ATSO: Asynchronous Teacher-Student Optimization for Semi-Supervised Image Segmentation

Xinyue Huo[1,2] Lingxi Xie[2] Jianzhong He[2] Zijie Yang[2,3]
Wengang Zhou[1] Houqiang Li[1] Qi Tian[2]
[1]University of Science and Technology of China, [2]Huawei Inc.
[3]Chinese Academy of Sciences
xinyueh@mail.ustc.edu.cn 198808xc@gmail.com jianzhonghe@pku.edu.cn
yangzijie@ict.ac.cn {zhwg,lihq}@ustc.edu.cn tian.qi1@huawei.com

Abstract

*Semi-supervised learning is a useful tool for image segmentation, mainly due to its ability in extracting knowledge from unlabeled data to assist learning from labeled data. This paper focuses on a popular pipeline known as self-learning, where we point out a weakness named **lazy mimicking** that refers to the inertia that a model retains the prediction from itself and thus resists updates. To alleviate this issue, we propose the Asynchronous Teacher-Student Optimization (**ATSO**) algorithm that (i) breaks up continual learning from teacher to student and (ii) partitions the unlabeled training data into two subsets and alternately uses one subset to fine-tune the model which updates the labels on the other. We show the ability of ATSO on medical and natural image segmentation. In both scenarios, our method reports competitive performance, on par with the state-of-the-arts, in either using partial labeled data in the same dataset or transferring the trained model to an unlabeled dataset.*

1. Introduction

Semantic segmentation plays an important role in image understanding. Recently, the fast development of deep learning [17] provides a powerful tool for dense image prediction [6, 20], but for many scenarios such as medical image analysis, data annotation is often expensive but there may exist abundant unlabeled data. In addition, it is a common requirement of transferring a segmentation model from one domain to another without extra annotations. Both scenarios fall into the area of *semi-supervised learning* which focuses on learning from both labeled data and unlabeled data while the labeled part is often smaller. An effective pipeline is known as *self-learning*, in which an initial model is trained on the labeled part (training set) and fine-tuned on the unlabeled part (reference set) with the pseudo labels generated by itself. We refer to this pipeline as *teacher-student optimization*, a variant of knowledge distillation [14] that has straightforward applications on medical image analysis [47].

However, we notice a factor that harms the efficiency of utilizing unlabeled data. In the self-learning procedure, the similarity between the teacher and student, two variants of the target model, tend to increase. Consequently, the supervision that the student model obtains from the pseudo labels becomes weak and the learning process quickly arrives at a plateau. We call this phenomenon **lazy mimicking**: the teacher model stores knowledge in the pseudo labels for the student model to learn; once a prediction error appears, it is likely to persist throughout the self-learning procedure; therefore, inaccuracy accumulates and finally downgrades the quality of the generated pseudo labels. We find that lazy mimicking quantitatively reflects in that the pseudo labels are not improved during the learning process – in other words, *the accuracy on the reference set stops growing but the model itself does not know*. From the viewpoint of optimization, lazy mimicking is caused by the self-learning process gradually pushing the teacher and student models, as a whole, towards a local optimum.

To break up the optimization trap and alleviate lazy mimicking, we present the **asynchronous teacher-student optimization** (ATSO) algorithm. ATSO puts forward two simple modifications beyond the self-learning pipeline to break up the chain of 'error inheritance'. **First**, we switch off continual learning and start each generation from the same initialized model. **Second**, we prevent using the pseudo labels generated by a teacher model to supervise its direct student, which involves partitioning the reference set into two subsets – in each round of teacher-student optimization, we generate the pseudo labels on any subset based on a teacher model that was not trained on the same set of data. As we

shall see in experiments, both strategies are helpful to improve the quality of the pseudo labels and, consequently, boost the final segmentation accuracy.

We evaluate ATSO on two kinds of segmentation data, medical images and autonomous driving images. For medical analysis, we use the NIH and MSD datasets for pancreas segmentation from CT scans. ATSO shows promising segmentation results using 10% or 20% of labeled data of NIH, surpassing the previous state-of-the-arts and approaching the fully-supervised upper-bound. ATSO also works well in transferring a model trained on NIH to MSD that is completely unlabeled. For autonomous driving, two popular datasets named Cityscapes and Mapillary are investigated. Compared to the state-of-the-arts that used strong data augmentations on Cityscapes, ATSO produces competitive segmentation accuracy with just basic-level augmentations. In transferring a model from Cityscapes and Mapillary, ATSO makes use of super-class pseudo labels to avoid the instability of training, and achieves satisfying results.

In summary, the contribution of this paper is two fold. **First**, this is the first work to reveal the lazy mimicking phenomenon in the self-learning pipeline. **Second**, the ATSO pipeline is presented that alleviates the above burden and improves semi-supervised image segmentation. **Third**, the idea of super-class pseudo labels is helpful to stabilize knowledge distillation in multi-class segmentation tasks.

2. Related Work

Image segmentation is a fundamental task in computer vision. Recently, with the fast development of deep neural networks [16, 29, 13], researchers developed effective algorithms [20, 45, 6] for natural image segmentation. These techniques quickly propagated to the area of medical images [27, 22]. One of the major differences between natural and medical images lies in the dimensionality, where researchers have investigated 2D-based [27, 28, 48, 43, 21] and 3D-based [7, 22, 50] pipelines and tried to integrate them into one framework to absorb benefits from both of them [33, 19, 24, 37].

