

MALPOLON: A Framework for Deep Species Distribution Modeling

Theo Larcher¹

larchertheo@gmail.com

Lukas Pícek^{1,2}

lukaspicek@gmail.com

Benjamin Deneu³

benjamin.deneu@wsl.ch

Titouan Lorieul

titouan.lorieul@gmail.com

Maximilien Servajean^{1,4}

maximiliense@gmail.com

Alexis Joly¹

alexis.joly@inria.fr

¹ INRIA

Montpellier, France

² University of West Bohemia

Pilsen, Czechia

³ Swiss Federal Institute for Forest,

Snow and Landscape Research

Birmensdorf, Zurich, Switzerland

⁴ Université Paul Valéry, Montpellier

Montpellier, France

Abstract

This paper describes a deep-SDM framework, MALPOLON. Written in Python and built upon the PyTorch library, this framework aims to facilitate training and inferences of deep species distribution models (deep-SDM) and sharing for users with only general Python language skills (e.g., modeling ecologists) who are interested in testing deep learning approaches to build new SDMs. More advanced users can also benefit from the framework's modularity to run more specific experiments by overriding existing classes while taking advantage of press-button examples to train neural networks on multiple classification tasks using custom or provided raw and pre-processed datasets. The framework is open-sourced on [GitHub](#) and [PyPi](#) along with extensive documentation and examples of use in various scenarios. MALPOLON offers straightforward installation, YAML-based configuration, parallel computing, multi-GPU utilization, baseline and foundational models for benchmarking, and extensive tutorials/documentation, aiming to enhance accessibility and performance scalability for ecologists and researchers.

1 Introduction

Species distribution models (SDMs) are popular numerical tools [[0](#), [33](#)] used for predicting the distribution of species over a geographic area and temporal frame, by trying to find correlations between environmental data and observation data. These models can be used to describe how environmental conditions or anthropogenic actions influence the occurrence or abundance of species and allow for future predictions [[18](#)]. Historically, this work has been carried out by ecologists, botanists, or environmental researchers with a strong statistical background and expertise with mechanistic (*process-based*) or statistical algorithms.

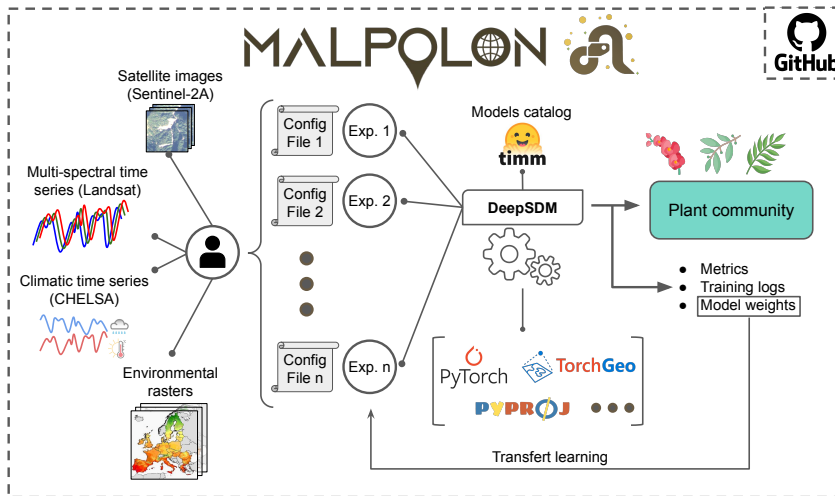


Figure 1: **Graphical abstract.** MALPOLON allows straightforward: (i) loading of various predictors, such as environmental rasters (e.g., land cover, human footprint), remote sensing data (e.g., Sentinel-2A and Landsat), and bioclimatic time-series, (ii) use of geospatial foundational models (e.g., SatCLIP, GeoCLIP), (iii) model training with the press of a button.

Common methods used in species distribution modeling include BIOMOD [40], Maximum Entropy (MAXENT) [52, 53, 54], Generalized Linear Models (GLM), Generalized additive model (GAM) [29], Random forests [8], Boosted regression trees, Gradient Boosting, and Support Vector Machines. However, recent research has highlighted the potential of using deep learning methods such as CNNs to perform Presence Only (PO) or Presence Absence (PA) species prediction with competitive performances [4, 9, 9, 10]. So-called deep-SDM models offer an advantage in predicting suitability scores for many species using a unified model rather than relying on multiple individual single-species models. They excel in capturing relations among diverse environmental data [5, 10, 10]. Although promising, such models are often developed and trained independently, lacking a shared foundation, which complicates reproducibility and accessibility. Furthermore, within the environmental research community focused on SDMs, the predominant tools and packages are rooted in the CPU-oriented language R. Many established SDM frameworks and widely-used statistical algorithms [1, 5, 15] have historically been developed within R, fostering community growth and the evolution of new methodologies over time. However, this reliance on R presents limitations in terms of computational capabilities and access to emerging methods like Deep Neural Networks (DNNs), which are predominantly implemented in Python and C++.

