Self-supervised Learning from Semantically Imprecise Data

Clemens-Alexander Brust[1][a], Björn Barz[2][b] and Joachim Denzler[1,2][c]

[1]*DLR Institute of Data Science, Jena, Germany*
[2]*Friedrich Schiller University Jena, Jena, Germany*

Abstract: Learning from imprecise labels such as "animal" or "bird", but making precise predictions like "snow bunting" at inference time is an important capability for any classifier when expertly labeled training data is scarce. Contributions by volunteers or results of web crawling lack precision in this manner, but are still valuable. And crucially, these weakly labeled examples are available in larger quantities for lower cost than high-quality bespoke training data. CHILLAX, a recently proposed method to tackle this task, leverages a hierarchical classifier to learn from imprecise labels. However, it has two major limitations. First, it does not learn from examples labeled as the root of the hierarchy, *e.g.,* "object". Second, an extrapolation of annotations to precise labels is only performed at test time, where confident extrapolations could be already used as training data. In this work, we extend CHILLAX with a self-supervised scheme using constrained semantic extrapolation to generate pseudo-labels. This addresses the second concern, which in turn solves the first problem, enabling an even weaker supervision requirement than CHILLAX. We evaluate our approach empirically, showing that our method allows for a consistent accuracy improvement of 0.84 to 1.19 percent points over CHILLAX and is suitable as a drop-in replacement without any negative consequences such as longer training times.

1 INTRODUCTION

High-quality training data labeled by domain experts is an essential ingredient for a successful application of contemporary deep learning methods. However, such data is not always available or affordable in sufficient quantities. And while there exist a number of effective strategies to maximize sample efficiency such as data augmentation, transfer learning, or active learning, there are also limits to the information that can be extracted from a small dataset.

If larger quantities of training data are required, a compromise w.r.t. the quality has to be made, *i.e.,* by allowing noisy labels. Such data is available at lower cost and greater quantity, *e.g.,* by employing volunteer labelers instead of experts or crawling the web for training examples.

In this case, lower quality means that the labels are noisy w.r.t. to two aspects: accuracy and precision. Inaccuracy means that labels are simply incorrect, *i.e.,* confused with other classes. Imprecise labels are correct, but carry less semantic information in terms of

depth in a class hierarchy (see fig. 1), *e.g.,* "animal" vs. "bird". In (Brust et al., 2020), this weakly supervised task is formally defined as "learning from imprecise data". It is defined such that the training data can contain imprecise labels, but predictions must always be as precise as possible, *i.e.,* leaf nodes of the hierarchy. At test time, labels are said to be *extrapolated* (from imprecise to precise) by their method CHILLAX, which we briefly explain in section 3.1.

While their method can perform the task reliably, it has two main limitations. The disadvantages come from the underlying probabilistic hierarchical classifier (Brust and Denzler, 2019). The classifier is modified to perform the extrapolation at test time and to accept imprecise labels during training. However, it cannot learn from examples that are labeled at root of the hierarchy, *e.g.,* as "object", even though it is clearly capable of the necessary extrapolation at test time. And while it can learn from inner node examples, it does not take any advantage of training time extrapolation for such examples either.

Our main contribution in this work is a self-supervised approach to learning from imprecise data based on pseudo-labels. To avoid learning mispredictions and feedback loops, we describe several strate-

[a] https://orcid.org/0000-0001-5419-1998
[b] https://orcid.org/0000-0003-1019-9538
[c] https://orcid.org/0000-0002-3193-3300

27

gies constraining the extrapolation from imprecise labels to more precise pseudo-labels. These strategies are based on prediction confidence scores and on the structure of the hierarchy. We also propose methods that are less sensitive to changes in confidence score distributions over time, which we call adaptive.

The experimental evaluation concerns two areas. First, we assess the potential and limits of extrapolation techniques by examining a best case scenario free of feedback loops. We then evaluate the performance of our methods against CHILLAX and observe the effects of a large range of parameters. The results show consistent improvements of 0.84 to 1.19 percent points in accuracy. All experiments are performed on the North American Birds dataset (Van Horn et al., 2015) for direct comparison to (Brust et al., 2020).