Semi-supervised learning lies between supervised and unsupervised learning, which assumes that a small fraction of data are labeled, while the remaining part are unlabeled but closely related to the labeled subset [4, 49]. Researchers designed some generalized frameworks including self-learning [1], multi-view learning [32, 40], co-training [4], *etc*. The idea of **self-learning** is to use an initial model trained on labeled data to predict the labels on unlabeled data, so that these labels, though less accurate, can be used for training an updated, more powerful model [1]. This is related to knowledge distillation [14] and teacher-student optimization [12], but since unlabeled data was introduced, it is crucial to maximally improve the quality of the predicted labels [34, 41]. The idea of self-learning is also widely used for natural image recognition. [3] obtained the final prediction by averaging the representation of multiple transformation from one image. [38], [2], and [30] injected noise into the network training process with different degrees of data augmentation to enhance the robustness of the model and further improve the reliability of pseudo labels. [39] only introduced noise into the student model to highlight the inconsistency between the teacher and student models and prevent the iterative process from moving towards a local optimum.

As another line of research, both **co-training** and **multi-view learning** aim to use the consistency within the task itself to assist learning. Differently, co-training often assumed that different models should produce the same output on the same data [26], but multi-view learning assumed that the same model should produce the same result on various views of the same data [32]. Sometimes, these assumptions were combined into one framework [31, 26]. Semi-supervised learning is of great interest to the researchers of medical image analysis, mainly because accurate annotations are often difficult to acquire. There exists large-scale datasets with inaccurate [35] and/or partial data annotations [46], and researchers also developed practical semi-supervised algorithms for learning from these data [47, 36].

ATSO aims at improving the quality of 'pseudo labels' in the self-training pipeline of semi-supervised learning. The key principle is to enlarge the difference between the teacher and student signals so that the student model learns non-trivial knowledge from the teacher model. A similar idea was presented by a recent work [42] which studied fully-supervised learning tasks. Differently, [42] facilitates the difference by manipulating learning rates, while ATSO by isolating reference data between iterations. ATSO is also related to other knowledge distillation approaches which trained a few models simultaneously so that each model can be used to supervise others. Examples include deep mutual learning [44] and deep co-training [26]. In particular, deep co-training [26] added an adversarial loss term to enlarge the gain between teacher and student models. Differently, we train two models individually on two subsets of the reference set, which naturally guarantees diversity and enjoys the ability of being parallelized when the number of individually-optimized models is large.

3. Our Approach

3.1. Problem Setting and Baselines

We study the problem of image segmentation. The goal is to train a segmentation model, $\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$, in which $\mathbf{x}$ and $\mathbf{y}$ denote the input images and output predictions and $\mathbf{f}(\cdot)$ is the deep network parameterized by $\boldsymbol{\theta}$. There are some off-the-shelf choices of $\mathbf{f}(\cdot)$ including FCN [20] and DeepLab [6]. Both FCN and DeepLab process 2D images.

In case of analyzing 3D medical data, *e.g.*, CT scans, we follow a practical pipeline, RSTN [43], that partitions each 3D volume into 2D slices, known as the *coronal*, *sagittal*, and *axial* views, uses a 2D network for segmentation, and stacks the outputs into a 3D volume as the final prediction. The accuracy of image segmentation is often measured by the overlapping ratio between the prediction and ground-truth. Throughout the rest part of this paper, we use the term of *segmentation accuracy* or briefly *accuracy* to refer to the mean IOU value for the natural images and the Dice-Sørensen coefficient (DSC) for the medical images. Provided $\mathcal{Y}$ and $\mathcal{Z}$ being the predicted and ground-truth set of object pixels, we have $\text{IOU} = |\mathcal{Y} \cap \mathcal{Z}| \,/\, |\mathcal{Y} \cup \mathcal{Z}|$ and $\text{DSC} = 2 \times |\mathcal{Y} \cap \mathcal{Z}| \,/\, (|\mathcal{Y}| + |\mathcal{Z}|)$, respectively.

Besides the fully-supervised image segmentation on the fully annotated data, another important setting is semi-supervised learning in which a large portion of training data do not have labels but we need to learn as much knowledge as possible from them. As a formal definition, the training set $\mathcal{T}$ is partitioned into two parts, namely, the supervised (labeled) set $\mathcal{S}$ and the reference (unlabeled) set $\mathcal{R}$. Most often, we have $|\mathcal{S}| \ll |\mathcal{R}|$. Also, there is a testing set, $\mathcal{E}$, which is invisible during the model training procedure.

3.2. Lazy Mimicking: the Devil in Self-Learning

For the simplicity of description, throughout this section, we illustrate our approach using the scenario of medical image segmentation – all data are from the NIH pancreas segmentation dataset [28] and all results are from RSTN [43], while almost all our statements can be directly transplanted to natural image segmentation.

We first show that image segmentation accuracy can drop dramatically in the semi-supervised setting. We start with training the model using 10% of data labeled and the remaining 90% unlabeled on the NIH dataset. The detailed configurations are elaborated in the experimental section. Although RSTN achieves an average accuracy of over 84% with full supervision (60 training samples) on the NIH dataset, but if only 10% of training data (6 training samples) are preserved, the accuracy quickly drops to nearly 70%.

A simple and effective pipeline for semi-supervised learning is named self-learning. It starts with an initial model, denoted by $\mathbb{M}_0$, which is trained under supervised learning on $\mathcal{S}$. $\mathbb{M}_0$ gets updated for a total of T rounds. In the t-th ($t = 1, 2, \ldots, T$) round (*a.k.a.*, generation), the reference subset, $\mathcal{R}$, is sent into the old model $\mathbb{M}_{t-1}$ (often referred to as the teacher model), and the prediction is named the pseudo label in the current round. The training process of the student model, $\mathbb{M}_t$, then follows a regular supervised learning procedure on both $\mathcal{S}$ and $\mathcal{R}$, with the supervision on $\mathcal{R}$ coming from the pseudo labels. The trained student model of the current round is used as the teacher model of the next round and the iteration continues till the end.

However, the results are below satisfaction. Self-learning gets saturated after 2 generations, when it achieves an accuracy of 78.98% on the test set. Although the accuracy is significantly higher than the base model, it is still far away from the fully-supervised upper-bound which is over 84%. In other words, self-learning extracts knowledge from the reference set, but the efficiency is below satisfaction.

