

Comparing Multi-Task Strategies for Multiple Dataset Learning in Bioacoustics

Céline Angonin¹[0000–0002–3073–4086], Grzegorz Chrupała¹[0000–0001–9498–6912],
and Dan Stowell^{2,3}[0000–0001–8068–3769]

¹ Department of Intelligent Systems, Tilburg University, Tilburg, The Netherlands
`c.angonin@tilburguniversity.edu`

² Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

³ Evolutionary Ecology, Naturalis Biodiversity Center, Leiden, The Netherlands

Abstract. Multi-task learning describes a training paradigm in which models learn several tasks together, with the underlying assumption that the knowledge of one task can be leveraged to better learn other tasks and produce more robust, task-independent features. This approach is well-suited in bioacoustics, where diverse datasets are currently publicly available but generally used in isolation. In this work, we explore multi-task learning as a means to leverage these datasets simultaneously and train more generalisable models. We compare three multi-task learning approaches to single-task models for the species, individual, and vocalisation type classification tasks. Then, we evaluate the generalisation of all the resulting models on the BEncmark of ANimal Sounds (BEANS). Our results show that single-task models generally outperform multi-task models, with the exception of one multi-task model which generalises better on the BEANS benchmark than one single-task model. This indicates that while multi-task learning may surpass single-task learning in some cases, the latter displays better results overall. We also show that the success of multi-task training depends on the relative task weightings, and find limited benefit from automatic task weighting (GradNorm).

Keywords: Multi-task learning · Bioacoustics · Classification · Deep learning

1 Introduction

Many diverse bioacoustics datasets are publicly available online but only few of them are exploited. These datasets cover a wide range of taxa such as primates, anurans, birds, or insects and can vary from a few minutes to hundreds of hours of data. The annotated tasks are also varied, including detection or classification of species, individuals, and repertoire, or syllable classification. However, most of these datasets are recorded for a specific study, and are not used thereafter [1]. Moreover, bioacoustics models are trained using only a few datasets [2]. For instance, a typical training method consists in using an audio encoder pre-trained on one large repository containing either generic audio like AudioSet [3]

or bioacoustics data such as Xeno-Canto, and finetune on the dataset of interest. Several studies evaluating audio encoders bioacoustics have compared training paradigms, model architectures, and pre-training datasets on several bioacoustics benchmarks [2, 4, 5]. However, these studies do not reach a consensus on how to best use bioacoustics datasets to train models most effectively.

Multi-task learning (MTL) is a promising paradigm to combine these datasets to train generalisable models. MTL [6] is a machine learning approach in which a model is trained on several tasks simultaneously, with the underlying assumption that tasks share features that the model will be able to extract, leading to more robust model parameters and a better generalisation. MTL acts then as regularisation technique because the model is pushed to concentrate on features that are common to all tasks. Because of this regularisation ability, MTL can also reduce the computational resources needed to train a model and the amount of labelled data [7]. The latter point is especially useful for some bioacoustics tasks like individual identification, where labelled data are scarce [5]. MTL has already shown successful results in other domains, such as natural language processing [7] or medical image analysis [8]. A few papers employ MTL in bioacoustics for various tasks. Rookognise [9] is a multi-task network developed to detect the vocalisations of individual rooks in a captive colony with custom-made tasks derived from the data and existing annotations. MTL has also been leveraged to predict different taxonomy levels [10], where each task corresponds to one level and finer taxonomy levels are predicted at deeper layers in the network. Somervuo [11] built a two-head convolutional neural network to simultaneously classify species and estimate their distance from the microphone. Finally, the authors of the bioacoustics encoder Perch 2.0 [12] also used multi-task during its supervised training by predicting both the species and the source recording of the audio. However, these papers all derive several tasks from a single dataset, thus do not investigate how to train networks productively from the many diverse bioacoustics datasets that are available. Combining multiple datasets raises issues such as how to ensure well-balanced learning from the different evidence sources.

