Spatial Ensemble: a Novel Model Smoothing Mechanism for Student-Teacher Framework

Tengteng Huang  Yifan Sun  Xun Wang  Haotian Yao∗  Chi Zhang
Megvii Technology
{huangtengteng, yaohaotian, zhangchi}@megvii.com,
sunyf15@tsinghua.org.cn, bnuwangxun@gmail.com

Abstract

Model smoothing is of central importance for obtaining a reliable teacher model in the student-teacher framework, where the teacher generates surrogate supervision signals to train the student. A popular model smoothing method is the Temporal Moving Average (TMA), which continuously averages the teacher parameters with the up-to-date student parameters. In this paper, we propose “Spatial Ensemble”, a novel model smoothing mechanism in parallel with TMA. Spatial Ensemble randomly picks up a small fragment of the student model to directly replace the corresponding fragment of the teacher model. Consequentially, it stitches different fragments of historical student models into a unity, yielding the “Spatial Ensemble” effect. Spatial Ensemble obtains comparable student-teacher learning performance by itself and demonstrates valuable complementarity with temporal moving average. Their integration, named Spatial-Temporal Smoothing, brings general (sometimes significant) improvement to the student-teacher learning framework on a variety of state-of-the-art methods. For example, based on the self-supervised method BYOL, it yields +0.9% top-1 accuracy improvement on ImageNet, while based on the semi-supervised approach FixMatch, it increases the top-1 accuracy by around +6% on CIFAR-10 when only few training labels are available. Codes and models are available at: https://github.com/tengteng95/Spatial_Ensemble

1 Introduction

Recent years have witnessed the success of the student-teacher framework in self-supervised [1–5] and semi-supervised [6–8] learning tasks. In this framework, the teacher is responsible for generating surrogate supervision signals to guide the learning of the student on unlabeled data, for example, category IDs for classification, feature patterns for contrastive learning, etc.

Many important works under this framework share a common model smoothing technique, i.e., the Temporal Moving Average (TMA), to generate the teacher from the student model. For example, Mean Teacher [6] for semi-supervision averages historical student weights as the teacher. MoCo [1] for self-supervision employs a slowly progressing encoder (teacher), driven by a momentum update with the query encoder (student). Another state-of-the-art method for self-supervision BYOL [2] employs a similar strategy to update the target (teacher) network with a slow-moving average of the online (student) network. We thus recognize that model smoothing with TMA is of central importance for obtaining a reliable teacher for these works.

In this paper, we rethink the mechanism of TMA and summarize its importance as two-fold. First, TMA enables the teacher to absorb parameters from different student editions and thus benefits from

∗corresponding author

We propose a novel model smoothing mechanism named “Spatial Ensemble”. Compared with TMA, Spatial Ensemble facilitates similar model smoothing benefits with a different mechanism, as illustrated in Figure 1. In TMA, the teacher absorbs each historical student as a whole. A temporally near student is absorbed into the teacher with a larger weighting factor due to the Exponential Moving Average effect. In contrast, in Spatial Ensemble, the teacher randomly picks up one or more small fragments of the student to replace the corresponding fragments of the teacher during each update. All the fragments are weighted equally in the teacher model. In this way, Spatial Ensemble stitches different fragments of previous student models into a unity, yielding the “Spatial Ensemble” effect. In spite of their fundamental differences, Spatial Ensemble facilitates similar benefits of model ensemble and smoothed update. The smoothed update is maintained by replacing only a small portion of parameters at each iteration. On the challenging self-supervised learning task, which is very sensitive to the quality of surrogate supervision signals, we observe that Spatial Ensemble achieves comparable performance with temporal moving average. It shows that our Spatial Ensemble can serve as a basic model smoothing technique for the student-teacher framework.

An important advantage of Spatial Ensemble lays in its complementarity to Temporal Moving Average. To explore their complementary benefits, we integrate both the temporal smoothing and spatial smoothing mechanism into a unified method named Spatial-Temporal Smoothing (STS). STS does not directly absorb a whole student, or simply replace a fragment. Instead, it updates a fragment of the teacher model with a temporal moving average of the corresponding fragment of the student. This “mix-up” model smoothing benefits from the mutual complementarity of temporal moving average and Spatial Ensemble. Experiments on both self-supervised and semi-supervised tasks show that using STS for model smoothing obtains general (sometimes significant) improvement over a majority of state-of-the-art methods.