To investigate the reason, we diagnose the accuracy of the reference set, which is expected to grow with the learning procedure. However, we find that the accuracy quickly arrives at a plateau. After the 0th (the initial training stage with only labeled data), 1st, 2nd, and 3rd generations, the accuracy of the reference set is 71.41%, 74.54%, 75.44%, and 74.72%, respectively. Compared to the scenario when 100% training data are labeled, the accuracy on the same subset of training data is 86.70%. In other words, the training procedure has entered a 'trap' that the pseudo labels of the reference set stop at a low accuracy (what is worse, the accuracy starts to drop in the 3rd generation), but the algorithm 'does not know' because the ground-truth is missing.

From the viewpoint of optimization, this phenomenon can be explained as a local optimum of the self-learning system. Let $\mathbf{x}_n$ be a training sample in the reference set, $\mathcal{R}$, $\mathbf{y}_n^\star$ is the ground-truth label, and $\mathbf{y}_n$ is the predicted output by the teacher model. We assume that $\mathbf{y}_n = \mathbf{y}_n^\star + \boldsymbol{\epsilon}_n$ where $\boldsymbol{\epsilon}_n$ is the prediction error. When $\mathcal{R}$ is labeled, *i.e.*, $\mathbf{y}_n^\star$ is known, $\boldsymbol{\epsilon}_n$ follows a zero-mean distribution because the optimization goal is to minimize $|\boldsymbol{\epsilon}_n|$ on the training set. However, in the scenario of self-learning, $\mathbf{y}_n^\star$ remains unknown and thus $\boldsymbol{\epsilon}_n$ may follow a non-zero-mean distribution. Since $\mathbf{y}_n$ is used as the pseudo label, such noise can persist across in the student model. What makes things worse, each generation of the teacher-student optimization can introduce new noise which accumulate on the reference set. We use the name of **lazy mimicking** to refer to the behavior that ***the student model is unable to identify and eliminate the noise of the teacher model***. We show a typical example of lazy mimicking in Figure 1.

3.3. Asynchronous Optimization: Escaping from the Optimization Trap

To alleviate lazy mimicking, *i.e.*, escaping from the optimization trap, we point out two key factors that assist the propagation of the noise, $\boldsymbol{\epsilon}_n$: (i) the pre-trained weights of the current snapshot from the teacher model and (ii) the reference set that has just been used by the teacher model. This inspires us to weaken the correlation between the teacher and student models from these two aspects.

First, we weaken the correlation from the **model** perspective, namely, preventing the teacher and student model from being too close. This is easily implemented by breaking up the setting of continual learning and initializing each student model from the same checkpoint – in practice, we

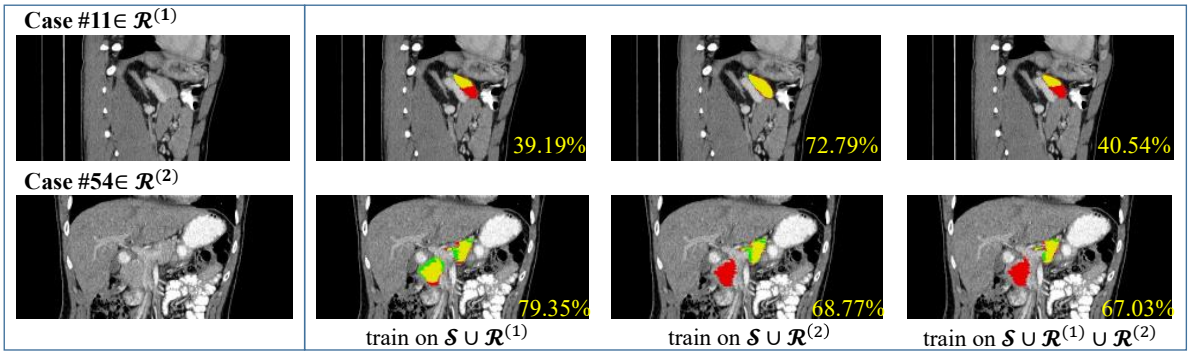

Figure 1. Examples of lazy mimicking and a simple method to alleviate it *(best viewed in color)*. The leftmost column shows two unlabeled images in the reference set, and the right columns show the segmentation results when the full reference set has been used for self-learning and when half of the reference set has been used. Segmentation accuracy is significantly improved when the reference set does not contain the test case. The red, green, and yellow masks indicate the true label, prediction, and overlapping region, respectively. The numbers in the bottom-right corner denote the DSC accuracy of the entire 3D volume.

use the first model optimized on the labeled training set, $\mathcal{S}$, to be the default checkpoint. As shown in the experiments, this simple modification brings significant accuracy gain.

Second, we weaken the correlation from the **data** perspective, namely, preventing using the same set of reference data continuously, *i.e.*, always using the pseudo labels generated by a teacher model to supervise its direct student. To verify this assumption, we first notice that in the aforementioned self-learning procedure, the accuracy gained on the test set is much higher than that on the reference set, *e.g.*, 6.36% test accuracy gain vs. 4.03% reference accuracy gain after 2 self-learning generations. To make things clearer, we partition the reference set into two parts, and perform self-learning on different combinations of reference data. Results of two hard examples are shown in Figure 1. *Interestingly, the segmentation accuracy is significantly improved when the example is not contained in the reference set.* This aligns with the observation that the improvement on the test set is larger than that on the reference set. Back to the optimization perspective, *the noise on the reference set does not transfer to the data that the current generation does not use for self-learning*. This inspires us to partition the reference set into two subsets, denoted by $\mathcal{R} = \mathcal{R}^{(1)} + \mathcal{R}^{(2)}$. In each generation, we generate the pseudo labels on $\mathcal{R}^{(1)}$ using the model that was just self-trained on $\mathcal{R}^{(2)}$, and vice versa. After the last generation asynchronous update of both subsets, we combine the pseudo labels of both subsets into the complete one, based on which the final model is trained.

