A Appendix

A.1 Other Datasets

Below we describe a few other prevalent multi-label datasets and explain how the ML48S differs from them, hence they were excluded from comparison in this paper.

PASCAL VOC [11] was created for object detection and classification, covering 20 basic-level classes across 4,574 images, with most images containing a single prominent object. This dataset is much smaller than ML48S and also contains much fewer classes which are all coarse-grained.

VG500 is a modification of the Visual Genome dataset [19], a dataset focused on dense annotations linking images to respective captions. This dataset is not intended to be bounded by categories but has open-vocabulary annotations. To turn this into a multi-label task, only the top 500 most frequent categories are kept to make VG500, following the work in [21]. We choose not to compare to this dataset because the open-vocabulary nature of the task leaves ambiguity in annotations but no clarification is given between explicit negatives and unknowns.

OpenImages [20] is a large-scale dataset with 14.6M boxes across 1.7M images spanning over 600 categories. Similar to Visual Genome, a semantic hierarchy is given and both positives and negatives are given explicitly. This dataset is similar to ML48S in nature, but differs entirely in scale, containing about 10 times in the number of images in the training set. Since this dataset is used in a completely different context to ML48S due to the size, we chose not to compare to this dataset.

NUS-WIDE [5] is another multi-label dataset based on publicly-available internet images. This dataset contains images from Flickr which are labeled with corresponding tags for 81 concepts. This dataset is no longer available in its entirety due to many of the associated images being no longer accessible on Flickr. In addition, not all of the concepts are object-centric and can be associated with a bounding box, including abstract concepts such as "protest" and less clearly explicit events such as "earthquake." Based on these issues and differences from ML48S, we excluded NUS-WIDE in our comparison.

WIDER-Attributes [22] is a dataset focused on classifying human attributes, but only focuses on 14 attributes per person in an image. This task is much less fine-grained than the ML48S and contains far fewer classes than ML48S, which led to its exclusion in our analysis.

Caltech-UCSD-Birds (CUB200) [41] is conventionally used as a classification dataset, but can also be treated as an attribute prediction task for each bird. However, these attributes are non-binary (such as the shape of the bill being curved, hooked, cone, etc.), so to transform this into a multi-label problem, each of these attributes must be turned into a set of binary attributes equal to the number of choices where they are mutually exclusive. This is not an object-centric task like the ML48S, and we believe turning multiple classification problems into a single multi-label problem is contrived so we exclude it from our comparisons.

Visual Privacy (VISPR) [30] is a dataset which identifies personally revealing information within images, where each category signifies whether a given personal characteristic can be found within an image. While some of these attributes are explicit to identify such as phone number and eye color, others are abstract, such as religion, personal relationships, and hobbies. We primarily exclude this from our analysis because the labels are not object-centric like in ML48S and are more difficult to interpret.

A.2 ML48S Additional Information

We organize the ML48S by images in sets which come from recordings, which we also call clips and assets, respectively. We outline the metadata associated with each image and recording as well as our spectrogram generation process below.

A.2.1 Spectrogram Generation

To generate spectrograms from 1D waveforms, we use the Short-Time Fourier Transform with a window size of 512 and stride length of 128. This spectrogram is then converted to individual images which span 3 seconds and are disjoint. To input the spectrogram into our network, we copy the spectrogram into three channels and resize it to shape $448 \times 448 \times 3$.

Field	Possible Values	Description
id	[0, 9999]	The unique ID associated with the asset
split	[train, test]	Denotes training split or test split for an asset
target_species_code	6-letter-code	The target species for this asset
possible_species_codes	[6-letter-codes]	A list of possible species based on ranges
observed_species_codes	[6-letter-codes]	A list of species in the affiliated checklist
present_species_codes	[6-letter-codes]	A list of positively labeled species
unknown_species_codes	[6-letter-codes]	All species not in present or absent lists
absent_species_codes	[6-letter-codes]	A list of negatively labeled species

Table A1: A summary of asset metadata and their possible values.

Field	Possible Values	Description
id	[0, 416534]	The unique ID associated with the clip
asset_id	[0, 9999]	The asset ID from which this clip came
clip_order	[0, 1449]	The position of the clip within the asset
file_path	Relative filepath	The path to the image for the given clip
width	750	The image width
height	236	The image height
present_species_codes	[6-letter-codes]	A list species with positive labels
unknown_species_codes	[6-letter-codes]	All species not in present or absent lists
absent_species_codes	[6-letter-codes]	A list species with negative labels
boxes	[dictionaries]	Bounding box annotations for the clip, see Table A3

Table A2: A summary of clip metadata and their possible values.