We summarize our contribution as follows:
1. We propose a novel model smoothing mechanism named Spatial Ensemble. Compared with the popular temporal moving average, Spatial Ensemble achieves a comparable model smoothing effect with a different mechanism.
2. We integrate the spatial and temporal smoothing mechanism into a unified method named Spatial-Temporal Smoothing. STS benefits from the mutual complementarity of two mechanisms and consistently improves student-teacher methods on self- and semi-supervised learning tasks.
3. We notice two recognizable advantages of the model learned through STS, including strong robustness against various data corruptions, and superior transferring capability to the object detection task.

## 2 Related Work

In this section, we present a brief review of the development on semi- and self-supervised learning, as well as the student-teacher framework. We mainly focus on the recent deep learning literature related to our method.
**Semi-supervised learning** aims at achieving comparable performance to fully supervised methods, usually with a small portion of labeled data and the rest as unlabeled data. It is of great importance for real scenarios where annotations are difficult/expensive to acquire, such as medical data. Pseudo labels [9, 10] and consistency regularization [6, 7, 11] are two essential techniques in many recent semi-supervised approaches. Pseudo-Labeling [9] generates soft pseudo labels for the unlabeled data using the logits predicted by the model from the latest epoch. Laine et al. [10] improve the pseudo label generation process by leveraging a temporal ensembling prediction of different epochs. Mean-teacher [6] further simplifies this process by introducing a teacher model, which is a temporal ensembling of the student model through the exponential moving average, to generate pseudo labels directly. Besides, mean-teacher employs a consistency regularization to encourage similar predictions from the teacher and the student model. FixMatch [7] assumes that the consistency should also hold between images applied with augmentations of different degrees, and applies both a weak and a strong set of data augmentations.

**Self-supervised learning** largely narrows down its performance gap with supervised approaches and is receiving increasing attention in the community. Recent self-supervised methods can be roughly categorized into two genres, i.e., approaches based on pretext task and approaches based on contrastive learning. There are many kinds of insightful pretext task designs. One common method is generating surrogate supervision signals by feature clustering [12-14], image augmentation (usually rotation [15]), relative order of clipped image patches [16-18], etc. Another mainstream approach is to design a generative pretext task, e.g., image colorization [19, 20], image inpainting [21] and image denoising [22], and usually employ auto-encoders or generative adversarial networks [23] to transfer images. Instead of designing a pretext task, contrastive learning approaches [1, 4, 24-29] aim at maximizing the feature similarities with augmented views produced from the identical image sample, while minimizing those from different image samples.

**Student-teacher framework** is a popular architecture that has been extensively used in many recent state-of-the-art methods, especially for semi- and self-supervised learning tasks. Usually, temporal moving average is employed to generate the teacher from historical student models. The optimization goal is to ensure the prediction consistency between the student and the teacher. Besides pulling different views of the same image closer, MoCo [1] employs a memory bank to store negative samples and pushes views from different images apart. BYOL [2] removes negative samples and leverages an asymmetric student-teacher architecture to prevent collapsing. An extra predictor is appended to the online (student) network to regress the output of the target (teacher) network.

### 3 Method

**Preliminaries.** Given a same network architecture, $\Theta^T$ and $\Theta^S$ are the parameters of the teacher and student model. Both $\Theta^T$ and $\Theta^S$ contains $n$ units of parameters, i.e., $\Theta^T = \{\theta^T_1, \theta^T_2, \ldots, \theta^T_n\}$ and $\Theta^S = \{\theta^S_1, \theta^S_2, \ldots, \theta^S_n\}$. A unit may correspond to a layer, a neuron or a channel, depending on the desired spatial granularity. To analyze the model update in consecutive iterations, we further define them as functions of the iteration $t$, i.e., $\Theta^T(t) = \{\theta^T(t)\}$ and $\Theta^S(t) = \{\theta^S(t)\}$.
3.1 Spatial Ensemble