In this paper, we assess the potential of multiple dataset learning using MTL for three common bioacoustics classification tasks, namely species, individual, and vocalisation type (vox type) identification. More precisely, we make the following contributions:

- We compare single-task to multi-task learning on five datasets spanning these three bioacoustics tasks.
- We compare three strategies to set up the weights of each task for the multi-task models during the loss calculation: equal weights, different static weights, and dynamic weights adjustments using the GradNorm algorithm.
- We evaluate the resulting single and multi-task models on diverse taxa and tasks using the BEncmark of ANimal Sounds (BEANS)[13], to estimate whether MTL helps improving the generalisation abilities of the models.

The structure of the paper is as follows. We begin with presenting the datasets we use as well as our single and multi-task architectures. Then, we carry out

both an in-domain evaluation using the training datasets and a generalisation evaluation using the BEANS benchmark. Finally, we present some insights about MTL for bioacoustics. Our code will be released upon acceptance.

2 Methods

2.1 Datasets

We chose five datasets covering our three tasks of interest, all publicly available. Table 1 presents the number of annotations and classes of each dataset included in this work. Four of them label individual birds. The Wytham dataset [14] originally contains more than one million annotations of great tits spanning a period of several years in the United Kingdom. We took a subset of this dataset in this study and included the vocalisations of the fifteen individuals that were the most recorded during the same year. The three other datasets [15] label individual little owls, tree pipits and chiffchaffs, and were collected in the Czech Republic and Hungary for a common study. The train and test sets were already provided, and we randomly selected samples from the training set to create the validation set.

The last dataset [16] recorded dawn choruses of birds in three American states. Each audio file is five minutes long and annotated in two second segments with all the occurring species and vox types. Species or vox types present in only one file or two files were put in the train set and both the train and test set respectively. We also discarded classes that contained less than 10 samples. This dataset is highly unbalanced, both in terms of the number of annotations per species, and the number of annotations per vox type per species.

Table 1. Datasets information

	Task	# Classes	# Items (train / val / test)	Multi-label
Great tit	Ind. Id	15	5051 / 1017 / 1852	×
Chiffchaff	Ind. Id	23	4763 / 668 / 1331	×
Tree pipit	Ind. Id	10	362 / 47 / 303	×
Little owl	Ind. Id	16	484 / 61 / 407	×
Dawn chorus	Species	49	13111 / 2805 / 5726	✓
	Vox type	75		

2.2 Multi-task training

Multi-task training contains key differences compared to single-task training. The construction of the loss function is one of them, as the loss must consider all

the tasks. A very common method to build the loss function is to sum the loss of each task with a weighting parameter [17]. The loss function of the multi-task model can then be written as

$$L_{tot} = \sum_{i=1}^N w_i L_i \quad (1)$$

with L_i and w_i representing the loss function and the weight of task i respectively, and N the number of tasks.

We explored three different methods of setting the weights w_i in Equation 1. We first implemented the most naive approach, which summed the losses of the three tasks, leading to an equal importance of each task. This is equivalent to setting each weight to 1. The second method consisted in setting them manually at the beginning of the training. These weights are then fixed for the whole training. The optimal value for each weight can be found either through a grid search or by intuition. However, this is a time-consuming process which becomes increasingly cumbersome as we expand the number of tasks. In addition, these weights may need to be updated during the training as the tasks may not train at the same speed. Therefore, numerous multi-task optimisation approaches have been developed to dynamically tune these weights. The last method we explored is the GradNorm [18] algorithm, which automatically updates the magnitude of the loss weights according to the learning rate of each task while training. This algorithm is therefore suitable for handling cases where some tasks dominate the loss calculation. The weights for all the tasks were initially set to 1 and a single imbalance hyperparameter α controlled the magnitude of the change allowed for each update. The higher α , the more these weights can change for each iteration. The value of this parameter ranges from 0.1 for balanced tasks to 3 for very unbalanced tasks. In the rest of the paper, these models will be referred to as Equal weights, Static weights, and GradNorm models respectively.