A.2.2 Asset Metadata

Assets have associated metadata which we summarize in Table A1 and also explain in detail below.

Each asset is associated with a unique asset ID from 0 to 9999. Assets with an ID greater than or equal to 8000 are test assets, and each species has 80 training assets and 20 test assets. For our experiments, we randomly selected 10 training assets per species to serve as validation assets for hyperparameter tuning (given in the repository). Each asset contains a variable number of clips, with a minimum of 11 and a maximum of 1450. As discussed in the paper, every asset has a target species which is provided in the form of a 6-character target species code. The corresponding taxonomic information such as phylogeny, common name, and scientific name are given in `taxa.csv`.

Assets also contain compiled lists of positives, negatives, and unknowns, where positives are also known as present species and negatives are also known as absent species. The list of positives is the union of positives given across each clip in the asset, while the list of negatives is the intersection of clip negatives. The list of unknown species contains the species which are not in either of the previous two lists.

Assets also contain two additional fields, possible species given by geographic priors and observed species within the associated checklist. Using the location and time of year each recording was taken, we are able to generate a list of possible species based on species ranges. Though this list does not provide positive labels, absence of a species on this list implies a negative label for that species across the entire recording. This logic also applies for observed species within the associated checklist. Any species present in the recording should also be reported in the associated checklist, so species not on the checklist should have negative labels for the recording. The negative labels generated through checklist data is a superset of the negative labels generated from geographical priors. Hence, geographical priors and checklist data provide two additional levels of weak supervision which falls between SPML and full-labels. We apply negative labels from geographical and range priors to the clip level, even for unlabeled data.

Field	Possible Values	Description
`id`	int	Box ID unique to each clip
`species_code`	6-letter-code	The species which this vocalization belongs to
`status`	[“passive”, “active”, “ignore”]	Species prevalence in the clip
`bbox`	$[0, 1]^4$	Box coordinates [xmin, ymin, xmax, ymax]

Table A3: A summary of box data and their possible values.

Dataset	Boxes/image	Small	Medium	Large
VOC	3.28	2.96%	19.79%	77.24%
COCO	9.17	19.95%	34.36%	45.69%
ML48S +	2.38	0.97%	7.85%	91.18%

Table A4: An overview of each datasets' box statistics in terms of sizes and quantities for the training set. To standardize which boxes are small, medium, and large, we resize each image and its bounding boxes such that the minimum dimension of the image is 640, then we threshold by bounding box area. Small boxes have area less than 32^2, large boxes have area greater than 96^2, and all other boxes are medium boxes. ML48S + signals images with no boxes are not considered.

A.2.3 Clip Metadata

Clips also have corresponding metadata which is summarized in Table A2. The bounding box annotations for each clip are provided, where each box is specified with an ID, species code, status, and coordinates. The box ID is unique to a clip, so no two boxes within the same clip share the same ID. The bounding box coordinates are given in relative coordinates falling within [0, 1] and are provided as [xmin, ymin, xmax, ymax]. For box status, sounds which are longer than 80 ms which are only present in the first or last 200 ms of a window are labeled “ignore” while others are “active.”

Boxes which do not have status “ignore” are treated as positive labels for the multi-label task and are given in the list of positives. Any clip with positive labels is treated as fully-labeled, meaning all other species are negative, unless there are “Unknown bird” boxes, in which case we put treat other possible species as unknown (but retain negatives from geographical priors).

A.3 Additional Dataset Statistics

In this section, we compare additional statistics of the ML48S to VOC and COCO not covered in the main paper.

A.3.1 Bounding Box Statistics

We give basic statistics of bounding boxes quantity and sizes in Table A4. ML48S is most similar to VOC among the datasets which we compare to. On average, an image contains 2.25 boxes, and the vast majority of these boxes are usually large. This likely occurs because most vocalizations in a spectrogram span a wide range of frequencies due to overtones, so most boxes have a height comparable to the image height. The duration of these vocalizations can vary, depending on whether they encompass a single call or a longer bird song. These distributions are also visualized in Figure A1.