Despite being limited in number, recent development efforts have yielded a few solutions to establish and support deep learning SDM frameworks. For example, Pichler and Hartig [57] introduced *sjSDM*, which leverages PyTorch within R, enabling GPU utilization. However, integrating PyTorch code into an R package can decrease code and memory efficiency while also increasing the complexity and cognitive load associated with coding. Lu [27] proposed a Python implementation of Maxent through Deep Inverse Reinforcement Learning [44], however despite enabling a direct utilization of GPU capabilities via TensorFlow the framework relies on reward maps and does not handle observation / environmental predictor pairs as input data. Gillespie et al. [10] have developed a deep learning framework

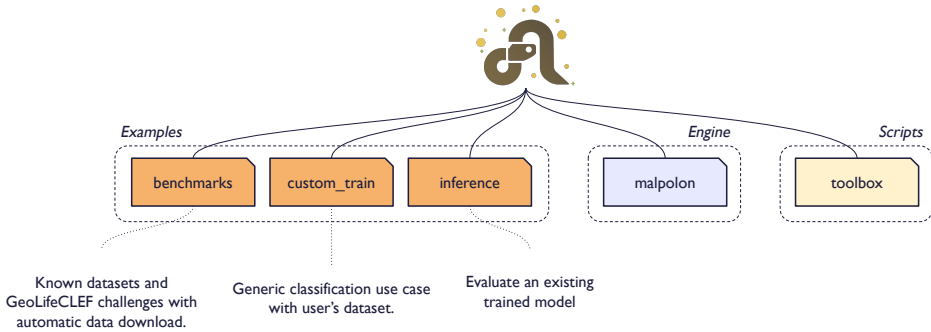


Figure 2: **Macro structure of MALPOLON.** The *Examples* consist of different use case experiments with pre-written plug-and-play examples and include training, inference, etc. The *Engine* contains everything important for datasets and models loading and usage. The *Toolbox* provides a collection of useful scripts to perform data pre-processing.

to perform plant prediction using citizen science and remote sensing data from California, enabling observation of species distribution shifts. However, this work is more focused on the model they propose and less about the framework in itself. While it is well-documented with comments, tutorials, and troubleshooting guidance, users need a solid understanding of Python and PyTorch to adapt it for custom use cases, as the framework is not modular.

In light of that, we have developed MALPOLON, a new framework for deep species distribution modeling (see Figure 1). It allows training various types of multi-modal SDMs based on deep learning using geolocated observations combined with a variety of input variables such as environmental rasters (e.g., land cover, human footprint, etc.), remote sensing images (e.g., Sentinel-2A [13] and Landsat [14]), and time series (e.g., Landsat, and bioclimatic). MALPOLON is built on top of PyTorch Lightning and is distributed through PyPi and available on GitHub, alongside installation guidelines, tutorials, and documentation.

2 The framework

MALPOLON is a new Python-based framework for deep species distribution modeling, which is designed not only for ML researchers but also for people without extensive knowledge of Python and PyTorch. The framework is built on top of PyTorch Lightning, which makes it highly modular and customizable, as its classes can be re-defined to better control the functions handling the datasets, data module, training loops, and optimizer. MALPOLON is compatible with TorchGeo, a PyTorch domain library providing datasets, samplers, transforms, and pre-trained models specific to geospatial data. Furthermore, it allows using various types of neural networks ranging from simple MLP to more complex Transformer- and Convolutional-based architectures. We also provide a variety of geospatial foundational models such as GeoCLIP [13] and SatCLIP [14]. More importantly, MALPOLON offers straightforward access to various standards predictors as well as new ones, including but not limited to satellite data (e.g., Sentinel-2A and Landsat), climatic rasters (e.g., temperature and precipitation), and environmental rasters (e.g., soil grids, land cover, and human footprint). A collection of standalone data processing scripts is also provided under the name of *toolbox*. We visualize MALPOLON’s meta-class diagram in Figure 2.

