Supplementary Information

Wild Boar Collision Data and Satellite Computer Vision Refine Habitat Suitability Mapping across France

Adrian Ferdinand Meyer [1,2]; Théo Reibel [1]; Kevin Morelle [3,4]; Mathias Kneubühler [2]; Denis Jordan [1]

1) University of Applied Sciences and Arts Northwestern Switzerland FHNW, Muttenz, Switzerland, Institute Geomatics
2) University of Zurich, Zurich, Switzerland, Remote Sensing Laboratories, Department of Geography
3) Max Planck Institute of Animal Behavior, Radolfzell, Germany
4) Minuartia, Barcelona, Spain

Corresponding author: Adrian Meyer, adrian.meyer@fhnw.ch, Hofackerstr. 30, 4132 Muttenz, Switzerland

Table of Contents

S1 – Occurrence Point Data Thinning Algorithm

We compiled presence-only occurrences (2017-2023) from SNCF Réseau railway WVCs (Brehat), RRN roadkill records, and GBIF observations. Records were filtered for coordinate uncertainty < 0.5 km and temporal resolution of at least one month, de-duplicated per 1 km² grid cell, and assigned to summer (Mar-Aug) or winter (Sep-Feb).

To reduce spatial clustering and environmental redundancy while preserving niche coverage, we applied environmental-space thinning (Aiello-Lammens et al. 2015). The method operates in multivariate predictor space using a Mahalanobis distance representation (Beaugrand et al. 2011; de Oliveira et al. 2014; Sillero and Barbosa 2021):

1. Represent each occurrence i by its predictor vector x (for the relevant year and season):

$$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,p}]^\top, i = 1, \dots, n$$

   x: Environmental predictor vector of occurrence i
   n: Number of occurrences before thinning
   p: Number of environmental predictors used (e.g., temperature, NDVI, elevation)

2. Compute the covariance matrix as Σ of occurrence vectors and whiten the data via $\Sigma^{-\frac{1}{2}}$ so that Mahalanobis distances become Euclidean distances in the transformed space (Etherington 2019). The covariance matrix Σ of all occurrence vectors is computed as:

$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

   where the mean vector $\boldsymbol{\mu}$ is:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

   The whitening transformation then applies as:

$$\mathbf{x}_i^* = \Sigma^{-\frac{1}{2}}(\mathbf{x}_i - \boldsymbol{\mu})$$

   In the whitened space, Mahalanobis distances reduce to Euclidean distances:

$$D_{ij} = \| \mathbf{x}_i^* - \mathbf{x}_j^* \|_2 = \sqrt{(\mathbf{x}_i^* - \mathbf{x}_j^*)^\top(\mathbf{x}_i^* - \mathbf{x}_j^*)}$$

3. Compute pairwise distances in whitened space and select a subset using greedy farthest-point sampling (Eldar et al. 1997; Chamon and Ribeiro 2017): start from a random occurrence and iteratively retain the point with maximum minimum-distance to all currently retained points.

   Initially, select a random occurrence vector as the starting point. Iteratively select the next occurrence that maximizes the minimal distance to the set R of previously selected points: ($R = \{\mathbf{x}_r^*\}$, randomly chosen), where r belongs to an uniform distribution and $\mathbf{x}_r^*$ is the whitened environmental predictor vector of occurrence at index r.

   Iterative selection step:

$$\mathbf{x}_{\text{next}}^* = \arg \max_{\mathbf{x}_i^* \notin R}\left(\min_{\mathbf{x}_j^* \in R} D_{ij}\right)$$

   Update retained set:

$$R \leftarrow R \cup \{\mathbf{x}_{\text{next}}^*\}$$

4. Stop when a tunable target retained fraction is reached; in practice, we retained approximately 40% of records per stream to balance bias reduction and sample size (Fourcade et al. 2014).

This strategy preferentially retains occurrences near the convex hull of the environmental cloud, maintaining broad coverage of the realized niche while down-weighting densely repeated conditions (Preparata and Shamos 1985; Jiménez and Soberón 2022).

S2 – Bias Correction Algorithm

Background point sampling and selection is a key step in presence-only based SDMs, because it can significantly distort suitability predictions unless sampling effort is explicitly modeled (Lobo et al. 2010; Barbet-Massin et al. 2012). Uneven observer effort and accessibility can lead models to interpret sampling patterns as habitat preference requiring effective spatial correction strategies (Baker et al. 2024). To mitigate such effects, we construct spatially explicit sampling bias surfaces for each occurrence source and each year/season to model observation effort as a proxy for realistic sampling intensity (Lobo et al. 2010). This increasingly endorsed practice in presence-only SDMs aligns background point generation with actual detectability (Phillips and Dudík 2008; Beck et al. 2014). Syfert et al. (2013) found that incorporating a sampling bias grid significantly enhanced goodness-of-fit metrics, even though it did not completely eliminate bias effects. Meta-analyses and simulations also support modest but meaningful gains from bias correction on independent test data (Baker et al. 2024).

GBIF occurrence data are available in large quantities for wild boar in Europa (Croft et al. 2025), but are known to suffer from strong spatial biases such as over-representation near roads, urban centers and accessible sites (Beck et al. 2014; Meyer et al. 2016). These biases may correlate with environmental predictors and lead to misleading niche estimations (Baker et al. 2022; Dubos et al. 2022). To mitigate such effects, recent work recommends modeling sampling effort explicitly (Sicacha-Parada et al. 2019) and implementing density surfaces or accessibility proxies to weight background sampling (Barber et al. 2022).

For our GBIF data, a Gaussian kernel density estimation (KDE) (Tong et al. 2023) is applied to all wild boar records of that particular season. This layer is denoting sampling density and is then multiplied by the Human Footprint Index (HFI) (Sanderson et al. 2002) to account for human accessibility (Hughes et al. 2021). The HFI is an ~1 km² GSD global composite of human population density, LU, infrastructure and serves as a measure of human presence. We assume that a higher HFI (more observers, roads, settlements) correlates with higher likelihood of reporting a sighting to the GBIF database (Gallardo et al. 2015; Hughes et al. 2021). We merge these two bias components (KDE density and HFI accessibility proxy) using a logit-based (sigmoid) function, where we prioritize HFI as an external density estimate by setting $w_{\mathrm{KDE}} = 1.0$, $w_{\mathrm{HFI}} = 2.0$, and define the logit function as $\mathrm{logit}(p) = \log\left(p/(1-p)\right)$:

$$\hat{b}_{GBIF}(x) = \mathrm{logit}^{-1}\left(\frac{w_{\mathrm{KDE}} \cdot \mathrm{logit}(\mathrm{KDE}(x)) + w_{\mathrm{HFI}} \cdot \mathrm{logit}(\mathrm{HFI}(x))}{w_{\mathrm{KDE}} + w_{\mathrm{HFI}}}\right)$$

The resulting GBIF bias-correction map exhibits high bias values around intensively sampled hotspots and areas of high human accessibility (urban centers, roads), while assigning lower bias weights to remote and less accessible regions.

WVCs and traffic infrastructure data are a highly valuable data source and monitoring prior (Schwartz et al. 2020). Because WVCs are exclusively reported exactly on infrastructure networks we assume that higher traffic volume generally leads to a higher likelihood of detecting collisions (Morelle et al. 2013). We acknowledge that traffic density is only one of several interacting factors influencing WVCs: Collision likelihood involves complex ecological, behavioral, and spatial dynamics, such as animal movement, habitat fragmentation, landscape connectivity, and behavioral adaptations (Benard 2023). However, given data availability constraints and the intrinsic value of these collision datasets as reliable and consistently monitored wildlife records (Schwartz et al., 2020), we believe traffic volume currently represents an accessible and quantifiable proxy to detection and reporting biases along road and rail infrastructures. The importance of traffic volume as a key determinant for wildlife collision detection is well-established (Morelle et al. 2013).

The **RRN roadkill WVC data** used in this study is exclusively collected on a limited part of the French road network: The national toll-free highways. We therefore map an elevated bias value only to pixels with a direct adjacency to the nearest RRN road vector. As these roads are centrally governed, we can leverage the openly available annual average daily traffic volume counts (*TMJA*, French Ministry for Ecological Transition, Aug 2025) for the largest parts of the network. We linearly map bias correction with a min-max stretch from 0 (lowest traffic) up to the value of 1 (full correction for the maximum traffic volume encountered).

Similarly, the **SNCF railway WVC data** bias map is based on direct pixel adjacency to railway lines, and the bias correction is linearly mapped to average annual train frequencies. Both datasets, the geographical extent of the railway network and the train frequency per section were provided by SNCF

Réseau. Traffic density for both datasets is spatially sampled in 1 km resolution where section counts are available. As WVC reports for both datasets are never placed outside of the immediate vicinity of the network vector features this alleviates the need to create additional fading buffer zones. Missing values for both datasets are interpolated using Inverse Distance Weighting (IDW) on the activated network pixels.

Consequentially we combine the GBIF, road, and rail bias raster into a single weighted composite bias grid for each season. This involves scaling each bias map to a comparable range and summing them, such that for example a cell with dual high GBIF and high road bias correction retains a large composite value. Source-specific summand weights (α, β, γ) equal the fraction of points in the specific seasonal dataset:

$$B(x) = \alpha\, \hat{b}_{\text{GBIF}}(x) + \beta\, T_{\text{road}}(x) + \gamma\, T_{\text{rail}}(x), \pi(x) \propto (1 - \rho) + \rho\, \tilde{B}(x)$$

During model initialization (background sampling), we use this composite surface as a probability weighting for creating background points with a tunable blending factor (ρ) reflecting correction strength. In effect, the background points are drawn in proportion to local bias strength, which corrects SDMs for the non-uniform detectability of wild boar across the landscape.

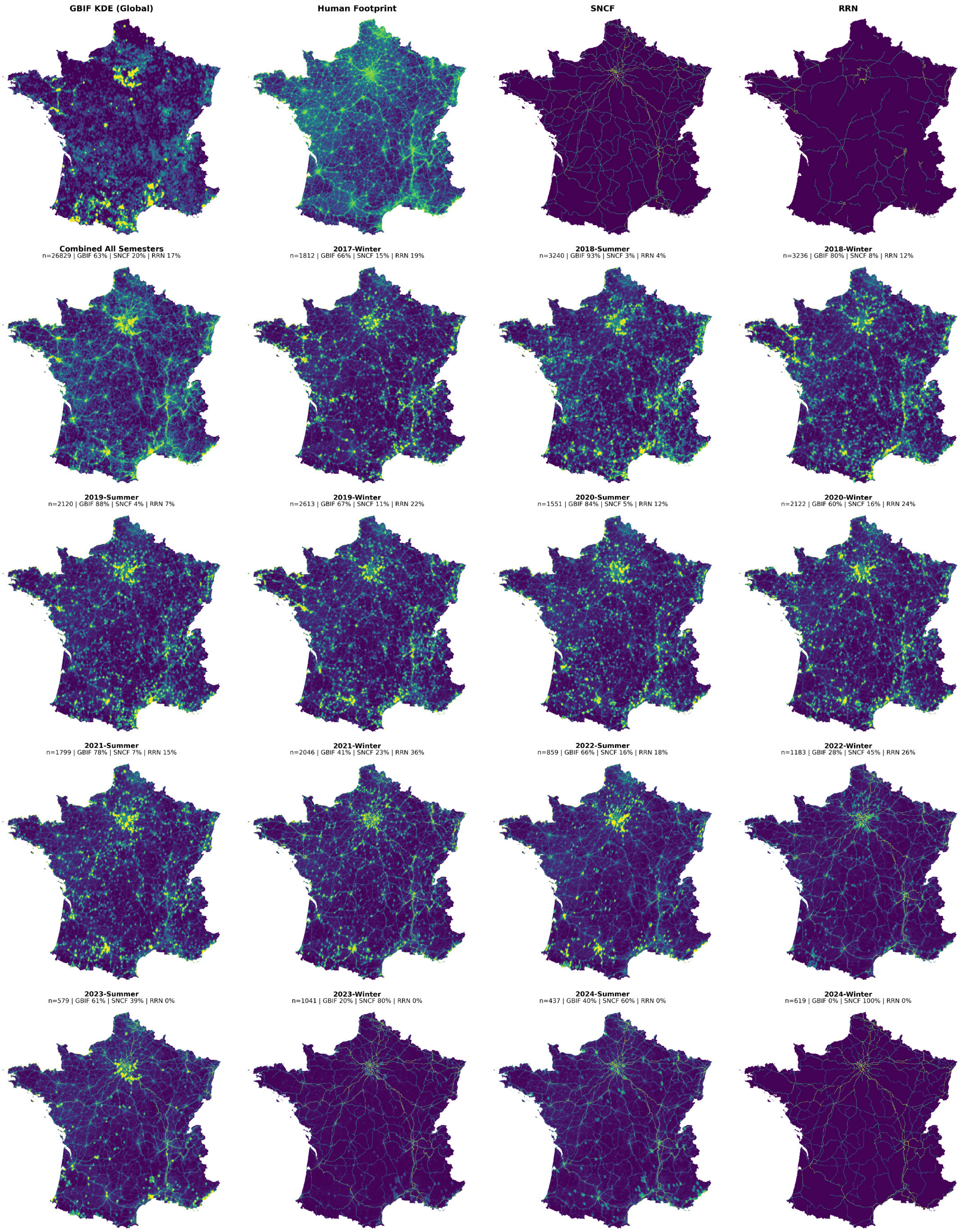


Fig. S2.1: Constructed Bias Maps.

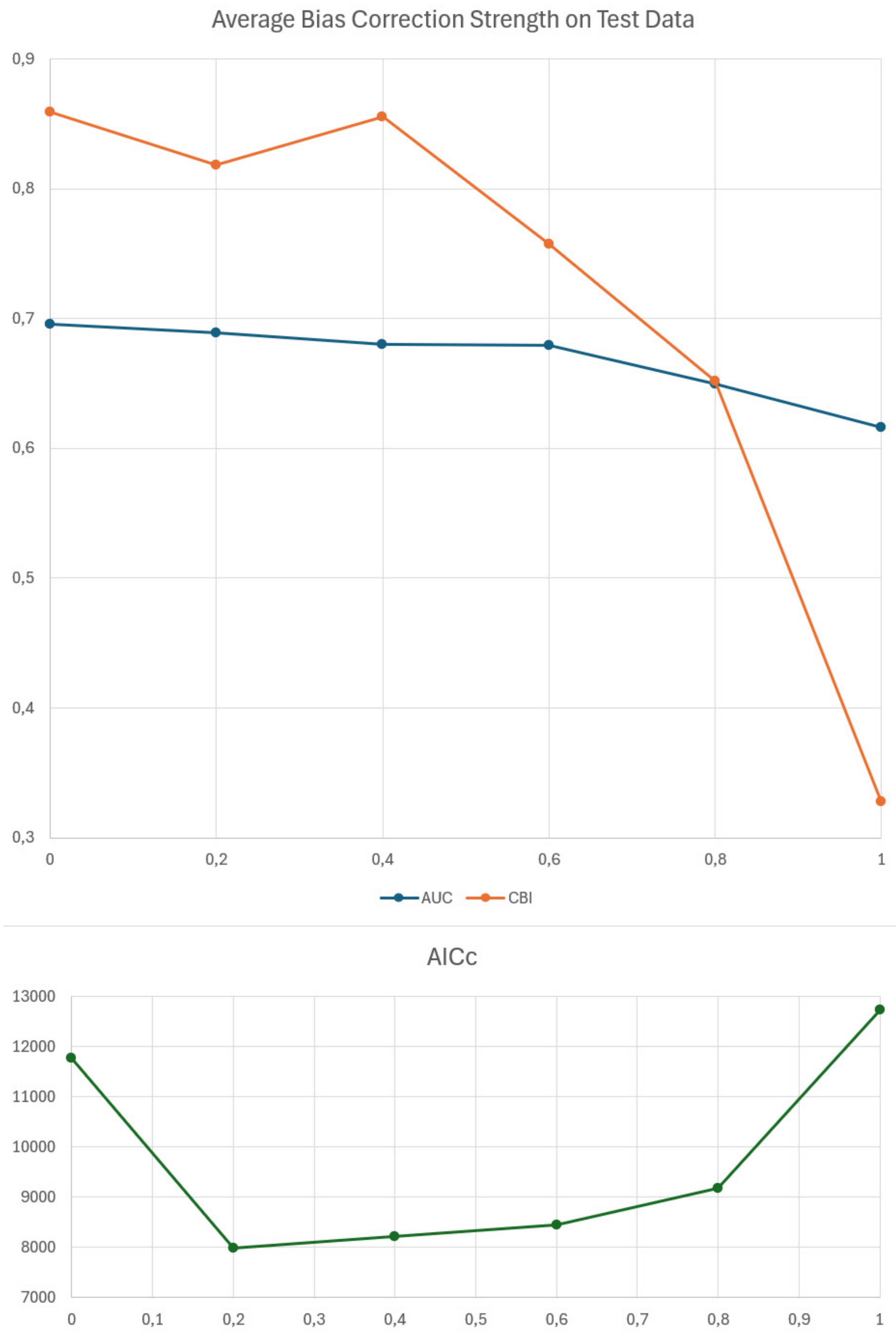


Fig. S2.2: Example Test series for tunable bias correction mixing parameter ρ.
Final choice in this case was ρ = 0.4, because of a combined low AICc and high CBI.

S3 – Vision Transformer Configuration

S3a – Ground Truth Composition

We produced annually updated land-cover maps at 30 m resolution by fine-tuning the multispectral Vision Transformer (ViT) foundation model Prithvi-100M (Jakubik et al., 2023) on harmonized Landsat-Sentinel-2 surface reflectance imagery (HLS v2) (Claverie et al. 2018). For each year, input imagery comprises three phenological mosaics (April, July, October) with six reflectance bands (R, G, B, NIR, SWIR1, SWIR2) per mosaic, yielding an 18-band multitemporal stack.

Training labels were assembled from national-scale GIS products to create reference classes aligned with wild boar resources and refuge (Brandt et al. 2006; Ballari and Barrios-García 2014). Crop labels (wheat, barley, maize, rapeseed) were sourced from the Registre Parcellaire Graphique (RPG) (Cantelaube and Carles 2014). Forest labels were derived from IGN BD Forêt v2 and augmented with forest-loss updates (Hansen et al. 2013). Permanent grasslands and additional agricultural land-use classes were integrated from annual Theia land cover products (Inglada et al. 2017; Puissant et al. 2019). Urban areas were delineated by combining Corine Land Cover and Theia layers; open water was derived from OpenStreetMap. An 'other' class captured minor land covers and no-data areas but was excluded from further processing.

Class Mapping

Ground Truth labels were generated through a deterministic multi-source raster fusion workflow (Fig. S3a.1) that harmonizes national GIS products into a common 30 m categorical grid with 12 target classes (1-4: wheat/corn/barley/rapeseed; 5: other agriculture and post-forest-loss; 6: grassland; 7-10: forest types; 11: urban; 12: water/sea).

For each year, RPG parcel data were reclassified (codes 1/2/3/5 → classes 1/2/3/4; codes 17-18 → class 6; all other codes → class 5) and rasterized.

Forest labels were derived from BDForêt by rasterizing species-based classes (oak=7, coniferous=8, mixed=9, beech=10) and by separately rasterizing “Essence NC” as grassland (class 6). A Masque Forêt layer was rasterized as mixed forest (class 9) to cover forest areas not attributed in BDForêt. Annual forest-loss rasters were produced (one per year), resampled to the 30 m grid, and used to remove lost forest pixels from both the BDForêt and Masque Forêt forest rasters; forest-loss pixels were also introduced as a dedicated class-5 layer in the final composite.

Additional agricultural and grassland labels were taken from annual Theia products (class 5 and 6), with code mappings adjusted for pre-2018 nomenclature, and resampled to 30 m.

Settlements were labeled as class 11 using CLC urbanization (static) complemented by annual Theia urbanization.

Water was labeled as class 12 by combining a regional water layer and a global sea layer (static) prior to rasterization.

The final ground-truth raster for each year was obtained by combining the prepared rasters in the explicit order shown in Fig. S3a.1 (BDForêt−loss, Masque Forêt−loss, water/sea, RPG, Theia agriculture, Theia grassland, Essence NC, loss forest, CLC urban, Theia urban).

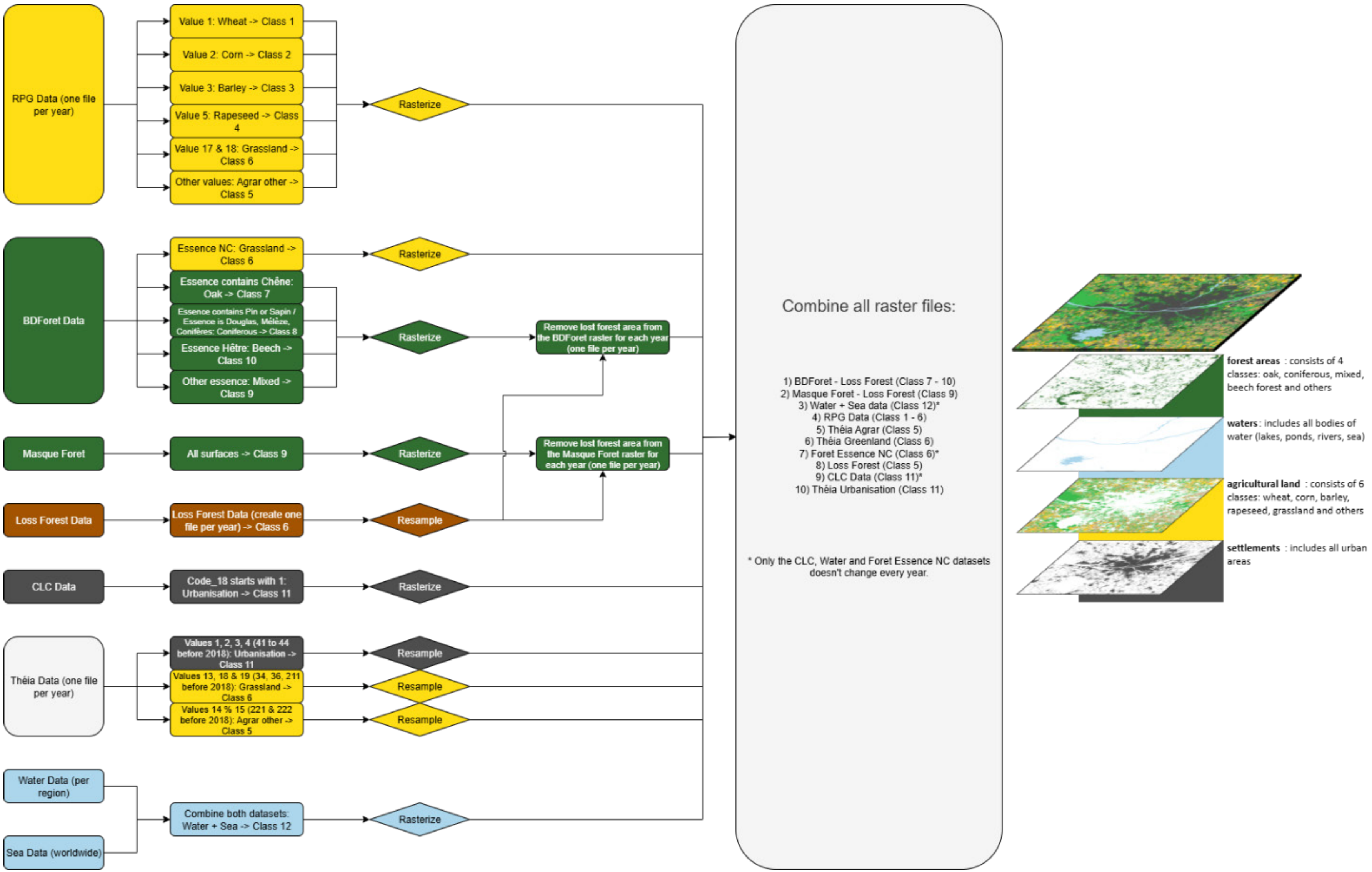


Fig. S3a.1: Annotation data provenance overview flowchart.