2 RELATED WORK

The task of learning from *semantically imprecise* labels is proposed in (Brust et al., 2020), where a class hierarchy is used to formally define it. It is then tackled using a modified hierarchical classifier (Brust and Denzler, 2019). However, if an image is labeled at the root of the hierarchy, this classifier cannot leverage it. Instead, the image is ignored entirely. This property is a result of the closed world assumption, which most hierarchical classifiers make (Silla and Freitas, 1 01). In this work, we resolve this deficiency by extending the method in (Brust et al., 2020) with a self-supervision scheme.

Labels can be imprecise in other respects, *e.g.,* missing labels in multi-label classification (Abassi and Boukhris, 2020), or a set of labels where only one is expected (Ambroise et al., 2001). An important problem of semantically imprecise labels is that the classes are no longer mutually exclusive. The associated consequences are discussed in detail in (McAuley et al., 2013), although this work does not consider label extrapolation. Instead, it allows imprecise predictions. In (Deng et al., 2012), the authors explicitly mention the trade-off between accuracy and precision (specificity in their terms) and propose an algorithm that can reduce the precision of predictions such that a certain accuracy is guaranteed. This task is the opposite of ours, where the precision of labels is reduced, but the predictions are as precise as possible.

The term *self-supervised learning* has different meanings depending on the specific field. It is commonly used in unsupervised tasks such as visual representation learning (Kolesnikov et al., 2019). In this setting, the supervision that makes training a deep neural network possible comes from solving auxil-
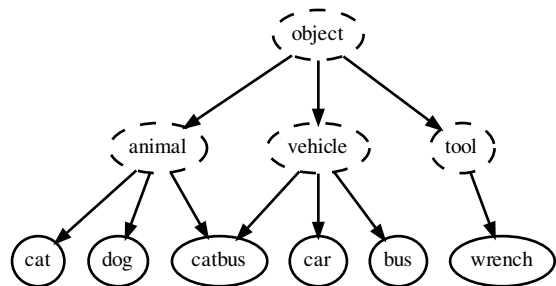

Figure 1: A class hierarchy with semantically imprecise (dashed) and precise (bold) classes.

liary or "pretext" tasks like predicting the previously applied rotation of an image. Real applications should benefit from the representations learned on the auxiliary tasks because unlabeled images are ubiquitous.

Another common interpretation of self-supervision is also known as *pseudo-labeling*, where confident predictions of a model are used as training data (Lee, 2013; Sohn et al., 2020; Wang et al., 2016). We use this definition in our work and focus on interpreting the confidence scores at each level and node in the class hierarchy correctly to maximize the reliability of our pseudo-labels. Auxilliary tasks and pseudo-labeling approaches can also be combined (Zhai et al., 2019).

3 SELF-SUPERVISED METHOD

In this section, we describe our proposed methods and their theoretical background. We first review the concept of semantically imprecise data and the existing method to learn from this data. Then, we introduce our self-supervised approaches.

3.1 Semantically Imprecise Data

Given a class hierarchy, *e.g.,* fig. 1, we can distinguish between semantically precise and imprecise labels. Precise labels are leaf nodes in the hierarchy, while imprecise labels consist of inner nodes and the root. A useful analogy is the number of digits of a measurement. Like the depth in a class hierarchy, a high number does not guarantee an accurate measurement – only a precise one.

The term *imprecise data* is used as a shorthand to describe training data that can contain examples with semantically imprecise labels (Brust et al., 2020). This relaxation does not apply to the predictions, *i.e.,* we still expect them to be as precise as possible. If the goal is to allow more training data to be used in order to improve an existing application, the predictions

should remain unchanged. An extrapolation from imprecise training data to precise output is taking place.