We present a new model smoothing method named Spatial Ensemble (SE), as illustrated in Figure 2(b). During each iteration, SE randomly picks up some units of the student model to replace the corresponding units of the teacher model, leaving the remaining parts of the teacher unchanged. We formulate this process as:

$$\Theta_T(t) = \{P_i \theta^T_i(t-1) + (1 - P_i) \theta^S_i(t-1)\},$$

where $P_i \sim Ber(p)$ (i.e., the Bernoulli distribution with success probability $p$) is a binary variable. We note that all $P_i \sim Ber(p) (i = 1, 2, \cdots, n)$ are mutually independent to each other and have the same success probability $p$. It determines whether the $i$-th unit of the teacher parameters is to be preserved. Specifically, if $P_i = 1$, the $i$-th unit parameters is to be preserved. If $P_i = 0$, the $i$-th unit is to be replaced. Since $P_i \sim Ber(p)$, larger $p$ results in lower replacing frequency and higher proportion of preserved units. Therefore, we name $p$ as the “preserving probability”.

For clarity, we omit the iteration indicator in all the equations in the following part of this paper and rewrite Equation 1 as:

$$\Theta_T = \{P_i \theta^T_i + (1 - P_i) \theta^S_i\}.$$  

(2)

To investigate the mechanism of SE, we quantitatively analyze the smoothing effect on both the model parameters and the generated surrogate supervision signals. Specifically, we calculate the Mean-Square Errors (MSE) between teachers at two adjacent epochs, as well as the MSE between their outputs. We visualize the MSE values at the early training stages (epoch 1 to 20) in Fig. 3, from which we make two observations:

**Remark 1: Spatial Ensemble smooths the teacher update.** It is observed that without model smoothing, the MSE of both model parameters and supervision signals maintain relatively high values, indicating strong vibration during the teacher update. In another word, the teacher model and its output signals undergo substantial instability. In contrast, when using the TMA and the proposed SE to update the teacher, the MSE of both model parameters and supervision signals are much smaller (than Non-smoothing), yielding a smoothed teacher update.

**Remark 2: Spatial Ensemble promotes convergence.** Using TMA and SE, the MSE values gradually decrease as the training epochs increases. Such result indicates that the up-to-date teacher model gradually become closer to its prior edition, therefore approaching convergence. Without TMA or SE, the Non-smoothing mode maintains a roughly constant MSE value. We thus infer that Spatial Ensemble (as well as the TMA) promotes convergence of student-teacher framework.

We provide the pseudocode of SE in Algorithm 1, which is easy to implement with only a few lines of code. In Section 4.3 we further analyze the effect of $p$ on student-teacher learning performance.

### Algorithm 1 Pseudocode of SE.

```python
# net_s, net_t: student/teacher network  
# m: momentum for TMA  
# p: probability of a student layer belonging to the unpicked fragment.
params_s = net_s.parameters()  
params_t = net_t.parameters()  
for s, t in zip(params_s, params_t):  
    if random.rand() < p:  
        continue  
    else:  
        t = s
```

### Algorithm 2 Pseudo code of STS.

```python
# net_s, net_t: student/teacher network  
# m: momentum for TMA  
# p: probability of a student layer belonging to the unpicked fragment.
params_s = net_s.parameters()  
params_t = net_t.parameters()  
for s, t in zip(params_s, params_t):  
    if random.rand() < p:  
        continue  
    else:  
        t = m * t + (1 - m) * s
```

3.2 Spatial-Temporal Smoothing

We further integrate TMA and SE into a unified model smoothing method named Spatial-Temporal Smoothing (STS), to benefit from their mutual complementarity. STS picks up several units in
Figure 3: Illustration of mean square error of teacher parameters (Left) and generated supervision signals (Right) at two adjacent epochs. None represents no model smoothing is used. TMA adopts a momentum of 0.999. p is set to 0.999 for SE. The vertical coordinates are arranged in Log scales.

the student model and averages them with the corresponding units in the teacher model, which is formulated as:

$$\Theta^T \leftarrow \{P_i \theta^T_i + (1 - P_i)m \theta^T_i + (1 - P_i)(1 - m)\theta^S_i\},$$  \hspace{1cm} (3)

where $m$ is the momentum for TMA.