Tab. S3a.2: Class list.

Land-Cover Class	Scientific Names of Target Species	Ecological Relevance to Wild Boar (*Sus scrofa*)	GIS Data Sources	References
1 **Wheat**	*Triticum aestivum*	Key cultivated crop, energy-rich grains consumed during summer/autumn.	Registre Parcellaire Graphique (RPG)	(Schley and Roper 2003; Brandt et al. 2006; Cantelaube and Carles 2014)
2 **Maize (Corn)**	*Zea mays*	High-energy cultivated crop; important food from summer to early winter.	Registre Parcellaire Graphique (RPG)	(Schley and Roper 2003; Brandt et al. 2006; Ballari and Barrios-García 2014; Cantelaube and Carles 2014)
3 **Barley**	*Hordeum vulgare*	Cultivated grain; supplementary forage, mainly early summer resource.	Registre Parcellaire Graphique (RPG)	(Schley and Roper 2003; Brandt et al. 2006; Cantelaube and Carles 2014)
4 **Rapeseed**	*Brassica napus*	Energy-rich seeds; valuable nutritional resource in summer/autumn.	Registre Parcellaire Graphique (RPG)	(Schley and Roper 2003; Brandt et al. 2006; Cantelaube and Carles 2014)
5 **General Agricultural Land**	Mixed cultivated species	Rotational crops and mixed agriculture; transient seasonal forage.	Registre Parcellaire Graphique (RPG)	(Schley and Roper 2003; Brandt et al. 2006; Cantelaube and Carles 2014)
6 **Permanent Grassland**	Mixed perennial grass species	Year-round forage and shelter; complementary resource.	Theia Land Cover	(Schley and Roper 2003; Brandt et al. 2006; Inglada et al. 2017; Puissant et al. 2019)
7 **Oak-dominated Forest**	*Quercus robur*, *Q. petraea*	Acorns provide high-value mast resource; stable autumn–winter forage.	IGN BD Forêt v2; Global Forest Change	(Schley and Roper 2003; Cutini et al. 2013; Hansen et al. 2013; Bisi et al. 2018)
8 **Coniferous Forest**	*Picea abies*, *Pinus sylvestris*, *Abies alba*	Provides year-round shelter, refuge from humans, limited forage value.	IGN BD Forêt v2; Global Forest Change	(Thurfjell et al. 2009; Hansen et al. 2013; Ballari and Barrios-García 2014)
9 **Mixed Forest**	Mixed broadleaf/coniferous species	Shelter, moderate forage value; seasonally variable mast availability.	IGN BD Forêt v2; Global Forest Change	(Cutini et al. 2013; Hansen et al. 2013; Ballari and Barrios-García 2014)
10 **Beech-dominated Forest**	*Fagus sylvatica*	Critical autumn mast resource (beech nuts); rapidly deteriorate after autumn.	IGN BD Forêt v2; Global Forest Change	(Cutini et al. 2013; Hansen et al. 2013; Ballari and Barrios-García 2014)
11 **Urban and Built-up Areas**	Anthropogenic structures	Provide supplemental food and shelter; urban-tolerant boar groups documented.	Corine Land Cover; Theia Land Cover	(Cahill et al. 2012; Inglada et al. 2017; Stillfried et al. 2017; Marin et al. 2024)
12 **Open Water Surfaces**	Lakes, rivers, reservoirs	Critical drinking water source; thermoregulation, associated wetland habitats.	OpenStreetMap	(Giménez-Anaya et al. 2008; Barasona et al. 2021)
13 **Other (excluded)**	Rocky areas, beaches, no-data zones	Excluded due to limited ecological relevance for wild boar.	(Excluded from further processing)	—

S3b – Model Training

We fine-tuned the model using 224×224-pixel tiles sampled across 2017-2023 with the MMSegmentation framework (MMSegmentation Contributors, 2020) on an NVIDIA RTX 3090 GPU (24 GB VRAM), training for 100-120 epochs until convergence. The resulting 30 m maps were aggregated to the 1 km grid as fractional cover per class and complemented with distance-to-class layers for edge/proximity effects.

Tab. S3b.1: Hyperparameter Configuration File (python, mmsegmentation) for training and inference.

```
dist_params = dict(backend='nccl')
log_level = 'INFO'
load_from = None
resume_from = None
cudnn_benchmark = True
custom_imports = dict(imports=['geospatial_fm'])
num_frames = 3
img_size = 224
num_workers = 2
pretrained_weights_path = './multi-temporal-crop-classification/Prithvi_100M.pt'
num_layers = 6
patch_size = 16
embed_dim = 768
num_heads = 8
tubelet_size = 1
max_epochs = 100
eval_epoch_interval = 1
loss_weights_multi = ([
    0.84986144, 1.4526042, 2.1749167, 3.180532, 0.5387366, 0.3930211,
    1.3835843, 1.2408366, 0.3509285, 6.4880056, 1.1104473, 2.2651799, 11.662636
], )
loss_func = dict(
    type='CrossEntropyLoss',
    use_sigmoid=False,
    class_weight=([
        0.84986144, 1.4526042, 2.1749167, 3.180532, 0.5387366, 0.3930211,
        1.3835843, 1.2408366, 0.3509285, 6.4880056, 1.1104473, 2.2651799,
        11.662636
    ], ),
    avg_non_ignore=True)
output_embed_dim = 2304
experiment = 'exp_12_2017-2023'
project_dir = 'Finetuning'
work_dir = 'Finetuning\exp_12_2017-2023'
save_path = 'Finetuning\exp_12_2017-2023'
dataset_type = 'GeospatialDataset'
data_root = 'C:/Users/ady/Documents/05_KI-Daten/multi-temporal-crop-classification/hls-foundation-os/multi-temporal-crop-
classification'
img_norm_cfg = dict(
    means=[
        458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449, 1160.1824,
        410.4749, 725.431, 736.8183, 3288.6973, 2219.1045, 1255.3219, 374.5087,
        649.1173, 665.73065, 2633.4585, 1988.4376, 1196.2112
    ],
    stds=[
        832.50885, 861.7658, 904.2468, 1183.4471, 768.0135, 681.59894,
        355.74118, 424.48578, 594.4535, 921.59155, 883.3909, 706.4679,
        378.2199, 441.80365, 557.4787, 927.7923, 878.57825, 737.9002
    ])
bands = [0, 1, 2, 3, 4, 5]
tile_size = 224
orig_nsize = 512
crop_size = (224, 224)
train_pipeline = [
    dict(type='LoadGeospatialImageFromFile', to_float32=True),
    dict(type='LoadGeospatialAnnotations', reduce_zero_label=True),
    dict(type='RandomFlip', prob=0.5),
    dict(type='ToTensor', keys=['img', 'gt_semantic_seg']),
    dict(type='TorchPermute', keys=['img'], order=(2, 0, 1)),
    dict(
        type='TorchNormalize',
        means=[
            458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449, 1160.1824,
            410.4749, 725.431, 736.8183, 3288.6973, 2219.1045, 1255.3219,
            374.5087, 649.1173, 665.73065, 2633.4585, 1988.4376, 1196.2112
        ],
        stds=[
            832.50885, 861.7658, 904.2468, 1183.4471, 768.0135, 681.59894,
            355.74118, 424.48578, 594.4535, 921.59155, 883.3909, 706.4679,
            378.2199, 441.80365, 557.4787, 927.7923, 878.57825, 737.9002
        ]),
    dict(type='TorchRandomCrop', crop_size=(224, 224)),
    dict(type='Reshape', keys=['img'], new_shape=(6, 3, 224, 224)),
    dict(type='Reshape', keys=['gt_semantic_seg'], new_shape=(1, 224, 224)),
    dict(
        type='CastTensor',
        keys=['gt_semantic_seg'],
        new_type='torch.LongTensor'),
    dict(type='Collect', keys=['img', 'gt_semantic_seg'])
]
test_pipeline = [
    dict(type='LoadGeospatialImageFromFile', to_float32=True),
    dict(type='ToTensor', keys=['img']),
    dict(type='TorchPermute', keys=['img'], order=(2, 0, 1)),
    dict(
        type='TorchNormalize',
        means=[
            458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449, 1160.1824,
            410.4749, 725.431, 736.8183, 3288.6973, 2219.1045, 1255.3219,
            374.5087, 649.1173, 665.73065, 2633.4585, 1988.4376, 1196.2112
```

```
        ],
        stds=[
            832.50885, 861.7658, 904.2468, 1183.4471, 768.0135, 681.59894,
            355.74118, 424.48578, 594.4535, 921.59155, 883.3909, 706.4679,
            378.2199, 441.80365, 557.4787, 927.7923, 878.57825, 737.9002
        ]),
    dict(
        type='Reshape',
        keys=['img'],
        new_shape=(6, 3, -1, -1),
        look_up=dict({
            '2': 1,
            '3': 2
        })),
    dict(type='CastTensor', keys=['img'], new_type='torch.FloatTensor'),
    dict(
        type='CollectTestList',
        keys=['img'],
        meta_keys=[
            'img_info', 'seg_fields', 'img_prefix', 'seg_prefix', 'filename',
            'ori_filename', 'img', 'img_shape', 'ori_shape', 'pad_shape',
            'scale_factor', 'img_norm_cfg'
        ])
]
CLASSES = ('Wheat', 'Cornmaize', 'Barley', 'Rapeseed', 'Agriculture_Other',
           'Greenland', 'Oak_Grove', 'Coniferous_Forest', 'Mixed_Forest',
           'Beech_Grove', 'Urbanisation', 'Water', 'Other')
dataset = 'GeospatialDataset'
data = dict(
    samples_per_gpu=4,
    workers_per_gpu=2,
    train=dict(
        type='GeospatialDataset',
        CLASSES=('Wheat', 'Cornmaize', 'Barley', 'Rapeseed',
                 'Agriculture_Other', 'Greenland', 'Oak_Grove',
                 'Coniferous_Forest', 'Mixed_Forest', 'Beech_Grove',
                 'Urbanisation', 'Water', 'Other'),
        reduce_zero_label=True,
        data_root=
        './multi-temporal-crop-classification/hls-foundation-os/multi-temporal-crop-classification',
        img_dir='train',
        ann_dir='train',
        pipeline=[
            dict(type='LoadGeospatialImageFromFile', to_float32=True),
            dict(type='LoadGeospatialAnnotations', reduce_zero_label=True),
            dict(type='RandomFlip', prob=0.5),
            dict(type='ToTensor', keys=['img', 'gt_semantic_seg']),
            dict(type='TorchPermute', keys=['img'], order=(2, 0, 1)),
            dict(
                type='TorchNormalize',
                means=[
                    458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449,
                    1160.1824, 410.4749, 725.431, 736.8183, 3288.6973,
                    2219.1045, 1255.3219, 374.5087, 649.1173, 665.73065,
                    2633.4585, 1988.4376, 1196.2112
                ],
                stds=[
                    832.50885, 861.7658, 904.2468, 1183.4471, 768.0135,
                    681.59894, 355.74118, 424.48578, 594.4535, 921.59155,
                    883.3909, 706.4679, 378.2199, 441.80365, 557.4787,
                    927.7923, 878.57825, 737.9002
                ]),
            dict(type='TorchRandomCrop', crop_size=(224, 224)),
            dict(type='Reshape', keys=['img'], new_shape=(6, 3, 224, 224)),
            dict(
                type='Reshape',
                keys=['gt_semantic_seg'],
                new_shape=(1, 224, 224)),
            dict(
                type='CastTensor',
                keys=['gt_semantic_seg'],
                new_type='torch.LongTensor'),
            dict(type='Collect', keys=['img', 'gt_semantic_seg'])
        ],
        img_suffix='_merged.tif',
        seg_map_suffix='.mask.tif'),
    val=dict(
        type='GeospatialDataset',
        CLASSES=('Wheat', 'Cornmaize', 'Barley', 'Rapeseed',
                 'Agriculture_Other', 'Greenland', 'Oak_Grove',
                 'Coniferous_Forest', 'Mixed_Forest', 'Beech_Grove',
                 'Urbanisation', 'Water', 'Other'),
        reduce_zero_label=True,
        data_root=
        'C:/Users/ady/Documents/05_KI-Daten/multi-temporal-crop-classification/hls-foundation-os/multi-temporal-crop-
classification',
        img_dir='val',
        ann_dir='val',
        pipeline=[
            dict(type='LoadGeospatialImageFromFile', to_float32=True),
            dict(type='ToTensor', keys=['img']),
            dict(type='TorchPermute', keys=['img'], order=(2, 0, 1)),
            dict(
                type='TorchNormalize',
                means=[
                    458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449,
                    1160.1824, 410.4749, 725.431, 736.8183, 3288.6973,
                    2219.1045, 1255.3219, 374.5087, 649.1173, 665.73065,
                    2633.4585, 1988.4376, 1196.2112
                ],
                stds=[
                    832.50885, 861.7658, 904.2468, 1183.4471, 768.0135,
                    681.59894, 355.74118, 424.48578, 594.4535, 921.59155,
                    883.3909, 706.4679, 378.2199, 441.80365, 557.4787,
```

```
                927.7923, 878.57825, 737.9002
            ]),
        dict(
            type='Reshape',
            keys=['img'],
            new_shape=(6, 3, -1, -1),
            look_up=dict({
                '2': 1,
                '3': 2
            })),
        dict(
            type='CastTensor', keys=['img'], new_type='torch.FloatTensor'),
        dict(
            type='CollectTestList',
            keys=['img'],
            meta_keys=[
                'img_info', 'seg_fields', 'img_prefix', 'seg_prefix',
                'filename', 'ori_filename', 'img', 'img_shape',
                'ori_shape', 'pad_shape', 'scale_factor', 'img_norm_cfg'
            ])
    ],
    img_suffix='_merged.tif',
    seg_map_suffix='.mask.tif'),
test=dict(
    type='GeospatialDataset',
    CLASSES=('Wheat', 'Cornmaize', 'Barley', 'Rapeseed',
             'Agriculture_Other', 'Greenland', 'Oak_Grove',
             'Coniferous_Forest', 'Mixed_Forest', 'Beech_Grove',
             'Urbanisation', 'Water', 'Other'),
    reduce_zero_label=True,
    data_root=
    './multi-temporal-crop-classification/hls-foundation-os/multi-temporal-crop-classification',
    img_dir='test',
    ann_dir='test',
    pipeline=[
        dict(type='LoadGeospatialImageFromFile', to_float32=True),
        dict(type='ToTensor', keys=['img']),
        dict(type='TorchPermute', keys=['img'], order=(2, 0, 1)),
        dict(
            type='TorchNormalize',
            means=[
                458.82156, 763.18054, 719.3139, 3070.6416, 1914.1449,
                1160.1824, 410.4749, 725.431, 736.8183, 3288.6973,
                2219.1045, 1255.3219, 374.5087, 649.1173, 665.73065,
                2633.4585, 1988.4376, 1196.2112
            ],
            stds=[
                832.50885, 861.7658, 904.2468, 1183.4471, 768.0135,
                681.59894, 355.74118, 424.48578, 594.4535, 921.59155,
                883.3909, 706.4679, 378.2199, 441.80365, 557.4787,
                927.7923, 878.57825, 737.9002
            ]),
        dict(
            type='Reshape',
            keys=['img'],
            new_shape=(6, 3, -1, -1),
            look_up=dict({
                '2': 1,
                '3': 2
            })),
        dict(
            type='CastTensor', keys=['img'], new_type='torch.FloatTensor'),
        dict(
            type='CollectTestList',
            keys=['img'],
            meta_keys=[
                'img_info', 'seg_fields', 'img_prefix', 'seg_prefix',
                'filename', 'ori_filename', 'img', 'img_shape',
                'ori_shape', 'pad_shape', 'scale_factor', 'img_norm_cfg'
            ])
    ],
    img_suffix='_merged.tif',
    seg_map_suffix='.mask.tif'))
optimizer = dict(type='Adam', lr=3e-05, betas=(0.9, 0.999), weight_decay=0.01)
optimizer_config = dict(grad_clip=None)
lr_config = dict(
    policy='poly',
    warmup='linear',
    warmup_iters=1500,
    warmup_ratio=1e-06,
    power=1.0,
    min_lr=1e-06,
    by_epoch=False)
log_config = dict(
    interval=10,
    hooks=[dict(type='TextLoggerHook'),
           dict(type='TensorboardLoggerHook')])
checkpoint_config = dict(
    by_epoch=True, interval=20, out_dir='Finetuning\exp_12_2017-2023')
evaluation = dict(
    interval=1, metric='mIoU', pre_eval=True, save_best='mIoU', by_epoch=True)
reduce_train_set = dict(reduce_train_set=False)
reduce_factor = dict(reduce_factor=1)
runner = dict(type='EpochBasedRunner', max_epochs=100)
workflow = [('train', 1)]
norm_cfg = dict(type='BN', requires_grad=True)
model = dict(
    type='TemporalEncoderDecoder',
    frozen_backbone=False,
    backbone=dict(
        type='TemporalViTEncoder',
        pretrained=
        './multi-temporal-crop-classification/Prithvi_100M.pt',
        img_size=224,
```

```
        patch_size=16,
        num_frames=3,
        tubelet_size=1,
        in_chans=6,
        embed_dim=768,
        depth=6,
        num_heads=8,
        mlp_ratio=4.0,
        norm_pix_loss=False),
    neck=dict(
        type='ConvTransformerTokensToEmbeddingNeck',
        embed_dim=2304,
        output_embed_dim=2304,
        drop_cls_token=True,
        Hp=14,
        Wp=14),
    decode_head=dict(
        num_classes=13,
        in_channels=2304,
        type='FCNHead',
        in_index=-1,
        channels=256,
        num_convs=1,
        concat_input=False,
        dropout_ratio=0.1,
        norm_cfg=dict(type='BN', requires_grad=True),
        align_corners=False,
        loss_decode=dict(
            type='CrossEntropyLoss',
            use_sigmoid=False,
            class_weight=([
                0.84986144, 1.4526042, 2.1749167, 3.180532, 0.5387366,
                0.3930211, 1.3835843, 1.2408366, 0.3509285, 6.4880056,
                1.1104473, 2.2651799, 11.662636
            ], ),
            avg_non_ignore=True)),
    auxiliary_head=dict(
        num_classes=13,
        in_channels=2304,
        type='FCNHead',
        in_index=-1,
        channels=256,
        num_convs=2,
        concat_input=False,
        dropout_ratio=0.1,
        norm_cfg=dict(type='BN', requires_grad=True),
        align_corners=False,
        loss_decode=dict(
            type='CrossEntropyLoss',
            use_sigmoid=False,
            class_weight=([
                0.84986144, 1.4526042, 2.1749167, 3.180532, 0.5387366,
                0.3930211, 1.3835843, 1.2408366, 0.3509285, 6.4880056,
                1.1104473, 2.2651799, 11.662636
            ], ),
            avg_non_ignore=True)),
    train_cfg=dict(),
    test_cfg=dict(mode='slide', stride=(112, 112), crop_size=(224, 224)))
auto_resume = False
gpu_ids = range(0, 1)


# ---- fix int8 overflow: use -1 as ignore index ----
ignore_index = -1

# Update heads to use -1 as ignore_index
model['decode_head']['ignore_index'] = ignore_index
model['auxiliary_head']['ignore_index'] = ignore_index

# ONLY set dataset-level ignore_index; pipeline ops don't support it directly.
for split in ['train', 'val', 'test']:
    data[split]['ignore_index'] = ignore_index
# --------------------------------------------------
```

S3c – Vision Transformer Results

Label Consistency

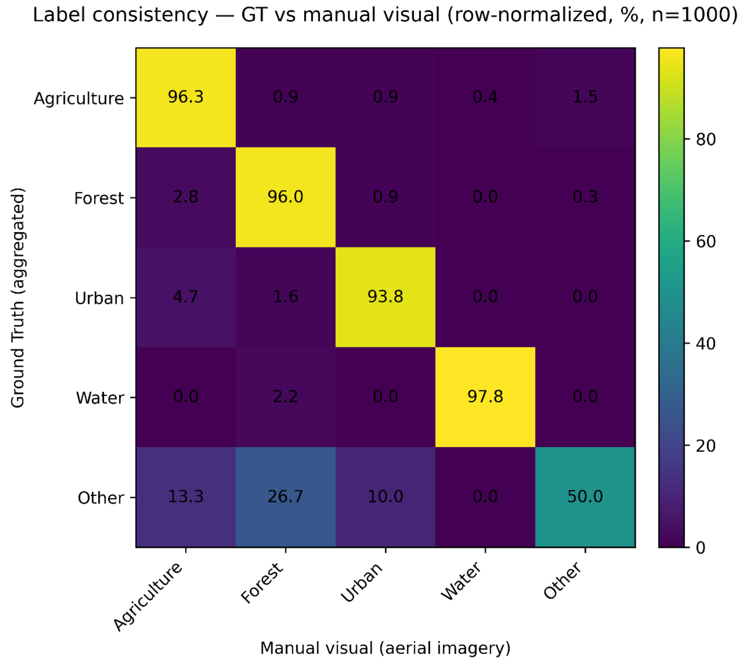


Fig. S3c.1: Domain-pooled confusion matrix manually validating the training labels against aerial imagery on 1000 points.

Manual Validation of Predictions

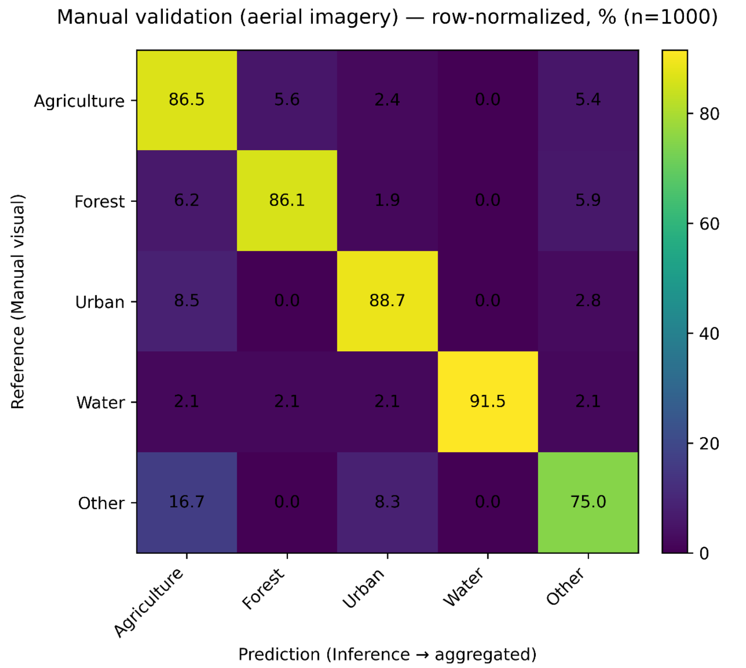


Fig. S3c.2: Domain-pooled confusion matrix validating predictions against aerial imagery on 1000 points.

