A data-centric approach for improving ambiguous labels with combined semi-supervised classification and clustering

Supplementary Material

A Further insights into hyperparameters

During method development, we looked at a larger variety of hyperparameters for Mean-Teacher [53] and the Plankton dataset [43]. We found in general that the model was quite robust to changes of individual parameters. The most impact we noticed from design decision where and where not a gradient is propagated. For example, if we would propagate the error along the pseudo labels for the ambiguity loss calculation the system degenerates almost always. Aside from these design decisions, the weight for $\lambda_{CE^{-1}}$ had the most impact. While we use the same value for the labeled and the unlabeled data, we see slight evidence that a lower value could be beneficial on the labeled data. Moreover, slightly lower or higher weights for the other hyperparameters showed promising results under certain circumstances. As stated in the paper, we aimed at providing general parameters across different datasets and methods and thus did not fine-tune the parameters to a specific combination.

B Pseudocode

In our pseudo code block 1, we give the main parts of our proposed method as pseudocode. The code is similar to python and Tensorflow code. In the following, we will describe the used parameters and methods. `tf` is an abbrevation for tensorflow and refers to that function. `prob_ambiguous` is the output $p_a(x)$ for a complete batch. `logits_x_over`, `logits_u_over` and `logits_u2_over` are the overclustering outputs $p_o(x)$ for a complete batch for the labeled data, the unlabeled data and the possible additional second unlabeled input respectively. The labels for the labeled data are given in `l`. `logits_u` is the output $p_n(x)$ for a complete batch of unlabeled data. The parameters `prior_ambiguity`, `wou`,`wa` and `ws` correspond to the hyperparameters $p_A, \lambda_{CE^{-1}}, \lambda_a$ and λ_s in the paper respectively. The parameter `wol` is the weight $\lambda_{CE^{-1}}$ on the labeled data. The parameter `loss_tensor` is SSL loss as tensor (L_{SSL}). The function `threshold()` thresholds the elements of the given vector (first argument) based on the given threshold (second argument). The functions `inverse_ce()`, `ce()` and `be()` calculate the loss value for inverse cross-entropy (CE^{-1}), cross-entropy and binary cross-entropy respectively. The function `get_different_logits()` selects from the given batch logits (first argument) one logit for each image in the batch. A logit is randomly selected of all logits in the batch which to do not share

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 CIFAR-10H 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			CIFAR-10H		
	A1	A	C	A1	A	C
CE	86.71	88.35	96.10	67.71	68.89	86.57
Mean-Teacher [53]	88.72	88.94	96.00	73.56	75.06	86.96
Pi-Model [31]	87.57	89.03	96.41	71.53	72.75	87.19
Pseudo-Label [32]	87.62	88.41	96.20	69.70	71.82	87.15
FixMatch [49]	80.29	90.24	98.86	76.15	79.15	90.37

the same label (`l`) or the same pseudo label based on `logits_u`. The function `get_pseudo_ambiguity_labels()` gets pseudo labels for every ambiguity prediction $p_a(x)$ based on the given prior ambiguity estimate p_A as described in the main paper.

The different outputs can easily be realized in a model by extending the dense output layer to the sum of the number of classes and the number of output clusters. Before calculating the loss or applying softmax activation the output can be separated in the desired input to our method.