The construction of batches also differs from single-task learning. One can train tasks in parallel, sequentially, or with a combination of both. In this study, we sampled from all the tasks for each batch randomly. Since our models predict labels for all the tasks regardless of the input labels, we masked the irrelevant outputs during the loss calculation.

2.3 Architecture and training details

The overview of our model is illustrated in Fig 1 a). We first selected a robust pre-trained encoder to get a representation of the input data. We chose the popular general-purpose encoder Audio Spectrogram Transformer (AST) [19], that we kept frozen for all of our experiments. As a baseline, we used this frozen encoder followed by a linear probe, which will be referred to as "AST-frozen" in the rest of the paper. We resampled our data to 16 kHz and padded it to 10-second long segments to comply with the sample rate and input length required by the AST. The audio was then converted to a spectrogram that the AST divided into patches and encoded, leading to a two-dimensional embedding output. We

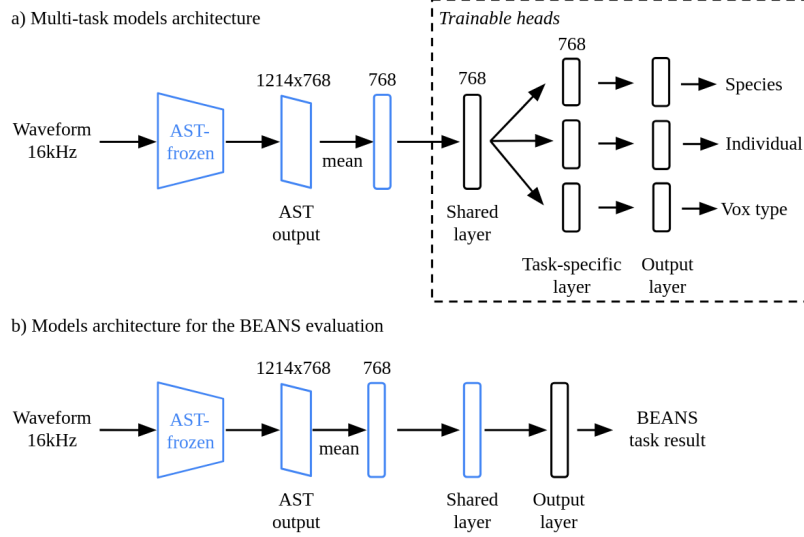


Fig. 1. Architecture of the multi-task models. Single-task models only contain the head corresponding to their task. The blue layers indicate trained and frozen layers.

averaged this embedding along the spatial axis to get an audio representation for the whole spectrogram before propagating it to a small fully trainable feed-forward network. The latter includes two hidden layers, both followed by a ReLU activation function. The first layer is shared between all tasks to learn common features, while the second one is task-specific, in order to learn task-specific features. We kept our architecture simple on purpose so that the differences in our results would arise from the difference in training paradigms.

We trained nine models in total: the baseline for the three tasks, the three single-task models and three multi-task models. We used the Optuna framework for the hyperparameter optimisation to select an appropriate batch size, dropout rate, and learning rate for each model. The resulting hyperparameters after 30 trials are presented in Table 4 in the Appendix. Two multi-task models had specific hyperparameters that we tuned together with the other hyperparameters shown in Table 4. The first model was the static weight model, for which each task weight was specified before the training. We constrained each weight to take a value between 0 and 1, and scaled the resulting weights so that their sum amounted to 3, in order to have the same scaling as the other multi-task models. The final weights were 0.327, 2.32, and 0.330 for the individual, species, and vox type classification respectively. The second model was the GradNorm model, which included two hyperparameters related to the GradNorm algorithm. One of the hyperparameters controlled the number of batches in-between each task weight update. Since the application of this algorithm can be computationally intensive, we set this hyperparameter to 10 in order to frequently update without

slowing down the training excessively. The other hyperparameter α was included in the hyperparameter search with values ranging from 0.1 to 3. This resulted in a final value of 2.3, indicating an asymmetry between the tasks.