STS degenerates to SE or TMA, given specification of hyper-parameters. When $m = 0$, STS degenerates into SE. Similarly, when we set the preserving probability $p = 1$, which results in $P_i \equiv 0$, STS degenerates into TMA:

$$\Theta^T \leftarrow m\Theta^T + (1 - m)\Theta^S,$$ \hspace{1cm} (4)

Algorithm 2 illustrates the pseudocode of STS. Similar to SE, STS is easy to implement and can be integrated into most teacher-student frameworks. In Section 4.3, we present detailed analysis of how $p$ and $m$ influence the performance of STS.

4 Experiments

4.1 Baselines and Implementation Details

We evaluate the effectiveness of the proposed Spatial-Temporal Smoothing by applying it to the state-of-the-art self-supervised approaches (MoCo [1] and BYOL [2]) and semi-supervised method (FixMatch [7]). We use the official implementation of MoCo [1] and re-implement BYOL and FixMatch using Pytorch [30]. All experiments are conducted on a machine with 8 RTX2080 GPUs. The masking probability $p$ is set to 0.7/0.5/0.5 for BYOL/MoCo/FixMatch, respectively. We dump the teacher model at the end of training. The self-supervised models are evaluated by following the common protocol of linear classification [1, 2, 4, 31] while semi-supervised performance are assessed with top-1 accuracy metric. We carefully follow both the training and evaluation settings of [1, 2, 7] in their original paper for fair comparison. More details are given in the following.

MoCo [1] consists of a query model and a key model, which simulates the same role as the student and the teacher model, respectively. We adopt MoCo v2 with ResNet-50 as our baseline and use a memory bank of size 65536. The initial learning rate is set to 0.03 and adjusted by cosine learning rate scheduler [32]. Following the original paper, we train the model using SGD with momentum of 0.9, weight decay of 0.0001, and a mini-batchsize of 256. Both the query and key model adopt the same data augmentations as SimCLR [4], i.e., random crop, horizontal flip, color jitter, Gaussian blur, and gray-scale conversion.

BYOL [2] reformulates the self-supervised learning task as a regression task which maximizes the similarity of feature representations from the teacher and the student model. We train BYOL with Synchronized batch normalization [33] and use a mini-batchsize of 256. The basic learning rate is 0.1 and is decayed by cosine strategy [32]. The first 10 epochs are warmed up with a factor of 0.001 and
gradually increased to the basic learning rate. We adopt SGD with momentum of 0.9 as the optimizer and set weight decay to 1e-4. BYOL uses the same set of data augmentations as SimCLR [4] as listed above, and adopts solarization with probability of 0.2 for the teacher model.

FixMatch [7] uses identical student and teacher network in the training phase while employing the TMA teacher for evaluation. We use Wide ResNet-28-2 [34] and Wide ResNet-28-8 as the backbone networks for CIFAR-10 and CIFAR-100, respectively. We use SGD with momentum 0.9 as the optimizer and train the model for 200 iterations following the original paper. Each mini-batch comprises of 64 labeled images and 448 unlabeled images (i.e., 1:7). The initial learning rate is 0.03 and is decayed by cosine scheduler [32]. The weight decay is set to 0.0005 and 0.001 for CIFAR-10 and CIFAR-100, respectively. For data augmentation, the student only uses random crop and random horizontal flip. Besides these two augmentations, the teacher additionally adopts random augmentation [35].

Evaluation. Linear Evaluation is a common metric for measuring whether the feature representation learned by the self-supervised model is good. Typically, a classifier is appended to the feature encoder and finetuned on the down-streaming task. During the whole process, the parameters of the feature encoder, as well as the BN statistics, are fixed. We train the classifier appended to MoCo using the SGD optimizer for 100 epochs. The initial learning rate is 30 and is decayed at the 60th and 80th epoch with factor 0.1. For BYOL, the basic learning rate is 0.2 and is adjusted according to the cosine strategy during the full 80 epochs.

4.2 The Effect of SE and STS

We first evaluate the effect of SE on the self-supervised task. We select MoCo and BYOL as the baseline models and train them for 200 epochs. As shown in Table 1, disabling the model smoothing mechanism in MoCo and BYOL leads to failure of convergence, which is also observed in their original papers. The experimental results indicate the importance of model smoothing in these self-supervised approaches. When combined with SE, MoCo/BYOL achieves 66.7%/67.8% top-1 accuracy, demonstrating the effectiveness of SE in model smoothing.

