A APPENDIX

A.1 BIOACOUSTICS DATASETS

We use **Xeno-Canto** (XC; Vellinga & Planqué, 2015) as the source dataset for bird species classification in the audio domain. XC is a growing, user-contributed collection of Creative Commons recordings of wild birds across the world. Our snapshot, downloaded on July 18, 2022, contains around 675,000 files spanning 10,932 bird species. Recordings are *focal* (purposefully capturing an individual's vocalizations in natural conditions, as opposed to *passively* capturing all ambient sounds), and each is annotated with a single foreground label (for the recording's main subject) and optionally a varying number of background labels (for other species vocalizing in the background).

For our distributionally-shifted datasets, we use multiple collections of passive (also called *soundscape*) recordings from various geographical locations. We use a 75/25 % split to obtain $\mathbb{D}_t^{adapt}$—used to adapt the model—and $\mathbb{D}_t^{test}$—used to evaluate the adapted model.

- **Sapsucker Woods** (SSW; Kahl et al., 2022a) contains soundscape recordings from the Sapsucker Woods bird sanctuary in Ithaca, NY, USA.
- **Sierra Nevada** (Kahl et al., 2022b) contains soundscape recordings from the Sierra Nevada in California, USA.
- **Hawai'i** (Navine et al., 2022) contains soundscape recordings from Hawai'i, USA. Some species, particularly endangered honeycreepers, are endemic to Hawai'i and many are under-represented in the Xeno-Canto training set.
- **Powdermill** (Chronister et al. (2021)) contains high-activity dawn chorus recordings captured over four days in Pennsylvania, USA.
- **Caples** is an unreleased dataset collected by the California Academy of Science at the Caples Creek area in the central Californian Sierra Nevadas. Work is underway to open-source this dataset.
- **Colombia** is an unreleased dataset, previously used as part of the test set for the BirdCLEF 2019 competition.
- **High Sierras** is an unreleased dataset, previously used as part of the test set for the Kaggle Cornell Birdcall Identification challenge. Recordings are typically sparse, but with very low SNR due to wind noise. Work is underway to open-source this dataset.

A.2 XENO-CANTO DATA PROCESSING

Xeno-Canto recordings range from less than 1 second to several hours long. To extract 6-second segments we use a heuristic to identify segments with strong signal.

1. If the audio is shorter than 6 seconds, pad the recording evenly left and right using wrap-around padding.
2. Convert the audio into a log mel-spectrogram.
3. Denoise the spectrogram:
   (a) For each channel calculate the mean and standard deviation. All values that lie more than 1.5 standard deviations from the mean are considered outliers.
   (b) Calculate a robust mean and standard deviation using the inliers[3].
   (c) Any values that lie more than 0.75 robust standard deviations away from the robust mean are considered signal. Shift the signal in each channel by its robust mean, and set all noise to zero.
4. Sum all channels in the denoised spectrogram to create a signal vector.
5. Use SciPy's `find_peaks_cwt` function to retrieve peaks, using 10 Ricker wavelets with widths linearly spaced between 0.5 and 2 seconds

[3]In the calculation of the mean and variance we add 1 to the denominator to avoid division by zero in the case that all values are considered outliers.

Table 5: Summary of Bioacoustic Soundscape Dataset Characteristics.
'XC/Species' indicates the average number of Xeno-Canto training example files per species. 'Low Data Species' are species with fewer than 50 training examples available. Labels per example is computed on the peak-sliced data, while hours and number of labels refer to the original raw dataset.