The loss function and evaluation metrics we chose differed according to the task. Since the dawn chorus dataset contains multi-label annotations whereas the other datasets labelled one class for each sample, we chose the binary cross-entropy loss function for the species and the vox type tasks, and the cross-entropy loss function for the individual task. For the same reason, we evaluated the individual task with the weighted F1 score, and with the weighted mean average precision (mAP) metric for the two remaining tasks.

2.4 Generalisation evaluation with BEANS

The BEANS benchmark allows assessing the generalisation of bioacoustics models by evaluating them on a varied range of datasets. It comprises ten bioacoustics datasets with five detection and five classification datasets, as well as two more general audio classification datasets, namely speech-commands (sc) and ESC-50 (esc). Therefore, BEANS uses two evaluation metrics, the accuracy for the classification datasets, and mAP for the detection datasets. The bioacoustics datasets span many taxa (birds, marine mammals, primates, insects...) and three bioacoustics tasks, the same as in this work. All the datasets come with predetermined training, validation, and test splits.

To evaluate the single and multi-task models on BEANS, we discarded their task-specific and output layers, froze the models, and trained a linear probe on top of the shared layer. This allowed to observe whether joint training on multiple tasks resulted in better features in the shared part of the model. We applied the same hyperparameters as during the training. The baseline also consisted of the frozen AST followed by a linear probe. This evaluation procedure is shown on Fig 1 b).

3 Results

3.1 In-domain evaluation

Table 2 shows the scores of all models for the three tasks. The baseline delivered the best performance. Its training was also the most stable, as can be seen by its lower standard deviation. Single-task models closely followed, with almost the same score for the individual task, but a larger decrease for the two other tasks. Finally, all the multi-task models showed substantially lower scores than all other models for the species and vox type tasks. While the static weights and GradNorm models performed closely, with the static weight model displaying a slightly increase in scores for the species and vox type tasks, the equal weights model performed the worst.

The choice of the multi-task strategy mattered, which can be seen in the variation of scores of the multi-task models. The equal weights model training

Table 2. Weighted averages of F1 scores and mean Average Precision (mAP) for each task. The mean score and standard deviation are reported for three seeds. Task weights at the end of the training are written in parentheses.

	Individual F1 score	Species mAP	Vox type mAP
AST-frozen	0.643 $\pm$0.004	0.496 $\pm$0.002	0.444 $\pm$0.0009
Single-task	0.621 $\pm$ 0.011	0.452 $\pm$ 0.009	0.389 $\pm$ 0.009
Equal weights	0.604 $\pm$ 0.010 (1)	0.278 $\pm$ 0.013 (1)	0.240 $\pm$ 0.011 (1)
Static weights	0.596 $\pm$ 0.013 (0.327)	0.355 $\pm$ 0.006 (2.32)	0.309 $\pm$ 0.003 (0.329)
GradNorm	0.602 $\pm$ 0.012 (0.08)	0.331 $\pm$ 0.006 (1.48)	0.290 $\pm$ 0.011 (1.44)

Table 3. Results of all models on the BEANS benchmark for one seed. The average score on the classification and detection datasets has been computed, as well as the median rank of each model.

	Mean (classif.)	Mean (detection)	Median Rank
Ast-frozen	0.734	0.348	1
Single-task individual	0.450	0.199	5
Single-task species	0.527	0.260	2.75
Single-task vox type	0.534	0.260	2.25
Equal weights	0.394	0.156	7
Static weights	0.463	0.218	4
GradNorm	0.419	0.178	6

seemed to be dominated by the individual task, as is indicated by its lower scores for the species and vox type tasks. This assumption is further reinforced by the fact that although both the GradNorm and the static weights models attributed a lower weight for the individual task, they maintained the same performance, and improved their scores for the two remaining tasks. We can also observe that while the GradNorm and static weights models resulted in different end weights, their scores were relatively similar, implying that different set of weights may be possible. In a context of similar output, GradNorm has an advantage over the static weights model. It includes only two hyperparameters to tune for any number of tasks, while the hyperparameter search space grows with each added task for the static weight model.