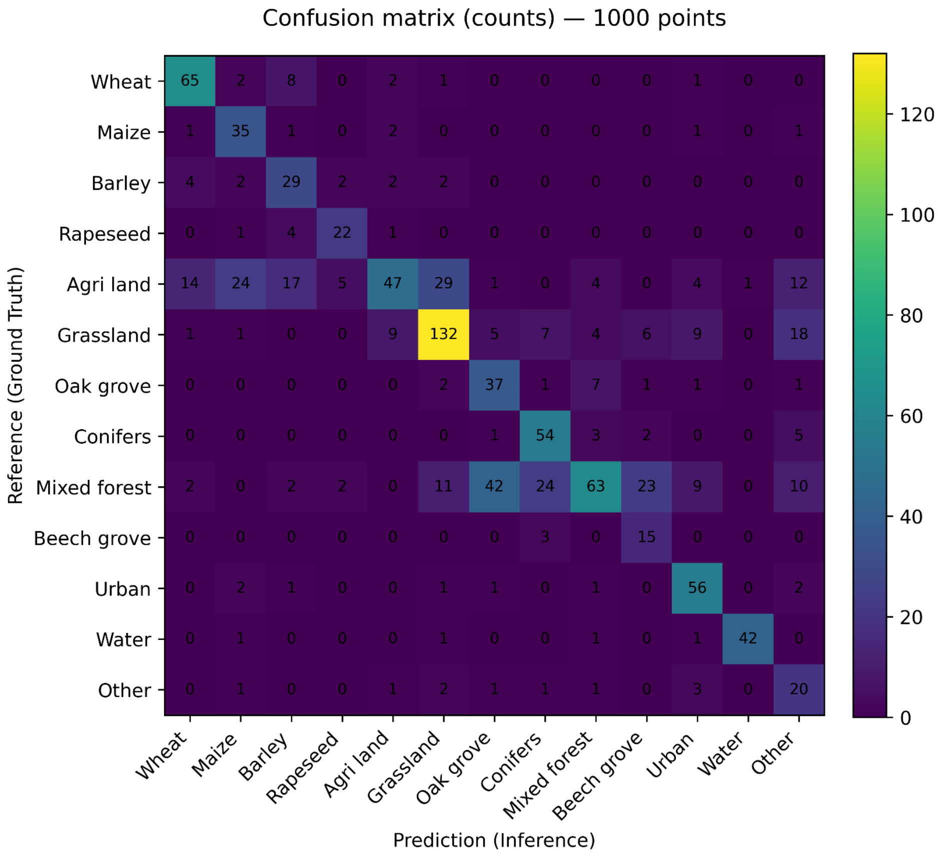


Fig. S3c.3: Absolute confusion matrix validating predictions against training labels on the 1000 interpreted points.

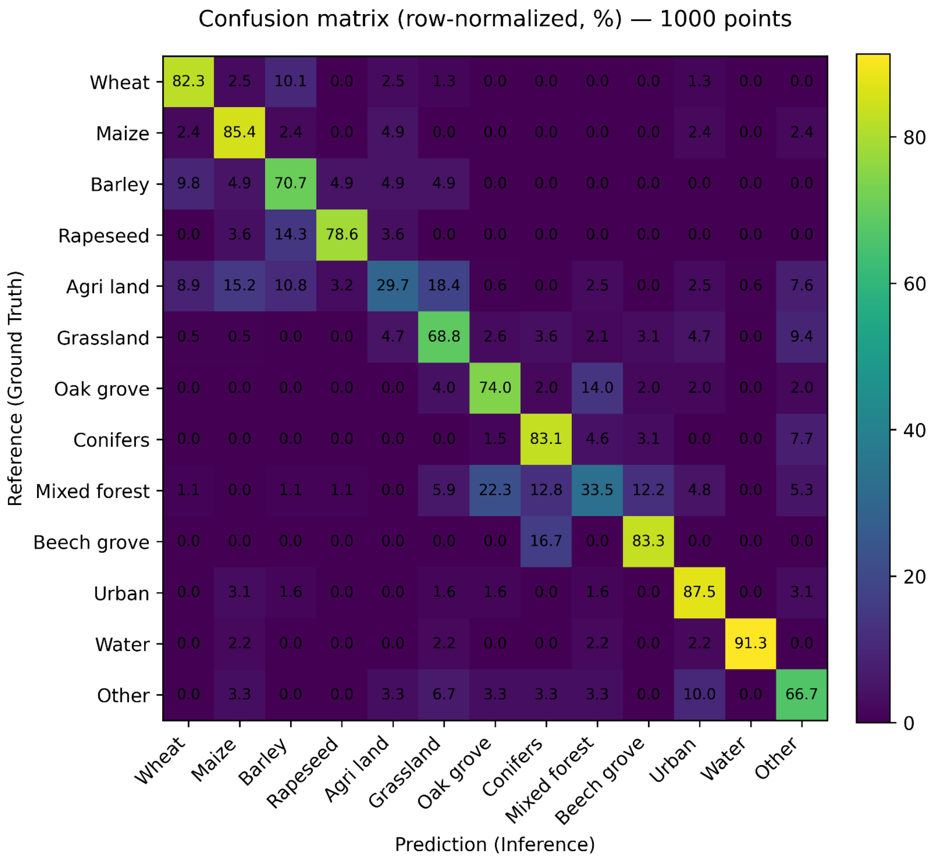


Fig. S3c.4: Row-normalized confusion matrix validating predictions against training labels on the 1000 interpreted points.

Pixelwise Confusion Matrices

Fig. S3c.5: Pixelwise row-normalized confusion matrix validating predictions against training labels on test data.

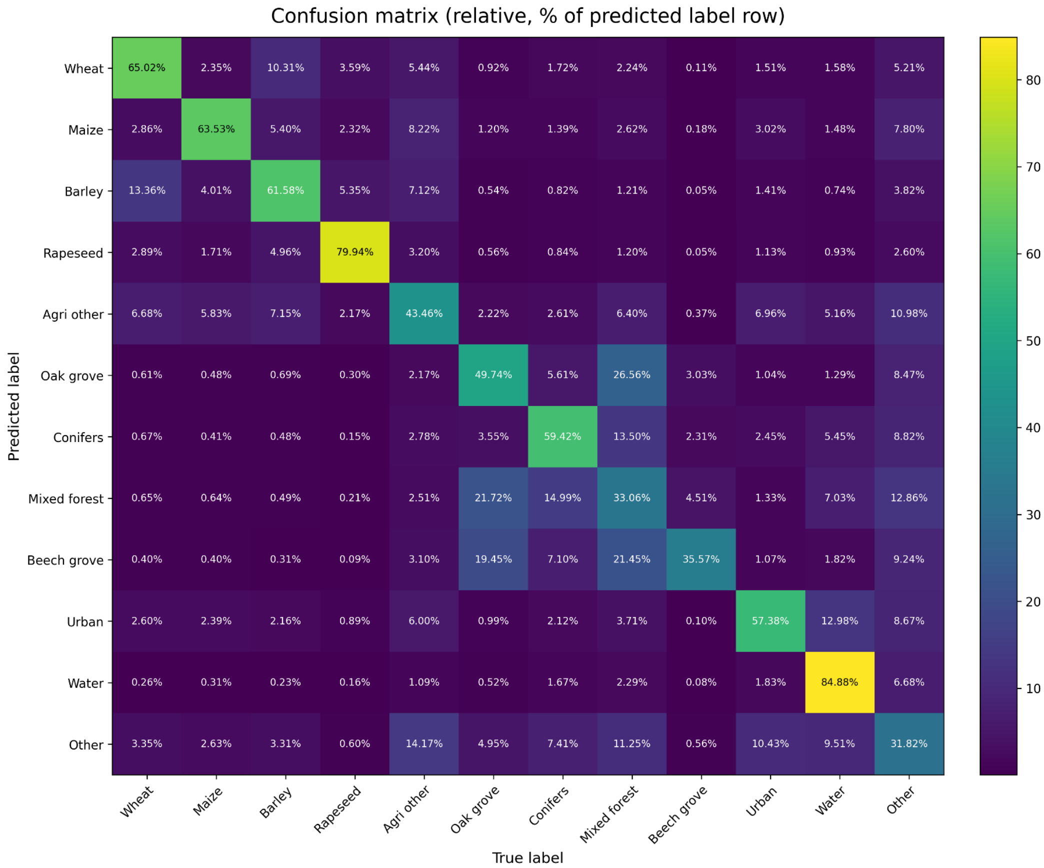


Fig. S3c.6: Pixelwise column-normalized confusion matrix validating predictions against training labels on test data.

S4 – Environmental SDM Features

Climate and Weather Predictors
Climate influences wild boar populations both directly through physiology and movement (Sabrina et al. 2009; Bergqvist et al. 2018; Vetter et al. 2020) and indirectly via vegetation dynamics such as masting or drought (Bruinderink and Hazebroek 1995; Cutini et al. 2013; Bisi et al. 2018). We derived two complementary climate datasets, dynamic ERA-5 anomalies with stable WorldClim normals, each filtered for collinearity with variance-inflation-factor (VIF) analysis: (i) Dynamic seasonal variables came from ERA-5 monthly reanalysis at ~30km GSD (Muñoz-Sabater et al. 2021), aggregated into composites relevant to ungulate ecology (Couriot et al. 2023). We include mean temperature and precipitation as well as surface solar radiation and top-layer soil moisture. For winter (Sep-Feb), we added absolute seasonal mean snow depth, minimum air and soil temperature to capture fundamental constraints such as thermal stress, snow-induced mobility limitations, and frozen soils restricting rooting and digging activity. (ii) Static climatic baselines were taken from WorldClim 2 (~800 m GSD, 1970-2000) describing mean annual temperature, temperature seasonality, and precipitation metrics. These provided a high-resolution representation of microclimatic regimes relevant to wild boar ecological requirements (Enetwild Consortium et al. 2019; Lee et al. 2022; James et al. 2023), such as mean annual temperature, temperature seasonality, annual precipitation, and precipitation of the driest quarter.

Digital Elevation Model
We incorporate topographic variables derived from the *SRTM DEM* digital elevation model (USGS 2018) – elevation, slope, and aspect – to account for terrain preferences. Wild boar are affected by landscape structure (Acevedo et al. 2006), even though slope and moderate altitude might not restrict their activity (Abaigar et al. 1994) and tolerance limits might be expanding (Bosch et al. 2020) we hypothesize an impact on habitat selection.

Auxiliary Ecological LU/LC Datasets
(i) To represent **landscape fragmentation**, we include the *Effective Mesh Density Landscape Fragmentation Index* seff (Dataset FGA2-S 2021) (EEA/FOEN 2011), resampled from its original 10 m resolution, capturing both potential barriers or corridors influencing wild boar movement (Thurfjell et al. 2009). Fragmentation may also increase the suitability of a habitat secondarily through anthropogenic food sources (Castillo-Contreras et al. 2018).
(ii) We also integrate the Copernicus European ***small woody features*** dataset (Faucqueur et al. 2019), providing fractional cover of hedgerows and shrublands, valuable as corridors, shelter, and supplementary feeding areas (Welander 2000; Ficetola et al. 2014).
(iii) Additional specialized LC layers include fractional coverage of **wetland areas** (“*zones humides”*) (Rapinel et al. 2023) derived from the French national environmental database Inventaire National du Patrimoine Naturel (INPN), which serve as important habitat refuges and seasonal resources especially for mediterranean populations (Rosell et al. 2004, 2012; Vanschoenwinkel et al. 2008; Giménez-Anaya et al. 2008; Barasona et al. 2021).
(iv) From the French Theia maps, we included fractional cover of specific **ecologically relevant LU** categories not detailed in our own LC analysis: Vineyards (Fournier-Chambrillon et al. 1995; Calenge et al. 2004), natural heathland (“*prairies*”) (Bruinderink and Hazebroek 1995; Leaper et al. 1999), fruit orchards (Schley and Roper 2003), as well as highly dense urban fabric and industrial zones (since we did not differentiate further between LU intensity categories of urbanization in our own LC maps).

Hunting Bag
We incorporate direct anthropogenic influence by integrating wild boar harvest resampled at the regional scale, derived from département-level hunting bag statistics corrected for reported absence by commune made publicly available by the Office Français de la Biodiversité (OFB). This provides a spatially explicit measure of mortality due to hunting as well as an opportunity to integrate behavioral shifts due to hunting activity (Calenge et al. 2002; Keuling et al. 2008; Thurfjell et al. 2013). Hunting predictor features are only activated in Winter models.

All predictors were harmonized to a 1 km grid for seamless integration.

Feature Correlation Analysis

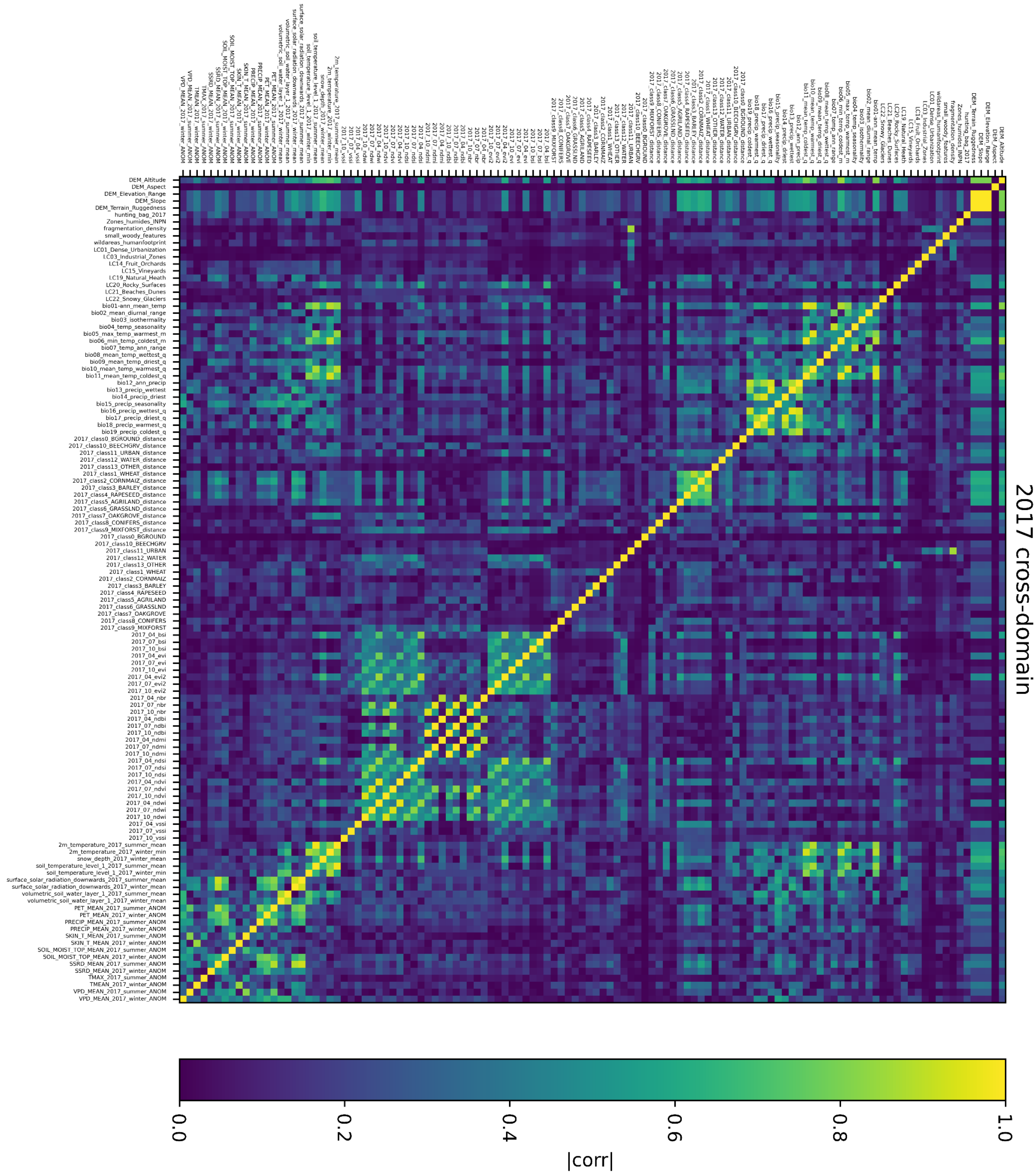


Fig. S4.1: Feature Correlation Matrix for 2017.

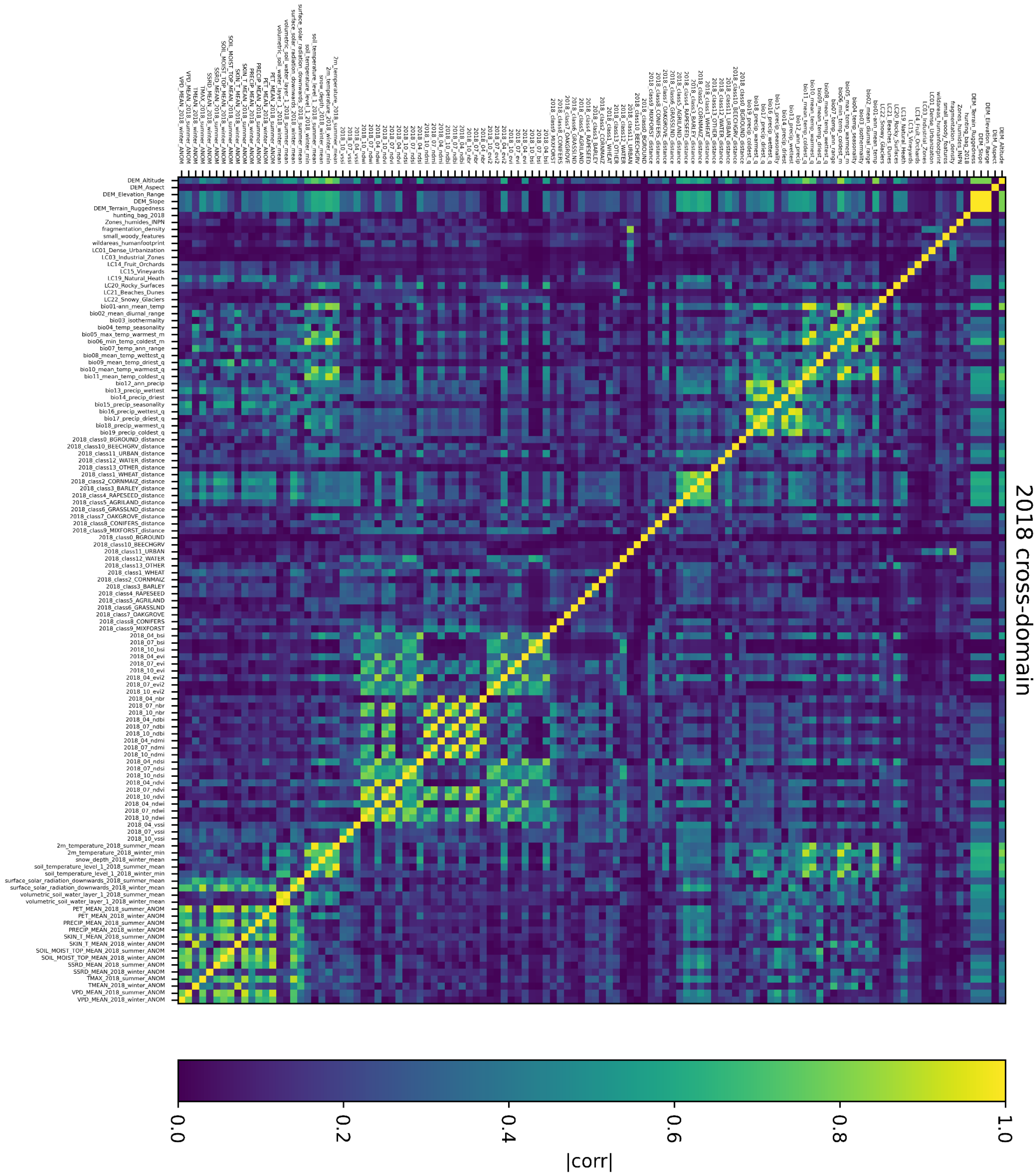


Fig. S4.2: Feature Correlation Matrix for 2018.

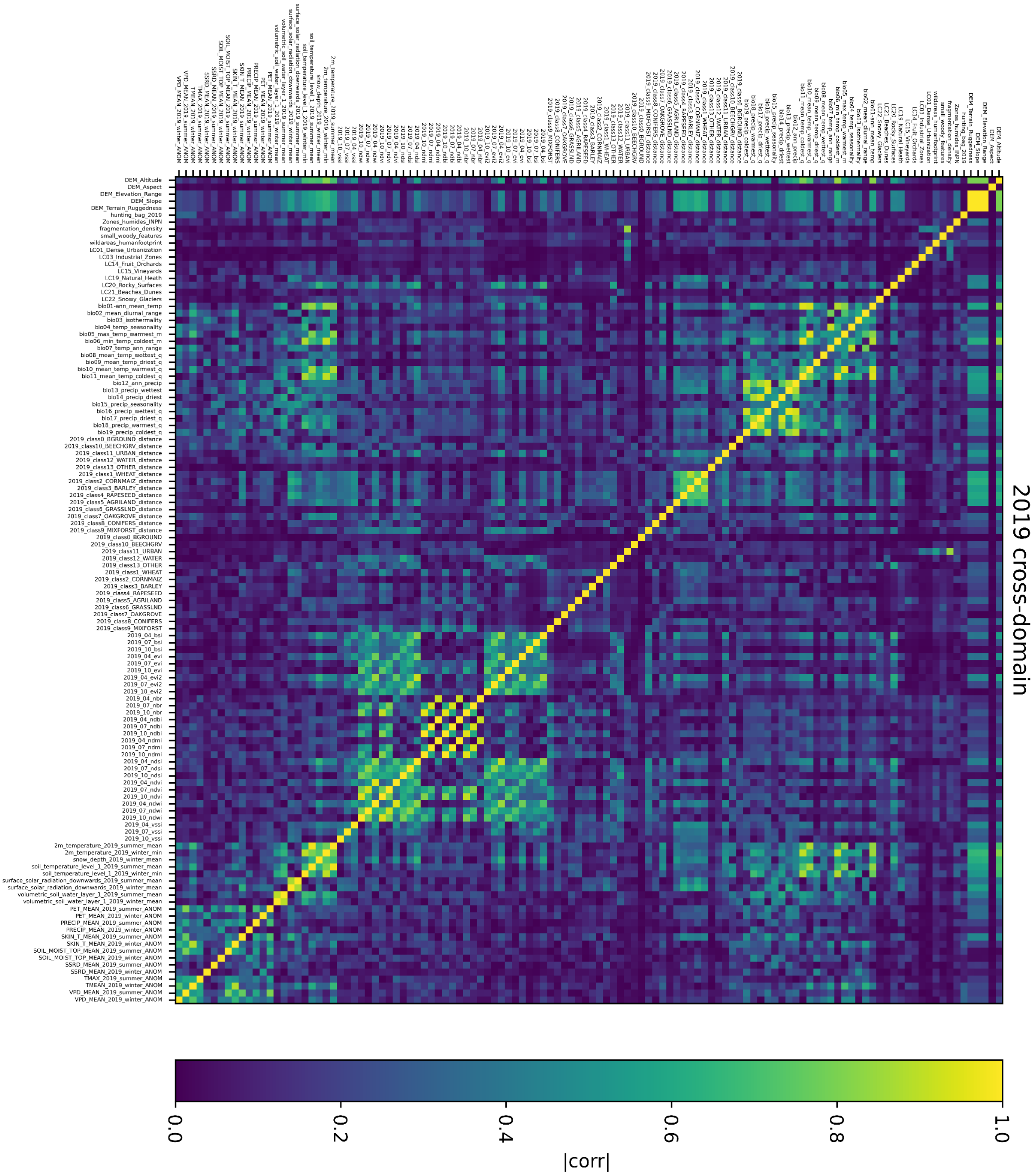


Fig. S4.3: Feature Correlation Matrix for 2019.

