S2C2 - AN ORTHOGONAL METHOD FOR SEMI-SUPERVISED LEARNING ON AMBIGUOUS LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Semi-Supervised Learning (SSL) can decrease the required amount of labeled image data and thus the cost for deep learning. Most SSL methods assume a clear distinction between classes, but class boundaries are often ambiguous in real-world datasets due to intra- or interobserver variability. This ambiguity of annotations must be addressed as it will otherwise limit the performance of SSL and deep learning in general due to inconsistent label information. We propose Semi-Supervised Classification & Clustering (S2C2) which can extend many deep SSL algorithms. S2C2 automatically estimates the ambiguity of an image and applies the respective SSL algorithm as a classification to certainly labeled data while partitioning the ambiguous data into clusters of visual similar images. We show that S2C2 results in a 7.6% better F1-score for classifications and 7.9% lower inner distance of clusters on average across multiple SSL algorithms and datasets. Moreover, the output of S2C2 can be used to decrease the ambiguity of labels with the help of human experts. Overall, a combination of Semi-Supervised Learning with our method S2C2 leads to better handling of ambiguous labels and thus real-world datasets.

1 INTRODUCTION

In recent years, Semi-Supervised Learning (SSL) has shown great potential in solving one of the main issues in deep learning for image classification: the required amount of labeled image data and the high cost associated with the labeled data generation. By leveraging unlabeled data, the amount of labeled data and its labeling cost can be decreased to 10% or even 1% while maintaining classification performance (35; 19; 8; 42; 6) or even boost performance further (40; 29) on already large labeled datasets like ImageNet (22).

However, these successes have been achieved on curated benchmark datasets where the labels have been manually cleaned (21). We focus in this paper on how to apply SSL to new uncurated and unlabeled datasets and the corresponding issues. When we annotate new data, we will often encounter a variability / inconsistency between the annotations of different annotators or over time. This issue is called *intra- or interobserver variability* (IIV) and is a common issue when annotating data (28; 18; 31; 33; 16; 26; 4; 11; 17; 10). The literature names different possible reasons for this variability such as low resolution (28), bad quality(14; 33), subjective interpretations of classes (17; 26) or mistakes (18; 25).

We assume that this variability can be modeled for each image with an unknown soft probability distribution $l \in [0, 1]^k$ for a classification problem with k classes. We call a label and its corresponding image *certain* if all annotators would agree on the classification ($l \in \{0, 1\}$) and *ambiguous* if they would disagree ($l \in (0, 1)$). In other words, ambiguous images are likely to have different annotations due to IIV while certain images do not. The issue of ambiguous images is that the unknown distribution l can only be estimated with expensive operations like actually acquiring multiple annotations. Real-world example images with certain and ambiguous labels are given in Figure 3 and detailed definitions are given in subsection 2.1.

The question is how to apply SSL to a dataset with expected IIV and thus ambiguous labels. The better we approximate the unknown label l the more annotations / annotation time we need and thus we negate the desirable benefit of SSL to require fewer labeled samples. However, low quality

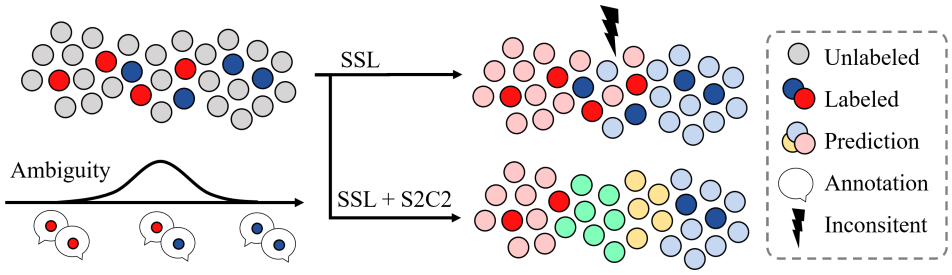


Figure 1: Benefit of Semi-Supervised Classification & Clustering (S2C2) over Semi-Supervised Learning (SSL) - Real-world datasets often suffer from intra- or interoberserver variability (IIV) during the annotation and thus no clear separation of classes is given as in common benchmark datasets. Images with a high variability between the annotations have therefore an ambiguous label. SSL can be confused by these ambiguous labels (see lightning bolt) which results in inconsistent predictions. Our method S2C2 can be used in combination with SSL to identify ambiguous images automatically and cluster them, while classifying the rest as usual. Therefore, we avoid the ambiguity of the labels during training and generated cluster proposals which can be used to create more consistent labels.

approximations of l are less consistent and lower the performance of trained models (3; 41) and additional experiments in subsection 3.6. One goal of this paper is to highlight samples that benefit from careful (re-)labeling to approximate l better while not increasing the required labels / annotations much.

For this purpose, we propose Semi-Supervised Classification & Clustering (S2C2) which simultaneously distinguishes between ambiguous and certain images, classifies the certain images and clusters visually similar ambiguous images. S2C2 is not just another SSL algorithm but can be combined with many existing deep SSL algorithms and is thus *orthogonal* to them. We will show that this approach leads to better classifications and more compact clusters across multiple semi-supervised algorithms and non-cureated datasets. Following previous literature (32), we will demonstrate the improved consistency of labels based on proposals from S2C2 and the beneficial insights into the model due to the ambiguity estimation. A graphical summary is provided in Figure 1.