3.2 Generalisation evaluation

Table 3 presents the results of the single and multi-task models together with the baseline on the BEANS benchmark. In order to allow for an easier model comparison, we calculated their average scores on the classification and detection datasets, as well as their median rank across all the datasets. The per-dataset results are displayed in Table 5 in the Appendix. Overall, these generalisation results are in line with the in-domain evaluation shown in Table 2. The baseline outperformed all the other models, and the single-task models performed also generally better than the multi-task models.

However, the performance of some models was not the same as the previous evaluation. First, the difference in scores between the baseline and the other models was more pronounced, especially for the classification datasets. Two reasons may explain this difference. On one hand, the single and multi-task models may have tailored their features towards the training datasets, losing some generalisation abilities compared to the pre-trained baseline. On the other hand, the per-dataset scores displayed in Table 5 showed that this score difference is partly driven by the speech-commands dataset for the classification datasets. This may come from the pre-training data of AST, AudioSet, where almost half of the samples were tagged with human speech. Second, the static weights models got better scores than the single-task individual model, indicating that multi-task learning may generalise better than single-task learning in some cases.

4 Discussion

Our results for both the in-domain and generalisation evaluations show that a wide variation in performance can arise from the details of the MTL configurations. Contrary to what could be expected, more sophisticated methods such as GradNorm do not necessarily give the best results. It also adds overhead in terms of computations and memory usage since the gradients for each task must be calculated and the weights of the loss function updated for each iteration. The best multi-task approach in our study is setting fixed weights at the beginning of the training. These two results are in line with the observations made in [20], in which the authors compared several multi-task optimisation methods, including GradNorm, to scale loss weights for multi-task models. They conclude that these methods do not give better results than fixing arbitrary weights.

In our study, the majority of the single-task models outperformed the multi-task models, with the exception of the static weights models on the BEANS benchmark which outperformed the single-task individual model. Multiple reasons could explain the lower performance of the multi-task models. For instance, methodological limitations of our experiments may have prevented taking full advantage of the multi-task learning possibilities. One such limitation consists in setting the same hyperparameters for all tasks, which is necessary with our architecture. This could explain why the species and vox type tasks results are lower compared to the individual task for our multi-task models. Furthermore, we also learnt all the tasks together during the whole training, while one could

imagine another approach where easy tasks or dominating tasks would be trained first. An interesting method alleviating both limitations mentioned above can be found in [21]. The authors designed a dynamic stop-and-go multi-task training procedure where each batch originated from one task only. When the model converges for one task, the authors stop training this task until its validation loss raises. This allowed them to set different training parameters and have a finer grain control over the training of each task. Future work could assess the value of such methods for these three tasks.

However, the most probable factor explaining the lower scores of the multi-task models could be that the tasks don't share enough common information that can be leveraged. On one side, the multiple datasets we chose for the three tasks may be too different, so do not contain enough common information for the multi-task models to learn from. On the other side, the tasks themselves may not benefit to learning from each other. Task relatedness is an important concept in MTL, and a whole line of research focuses on exploring tasks interaction and selecting relevant tasks [22–24]. In order to test whether our chosen tasks induce a positive transfer, we trained a multi-task model on the two tasks that originated from the same dataset, i.e. the individual and vox type classification. Following [17], we trained the same architecture with various complementary weights for the two tasks summing up to 1, and plotted the resulting performance of all models on both tasks. The result is shown on Figure 2. Each most left and right points of this graph represents the single-task case, with the weight of one of the two tasks equal to 0, and the other to 1. The blue line represents the evolution of the mAP score of the species task when its weight is increasing, while the orange line corresponds to the evolution of the mAP score of the vox type task when its weight is decreasing. If tasks would benefit from each other, we would observe bell-shaped curves for both tasks: multi-task cases would showcase better scores for than the single-task cases. Here, the graph is flat when both tasks weights are greater than 0, indicating that the tasks do not influence each other. Therefore, no positive transfer occurs. This may be due to the hierarchical nature of these tasks, where the knowledge of the vox type implies the knowledge of the species. For these two specific tasks, a hierarchical model may have been more appropriate, such as TaxoNet [10].