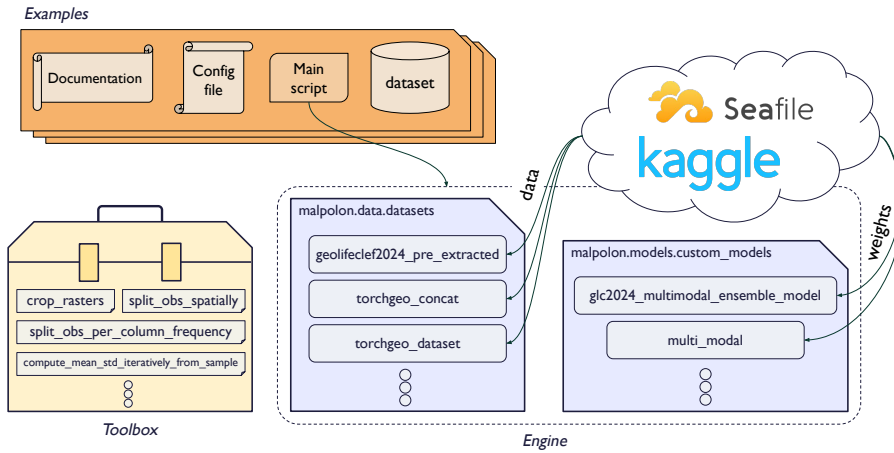


Figure 3: **Main components of MALPOLON.** The framework contains custom datasets and models (in blue), which data and weights are automatically retrieved from remote servers. The toolbox (in yellow), provides standalone data processing scripts. Examples (in orange) are provided when cloning the GitHub project and interact with the engine to run models for training or inference.

2.1 Tools and Methods

Experiments are framed around three entities: (i) **dataset**, (ii) **data module**, and (iii) **model**, that are later fed into a custom PyTorch Lightning pipeline.

The **datasets** define how data and labels are queried and returned when iterating through the dataset. The framework provides specialized dataset classes tailored for handling geospatial rasters, image patches, or combinations of both. Torchgeo-based datasets for geospatial rasters are particularly flexible with different Coordinate Reference Systems (CRS). They allow observations to be matched with rasters even if they are in different CRSs, eliminating the need for coordinate transformations or raster deformations.

The **data module** is responsible for creating PyTorch `dataloaders` for training, validation, and test sets, while also managing data pre-processing and loading according to the specified dataset. Once created, these PyTorch objects are then forwarded to a PyTorch Lightning `Trainer`, which handles the pipeline.

The **model** holds the architecture to train, redefining the `forward` and `step` functions to match the dataset’s tuple output for model computation. Additionally, it handles consistent computation and performance metrics logging throughout the optimization process.

By default, all experiments print a summary of the architecture, set the checkpoint saving strategy to validation loss decrease, and save the experiments’ metrics and loss in CSV and Tensorboard files respectively. While training, a progress bar with the training status and training time estimation is provided. Furthermore, the model’s loss and performance metrics are logged regularly to local [Tensorboard](#) files, which allows visualizing the training by using the bash command `tensorboard --logdir logs/`. After training, the validation metrics of the best checkpoint are displayed, and the last output files are logged in the output directory and, if selected, "uploaded" to [Tensorboard](#) or [Weights & Biases](#).

2.2 Data Availability

Recent advances in species distribution modeling have shown that on top of environmental data, time-series and satellite imagery also contribute strongly to new model performances [11, 30, 69]. However, the data is usually extremely disk space-demanding and time-consuming to pre-process, especially when combining multiple modalities and remote sensing data at the same time. Hence why, efforts have been made to synthesize and compress such data information in more conveniently compact data formats such as EcoDataCube [28], OpenDataCune [6], and EuroDataCube which still can include gigabytes or terabytes of data. In light of that, we build MALPOLON to provide straightforward access to all standard and even more complex data (see Figure 3). So far, we allow loading the data from Sentinel-2A, Landsat, CHELSA [24, 31], Worldclim [14, 19], Soilgrids [21], ASTER Global Digital Elevation Model [12], MODIS Terra+Aqua (land cover) [16], DRYAD (human footprint) [17] and observations from GBIF and PI@ntNet.

The time-series data (e.g., Landsat and CHELSA climatic variables) are also available as lightweight tensor *cubes* where the information has been compressed down to geolocated points extraction based on species observation files. The easy loading of these cubes with PyTorch and their readiness make them a strong asset of MALPOLON’s provided multi-modal datasets and baseline models.

2.3 Use Cases

While building MALPOLON, we have considered various scenarios and types of users. For the three most common "scenarios", we have prepared a set of experiments and comprehensive documentation. All three scenarios and how to work with them are further described below:

"Custom train". The approach for custom dataset training is straightforward. Duplicate an experiment’s main script, update its configuration file, and select a suitable data loader. If existing data loaders are insufficient, one could simply update the classes. Next, users choose a model by specifying the name and parameters in the config file (all TIMM models are available), and finalize the model parameters using custom or default values. Metrics, logs, and model weights are saved in a unique output folder.

"Inference". For a scenario with just a pre-trained model inference, we made available all models available in TIMM or Torchvision. Besides, we provide selected foundational models such as GeoCLIP [23]. The recommended way to perform inference with a trained model on a test dataset is to use or duplicate an experiment’s main script and update the dataset path and the model’s weight path. Similarly to the "Custom train" scenario, users may import their modifications to data loaders and other classes before running the inference pipeline. Metrics, logs, and predictions are then outputted in a unique output folder.

"Benchmarks". For a plug-and-play benchmarking on existing datasets such as those provided within the annual GeoLifeCLEF competition [65] (organized on Kaggle in conjunction with FGVC-CVPR and LifeCLEF workshops [14, 23]), we provide data loaders and training examples to allow an easier start for anyone interested in deep species distribution modeling.