In (Brust et al., 2020), the CHILLAX method based on (Brust and Denzler, 2019) is proposed. Instead of predicting probabilities for each class given some input, a deep neural network predicts the probability of a class s being present (the event Y_s^+), conditioned on both the input image X and the presence $Y_{S'}^+$ of *any* parent class (cf. (Brust and Denzler, 2019, eq. (5))):

$$P(Y_s^+|X) = \underbrace{P(Y_s^+|X,Y_{S'}^+)}_{\text{DNN output}} \cdot P(Y_{S'}^+|X) \qquad (1)$$

This equation is evaluated recursively to obtain the final unconditional probabilities for all "allowed" classes s, *i.e.,* leaf nodes in the hierarchy. The final prediction is the leaf node with the highest probability. Restricting the possible predictions is necessary, as leaf nodes cannot have a higher probability than associated inner nodes owing to the multiplication of more factors that are ≤ 1. Crucially, the recursion ends at the root where the probability is always one, encoding the closed-world assumption. Hence, training examples that are only labeled at the root have no effect on the classifier in terms of loss and thus no value.

3.2 Self-supervised Approach

We propose to make use of effectively unlabeled data at the root of the class hierarchy, and increase the value of other imprecise examples, by extrapolating their labels to pseudo-labels during training. The classifier's own predictions are used to generate the pseudo-labels. Results in (Brust et al., 2020) show that extrapolation at test time is reliable and we examine this in more detail in section 4.2.

However, not all predictions are correct, and using incorrect labels for training can be worse than ignoring the respective example altogether (although controlled amounts of incorrect labels can also be beneficial, see (Xie et al., 2016)). But because we work in a hierarchical setting, there is a middle ground between ignoring and using predictions as ground truth. Instead of leaf nodes, labels can also be extrapolated to internal nodes where there is less potential for confusion. In the following, we propose several methods to determine the appropriate level of extrapolation.

3.2.1 Non-adaptive Methods

While we require that predictions are precise, the same is not true for training data, and crucially, also not for pseudo-labels. We can potentially extrapolate an imprecise label (the source) to a slightly more

precise label (the target), while both are inner nodes of the hierarchy. Our main selection criteria for extrapolation targets are the unconditional probabilities $P(Y_s^+|X)$, which we compute for all classes in the hierarchy. Using predictions for all nodes is clearly not necessary, if the label is sufficiently precise, because that would ignore the label completely. Instead, we replace all predicted probabilities with 1 or 0 where appropriate, which also improves the predictions of related nodes through the recursion. This process ensures that the extrapolation target never "disagrees" with the source. Semantically, the source always subsumes the target.

However, because these unconditional probabilities are always higher for nodes closer to the root, we cannot simply choose the most confident prediction. Instead, we consider the following three approaches that work by building a list of candidates (initially always all classes subsumed by the source class), then excluding some, and finally sorting the remainder to make a selection.

(a) *Leaf Node* Extrapolate any label to the most probable leaf node. Effectively, all predictions are used as training data without any further consideration. This strategy has the highest potential for improvement, but also for inaccuracy. There are also no parameters to tune.

(b) *k Steps Down* Limit extrapolation to exactly k steps "down" the hierarchy from the label, *i.e.,* in the more precise direction. Leaf nodes that are less than k steps down are also allowed because there might not be any possible extrapolations otherwise. The label is selected based on the highest predicted unconditional probability [1]. A threshold on the probabilities can be applied optionally to further exclude unconfident predictions.

(c) *Fixed Threshold* Extrapolate only to labels with a predicted unconditional probability greater than or equal to a given, fixed threshold, *e.g.,* 0.8. We sort all candidates by this probability [1]. Afterwards, a stable sort of the candidate labels by information content (IC, a measure of semantic precision) is performed, such that the candidate with the highest overall IC is first. Thus, if two candidates have the same IC, we select the candidate

---

[1]Note that we add gaussian noise with $\sigma = 0.0001$ to the probabilities before sorting to make it intentionally unstable. This is necessary because many predicted probabilities are exactly 0.5 during initial training as a result of intially zero weights and a sigmoid activation function. If the sorting were stable, the resulting order would be biased by outside factors such as memory layout.

with higher predicted probability. We use the formulation of IC given in (Harispe et al., 2015, 55, eq. 3.8).

If no candidate remains in either (b) or (c), the extrapolation source is used as the target. The approaches (a-c) are *stateless* and require no information other than the extrapolation source, class hierarchy, and predicted probablities from the deep neural network.

3.2.2 Adaptive Methods

A fixed treshold as described in the previous section is intuitive and easy to implement. However, it has major disadvantages. First, it has to be fine-tuned carefully. Second, a threshold that is optimal in one training step may not be optimal for the next because of the continually increasing confidence during training.