	Hours	#Species	#Labels	Labels/ Example	Climate	XC/ Species	Low data
Sapsucker	285	96	50,760	1.5	Temperate	367.9	3%
Sierra Nevada	33	56	20,147	1.9	Temperate	416.0	0%
Hawai'i	51	27	59,583	1.5	Tropical	166.3	44%
Powdermill	6	48	16,052	3.2	Temperate	360.0	0%
Caples	6	78	4,993	1.4	Temperate	334.8	10%
Colombia	8	63	1,489	1.4	Tropical	215.8	6%
High Sierras	34	19	14,494	1.2	Alpine	323.5	5%

6. Select windows of 0.6 seconds centred at each peak and discard the peak if the maximum value in this window is smaller than 1.5 times the mean of the signal vector.
7. Keep only up to 5 peaks, with the highest corresponding values in the signal vector.
8. Select a 6 second window centred at each peak. If the window overlaps the start or beginning of the boundary, shift the window accordingly.

A.3 SOUNDSCAPES DATA PROCESSING

We extract 5-second segments from soundscapes recording by cross-referencing the bounding box labels with the same heuristic used to extract 6-second segments from XC recordings:

1. Use the procedure outlined in Appendix A.2 to extract 5-second (rather than 6-second) windows from up to 200 (rather than 5) high-signal peaks per source file.
2. For all 5-second windows:
   (a) If it does not overlap in time with any bounding box label, drop it.
   (b) Otherwise, find all overlapping bounding box labels and label the window with the union of all their labels.

A.4 METRICS

A.4.1 MAP

Define $\mathrm{Prec}_X(s, c)$ as the generalized inverse rank of a ground-truth positive label c in an observation s, computed as:

$$\mathrm{Prec}_X(s,c) = \frac{1}{\mathrm{Rank}_X(s,c)} \sum_{r=1}^{\mathrm{Rank}_X(s,c)} \mathbf{1}[\mathrm{Label}(s,c) \in \mathcal{C}(s)]$$

where X is the corpus over which we perform rankings, $\mathrm{Rank}_X\, s, c$ is the rank of the score for class c in observation s in the corpus X, $\mathrm{Label}(s, r)$ is the ground-truth label of class r in observation s, and $\mathcal{C}(s)$ is the set of ground-truth positive classes in observation s. Finally, $\mathbf{1}[\mathrm{Label}(s, r) \in \mathcal{C}(s)]$ is the indicator function for whether observation s is a ground-truth member of class c.

The mAP metric measures the per-example precision, averaged over the set $\mathcal{E}$ of all examples in the dataset:

$$\frac{1}{|\mathcal{E}|} \sum_{s \in \mathcal{E}} \frac{1}{|\mathcal{C}(s)|} \sum_{c \in \mathcal{C}(s)} \mathrm{Prec}_{\mathcal{E}}(s, c)$$

Table 6: Grid used for tuning hyper-parameters.

Method	Hyperparameter	Grid
All	Learning rate	{ 1e-5, 1e-4, 1e-3 }
	Trainable parameters	{ BatchNorm scale and bias, all }
	Use of dropout	{True, False }
	Use source BN statistics	{True, False }
	Learning rate cosine decay	{True, False }
SHOT (Liang et al., 2020)	Pseudo-labels weight β	{0., 0.3, 0.6, 0.9}
Pseudo-labelling Lee et al. (2013)	Confidence threshold	{0., 0.5, 0.9, 0.95}
Dropout Student	Softness weight α	{0.1, 1.0}
	Pseudo-label update frequency	{Every iteration, Every epoch}
NOTELA	k nearest neighbors	{5, 10}
	Softness weight α	{0.1, 1.0}
	Pseudo-label update frequency	{Every iteration, Every epoch}

A.4.2 CMAP

Class-wise mean average precision (cmAP) is defined as:

$$\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{|c\in\mathcal{E}|}\sum_{s\in\mathcal{E}}\mathrm{Prec}_{\mathcal{C}}(s,c)$$

Here $|c\in\mathcal{E}|$ denotes the total number of ground-truth positive examples of class c in the dataset. Notice that in this case, the precision ranking is over the logits for each target class, instead of ranking the logits in a single observation.