C Additional Results

Impact of ambiguous labels We stated that high quality labels lead to better model training [4] and verify this statement on the Plankton and CIFAR-10H datasets in Table 5. We see for all supervised and semi-supervised methods that used training 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). We used the approximation based on a sample from $\hat{l}$ because we normally would have access to $\hat{l}$ only at a high cost. If we remove 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 [20]. We qualitatively illustrate in Figure 5 and Figure 6 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

```
def calculate_loss(prob_ambiguous, l,
    logits_x_over, logits_u, logits_u_over, logits_u2_over,
    prior_ambiguity, wou, wol, wa, ws,
    loss_tensor):

    # stop gradient on ambiguity scale
    ambiguous_scale = tf.stop_gradient(prob_ambiguous)
    certain_scale = 1 - ambiguous_scale


    pseudo_labels = tf.stop_gradient(tf.nn.softmax(logits_u))
    args_pseudo = tf.argmax(pseudo_labels, axis=1)
    pseudo_mask = threshold(pseudo_labels, 0.95)

    #  get different image based on label or pseudo-label
    # for elements in the batch from the batch
    logits_x_over_inverse = get_different_logits(logits_x_over, l)
    logits_u_over_inverse = get_different_logits(logits_u_over, l)

    loss_xeil = inverse_ce(tf.nn.softmax(logits_x_over),
        tf.nn.softmax(logits_x_over_inverse))
    loss_xeil = tf.reduce_mean(loss_xeil)

    loss_xeiu = inverse_ce(tf.nn.softmax(logits_u_over),
        tf.nn.softmax(logits_u_over_inverse))
    loss_xeiu = tf.reduce_mean(loss_xeiu * pseudo_mask * certain_scale)

     # use pseudo labels based on the number of ambiguous elements
     # in each batch to calculate ambiguity loss
    pseudo_ambiguity_label =
        get_pseudo_ambiguity_labels(prob_ambiguous, prior_ambiguity)
    loss_ambiguity = be(tf.stop_gradient(pseudo_ambiguity_label),
        prob_ambiguous)
    loss_ambiguity = tf.reduce_mean(loss_ambiguity)

    # calculate similarity loss
    # use ce to add entropy based on the logits u over
    sim_loss = ce(tf.nn.softmax(logits_u_over),
        tf.nn.softmax(logits_u2_over))
    sim_loss = tf.reduce_mean(sim_loss * ambiguous_scale)

    loss = loss_tensor * certain_scale + wou * loss_xeiu + wol * loss_xeil
        + wa * loss_ambiguity + ws * loss_ambiguous_similarity

    return loss
```

Pseudocode 1: Main pseudocode for the proposed method. The expected parameters and functions are explained in the corresponding subsection.

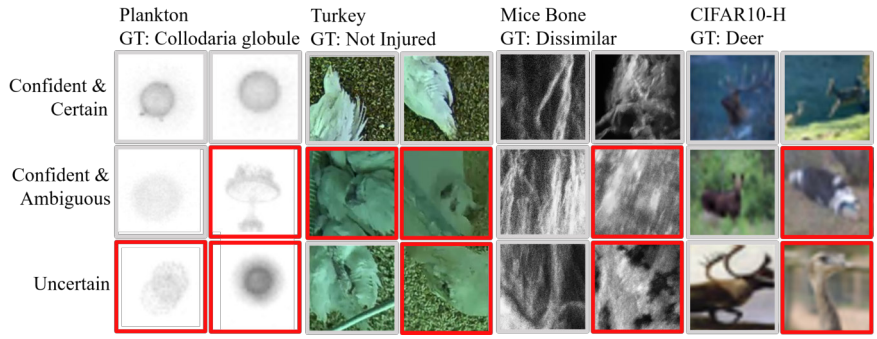


Fig. 5: Qualitative Results for selected classes across different confidences and ambiguity predictions

Table 6: Impact of unlabeled data ratio – The first column unlabeled data ratio. Better results compared to baseline are bold.

		F1		d		d−F1	
		best	mean ± std	best	mean ± std	best	mean ± std
10%	Mean-Teacher	0.5850	0.5627 +- 0.0181	0.4639	0.4889 +- 0.0275	-0.1095	-0.0738 +- 0.0304
	+ DC3	**0.6044**	0.5277 +- 0.0655	**0.2590**	**0.3011 +- 0.0319**	**-0.3455**	**-0.2266 +- 0.0879**
20%	Mean-Teacher	0.5643	0.5611 +- 0.0044	0.4721	0.5157 +- 0.0326	-0.0827	-0.0454 +- 0.0284
	+ DC3	**0.6111**	0.5380 +- 0.0519	**0.2175**	**0.3772 +- 0.1229**	**-0.3936**	**-0.1608 +- 0.1731**
50%	Mean-Teacher	0.7348	0.6700 +- 0.0856	0.4388	0.4603 +- 0.0270	-0.2910	-0.2097 +- 0.1124
	+ DC3	0.6862	**0.6839 +- 0.0023**	**0.3298**	0.4601 +- 0.1303	**-0.3518**	**-0.2238 +- 0.1279**
100%	Mean-Teacher	0.7254	0.6524 +- 0.0645	0.3875	0.4373 +- 0.0355	-0.3379	-0.2150 +- 0.0971
	+ DC3	0.5692	0.6017 +- 0.0325	**0.1147**	**0.2995 +- 0.1848**	**-0.4545**	**-0.3022 +- 0.1523**

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.