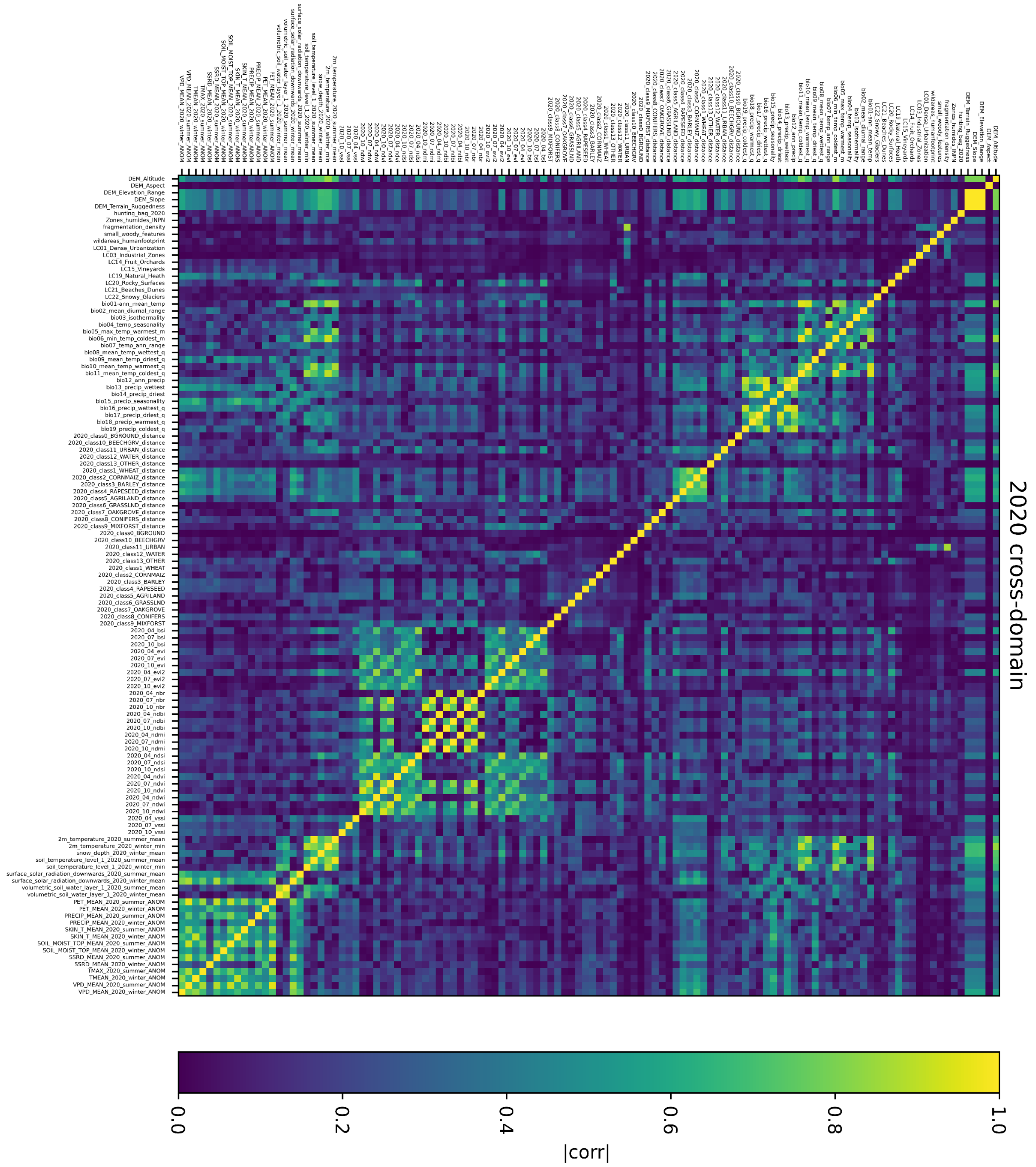


Fig. S4.4: Feature Correlation Matrix for 2020.

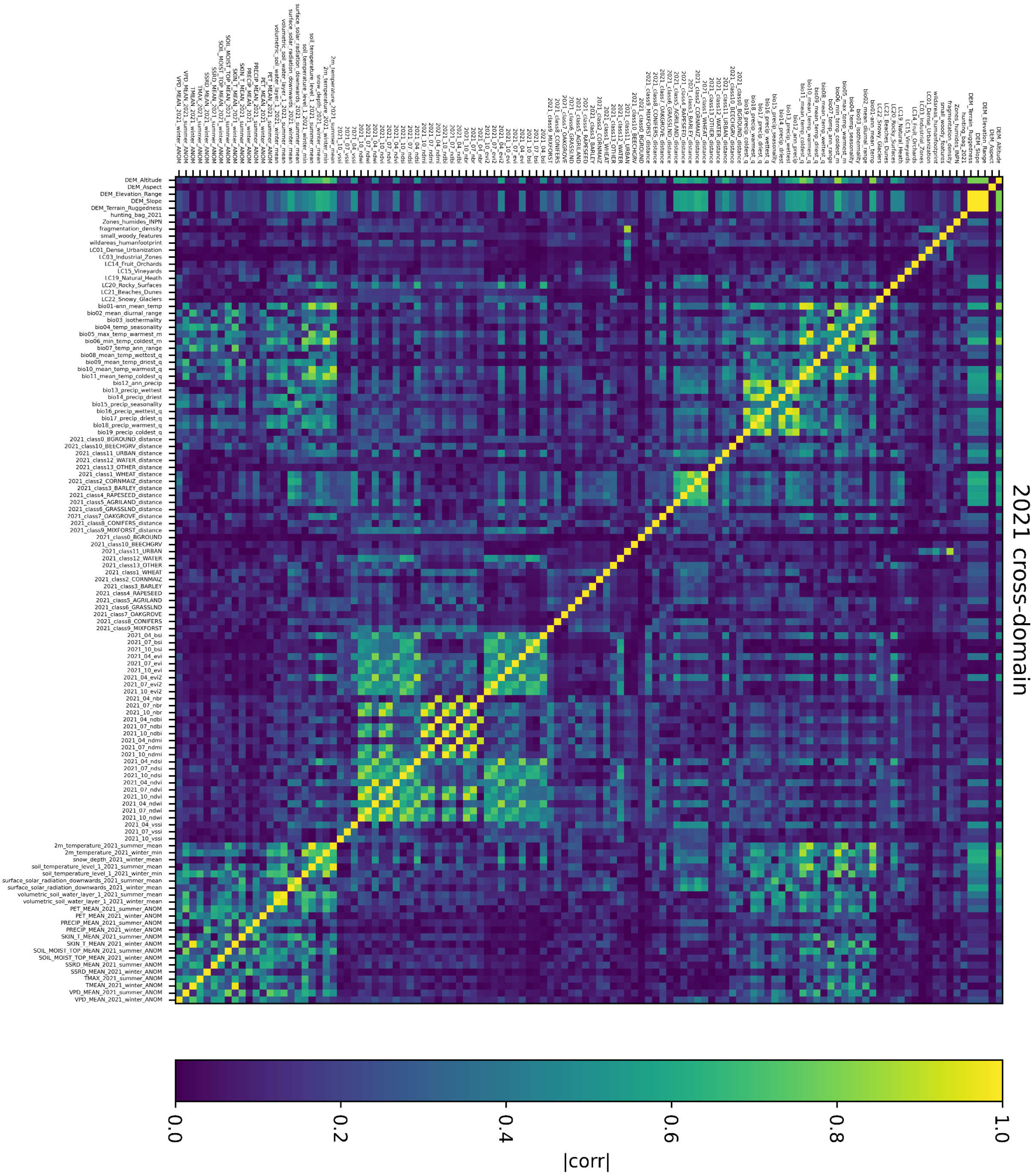


Fig. S4.5: Feature Correlation Matrix for 2021.

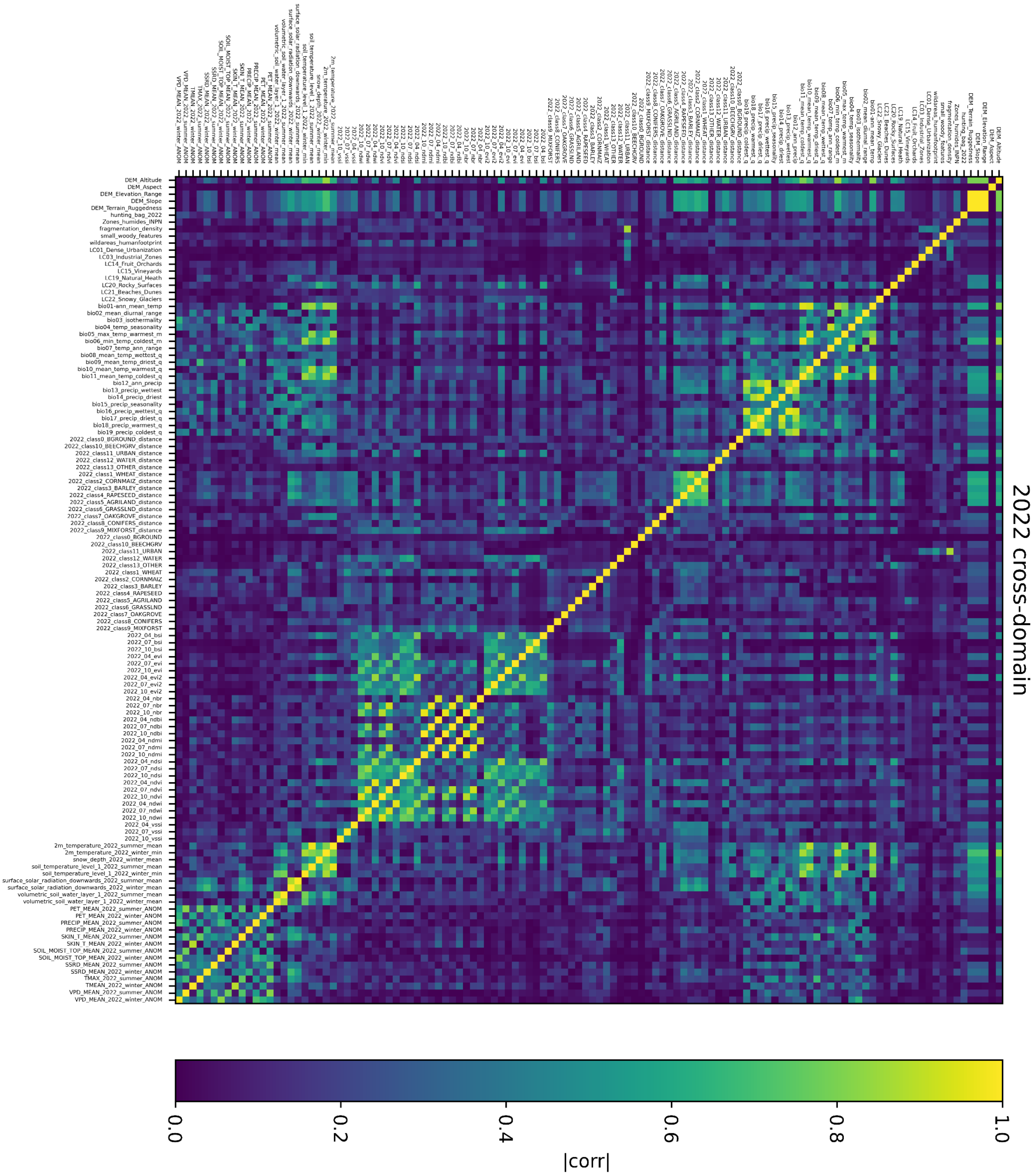


Fig. S4.6: Feature Correlation Matrix for 2022.

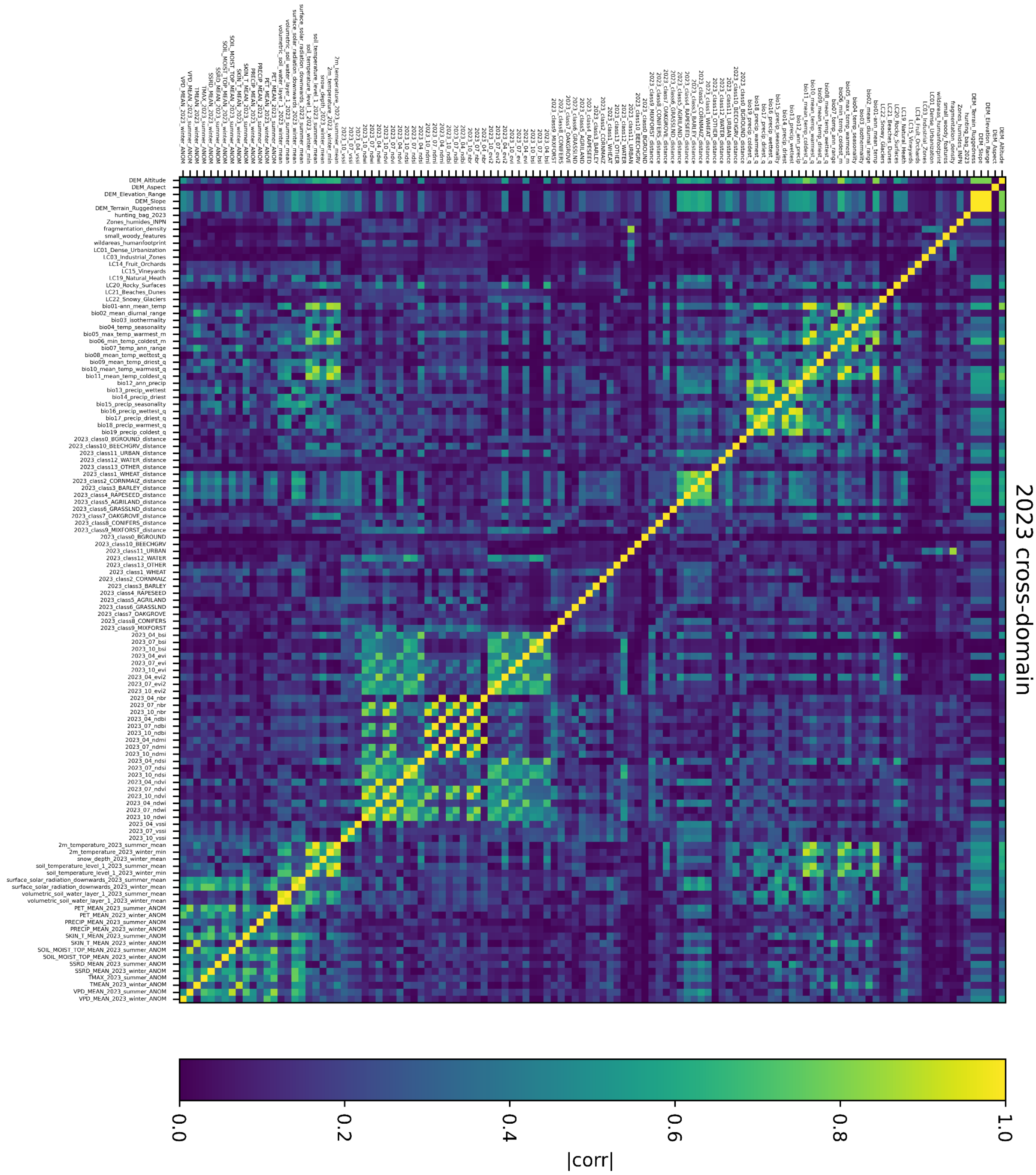
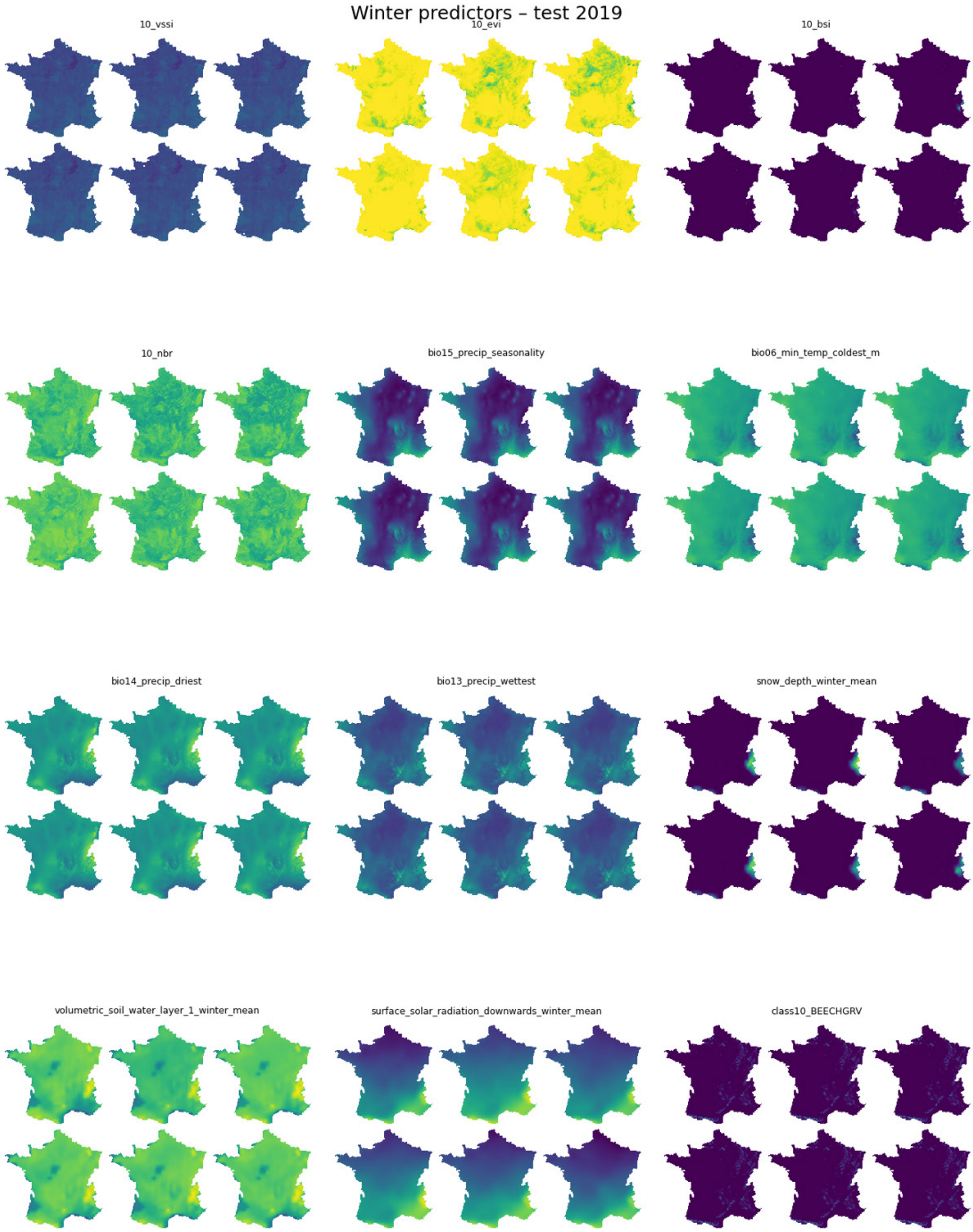
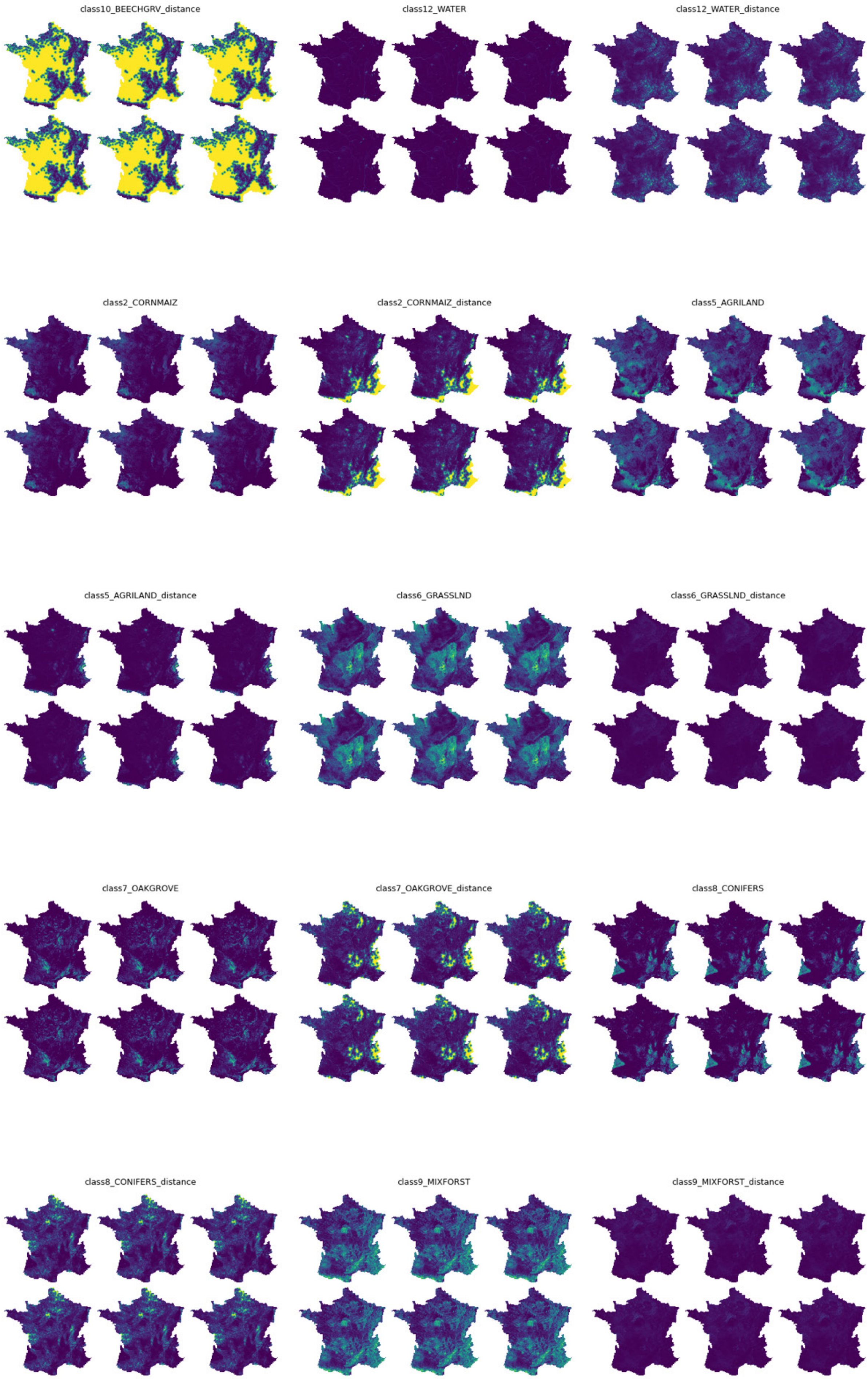


Fig. S4.7: Feature Correlation Matrix for 2023.