A.3.2 Known and Unknown Label Statistics

We give statistics for the breakdown of images which are fully-labeled, images which contain at least one positive label, and images with any labels in Table A5. Though negative labels are generated for all images using the metadata outlined in Section A.2.2, bounding boxes are all hand-drawn by expert annotators, who focus on annotating various segments of a recording instead of the entire thing. Hence, only 45% of the training set is fully-annotated for all species. Our original data contained "Unknown Bird" boxes for vocalizations which were unable to be identified to species level. As a result, we cannot generate negative labels reliably for these images, and they remain partially-labeled despite containing positive labels. We train our model with all images with at least one positive

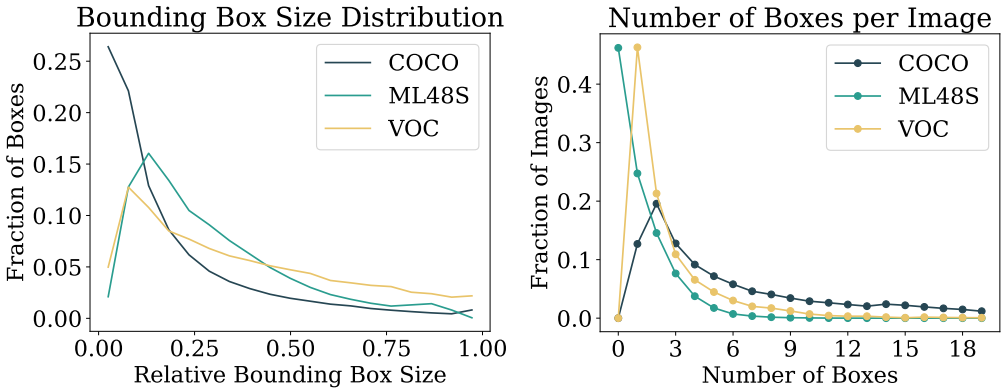

Figure A1: Visualization of bounding box distributions for each dataset. The left plot shows the bounding box size distribution, where the relative size gives the area of the box divided by the total image area. The right box shows the number of boxes per image. ML48S mirrors the distribution of VOC closely in terms of boxes per image, but has a unique bounding box size distribution.

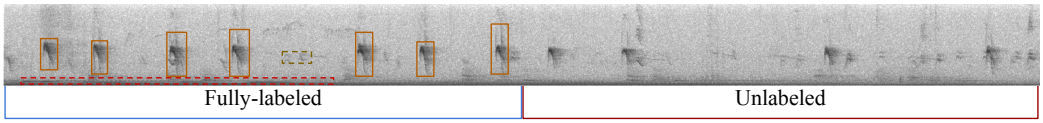

Figure A2: An excerpt from a partially-labeled asset in ML48S. The first half of this snippet is fully-labeled while the last half is unlabeled. For our experiments we train only on the first half, but we release the full asset for future work on semi-supervised and unsupervised learning. The vocalizing birds are Mourning Dove, Canyon Wren, and House Finch in order of first appearance from left to right.

label, which is 53% of the dataset. We do not use the remaining data for training in this paper, but we include it in the dataset release for future work. One such example is shown in Figure A2. Furthermore, the distributions of unknown labels are visualized in Figure A3.

A.3.3 Positive and Negative Label Statistics

In Table A6 we give positive and negative label statistics across all splits of each datasets. All images in ML48S contain at least 24 negative labels derived from metadata discussed in Section A.2.2. We also plot the distributions of positives, negatives, and unknowns individually for each dataset in Figure A3. ML48S shows a bimodal distribution for negatives and unknowns, because each image is either fully-labeled or labels are generated through metadata. The negative labels generated by checklist and location data vary, but on average around 45 negative labels can be generated through this method.

Known Labels	# Images	% Images
Fully-labeled	38,975	45.75%
At least one box	45,178	53.03%
Any labels	85,193	100%

Table A5: ML48S degree of annotation for the training set. All images contain negative labels generated from checklist and geographic information, but positives labels must be manually labeled. Images with "Unknown Bird" labels are not considered fully-labeled.