Integrating the above two aspects obtains the **asynchronous teacher-student optimization** (ATSO) algorithm, described in Algorithm 1. Compared to the self-learning baseline, each generation of ATSO takes approximately the same computational costs. This makes ATSO easily plugged into any self-learning scenarios.

Of course, there may exist other solutions that alleviate lazy mimicking, yet our solution is simple and effective. Moreover, ATSO also has the option of dividing the reference set into more subsets, but this can slow down the training procedure. In practice, we find that using two folds of reference data performs well for semi-supervised learning.

Algorithm 1: ATSO
Asynchronous Teacher-Student Optimization

Input : a training set $\mathcal{T} = \mathcal{S} \cup \mathcal{R}$ ($\mathcal{S}$ is labeled and $\mathcal{R}$ is unlabeled), # of iterations T;
Output: a model $\mathbb{M}$ trained on $\mathcal{T}$;

Train an initial model $\mathbb{M}_0$ *from scratch* on $\mathcal{S}$;
Divide $\mathcal{R}$ into two subsets, $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$, $t \leftarrow 0$;
$\mathbb{M}_0^{(1)} \leftarrow \mathbb{M}_0$, $\mathbb{M}_0^{(2)} \leftarrow \mathbb{M}_0$;
repeat
  Update $\mathcal{R}_{t+1}^{(1)}$ using the prediction of $\mathbb{M}_t^{(2)}$;
  Update $\mathcal{R}_{t+1}^{(2)}$ using the prediction of $\mathbb{M}_t^{(1)}$;
  $\mathbb{M}_{t+1}^{(1)} \leftarrow \mathbb{M}_0$, $\mathbb{M}_{t+1}^{(2)} \leftarrow \mathbb{M}_0$;
  Fine-tune $\mathbb{M}_{t+1}^{(1)}$ on $\mathcal{S} \cup \mathcal{R}_{t+1}^{(1)}$;
  Fine-tune $\mathbb{M}_{t+1}^{(2)}$ on $\mathcal{S} \cup \mathcal{R}_{t+1}^{(2)}$;
  $t \leftarrow t + 1$;
until $t = T$;
Train $\mathbb{M}_T$ on $\mathcal{S} \cup \mathcal{R}_t^{(1)} \cup \mathcal{R}_t^{(2)}$;
Return: $\mathbb{M} \leftarrow \mathbb{M}_T$.

4. Experiments on Medical Images

This section demonstrates the effectiveness of ATSO in two medical segmentation datasets. We investigate two scenarios, namely, semi-supervised segmentation (with part of training data labeled) on a single dataset and transferring the trained model from one dataset to another that does not have labels at all.

4.1. Results on the NIH Dataset

We evaluate ATSO on the NIH dataset [28] for pancreas segmentation. It contains 82 normal CT scans, each of which is a 3D volume of $512 \times 512 \times L$ voxels, where L is the length of the long axis. We follow the prior work [48, 43] to partition each dataset into four folds and use the first three folds (62 cases) as the training data. For semi-supervised learning, we follow the prior work [47, 36] to use a small portion (10% or 20%) of training data as the labeled set and leave the remaining part to the reference set. We report the average DSC over all test cases.

The configuration of the deep networks follows that in the original RSTN paper [43]. In the training stage, we optimize three individual networks for segmentation along with the *coronal*, *sagittal* and *axial* views, respectively. In the inference stage, either on the reference set or the test set, predictions from these three views are fused into the final segmentation by majority voting. Please refer to the original paper for further technical details.

We first investigate semi-supervised segmentation on the NIH dataset with 10% of the training set (6 cases) labeled. The naive baseline, by only using the labeled data for training, reports a 72.62% test accuracy which is far lower than the upper-bound, 85.04%, when the labels of all training data are available. In what follows, we gradually add the key components of ATSO into the baseline and show how these components push the training procedure towards higher accuracy.

- **Breaking up Continual Learning Brings Benefits**

We first compare the options with and without continual learning, which we refer to as the **self-learning** baseline and **synchronous teacher-student optimization** (STSO), respectively. The former uses the last snapshot of the teacher model to initialize the student model, while the latter fine-tunes the student model from $\mathbb{M}_0$. To save computational costs, we only perform the third training stage of RSTN beyond initialization. Note that the major difference between the self-learning baseline and STSO lies in whether continual learning is used – we use the comparison to reveal the relationship between continual learning and 'lazy mimicking' and thus offer a new understanding to the prior work that either used continual learning or not.

Results are summarized in Table 1. The difference between the self-learning baseline and STSO is significant. After two generations, the self-learning baseline achieves a 78.98% accuracy on the testing set and 75.44% on the reference set. Starting from the third generation, these numbers start to drop, demonstrating that lazy mimicking has obstructed the model from obtaining useful information and conducted a fallacious direction to the student model. The best accuracy of STSO, 79.67%, (a non-trivial 0.69% improvement) is obtained after 4 generations.

- **Asynchronous Optimization Improves Accuracy**

Generation	Self-learning		STSO		ATSO	
	$@\mathcal{R}$	$@\mathcal{E}$	$@\mathcal{R}$	$@\mathcal{E}$	$@\mathcal{R}$	$@\mathcal{E}$
G0	71.41	72.62	71.41	72.62	71.41	72.62
G1	74.54	76.82	74.54	76.82	75.69	78.82
G2	75.44	78.98	75.42	77.88	77.05	80.81
G3	74.72	78.27	76.42	79.27	77.81	**81.69**
G4	74.38	77.78	76.93	79.67	77.73	81.41
G5	73.38	77.22	77.15	79.57	**78.07**	81.57

Table 1. Segmentation results (DSC, %) on the NIH pancreas segmentation datasets with 10% labeled training data (6 cases). The results of the reference set and the test set are compared during 5 generations.