2.4 Baseline Experiments and How to Use Them

To make the framework accessible to ecologists, each experiment contains a CLI script ready to be run in the terminal with a single line command and a tutorial file detailing its purpose, how to configure it, the expected data, and running instructions. Three baseline experiments were provided for the "Custom train" and "Benchmark" scenarios for which users only need to update the following sections in the YAML file if needed:

- **run**: parameters related to prediction and transfer learning
- **data**: data-related information, such as the path to a dataset or batch size
- **task**: type of experiment, e.g., binary, multiclass, or multilabel classification.
- **trainer**: parameters used for tweaking a training or inference session
- **model**: defines the provider, architecture, and hyperparameters listed in the given provider's model builder. Additionally, it allows to call model modifiers to adapt their model structure. Namely, three modifiers are provided by MALPOLON to change a model's input and output shape.
- **optimizer**: defines the optimization algorithm, the loss and their hyperparameters

2.5 Assets

The MALPOLON GitHub repository contains the mentioned datasets, models, baseline scripts and notebooks as well as instructions on how to install and use the framework for different use cases. Additionally, each "use case" contains detailed documentation explaining how to run and tune the experiment and giving insights about the data used by it. Additionally, there are handy data transformation and visualization scripts being used to prepare the datasets used in many examples. Besides, we provide standardized code documentation, which is being updated every time new content is pushed to the main branch.

3 Baseline performance

To test the suitability and versatility of MALPOLON, we opt to train on the data and models provided in the context of the recent GeoLifeCLEF competition [65] organized in conjunction with FGVC-CVPR and LifeCLEF workshops [22, 23]. We have trained a multimodal ensemble model (MME) and single-modality models using the Sentinel-2A image patches, Landsat time series, and CHELSA bioclimatic time series¹. The architecture consists of two ResNet-18 [20] encoders for time series data and one Swin-v2 Transformer [26] that encodes Sentinel-2A image patches; feature vectors are then concatenated and forwarded into a MLP. For training, we used the GLC24 dataset [65, 66] consisting of 1.4M+ observations from 80k Presence Absence plots with 11k+ species scattered over Europe. The dataset is highly imbalanced as labels are characterized by a long-tailed distribution. The data have been spatially split into training (85%) and testing (15%) sets, using a spatial block holdout technique excluding training set points from validation "zones" of size 10 *arcminutes*, which aims at reducing spatial bias. See Figure 4 for the geospatial distribution of the data.

¹All architectures were provided by GeoLifeCLEF organizers.

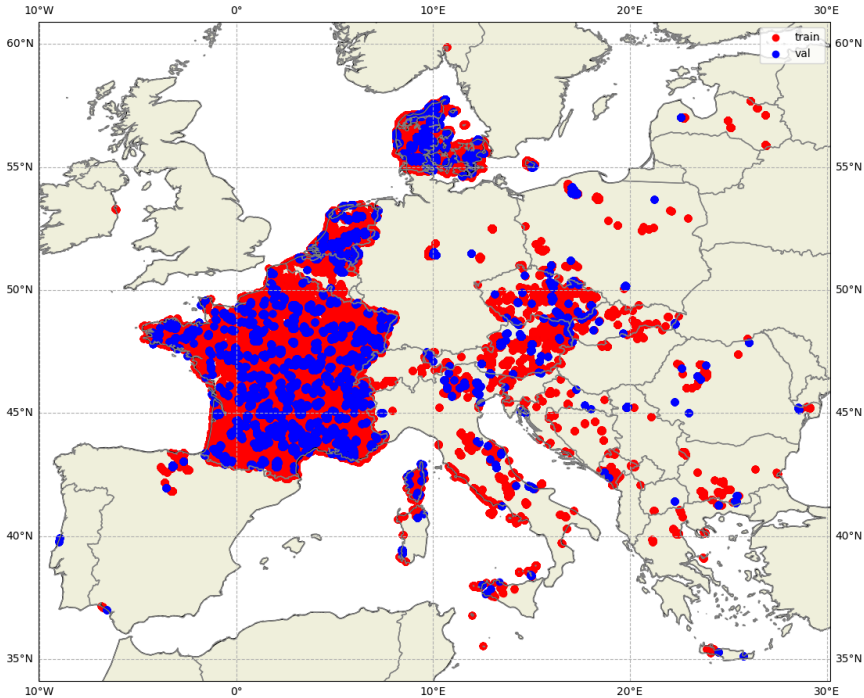


Figure 4: Spatial split of training and validation data points.

Experimental Settings: To allow direct comparability, we use the same architectures as used in the provided GLC24 baselines, i.e., ResNet-18 for Landsat and Bioclimatic data and Swin-v2-b for Sentinel2 images. All ResNet and Swin models were fine-tuned from pre-trained ImageNet-1k weights. The ResNet-18 first convolutional layer input sizes were adapted to fit the expected size of the provided input variables, i.e., 3d cubes². The MME model is a straightforward concatenation of the single-modality embeddings followed by two linear layers (first with a dropout with 10% probability).