Impact of unlabeled data ratio In this study, we fixed the supervision to a fixed ratio to simplify the analysis. However, we conducted some preliminary studies on a variant of the MiceBone dataset and show the results in Table 6. Be aware that the numbers are not directly comparable due to a different data split algorithm. We see that DC3 improves the results over all supervision percentages. It seems also be more robust to less data than SSL alone, but additional data is required to confirm this trend.

Social Impact Discussion Improving the annotation process by increasing the data quality and reducing the required time leads indirectly to improved performance in broad range of deep learning applications. However, these improvements are achieved by leveraging the confirmation bias for given proposals. We propose that network predictions are validated by a human but introducing a bias without considering the consequences could lead to undesirable behaviour

in concrete cases. For example, if people are classified / graded based on provided proposals a negative bias could be introduced for certain people. In such a case, the user should consider investigating more annotation resources into an unbiased consensus process. We believe a small bias can be accepted in most applications because it is human controlled and systematically.

D Extended results

In this subsection, we give the complete result tables used for the tables in the main paper. The definitions of the metrics are given in the main paper. Detailed scatter plots are given in Figure 7. The individual tables are Table 7, Table 8, Table 9, Table 10 and Table 11.

Table 7: Complete ablation results for Cross-Entropy – The vanilla algorithm is highlighted in light grey. The row below that extend the algorithm with CE$^-$1 [43], Clustering & Classification (CC) or both (DC3). Better results in comparison to the vanilla algorithm are marked bold