Dataset	Split	# Images	+ (min)	+ (max)	+ (avg)	+ (med)	- (min)	- (max)	- (avg)	- (med)
VOC	Train	4574	1	5	1.46	1	15	19	18.54	19
VOC	Val	1143	1	5	1.46	1	15	19	18.54	19
VOC	Test	5823	1	5	1.43	1	15	19	18.57	19
VOC	All	11540	1	5	1.45	1	15	19	18.55	19
COCO	Train	65665	1	18	2.94	2	62	79	77.06	78
COCO	Val	16416	1	16	2.92	2	64	79	77.08	78
COCO	Test	16416	1	16	2.92	2	64	79	77.08	78
COCO	All	98497	1	18	2.93	2	62	79	77.07	78
ML48S	Train	85193	0	8	0.84	1	24	100	69.59	59
ML48S	Train+	45178	1	8	1.58	1	24	99	85.33	98
ML48S	Val	12448	0	7	0.81	1	25	100	67.72	53
ML48S	Test	31365	0	8	0.78	0	24	100	68.73	53
ML48S	All	129006	0	8	0.82	1	24	100	68.47	56

Table A6: An overview of each datasets' positive and negative labels in terms of minimum per image, maximum per image, average, and median for training, validation, and testing splits as well as all three splits combined. "+" signifies the number of positive labels and "-" signifies the number of negative labels. The number of unknown labels can implicitly be calculated using these two values by subtracting by the total number of classes for the dataset. For ML48S, "Train+" signifies the training set with images with at least one positive. On VOC and COCO, the validation sets used are the ones used in our experiments, which are a randomly selected subset of the original training set.

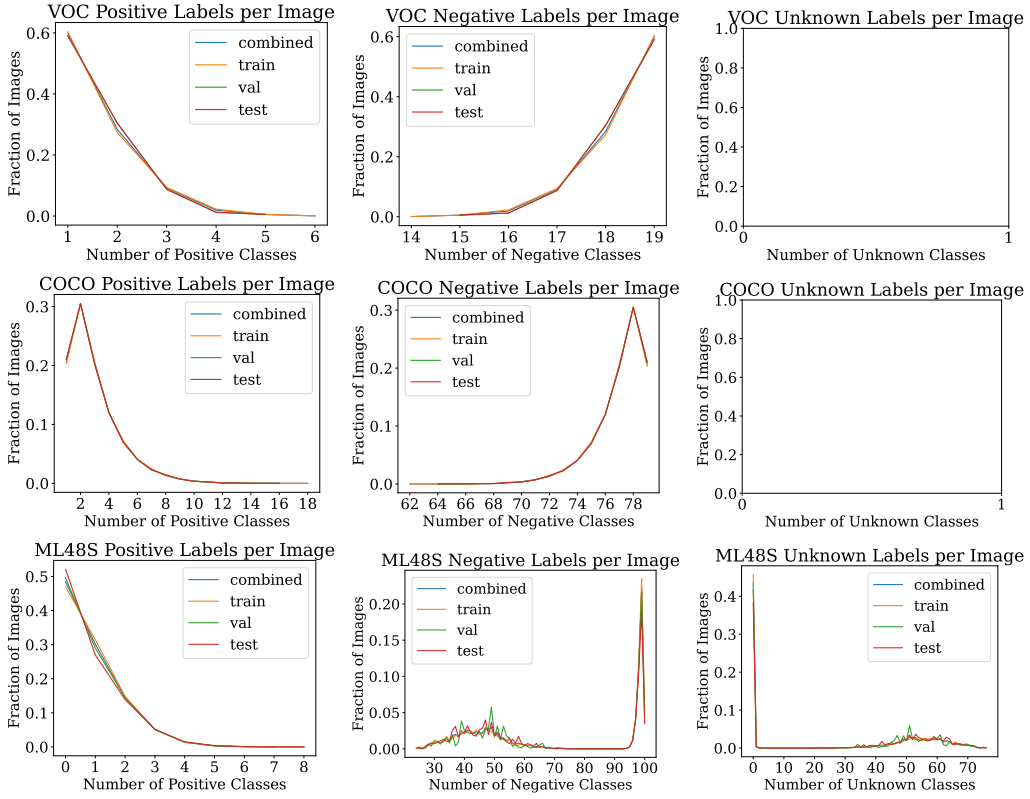

Figure A3: Visualization of dataset positives, negatives, and unknown labels per image for each split and the combined splits. For COCO and VOC, unknown label graphs are left blank because all images in these datasets are fully-labeled.