A.5 HYPERPARAMETER VALIDATION

As mentioned in section 5, we reproduce all methods and ensure fairness of comparisons by (i) using the same pre-trained models and (ii) re-tuning each method's hyperparameters. All experiments are carried out with a batch size set to 64 (both audio and vision). Table 6 displays the grids used for hyper-parameter tuning (also both for audio and vision tasks). Note that to reduce the load of hyperparameters to tune in NOTELA, we make the design choice of using $\lambda=\alpha$, which effectively removes one degree of freedom.

A.6 PROOF OF EQUATION 3

We hereby provide the proof, as well as a more formal justification for the updates given in Equation 3. We start by restating the objective we want to minize:

$$\begin{aligned}\min_{\mathbf{y}_{1:N}}\quad & \mathrm{Tr}\left(-\frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top\log(\mathbf{p}_i)+\frac{\alpha}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top\log(\mathbf{y}_i)-\frac{\lambda}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}\,\mathbf{y}_i^\top\mathbf{y}_j\right)\\ \text{s.t}\quad & \mathbf{1}^\top\mathbf{y}_i=1,\ \mathbf{y}_i\geq 0\,.\end{aligned}\tag{4}$$

General case. Let us consider an easier problem, in which the Laplacian term has been linearized around the current solution $\mathbf{y}_i=\mathbf{p}_i$,

$$\begin{aligned}\min_{\mathbf{y}_{1:N}}\quad & \mathrm{Tr}\left(-\frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top\log(\mathbf{p}_i)+\frac{\alpha}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top\log(\mathbf{y}_i)-\frac{\lambda}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}\,\mathbf{y}_i^\top\mathbf{p}_j\right)\\ \text{s.t}\quad & \mathbf{1}^\top\mathbf{y}_i=1\,.\end{aligned}\tag{5}$$

Table 7: Top-1 accuracy, averaged over 5 random seeds, along with 95% confidence interval for each corruption in CIFAR-10-C.

Corruption	Baseline	AdaBN	Pseudo-Labelling	SHOT	TENT	NOTELA
brightness	91.28 ± 0.0	92.2 ± 0.03	92.55 ± 0.19	92.67 ± 0.06	92.58 ± 0.12	**92.96** ± 0.14
contrast	52.84 ± 0.0	87.3 ± 0.02	89.73 ± 1.26	89.22 ± 0.08	89.34 ± 0.35	**90.23** ± 0.3
defocus	53.16 ± 0.0	87.4 ± 0.09	**89.41** ± 0.21	88.42 ± 0.08	89.17 ± 0.09	89.38 ± 0.23
elastic	73.28 ± 0.0	77.02 ± 0.1	**80.08** ± 0.43	78.97 ± 0.08	79.74 ± 0.21	79.98 ± 0.19
fog	74.08 ± 0.0	85.92 ± 0.04	**88.37** ± 0.51	87.34 ± 0.16	87.95 ± 0.16	88.17 ± 0.18
frost	59.6 ± 0.0	82.45 ± 0.05	85.18 ± 0.6	84.62 ± 0.17	85.28 ± 0.2	**85.4** ± 0.22
frosted	46.36 ± 0.0	65.38 ± 0.03	71.29 ± 0.81	70.75 ± 0.12	**71.77** ± 0.18	71.57 ± 0.32
gaussian	28.16 ± 0.0	73.31 ± 0.03	**78.72** ± 0.53	76.51 ± 0.12	77.13 ± 0.12	77.35 ± 0.05
impulse	28.04 ± 0.0	64.36 ± 0.05	**70.9** ± 0.93	68.86 ± 0.14	70.05 ± 0.27	70.18 ± 0.17
jpeg	69.12 ± 0.0	72.57 ± 0.06	**80.37** ± 0.21	75.86 ± 0.15	77.26 ± 0.41	77.71 ± 0.1
motion	65.16 ± 0.0	86.29 ± 0.05	88.1 ± 0.54	87.94 ± 0.22	88.38 ± 0.15	**88.56** ± 0.14
pixelate	41.84 ± 0.0	80.7 ± 0.04	**85.77** ± 0.45	83.62 ± 0.28	85.14 ± 0.36	85.5 ± 0.2
shot	34.52 ± 0.0	75.68 ± 0.07	**81.24** ± 0.59	79.7 ± 0.12	80.8 ± 0.05	81.07 ± 0.13
snow	75.12 ± 0.0	82.88 ± 0.03	86.02 ± 0.88	85.14 ± 0.14	86.42 ± 0.2	**86.66** ± 0.18
zoom	57.36 ± 0.0	88.02 ± 0.05	90.09 ± 0.46	89.67 ± 0.19	90.15 ± 0.22	**90.38** ± 0.13