The key contributions of this paper are:
(1) S2C2 allows a SSL algorithm to predict on avarage a 7.6% better F1-score for classifications and a 7.9% lower inner distance of clustering across multiple algorithms and non-curated datasets. This results in better handling of ambiguous images and thus real-world datasets.
(2) Our method S2C2 can easily extend many deep SSL algorithms and is therefore orthogonal to them. It can be implemented without a noticeable trade-off in terms of run-time or memory consumption.
(3) We give a proof-of-concept that the S2C2 proposals can be used to create faster and more consistent labels in comparison to the non-extended algorithms and a consensus process. This leads to higher quality data for further evaluation or model training.

1.1 Related Work

Our method is mainly related to Semi-Supervised Learning, Handling Noisy Data and Classification & Clustering.

Semi-Supervised Learning (7) is mainly developed on curated benchmark datasets (22; 9; 21) where the issue of IIV is not considered. In contrast to other SSL research (8; 42; 6; 12; 35), we are not evaluating on these curated benchmarks but work with new real-world datasets for two reasons. Firstly, curated datasets do not suffer so much from IIV because they were already cleaned. Recent research indicates that even these datasets suffer from erros in the labels which negatively impact the performance (28; 3). Secondly, if we want to evaluate the IIV issue, we need an approximation of the variability of the label for each image e.g. in the form of multiple annotations per image. However, this information is often not provided at current state-of-the-art benchmarks.

Handling Noisy Data is often defined as handling mistakes in the labels (36; 18; 1). Our method S2C2 automatically trains the extended SSL algorithm on only the certain images and thus filters for the difficult and possible noisy ambiguous images. However, these ambiguous images are not necessarily wrongly labeled but we still consider them uninformative for SSL. Instead, we only require them to form visually homogeneous clusters. In contrast to common noise estimation (36; 18; 1; 25), we do not just ignore or relabel these images because a ambiguous label describes a variability between multiple classes and we do not want to loose this knowledge during training.

The combination of **Classification & Clustering** was investigated in (30; 27; 5). However, the methods only use classical approaches and no deep neural network which makes it more difficult to extend the methods to real-world image data. Clustering with deep neural networks has been used successfully in image classification (38; 15; 31) and has been combined with pairwise classification constraints (34). We simultaneously calculate a classification and a clustering with a deep neural network on real-world image data.

2 METHOD

Our method Semi-Supervised Classification & Clustering (S2C2) is not a individual method but an extension for most SSL algorithms such as (2; 35; 37; 24; 23). We can extend any image classification model with S2C2 as long as it is compatible with the definition of an arbitrary SSL algorithm below.

2.1 DEFINITIONS

We assume that every image $x \in X$ has an unknown soft probability distribution $l \in [0,1]^k$ for a classification problem with k classes. This assumption is based on two main reasons. Firstly, inconsistent annotations exist due to subjective opinions from the annotators, e.g. the grading of an illness (17). A hard label $l \in \{0,1\}^k$ could not model such a difference over the complete annotator population. Secondly, if we look for example at biological processes, there exist images of intermediate transition stages between two classes, e.g. the degeneration of a living underwater organism to dead biomass (31).

An image and its corresponding label l are *ambiguous* if $i, j \in \{1, .., k\}$ exist with $i \neq j, l_i > 0$ and $l_j > 0$. Otherwise the image and its label are *certain*. The ambiguity of a label is $1 - max_{i \in \{1,..,k\}} l_i$. An image might be ambiguous because it is actually an intermediate or uncertain combination of different classes as stated above. For this reason, we view ambiguous images not just as wrongly assigned images.

A SSL algorithm uses a labeled dataset X_l and an unlabeled dataset X_u for the training of a neural network Φ with $X = X_l \dot{\bigcup} X_u$. For all images $x \in X_l$ a hard label l is available while no label information is available for $x \in X_u$. The output $p_n(x) := \Phi(x)$ is a probability distribution over the k classes.

2.2 S2C2

Our method S2C2 extends an arbitrary SSL algorithm. This SSL algorithm passes an image x through the network Φ and predicts a classification $p_n(x) \in [0,1]^k$. S2C2 calculates two additional outputs without a noticeable impact on training time or memory consumption: a clustering assignment $p_o(x) \in [0,1]^{k'}$ with $k' > k$ and an ambiguity estimation $p_a(x) \in [0,1]$. The cluster assignment partitions visually similar images in more clusters than classes exist (overclustering with $k' > k$). The ambiguity estimation is used to determine if a classification ($p_n(x)$) or an (over)clustering ($p_o(x)$) is used as the final output. If $p_a(x) < 0.5$ the image is predicted as certain and thus the classification is used. Otherwise, the image is estimated as ambiguous and the clustering is used as output.

The network is trained by minimizing the following loss function:

$$\begin{aligned} L(x) = & L_{SSL}(x) \cdot [1 - p_a(x)] + \lambda_a L_A(x) \\ & + \lambda_{CE^{-1}} L_{CE^{-1}}(x) \cdot [1 - p_a(x)] + \lambda_s L_S(x) \cdot p_a(x) \end{aligned} \tag{1}$$

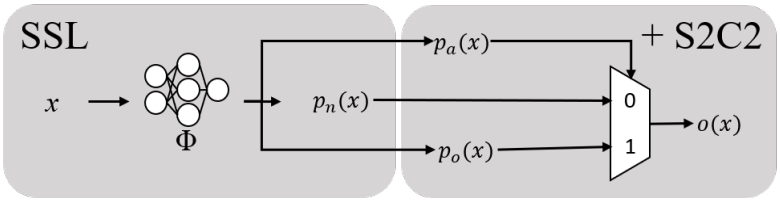


Figure 2: Our method S2C2 and an extended arbitrary SSL method – The SSL algorithm passes an image x through the network Φ and outputs a classification $p_n(x)$. We add two additional outputs: an overclustering $p_o(x)$ and a ambiguity estimation $p_a(x)$. The ambiguity estimation $p_a(x)$ is used to determine if the classification or the overclustering output is used for our method S2C2.