We further validate the effectiveness of STS which integrates both the temporal and spatial smoothing strategies. As is shown in Table 2, compared to typical TMA, our method boosts the performance of MoCo (BYOL) from 67.5% (71.8%) to 67.8% (72.7%) in terms of top-1 accuracy, indicating the valuable complementarity of the spatial and temporal smoothing mechanism.

<table>
<thead>
<tr>
<th>Method</th>
<th>Smoothing</th>
<th>top-1</th>
<th>top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCo</td>
<td>-</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>MoCo</td>
<td>SE</td>
<td>66.7</td>
<td>87.0</td>
</tr>
<tr>
<td>BYOL</td>
<td>-</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>BYOL</td>
<td>SE</td>
<td>67.8</td>
<td>88.3</td>
</tr>
</tbody>
</table>

Table 1: Analysis of the effectiveness of SE as a model smoothing approach.

<table>
<thead>
<tr>
<th>Method</th>
<th>Smoothing</th>
<th>top-1</th>
<th>top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCo</td>
<td>TMA</td>
<td>67.5</td>
<td>88.0</td>
</tr>
<tr>
<td>MoCo</td>
<td>STS</td>
<td>67.8</td>
<td>88.1</td>
</tr>
<tr>
<td>BYOL</td>
<td>TMA</td>
<td>71.8</td>
<td>90.6</td>
</tr>
<tr>
<td>BYOL</td>
<td>STS</td>
<td>72.7</td>
<td>90.9</td>
</tr>
</tbody>
</table>

Table 2: Analysis of the superiority of STS over TMA.

4.3 Ablation Studies

In this section, we evaluate the effect of hyperparameters in STS and then investigate Spatial Ensemble of different granularities. BYOL is adopted as the default network for the following experiments. All the ablation experiments are conducted on the ImageNet dataset [36] and trained for 200 epochs unless noted otherwise.

Effect of hyperparameters $p$ and $m$. To explore how $p$ and $m$ influences the student-teacher learning performance, we conduct extensive experiments by selecting $m$ from {0, 0.9, 0.99, 0.999} and varying $p$ from 0.1 to 0.9 with an interval of 0.2. We first investigate the effect of $p$ by setting $m = 0$. Under this case, STS degenerates into SE. As is shown in Figure 4, BYOL fails to converge when $p = 0$ while achieves increasing performance with the increase of $p$, demonstrating the importance of model smoothing. Figure 4 illustrates the results with different combinations of $p$ and $m$. It can be observed that STS yields better performance than typical TMA ($p = 0$) in most cases. We also notice an interesting fact that with the increase (decrease) of $m$, the model achieves its best performance
at relatively smaller (larger) value of $p$. For example, $p = 0.99$ is the best choice when $m$ is set to 0.9. However, for $m = 0.99$, a smaller $p$ of 0.7 yields much better performance than larger $p$ of 0.99. These results demonstrate the complementarity of spatial and temporal smoothing, as well as the important role of model smoothing for self-supervised learning based on the student-teacher framework.

**Granularity of Spatial Ensemble.** SE may have various granularities, corresponding to a layer, a channel, or a neuron as the smallest unit. When integrating SE into STS, we set the granularity to layer-wise by default. In this section, we investigate the impacts of different granularities, i.e., layer-wise (LW), channel-wise (CW) and neuron-wise (NW). Specifically, for a student parameter with shape $C_{in} \times C_{out}$, where $C_{in}$ and $C_{out}$ denote the number of input and output channels, respectively. For CW, we randomly pick up a small set of indexes along with the $C_{out}$ dimension and replace the corresponding teacher parameters. Similarly for NW, we randomly select neurons from the whole student model and replace the corresponding neurons of the teacher model. As shown in the top of Table 4, SE using NW outperforms its counterpart using LW and CW and achieves comparable performance with TMA. The result reveals that SE itself benefits from the fine-grained spatial ensemble. Besides, three variants of STS result in comparable performance, indicating the complementarity of temporal smoothing and spatial ensemble. When combined with temporal smoothing, LW is capable to produce a representative teacher model as good as its fine-grained counterpart NW. Considering that the time cost of LW is lower than both CW and NW, yet their performances are close, we use LW as default.