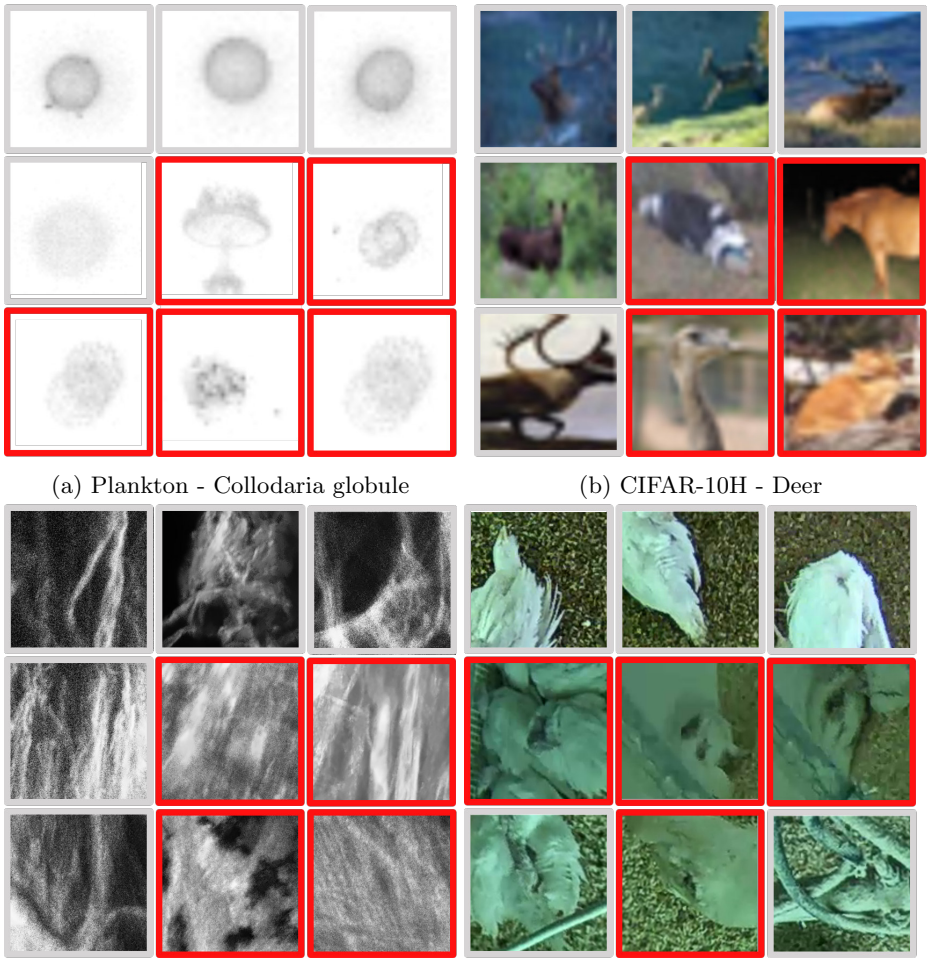
	F1		d		(d−F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR-10H									
Baseline	0.6771	0.6704 ± 0.0062	0.5580	0.5627 ± 0.0047	-0.1191	-0.1077 ± 0.0099	-	-	3
+ CE$^-$1	**0.7383**	**0.7329 ± 0.0078**	**0.4692**	**0.4712 ± 0.0044**	**-0.2691**	**-0.2618 ± 0.0120**	-	-	3
+ CC ($p_A = 0.6$)	**0.8570**	**0.8518 ± 0.0049**	0.8666	0.8662 ± 0.0005	0.0096	0.0144 ± 0.0049	0.6240	0.6197 ± 0.0045	3
+ DC3 ($p_A = 0.32$)	0.6656	0.6656 ± 0.0055	**0.2155**	**0.2498 ± 0.0399**	**-0.4501**	**-0.4158 ± 0.0364**	0.2910	0.2960 ± 0.0044	3
+ DC3 ($p_A = 0.6$)	**0.7827**	0.6474 ± 0.1178	**0.5452**	**0.5096 ± 0.0382**	**-0.2375**	**-0.1378 ± 0.0870**	0.6240	0.5775 ± 0.0404	3
Plantkon									
Baseline	0.8671	0.8632 ± 0.0034	0.3045	0.3057 ± 0.0044	-0.5626	-0.5574 ± 0.0062	-	-	3
+ CE$^-$1	**0.8896**	**0.8880 ± 0.0023**	**0.2540**	**0.2602 ± 0.0087**	**-0.6356**	**-0.6278 ± 0.0110**	-	-	2
+ CC ($p_A = 0.6$)	**0.9596**	**0.9221 ± 0.0337**	0.8321	0.8419 ± 0.0085	-0.1274	-0.0802 ± 0.0422	0.5908	0.5949 ± 0.0036	3
+ DC3 ($p_A = 0.44$)	**0.8625**	**0.9148 ± 0.0461**	**0.2192**	**0.3090 ± 0.0810**	**-0.6433**	**-0.6058 ± 0.0354**	0.4365	0.4511 ± 0.0127	3