The first three loss terms correspond to the outputs $p_n(x), p_a(x)$ and $p_o(x)$ respectively. The last term is optional and stabalizes the training. The λ values are weights to balance the impact of each term. The first loss L_{SSL} is the loss calculated by the original SSL algorithm and is only scaled with $[1 - p_a(x)]$ to prevent the original SSL training on images the network predicts as ambiguous.

The second loss L_A improves the ambiguity estimation. As stated above, the underlying distribution l is unknown and thus we do not know during training if x is ambiguous or certain. However, we can expect to know or be given an prior probability $p_A \in [0, 1]$ of the expected percentage of ambiguous images in the total dataset. Based on this probability, we can estimate a Pseudo-Label of the ambiguity of each image in a batch during training. The loss L_F is the binary cross-entropy between the Pseudo-Label $h(x)$ and $p_a(x)$. The formulation is given below with i the index of the image x inside the given batch when all images inside the batch are sorted in ascending order based on p_a.

$$\begin{aligned} L_A(x) &= CE(h(x), p_a(x)) \\ &= -(1 - h(x)) \cdot ln(p_a(x = 0)) - h(x) \cdot ln(p_a(x = 1)) \text{ with} \\ h(x) &= \begin{cases} 1 & i \leq \text{batch size} \cdot p_A \\ 0 & \text{else} \end{cases} \end{aligned} \tag{2}$$

The third loss $L_{CE^{-1}}$ incentivises visually homogeneous clusters of the images by pushing images from different classes into different clusters and it has been shown to improve classification based on overclustering (31).

$$CE^{-1}(p_o(x), p_o(x')) = -\sum_{c=1}^{k} p_o(x)_c \cdot ln(1 - p_o(x')_c)\,. \tag{3}$$

The loss was presented in (31) and is also scaled with $[1 - p_a(x)]$ because it uses an estimate of the class for an image which could be wrong for ambiguous images. In contrast to (31), we select the image x' from the same batch as x and not as an additional input with a different class label. For the selection, we use either the ground-truth label l of x if it is available or the Pseudo-Label based on the network prediction $p_n(x)$

The forth term L_S is the cross-entropy (CE) between $p_o(x)$ and $p_o(x')$ for two differently augmented versions x, x' of the same image. This loss is scaled with $f(x)$ and incentivises that augmented versions of the same ambiguous image are in the same output cluster. We use CE because it indirectly minimizes also the entropy of $p_o(x)$ which leads to sharper predictions. Many SSL algorithms already use a differently augmented version x' of x as secondary input (2; 35; 37; 23; 15) which allows an easy computation. Otherwise, the forth term is not calculated and treated as zero.

3 Experiments

3.1 Datasets

A main contribution of this paper is that our method can be applied to many semi-supervised algorithms across different real-world ambiguous datasets without major changes. While many datasets

Table 1: Overview of the used datasets – # is an abbreviation for number. The class imbalance is given as the percentage of the smallest and largest class with regard to the complete dataset. $\hat{p}_A$ is the expected prior ambiguity probability of the dataset. n is the average of annotations per image.

Name	# classes	Input size $[px]$	# Images			Class Imbalance $[\%]$		$\hat{p}_A$ $[\%]$	n
			Train	Val	Unlabeled	Smallest	Largest		
Plankton (31)	10	96x96	1964	2456	7860	4.16	30.37	44	24
Turkey (39)	2	96x96	1299	1542	5199	9.66	90.33	22	3
Mice Bone (33)	3	224x224	277	169	278	10.81	63.98	65	3
CIFAR-10H (28)	10	32x32	1600	2000	6400	9.88	10.16	32	51

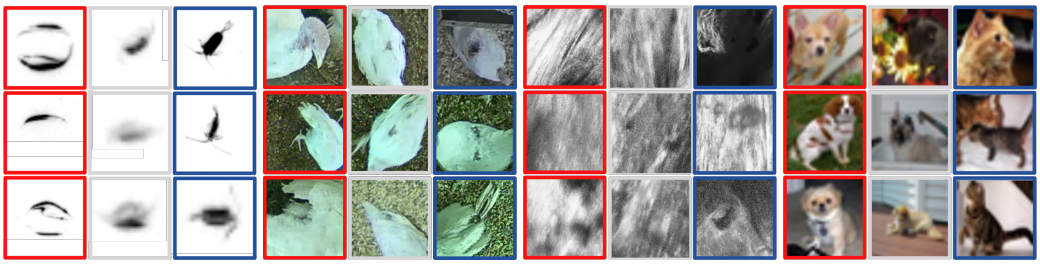
(28; 18; 31; 33; 16; 26; 4; 11; 17; 10) suffer from annotation variability, we do not know the unknown underlying distribution l to evaluate the ambiguity or any related metrics. We can approximate l with the average over multiple annotations from humans. An annotation is the hard coded guess $a = (a_1, ..., a_k) \in \{0, 1\}^k$ of the class for an image from a human with exactly one $i' \in \{1, ..., k\} : a'_i = 1$ and for all $j \in \{1, ..., k\} \setminus \{i'\} : a_j = 0$. We assume that the approximation $\hat{l}$ as the average of n annotations is identical to the unknown distribution l for $n \longrightarrow \infty$. This leaves the issue that we need multiple annotations per image for a dataset with ambiguous labels which are often not available. However, all datasets summarized in Table 1 have multiple annotations and thus allow the approximation of $\hat{l}$. Nine visual examples for all datasets are given in Figure 3 and the datasets are shortly introduced below.