Method	Dataset	Learning Rate	Method Hyperparameter
BCE	COCO	$1e-5$	N/A
BCE-AN	COCO	$1e-5$	N/A
WAN	COCO	$1e-5$	$\gamma = 1/79$
LS	COCO	$1e-5$	$\epsilon = 0.1$
ROLE	COCO	$1e-5$	$\lambda = 1$
EM	COCO	$1e-5$	$\alpha = 0.1$
LL-R	COCO	$1e-5$	$\Delta_{\text{rel}} = 0.4$
LL-Ct	COCO	$1e-5$	$\Delta_{\text{rel}} = 0.2$
LL-Cp	COCO	$1e-5$	$\Delta_{\text{rel}} = 0.2$
BCE	ML48S	$1e-4$	N/A
BCE-AN	ML48S	$1e-4$	N/A
WAN	ML48S	$1e-4$	$\gamma = 1/99$
LS	ML48S	$1e-4$	$\epsilon = 0.1$
ROLE	ML48S	$1e-4$	$\lambda = 1$
EM	ML48S	$1e-4$	$\alpha = 0.2$
LL-R	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$
LL-Ct	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$
LL-Cp	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$

Table A7: Testing hyperparameters used for the target-only regime.

A.4 Hyperparameters

We use mean average precision (mAP, *i.e.* the mean of per-class average precision), as our evaluation metric. For COCO, we use 20% of the training set as a validation set for hyperparameter tuning. For the ML48S, we select 10 training assets per species to make up the validation set which are specified in the repository. Since ML48S has incomplete labels, we calculate mAP only using images which have labels for all species.

Following the training procedure of prior work [7, 18, 47], each of our experiments was trained using an ImageNet [10] pretrained ResNet50 [12] architecture using the Adam optimizer on Pytorch. Prior works [4, 37] have shown substantial improvements from ImageNet pretraining for spectrogram classification despite the domain shift. We preprocess each image by resizing the image to shape (448, 448) and normalizing the image to ImageNet statistics. For COCO only, at training time, we flip the image horizontally at random. We train for 10 epochs using a fixed batch size of 16 and a constant learning rate, which we sweep using values in $\{1e-2, 1e-3, 1e-4, 1e-5\}$. For WAN on ML48S we found convergence was slower so we trained these experiments for 20 epochs instead of 10. We monitor performance on the validation set, and the best performing configuration is used for evaluation on the test set. For other SPML methods, to reduce the amount of trials required for hyperparameter tuning, we first tune the learning rate of each loss function with the hyperparameters reported for each method on COCO before sweeping the suggested range of hyperparameters given in each respective work. Once these settings are chosen, each experiment is repeated 5 times to calculate mean and standard deviation performance. The settings used in each of our experiments can be found in Tables A7, A8, and A9. For COCO experiments, we use a different randomly-generated SPML dataset each time, though these are the same across methods. For ML48S experiments, we only train with images containing at least one positive, meaning we remove images with only confirmed negatives. Following [18], we increase the learning rate of the last layer by 10x for training the LL-variants.

For $\mathcal{R}_P$ hyperparameters, we run a grid search with $\alpha \in \{1e-1, 1e-2, 1e-3\}, \epsilon \in \{1e-2, 1e-3, 1e-4\}$. We initialize $\overline{y_0^i}$ following ROLE initialization [7], $\overline{y_0^i} \sim \mathcal{U}(0.4, 0.6)$.

Method	Dataset	Learning Rate	Method Hyperparameter	α	ϵ
BCE	ML48S	$1e-4$	N/A	$1e-1$	$1e-2$
BCE-AN	ML48S	$1e-4$	N/A	$1e-1$	$1e-3$
WAN	ML48S	$1e-4$	$\gamma = 1/99$	$1e-2$	$1e-3$
LS	ML48S	$1e-4$	$\epsilon = 0.1$	$1e-2$	$1e-3$
ROLE	ML48S	$1e-4$	$\lambda = 1$	$1e-1$	$1e-4$
EM	ML48S	$1e-4$	$\alpha = 0.2$	$1e-1$	$1e-4$
LL-R	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$	$1e-1$	$1e-2$
LL-Ct	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$	$1e-2$	$1e-2$
LL-Cp	ML48S	$1e-4$	$\Delta_{\text{rel}} = 0.1$	$1e-2$	$1e-4$

Table A8: Testing hyperparameters used for asset regularization.