+ DC3 ($p_A = 0.6$)	0.7824	**0.8942 ± 0.0974**	**0.2341**	**0.3580 ± 0.1074**	-0.5484	-0.5361 ± 0.0155	0.5615	0.5873 ± 0.0241	3
Turkey									
Baseline	0.8384	0.8307 ± 0.0071	0.4298	0.4357 ± 0.0058	-0.4086	-0.3951 ± 0.0117	-	-	3
+ CE$^-$1	0.7998	0.7998 ± 0.0000	**0.3338**	**0.3338 ± 0.0000**	**-0.4660**	**-0.4660 ± 0.0000**	-	-	3
+ CC ($p_A = 0.6$)	**0.8452**	0.8033 ± 0.0550	**0.3565**	**0.3213 ± 0.0542**	**-0.4887**	**-0.4820 ± 0.0068**	0.5156	0.5409 ± 0.0295	3
+ DC3 ($p_A = 0.22$)	0.7998	0.7998 ± nan	**0.2705**	**0.2705 ± nan**	**-0.5293**	**-0.5293 ± nan**	0.1087	0.1087 ± nan	1
+ DC3 ($p_A = 0.6$)	**0.8579**	0.8195 ± 0.0624	**0.2764**	**0.2653 ± 0.0627**	**-0.5814**	**-0.5543 ± 0.0252**	0.5694	0.5841 ± 0.0776	3
Mice Bone									
Baseline	0.6955	0.6753 ± 0.0198	0.5475	0.5667 ± 0.0166	-0.1479	-0.1086 ± 0.0353	-	-	3
+ DC3 ($p_A = 0.6$)	**0.9388**	**0.7373 ± 0.3046**	**0.3658**	**0.3018 ± 0.1005**	**-0.5730**	**-0.4355 ± 0.2041**	0.5680	0.5365 ± 0.0448	3
STL-10									
Baseline	0.8048	0.7918 ± 0.0125	-	-	-0.8048	-0.7918 ± 0.0125	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8845**	**0.8671 ± 0.0166**	-	-	**-0.8845**	**-0.8671** ± 0.0166	0.5919	0.4727 ± 0.2643	3