The *Plankton* dataset was introduced in (31). The dataset contains 10 plankton classes and has multiple labels per image due to the help of citizen scientists. In contrast to (31), we include ambiguous images in the training and validation set and do not enforce a class balance which results in a slighlty different data split as shown in Figure 3.

The *Turkey* dataset was used in (39). The dataset contains cropped images of potential injuries which were separately annotated by three experts as not injured or injured.

The *Mice Bone* dataset is based on the raw data which was published in (33). The raw data are 3D scans from collagen fibers in mice bones. The three proposed classes are similar and dissimilar collagen fiber orientations and not relevant regions due to noise or background. We used the given segmentations to cut image regions from the original 2D image slices which mainly consist of one class. We generated ambiguous GT labels on 10% of the generated images by averaging over three own classifications from an expert.

The *CIFAR-10H* (28) dataset provides multiple annotations for the test set of CIFAR-10(21). This dataset is interesting because it illustrates that even the hard labels from benchmark datasets like CIFAR-10 are based on soft labels due to IIV.



(a) Plankton (31) (b) Turkey (39) (c) Mice Bone (33) (d) CIFAR-10H (28)

Figure 3: Example images for the ambiguous real-world datasets – All datasets have certain images (red & blue) and ambiguous images between these classes (grey). The classes are Bubble & Copepod, Not Injured & Injured, Similar & Dissimilar Orientations and Dog & Cat respectively.

For all datasets, we split our images X into a labeled X_l and unlabeled X_u training set. We keep additional images as a validation subset. On X_l, we use for each image a random hard label sampled from the corresponding $\hat{l}$. This simulates that we only have a noisy approximation of the true ground

truth label l. On X_u, we can only use the image information and not any label information. The validation data is used to compare the trained networks and to detect issues like overfitting.

As stated above the approximation of $\hat{l}$ is only possible with multiple annotations per image. For this reason the inclusion of classical benchmarks like (21; 9) is difficult because no multiple annotations or labels are given. For the CIFAR-10 dataset (21), multiple annotations are only available for the test set as the above mentioned CIFAR-10H dataset (28) and thus we can evaluate on this subset. For the STL-10 dataset (9), only one annotation / label per image is given. We still include some results of this dataset to illustrate the performance on previous benchmarks.

3.2 METRICS

We want to measure the quality of classification and clusters over the certain and ambiguous data respectively which we assume are better proposals in the annotation process or evaluation by experts. Based on this reasoning, we decided to use the weighted F1-Score on certain data and the mean inner distance on ambiguous data, The ambiguity is determined by the network output p_a. We define the metrics in detail below and give in subsection 3.6 a proof-of-concept for the higher consistency of labels based on proposals selected by the defined metrics.

During training we do not enforce a balance between ambiguous and certain predictions to keep the required prior knowledge minimal and give more flexibility to the model. This leads to two issues. Firstly, our network might predict almost all samples as certain (or ambiguous). This would make the clustering (or classification) metric not meaningful due to too few samples. Hence, we avoid this issue by calling a training *degenerated* if no more than 10% of the validation data are either predicted as ambiguous or certain. Secondly, due to the definition of the datasets, some classes might have no or very few certain (predicted) images and an averaging over their classes can lead to non-intuitive metrics. This issue can also cause instability of a metric due to a low sampling size in a class.

Hence, we use the *weighted F1-Score* on certain images. We use the weighted version based on the number of images per class to avoid instability due to the second issue mentioned above. For the ambiguous images, we use the mean inner euclidean distance (d) to the centroid on the soft / ambiguous gt labels. The equation for a set of clusters of images X is given in Equation 4 with sets $C \in X$ as clusters and the corresponding approximated soft label distribution $\hat{l}_x$ for each image $x \in C$. The centroid per cluster is given as μ_C.

$$d(X) := \frac{1}{|X|} \sum_{C \in X} \frac{1}{|C|} \sum_{x \in C} ||\hat{l}_x - \mu_C||_2 \text{ with } \mu_C := \frac{1}{|C|} \sum_{x \in C} \hat{l}_x \tag{4}$$

We use the vanilla (unchanged) SSL algorithms as baseline experiments. For these experiments and some ablation experiments, we have no ambiguity prediction $p_a(x)$. In this case, we assume all images to be certain and use $p_n(x)$ as output. We often noticed that the classification improved while the clustering degenerated and the other way round. Therefore, we determine the best performance by looking at the difference ($d-$F1) between distance and F1-Score (smaller is better). In general, we have 3 runs per setup but we exclude results that degenerate as described above. We report the best of these runs based on the ($d-$F1)-score or the mean and standard deviation over all non-degenerated runs. All scores are calculated on the validation data which is in general about 20% of all the data (see details in Table 1).

3.3 IMPLEMENTATION DETAILS

All methods use the same code base[1] and share major hyperparameters which is crucial for valid comparisons (20). We use the prior ambiguity $p_A = 0.6$ and loss weights $\lambda_{CE^{-1}} = 10$, $\lambda_f = 0.1$ and $\lambda_s = 0.1$ across all experiments. The additional losses L_A and L_S are only applied on the unlabeled data while $L_{CE^{-1}}$ is also calculated on the labeled data. These hyperparameters were determined heuristically on the Plankton Dataset with Mean-Teacher and show strong results across different methods and datasets as shown in subsection 3.4. Most likely these parameters are not optimal for an individual combination of a method and a dataset but they show the general usability across methods and datasets. We refer to the supplementary for more detailed insights about individual hyperparameters.