Method	Dataset	Learning Rate	Method Hyperparameter	a	b
WAN	COCO Geo	$1e-5$	$\gamma = 0.1$	0	0.01
LS	COCO Geo	$1e-5$	$\epsilon = 0.2$	0	0.05
ROLE	COCO Geo	$1e-5$	$\lambda = 0.1$	0	0.01
EM	COCO Geo	$1e-5$	$\alpha = 0.1$	1	0.01
LL-R	COCO Geo	$1e-5$	$\Delta_{\text{rel}} = 0.4$	N/A	N/A
LL-Ct	COCO Geo	$1e-5$	$\Delta_{\text{rel}} = 0.2$	N/A	N/A
LL-Cp	COCO Geo	$1e-5$	$\Delta_{\text{rel}} = 0.2$	N/A	N/A
WAN	COCO Checklist	$1e-5$	$\gamma = 0.1$	1	0.01
LS	COCO Checklist	$1e-5$	$\epsilon = 0.1$	1	0.5
ROLE	COCO Checklist	$1e-5$	$\lambda = 1$	1	1
EM	COCO Checklist	$1e-5$	$\alpha = 0.1$	1	0.02
LL-R	COCO Checklist	$1e-5$	$\Delta_{\text{rel}} = 0.4$	N/A	N/A
LL-Ct	COCO Checklist	$1e-5$	$\Delta_{\text{rel}} = 0.2$	N/A	N/A
LL-Cp	COCO Checklist	$1e-5$	$\Delta_{\text{rel}} = 0.2$	N/A	N/A
WAN	ML48S Geo	$1e-4$	$\gamma = 0.05$	1	0.5
LS	ML48S Geo	$1e-4$	$\epsilon = 0.1$	1	0.2
ROLE	ML48S Geo	$1e-4$	$\lambda = 0.5$	0	0.05
EM	ML48S Geo	$1e-4$	$\alpha = 0.1$	0	0.01
LL-R	ML48S Geo	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A
LL-Ct	ML48S Geo	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A
LL-Cp	ML48S Geo	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A
WAN	ML48S Checklist	$1e-4$	$\gamma = 1/99$	0	0.5
LS	ML48S Checklist	$1e-4$	$\epsilon = 0.1$	1	1
ROLE	ML48S Checklist	$1e-4$	$\lambda = 2$	0	0.05
EM	ML48S Checklist	$1e-4$	$\alpha = 0.02$	1	0.01
LL-R	ML48S Checklist	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A
LL-Ct	ML48S Checklist	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A
LL-Cp	ML48S Checklist	$1e-4$	$\Delta_{\text{rel}} = 0.1$	N/A	N/A

Table A9: Testing hyperparameters used for the geo/checklist regime.

Method	$\mathcal{L}^+$	$\mathcal{L}^?$
BCE	$-\log(f_\theta^i)$	$-\log(1-f_\theta^i)$
BCE-AN	$\mathcal{L}^+_{\text{BCE}}$	$\mathcal{L}^-_{\text{BCE}}$
WAN	$\mathcal{L}^+_{\text{BCE}}$	$\gamma\mathcal{L}^-_{\text{BCE}}$
LS	$\frac{1-\epsilon}{2}\mathcal{L}^+_{\text{BCE}} + \frac{\epsilon}{2}\mathcal{L}^-_{\text{BCE}}$	$\frac{1-\epsilon}{2}\mathcal{L}^-_{\text{BCE}} + \frac{\epsilon}{2}\mathcal{L}^+_{\text{BCE}}$
ROLE	See [7]	See [7]
EM	$\mathcal{L}^+_{\text{BCE}}$	$-\alpha(f_\theta^i\mathcal{L}^+_{\text{BCE}} + (1-f_\theta^i)\mathcal{L}^-_{\text{BCE}})$
LL-R	$\mathcal{L}^+_{\text{BCE}}$	$\mathbb{1}_{[\neg\text{LL}]}\mathcal{L}^-_{\text{BCE}}$
LL-Ct	$\mathcal{L}^+_{\text{BCE}}$	$\mathbb{1}_{[\neg\text{LL}]}\mathcal{L}^-_{\text{BCE}} + \mathbb{1}_{[\text{LL}]}\mathcal{L}^+_{\text{BCE}}$
LL-Cp	$\mathcal{L}^+_{\text{BCE}}$	$\mathbb{1}_{[\neg\text{LL}]}\mathcal{L}^-_{\text{BCE}} + \mathbb{1}_{[\text{LL}]}\mathcal{L}^+_{\text{BCE}}$