Table 8: Complete ablation results for Mean-Teacher [53] – The vanilla algorithm is highlighted in light grey. The row below that extend the algorithm with CE$^-$1 [43], Clustering & Classification (CC) or both (DC3). Better results in comparison to the vanilla algorithm are marked bold

	F1		d		(d–F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR-10H									
Baseline	0.7353	0.7280 ± 0.0065	0.4693	0.4807 ± 0.0106	-0.2659	-0.2473 ± 0.0171	-	-	3
+ CE$^-$1	**0.7360**	**0.7297 ± 0.0054**	**0.4747**	**0.4753 ± 0.0011**	-0.2613	**-0.2544 ± 0.0061**	-	-	3
+ CC ($p_A = 0.6$)	**0.8565**	**0.7791 ± 0.1243**	0.8657	0.8747 ± 0.0153	0.0092	0.0956 ± 0.1396	0.6145	0.5962 ± 0.0375	3
+ DC3 ($p_A = 0.32$)	0.6614	0.7243 ± 0.0554	**0.3197**	**0.4615 ± 0.1272**	**-0.3417**	**-0.2628 ± 0.0805**	0.2910	0.3070 ± 0.0151	3
+ DC3 ($p_A = 0.6$)	**0.8513**	0.7066 ± 0.1260	0.5244	**0.4328 ± 0.0839**	**-0.3269**	**-0.2738 ± 0.0610**	0.6145	0.5757 ± 0.0336	3
Plankton									
Baseline	0.8872	0.8828 ± 0.0044	0.2584	0.2620 ± 0.0037	-0.6287	-0.6208 ± 0.0080	-	-	3
+ CE$^-$1	0.8846	0.8821 ± 0.0035	**0.2568**	0.2623 ± 0.0050	-0.6278	-0.6198 ± 0.0082	-	-	3
+ CC ($p_A = 0.6$)	**0.9645**	**0.9345 ± 0.0260**	0.8303	0.8377 ± 0.0065	-0.1342	-0.0968 ± 0.0324	0.5928	0.5898 ± 0.0029	3
+ DC3 ($p_A = 0.44$)	0.8690	0.8634 ± 0.0080	0.2753	0.2966 ± 0.0301	-0.5937	-0.5667 ± 0.0381	0.4064	0.4098 ± 0.0049	2
+ DC3 ($p_A = 0.6$)	**0.9130**	**0.9056 ± 0.0087**	**0.2484**	0.2699 ± 0.0267	**-0.6646**	**-0.6357 ± 0.0282**	0.6164	0.6124 ± 0.0143	3
Turkey									
Baseline	0.8182	0.8158 ± 0.0049	0.4512	0.4579 ± 0.0078	-0.3670	-0.3579 ± 0.0079	-	-	3
+ CE$^-$1	0.7998	0.7998 ± 0.0000	**0.3338**	**0.3338 ± 0.0000**	**-0.4660**	**-0.4660 ± 0.0000**	-	-	3
+ CC ($p_A = 0.6$)	**0.8527**	**0.8829 ± 0.0295**	**0.3400**	**0.3816 ± 0.0384**	**-0.5127**	**-0.5013 ± 0.0098**	0.5837	0.5428 ± 0.0378	3
+ DC3 ($p_A = 0.22$)	0.7998	0.7998 ± nan	**0.1719**	**0.1719 ± nan**	**-0.6278**	**-0.6278 ± nan**	0.5252	0.4748 ± nan	1
+ DC3 ($p_A = 0.6$)	**0.8645**	**0.8639 ± 0.0008**	**0.3392**	**0.3439 ± 0.0067**	**-0.5253**	**-0.5200 ± 0.0075**	0.7691	0.7970 ± 0.0394	2
Mice Bone									
Baseline	0.6641	0.6688 ± 0.0217	0.4883	0.5209 ± 0.0284	-0.1758	-0.1479 ± 0.0301	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8984**	**0.8940 ± 0.0124**	**0.3511**	**0.4300 ± 0.0887**	**-0.5473**	**-0.4641 ± 0.0848**	0.5266	0.5444 ± 0.0178	3
STL-10									
Baseline	0.8067	0.7863 ± 0.0173	-	-	-0.8067	-0.7863 ± 0.0173	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8928**	**0.8751 ± 0.0188**	-	-	**-0.8928**	**-0.8751 ± 0.0188**	0.5897	0.4732 ± 0.2646	3