[1] https://github.com/google-research/fixmatch + own S2C2 code, main pseudo code in supplementary

Table 2: Performance across different methods and datasets – The vanilla algorithm is highlighted in light grey. Better results in comparison to the vanilla algorithm are marked bold. The definition of the metrics are given in subsection 3.2. CE stands for supervised Cross-Entropy training. All values are given in %. Reasons for exclusion: H - Hardware Restrictions

Methods	Plankton			Turkey			Mice Bone			CIFAR10H			STL-10
	F1 ↑	d ↓	$(d-\text{F1})$ ↓	F1 ↑	d ↓	$(d-\text{F1})$ ↓	F1 ↑	d ↓	$(d-\text{F1})$ ↓	F1 ↑	d ↓	$(d-\text{F1})$ ↓	F1 ↑
CE	86.71	30.45	-56.26	83.84	42.98	-40.86	69.55	54.75	-14.80	67.71	55.80	-11.91	80.48
CE + S2C2	78.24	**23.41**	-54.84	**85.79**	**27.64**	**-58.14**	**93.88**	**36.58**	**57.30**	**78.27**	**54.52**	**-23.75**	**88.45**
Mean-Teacher (37)	88..72	25.84	-62.88	81.82	45.12	-36.70	66.41	48.83	-17.58	73.53	46.93	-26.59	80.67
Mean-Teacher (37) + S2C2	**91.30**	**24.84**	**-66.46**	**86.45**	**33.92**	**-52.53**	**89.4**	**35.11**	**-54.73**	**85.13**	52.44	**-32.69**	**89.28**
Pi-Model (23)	87.57	28.43	-59.14	82.11	39.46	-42.65	68.15	54.11	-14.04	71.53	49.13	-22.40	82.56
Pi-Model (23) + S2C2	79.79	**19.08**	**-60.71**	**87.43**	**23.33**	**-64.10**	**88.01**	**30.99**	**-57.02**	**83.05**	**43.40**	**-39.65**	**89.54**
Pseudo-Label (24)	87.62	27.42	-60.20	82.37	44.88	-37.49	66.60	57.03	-9.57	69.70	53.30	-16.40	82.48
Pseudo-Label (24) + S2C2	**89.31**	31.76	-57.55	**83.44**	**35.04**	**-48.41**	**86.58**	**37.52**	**-49.06**	**83.74**	**51.32**	**-32.42**	**88.87**
FixMatch (35)	85.81	30.29	-55.52	82.14	43.33	-38.81	H	H	H	78.09	41.99	-36.10	89.35
FixMatch (35) + S2C2	**87.20**	31.28	**-55.92**	**83.56**	**28.17**	**-55.39**	H	H	H	**83.09**	49.49	-33.60	**91.45**

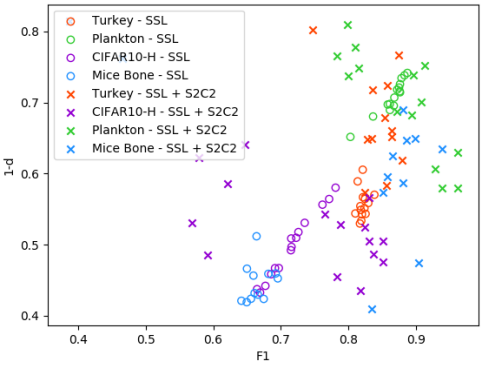


Figure 4: Comparison of SSL and S2C2 – Each datapoint represents an independent training run depending on the weighted F1-Score (F1) and the mean inner distance (d). The color and marker types define the used dataset and method respectively.

Table 3: Consistency comparison of generated labels from proposals for the Mice Bone dataset – The first column describes by which algorithm the propsals were generated. The Cohen's kappa coefficient κ measures the agreement of two repetitions and Time gives the averaged annotation time in minutes.

Methods	κ	Time
No Proposals	0.7135	13.95
SSL	0.7047	7.5
SSL + S2C2	0.8437	7.13

3.4 Results

The comparison between different semi-supervised algorithms and their extension with S2C2 is given in Table 2. The best results were selected as described in subsection 3.2 and the complete results are given in the supplementary. We see that S2C2 improves the classification and clustering performance across the majority of classes and methods by 5 to 10%. $(d-\text{F1})$ is improved by up to 40% for 16 out of 19 method-dataset-combinations. On average, we achieve a 7.6% higher F1-score for certain classifications and a 7.9% lower inner distance for clusterings of ambiguous images if we look at all non excluded method-dataset-combinations. Even on STL-10 a dataset without the possibility to evaluate ambiguous labels S2C2 creates up to 9% better classifications.

In Figure 4, we plot all used runs for the evaluation based on their F1 and d-score. Overall, we see the most benefit on the Mice Bone and Turkey dataset which we attribute to the worse initial approximation of $\hat{l}$. In most cases, we can see an improvement in F1 and d metric but also a greater variability in the training results. We think this variability is introduced because the network has to decide automatically what a certain and ambiguous image is without any direct guidance. The different vanilla algorithms achieve quite similar results for each dataset. Only FixMatch achieves a more than 5% better F1-Score on the curated STL-10 and CIFAR-10H dataset. In general, we see that S2C2 can be beneficially applied to a variety of datasets and methods and predicts better classifications and more compact clusters. In subsection 3.6, we will show that this defines also better proposals during the annotation process.