Next, we study the difference between STSO and ATSO. Results are summarized in Table 1. ATSO improves segmentation accuracy on both the reference and test sets. Interestingly, ATSO enjoys faster growth in both numbers: after only two generations, the test accuracy has increased to over 80%, claiming a nearly 3% advantage over the corresponding number of STSO. After five generations, ATSO still enjoys a 2% advantage over STSO. That being said, ATSO has a broad range of applications in the scenario of limited computational resource for model training.

To quantify the impact of lazy mimicking, we refer to the DSC between the final model and the model trained only on labeled data, which is 84.40%, 83.47%, and 81.13% for the self-learning baseline, STSO, and ATSO, respectively, implying that 'teacher and student being too close' is a negative factor to the segmentation accuracy (77.22%, 79.57%, 81.57%). Therefore, *by not generating pseudo labels on the reference set that was just used*, the algorithm can escape from the optimization trap.

- **Comparison to State-of-the-Arts and Visualization**

In Table 2, we compare ATSO against state-of-the-art approaches, and show that ATSO outperforms all of them. In particular, ASTO surpasses [36] by more than 2.5% in both scenarios that 10% and 20% labeled data have been used. Note that [36] is a recently published method which involved uncertainty in multi-view learning – in comparison, our solution is easier and more effective. In addition, being simple and easily implemented, ATSO can be combined with other training strategies, *e.g.*, adversarial training [26] or uncertainty evaluation [36], towards better performance.

Figure 2 shows some typical examples of how segmentation errors are fixed with semi-supervised learning. When the labeled set is small, it is very likely that the labeled training set does cover sufficient situations, causing some failure cases in the reference set. In the self-learning baseline or even STSO, it is relatively difficult for the model to fix these errors during iteration, and the persisted errors can hinder the ability of the student model. In comparison, ATSO offers extra opportunities to jump out of the current distribution and thus get rid of the failure case. Hence, the efficiency of utilizing unlabeled training data is improved.

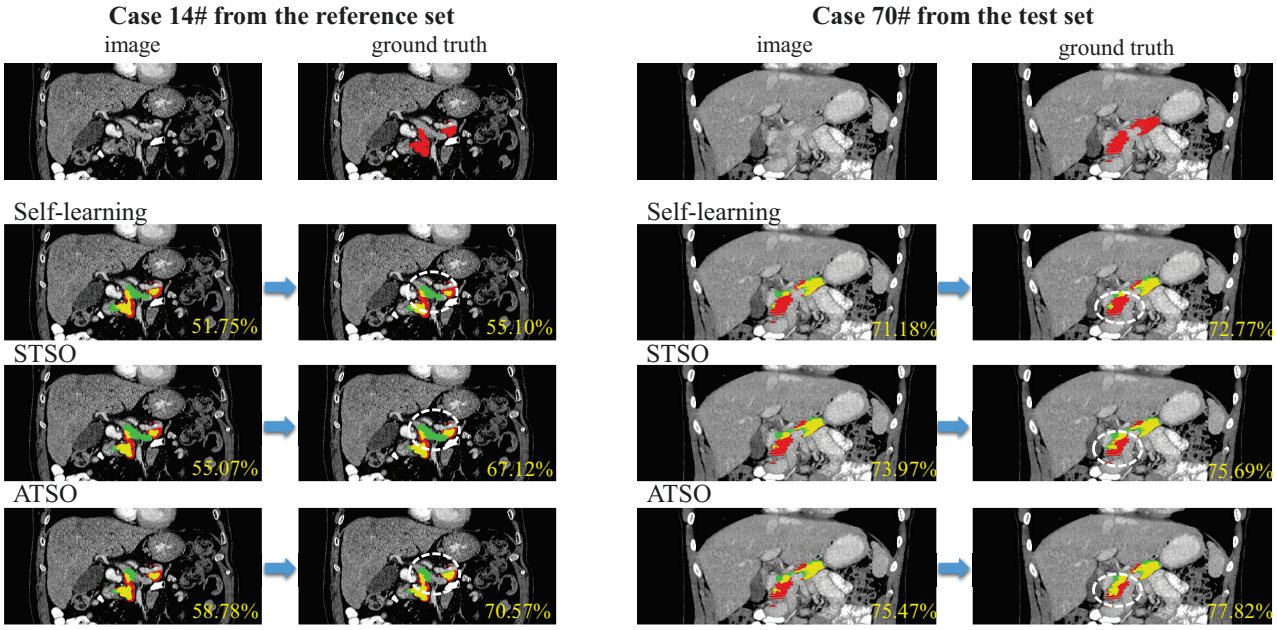

Figure 2. Visualization of the improvement on the reference and test sets (best viewed in color). For each case, the second to the last row show the results produced by the self-learning baseline, STSO, and ATSO, respectively. In each pair, the left and right sides of the arrow are the outputs of an intermediate and the final generations. We show typical 2D slices that reflect the difference, while the DSC numbers in the bottom-right corner are computed in the entire 3D volume. The red, green, and yellow masks indicate the true label, prediction, and overlapping region, respectively. Please also zoom in to see the white dashed circles that mark the regions with significant accuracy gain.