Table 9: Complete ablation results for Pi-Model [31] – The vanilla algorithm is highlighted in light grey. The row below that extend the algorithm with CE$^-$1 [43], Clustering & Classification (CC) or both (DC3). Better results in comparison to the vanilla algorithm are marked bold

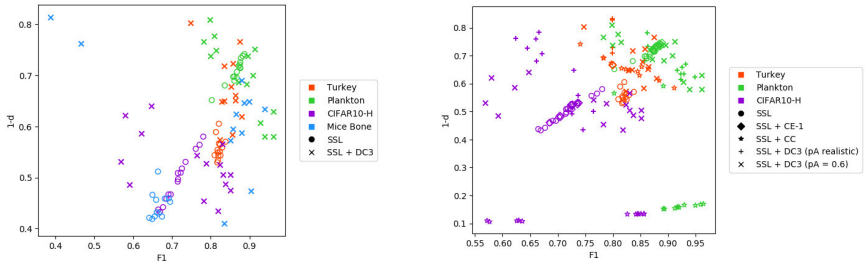
	F1		d		(d–F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR-10H									
Baseline	0.7153	0.7153 ± 0.0005	0.4913	0.5008 ± 0.0085	-0.2240	-0.2145 ± 0.0087	-	-	3
+ CE$^-$1	**0.7255**	**0.7163 ± 0.0109**	0.4917	**0.4988 ± 0.0138**	**-0.2337**	**-0.2175 ± 0.0244**	-	-	3
+ CC ($p_A = 0.6$)	**0.8420**	0.6991 ± 0.1238	0.8670	0.8823 ± 0.0133	0.0250	0.1832 ± 0.1371	0.6120	0.5728 ± 0.0349	3
+ DC3 ($p_A = 0.32$)	**0.7291**	**0.7038 ± 0.0506**	**0.3528**	**0.3564 ± 0.0849**	**-0.3764**	**-0.3474 ± 0.0466**	0.3230	0.3193 ± 0.0095	3
+ DC3 ($p_A = 0.6$)	**0.8305**	**0.8352 ± 0.0143**	**0.4340**	**0.4680 ± 0.0311**	**-0.3965**	**-0.3672 ± 0.0258**	0.6125	0.6122 ± 0.0035	3
Plankton									
Baseline	0.8757	0.8747 ± 0.0024	0.2843	0.2840 ± 0.0019	-0.5914	-0.5907 ± 0.0007	-	-	3
+ CE$^-$1	**0.8826**	**0.8783 ± 0.0043**	**0.2700**	**0.2745 ± 0.0084**	**-0.6126**	**-0.6038 ± 0.0122**	-	-	3
+ CC ($p_A = 0.6$)	0.8027	**0.8899 ± 0.0773**	0.4346	0.7044 ± 0.2336	-0.3681	-0.1855 ± 0.1594	0.5708	0.5869 ± 0.0159	3
+ CC ($p_A = 0.6$)	**0.9260**	**0.9260 ± 0.0028**	0.3411	0.3633 ± 0.0216	-0.5850	-0.5627 ± 0.0231	0.4597	0.4585 ± 0.0064	3
+ DC3 ($p_A = 0.6$)	0.7979	0.8561 ± 0.0910	**0.1908**	**0.2613 ± 0.0960**	**-0.6071**	**-0.5948 ± 0.0108**	0.5782	0.5829 ± 0.0042	3
Turkey									
Baseline	0.8211	0.8172 ± 0.0038	0.3946	0.4253 ± 0.0398	-0.4265	-0.3919 ± 0.0410	-	-	3
+ CE$^-$1	0.7998	0.7998 ± 0.0000	**0.3338**	**0.3338 ± 0.0000**	**-0.4660**	**-0.4660 ± 0.0000**	-	-	3
+ CC ($p_A = 0.6$)	0.7828	0.7987 ± 0.0166	**0.3071**	**0.3266 ± 0.0192**	**-0.4758**	**-0.4722 ± 0.0031**	0.6025	0.5800 ± 0.0289	3
+ DC3 ($p_A = 0.22$)	-	-	-	-	-	-	-	-	0
+ DC3 ($p_A = 0.6$)	**0.8743**	**0.8768 ± 0.0035**	**0.2333**	**0.3071 ± 0.1044**	**-0.6410**	**-0.5697 ± 0.1009**	0.8093	0.9047 ± 0.1348	2
Mice Bone									
Baseline	0.6815	0.6673 ± 0.0123	0.5411	0.5510 ± 0.0150	-0.1403	-0.1164 ± 0.0237	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8801**	**0.8575 ± 0.0228**	**0.3099**	**0.4348 ± 0.1424**	**-0.5702**	**-0.4227 ± 0.1649**	0.5621	0.5266 ± 0.0307	3
STL-10									
Baseline	0.8256	0.8013 ± 0.0169	-	-	-0.8256	-0.8013 ± 0.0169	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8954**	0.7951 ± 0.0856	-	-	**-0.8954**	-0.7951 ± 0.0856	0.5823	0.3425 ± 0.3128	3