All models were trained for 20 epochs with a batch size of 64 using AdamW optimizer³. In order to predict multi-label species present in the plots, we used *BCEWithLogitsLoss* with a sigmoid activation function. Additionally, in order to strengthen the rewarding process of correctly predicted species, the loss positive weight parameter was set to 10.

Models are being evaluated by 2 main metrics adapted for imbalanced datasets: F1-score (F1) and ROC-AUC (as defined per Scikit-learn). The F1-score is more appropriate than a basic Accuracy as it puts more emphasis on False negatives and False positives. The ROC-AUC tracks the balance between True positives and False positives. Since we are most interested in avoiding False negatives, it makes sense to evaluate the model with such metrics. Additionally, we provide Precision (P) and Recall (R) to give more insight on the positive/negative rates. Finally, as each survey contains in average 20 species, we only compute F1, P and R on the Top-25 returned species.

²Conv2D parameters, e.g., kernel size, stride, and padding, were also changed; for more details, refer to [15]

³AdamW was parametrized with a learning rate of $2.5e10^{-4}$ and associated with a *CosineAnnealingLR* scheduler parametrized with a max temperature of 25

Results: As expected, a model trained using MALPOLON performed similarly to the model trained in the provided baseline notebook. Additionally, MALPOLON outputs the best model’s weights, handles multi-process and multi-GPU usage seamlessly, and logs training progress to Tensorboard. Switching from training to prediction is as easy as changing two values in the config file. Finally, the framework provides convenient plotting functions to help visualize the model’s predictions. Regarding performances, since MALPOLON runs Python scripts, it is more efficient than notebooks, and PyTorch Lightning optimizes the distributed computations across CPU cores and GPU units. The main bottleneck resides in the datasets, which, when building custom ones, users should be mindful of making optimized.

Model	<i>Micro averaged</i>				<i>Sample averaged</i>				<i>Macro averaged</i>			
	AuC	P	R	F1	AuC	P	R	F1	AuC	P	R	F1
<i>ResNet-18_l</i>	94.7	23.4	32.7	27.3	94.8	23.4	39.1	26.5	88.5	12.4	11.0	7.7
<i>ResNet-18_b</i>	95.1	21.1	29.4	24.5	94.9	21.1	34.1	23.5	90.0	16.0	12.1	7.6
<i>Swin-v2_s</i>	94.3	20.4	28.2	23.7	94.5	20.4	34.8	23.3	88.3	10.3	8.7	5.7
MME	96.7	26.2	36.5	30.5	96.3	26.2	43.4	29.6	93.2	18.9	14.5	9.4

Table 1: **Baseline performance for selected custom architectures.** The Multimodal Ensemble (MME) model provides better performances than standard CNN and transformers on our metrics, i.e., Top-25 Precision (P), Top-25 Recall (R), Top-25 F1-score (F1), and Area under the Curve (AuC) computed in samples, macro and micro averaged. *CNN_l*: ResNet-18 trained on Landsat data; *CNN_b*: ResNet-18 trained on CHELSA bioclimatic data; *Swin-v2_s*: Swin-v2 transformer trained on Sentinel-2A data.

3.1 Benchmarking foundational models with MALPOLON

Deep learning foundational models are trained on large amounts of broad data such that they can be applied across a wide range of use cases. As such, they are very much prized and beneficial to bootstrap topic-specific training as they allow reducing training times, which can range from hours to weeks if trained from scratch, depending on the data used. In the domain of species distribution modeling, such models are scarce, but interesting options, such as SatCLIP proposed by Klemmer et al. [15] and GeoCLIP proposed by Vivanco Cepeda et al. [16], have emerged. Hereafter, we investigate the added value of these foundational models within the MME model in place of the Swin-v2 transformer, as well as an individual model to handle the Sentinel-2A modality.

Results: GeoCLIP outperforms the other two models in micro and sample-averaged metrics, with higher precision and recall, but like Swin-v2, it also falls short in macro-averaged performance, indicating limitations in its ability to generalize across all classes. On the other hand, SatCLIP underperformed heavily. The variant MME[†], which replaces Swin-v2 with GeoCLIP, also demonstrates competitive performance, particularly excelling in precision metrics. It achieves the highest micro-averaged precision of 30.1% and sample-averaged precision of 29.9%, suggesting that integrating GeoCLIP enhances the model’s ability to accurately identify positive instances. However, it shows slightly lower performance in macro-averaged metrics compared to the standard MME, indicating some variance in performance across different classes. For a more comprehensive evaluation, see Table 2.