We purposefully omitted the $\mathbf{y}_i \geq 0$ constraint, which will be satisfied later on by our solution to Equation 4. The Lagrangian of this problem is

$$\mathcal{L}(\mathbf{y}_{1:N}) = \mathrm{Tr}\left(-\frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top \log(\mathbf{p}_i) + \frac{\alpha}{N}\sum_{i=1}^{N}\mathbf{y}_i^\top \log(\mathbf{y}_i) - \frac{\lambda}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\, \mathbf{y}_i^\top \mathbf{p}_j\right) \tag{6}$$

$$+ \frac{1}{N}\sum_{i=1}^{N}\gamma_i(\mathbf{1}^\top \mathbf{y_i} - \mathbf{1})\,, \tag{7}$$

and the gradient of this Lagrangian with respect to $\mathbf{y}_i$ is

$$N.\nabla_{\mathbf{y}_i}\mathcal{L} = -\log(\mathbf{p}_i) + \alpha(\log(\mathbf{y}_i) + \mathbf{1}) - \lambda\sum_{j=1}^{N} w_{ij}\mathbf{p}_j + \gamma_i\mathbf{1}\,. \tag{8}$$

Solving for $\mathbf{y}_i$ yields

$$\mathbf{y}_i = \exp\left(-\frac{\alpha+\gamma_i}{\alpha}\right)\exp\left(\frac{\lambda}{\alpha}\sum_{j=1}^{N} w_{ij}\mathbf{p}_j\right) \odot \mathbf{p}_i^{1/\alpha}\,. \tag{9}$$

Now γ_i is chosen such that the constraint $\mathbf{y}_i^\top \mathbf{1} = 1$ is satisfied, resulting in

$$\mathbf{y}_i \propto \mathbf{p}_i^{1/\alpha} \odot \exp\left(\frac{\lambda}{\alpha}\sum_{j=1}^{N} w_{ij}\mathbf{p}_j\right)\,. \tag{10}$$

Concavity. Let $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{N\times N}$ be the matrix of affinity weights. An additional assumption on $\mathbf{W}$ allows a more formal justification of the linearization of the Laplacian term. Specifically, we can justify that if $\mathbf{W} + \mathbf{W}^\top$ is positive semi-definite, then the last term in Equation 4 is concave.

To show this rewrite $\mathrm{Tr}\left(\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\, \mathbf{y}_i^\top \mathbf{y}_j\right)$ as $\mathbf{1}^\top(\boldsymbol{W} \odot \boldsymbol{Y}^\top\boldsymbol{Y})\mathbf{1}$ where $\boldsymbol{Y} = (\mathrm{vect}(\mathbf{y}_i))$, which is in $\mathbb{R}^{N\times C}$ or $\mathbb{R}^{N\times 2C}$ for the single- and multi-label case respectively. The Hessian of this function is $\boldsymbol{W}\otimes\mathbb{I} + \boldsymbol{W}^\top\otimes\mathbb{I}$, whose eigenvalues are multiplicities of those of $\boldsymbol{W} + \boldsymbol{W}^\top$.