(a) Plankton - Collodaria globule (b) CIFAR-10H - Deer

(c) Mice Data - Dissimilar (d) Turkey - Not Injured

Fig. 6: Qualitative Results for selected classes across different confidences and ambiguity predictions – Wrong classifications based on the normal head are highlighted in red. The ground-truth class is given in the subcaption. Larger and extended version of Figure 5.



(a) SSL vs. SSL + DC3 (b) Ablation study

Fig. 7: 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. For the ablation, we evaluate DC3 with the loss CE^{-1} (CE-1), only the classification and clustering without the loss (CC) and the combination (DC3). We usually use the prior probability for the ambiguity (p_A,pA) of 60% but show also an ablation with the realistic prior $\hat{p}_A$ from Table 1.

Table 10: Complete ablation results for Pseudo-Label [32] – The vanilla algorithm is highlighted in light grey. The row below that extend the algorithm with CE$^-$1 [43], Clustering & Classification (CC) or both (DC3). Better results in comparison to the vanilla algorithm are marked bold

	F1		d		(d–F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR-10H									
Baseline	0.6970	0.6914 ± 0.0057	0.5330	0.5359 ± 0.0050	-0.1640	-0.1554 ± 0.0103	-	-	3
+ CE$^-$1	**0.7054**	**0.6977 ± 0.0108**	**0.5194**	**0.5264 ± 0.0113**	**-0.1860**	**-0.1713 ± 0.0221**	-	-	3
+ CC ($p_A = 0.6$)	**0.8265**	0.6583 ± 0.1457	0.8670	0.8838 ± 0.0147	0.0404	0.2255 ± 0.1603	0.6190	0.5803 ± 0.0337	3
+ DC3 ($p_A = 0.32$)	0.6236	**0.6941 ± 0.0611**	**0.2382**	**0.4058 ± 0.1464**	**-0.3854**	**-0.2883 ± 0.0870**	0.3245	0.3237 ± 0.0063	3
+ DC3 ($p_A = 0.6$)	**0.8374**	**0.7448 ± 0.1440**	**0.5132**	**0.4854 ± 0.0967**	**-0.3242**	**-0.2594 ± 0.0618**	0.6100	0.5957 ± 0.0311	3
Plankton									
Baseline	0.8762	0.8730 ± 0.0045	0.2742	0.2821 ± 0.0098	-0.6020	-0.5908 ± 0.0143	-	-	3
+ CE$^-$1	**0.8737**	0.8727 ± 0.0015	0.2788	**0.2794 ± 0.0009**	-0.5949	**-0.5932 ± 0.0024**	-	-	2
+ CC ($p_A = 0.6$)	**0.8919**	**0.9046 ± 0.0220**	0.4085	0.6967 ± 0.2497	-0.4833	-0.2078 ± 0.2400	0.6242	0.5991 ± 0.0221	3
+ DC3 ($p_A = 0.44$)	0.8640	**0.9016 ± 0.0327**	**0.2661**	0.3285 ± 0.0545	-0.5979	-0.5731 ± 0.0217	0.4304	0.4491 ± 0.0167	3
+ DC3 ($p_A = 0.6$)	**0.8931**	0.8539 ± 0.0555	0.3176	0.2844 ± 0.0469	-0.5755	-0.5695 ± 0.0085	0.5945	0.5843 ± 0.0144	2
Turkey									
Baseline	0.8237	0.8245 ± 0.0012	0.4488	0.4527 ± 0.0056	-0.3749	-0.3718 ± 0.0044	-	-	2
+ CE$^-$1	0.7998	0.7998 ± 0.0000	**0.3338**	**0.3338 ± 0.0000**	**-0.4660**	**-0.4660 ± 0.0000**	-	-	3
+ CC ($p_A = 0.6$)	**0.8486**	0.8207 ± 0.0334	**0.3708**	**0.3445 ± 0.0319**	**-0.4778**	**-0.4762 ± 0.0016**	0.6310	0.5947 ± 0.0601	3
+ DC3 ($p_A = 0.22$)	0.7998	0.7998 ± 0.0000	**0.1675**	**0.2291 ± 0.0871**	**-0.6322**	**-0.5706 ± 0.0871**	0.5000	0.4783 ± 0.0307	2
+ DC3 ($p_A = 0.6$)	**0.8344**	**0.8292 ± 0.0074**	**0.3504**	**0.3883 ± 0.0536**	**-0.4841**	**-0.4409 ± 0.0610**	0.8560	0.5305 ± 0.4604	2
Mice Bone									
Baseline	0.6660	0.6524 ± 0.0124	0.5703	0.5768 ± 0.0057	-0.0957	-0.0756 ± 0.0176	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8658**	**0.7275 ± 0.2267**	**0.3752**	**0.3465 ± 0.0978**	**-0.4906**	**-0.3810 ± 0.1363**	0.5444	0.5444 ± 0.0118	3
STL-10									
Baseline	0.8248	0.8016 ± 0.0184	-	-	-0.8248	-0.8016 ± 0.0184	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8887**	0.7903 ± 0.0854	-	-	**-0.8887**	-0.7903 ± 0.0854	0.5921	0.4606 ± 0.2581	3

Table 11: Complete ablation results for FixMatch [49] – The vanilla algorithm is highlighted in light grey. The row below that extend the algorithm with CE^{-}1 [43], Clustering & Classification (CC) or both (DC3). Better results in comparison to the vanilla algorithm are marked bold

	F1		d		(d–F1)		Ambiguous		# Runs
	best	mean ± std	best	mean ± std	best	mean ± std	best	mean ± std	
CIFAR-10H									
Baseline	0.7809	0.7713 ± 0.0097	0.4199	0.4332 ± 0.0122	-0.3611	-0.3381 ± 0.0218	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8309**	0.7947 ± 0.0335	0.4949	0.4746 ± 0.0190	-0.3360	-0.3200 ± 0.0145	0.5805	0.5688 ± 0.0169	3
Plankton									
Baseline	0.8581	0.8324 ± 0.0278	0.3029	0.3237 ± 0.0231	-0.5552	-0.5088 ± 0.0509	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8720**	**0.8666 ± 0.0649**	0.3128	**0.3228 ± 0.0659**	**-0.5592**	**-0.5438 ± 0.0134**	0.5770	0.5782 ± 0.0259	3
Turkey									
Baseline	0.8214	0.8196 ± 0.0019	0.4333	0.4455 ± 0.0121	-0.3881	-0.3741 ± 0.0130	-	-	3
+ DC3 ($p_A = 0.6$)	**0.8356**	**0.8401 ± 0.0146**	**0.2817**	**0.3499 ± 0.0673**	**-0.5539**	**-0.4902 ± 0.0581**	0.2691	0.4827 ± 0.1856	3
STL-10									
Baseline	0.8957	0.8948 ± 0.0011	-	-	-0.8957	-0.8948 ± 0.0011	-	-	3
+ DC3 ($p_A = 0.6$)	**0.9145**	0.8690 ± 0.0440	-	-	**-0.9145**	-0.8690 ± 0.0440	0.5746	0.5586 ± 0.0266	3