Model	<i>Micro averaged</i>				<i>Sample averaged</i>				<i>Macro averaged</i>			
	AuC	P	R	F1	AuC	P	R	F1	AuC	P	R	F1
<i>Swin-v2_s</i>	94.3	20.4	28.2	23.7	94.5	20.4	34.8	23.3	88.3	<u>10.3</u>	<u>8.7</u>	<u>5.7</u>
<i>SatCLIP</i>	89.5	11.5	14.5	12.8	89.5	10.4	13.3	11.1	85.0	0.9	2.4	1.0
<i>GeoCLIP</i>	95.1	24.1	30.4	26.9	95.1	22.8	30.1	24.6	<u>88.5</u>	2.8	3.1	2.6
<i>MME</i>	96.7	<u>26.2</u>	<u>36.5</u>	<u>30.5</u>	96.3	26.2	43.4	29.6	93.2	18.9	14.5	9.4
<i>MME</i> [†]	<u>95.8</u>	30.1	37.9	33.5	96.0	29.9	<u>40.6</u>	32.2	<u>88.5</u>	3.6	3.9	3.4

Table 2: **MME baseline performance integrating foundational models** The MME model integrating GeoCLIP provides yet better performance in terms of micro and sample average evaluation, highlighting the relevance of foundational models paired with CNNs. The metrics used are the same as mentioned in Table 1. [†]MME with Swin-v2 replaced for GeoCLIP.

4 Conclusion

In this work, we introduced "MALPOLON", a new PyTorch-based framework that enables ecologists to easily train deep-SDMs. The framework supports multi-GPU computation and provides access to neural networks to train models on classification and regression tasks to predict species distribution over geographical areas. The framework is modular and follows encapsulation principles which makes it adapted for expert users knowledgeable about Python or PyTorch. However, it is also adapted for less experienced users coming from R, thanks to its online documentation, tutorial files and plug-and-play examples provided with the project’s repository. We showed that the framework can train baselines with complex deep architectures on a real-world use case, GeoLifeCLEF 2024, with strong metrics performance relative to other methods submitted on the [Kaggle challenge page](#). Such architectures are then provided as part of the framework’s available models and are coupled with additional examples for easy training. Furthermore, the data used in all examples is made accessible for reproducibility and alleviates the need to manually gather and transform data from different sources. The framework is open-sourced on [GitHub](#) and [PyPi](#), which enables it to use the wide Python community for future development.

Limitations: As of now, too few models were trained using MALPOLON to provide a deep comparison analysis with the MME model. Likewise, the lack of similar open-sourced deep learning framework available makes it difficult to compare MALPOLON’s features to those of existing frameworks, which do not operate using the same language or model types.

Next Steps: Preliminary results have been conducted to predict *EUNIS habitat types* instead of species. A convergence of efforts will be made to enable MALPOLON to support not only deep-SDM but also deep Habitat Distribution Modeling (deep-HDM).

Acknowledgement

The research described in this paper was partly funded by the European Commission via the GUARDEN and MAMBO projects, which have received funding from the European Union’s Horizon Europe research and innovation program under grant agreements 101060693 and 101060639. The opinions expressed in this work are those of the authors and are not necessarily those of the GUARDEN or MAMBO partners or the European Commission.

References

- [1] Adde Antoine, Rey Pierre-Louis, Brun Philipp, Külling Nathan, Fopp Fabian, Altermatt Florian, Broennimann Olivier, Lehmann Anthony, Petitpierre Blaise, Zimmermann Niklaus, Pellissier Loïc, and Guisan Antoine. N-sdm: a high-performance computing pipeline for nested species distribution modelling. *Ecography*, 2023(6), 2023.
- [2] Miguel B. Araújo, Robert P. Anderson, A. Márcia Barbosa, Colin M. Beale, Carsten F. Dormann, Regan Early, Raquel A. Garcia, Antoine Guisan, Luigi Maiorano, Babak Naimi, Robert B. O'Hara, Niklaus E. Zimmermann, and Carsten Rahbek. Standards for distribution models in biodiversity assessments. *Standards for distribution models in biodiversity assessments*, 5(1), 2019.
- [3] Trevor H. Booth, Henry A. Nix, John R. Busby, and Michael F. Hutchinson. bioclim: the first species distribution modelling package, its early applications and relevance to most current maxent studies. *Diversity and Distributions*, 20(1):1–9, 2014.
- [4] Christophe Botella, Alexis Joly, Pierre Bonnet, Pascal Monestiez, and François Munoz. A deep learning approach to species distribution modelling. *Multimedia Tools and Applications for Environmental & Biodiversity Informatics*, page 169–199, June 2018.
- [5] Christophe Botella, Benjamin Deneu, Diego Marcos, Maximilien Servajean, Théo Larcher, César Leblanc, Pierre Bonnet, and Alexis Joly. Overview of geolifeclef 2023: Species composition prediction with high spatial resolution at continental scale using remote sensing. *CLEF 2023: Conference and Labs of the Evaluation Forum*, 12, December 2023.
- [6] Killough Brian. Overview of the open data cube initiative. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 8629–8632, 2018. doi: 10.1109/IGARSS.2018.8517694.
- [7] Di Chen, Yexiang Xue, S Chen, Daniel Fink, and Gomes CP. Deep multi-species embedding. *CoRR*, abs/1609.09353, February 2017.
- [8] D. Richard Cutler, Thomas C. Edwards Jr., Karen H. Beard, Adele Cutler, Kyle T. Hess, Jacob Gibson, and Joshua J. Lawler. Random forests for classification in ecology. *Ecology*, 88(11), November 2007.
- [9] Benjamin Deneu, Maximilien Servajean, Christophe Botella, and Alexis Joly. Location-based species recommendation using co-occurrences and environment- geolifeclef 2018 challenge. *CLEF 2018 - Conference and Labs of the Evaluation Forum*, 2018.
- [10] Benjamin Deneu, Maximilien Servajean, Pierre Bonnet, Christophe Botella, François Munoz, and Alexis Joly. Convolutional neural networks improve species distribution modelling by capturing the spatial structure of the environment. *PLOS Computational Biology*, 2021.
- [11] Joaquim Estopinan, Maximilien Servajean, Pierre Bonnet, François Munoz, and Alexis Joly. Deep species distribution modeling from sentinel-2 image time-series: a global scale analysis on the orchid family. *Frontiers in Plant Science*, 3, 2022.