While methods (a) and (b) do not rely on a threshold and thus don't suffer from these two effects, those methods are often outperformed by a fixed threshold. The experiments in section 4.3 show that constraining possible extrapolations is critical, and that a fixed structural criterion as in (b) is not always sufficient.

Instead of relying on probabilities directly, we propose considering the difference in IC of the label before and after extrapolation, IC gain in short. IC is a more meaningful measure than hierarchical distance because it also takes global and local properties, *e.g.,* fan-out, of the graph into account. However, we still want to consider the predicted probabilities as an indicator of confidence in certain classes.

To achieve both goals at the same time, we propose two *adaptive* methods. They are adaptive in the sense that the specific selection criteria vary based on the source label:

(d) *Adaptive Threshold* We maintain a moving average of the last 64 IC gains $\bar{h}$ and strive for a target IC gain h^*. This average is used to calculate a probability threshold θ for each time step t (representing weight updates or minibatches) by applying a simple update rule:

$$\theta^{(t)} = \theta^{(t-1)} + \bar{h} - h^*. \quad (2)$$

The threshold is bounded by $[0.55, 1.0]$ and initially set to the lower bound. We then apply the sorting algorithm in (c) to perform the actual extrapolation using the current value of θ.

(e) *IC Range* An interval of allowed IC differences is defined, *e.g.,* $[0.1, 0.3]$, with the target IC gain in the middle. We use the interval to preselect extrapolation candidates. Depending on the extrapolation source, the IC difference range allows varying hierarchical distances to the target, as opposed

to the fixed criterion in (b). Thus, we still consider this approach adaptive even though the parameters stay constant throughout training. A final selection on the preselected candidates is made using (c) with a threshold of 0.55 to remove further spurious predictions.

Both proposed methods make use of the predicted probabilities through application of the algorithm in (c), but rely mainly on its sorting and less on the threshold. Furthermore, (d) is *stateful* because the value of θ needs to be preserved across iterations, which may be a disadvantage from an implementation perspective.

4 EXPERIMENTS

This section contains the experimental evaluation of our methods. We start with a study that determines an upper limit for any gains from self-supervision by examining the pseudo-labels generated by our methods with hierarchical error measures. Then, we test the efficacy of our approaches in a benchmark setting as well as their sensitivity to parameters. This detailed evaluation is seperated into non-adaptive and adaptive approaches as described in section 3.2.1 and section 3.2.2, respectively.

4.1 Setup

We evaluate the effects of adding our proposed self-supervised schemes to the CHILLAX method (Brust et al., 2020). A conventional one-hot softmax classifier is added as a baseline for comparison. Our experimental setup generally matches that of (Brust et al., 2020), except for adjustments to the learning rate (η) and the ℓ_2 regularization coefficient β. We use $\eta = 0.0044, \beta = 5 \cdot 10^{-5}$ and $\eta = 0.003, \beta = 1.58114 \cdot 10^{-5}$ for CHILLAX and the one-hot baseline, respectively.

We report results on the North American Birds (NABirds) dataset (Van Horn et al., 2015) for comparison to the original CHILLAX method. This fine-grained classification dataset consists of approx. 48500 images of 555 species of birds. Crucially, it is also equipped with a class hierarchy. The training portion of the NABirds dataset is modified in different ways according to a selection of noise models from (Brust et al., 2020):

(i) No noise (100% precise labels),

(ii) Relabeling to parent with $p = 0.99$ (Deng et al., 2014) (1% precise labels),

(iii) Geometric distribution with $q = 0.5$ (9.6% precise labels),

(iv) Poisson distribution with $\lambda = 1$ (4.8% precise labels),

(v) Poisson distribution with $\lambda = 2$ (22.7% precise labels).

These noise models cover a wide range of scenarios from expertly labeled high-quality data (i) to web crawling (iii) and volunteer labelers (iv,v). We also include the protocol (ii) which is proposed in (Deng et al., 2014). The models are distributions over depths in the class hierarchy, which are realized by replacing the original precise labels with parent classes to match the distribution. Note that our experiments are limited to label noise in terms of imprecision. Inaccuracy is beyond the scope of this work, but the extent of CHILLAX's relative robustness against inaccurate labels is demonstrated in (Brust et al., 2020).