Table A10: Positive and unknown losses for the SPML methods. For BCE row, $\mathcal{L}^?$ signifies $\mathcal{L}^-$ since BCE is trained on full labels. The variables γ, ϵ, α are all hyperparameters for each respective method. For the LL-variants, LL signifies whether the loss term falls in the top $((t-1)\cdot\Delta)\%$ of losses in the batch.

Method	ML48S	ML48S $+\mathcal{R}_P$	ML48S $+\mathcal{R}_E$
BCE	62.44 ± 0.51	(+3.93) 66.37 ± 0.48	(+0.59) 63.03 ± 0.42
BCE-AN	52.23 ± 0.48	(+3.83) 56.06 ± 1.11	(+0.12) 52.35 ± 0.54
WAN	51.96 ± 0.55	(+3.70) 55.66 ± 0.72	(-0.07) 51.89 ± 0.45
LS	**56.42 ± 0.67**	(+0.02) **56.44 ± 0.73**	(-0.08) **56.34 ± 0.51**
ROLE	54.00 ± 0.95	(+0.14) 54.14 ± 0.54	(-0.51) 53.49 ± 0.73
EM	55.27 ± 0.97	(-0.08) 55.19 ± 0.66	(+0.35) 55.62 ± 0.21
LL-R	50.06 ± 0.79	(+4.90) 54.96 ± 0.35	(+0.07) 50.13 ± 0.84
LL-Ct	47.98 ± 0.90	(+6.11) 54.09 ± 0.55	(+2.09) 50.07 ± 1.41
LL-Cp	43.80 ± 0.80	(+0.60) 44.40 ± 0.58	(+0.58) 44.38 ± 0.61

Table A11: Compiled mAP results (given in percentages) on the test set for each method, averaged across five runs for unmodified, probability regularized, and embedding regularized SPML methods.

A.5 Additional Experiments and Analysis

A.5.1 Embedding Asset Regularization

We extend the idea of asset-level similarity to the embedding level, by enforcing embeddings of a clip to be similar to the average embeddings across the entire asset, with a regularization term $\mathcal{L}_E$:

$$\mathcal{R}_E(x_j^i) = \text{MSE}(d_j^i, \overline{d_t^i}) \tag{1}$$

where MSE is mean-squared error and d_j^i is the last layer embedding of the network for the j-th clip of recording i. We use a similar equation to calculate the running average of the embedding as for the probability asset regularization but use a different ϵ_2.

Probability regularization is generally more effective than embedding regularization. In Table A11, we see the average performance boost for embedding regularization is much less than the boost for probability regularization. We attribute this to the recurrence of background species at the asset-level giving an accurate and more direct training signal. The supervision provided at the prediction-level is a strong prior because species positives in one clip are very likely to reoccur. In contrast, embedding regularization is more indirect than probability regularization, as the model can learn spurious correlations at the embedding level like fixed background noise within an asset. Regardless, we do find that embedding regularization still has a minor positive effect on training, potentially working as a weaker form of ℓ_2 weight decay to prevent overfitting on noisy target-only data.