3.5 Ablation

We pooled the runs between all methods to evaluate the impact of the individual components of our method S2C2. We report the best and the average across all methods in Table 4. We averaged over

Table 4: Complete ablation results for averaging over different methods – The vanilla algorithms as baseline are highlighted in light grey. Each row below that extend this baseline individually with CE^{-1} (31), Clustering & Classification (CC) or both (S2C2). CC can be interpreted as S2C2 without CE^{-1}. The prior ambiguity estimate p_A is given if used in brackets. Results that improve over the baseline are marked in bold. The definition of the metrics are given in subsection 3.2. The column 'Ambiguous' gives the percentage of predicted ambiguous data and the last column gives the number of runs over which we averaged.

	F1		d		(d−F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR10-H									
Baseline	0.7809	0.7153 ± 0.0359	0.4199	0.5027 ± 0.0469	-0.3611	-0.2126 ± 0.0827	-	-	15
+ CE^{-1}	0.7383	**0.7191 ± 0.0164**	0.4692	**0.4929 ± 0.0243**	-0.2691	**-0.2262 ± 0.0404**	-	-	12
+ CC ($p_A = 0.6$)	**0.8565**	**0.7471 ± 0.1246**	0.8657	0.8768 ± 0.0129	0.0092	0.1297 ± 0.1374	0.6145	0.5923 ± 0.0322	12
+ S2C2 ($p_A = 0.32$)	0.6656	0.6970 ± 0.0469	**0.2155**	**0.3684 ± 0.1227**	**-0.4501**	**-0.3286 ± 0.0836**	0.2910	0.3115 ± 0.0140	12
+ S2C2 ($p_A = 0.6$)	**0.8305**	**0.7457 ± 0.1097**	0.4340	**0.4741 ± 0.0584**	**-0.3965**	**-0.2716 ± 0.0928**	0.6125	0.5860 ± 0.0290	15
Plankton									
Baseline	0.8872	0.8652 ± 0.0212	0.2584	0.2915 ± 0.0240	-0.6287	-0.5737 ± 0.0444	-	-	15
+ CE^{-1}	**0.8896**	**0.8803 ± 0.0060**	**0.2540**	**0.2690 ± 0.0098**	**-0.6356**	**-0.6113 ± 0.0154**	-	-	12
+ CC ($p_A = 0.6$)	**0.8919**	**0.9128 ± 0.0427**	0.4085	0.7702 ± 0.1630	-0.4833	-0.1426 ± 0.1375	0.6242	0.5927 ± 0.0127	12
+ S2C2 ($p_A = 0.44$)	0.8625	**0.9049 ± 0.0340**	**0.2192**	0.3269 ± 0.0526	**-0.6433**	**-0.5780 ± 0.0305**	0.4365	0.4451 ± 0.0204	11
+ S2C2 ($p_A = 0.6$)	**0.9130**	**0.8768 ± 0.0640**	**0.2484**	0.3004 ± 0.0750	**-0.6646**	**-0.5764 ± 0.0416**	0.6164	0.5893 ± 0.0202	14
Turkey									
Baseline	0.8211	0.8213 ± 0.00s69	0.3946	0.4428 ± 0.0209	-0.4265	-0.3786 ± 0.0230	-	-	15
+ CE^{-1}	0.7998	0.7998 ± 0.0000	**0.3338**	**0.3338 ± 0.0000**	**-0.4660**	**-0.4660 ± 0.0000**	-	-	12
+ CC ($p_A = 0.6$)	**0.8527**	**0.8264 ± 0.0469**	**0.3400**	**0.3435 ± 0.0408**	**-0.5127**	**-0.4829 ± 0.0128**	0.5837	0.5646 ± 0.0427	12
+ S2C2 ($p_A = 0.22$)	0.7998	0.7998 ± 0.0000	**0.1675**	**0.2252 ± 0.0646**	**-0.6322**	**-0.5746 ± 0.0646**	0.5000	0.3674 ± 0.2054	4
+ S2C2 ($p_A = 0.6$)	**0.8743**	**0.8432 ± 0.0350**	**0.2333**	**0.3270 ± 0.0692**	**-0.6410**	**-0.5162 ± 0.0643**	0.8093	0.6387 ± 0.2354	12

15 runs except for the additional ablations with 12 runs where we excluded FixMatch due to the 12 times higher required GPU hours. We excluded degenerated runs which lowers the number for some ablations further. The complete results are given in the supplementary. Across the datasets, we see the best (d−F1)-scores are achieved by S2C2. The impact of the components varies between the datasets. We see that CE^{-1} positively impacts the clustering results which confirms the benefit of using CE^{-1} for overclustering (31). CC reaches in all cases a better F1-Score than the baseline and even surpasses S2C2 sometimes. However, the inner distance (d) may increase as well. We conclude that CC and CE^{-1} on their own can lead to improvements but only the combination of both parts results in a stable algorithm across datasets and methods. If we use the correct amount of ambiguity $\hat{p}_A$ in each dataset as p_A, we see that in general the F1-Score decreases and d-score improves. We attribute this difference to the lower prior ambiguity p_A because S2C2 tries to predict more certain than ambiguous images. This leads to a lower inner distance but also includes more difficult images in the classification of the certain data. We believe this parameter is essential for balancing the improvements in the F1- and d-score for a specified usecase.

3.6 INSIGHTS

Consistency - We showed that S2C2 can lead to better classifications and clusterings than SSL alone. Our assumption from above was that using the predictions of S2C2 as proposals during the annotation process leads to more consistent and thus higher quality labels. We give a proof-of-concept on the Mice Bone dataset and the SSL algorithm MeanTeacher (37) in Figure 3. The data was reannotated by an expert two times for each given proposal. We see that the consistency is similar between using no proposals and the SSL classification predicitions. While the SSL proposals can decrease the annotation time by about 50%, too many ill predicted images exist which need to be corrected. The usage of the better classification and visually homogeneous clusters of S2C2 lead to an increased κ coefficient of over 10% while achieving an even faster annotation time than SSL.