Selected Features
Example of Activated Features for Multitemporal Winter Models after Filtering: 2017-2013, with 2019 left out from training stack for testing.




class10_BEECHGRV_distance
class12_WATER
class12_WATER_distance
class2_CORNMAIZ
class2_CORNMAIZ_distance
class5_AGRILAND
class5_AGRILAND_distance
class6_GRASSLND
class6_GRASSLND_distance
class7_OAKGROVE
class7_OAKGROVE_distance
class8_CONIFERS
class8_CONIFERS_distance
class9_MIXFORST
class9_MIXFORST_distance

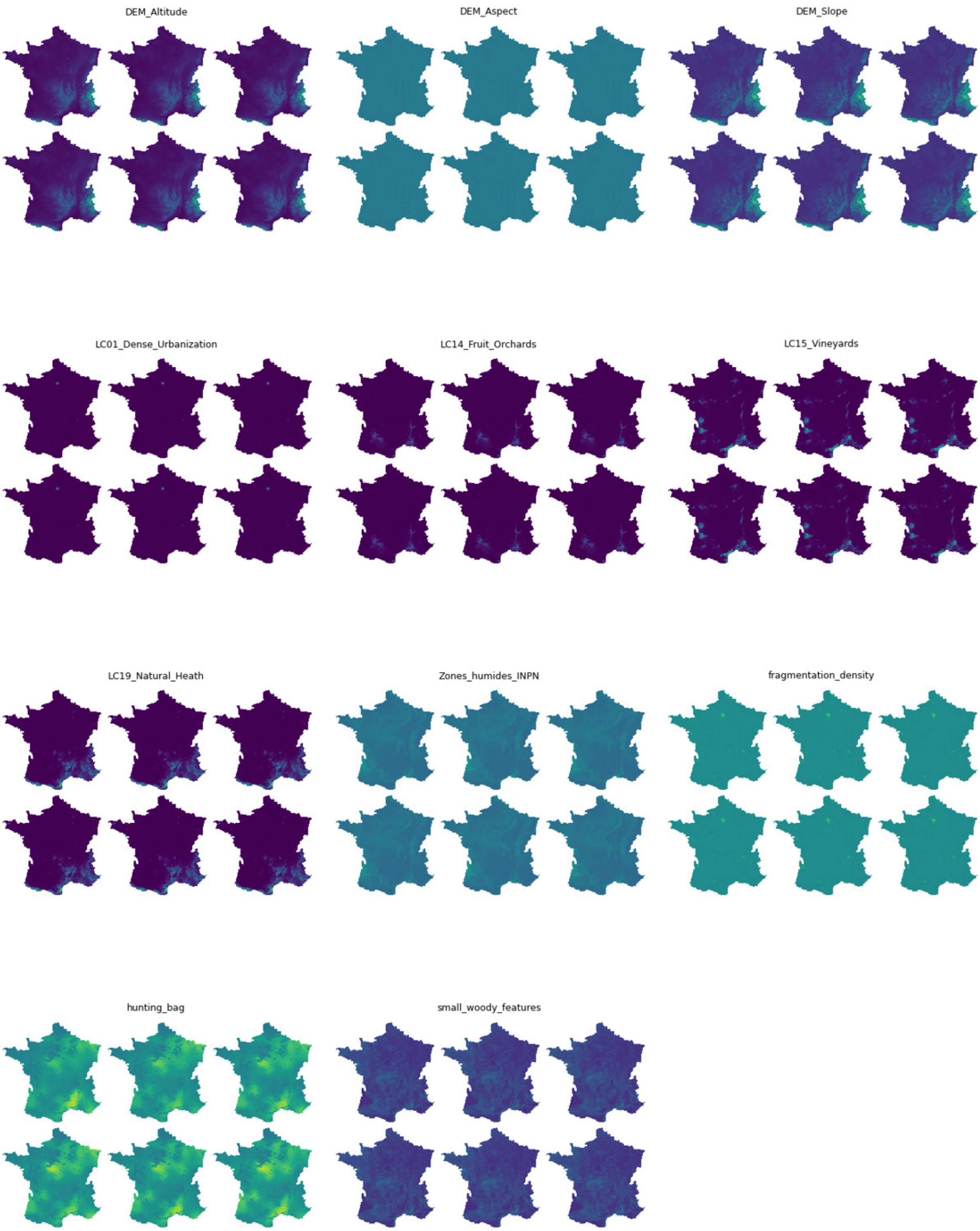


Fig. S4.8: Parallel Feature Stack for Multitemporal Winter Models. Color ramp min-max stretched.

Example of Activated Features for Monotemporal Winter Models after Filtering:
2017-2023 averaged, with 2019 left out from training stack for testing.

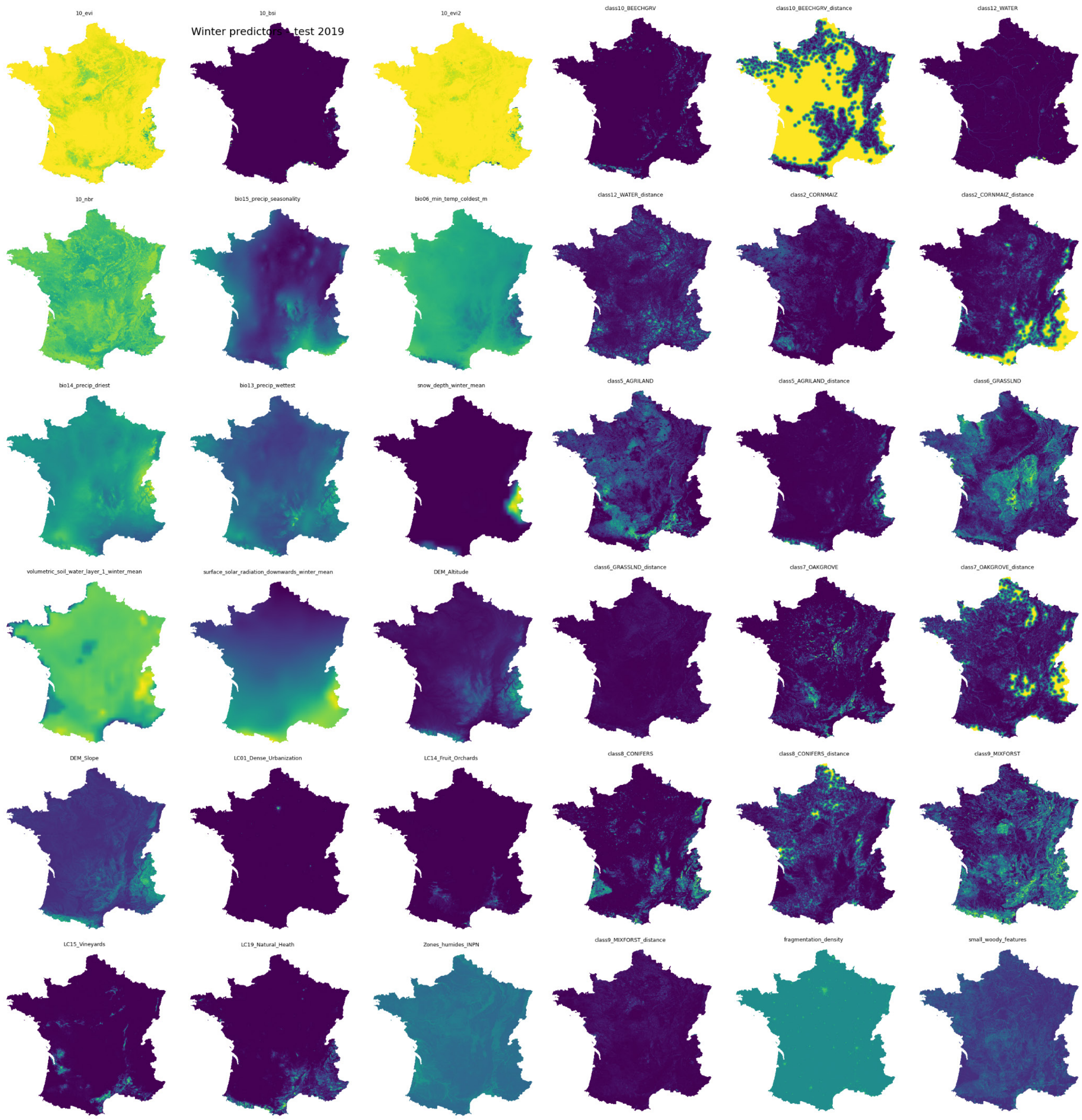
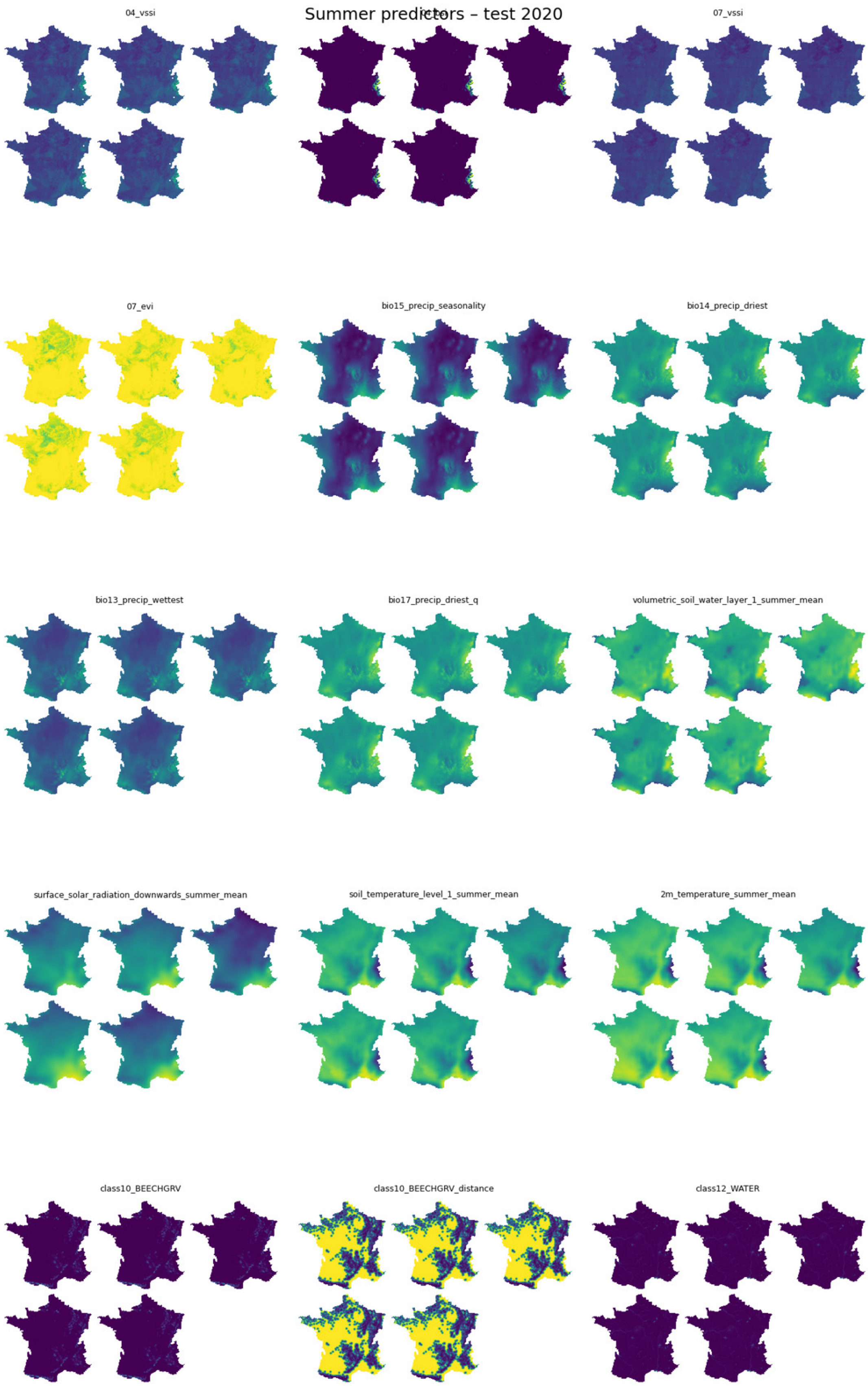
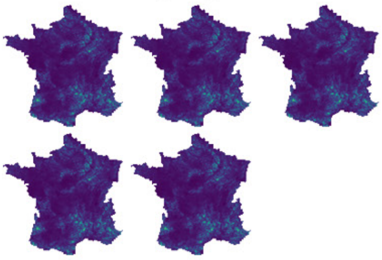


Fig. S4.9: Feature Stack for Monotemporal Winter Models. Color ramp min-max stretched.

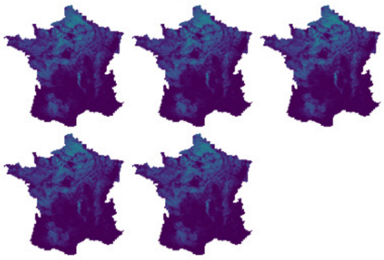
Example of Activated Features for Multitemporal Summer Models after Filtering: 2018-2023, with 2020 left out from training stack for testing.



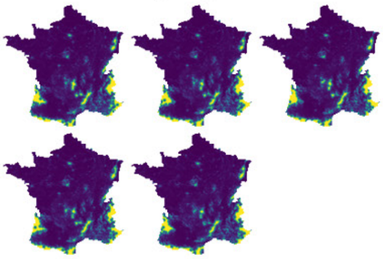
class12_WATER_distance



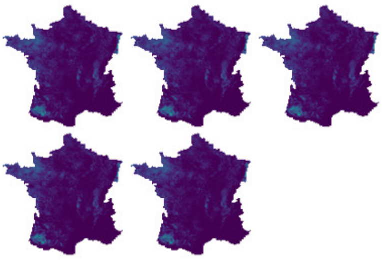
class1_WHEAT



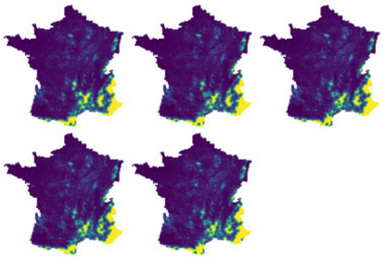
class1_WHEAT_distance



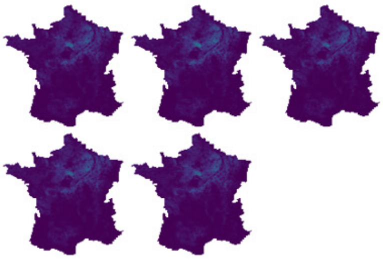
class2_CORNMAIZ



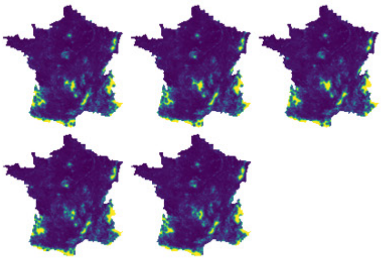
class2_CORNMAIZ_distance



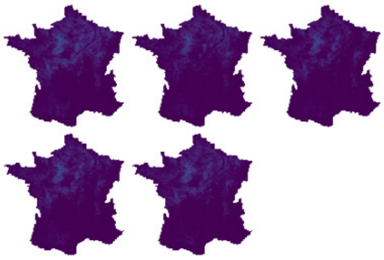
class3_BARLEY



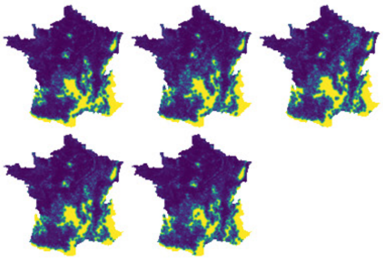
class3_BARLEY_distance



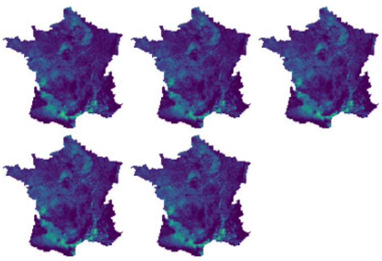
class4_RAPESEED



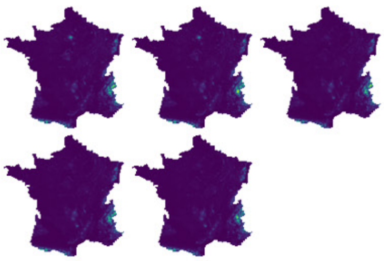
class4_RAPESEED_distance



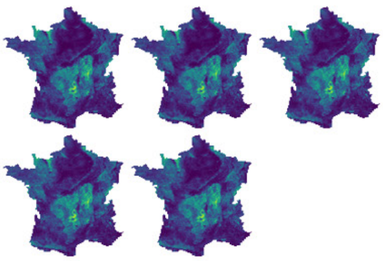
class5_AGRILAND



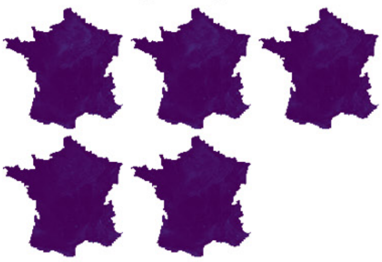
class5_AGRILAND_distance



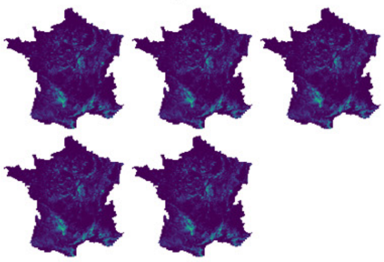
class6_GRASSLND



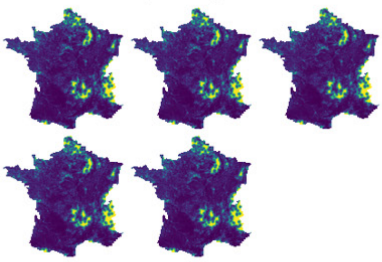
class6_GRASSLND_distance



class7_OAKGROVE



class7_OAKGROVE_distance



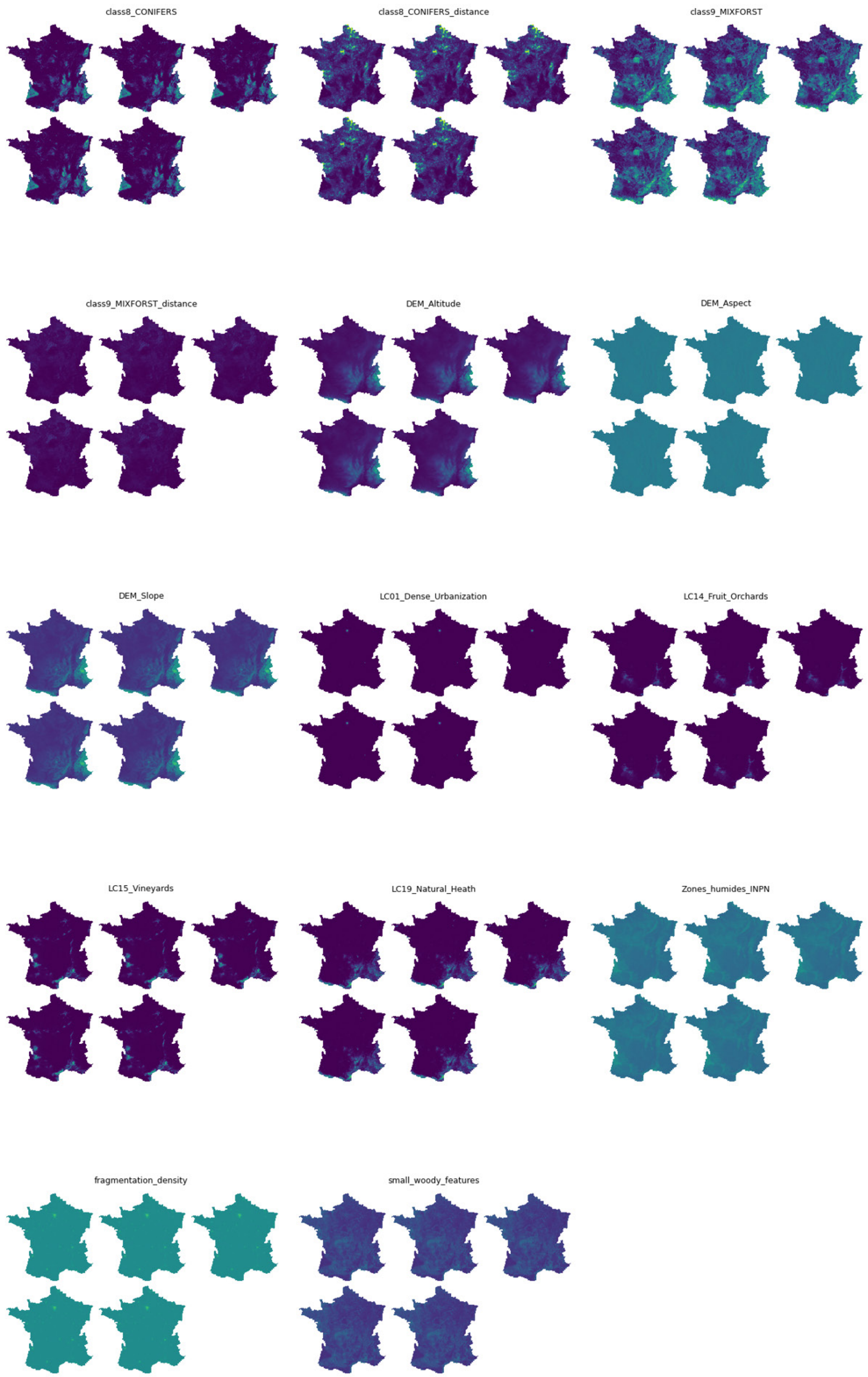


Fig. S4.10: Parallel Feature Stack for Multitemporal Summer Models. Color ramp min-max stretched.

Example of Activated Features for Monotemporal Summer Models after Filtering:
2018-2023 averaged, with 2020 left out from training stack for testing.