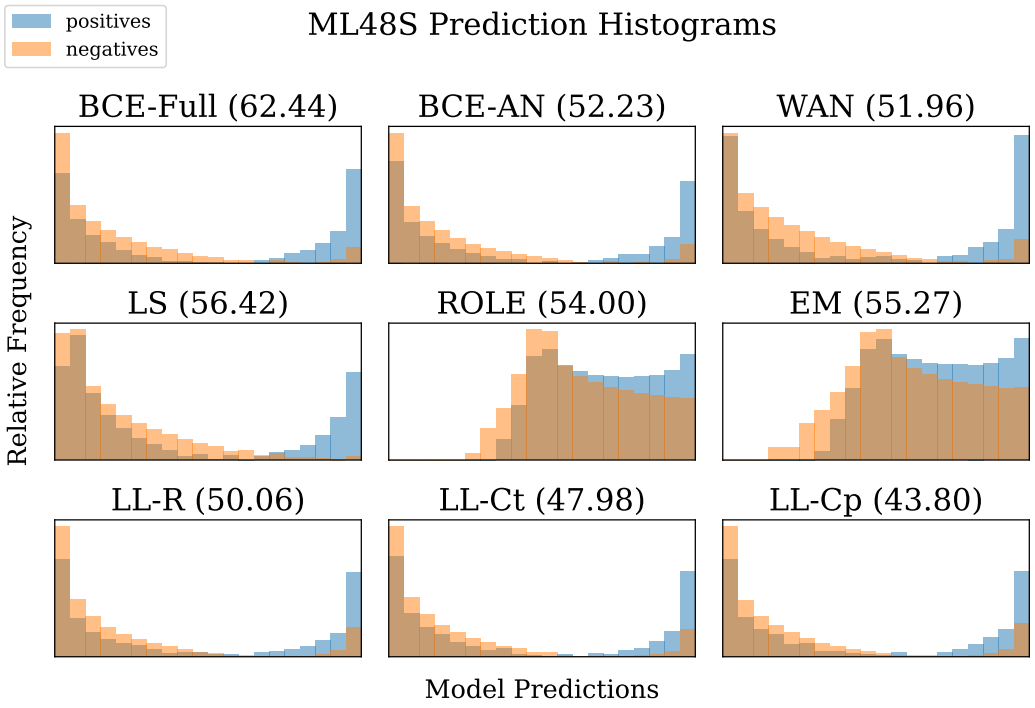

Figure A4: Histogram of model outputs, log-scaled, shown separately for positive and negative labels on the test set of ML48S. We see ROLE and EM have significantly different distributions from the other methods and the LL-variants all have higher rates of high confidence false negatives compared to the other methods.

A.5.2 Model Output Histograms

In Figure A4, we show the model prediction distributions for positive and negative labels on the ML48S test set. We see a higher rate of high confidence false positives for the LL methods and a significantly shifted probability distribution for ROLE and EM.

A.5.3 Analysis of Specific Species

In Figure A5, we include PR curves for the LL-R method in the three data regimes and with regularization for five different species. Interestingly, we see different patterns for the two groups of species. In the three plots on the left, we compare the PR curves for Carolina Chickadee, Black-capped Chickadee, and Mountain Chickadee, which all are geographically separated but are vocally similar. We see that providing negative labels through geographical priors gives a large increase to the model's precision, indicating the labels are helping with this fine-grained confusion. In contrast, in the two right plots we see the opposite effect. Yellow-bellied Sapsucker and Red-breasted Sapsucker are also geographically separated and nearly vocally identical, but we see providing the model with negative labels through geographical priors decreases performance. Our hypothesis for this distinction is the model is unable to learn the sapsucker task because it is more difficult than the chickadee task. In the chickadee task, the species are similar-sounding, but have known differences in vocal patterns. As a result, providing the model explicit negatives prevents LL-R from rejecting the losses for the similar chickadees and the model learns to distinguish the two. In contrast, the sapsucker task is more difficult than the chickadee task, as the two species do not have distinctive vocal differences. As a result, rejecting the loss in this case may prevent the model from being forced to learn an intractable task and instead allow it to accept ambiguity on these two species instead of having to predict confidently.

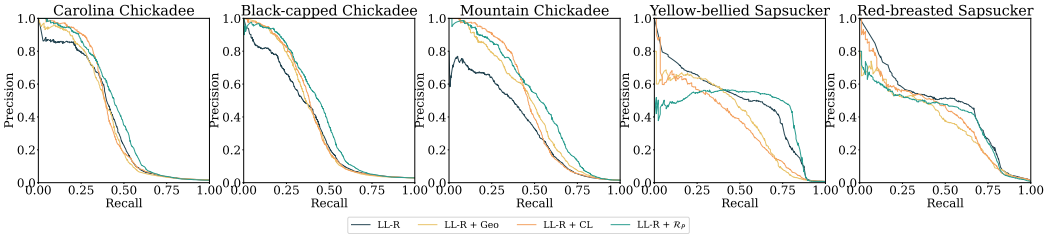

Figure A5: Precision-recall curves for the LL-R method in target-only, target-only with regularization, geo, and checklist regimes for five different species, where the species name is given as the graph title.