Hence equation 4 can be solved by a concave-convex procedure (CCP; Yuille & Rangarajan, 2003; Ziko et al., 2018; Boudiaf et al., 2022), which is suited to cases in which one part of the objective is convex (the two first terms in our case) and the other is concave (the Laplacian term). CCP proceeds by minimizing a sequence of *pseudo-bounds*, i.e., an upper bound that is tight at the current solution, obtained by linearizing the concave part of the objective at the current solution. Therefore, our

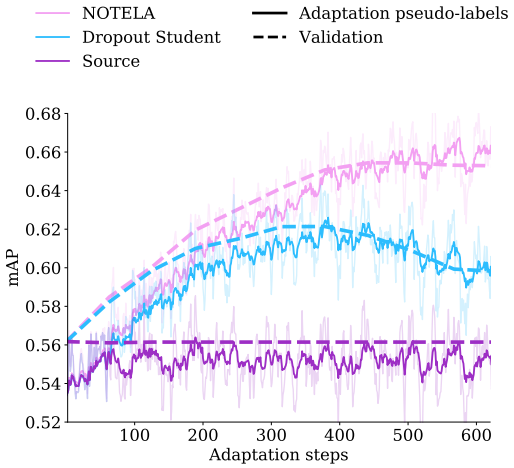

Figure 3: NOTELA improves pseudo-label quality over the Source model's predictions and Dropout Student approach, as evidenced by the evolution of mAP (computed using pseudo-labels and ground-truth annotations) over adaptation steps (solid curves) on the Sierra-Nevada dataset (Kahl et al., 2022b). This directly translates into higher mAP on the test set (dashed curves).

proposed updates from Equation 3 can be interpreted as the first iteration of this procedure. Starting from the initial solution, $\mathbf{y}_i^{(0)} = \mathbf{p}_i$, unrolling the CCP procedure would consist of performing the following updates until convergence:

$$\mathbf{y}_i^{(t)} \propto \mathbf{p}_i^{1/\alpha} \odot \exp\left(\frac{\lambda}{\alpha} \sum_{j=1}^{N} w_{ij} \mathbf{y}_j^{(t-1)}\right) . \tag{11}$$

Affinity weights. Let $\boldsymbol{A}$ be the adjacency matrix of the mutual k-nearest neighbours graph, and $\boldsymbol{D}$ the diagonal matrix with the node degrees (i.e., $(\boldsymbol{D})_{ii}$ is the number of mututal neighbours for sample i). We know that $\boldsymbol{A} + \boldsymbol{D}$ is positive semi-definite (Desai & Rao, 1994) and hence matrix $\boldsymbol{W} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{A} + \boldsymbol{D})\boldsymbol{D}^{-\frac{1}{2}}$ is also positive semi-definite. Note that this is equivalent to the case where w_{ij} is set to the reciprocal of the number of mutual neighbours, and $w_{ii} = 1$.

The terms w_{ii} act as an L2 regularizer on the values $\mathbf{y}_i$. Note that in our experiments we deviate from the theory and set $w_{ii} = 0$ to have better control over the regularization of $\mathbf{y}_i$.

A.7 Additional results

CIFAR-10-C per-corruption results. We provide the per-corruption results on CIFAR-10-C in Table 7, as well as the 95% confidence intervals.

Comparison of pseudo-labels. We provide in Figure 3 additional evidence that our NOTELA(LD) yields more accurate and stable pseudo-labels on the adaptation set $\mathbb{D}_t^{\text{adapt}}$ than Dropout Student (DS), which directly translates into better validation performances.

Table 8: NOTELA still works with lower batch size.

Batch size	mAP	cmAP
8	72.8	50.9
16	76.4	52.9
32	75.1	50.9
64	75.0	50.9

Sensitivity to batch size. Our results presented in the main paper used a batch size of 64 by default. Considering the relevance of the SFDA setting for *resource-constrained* practitioners, we study the sensibility of our method to changes in batch size. Our rational for studying this particular axis is that GPU memory can be an important bottleneck for individual practitioners and non-specialized research labs. We present results in Table 8, and show that NOTELA does not suffer

from using smaller batch sizes. As a matter of fact, it even seems that reducing batch size down to 16 provides a noticeable boost in performances. We hypothesize that using smaller batch size implicitly injects additional noise during the student step, which may help up to a certain extent.