Source code will be made available publicly upon formal publication.

4.2 Limits of Self-supervision

Self-supervised learning with pseudo-labels is subject to a feedback loop when high-confidence predictions are used as training data. The scores, *e.g.,*:

$$(0.02, 0.73, 0.11, \ldots),$$

are extrapolated to a maximally confident label:

$$(0.0, 1.0, 0.0, \ldots),$$

in addition to a potential extrapolation from one class to another. Subsequent predictions of the same image are likely to be even more confident, thereby increasing the chance of selecting the same target class for training again. This feedback loop can lead to overfitting, learning of potentially false pseudo-labels, and overrepresentation of a subset of training data. While not all methods proposed in this work are subject to this effect, it is important to investigate the potential and limits of extrapolated pseudo-labels in a controlled manner such that feedback effects do not influence the evaluation.

To this end, we train one CHILLAX classifier on a noisy NABirds training set for each of the noise models (i)-(v). We then make predictions for the unseen NABirds validation data, whose labels we also modify using (i)-(v) and use as extrapolation sources.

Table 1 shows the hierarchical F1 (hF1) score between the noise-free ground truth and the extrapolated noisy label from our non-adaptive methods as well as a "do nothing" baseline. In terms of hF1, all methods outperform the baseline. This is not surprising in and of itself. Since no method will select an extrapolation target that disagrees with the imprecise source, it is

impossible for them to perform worse than the baseline, at least in terms of hierarchical recall. Still, the level of outperformance is substantial in all cases and hierarchical precision could still decrease.

Observing noise model (ii), the theoretically largest possible improvement over the baseline is 12.51 percent points (pp), if all labels are correctly extrapolated to their precise origin. The actual improvements realized by our approaches range from 3.32 pp to 7.26 pp over the baseline. In contrast, we observe the largest overall improvement of 28.08 pp over the baseline out of a theoretically possible 40.21 pp in setting (iii). This is the noisiest model, *i.e.,* the one with the lowest expected depth in the class hierarchy.

In all cases, the best evaluated approaches can fill more than half of the performance gap from the baseline to the precise labels in terms of hF1.

We can also evaluate the extrapolated labels in terms of classification accuracy, but this is only possible for the "leaf node" approach. The other approaches produce imprecise labels as output which can only be compared to the precise validation set in terms of hierarchical measures. The results are presented in table 2. We include a variation where even the noisy label is withheld from the method, such that all labels must be extrapolated from the root (effectively unsupervised learning). Including the noisy label produces an improvement in accuracy of up to 13.9 pp and is always beneficial.

Even in the worst case (iv), more than half of the labels that are extrapolated as far as possible are correct. This is a strong result considering the 555 classes and only 4.8% precise labels in the training set.

4.3 Non-adaptive Methods

The study in section 4.2 shows that most cases, except possibly (ii), have a strong potential for improvement by using extrapolated predictions as pseudo-labels. However, a high hierarchical F1 score in one step does not necessarily generalize to high accuracy when using self-supervision continuously during training. On the one hand, the aforementioned feedback loop could negatively affect training by overfitting and unbalancing the training data. On the other hand, the quality of extrapolations might improve over time as the model learns from correct pseudo-labels.

We first apply the non-adaptive methods described in section 3.2.1 to CHILLAX on NABirds. Ground truth labels are replaced with pseudo-labels at all times during training. The resulting model is then evaluated in terms of accuracy on the NABirds validation set. We repeat the experiment six times per

Table 1: Hierarchical F1 (%) on NABirds validation set. Comparison between ground truth and extrapolated noisy validation labels after learning noisy training data. (i) is not included as all results are 100%.

Method / Noise	(ii)	(iii)	(iv)	(v)
Baseline (No Extrapolation)	87.49	59.79	61.73	78.06
1 Step Down	94.74	73.97	74.81	87.27
2 Steps Down	94.74	79.29	79.03	91.31
3 Steps Down	94.72	80.88	79.29	92.91
Fixed Threshold 0.55	90.99	**82.87**	**81.37**	**93.66**
Fixed Threshold 0.8	90.72	81.19	80.03	92.99
Leaf Node	**94.75**	81.15	79.12	93.20

Table 2: Accuracy (%) on NABirds validation set. Comparison between ground truth and extrapolated noisy validation labels after learning noisy training data. (i) is not included as all results are 100%.