### 4.4 Self-supervised evaluation of STS

In this part, we substitute the model smoothing mechanism in BYOL [2] with our STS and compare the resulted model to other state-of-the-art methods on self-supervised learning. Furthermore, we investigate the robustness of our method to corrupted data. Finally, we evaluate the transfer capability of our method on the object detection task.

**Comparison with State-of-the-art method.** Table 5 presents the linear evaluation results of our method compared with the state-of-the-art self-supervised method on ImageNet [36]. Using BYOL as the base model, STS yields competitive top-1 accuracies of 73.0% and 74.5% under 300 and 1000 epochs, outperforming previous state-of-the-art approaches. It is worth noting that our model is trained with a smaller batch size of 256, while BYOL requires 4096. According to the results in the original BYOL paper, when the batch size decreases from 4096 to 256, its top-1 accuracy drops around than 0.6%.

**Robust to corruptions on ImageNet-C.** We further evaluate the robustness of our method to data corruptions. Specifically, we select ImageNet-C [37], which is a standard benchmark dataset for evaluating the model robustness to common corruptions and perturbations, as our testbed. There are 15 different corruption types in ImageNet-C. Besides, each type has 5 corruption intensities. In the left part of Figure 5, we give some image examples of corruptions from ImageNet-C and the corresponding top-1 accuracy improvements of STS over TMA. Combining the right part of Figure 5, we observe that STS outperforms TMA under all the five corruption intensity settings and yields an improvement of around 1.5% in terms of the mean top-1 accuracy of 75 corruption combinations. Besides, considering a single corruption from these 75 combinations, STS leads to an improvement of around 0.5% $\sim$ 5.6%. These results demonstrate the superiority of our method in terms of the robustness to corruptions and further reveal the advantage of STS in enhancing the representative capability of the teacher model.

**Transfer learning in object detection.** We further evaluate the generalization ability of the learned representation using our approach for the more challenging object detection task. In Table 5, we compare our method with state-of-the-art self-supervised approaches. All the self-supervised models are trained for 200 epochs on the ImageNet dataset and then finetuned with the joint split of VOC2007 trainval and VOC2012 train. And their results are taken from SimSiam [38]. For fair comparison, we carefully follow the training and evaluation settings in SimSiam [38]. In more detail, we use the same detection codebase [3] as SimSiam [38] and MoCo [1]. The default detector is Faster-RCNN [39] with a backbone of R50-C4. We finetune all the layers and BN statistics for 24,000 iterations with 16 images per batch. The image scales is set to [480, 800] pixels in the training stage while 800
for inference. The initial learning rate is 0.02 and is decreased by a factor of 0.1 and 0.01 at when reaching 18,000 and 22,000 iterations. We can observe from Table 5 that the representation learned by our approach can be well transferred to downstream detection task and achieves comparable or superior performance than previous approaches, especially on the AP\textsubscript{75} metric.

<table>
<thead>
<tr>
<th>Method</th>
<th>Epoch</th>
<th>top-1</th>
<th>top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jigsaw [40]</td>
<td>90</td>
<td>45.7</td>
<td>-</td>
</tr>
<tr>
<td>InstDis [41]</td>
<td>200</td>
<td>56.5</td>
<td>-</td>
</tr>
<tr>
<td>BigBiGAN [42]</td>
<td>-</td>
<td>56.5</td>
<td>-</td>
</tr>
<tr>
<td>Local Agg. [43]</td>
<td>200</td>
<td>60.2</td>
<td>-</td>
</tr>
<tr>
<td>CPC v2 [25]</td>
<td>200</td>
<td>63.8</td>
<td>85.3</td>
</tr>
<tr>
<td>SimCLR [4]</td>
<td>200</td>
<td>66.6</td>
<td>-</td>
</tr>
<tr>
<td>MoCo v2 [44]</td>
<td>200</td>
<td>67.5</td>
<td>-</td>
</tr>
<tr>
<td>PCL [44]</td>
<td>200</td>
<td>67.6</td>
<td>-</td>
</tr>
<tr>
<td>InfoMin Aug [24]</td>
<td>200</td>
<td>70.1</td>
<td>89.4</td>
</tr>
<tr>
<td>Swav [3]</td>
<td>400</td>
<td>70.7</td>
<td>-</td>
</tr>
<tr>
<td>SimSiam [38]</td>
<td>400</td>
<td>70.8</td>
<td>-</td>
</tr>
<tr>
<td>BYOL [2]</td>
<td>300</td>
<td>72.5</td>
<td>90.8</td>
</tr>
<tr>
<td><strong>Ours</strong> (BYOL + STS)</td>
<td>300</td>
<td><strong>73.0</strong></td>
<td><strong>91.3</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Smoothing</th>
<th>top-1</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMA</td>
<td>-</td>
<td>71.8</td>
<td>1.0\times</td>
</tr>
<tr>
<td>SE LW</td>
<td>67.8</td>
<td>0.97\times</td>
<td></td>
</tr>
<tr>
<td>SE CW</td>
<td>70.8</td>
<td>1.02\times</td>
<td></td>
</tr>
<tr>
<td>SE NW</td>
<td>71.6</td>
<td>1.80\times</td>
<td></td>
</tr>
<tr>
<td><strong>STS</strong> LW</td>
<td>72.7</td>
<td>0.99\times</td>
<td></td>
</tr>
<tr>
<td><strong>STS</strong> CW</td>
<td>72.8</td>
<td>1.04\times</td>
<td></td>
</tr>
<tr>
<td><strong>STS</strong> NW</td>
<td>72.8</td>
<td>1.82\times</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of SE and STS with different granularities.