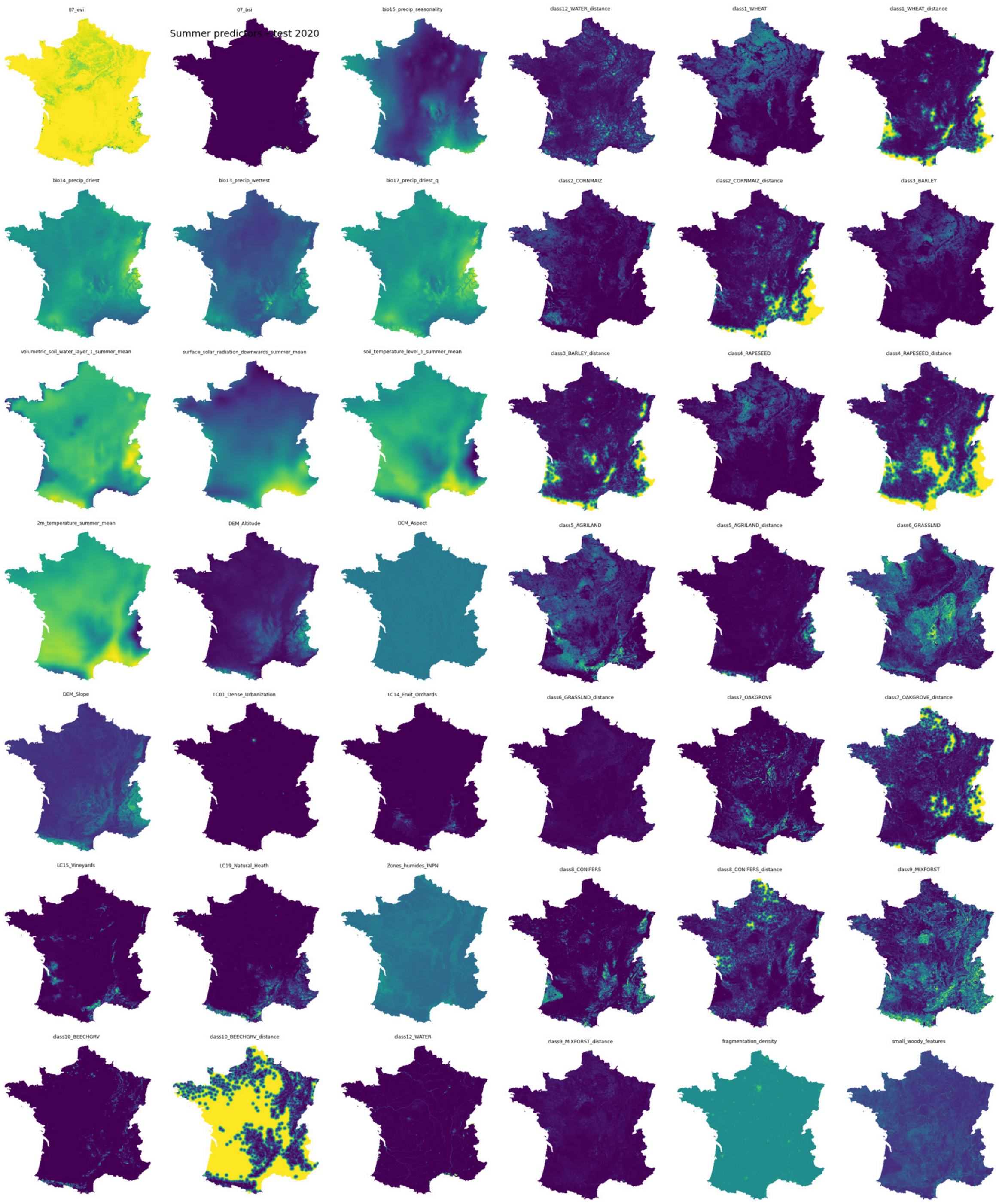


Fig. S4.11: Feature Stack for Monotemporal Summer Models. Color ramp min-max stretched.

S5 – SDM Parameters and Optuna Search Space

Feature Processing and Training Configuration for SDM

Tab. S5.1: Example Configuration YAML file for a typical comparative LOYO run.


```yaml
# ============================================================================
# Project Title: Wild Boar Distribution Model for France
# Path: config_elapid_gbm_loyo_compare.yaml
# Purpose: Aggregate configuration housing the four seasonal pipelines used for
#          GBM vs MaxEnt comparisons across all leave-one-year-out rotations.
# Process Step: Allows meta_run_leave_one_year.py to iterate winter/summer and
#               multi-/monotemporal scenarios from a single YAML document.
# Created by Adrian Meyer, Institute Geomatics, FHNW (adrian.meyer@fhnw.ch)
# Created: July 2025
# Version: v.0.1.0
# ============================================================================

meta:
  description: |
    Central configuration that bundles the individual scenario definitions
    used for GBM vs MaxEnt comparisons. Each scenario mirrors the previous
    standalone YAML files (configmultitemporal_winter.yaml,
    configwinter_monotemporal.yaml, configmultitemporal_summer.yaml and
    configsummer_monotemporal.yaml).
    The combinations explicitly cover two seasons (winter and summer),
    two temporal modes (multitemporal and monotemporal) and two modelling
    engines (elapid MaxEnt and GBM). Unlike the single-year comparison setup,
    this configuration enables leave-one-year-out rotations for every
    available year so that meta_run_leave_one_year.py can benchmark all
    combinations consistently.
  seasons: [winter, summer]
  temporal_modes: [multitemporal, monotemporal]
  engines: [elapid, gbm]
  test_year: all

scenarios:
  - key: winter_multitemporal
    label: Winter multitemporal
    script: multitemporal
    config:
      # ============================================================================
            # Configuration for multitemporal seasonal models (winter example).
      # Process Step: Governs 3x2 stacking of seasonal rasters and model settings.
      # ============================================================================

      experiment:
        name: winter_multitemporal
        random_seed: 42

      run_test_year: all  # options: all, 2017, 2018, 2019, 2020, 2021, 2022, 2023

      export_java_compatible_input_data: true

      multitemporal:
        seasons: ["Winter"]          # Seasons to process
        years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
        shift_km: 1000               # Spatial offset between copies
        stack_pattern: data/stacks/{season}_{year}.zarr
        include_hunting: true       # Toggle hunting bag predictor
        ignore_featureconfig_drop: false
        output_dir: outputs/multitemporal
        background_per_year: 15000
        validation_fraction: 0.1
        vif:
          enable: true
          n_satellite: 4
          n_bioclim: 4
          n_era5: 4
          sample_fraction: 0.05
          feature_config: featureconfig.yaml

      presence_data:
        file: data/allpoints.csv
        crs: EPSG:2154
        year_column: year
        season_column: season
        thinning:
          enable: true
          equalize_counts: false

      boundary_mask: data/boundary_france.asc

      bias_grid:
        enable: true
        pattern: data/bias-maps/Combined_{year}-{season}.asc
        output_file: outputs/bias/bias.tif

      background:
        use_bias_grid: true
        bias_strength: 0.6

      predictors:
        target_resolution_m: 1000
        resampling_method:
          continuous: bilinear
          categorical: nearest
        nodata_value: -9999
        variable_selection:
          correlation_cutoff: 0.7
```

```yaml
    vif_threshold: 12
    linkage: average
    sample_fraction: 0.01
    reference_year: 2022
    lc_reference_year_only: true
    save_selected_list: outputs/selected_predictors.txt
    feature_config: featureconfig.yaml

  layers:
    bioclim:
      files:
        - data/bioclim/bio01-ann_mean_temp.asc
        - data/bioclim/bio02_mean_diurnal_range.asc
        - data/bioclim/bio03_isothermality.asc
        - data/bioclim/bio04_temp_seasonality.asc
        - data/bioclim/bio06_min_temp_coldest_m.asc
        - data/bioclim/bio07_temp_ann_range.asc
        - data/bioclim/bio11_mean_temp_coldest_q.asc
        - data/bioclim/bio12_ann_precip.asc
        - data/bioclim/bio13_precip_wettest.asc
        - data/bioclim/bio14_precip_driest.asc
        - data/bioclim/bio15_precip_seasonality.asc
        - data/bioclim/bio16_precip_wettest_q.asc
        - data/bioclim/bio17_precip_driest_q.asc
        - data/bioclim/bio19_precip_coldest_q.asc
    landcover_fraction:
      pattern: data/land-cover/{year}_class{class}.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      classes:
        # - 1_WHEAT
        - 2_CORNMAIZ
        # - 3_BARLEY
        # - 4_RAPESEED
        - 5_AGRILAND
        - 6_GRASSLND
        - 7_OAKGROVE
        - 8_CONIFERS
        - 9_MIXFORST
        - 10_BEECHGRV
        - 11_URBAN
        - 12_WATER
    landcover_distance:
      pattern: data/lc-distances/{year}_class{class}_distance.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      classes:
        #- 1_WHEAT
        - 2_CORNMAIZ
        #- 3_BARLEY
        #- 4_RAPESEED
        - 5_AGRILAND
        - 6_GRASSLND
        - 7_OAKGROVE
        - 8_CONIFERS
        - 9_MIXFORST
        - 10_BEECHGRV
        - 11_URBAN
        - 12_WATER
    veg_index:
      pattern: data/sat-indices/{index}/{year}_{month}_{index_lc}.tif
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      months: [10]
      indices: [BSI, EVI, EVI2, NBR, NDBI, NDMI, NDSI, NDVI, NDWI, VSSI]
    era5:
      pattern: data/era5/*_{year}_{season}_*.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      seasons: [summer, winter]
    hunting_bag:
      pattern: data/hunting-bag/hunting_bag_{year}.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
    theia_special:
      files:
        - data/theia-special/LC01_Dense_Urbanization.asc
        - data/theia-special/LC14_Fruit_Orchards.asc
        - data/theia-special/LC15_Vineyards.asc
        - data/theia-special/LC19_Natural_Heath.asc
      categorical: false
    special:
      files:
        - data/special/fragmentation_density.asc
        - data/special/small_woody_features.asc
        - data/special/Zones_humides_INPN.asc
    dem:
      - data/dem/DEM_Altitude.asc
      - data/dem/DEM_Slope.asc
      - data/dem/DEM_Aspect.asc

maxent:
  engine: elapid
  output_type: cloglog
  feature_types: [linear, hinge, quadratic, product]
  product_features: true       # Toggle inclusion of product features
  n_hinge_features: auto
  regularization_multiplier: 1.8
  beta_lqp: 1               # elapid-specific shape parameter
  max_iterations: 1500
  threads: -1

gbm:
  boosting_type: gbdt
  objective: binary
  metric: [boyce]
  learning_rate: 0.03
  num_leaves: 40
  max_depth: 6
```

```yaml
      min_data_in_leaf: 48
      feature_fraction: 0.8
      bagging_fraction: 0.65
      bagging_freq: 1
      lambda_l1: 0.2
      lambda_l2: 3.0
      min_gain_to_split: 0.01
      num_boost_round: 1600
      early_stopping_rounds: 100
      validation_fraction: 0.2
      verbose_eval: 200
      num_threads: -1
      calibration:
        enable: true
        method: platt
        dataset: validation
        regularization: 1000

    prediction:
      export_geotiff: true

    evaluation:
      classification_threshold: auto
      cross_validation:
        enable: false
        method: spatial_block
        params:
          k_folds: 5
          block_size_km: 50

    variable_importance:
      permutation:
        enable: true
        n_repeats: 5

    outputs:
      root: outputs/
      logs_dir: outputs/logs/
      model_dir: outputs/models/
      maps_dir: outputs/maps/
      figures_dir: outputs/figures/
      model_paths:
        elapid: outputs/models/winter_multitemporal_elapid.pkl
        gbm: outputs/models/winter_multitemporal_gbm.pkl

- key: winter_monotemporal
  label: Winter monotemporal
  script: monotemporal
  config:
    # ============================================================================
    # Purpose: Configuration for monotemporal seasonal models (winter example).
    # Process Step: Governs 1x1 averaged stacking of seasonal rasters and model settings.
    # ============================================================================

    experiment:
      name: winter_monotemporal
      random_seed: 42

    monotemporal: true

    run_test_year: all  # options: all, 2017, 2018, 2019, 2020, 2021, 2022, 2023

    export_java_compatible_input_data: true

    multitemporal:
      seasons: ["Winter"]          # Seasons to process
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      shift_km: 1000               # Spatial offset between copies
      stack_pattern: data/stacks/{season}_{year}.zarr
      include_hunting: false       # Toggle hunting bag predictor
      ignore_featureconfig_drop: false
      output_dir: outputs/multitemporal
      background_per_year: 15000
      validation_fraction: 0.1
      vif:
        enable: true
        n_satellite: 4
        n_bioclim: 4
        n_era5: 4
        sample_fraction: 0.05
        feature_config: featureconfig.yaml

    presence_data:
      file: data/allpoints.csv
      crs: EPSG:2154
      year_column: year
      season_column: season
      thinning:
        enable: true
        equalize_counts: false

    boundary_mask: data/boundary_france.asc

    bias_grid:
      enable: true
      pattern: data/bias-maps/Combined_{year}-{season}.asc
      output_file: outputs/bias/bias.tif

    background:
      use_bias_grid: true
      bias_strength: 0.6

    predictors:
      target_resolution_m: 1000
```

```
  resampling_method:
    continuous: bilinear
    categorical: nearest
  nodata_value: -9999
  variable_selection:
    correlation_cutoff: 0.7
    vif_threshold: 12
    linkage: average
    sample_fraction: 0.01
    reference_year: 2022
    lc_reference_year_only: true
    save_selected_list: outputs/selected_predictors.txt
    feature_config: featureconfig.yaml

  layers:
    bioclim:
      files:
        - data/bioclim/bio01-ann_mean_temp.asc
        - data/bioclim/bio02_mean_diurnal_range.asc
        - data/bioclim/bio03_isothermality.asc
        - data/bioclim/bio04_temp_seasonality.asc
        - data/bioclim/bio06_min_temp_coldest_m.asc
        - data/bioclim/bio07_temp_ann_range.asc
        - data/bioclim/bio11_mean_temp_coldest_q.asc
        - data/bioclim/bio12_ann_precip.asc
        - data/bioclim/bio13_precip_wettest.asc
        - data/bioclim/bio14_precip_driest.asc
        - data/bioclim/bio15_precip_seasonality.asc
        - data/bioclim/bio16_precip_wettest_q.asc
        - data/bioclim/bio17_precip_driest_q.asc
        - data/bioclim/bio19_precip_coldest_q.asc
    landcover_fraction:
      pattern: data/land-cover/{year}_class{class}.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      classes:
        #- 1_WHEAT
        - 2_CORNMAIZ
        #- 3_BARLEY
        #- 4_RAPESEED
        - 5_AGRILAND
        - 6_GRASSLND
        - 7_OAKGROVE
        - 8_CONIFERS
        - 9_MIXFORST
        - 10_BEECHGRV
        - 11_URBAN
        - 12_WATER
    landcover_distance:
      pattern: data/lc-distances/{year}_class{class}_distance.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      classes:
        #- 1_WHEAT
        - 2_CORNMAIZ
        #- 3_BARLEY
        #- 4_RAPESEED
        - 5_AGRILAND
        - 6_GRASSLND
        - 7_OAKGROVE
        - 8_CONIFERS
        - 9_MIXFORST
        - 10_BEECHGRV
        - 11_URBAN
        - 12_WATER
    veg_index:
      pattern: data/sat-indices/{index}/{year}_{month}_{index_lc}.tif
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      months: [10]
      indices: [BSI, EVI, EVI2, NBR, NDBI, NDMI, NDSI, NDVI, NDWI, VSSI]
    era5:
      pattern: data/era5/*_{year}_{season}_*.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
      seasons: [summer, winter]
    hunting_bag:
      pattern: data/hunting-bag/hunting_bag_{year}.asc
      years: [2017, 2018, 2019, 2020, 2021, 2022, 2023]
    theia_special:
      files:
        - data/theia-special/LC01_Dense_Urbanization.asc
        - data/theia-special/LC14_Fruit_Orchards.asc
        - data/theia-special/LC15_Vineyards.asc
        - data/theia-special/LC19_Natural_Heath.asc
      categorical: false
    special:
      files:
        - data/special/fragmentation_density.asc
        - data/special/small_woody_features.asc
        - data/special/Zones_humides_INPN.asc
    dem:
      - data/dem/DEM_Altitude.asc
      - data/dem/DEM_Slope.asc
      #- data/dem/DEM_Aspect.asc

maxent:
  engine: elapid
  output_type: cloglog
  feature_types: [linear, hinge, quadratic, product]
  product_features: true       # Toggle inclusion of product features
  n_hinge_features: auto
  regularization_multiplier: 1.8
  beta_lqp: 1              # elapid-specific shape parameter
  max_iterations: 1500
  threads: -1

gbm:
```

```yaml
      boosting_type: gbdt
      objective: binary
      metric: [boyce]
      learning_rate: 0.03
      num_leaves: 40
      max_depth: 6
      min_data_in_leaf: 48
      feature_fraction: 0.8
      bagging_fraction: 0.65
      bagging_freq: 1
      lambda_l1: 0.2
      lambda_l2: 3.0
      min_gain_to_split: 0.01
      num_boost_round: 1600
      early_stopping_rounds: 100
      validation_fraction: 0.2
      verbose_eval: 200
      num_threads: -1
      calibration:
        enable: true
        method: platt
        dataset: validation
        regularization: 1000

    prediction:
      export_geotiff: true

    evaluation:
      classification_threshold: auto
      cross_validation:
        enable: false
        method: spatial_block
        params:
          k_folds: 5
          block_size_km: 50

    variable_importance:
      permutation:
        enable: true
        n_repeats: 5

    outputs:
      root: outputs/
      logs_dir: outputs/logs/
      model_dir: outputs/models/
      maps_dir: outputs/maps/
      figures_dir: outputs/figures/
      model_paths:
        elapid: outputs/models/winter_monotemporal_elapid.pkl
        gbm: outputs/models/winter_monotemporal_gbm.pkl

- key: summer_multitemporal
  label: Summer multitemporal
  script: multitemporal
  config:
    # ==========================================================================
    # Configuration for multitemporal seasonal models (summer example).
    # Process Step: Governs 3x2 stacking of seasonal rasters and model settings.
    # ==========================================================================

    experiment:
      name: summer_multitemporal
      random_seed: 42

    run_test_year: all  # options: all, 2018, 2019, 2020, 2021, 2022, 2023

    export_java_compatible_input_data: true

    multitemporal:
      seasons: ["Summer"]          # Seasons to process
      years: [2018, 2019, 2020, 2021, 2022, 2023]
      shift_km: 1000               # Spatial offset between copies
      stack_pattern: data/stacks/{season}_{year}.zarr
      include_hunting: false       # Toggle hunting bag predictor
      ignore_featureconfig_drop: false
      output_dir: outputs/multitemporal
      background_per_year: 15000
      validation_fraction: 0.1
      vif:
        enable: true
        n_satellite: 4
        n_bioclim: 4
        n_era5: 4
        sample_fraction: 0.05
        feature_config: featureconfig.yaml

    presence_data:
      file: data/allpoints.csv
      crs: EPSG:2154
      year_column: year
      season_column: season
      thinning:
        enable: true
        equalize_counts: false

    boundary_mask: data/boundary_france.asc

    bias_grid:
      enable: true
      pattern: data/bias-maps/Combined_{year}-{season}.asc
      output_file: outputs/bias/bias.tif

    background:
      use_bias_grid: true
```

```yaml
    bias_strength: 0.6

  predictors:
    target_resolution_m: 1000
    resampling_method:
      continuous: bilinear
      categorical: nearest
    nodata_value: -9999
    variable_selection:
      correlation_cutoff: 0.7
      vif_threshold: 12
      linkage: average
      sample_fraction: 0.01
      reference_year: 2022
      lc_reference_year_only: true
      save_selected_list: outputs/selected_predictors.txt
      feature_config: featureconfig.yaml

    layers:
      bioclim:
        files:
          - data/bioclim/bio01-ann_mean_temp.asc
          - data/bioclim/bio02_mean_diurnal_range.asc
          - data/bioclim/bio03_isothermality.asc
          - data/bioclim/bio04_temp_seasonality.asc
          - data/bioclim/bio05_max_temp_warmest_m.asc
          - data/bioclim/bio07_temp_ann_range.asc
          - data/bioclim/bio10_mean_temp_warmest_q.asc
          - data/bioclim/bio12_ann_precip.asc
          - data/bioclim/bio13_precip_wettest.asc
          - data/bioclim/bio14_precip_driest.asc
          - data/bioclim/bio15_precip_seasonality.asc
          - data/bioclim/bio16_precip_wettest_q.asc
          - data/bioclim/bio17_precip_driest_q.asc
          - data/bioclim/bio18_precip_warmest_q.asc
      landcover_fraction:
        pattern: data/land-cover/{year}_class{class}.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        classes:
          - 1_WHEAT
          - 2_CORNMAIZ
          - 3_BARLEY
          - 4_RAPESEED
          - 5_AGRILAND
          - 6_GRASSLND
          - 7_OAKGROVE
          - 8_CONIFERS
          - 9_MIXFORST
          - 10_BEECHGRV
          - 11_URBAN
          - 12_WATER
      landcover_distance:
        pattern: data/lc-distances/{year}_class{class}_distance.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        classes:
          - 1_WHEAT
          - 2_CORNMAIZ
          - 3_BARLEY
          - 4_RAPESEED
          - 5_AGRILAND
          - 6_GRASSLND
          - 7_OAKGROVE
          - 8_CONIFERS
          - 9_MIXFORST
          - 10_BEECHGRV
          - 11_URBAN
          - 12_WATER
      veg_index:
        pattern: data/sat-indices/{index}/{year}_{month}_{index_lc}.tif
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        months: [04, 07]
        indices: [BSI, EVI, EVI2, NBR, NDBI, NDMI, NDSI, NDVI, NDWI, VSSI]
      era5:
        pattern: data/era5/*_{year}_{season}_*.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        seasons: [summer, winter]
      hunting_bag:
        pattern: data/hunting-bag/hunting_bag_{year}.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
      theia_special:
        files:
          - data/theia-special/LC01_Dense_Urbanization.asc
          - data/theia-special/LC14_Fruit_Orchards.asc
          - data/theia-special/LC15_Vineyards.asc
          - data/theia-special/LC19_Natural_Heath.asc
        categorical: false
      special:
        files:
          - data/special/fragmentation_density.asc
          - data/special/small_woody_features.asc
          - data/special/Zones_humides_INPN.asc
      dem:
        - data/dem/DEM_Altitude.asc
        - data/dem/DEM_Slope.asc
        - data/dem/DEM_Aspect.asc

  maxent:
    engine: elapid
    output_type: cloglog
    feature_types: [linear, hinge, quadratic]
    product_features: true
    n_hinge_features: auto
    regularization_multiplier: 1.4
    max_iterations: 1500
```

```yaml
      threads: -1

    gbm:
      boosting_type: gbdt
      objective: binary
      metric: [boyce]
      learning_rate: 0.03
      num_leaves: 40
      max_depth: 6
      min_data_in_leaf: 48
      feature_fraction: 0.8
      bagging_fraction: 0.65
      bagging_freq: 1
      lambda_l1: 0.2
      lambda_l2: 3.0
      min_gain_to_split: 0.01
      num_boost_round: 1600
      early_stopping_rounds: 100
      validation_fraction: 0.2
      verbose_eval: 200
      num_threads: -1
      calibration:
        enable: true
        method: platt
        dataset: validation
        regularization: 1000

    prediction:
      export_geotiff: true

    evaluation:
      classification_threshold: auto
      cross_validation:
        enable: false
        method: spatial_block
        params:
          k_folds: 5
          block_size_km: 50

    variable_importance:
      permutation:
        enable: true
        n_repeats: 5

    outputs:
      root: outputs/
      logs_dir: outputs/logs/
      model_dir: outputs/models/
      maps_dir: outputs/maps/
      figures_dir: outputs/figures/
      model_paths:
        elapid: outputs/models/summer_multitemporal_elapid.pkl
        gbm: outputs/models/summer_multitemporal_gbm.pkl

- key: summer_monotemporal
  label: Summer monotemporal
  script: monotemporal
  config:
    # ==========================================================================
    # Purpose: Configuration for monotemporal seasonal models (summer example).
    # Process Step: Governs 1x1 averaged stacking of seasonal rasters and model settings.
    # ==========================================================================

    experiment:
      name: summer_monotemporal
      random_seed: 42

    monotemporal: true

    run_test_year: all  # options: all, 2018, 2019, 2020, 2021, 2022, 2023

    export_java_compatible_input_data: true

    multitemporal:
      seasons: ["Summer"]          # Seasons to process
      years: [2018, 2019, 2020, 2021, 2022, 2023]
      shift_km: 1000               # Spatial offset between copies
      stack_pattern: data/stacks/{season}_{year}.zarr
      include_hunting: false       # Toggle hunting bag predictor
      ignore_featureconfig_drop: false
      output_dir: outputs/multitemporal
      background_per_year: 15000
      validation_fraction: 0.1
      vif:
        enable: true
        n_satellite: 4
        n_bioclim: 4
        n_era5: 4
        sample_fraction: 0.2
        feature_config: featureconfig.yaml

    presence_data:
      file: data/allpoints.csv
      crs: EPSG:2154
      year_column: year
      season_column: season
      thinning:
        enable: true
        equalize_counts: false

    boundary_mask: data/boundary_france.asc

    bias_grid:
      enable: true
```

```yaml
      pattern: data/bias-maps/Combined_{year}-{season}.asc
      output_file: outputs/bias/bias.tif

  background:
    use_bias_grid: true
    bias_strength: 0.6

  predictors:
    target_resolution_m: 1000
    resampling_method:
      continuous: bilinear
      categorical: nearest
    nodata_value: -9999
    variable_selection:
      correlation_cutoff: 0.7
      vif_threshold: 12
      linkage: average
      sample_fraction: 0.01
      reference_year: 2022
      lc_reference_year_only: true
      save_selected_list: outputs/selected_predictors.txt
      feature_config: featureconfig.yaml

    layers:
      bioclim:
        files:
          - data/bioclim/bio01-ann_mean_temp.asc
          - data/bioclim/bio02_mean_diurnal_range.asc
          - data/bioclim/bio03_isothermality.asc
          - data/bioclim/bio04_temp_seasonality.asc
          - data/bioclim/bio05_max_temp_warmest_m.asc
          - data/bioclim/bio07_temp_ann_range.asc
          - data/bioclim/bio10_mean_temp_warmest_q.asc
          - data/bioclim/bio12_ann_precip.asc
          - data/bioclim/bio13_precip_wettest.asc
          - data/bioclim/bio14_precip_driest.asc
          - data/bioclim/bio15_precip_seasonality.asc
          - data/bioclim/bio16_precip_wettest_q.asc
          - data/bioclim/bio17_precip_driest_q.asc
          - data/bioclim/bio18_precip_warmest_q.asc
      landcover_fraction:
        pattern: data/land-cover/{year}_class{class}.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        classes:
          - 1_WHEAT
          - 2_CORNMAIZ
          - 3_BARLEY
          - 4_RAPESEED
          - 5_AGRILAND
          - 6_GRASSLND
          - 7_OAKGROVE
          - 8_CONIFERS
          - 9_MIXFORST
          - 10_BEECHGRV
          - 11_URBAN
          - 12_WATER
      landcover_distance:
        pattern: data/lc-distances/{year}_class{class}_distance.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        classes:
          - 1_WHEAT
          - 2_CORNMAIZ
          - 3_BARLEY
          - 4_RAPESEED
          - 5_AGRILAND
          - 6_GRASSLND
          - 7_OAKGROVE
          - 8_CONIFERS
          - 9_MIXFORST
          - 10_BEECHGRV
          - 11_URBAN
          - 12_WATER
      veg_index:
        pattern: data/sat-indices/{index}/{year}_{month}_{index_lc}.tif
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        months: [04, 07]
        indices: [BSI, EVI, EVI2, NBR, NDBI, NDMI, NDSI, NDVI, NDWI, VSSI]
      era5:
        pattern: data/era5/*_{year}_{season}_*.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
        seasons: [summer, winter]
      hunting_bag:
        pattern: data/hunting-bag/hunting_bag_{year}.asc
        years: [2018, 2019, 2020, 2021, 2022, 2023]
      theia_special:
        files:
          - data/theia-special/LC01_Dense_Urbanization.asc
          - data/theia-special/LC14_Fruit_Orchards.asc
          - data/theia-special/LC15_Vineyards.asc
          - data/theia-special/LC19_Natural_Heath.asc
        categorical: false
      special:
        files:
          - data/special/fragmentation_density.asc
          - data/special/small_woody_features.asc
          - data/special/Zones_humides_INPN.asc
      dem:
        - data/dem/DEM_Altitude.asc
        - data/dem/DEM_Slope.asc
        - data/dem/DEM_Aspect.asc

  maxent:
    engine: elapid
    output_type: cloglog
```

```
    feature_types: [linear, hinge, quadratic]
    product_features: true       # Toggle inclusion of product features
    n_hinge_features: auto
    regularization_multiplier: 1.4
    beta_lqp: 1              # elapid-specific shape parameter
    max_iterations: 1500
    threads: -1

  gbm:
    boosting_type: gbdt
    objective: binary
    metric: [auc]
    learning_rate: 0.03
    num_leaves: 40
    max_depth: 6
    min_data_in_leaf: 48
    feature_fraction: 0.8
    bagging_fraction: 0.65
    bagging_freq: 1
    lambda_l1: 0.2
    lambda_l2: 3.0
    min_gain_to_split: 0.01
    num_boost_round: 1600
    early_stopping_rounds: 100
    validation_fraction: 0.2
    verbose_eval: 200
    num_threads: -1
    calibration:
      enable: true
      method: platt
      dataset: validation
      regularization: 1000

  prediction:
    export_geotiff: true

  evaluation:
    classification_threshold: auto
    cross_validation:
      enable: false
      method: spatial_block
      params:
        k_folds: 5
        block_size_km: 50

  variable_importance:
    permutation:
      enable: true
      n_repeats: 5

  outputs:
    root: outputs/
    logs_dir: outputs/logs/
    model_dir: outputs/models/
    maps_dir: outputs/maps/
    figures_dir: outputs/figures/
    model_paths:
      elapid: outputs/models/summer_monotemporal_elapid.pkl
      gbm: outputs/models/summer_monotemporal_gbm.pkl
```