Method / Noise	(ii)	(iii)	(iv)	(v)
Leaf Node	76.99	58.52	51.03	83.81
Leaf Node From Root	63.09	49.55	43.36	70.99

individual setting and include a CHILLAX baseline for comparison.

The results are shown in table 3, where we first observe the threshold-based extrapolation method. It is very sensitive to the confidence threshold and requires substantial fine-tuning. The optimal threshold strongly depends on the noise model. For example, the Poisson noise (iv) only has a working range of thresholds from 0.97 to 0.998 where it matches or outperforms the baseline, with the optimum at 0.99.

Overall, improvements w.r.t. the baseline range from 0.84 pp to 1.05 pp.

The "leaf node" and "steps down" methods are somewhat competitive, specifically for the geometric noise model (iii). Going k steps down the hierarchy is only beneficial when combined with a confidence threshold. However, this combination suffers from the aforementioned fine-tuning problem.

Always selecting the most confident leaf node leads to a small improvement in all cases and requires no tuning.

4.4 Adaptive Self-supervision

This experiment compares our two proposed adaptive methods of limiting the increase in IC from extrapolation source to target (see section 3.2.2 for details). The first proposed adaptive method "adaptive threshold" changes a confidence threshold dynamically to achieve a given expected IC gain. Our experiment uses expected gains of $0.025, \ldots, 0.1$, where our choice of IC is naturally bounded between 0 and 1. The second method "IC range" uses a fixed range of allowed IC differences. We use the ranges $[0, 0.2], [0.1, 0.3], \ldots, [0.4, 0.6]$ and a minimum confi-

dence of 0.55 to reject spurious predictions. We perform six training repetitions for each combination of method and noise model.

Table 4 compares the results of both methods. Our adaptive threshold method performs better than the fixed range approach in the noisier settings (iii)-(v), even outperforming the fine-tuned non-adaptive fixed threshold method on setting (v) with in improvement in accuracy of 1.19 pp.

Furthermore, the parameter of our adaptive threshold method is much less sensitive to changes than the fixed threshold as evidenced by the large effective range.

The fixed IC gain range setup only works well for noise model (ii), which is the immediate parent relabeling scenario from (Deng et al., 2014). This result is expected, because this noise model leaves only two possibilities for IC gain. There can either be no gain at all, or the fixed amount when moving from the second to last level in the hierarchy to the last level. As such, the model fits the assumption of a fixed IC gain range perfectly. The other noise models lead to partly catastrophic results when the fixed range effectivly prohibits any extrapolation. However, if the noise distribution is known before training, it could be argued that setting a correct range is more straightforward.

Overall, we observe that a fixed threshold performs best, but only after significant fine-tuning. **Our "adaptive threshold" method is less sensitive to changes in its parameters, and performs slightly better than the parameter-free "leaf node" approach, which is why we recommend it as a drop-in replacement for fully supervised CHILLAX.**

Table 3: Accuracy (%) on NABirds validation set. Comparison between non-adaptive self-supervised methods on noisy training data. Baseline (i) result 81.63 ± 0.12.

