propose an effective multi-task aggregative strategy to fuse multiple proxy tasks and extract the complementary features. Last but not least, a self-aggregative SSL, which is simple but effective, is implemented to aggregate self-complementary feature to boost the performance of a single SSL method. Our extensive experiments on 2D natural and 3D medical image datasets show that the proposed aggregation strategies expose new insights for self-supervised learning and significantly improve the accuracy of learned features on the target tasks. Future work includes mining the feature complementarity among off-the-shelf networks for various vision tasks.

References

[1] M. Blendowski, H. Nickisch, and M. Heinrich. How to learn from unlabeled volume data: Self-supervised 3d context feature learning. In *Medical Image Computing and Computer Assisted Intervention*, 2019. 1

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E Hinton. A simple framework for contrastive learning of visual representations. *arXiv: Learning*, 2020. 1, 5

[3] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *ArXiv*, abs/2006.10029, 2020. 1

[4] Tianlong Chen, Sijia Liu, S. Chang, Y. Cheng, L. Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 696–705, 2020. 1

[5] Aaron Van Den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv: Learning*, 2018. 3

[6] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. *IEEE International Conference on Computer Vision*, pages 2070–2079, 2017. 1, 4

[7] J. Donahue and K. Simonyan. Large scale adversarial representation learning. In *Conference and Workshop on Neural Information Processing Systems*, 2019. 1

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1, 5

[9] Olivier J Henaff, Ali Razavi, Carl Doersch, S M Ali Eslami, and Aaron Van Den Oord. Data-efficient image recognition with contrastive predictive coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3, 5

[10] Tomas Jakab, A. Gupta, Hakan Bilen, and A. Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8784–8794, 2020. 1

[11] S. Jenni, H. Jin, and P. Favaro. Steering self-supervised feature learning beyond local pixel statistics. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6407–6416, 2020. 1

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[13] A. Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019. 1

[14] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019. 2, 4

[15] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6478–6487, 2020. 1

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 840–849, 2017. 1

[17] Ari S. Morcos, M. Raghu, and S. Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Conference on Neural Information Processing Systems*, 2018. 4

[18] Alejandro Newell and Jun Deng. How useful is self-supervised pretraining for visual tasks? *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7343–7352, 2020. 1

[19] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84, 2016. 1, 5

[20] Deepak Pathak, Philipp Krähenbühl, J. Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 5

[21] R. R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, D. Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2019. 8

[22] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *ArXiv*, abs/1906.05849, 2019. 1

[23] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1910–1919, 2019. 1

[24] D. Wilks. Canonical correlation analysis (CCA). *International Geophysics*, 100:563–582, 2011. 4

[25] Zongwei Zhou, Vatsal Sodha, Jiaxuan Pang, Michael B. Gotway, and Jianming Liang. Models genesis. *Medical Image Analysis*, page 101840, 2020. 1

[26] Zongwei Zhou, Vatsal Sodha, Mahfuzur Rahman Siddiquee, Ruibin Feng, Nima Tajbakhsh, Michael B Gotway, and Jianming Liang. Models genesis: Generic autodidactic models for 3d medical image analysis. In *Medical Image Computing and Computer Assisted Intervention*, pages 384–393, 2019. 1, 5

[27] Jiuwen Zhu, Yuexiang Li, Yifan Hu, Kai Ma, S. Kevin Zhou, and Yefeng Zheng. Rubik's Cube+: A self-supervised feature learning framework for 3D medical image analysis. In *Medical Image Analysis*, volume 64, page 101746, 2020. 1

[28] X. Zhuang, Y. Li, Y. Hu, K. Ma, Y. Yang, and Y. Zheng. Self-supervised feature learning for 3D medical images by

playing a Rubik's cube. In *Medical Image Computing and Computer Assisted Intervention*, 2019. 1, 5