As observed with this work, using MTL together with multiple datasets on these three tasks generally doesn't yield better results than the single-task models. Only the static weights model outperformed the single-task individual model on the generalisation evaluation. However, MTL may benefit the pre-training stage more than the fine-tuning stage, which would make a better use of its generalisation potential. Thus, investigating the use of MTL for pretraining general bioacoustics encoder models is a promising direction for future research. Recently, one of these encoders, namely Perch 2.0 [12], was released, and used MTL in its pre-training phase by creating two objectives, one main supervised objective and one auxiliary self-supervised objective. The first supervised objective was species classification, while the self-supervised objective consisted in predicting the source recording of the incoming spectrograms. MTL could also

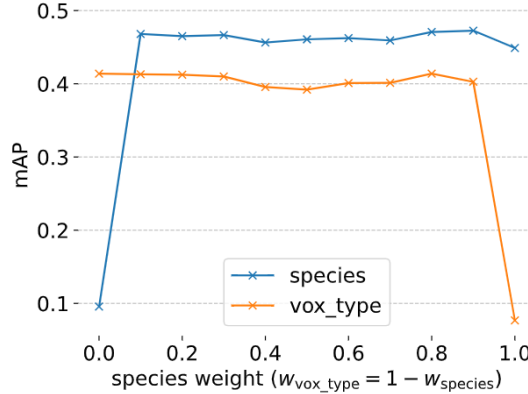


Fig. 2. mAP of multi-task models trained with various weights combinations on the species and vox type tasks. The tasks weights were constrained to sum up to 1. The middle of the graph corresponds to the performance of the multi-task model with weights for both tasks set to 0.5. Similarly, the left and right points of the graph correspond to the case where one of the tasks weight is set to 0, which is equivalent to a single-task model.

be used with unlabelled data. [5] explored different techniques to train robust bioacoustics encoders, and they determined that the best paradigm consisted in self-supervised pretraining followed by supervised fine-tuning on a mix of bioacoustics and general audio data. Multi-task training could then be easily integrated during the pre-training stage by using diverse datasets and create different tasks to predict from the data. Although not directly related to multi-task, another interesting topic would consist of exploring how to optimise multi-task training for a task whose dataset is constructed from several pre-existing datasets, as it is the case for the individual task in this study. This is especially relevant in bioacoustics, where many existing datasets cover the same task.

5 Conclusions

We compared three multi-task strategies against single-task training for species, vox type, and individual identification classification. We gathered five datasets covering these three tasks and evaluated our models both on these datasets and on the BEANS benchmark to estimate their generalisation abilities. We observed that single-task models generally outperformed multi-task models for both evaluations, with the exception of the static weights model which generalised better than the single-task individual model on the BEANS benchmark. Future work may investigate MTL for pre-training rather than fine-tuning. The use of many bioacoustics datasets in structured fashion, rather than generic pretraining, may then lead to better and more controlled generalisation capabilities.