Method / Noise	(ii)	(iii)	(iv)	(v)
CHILLAX Baseline	62.66 ± 0.82	49.04 ± 1.04	43.18 ± 0.20	70.91 ± 0.34
Leaf Node	63.05 ± 1.37	49.36 ± 0.48	43.49 ± 0.20	70.94 ± 0.42
k Steps Down				
1	61.78 ± 0.27	33.11 ± 0.87	23.49 ± 0.60	65.44 ± 0.83
1, conf. ≥ 0.8	61.75 ± 0.69	48.09 ± 0.75	40.85 ± 1.26	71.67 ± 0.23
1, conf. ≥ 0.9	63.13 ± 0.70	49.98 ± 0.55	41.54 ± 1.37	71.75 ± 0.26
2	61.31 ± 0.68	14.53 ± 0.82	12.12 ± 0.96	59.58 ± 0.60
2, conf. ≥ 0.8	62.07 ± 0.43	48.31 ± 0.84	37.21 ± 0.72	71.74 ± 0.81
2, conf. ≥ 0.9	62.52 ± 0.68	$\mathbf{50.68} \pm 0.44$	41.78 ± 0.47	71.54 ± 0.33
Threshold				
0.55	61.48 ± 0.36	26.56 ± 0.94	22.68 ± 0.29	65.32 ± 0.53
0.8	61.73 ± 0.54	39.80 ± 0.97	31.15 ± 1.65	69.86 ± 1.23
0.85	61.93 ± 0.25	43.35 ± 0.72	34.60 ± 0.73	70.59 ± 0.17
0.9	62.34 ± 0.33	46.74 ± 1.27	38.03 ± 0.78	$\mathbf{71.77} \pm 0.00$
0.95	62.75 ± 0.21	48.66 ± 1.03	42.11 ± 1.48	71.47 ± 0.44
0.97	63.00 ± 0.58	50.09 ± 0.26	43.20 ± 0.57	71.40 ± 0.25
0.99	63.51 ± 0.52	49.37 ± 0.28	$\mathbf{44.02} \pm 0.12$	71.14 ± 0.15
0.992	63.02 ± 0.57	48.88 ± 0.65	43.78 ± 0.33	71.21 ± 0.51
0.994	$\mathbf{63.54} \pm 0.51$	49.23 ± 0.55	43.61 ± 0.94	70.99 ± 0.27
0.996	63.11 ± 0.60	49.37 ± 0.61	43.69 ± 1.00	71.08 ± 0.45
0.998	63.10 ± 0.83	49.16 ± 0.63	43.81 ± 0.25	70.96 ± 0.22
0.999	62.78 ± 0.48	49.56 ± 1.04	42.92 ± 0.68	71.37 ± 0.39

Table 4: Accuracy (%) on NABirds validation set. Comparison between adaptive self-supervised methods on noisy training data. Baseline (i) result 81.63 ± 0.12. **Best adaptive result**, <u>Best overall result</u>.

Method / Noise	(ii)	(iii)	(iv)	(v)
CHILLAX Baseline	62.66 ± 0.82	49.04 ± 1.04	43.18 ± 0.20	70.91 ± 0.34
Best non-adaptive	<u>63.54 ± 0.51</u>	<u>50.68 ± 0.44</u>	<u>44.02 ± 0.12</u>	71.77 ± 0.00
Adaptive Threshold				
0.025	61.80 ± 0.69	49.07 ± 0.93	$\mathbf{43.35} \pm \mathbf{0.82}$	71.58 ± 0.40
0.0375	61.43 ± 0.51	$\mathbf{49.75} \pm \mathbf{1.13}$	43.11 ± 1.49	72.00 ± 0.43
0.05	61.80 ± 0.44	49.52 ± 0.90	42.92 ± 0.42	$\mathbf{72.10} \pm \mathbf{0.31}$
0.0625	61.87 ± 0.91	49.57 ± 1.55	42.30 ± 0.35	71.15 ± 0.92
0.075	62.20 ± 0.76	49.97 ± 0.59	41.71 ± 1.13	71.37 ± 0.40
0.1	61.79 ± 0.30	48.57 ± 0.61	39.72 ± 0.96	70.77 ± 0.34
Fixed Range				
$[0, 0.2]$	62.11 ± 0.44	46.42 ± 0.71	40.50 ± 0.50	68.39 ± 0.47
$[0.1, 0.3]$	61.78 ± 0.42	29.77 ± 0.98	26.07 ± 0.67	64.31 ± 1.05
$[0.2, 0.4]$	$\mathbf{63.43} \pm \mathbf{0.42}$	36.00 ± 1.29	30.64 ± 0.86	68.61 ± 0.22
$[0.3, 0.5]$	63.23 ± 0.10	35.00 ± 0.46	31.45 ± 0.60	68.55 ± 0.94
$[0.4, 0.6]$	62.96 ± 0.98	33.24 ± 0.65	27.31 ± 1.41	68.46 ± 0.33

5 CONCLUSION

Learning from imprecise data is proposed in (Brust et al., 2020) as a way of maximizing training data utilization. However, their method CHILLAX does not utilize *all* training data as it ignores examples labeled at the root. This is a consequence of the closed-world assumption made by the underlying classifier. To avoid such meaningless labels, we propose to use CHILLAX's label extrapolation not just at test time,

but during training to generate pseudo-labels. This increases the precision of examples labeled not only at the root, but also at inner nodes of the class hierarchy.