<table>
<thead>
<tr>
<th>pretrain</th>
<th>VOC07+12 detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AP\textsubscript{50}</td>
</tr>
<tr>
<td>supervised</td>
<td>81.3</td>
</tr>
<tr>
<td>SimCLR</td>
<td>81.8</td>
</tr>
<tr>
<td>MoCo v2</td>
<td>82.3</td>
</tr>
<tr>
<td>SwAV</td>
<td>81.5</td>
</tr>
<tr>
<td>SimSiam</td>
<td>82.4</td>
</tr>
<tr>
<td>BYOL</td>
<td>81.4</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>82.3</td>
</tr>
</tbody>
</table>

Table 5: Transfer learning on VOC object detection benchmark.

4.5 Semi-supervised Evaluation of STS

This part is to evaluate the performance of our STS when integrated with FixMatch [7] for semi-supervised learning. Following the common settings [6, 7], we use CIFAR-10 and CIFAR-100 as the benchmark datasets.

CIFAR-10. We evaluate our method on CIFAR-10 under four settings, i.e., using 2, 4, 25, 400 labels per class. As shown in Table 6, STS achieves consistently superior performance than TMA under these four different settings, yielding an improvement of 6.63\%, 1.83\%, 0.21\%, and 0.17\%,
Figure 5: Left: Examples of 4 corruption types, namely defocus blur, pixelate, fog, and spatter. The numbers attached to the left-bottom corner denote the improvement of STS over TMA on the corresponding corruption type and intensity in terms of top-1 accuracy. Right: Comparison of top-1 accuracy between TMA and STS on ImageNet-C.

Table 6: Comparison with state-of-the-art semi-supervised methods on the CIFAR-10 and CIFAR-100 benchmark datasets under the top-1 accuracy metric.

CIFAR-100. In Table 6 we further provide the results on CIFAR-100 benchmark dataset under four settings, i.e., using 2, 4, 25, and 100 labels per class. Similar to the phenomenon observed in CIFAR-10, STS achieves noticeable improvements when only a few labels (200/400 labels) are given, outperforming the baseline by 3.07% and 2.71%. Besides, with more labels (2500/10000 labels), STS also achieves consistent improvements of 1.47% and 1.39%, demonstrating the effectiveness of our approach.

5 Conclusion

This paper reveals a new mechanism for model smoothing from the spatial ensemble perspective. In addition to the capacity of model smoothing, an important advantage of spatial ensemble is its complementarity to the popular temporal moving average. To benefit from such complementarity, we integrate temporal moving average and spatial ensemble into a unified model smoothing method named Spatial-Temporal Smoothing (STS). Extensive experiments on self-supervised and semi-supervised tasks show that STS brings general improvement to many state-of-the-art methods. Besides, STS achieves competitive results on the ImageNet-C dataset and the downstream object detection task, demonstrating its good generalization ability.
Acknowledgment

This paper is supported by the National Key R&D Plan of the Ministry of Science and Technology (Project No. 2020AAA0104400).

References