Method	Backbone	10% D	20% D
DMPCT [47]	2D ResNet-101	63.45	66.75
DCT [26] (2v)	3D ResNet-18	71.43	77.54
TCSE [18]	3D ResNet-18	73.87	76.46
UMCT [36] (2v)	3D ResNet-18	75.63	79.77
UMCT [36] (6v)	3D ResNet-18	77.87	80.35
UMCT [36] (2v+)	3D ResNet-18	77.78	80.52
UMCT [36] (3v+)	3D ResNet-18	79.05	81.18
Self-Learning (ours)	2D FCN8s ×2	78.98	82.87
STSO (ours)	2D FCN8s ×2	79.67	83.21
ATSO (ours)	2D FCN8s ×2	**81.69**	**83.70**

Table 2. Accuracy (DSC, %) comparison between some recently published methods and our solutions, *i.e.*, the self-learning baseline, STSO, and ATSO. We have tested the accuracy using either 10% or 20% labeled training data. Some of the numbers are borrowed from [36]. 2v means that 2 views have been used in multi-view learning and '+' means muli-view fusion results.

4.2. Transferring to the MSD Dataset

The pancreatic tumor segmentation task of the MSD dataset (http://medicaldecathlon.com/) has 281 abnormal CT scans. Each sample contains the annotation of the pancreas and the tumor. Since we train the base model on the NIH dataset (normal pancreas), we only evaluate the pancreas segmentation accuracy on the MSD data. We use 62 cases from the NIH dataset as the labeled training data while 200 cases from the MSD dataset as the unlabeled training data. The rest of the MSD dataset is considered as the test set. Transferring from NIH to MSD is a challenging task since the distributions are very different between these two datasets: the scanners are different, and the MSD data contain abnormality (pancreatic tumor) while the NIH data do not. In this scenario, the key is to extract information from the unlabeled training set that has a close distribution to the test set. Since the resolution on the long axis varies significantly, we normalize the inter-slice distance along the long axis during training and testing, but rescale the final output to the original size for a fair comparison. We also compute the global DSC criterion (following the MSD standard, merging all test cases into a single volume) which reports the same trend as local DSC criterion.

We compare the segmentation results on the reference and test sets among the self-learning baseline, STSO and ATSO. Results are summarized in Table 3. Directly applying the pre-trained model for segmentation reports an average accuracy of 69.95% on the test set. After four generations, ATSO reports an average accuracy of 76.71% (an 6.76% gain over the direct transfer baseline) which is higher than that of STSO (75.48%) and the self-learning baseline (74.78%) after the same generations. Note that 76.71% is even higher than the accuracy (75.49%) obtained from us-

Generation	Self-learning		STSO		ATSO	
	$@\mathcal{R}$	$@\mathcal{E}$	$@\mathcal{R}$	$@\mathcal{E}$	$@\mathcal{R}$	$@\mathcal{E}$
G0	63.75	69.95	63.75	69.95	63.75	69.95
G1	68.24	75.35	68.24	75.35	68.86	74.53
G2	69.72	75.42	68.50	75.09	69.17	76.57
G3	69.77	74.78	68.04	74.17	70.49	76.10
G4	70.21	74.03	69.08	75.48	**71.02**	**76.71**

Table 3. Segmentation results (DSC,%) on the transfer learning from NIH to MSD. The results of the reference set and the test set are compared during 4 generations.

ing 62 labeled MSD cases for training in which the gap between the training and the testing is smaller. Similar to the NIH experiments, we also obtain more accurate pseudo labels. After four generations, ATSO achieves a 71.67% accuracy on the reference set, but both the self-learning baseline and STSO reports around 70%. This aligns with our expectation: ATSO has the ability of transferring knowledge from a labeled dataset to another unlabeled, even when the data distributions differ considerably.

5. Experiments on Natural Images

This section generalizes ATSO to natural image segmentation[1], in particular, on autonomous driving data. Compared to medical images, these datasets contain more semantic classes. Similar to the experiments on medical data, we show the superiority of the proposed ATSO in two semi-supervised learning scenarios.

5.1. Results on the Cityscapes dataset

- **The Finely-Labeled Subset**

To compare against other methods, we first evaluate ATSO on the 2,975 finely-labeled images, where we use around 1/30, 1/8, and 1/4 labels and leave the remaining as the reference set. The baseline is chosen to be DeepLab v2 [5] built on RseNet-101 [13]. The backbone weights are pre-trained on ImageNet [9]. We train the network for 100 epochs with the pixel-wise cross-entropy as the loss function. The initial learning rate is set to be 0.01 and adjusted by the `poly` schedule. We use a batch size of 32 and distribute the training on eight NVIDIA V100 GPUs. The entire training procedure takes around 3.5 hours for each generation. During the training stage, the input images are randomly cropped and rescaled to 512×1024, and the standard data augmentation including horizontal flipping and Gaussian blur are used. During the testing stage, each input image is rescaled into half width and height and then fed to the trained network. The output is up-sampled to the original size for evaluation (the mIOU value is used).

[1]We also evaluate ATSO on the PASCAL VOC dataset [10] and achieve a similar conclusion. Please find the results in the Appendix.

Labeled Samples	100 (1/30)	372 (1/8)	744 (1/4)
Adv-Learning [15]	-	57.1	60.5
s4GAN [23]	-	59.3	61.9
CutMix [11]	51.2	60.3	**63.9**
ClassMix [25]	**54.1**	61.4	63.6
Self-learning	52.4	60.5	63.0
STSO (ours)	52.9	60.7	63.1
ATSO (ours)	53.1	**61.8**	63.2

Table 4. Segmentation accuracy (mIoU, %) on the Cityscapes validation set. The results of other methods are directly borrowed from the previous papers [11, 25].

The segmentation accuracy and the comparison to other approaches are summarized in Table 4. One can observe that both STSO and ATSO bring consistent accuracy gain over the self-learning baseline. In particular, ATSO achieves competitive performance which is on par with CutMix [11] and ClassMix [25] that used strong data augmentations. On the other hand, we point out that ATSO can be freely integrated into data augmentation to further improve the segmentation accuracy – please refer to the experiments in the Appendix.

During the experiments, we observe that due to the finely-labeled subset is relatively small, the semi-supervised segmentation performance is not that stable. This motivates us to evaluate ATSO in the full dataset.