Optuna Search Space for MaxEnt (elapid)

Tab. S5.2: Config YAML with MaxEnt parameter boundaries.


```yaml
optuna:
  n_trials: 200
  n_jobs: 12
  objective_metric: boyce  # Options: boyce, auc, boyce+auc, aicc, omission

  search_space:
    regularization_multiplier:
      low: 0.5
      high: 4.0
    n_hinge_features:
      low: 10
      high: 40
    beta_lqp:
      low: 0.5
      high: 1.5
      step: 0.1
    background_per_year:
      low: 5000
      high: 20000
      step: 5000
    training_sample_factor:
      choices: [0.25, 0.5, 1.0, 2.0]
    product_features:
      choices: [true, false]
    max_iterations:
      low: 500
      high: 2000
      step: 500
```

Optuna Search Space for GBM (lightgbm)

Tab. S5.3: Config YAML with GBM parameter boundaries.


```yaml
optuna:
  n_trials: 200
  n_jobs: 12
  objective_metric: boyce   # Options: boyce, auc, boyce+auc, aicc, omission

  search_space:
    num_leaves:
      low: 16
      high: 64
    max_depth:
      choices: [-1, 4, 5, 6, 7]
    min_data_in_leaf:
      low: 20
      high: 120
      step: 10
    min_gain_to_split:
      low: 0.0
      high: 0.05
    learning_rate:
      low: 0.005
      high: 0.08
      log: true
    num_boost_round:
      low: 600
      high: 2500
      step: 200
    early_stopping_rounds:
      choices: [50, 100, 150]
    feature_fraction:
      low: 0.6
      high: 0.95
    bagging_fraction:
      low: 0.5
      high: 0.85
    bagging_freq:
      choices: [1, 5]
    lambda_l1:
      low: 0.0
      high: 2.0
    lambda_l2:
      low: 0.0
      high: 10.0
    scale_pos_weight:
      low: 0.5
      high: 5.0
    calibration:
      enable:
        choices: [true, false]
      method:
        choices: [platt, isotonic]
      regularization:
        low: 10
        high: 2000
        log: true
```

S6 – LOYO SDM Statistics

Tab. S6.1: Summer Models using points from all sources (WVCs + GBIF).

Year	Engine	Temporal	Dataset	Season	TrainCBI	ValCBI	TestCBI	TrainAUC	ValAUC	TestAUC
2018	GBM	Mono	All_Points	Summer	0.996	0.891	0.967	0.837	0.68	0.666
2019	GBM	Mono	All_Points	Summer	0.950	0.503	0.918	0.798	0.663	0.642
2020	GBM	Mono	All_Points	Summer	0.993	0.364	0.836	0.814	0.686	0.643
2021	GBM	Mono	All_Points	Summer	0.996	0.361	0.905	0.803	0.656	0.624
2022	GBM	Mono	All_Points	Summer	1.000	0.985	1.000	0.789	0.688	0.743
2023	GBM	Mono	All_Points	Summer	0.982	0.522	-0.088	0.794	0.663	0.710
2018	GBM	Multi	All_Points	Summer	0.993	0.386	0.618	0.782	0.625	0.706
2019	GBM	Multi	All_Points	Summer	0.903	0.422	0.351	0.802	0.615	0.631
2020	GBM	Multi	All_Points	Summer	0.880	0.169	0.593	0.762	0.633	0.642
2021	GBM	Multi	All_Points	Summer	0.915	0.168	0.977	0.769	0.616	0.627
2022	GBM	Multi	All_Points	Summer	0.953	0.913	0.958	0.816	0.616	0.723
2023	GBM	Multi	All_Points	Summer	0.907	0.732	0.113	0.756	0.617	0.718
2018	MaxEnt	Mono	All_Points	Summer	0.998	0.965	0.897	0.656	0.64	0.657
2019	MaxEnt	Mono	All_Points	Summer	0.998	0.966	0.991	0.667	0.638	0.617
2020	MaxEnt	Mono	All_Points	Summer	0.998	0.968	0.972	0.662	0.65	0.623
2021	MaxEnt	Mono	All_Points	Summer	0.996	0.936	0.958	0.670	0.618	0.588
2022	MaxEnt	Mono	All_Points	Summer	0.998	0.974	0.887	0.652	0.673	0.704
2023	MaxEnt	Mono	All_Points	Summer	0.998	0.950	0.952	0.655	0.624	0.679
2018	MaxEnt	Multi	All_Points	Summer	0.998	0.989	0.938	0.671	0.653	0.670
2019	MaxEnt	Multi	All_Points	Summer	0.982	0.984	0.982	0.702	0.668	0.610
2020	MaxEnt	Multi	All_Points	Summer	0.985	0.951	0.984	0.694	0.686	0.617
2021	MaxEnt	Multi	All_Points	Summer	0.997	0.968	0.841	0.688	0.655	0.579
2022	MaxEnt	Multi	All_Points	Summer	0.991	0.959	0.897	0.675	0.648	0.681
2023	MaxEnt	Multi	All_Points	Summer	0.984	0.886	0.669	0.676	0.657	0.705

Tab. S6.2: Winter Models Models using points from all sources (WVCs + GBIF).

Year	Engine	Temporal	Dataset	Season	TrainCBI	ValCBI	TestCBI	TrainAUC	ValAUC	TestAUC
2017	GBM	Mono	All_Points	Winter	0.982	0.911	-0.353	0.841	0.708	0.734
2018	GBM	Mono	All_Points	Winter	0.982	0.322	0.981	0.829	0.708	0.678
2019	GBM	Mono	All_Points	Winter	0.982	0.13	0.974	0.826	0.721	0.644
2020	GBM	Mono	All_Points	Winter	0.989	0.334	0.947	0.837	0.729	0.696
2021	GBM	Mono	All_Points	Winter	0.987	0.521	0.985	0.813	0.723	0.727
2022	GBM	Mono	All_Points	Winter	0.909	0.186	0.935	0.820	0.730	0.746
2023	GBM	Mono	All_Points	Winter	0.993	0.013	0.102	0.835	0.707	0.716
2017	GBM	Multi	All_Points	Winter	1.000	0.947	0.512	0.882	0.727	0.677
2018	GBM	Multi	All_Points	Winter	0.997	0.704	0.904	0.902	0.715	0.661
2019	GBM	Multi	All_Points	Winter	0.980	0.955	0.645	0.875	0.728	0.654
2020	GBM	Multi	All_Points	Winter	0.988	0.105	0.903	0.874	0.742	0.707
2021	GBM	Multi	All_Points	Winter	0.990	0.623	0.639	0.855	0.726	0.721
2022	GBM	Multi	All_Points	Winter	0.838	0.609	0.069	0.883	0.723	0.754
2023	GBM	Multi	All_Points	Winter	0.971	0.928	0.246	0.871	0.74	0.686
2017	MaxEnt	Mono	All_Points	Winter	0.989	0.964	0.919	0.675	0.652	0.683
2018	MaxEnt	Mono	All_Points	Winter	0.995	0.962	0.989	0.668	0.68	0.642
2019	MaxEnt	Mono	All_Points	Winter	0.939	0.962	0.981	0.687	0.679	0.608
2020	MaxEnt	Mono	All_Points	Winter	0.994	0.729	0.983	0.675	0.685	0.649
2021	MaxEnt	Mono	All_Points	Winter	0.997	0.774	0.974	0.667	0.68	0.681
2022	MaxEnt	Mono	All_Points	Winter	0.996	0.906	0.987	0.67	0.676	0.676
2023	MaxEnt	Mono	All_Points	Winter	0.998	0.942	0.936	0.676	0.668	0.687
2017	MaxEnt	Multi	All_Points	Winter	0.998	0.988	0.738	0.675	0.674	0.595
2018	MaxEnt	Multi	All_Points	Winter	0.992	0.968	0.993	0.677	0.645	0.631
2019	MaxEnt	Multi	All_Points	Winter	0.998	0.991	0.989	0.699	0.682	0.618
2020	MaxEnt	Multi	All_Points	Winter	0.997	0.984	0.988	0.684	0.665	0.652
2021	MaxEnt	Multi	All_Points	Winter	0.994	0.980	0.974	0.680	0.657	0.664
2022	MaxEnt	Multi	All_Points	Winter	0.994	0.956	0.984	0.678	0.663	0.690
2023	MaxEnt	Multi	All_Points	Winter	0.997	0.985	0.965	0.674	0.665	0.665

Tab. S6.3: Summer Models using only GBIF points.

Year	Engine	Temporal	Dataset	Season	TrainCBI	ValCBI	TestCBI	TrainAUC	ValAUC	TestAUC
2018	GBM	Mono	GBIF_only	Summer	0.995	0.130	0.204	0.783	0.696	0.679
2019	GBM	Mono	GBIF_only	Summer	0.943	0.783	0.909	0.852	0.672	0.647
2020	GBM	Mono	GBIF_only	Summer	0.950	0.528	0.927	0.826	0.682	0.659
2021	GBM	Mono	GBIF_only	Summer	0.995	0.594	0.191	0.815	0.655	0.625
2022	GBM	Mono	GBIF_only	Summer	0.982	-0.059	0.946	0.788	0.653	0.757
2023	GBM	Mono	GBIF_only	Summer	0.995	0.064	0.986	0.766	0.681	0.707
2018	GBM	Multi	GBIF_only	Summer	0.971	0.944	0.600	0.851	0.651	0.694
2019	GBM	Multi	GBIF_only	Summer	0.864	0.779	0.902	0.782	0.635	0.637
2020	GBM	Multi	GBIF_only	Summer	0.971	0.473	0.703	0.826	0.649	0.646
2021	GBM	Multi	GBIF_only	Summer	0.912	0.806	0.575	0.808	0.606	0.621
2022	GBM	Multi	GBIF_only	Summer	0.971	0.846	0.967	0.821	0.642	0.738
2023	GBM	Multi	GBIF_only	Summer	0.988	0.86	-0.131	0.837	0.666	0.709
2018	MaxEnt	Mono	GBIF_only	Summer	0.996	0.961	0.917	0.656	0.656	0.653
2019	MaxEnt	Mono	GBIF_only	Summer	0.997	0.966	0.991	0.664	0.637	0.613
2020	MaxEnt	Mono	GBIF_only	Summer	0.997	0.974	0.989	0.662	0.630	0.634
2021	MaxEnt	Mono	GBIF_only	Summer	0.997	0.919	0.907	0.667	0.627	0.587
2022	MaxEnt	Mono	GBIF_only	Summer	0.998	0.974	0.955	0.658	0.628	0.702
2023	MaxEnt	Mono	GBIF_only	Summer	0.992	0.968	0.885	0.653	0.644	0.670
2018	MaxEnt	Multi	GBIF_only	Summer	0.999	0.967	0.914	0.672	0.661	0.651
2019	MaxEnt	Multi	GBIF_only	Summer	0.989	0.962	0.985	0.694	0.669	0.610
2020	MaxEnt	Multi	GBIF_only	Summer	0.996	0.956	0.984	0.690	0.684	0.626
2021	MaxEnt	Multi	GBIF_only	Summer	0.998	0.939	0.851	0.690	0.638	0.585
2022	MaxEnt	Multi	GBIF_only	Summer	0.989	0.875	0.941	0.666	0.686	0.693
2023	MaxEnt	Multi	GBIF_only	Summer	0.993	0.878	0.811	0.670	0.676	0.695

Tab. S6.4: Winter Models using only GBIF points.

Year	Engine	Temporal	Dataset	Season	TrainCBI	ValCBI	TestCBI	TrainAUC	ValAUC	TestAUC
2017	GBM	Mono	GBIF_only	Winter	0.997	0.561	0.654	0.820	0.728	0.744
2018	GBM	Mono	GBIF_only	Winter	0.988	0.929	0.687	0.855	0.734	0.668
2019	GBM	Mono	GBIF_only	Winter	0.997	0.512	0.826	0.838	0.729	0.650
2020	GBM	Mono	GBIF_only	Winter	0.998	0.583	0.942	0.842	0.709	0.712
2021	GBM	Mono	GBIF_only	Winter	0.972	0.403	0.656	0.818	0.691	0.735
2022	GBM	Mono	GBIF_only	Winter	0.998	0.902	0.961	0.822	0.713	0.802
2023	GBM	Mono	GBIF_only	Winter	0.995	0.953	0.836	0.804	0.711	0.716
2017	GBM	Multi	GBIF_only	Winter	0.996	0.950	0.439	0.907	0.726	0.776
2018	GBM	Multi	GBIF_only	Winter	0.996	0.910	0.927	0.915	0.779	0.668
2019	GBM	Multi	GBIF_only	Winter	0.991	0.366	0.920	0.931	0.750	0.674
2020	GBM	Multi	GBIF_only	Winter	0.966	0.835	0.971	0.930	0.765	0.719
2021	GBM	Multi	GBIF_only	Winter	0.994	0.639	0.981	0.909	0.744	0.745
2022	GBM	Multi	GBIF_only	Winter	0.991	0.902	0.881	0.869	0.730	0.811
2023	GBM	Multi	GBIF_only	Winter	0.995	0.689	0.702	0.921	0.759	0.729
2017	MaxEnt	Mono	GBIF_only	Winter	0.993	0.959	0.951	0.654	0.650	0.696
2018	MaxEnt	Mono	GBIF_only	Winter	0.995	0.980	0.989	0.663	0.658	0.622
2019	MaxEnt	Mono	GBIF_only	Winter	0.998	0.874	0.984	0.671	0.658	0.598
2020	MaxEnt	Mono	GBIF_only	Winter	0.999	0.993	0.990	0.656	0.662	0.654
2021	MaxEnt	Mono	GBIF_only	Winter	0.994	0.972	0.991	0.656	0.644	0.666
2022	MaxEnt	Mono	GBIF_only	Winter	0.994	0.980	0.974	0.653	0.642	0.718
2023	MaxEnt	Mono	GBIF_only	Winter	0.997	0.985	0.960	0.657	0.646	0.684
2017	MaxEnt	Multi	GBIF_only	Winter	0.996	0.987	0.932	0.662	0.652	0.681
2018	MaxEnt	Multi	GBIF_only	Winter	0.999	0.952	0.988	0.677	0.677	0.607
2019	MaxEnt	Multi	GBIF_only	Winter	0.998	0.969	0.991	0.688	0.651	0.617
2020	MaxEnt	Multi	GBIF_only	Winter	0.995	0.973	0.991	0.668	0.673	0.653
2021	MaxEnt	Multi	GBIF_only	Winter	0.995	0.990	0.988	0.665	0.654	0.663
2022	MaxEnt	Multi	GBIF_only	Winter	0.998	0.983	0.975	0.661	0.636	0.721
2023	MaxEnt	Multi	GBIF_only	Winter	0.999	0.980	0.970	0.661	0.669	0.681

Tab. S6.5: Summary statistics (mean ± SD across years)