Impact of ambiguous labels - We stated that high quality labels lead to better model training (3). We verify this statement on the Plankton and CIFAR-10H dataets in Table 5. We see for all supervised and semi-supervised methods that used trainings labels based on the complete distribution of $\hat{l}$ (argmax $\hat{l}$, column A) leads to an improvement of up to 10% in comparison to sampling the training label from $\hat{l}$ (column A1). Nevertheless, in our work we used the approximation based on a sample from $\hat{l}$ because in a real-world task we would have access to $\hat{l}$ only at a high cost. If we remove

Table 5: Impact of ambiguous labels – Macro F1-Score for different methods and across three different subsets on the validation data from the Plankton and CIFAR10-H dataset. Columns: A1 – Labels are sampled from $\hat{l}$; A – Labels are the maximum class of $\hat{l}$; C – No ambiguous labels/images are used

Methods	Plankton			CIFAR10-H		
	A1	A	C	A1	A	C
CE	86.71	88.35	96.10	67.71	68.89	86.57
Mean-Teacher (37)	88.72	88.94	96.00	73.56	75.06	86.96
Pi-Model (23)	87.57	89.03	96.41	71.53	72.75	87.19
Pseudo-Label (24)	87.62	88.41	96.20	69.70	71.82	87.15
FixMatch (35)	80.29	90.24	98.86	76.15	79.15	90.37

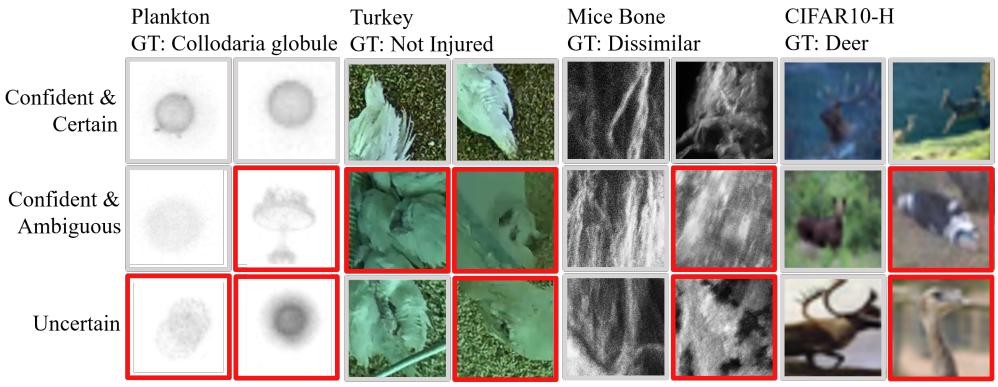


Figure 5: Qualitive Results for selected classes across different confidences and ambiguity predictions – Wrong classifications based on the normal head are highlighted in red.

the ambiguous images entirely from the dataset (column C), the results improve again by 8 to 15%. This indicates that ambiguous images are a major issue during the training process.

Interpretability - Many SSL algorithms interpret the probability of the largest value of $p_n(x)$ as confidence (13). We qualitatively illustrate in Figure 5 that using our ambiguity prediction $p_a(x)$ can lead to better interpretability and fewer errors. We show 6 randomly picked examples for selected classes across the datasets and extended results in the supplementary. The images in each row have a similar value for $p_n(x)$ and $p_a(x)$. The first row presents highly confident predictions on certain predicted images and shows no errors in the given random picks. The middle row shows highly confident predictions on ambiguous predicted images. Some of these images are false and would lower the performance without the additional ambiguity prediction. The last row shows non-confident or uncertain ($0.4 < p_a(x) < 0.6$) predictions which are often wrong.

4 Conclusion

In real-world datasets, we often encounter ambiguous labels for example due to intra- or interobserver variability. We propose our method S2C2 which is our orthogonal extension to many semi-supervised algorithms and allows to classify images with certain labels and cluster ambiguous ones. S2C2 also automatically determines which image to treat as certain or ambiguous only based on a given prior distribution p_A. On average, we achieve an increased F1-Score of 7.6% and a lower inner distance in clusterings of 7.9% over all method-dataset-combinations. We give a proof-of-concept that ambiguous labels can negatively impact the classification performance, annotations based on proposals from S2C2 are more consistent and the ambiguity prediction can give more insight in the model reasoning. Therefore, SSL algorithms with S2C2 are better suited to handle real-world datasets including ambiguous labeled images either by an improved classification / clustering or as a proposal during the annotation process with more insight.

REFERENCES

[1] Algan, G., Ulusoy, I.: Image Classification with Deep Learning in the Presence of Noisy Labels: A Survey. Knowledge-Based Systems (2020)

[2] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 5050–5060 (2019)

[3] Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., van den Oord, A.: Are we done with imagenet? (2020)

[4] Brünger, J., Dippel, S., Koch, R., Veit, C.: 'Tailception': using neural networks for assessing tail lesions on pictures of pig carcasses. Animal **13**(5), 1030–1036 (2019)

[5] Cai, W., Chen, S., Zhang, D.: A simultaneous learning framework for clustering and classification. Pattern Recognition **42**(7), 1248–1259 (2009)

[6] Caron, M., Goyal, P., Misra, I., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2020)

[7] Chapelle, O., Scholkopf, B., Zien, A., Schölkopf, B., Zien, A.: Semi-supervised learning. IEEE Transactions on Neural Networks **20**(3), 542 (2006)

[8] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big Self-Supervised Models are Strong Semi-Supervised Learners. Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020) (2020)

[9] Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223 (2011)