Acknowledgments. Céline Angonin is funded by the Bioacoustic AI Doctoral Network European Union’s Marie Skłodowska-Curie Action under grant agreement No 101116715.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Rauch, L., Schwinger, R., Wirth, M., Heinrich, R., Lange, J., Kahl, S., Sick, B., Tomforde, S., Scholz, C.: Birdset: A multi-task benchmark for classification in computational avian bioacoustics (Apr 2024). <https://doi.org/10.48550/arXiv.2403.10380>
2. Schwinger, R., Zadeh, P.V., Rauch, L., Kurz, M., Hauschild, T., Lapp, S., Tomforde, S.: Foundation models for bioacoustics – a comparative review (Aug 2025). <https://doi.org/10.48550/arXiv.2508.01277>
3. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 776–780. IEEE, New Orleans, LA (Mar 2017). <https://doi.org/10.1109/ICASSP.2017.7952261>
4. Ghani, B., Kalkman, V.J., Planqué, B., Vellinga, W.P., Gill, L., Stowell, D.: Impact of transfer learning methods and dataset characteristics on generalization in birdsong classification. *Scientific Reports* **15**(1), 1–17 (May 2025). <https://doi.org/10.1038/s41598-025-00996-2>
5. Miron, M., Robinson, D., Alizadeh, M., Gilsenan-McMahon, E., Narula, G., Chemla, E., Cusimano, M., Effenberger, F., Hagiwara, M., Hoffman, B., Keen, S., Kim, D., Lawton, J., Liu, J.Y., Raskin, A., Pietquin, O., Geist, M.: What matters for bioacoustic encoding (Aug 2025). <https://doi.org/10.48550/arXiv.2508.11845>
6. Caruana, R.: Multitask learning. *Machine Learning* **28**(1), 41–75 (Jul 1997). <https://doi.org/10.1023/A:1007379606734>
7. Chen, S., Zhang, Y., Yang, Q.: Multi-task learning in natural language processing: An overview. *ACM Comput. Surv.* **56**(12), 295:1–295:32 (Jul 2024). <https://doi.org/10.1145/3663363>
8. Zhao, Y., Wang, X., Che, T., Bao, G., Li, S.: Multi-task deep learning for medical image computing and analysis: A review. *Computers in Biology and Medicine* **153**, 106496 (Feb 2023). <https://doi.org/10.1016/j.compbimed.2022.106496>
9. Martin, K., Adam, O., Obin, N., Dufour, V.: Rookognise: Acoustic detection and identification of individual rooks in field recordings using multi-task neural networks. *Ecological Informatics* **72**, 101818 (Dec 2022). <https://doi.org/10.1016/j.ecoinf.2022.101818>
10. Cramer, A.L., LOSTANLEN, V., Farnsworth, A., Salamon, J., Bello, J.P.: Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 901–905 (May 2020). <https://doi.org/10.1109/ICASSP40776.2020.9052908>
11. Somervuo, P.: Simultaneous species classification and distance estimation for bird sounds using single-microphone recordings. In: 2024 32nd European Signal Processing Conference (EUSIPCO). pp. 1242–1246 (Aug 2024). <https://doi.org/10.23919/EUSIPCO63174.2024.10715289>