- **The Full Dataset**

We perform experiments on the Cityscapes dataset using 1,000 finely-annotated training images as the labeled training set and the remaining part (21,944 images, including finely-labeled and coarsely-labeled images) used as the unlabeled reference set.

The IOU numbers of all classes and different approached are summarized in Table 5. When only the 1K labeled images are used for training, the mIOU on the validation set is 54.68%. After the unlabeled part is incorporated into the training process, much higher accuracy is achieved. The baseline self-learning reports an mIOU of 68.48%, and the STSO gets a similar accuracy of 68.72%. ATSO improves the accuracy to 70.43%.

The overall improvement of ATSO over the baseline that only uses labeled data is over 15%, and the improvement over either the self-learning baseline and STSO is around 2%. From Table 5, we notice that the improvement on the hard semantic classes is more significant. This delivers different messages compared to the medical image segmentation task. When the proportions of different object categories vary a lot, it is important to mine richer information of the objects with limited training instances (*e.g.*, *train* or *truck*) from the reference set to improve the segmentation accuracy. Due to the space limit, we put some visualization results in the Appendix.

Method	*road*	*side*	*budg*	*wall*	*fence*	*pole*	*tr lt*	*tr sn*	*vegtr*	*terr*	*sky*	*pers*	*rider*	*car*	*truck*	*bus*	*train*	*motor*	*bike*	**mIoU**
Supervised-only	96.15	72.94	86.60	30.17	37.85	33.60	38.27	54.37	88.24	50.20	90.41	64.34	30.64	87.71	27.18	41.42	25.13	22.37	61.49	54.68
Self-learning	97.60	81.12	90.53	45.02	51.58	**53.47**	60.56	73.12	91.38	55.67	94.35	77.70	53.51	92.56	47.60	59.24	48.30	55.37	72.64	68.48
STSO	**97.65**	**81.77**	**90.70**	**51.52**	51.64	53.45	**60.75**	**73.48**	**91.46**	58.46	**94.46**	**78.08**	**55.18**	92.55	45.30	59.68	43.45	53.17	**72.92**	68.72
ATSO	97.64	81.74	89.97	43.18	**52.06**	51.09	54.74	69.72	91.15	**59.95**	94.03	76.18	51.90	**93.27**	**68.28**	**74.67**	**61.54**	**56.07**	70.99	**70.43**

Table 5. Class-wise and mean IOU (%) of Cityscapes, produced by different training strategies. 'Supervised-only' indicates that only the 1K labeled images are used. Mind the significant gain in the *truck* and *train* classes.

Method	*road*	*side*	*budg*	*wall*	*fence*	*pole*	*tr lt*	*tr sn*	*vegtr*	*terr*	*sky*	*pers*	*rider*	*car*	*truck*	*bus*	*train*	*motor*	*bike*	**mIoU**
transfer	61.82	10.23	46.98	4.73	13.98	17.48	16.66	19.08	64.53	32.09	54.58	25.51	7.88	54.46	9.36	10.20	0.07	10.44	9.51	24.70
$STSO_{19}$	60.11	7.68	57.64	0.57	10.18	10.73	14.11	24.77	71.50	**44.88**	54.88	43.54	7.77	59.68	6.79	20.82	0.02	21.28	17.18	28.11
$ATSO_{19}$	62.08	9.64	**62.46**	0.90	8.79	6.84	15.90	26.02	71.50	35.48	**63.36**	45.61	15.89	66.44	5.83	15.89	0.05	14.47	27.41	28.26
$STSO_5$	69.87	9.60	40.31	14.42	14.78	**26.36**	20.90	21.49	83.28	25.94	27.15	51.27	9.11	80.14	11.60	11.15	0.43	22.16	25.48	29.76
$ATSO_5$	**80.65**	**12.07**	40.24	**16.67**	21.62	25.40	**31.88**	**35.51**	**86.07**	16.75	30.09	56.83	14.27	82.91	23.25	15.78	2.22	28.09	32.68	34.36
$ATSO_{5\to19}$	80.01	7.37	42.87	13.64	**24.69**	19.82	23.46	20.03	84.81	34.79	19.72	**60.86**	**29.43**	**83.20**	**24.56**	**27.12**	**15.64**	**42.38**	**49.17**	**37.23**

Table 6. Class-wise and mean IOU (%) of Mapillary, produced by different training strategies. Please refer to the texts for details.

5.2. Transferring to the Mapillary Dataset

The transfer learning scenario is defined between two natural image segmentation datasets. We use two popular datasets for autonomous driving, *i.e.*, training the model on Cityscapes [8] and transferring it to Mapillary (https://www.mapillary.com/). The labeled training set contains 2,975 finely annotated images from Cityscapes and the reference and test sets are from Mapillary, which have 18,000 and 2,000 images, respectively. Note that Cityscapes has 19 semantic classes while Mapillary has 66. We train a model to infer 19 classes and evaluate it on the reduced ground-truth on Mapillary by map each of the 66 classes to one of the 19 classes. The direct transfer method reports a mIOU of 24.70% on the Mapillary test set that is dramatically lower than the number (77.86%) on the Cityscapes test set, revealing a significant distribution gap between these two datasets.

We directly apply STSO and ATSO in this scenario, and obtain mIOU values of 28.11% and 28.26% (see Table 6), respectively. ATSO does not show advantages over STSO. This is mainly due to the unsatisfying accuracy of the teacher model (directly transferred from another dataset). For example, for the objects with a small ground-truth area (*e.g.*, a *train*), the teacher model (not seeing the ground-truth) can easily ignore the entire object or consider it as another close category so that the pseudo label will guide the student models to make the same critical mistake.