- [12] Venter Oscar et al. Data from: Global terrestrial human footprint maps for 1993 and 2009. *Dryad*, 2016. doi: <https://doi.org/10.5061/dryad.052q5>.
- [13] European Space Agency. Copernicus sentinel-2 (processed by esa), msi level-2a boa reflectance product, 2021.
- [14] Stephen E. Fick and Robert J. Hijmans. Worldclim 2: new 1km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37(12): 4302–4315, 2017. doi: [doi:10.1002/joc.5086](https://doi.org/10.1002/joc.5086).
- [15] Scott D. Foster, David Peel, Geoffrey R. Hosack, Andrew Hoskins, David J. Mitchell, Kirstin Proft, Wen-Hsi Yang, David E Uribe-Rivera, and Jens G. Froese. ‘risdm’: species distribution modelling from multiple data sources in r. *Ecography*, 2024(6), 2024.
- [16] Mark A. Friedl, Damien Sulla-Menashe, Bin Tan, Annemarie Schneider, Navin Ramankutty, Adam Sibley, and Xiaoman Huang. Modis collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment*, 114:168–182, January 2010. doi: [10.1016/j.rse.2009.08.016](https://doi.org/10.1016/j.rse.2009.08.016).
- [17] Lauren E. Gillespie, Megan Ruffley, and Moises Exposito-Alonso. Deep learning models map rapid plant species changes from citizen science and remote sensing data. *Proceedings of the National Academy of Sciences*, 121(37):e2318296121, 2024. doi: [10.1073/pnas.2318296121](https://doi.org/10.1073/pnas.2318296121). URL <https://www.pnas.org/doi/abs/10.1073/pnas.2318296121>.
- [18] Antoine Guisan and Wilfried Thuiller. Predicting species distribution: offering more than simple habitat models. *Ecology letters*, 8(9), 2005.
- [19] Ian Harris, Timothy J. Osborn, Phil Jones, and David Lister. Version 4 of the cru ts monthly high-resolution gridded multivariate climate dataset. *Scientific Data*, 7(109), 2020. doi: [doi:10.1038/s41597-020-0453-3](https://doi.org/10.1038/s41597-020-0453-3).
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [21] ISRIC World Soil Information. Collection of soilgrid data, 2020. URL <https://soilgrids.org/>.
- [22] Alexis Joly, Lukáš Pícek, Stefan Kahl, Hervé Goëau, Vincent Espitalier, Christophe Botella, Benjamin Deneu, Diego Marcos, Joaquim Estopinan, Cesar Leblanc, et al. Lifeclef 2024 teaser: Challenges on species distribution prediction and identification. In *European Conference on Information Retrieval*, pages 19–27. Springer, 2024.
- [23] Alexis Joly, Lukáš Pícek, Stefan Kahl, Hervé Goëau, Vincent Espitalier, Christophe Botella, Diego Marcos, Joaquim Estopinan, Cesar Leblanc, Théo Larcher, Milan Šulc, Marek Hruží, Maximilien Servajean, Hervé Glotin, Robert Planqué, Willem-Pier Vellinga, Holger Klinck, Tom Denton, Ivan Eggel, Pierre Bonnet, and Henning Müller. Overview of lifeclef 2024: Challenges on species distribution prediction and identification. In Lorraine Goeuriot, Philippe Mulhem, Georges Quénot, Didier Schwab,