12. van Merriënboer, B., Dumoulin, V., Hamer, J., Harrell, L., Burns, A., Denton, T.: Perch 2.0: The bittern lesson for bioacoustics (Aug 2025). <https://doi.org/10.48550/arXiv.2508.04665>
13. Hagiwara, M., Hoffman, B., Liu, J.Y., Cusimano, M., Effenberger, F., Zacarian, K.: Beans: The benchmark of animal sounds (Oct 2022). <https://doi.org/10.48550/arXiv.2210.12300>
14. Recalde, N.M.: Wytham great tit song dataset (2023). <https://doi.org/10.17605/OSF.IO/N8AC9>
15. Stowell, D., Petrusková, T., Šálek, M., Linhart, P.: Datasets for automatic acoustic identification of individual birds (Oct 2018). <https://doi.org/10.5281/zenodo.1413495>
16. Weldy, M.J., Denton, T., Fleishman, A.B., Tolchin, J., Mckown, M., Spaan, R.S., Ruff, Z.J., Jenkins, J.M.A., Betts, M.G., Lesmeister, D.B.: Audio tagging of avian dawn chorus recordings in california, oregon, and washington (Jul 2023). <https://doi.org/10.5281/zenodo.8047850>
17. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics (Apr 2018). <https://doi.org/10.48550/arXiv.1705.07115>
18. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks (Jun 2018). <https://doi.org/10.48550/arXiv.1711.02257>
19. Gong, Y., Chung, Y.A., Glass, J.: Ast: Audio spectrogram transformer (Jul 2021). <https://doi.org/10.48550/arXiv.2104.01778>
20. Xin, D., Ghorbani, B., Garg, A., Firat, O., Gilmer, J.: Do current multi-task optimization methods in deep learning even help? (Sep 2022). <https://doi.org/10.48550/arXiv.2209.11379>
21. Lu, J., Goswami, V., Rohrbach, M., Parikh, D., Lee, S.: 12-in-1: Multi-task vision and language representation learning (Apr 2020)
22. Standley, T., Zamir, A.R., Chen, D., Guibas, L., Malik, J., Savarese, S.: Which tasks should be learned together in multi-task learning? (Sep 2020). <https://doi.org/10.48550/arXiv.1905.07553>
23. Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., Finn, C.: Efficiently identifying task groupings for multi-task learning (Oct 2021). <https://doi.org/10.48550/arXiv.2109.04617>
24. Bingel, J., Søgaard, A.: Identifying beneficial task relations for multi-task learning in deep neural networks. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 164–169. Association for Computational Linguistics, Valencia, Spain (2017). <https://doi.org/10.18653/v1/E17-2026>

Appendix: Supplementary Tables

Table 4. Hyperparameters after tuning common to all the models. The search range of each hyperparameter is specified under its name. Hyperparameters specific to the multi-task static weights and GradNorm models are not displayed.

Hyperparameter	Learning rate	Batch size	Dropout
	[1e-4, 1e-2]	{64,128}	[0,1]
AST-frozen ind.	7.3e-3	128	0.36
AST-frozen species	6.9e-3	64	0.29
AST-frozen vox type	6.2e-3	64	0.52
Single-task ind.	4.6e-4	64	0.26
Single-task species	2.0e-4	128	0.40
Single-task vox type	6.1e-4	64	0.19
Static weights	4.2e-4	64	0.22
Equal weights	6.8e-4	64	0.22
GradNorm	4.7e-4	64	0.25

Table 5. BEANS benchmark results, with each column representing one dataset. The best and second best performances are written in bold and underlined respectively.

	wtkn	bat	dogs	cbi	hbdb	esc	sc	dcase	enab	hiceas	gib	rfcx
Ast-frozen	0.85	0.654	0.82	0.388	0.767	0.922	0.74	0.399	0.521	0.503	0.195	0.121
single-task	0.56	0.419	0.561	0.081	0.639	0.688	0.199	0.251	0.248	0.432	0.024	0.041
Individual												
single-task	<u>0.69</u>	<u>0.473</u>	0.604	0.17	<u>0.684</u>	0.8	0.267	0.316	<u>0.361</u>	<u>0.495</u>	0.052	<u>0.076</u>
Species												
single-task	0.678	0.466	<u>0.647</u>	<u>0.176</u>	<u>0.684</u>	<u>0.818</u>	<u>0.268</u>	<u>0.318</u>	0.342	0.479	<u>0.084</u>	0.075
Vox type												
Equal weights	0.546	0.374	0.403	0.061	0.611	0.608	0.157	0.209	0.226	0.308	0.011	0.024
Static weights	0.64	0.432	0.54	0.092	0.651	0.688	0.2	0.248	0.278	0.488	0.029	0.045
GradNorm	0.546	0.41	0.496	0.08	0.607	0.623	0.171	0.21	0.26	0.371	0.01	0.037