To alleviate this issue, we propose a solution named **super-class pseudo label generation** that generates relatively coarse pseudo labels in the super-class level to get rid of the trouble. For this purpose, we perform a predefined mapping to reduce the number of classes from 19 to 5 (*i.e.*, *rode*, *vehicle*, *person*, *vegetation*, *others*, referred to as the super-classes) so as to reduce the risk of missing some class completely. In this reduced, 5-class segmentation task, ATSO reports a mIOU of 77.11% which is significantly higher than the mIOU produced by STSO, 69.94%.

We then use the 5-class pseudo label, generated by the final generation of either STSO or ATSO, to guide the target network. During the training procedure, the target network generates 19-class segmentation and then reduces it to 5 classes for computing the loss function with respect to the pseudo labels. With only one training generation, the target model produces mIOU of 29.76% and 34.36% using the pseudo labels from STSO and ATSO, respectively, *i.e.*, better pseudo labels lead to higher mIOU. Interestingly, when we continue fine-tuning the latter model (a 34.36% mIOU) with 19-class pseudo labels, its performance is further boosted to 37.23%, claiming an over 12% gain beyond the direct transfer baseline (24.70%). Class-wise IOU numbers are detailed in Table 6. These results provide new insights to apply semi-supervised segmentation to the challenging datasets, in particular, with difficult semantic classes that can be easily missed. More details and results can be found in the Appendix.

6. Conclusions

In this paper, we investigate semi-supervised image segmentation using teacher-student optimization. The core discovery is that the self-learning process can fall into a trap named lazy mimicking which downgrades the quality of prediction in the reference set. To alleviate this issue, we propose a simple yet effective pipeline named **asynchronous teacher-student optimization** (ATSO) which (i) switches off continual learning and (ii) avoids any unlabeled sample to be used in two consecutive fine-tuning rounds. Experiments on a few public datasets verify the effectiveness of our approach in both intra-dataset and inter-dataset semi-supervised learning tasks.

Our research sheds light on a new direction to improve semi-supervised learning, *i.e.*, design a better schedule of feeding unlabeled data to the model. We expect to generalize ATSO to a wider range of vision problems, *e.g.*, simultaneous detection and segmentation.

References

[1] 2

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019. 2

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 2

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 7

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1, 2

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 2

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[11] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, high-dimensional perturbations. *arXiv preprint arXiv:1906.01916*, 2019. 7

[12] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 7

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[15] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018. 7

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 1

[18] Xiaomeng Li, Lequan Yu, Hao Chen, Chi-Wing Fu, and Pheng-Ann Heng. Semi-supervised skin lesion segmentation via transformation consistent self-ensembling model. *arXiv preprint arXiv:1808.03887*, 2018. 6

[19] Siqi Liu, Daguang Xu, S Kevin Zhou, Olivier Pauly, Sasa Grbic, Thomas Mertelmeier, Julia Wicklein, Anna Jerebko, Weidong Cai, and Dorin Comaniciu. 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 851–858. Springer, 2018. 2

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 2

[21] Yunze Man, Yangsibo Huang, Junyi Feng, Xi Li, and Fei Wu. Deep q learning driven ct pancreas segmentation with geometry-aware u-net. *IEEE transactions on medical imaging*, 38(8):1971–1980, 2019. 2

[22] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016. 2

[23] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 7

[24] Tianwei Ni, Lingxi Xie, Huangjie Zheng, Elliot K Fishman, and Alan L Yuille. Elastic boundary projection for 3d medical image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2109–2118, 2019. 2

[25] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. *arXiv preprint arXiv:2007.07936*, 2020. 7

[26] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the european conference on computer vision (eccv)*, pages 135–152, 2018. 2, 5, 6

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[28] Holger R Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim B Turkbey, and Ronald M Summers. Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 556–564. Springer, 2015. 2, 3, 5

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[30] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 2

[31] Ricardo Teixeira Sousa and Joao Gama. Comparison between co-training and self-training for single-target regression in data streams using amrules. 2017. 2

[32] Shiliang Sun. A survey of multi-view machine learning. *Neural computing and applications*, 23(7-8):2031–2038, 2013. 2

[33] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016. 2

[34] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 2

[35] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 2

[36] Yingda Xia, Fengze Liu, Dong Yang, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, and Holger Roth. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3646–3655, 2020. 2, 5, 6

[37] Yingda Xia, Lingxi Xie, Fengze Liu, Zhuotun Zhu, Elliot K Fishman, and Alan L Yuille. Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 445–453. Springer, 2018. 2

[38] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2

[39] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2

[40] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013. 2

[41] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5628–5635, 2019. 2

[42] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019. 2

[43] Qihang Yu, Lingxi Xie, Yan Wang, Yuyin Zhou, Elliot K Fishman, and Alan L Yuille. Recurrent saliency transformation network: Incorporating multi-stage visual cues for small organ segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8280–8289, 2018. 2, 3, 5

[44] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 2

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[46] Yuyin Zhou, Zhe Li, Song Bai, Chong Wang, Xinlei Chen, Mei Han, Elliot Fishman, and Alan L Yuille. Prior-aware neural network for partially-supervised multi-organ segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10672–10681, 2019. 2

[47] Yuyin Zhou, Yan Wang, Peng Tang, Song Bai, Wei Shen, Elliot Fishman, and Alan Yuille. Semi-supervised 3d abdominal multi-organ segmentation via deep multi-planar co-training. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 121–140. IEEE, 2019. 1, 2, 5, 6

[48] Yuyin Zhou, Lingxi Xie, Wei Shen, Yan Wang, Elliot K Fishman, and Alan L Yuille. A fixed-point model for pancreas segmentation in abdominal ct scans. In *International conference on medical image computing and computer-assisted intervention*, pages 693–701. Springer, 2017. 2, 5

[49] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 2

[50] Zhuotun Zhu, Yingda Xia, Wei Shen, Elliot K Fishman, and Alan L Yuille. A 3d coarse-to-fine framework for automatic pancreas segmentation. *arXiv preprint arXiv:1712.00201*, 2, 2017. 2