[10] Culverhouse, P., Williams, R., Reguera, B., Herry, V., González-Gil, S.: Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. Marine Ecology Progress Series **247**, 17–25 (2003)

[11] De Fauw, J., Ledsam, J.R., Romera-Paredes, B., Nikolov, S., Tomasev, N., Blackwell, S., Askham, H., Glorot, X., O'Donoghue, B., Visentin, D., Others: Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature medicine **24**(9), 1342–1350 (2018)

[12] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised Learning. Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020) (2020)

[13] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning. pp. 1321–1330. PMLR (2017)

[14] Jenckel, M., Parkala, S.S., Bukhari, S.S., Dengel, A.: Impact of Training LSTM-RNN with Fuzzy Ground Truth. In: ICPRAM (2018)

[15] Ji, X., Henriques, J.F., Vedaldi, A., Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9865–9874. No. Iic (2019)

[16] Jungo, A., Meier, R., Ermis, E., Blatti-Moreno, M., Herrmann, E., Wiest, R., Reyes, M.: On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation. In: Medical Image Computing and Computer Assisted Interventions, MICCAI. pp. 682–690. Springer (2018)

[17] Karimi, D., Nir, G., Fazli, L., Black, P.C., Goldenberg, L., Salcudean, S.E.: Deep Learning-Based Gleason Grading of Prostate Cancer From Histopathology Images—Role of Multiscale Decision Aggregation and Data Augmentation. IEEE Journal of Biomedical and Health Informatics **24**(5), 1413–1426 (2020)

[18] Karimi, D., Dou, H., Warfield, S.K., Gholipour, A.: Deep learning with noisy labels: exploring techniques and remedies in medical image analysis. Medical Image Analysis **65** (2020)

[19] Kim, B., Choo, J., Kwon, Y.D., Joe, S., Min, S., Gwon, Y.: SelfMatch: Combining Contrastive Self-Supervision and Consistency for Semi-Supervised Learning (NeurIPS) (2021)

[20] Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1920–1929 (2019)

[21] Krizhevsky, A., Hinton, G., Others: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)

[22] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. vol. 60, pp. 1097–1105. Association for Computing Machinery (2012)

[23] Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (2017)

[24] Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 2 (2013)

[25] Li, J., Socher, R., Hoi, S.C.H.: DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In: International Conference on Learning Representations. pp. 1–14 (2020)

[26] Ooms, E., Zonderland, H., Eijkemans, M., Kriege, M., Mahdavian Delavary, B., Burger, C., Ansink, A.: Mammography: Interobserver variability in breast density assessment. The Breast **16**(6), 568–576 (dec 2007)

[27] Peikari, M., Salama, S., Nofech-mozes, S., Martel, A.L.: A Cluster-then-label Semi- supervised Learning Approach for Pathology Image Classification. Scientific Reports (April), 1–13 (2018)

[28] Peterson, J., Battleday, R., Griffiths, T., Russakovsky, O.: Human uncertainty makes classification more robust. Proceedings of the IEEE International Conference on Computer Vision **2019-Octob**, 9616–9625 (2019)

[29] Pham, H., Dai, Z., Xie, Q., Luong, M.T., Le, Q.V.: Meta Pseudo Labels (2020)

[30] Qian, Q., Chen, S., Cai, W.: Simultaneous clustering and classification over cluster structure representation. Pattern Recognition **45**(6), 2227–2236 (2012)

[31] Schmarje, L., Brünger, J., Santarossa, M., Schröder, S.M., Kiko, R., Koch, R.: Fuzzy Overclustering: Semi-supervised classification of fuzzy labels with overclustering and inverse cross-entropy. Sensors (2021)

[32] Schmarje, L., Koch, R.: Life is not black and white - Combining Semi-Supervised Learning with fuzzy labels. Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", (2021)

[33] Schmarje, L., Zelenka, C., Geisen, U., Glüer, C.C., Koch, R.: 2D and 3D Segmentation of Uncertain Local Collagen Fiber Orientations in SHG Microscopy. In: DAGM German Conference of Pattern Regocnition, vol. 11824 LNCS, pp. 374–386. Springer (2019)

[34] Śmieja, M., Struski, Ł., Figueiredo, M.A.T.: A Classification-Based Approach to Semi-Supervised Clustering with Pairwise Constraints (2020)

[35] Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020) (2020)

[36] Song, H., Kim, M., Park, D., Lee, J.: Learning from Noisy Labels with Deep Neural Networks: A Survey. arXiv preprint arXiv:1406.2080 (2020)

[37] Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: ICLR (2017)

[38] Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., Van Gool, L.: Scan: Learning to classify images without labels. In: Proceedings of the European Conference on Computer Vision. pp. 268–285 (2020)

[39] Volkmann, N., Brünger, J., Stracke, J., Zelenka, C., Koch, R., Kemper, N., Spindler, B.: SO MUCH TROUBLE IN THE HERD: DETECTION OF FIRST SIGNS OF CANNIBALISM IN TURKEYS. In: Recent advances in animal welfare science VII Virtual UFAW Animal Welfare Conference. p. 82 (2020)

[40] Xie, Q., Luong, M.T., Hovy, E., Le, Q.V., Luong, M.T., Le, Q.V., Hovy, E., Le, Q.V.: Self-Training With Noisy Student Improves ImageNet Classification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695. IEEE (jun 2020)

[41] Yun, S., Oh, S.J., Heo, B., Han, D., Choe, J., Chun, S.: Re-labeling imagenet: From single to multi-labels, from global to localized labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2340–2350 (June 2021)

[42] Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow Twins: Self-Supervised Learning via Redundancy Reduction (2021)