- Giorgio Maria Di Nunzio, Laure Soulier, Petra Galuščáková, Alba García Seco de Herrera, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 183–207, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-71908-0.
- [24] Dirk Nikolaus Karger, Olaf Conrad, Jürgen Böhrner, Tobias Kawohl, Holger Kreft, Rodrigo Wilber Soria-Auza, Niklaus E. Zimmermann, H. Peter Linder, and Michael Kessler. Climatologies at high resolution for the earth land surface areas. *Scientific Data*, 4(170122), 2017. doi: 10.1038/sdata.2017.122.
- [25] Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. Satclip: Global, general-purpose location embeddings with satellite imagery. *arXiv preprint arXiv:2311.17179*, 2023.
- [26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12009–12019, June 2022.
- [27] Yiren Lu. Implementations of inverse reinforcement learning algorithms in Python/Tensorflow, 7 2017. URL <https://github.com/yrlu/irl-imitation>.
- [28] Witjes M., Parente L., Križan J., Hengl T., and AntoniĆ L. Ecodatacube.eu: analysis-ready open environmental data cube for europe. *PeerJ Environmental Science*, June 2023. doi: 10.7717/peerj.15478.
- [29] Gretchen G. Moisen, Elizabeth A. Freeman, Jock A. Blackard, Tracey S. Frescino, Niklaus E. Zimmermann, and Thomas C. Edwards Jr. Predicting tree species presence and basal area in utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods. *Ecological*, 199(2):176–187, November 2006.
- [30] Gaetan Morand, Alexis Joly, Tristan Rouyer, Titouan Lorieul, and Julien Barde. Predicting species distributions in the open oceans with convolutional neural networks. *bioRxiv*, 2023.
- [31] Karger Dirk N., Conrad Olaf, Böhrner Jürgen, Kawohl Tobias, Kreft Holger, Soria-Auza Rodrigo, Zimmermann Niklaus, Linder H. Peter, and Kessler Michael. Data from: Climatologies at high resolution for the earth’s land surface areas. *Dryad Digital Repository*, August 2018. doi: doi:10.5061/dryad.kd1d4.
- [32] Steven J Phillips and Miroslav Dudík. Modeling of species distributions with maxent: new extensions and a comprehensive evaluation. *Ecography*, 31(2):161–175, 2008.
- [33] Steven J. Phillips, Robert P. Anderson, and Robert E. Schapire. Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190(3):231–259, 2006. ISSN 0304-3800. doi: 10.1016/j.ecolmodel.2005.03.026.
- [34] Steven J. Phillips, Robert P. Anderson, and Robert E. Schapire. Maximum entropy modeling of species geographic distributions. *Ecological modelling*, 190(3-4):231–259, January 2006.

- [35] Lukas Picek, Christophe Botella, Maximilien Servajean, Benjamin Deneu, Diego Marcos Gonzalez, Rémi Palard, Théo Larcher, César Leblanc, Joaquim Estopinan, Pierre Bonnet, and Alexis Joly. Overview of GeoLifeCLEF 2024: Species presence prediction based on occurrence data and high-resolution remote sensing images. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, 2024.
- [36] Lukas Picek, Christophe Botella, Maximilien Servajean, César Leblanc, Rémi Palard, Théo Larcher, Benjamin Deneu, Diego Marcos, Pierre Bonnet, and Alexis Joly. Geoplant: Spatial plant species prediction dataset. *arXiv preprint arXiv:2408.13928*, 2024.
- [37] Maximilian Pichler and Florian Hartig. A new joint species distribution model for faster and more accurate inference of species associations from big community data. *Software Impacts*, 12(11):2159–2173, November 2021. doi: 10.1111/2041-210X.13687.
- [38] Vivek Srivastava, Valentine Lafond, and Verena C. Griess. Species distribution models (sdm): applications, benefits and challenges in invasive species management. *CABI Reviews*, 5(1), April 2019.
- [39] S. B. Stewart, J. Elith, M. Fedrigo, S. Kasel, S. H. Roxburgh, L. T. Bennett, M. Chick, T. Fairman, S. Leonard, M. Kohout, J. K. Cripps, L. Durkin, and C. R. Nitschke. Climate extreme variables generated using monthly time-series data improve predicted distributions of plant species. *Ecography*, 44(4):626–639, 2021. doi: <https://doi.org/10.1111/ecog.05253>. URL <https://nsojournals.onlinelibrary.wiley.com/doi/abs/10.1111/ecog.05253>.
- [40] Wilfried Thuiller, Bruno Lafourcade, Robin Engler, and Miguel B. Araújo. Biomod – a platform for ensemble forecasting of species distributions. *Ecography*, 32(3):369–373, June 2009.
- [41] U.S. Geological Survey. Landsat collections, 2019. URL <https://www.usgs.gov/landsat-missions>.
- [42] U.S./Japan ASTER Science Team. Aster global digital elevation model 1 arc second, 2013. URL <https://lpdaac.usgs.gov/products/astgtmv003/>.
- [43] Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. Geoclip: Clip-inspired alignment between locations and images for effective worldwide geolocalization. *Advances in Neural Information Processing Systems*, 36, 2024.
- [44] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *CoRR*, abs/1507.04888, 2015. URL <http://arxiv.org/abs/1507.04888>.