To implement this self-supervised learning scheme, we describe several possible strategies of deciding which candidate pseudo-labels are reliable enough for training. These strategies employ heuristic, structural and statistical criteria. Our experiments show that an increase in accuracy of around one percent point can be expected by simply using one of our self-supervised strategies on top of CHILLAX. This improvement comes without any requirement of fine-tuning unrelated parameters or undue computational efforts.

Future Work. In the future, these methods could also be applied to semi-supervised learning tasks in general, *e.g.,*, by assigning a root label to the unlabeled images as long as a closed-world scenario can be assumed. Furthermore, the individual heuristics could be combined into a meta-heuristic. In contrast, relaxing the closed-world assumption is another important research direction. Asking a hierarchical classifier for its confidence in the root node is a first step towards open-set models from a semantic perspective, as long as the predicted confidence has a reasonable basis. A fixed hierarchy is a further limiting assumption, which could be relaxed, *e.g.,*, in a lifelong learning setting.

The research on semantically imprecise data in general could be expanded to domains beyond natural images. For example, we expect source code to have a stronger feature-semantic correspondence, which is crucial for the hierarchical classifier. In particular, human-made hierarchies such as the Common Weakness Enumeration (CWE, (The MITRE Corporation, 2021)) explicitly consider certain features of program code to determine categories. And even in the visual domain, there are efforts to construct more visual-feature-oriented hierarchies, *e.g.,*, accompanying WikiChurches (Barz and Denzler, 2021).

ACKNOWLEDGMENTS

REFERENCES

Abassi, L. and Boukhris, I. (2020). Imprecise label aggregation approach under the belief function theory. In Abraham, A., Cherukuri, A. K., Melin, P., and Gandhi, N., editors, *Intelligent Systems Design and Applications*, pages 607–616. Springer International Publishing.

Ambroise, C., Denœux, T., and Govaert, G. (2001). Learning from an imprecise teacher: probabilistic and evidential approaches. *Applied Stochastic Models and Data Analysis*, page 6.

Barz, B. and Denzler, J. (2021). Wikichurches: A fine-grained dataset of architectural styles with real-world challenges. In *Neural Information Processing Systems (NeurIPS)*.

Brust, C.-A., Barz, B., and Denzler, J. (2020). Making every label count: Handling semantic imprecision by integrating domain knowledge. In *International Conference on Pattern Recognition (ICPR)*.

Brust, C.-A. and Denzler, J. (2019). Integrating domain knowledge: using hierarchies to improve deep classifiers. In *Asian Conference on Pattern Recognition (ACPR)*.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., and Adam, H. (2014). Large-scale object classification using label relation graphs. In *European Conference on Computer Vision (ECCV)*.

Deng, J., Krause, J., Berg, A. C., and Fei-Fei, L. (2012). Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR)*.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2015). Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254.

Kolesnikov, A., Zhai, X., and Beyer, L. (2019). Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lee, D.-H. (2013). Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning Workshops (ICML-WS)*, page 6.

McAuley, J. J., Ramisa, A., and Caetano, T. S. (2013). Optimization of robust loss functions for weakly-labeled image taxonomies. *International Journal of Computer Vision (IJCV)*, 104(3):343–361.

Silla, C. N. and Freitas, A. A. (2011-01). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence.

The MITRE Corporation (2021). Common Weakness Enumeration (CWE).

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. (2015).

Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604.

Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2016). Cost-effective active learning for deep image classification. *Circuits and Systems for Video Technology (CSVT)*, PP(99):1–1.

Xie, L., Wang, J., Wei, Z., Wang, M., and Tian, Q. (2016). Disturblabel: Regularizing cnn on the loss layer. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. (2019). S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.