Dataset	Season	Split	Engine	Temporal	AUC (mean ± SD)	CBI (mean ± SD)
All_Points	Summer	Test	GBM	Monotemporal	0.671 ± 0.046	0.756 ± 0.418
All_Points	Summer	Test	GBM	Multitemporal	0.675 ± 0.046	0.602 ± 0.338
All_Points	Summer	Test	Maxent (elapid)	Monotemporal	0.645 ± 0.043	0.943 ± 0.042
All_Points	Summer	Test	Maxent (elapid)	Multitemporal	0.644 ± 0.049	0.885 ± 0.119
All_Points	Summer	Train	GBM	Monotemporal	0.806 ± 0.018	0.986 ± 0.019
All_Points	Summer	Train	GBM	Multitemporal	0.781 ± 0.024	0.925 ± 0.041
All_Points	Summer	Train	Maxent (elapid)	Monotemporal	0.661 ± 0.007	0.998 ± 0.001
All_Points	Summer	Train	Maxent (elapid)	Multitemporal	0.684 ± 0.012	0.989 ± 0.007
All_Points	Summer	Validation	GBM	Monotemporal	0.673 ± 0.013	0.604 ± 0.269
All_Points	Summer	Validation	GBM	Multitemporal	0.621 ± 0.007	0.465 ± 0.302
All_Points	Summer	Validation	Maxent (elapid)	Monotemporal	0.641 ± 0.020	0.960 ± 0.014
All_Points	Summer	Validation	Maxent (elapid)	Multitemporal	0.661 ± 0.014	0.956 ± 0.037
All_Points	Winter	Test	GBM	Monotemporal	0.706 ± 0.036	0.653 ± 0.548
All_Points	Winter	Test	GBM	Multitemporal	0.694 ± 0.036	0.560 ± 0.314
All_Points	Winter	Test	Maxent (elapid)	Monotemporal	0.661 ± 0.029	0.967 ± 0.028
All_Points	Winter	Test	Maxent (elapid)	Multitemporal	0.645 ± 0.032	0.947 ± 0.093
All_Points	Winter	Train	GBM	Monotemporal	0.828 ± 0.010	0.975 ± 0.029
All_Points	Winter	Train	GBM	Multitemporal	0.877 ± 0.014	0.966 ± 0.058
All_Points	Winter	Train	Maxent (elapid)	Monotemporal	0.674 ± 0.007	0.987 ± 0.021
All_Points	Winter	Train	Maxent (elapid)	Multitemporal	0.681 ± 0.009	0.996 ± 0.002
All_Points	Winter	Validation	GBM	Monotemporal	0.718 ± 0.010	0.345 ± 0.298
All_Points	Winter	Validation	GBM	Multitemporal	0.729 ± 0.010	0.696 ± 0.302
All_Points	Winter	Validation	Maxent (elapid)	Monotemporal	0.674 ± 0.011	0.891 ± 0.098
All_Points	Winter	Validation	Maxent (elapid)	Multitemporal	0.665 ± 0.012	0.979 ± 0.013
GBIF_only	Summer	Test	GBM	Monotemporal	0.679 ± 0.047	0.694 ± 0.385
GBIF_only	Summer	Test	GBM	Multitemporal	0.674 ± 0.046	0.603 ± 0.393
GBIF_only	Summer	Test	Maxent (elapid)	Monotemporal	0.643 ± 0.041	0.941 ± 0.045
GBIF_only	Summer	Test	Maxent (elapid)	Multitemporal	0.643 ± 0.045	0.914 ± 0.071
GBIF_only	Summer	Train	GBM	Monotemporal	0.805 ± 0.032	0.976 ± 0.024
GBIF_only	Summer	Train	GBM	Multitemporal	0.821 ± 0.024	0.946 ± 0.048
GBIF_only	Summer	Train	Maxent (elapid)	Monotemporal	0.660 ± 0.005	0.996 ± 0.002
GBIF_only	Summer	Train	Maxent (elapid)	Multitemporal	0.680 ± 0.012	0.994 ± 0.004
GBIF_only	Summer	Validation	GBM	Monotemporal	0.673 ± 0.017	0.340 ± 0.340
GBIF_only	Summer	Validation	GBM	Multitemporal	0.642 ± 0.020	0.785 ± 0.163
GBIF_only	Summer	Validation	Maxent (elapid)	Monotemporal	0.637 ± 0.011	0.960 ± 0.021
GBIF_only	Summer	Validation	Maxent (elapid)	Multitemporal	0.669 ± 0.018	0.929 ± 0.042
GBIF_only	Winter	Test	GBM	Monotemporal	0.718 ± 0.050	0.795 ± 0.131
GBIF_only	Winter	Test	GBM	Multitemporal	0.732 ± 0.052	0.831 ± 0.197
GBIF_only	Winter	Test	Maxent (elapid)	Monotemporal	0.663 ± 0.042	0.977 ± 0.016
GBIF_only	Winter	Test	Maxent (elapid)	Multitemporal	0.660 ± 0.039	0.976 ± 0.021
GBIF_only	Winter	Train	GBM	Monotemporal	0.829 ± 0.017	0.992 ± 0.010
GBIF_only	Winter	Train	GBM	Multitemporal	0.912 ± 0.021	0.990 ± 0.011
GBIF_only	Winter	Train	Maxent (elapid)	Monotemporal	0.659 ± 0.006	0.996 ± 0.002
GBIF_only	Winter	Train	Maxent (elapid)	Multitemporal	0.669 ± 0.010	0.997 ± 0.002
GBIF_only	Winter	Validation	GBM	Monotemporal	0.717 ± 0.015	0.692 ± 0.228
GBIF_only	Winter	Validation	GBM	Multitemporal	0.750 ± 0.019	0.756 ± 0.208
GBIF_only	Winter	Validation	Maxent (elapid)	Monotemporal	0.651 ± 0.008	0.963 ± 0.041
GBIF_only	Winter	Validation	Maxent (elapid)	Multitemporal	0.659 ± 0.015	0.976 ± 0.013

S7 – Variable Response

Multitemporal Winter Model

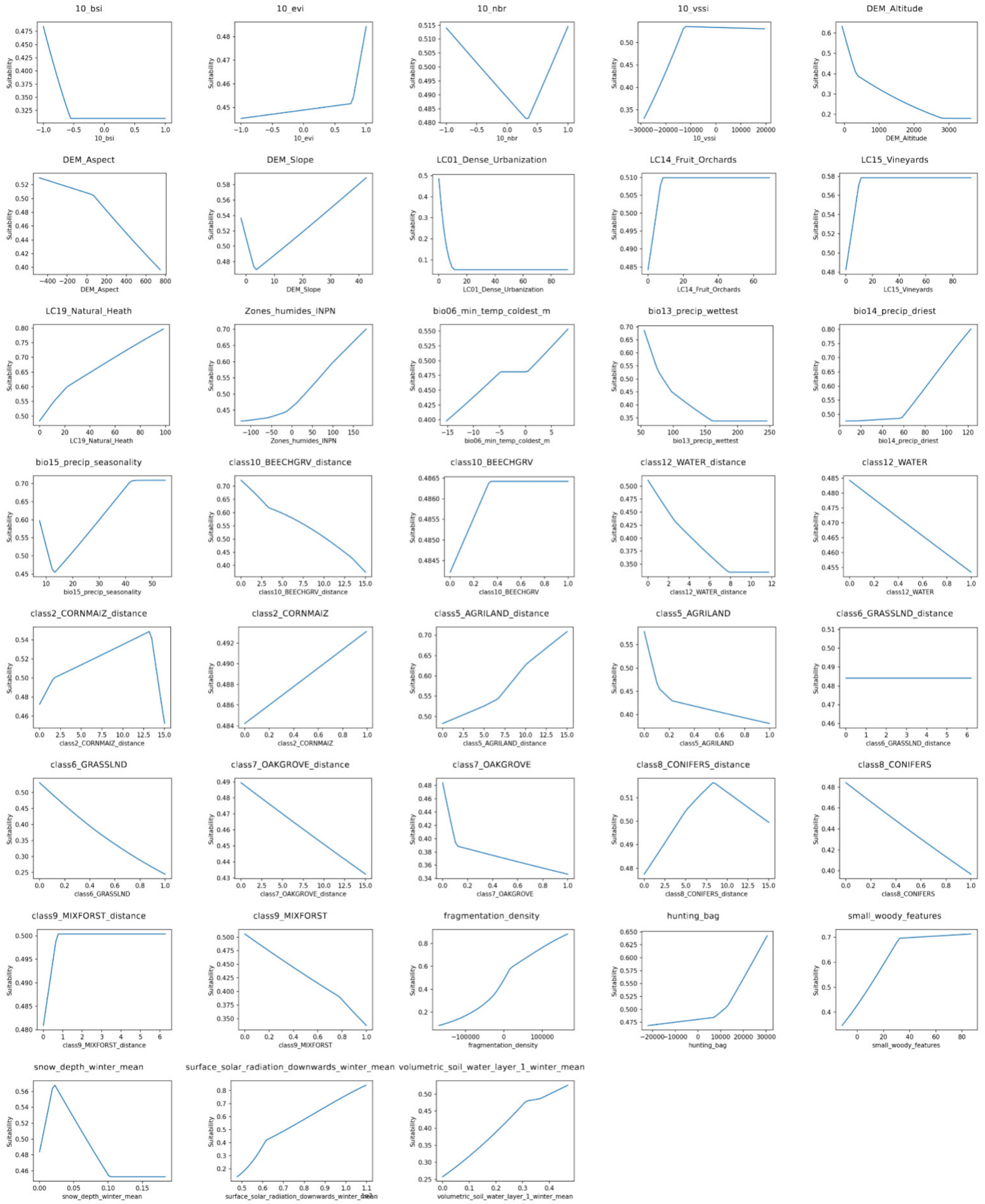


Fig. S7.1: Multitemporal winter model variable response curves (Typical Example); Feature scalar (x-axis) vs. Difference in Habitat Suitability Index (y-axis).

Monotemporal Winter Model

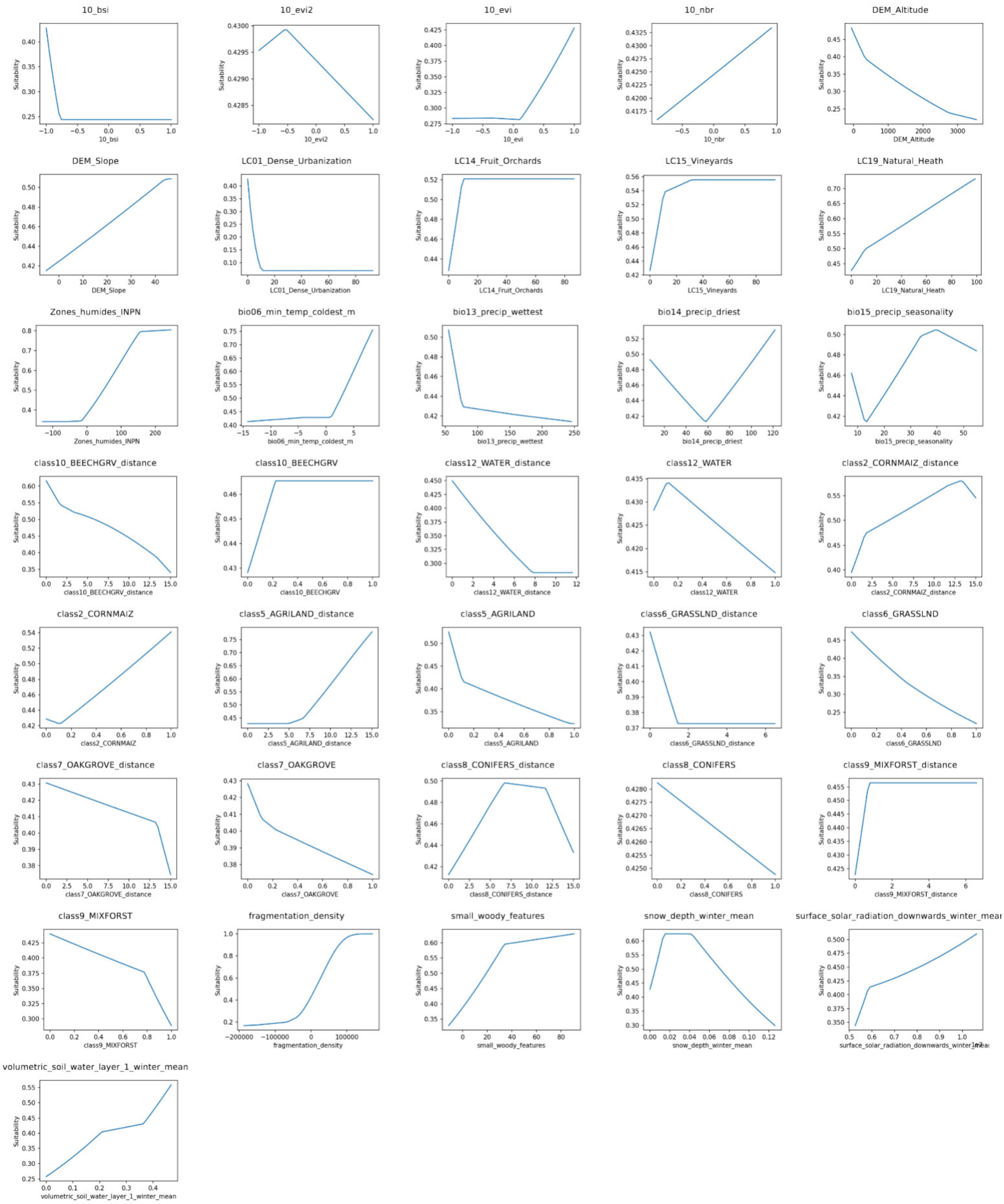


Fig. S7.2: Monotemporal winter model variable response curves (Typical Example); Feature scalar (x-axis) vs. Difference in Habitat Suitability Index (y-axis).

Multitemporal Summer Model

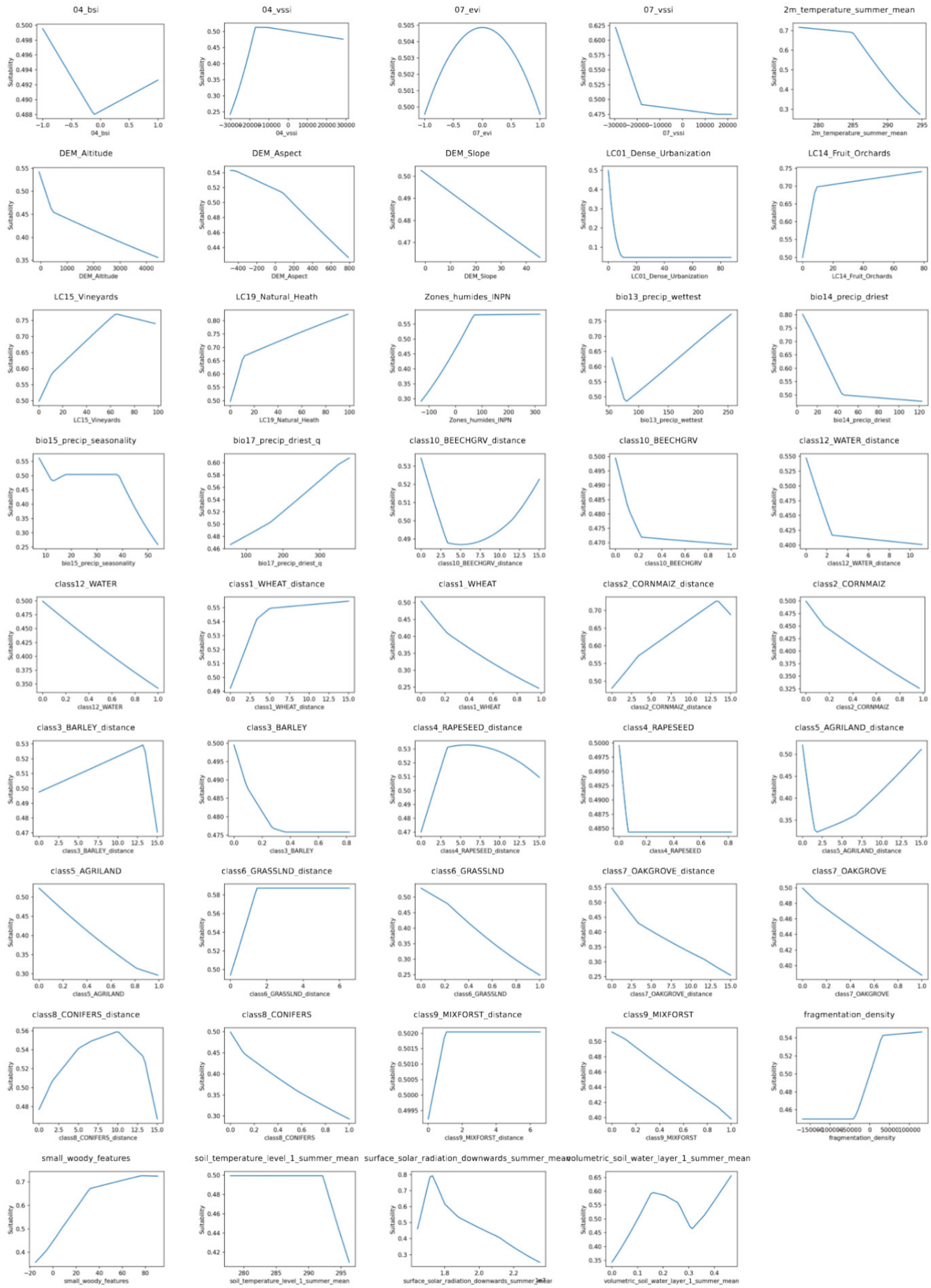


Fig. S7.3: Multitemporal summer model variable response curves (Typical Example); Feature scalar (x-axis) vs. Difference in Habitat Suitability Index (y-axis).

Monotemporal Summer Model

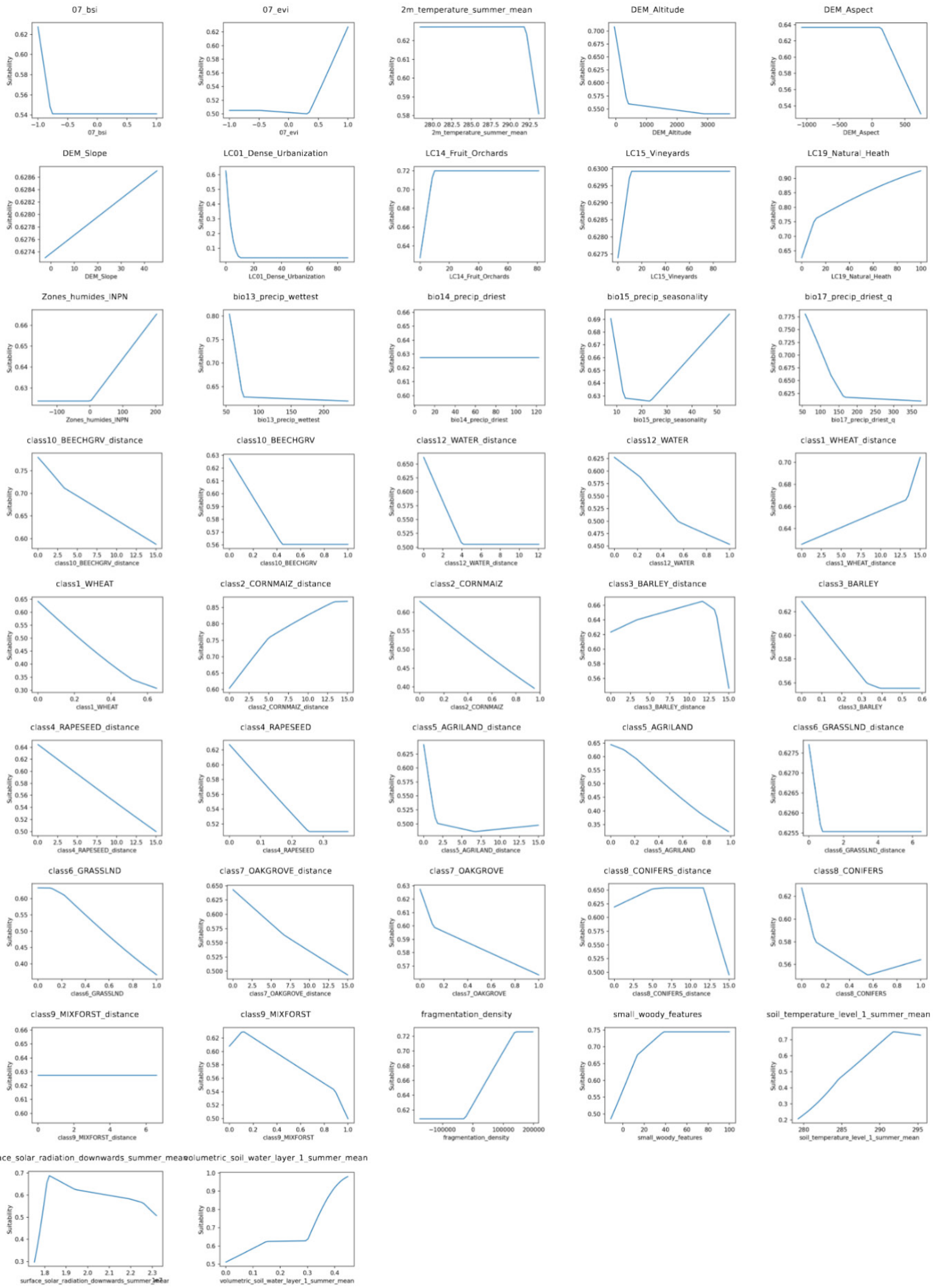


Fig. S7.4: Monotemporal summer model variable response curves (Typical Example); Feature scalar (x-axis) vs. Difference in Habitat Suitability Index (y-axis).

S8 – SDM Output Maps

Cloglog-Modulated Maps

Winter Multitemporal All Points MaxEnt

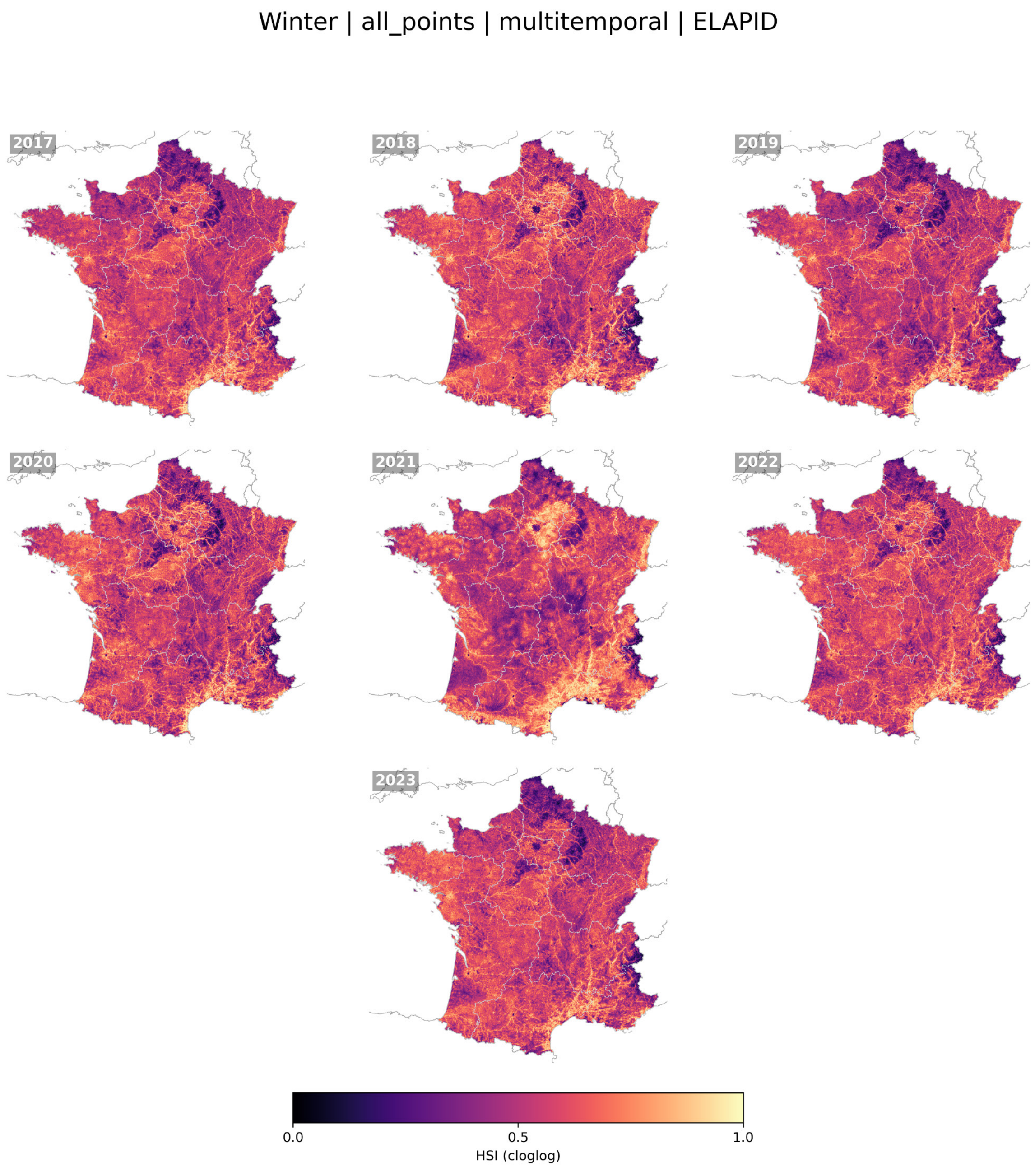


Fig. S8.1 – Winter habitat suitability (cloglog output) from multitemporal MaxEnt calibrated on all occurrence records, projected to independent test data of the annotated year.

Winter Monotemporal All Points MaxEnt

Winter | all_points | monotemporal | ELAPID

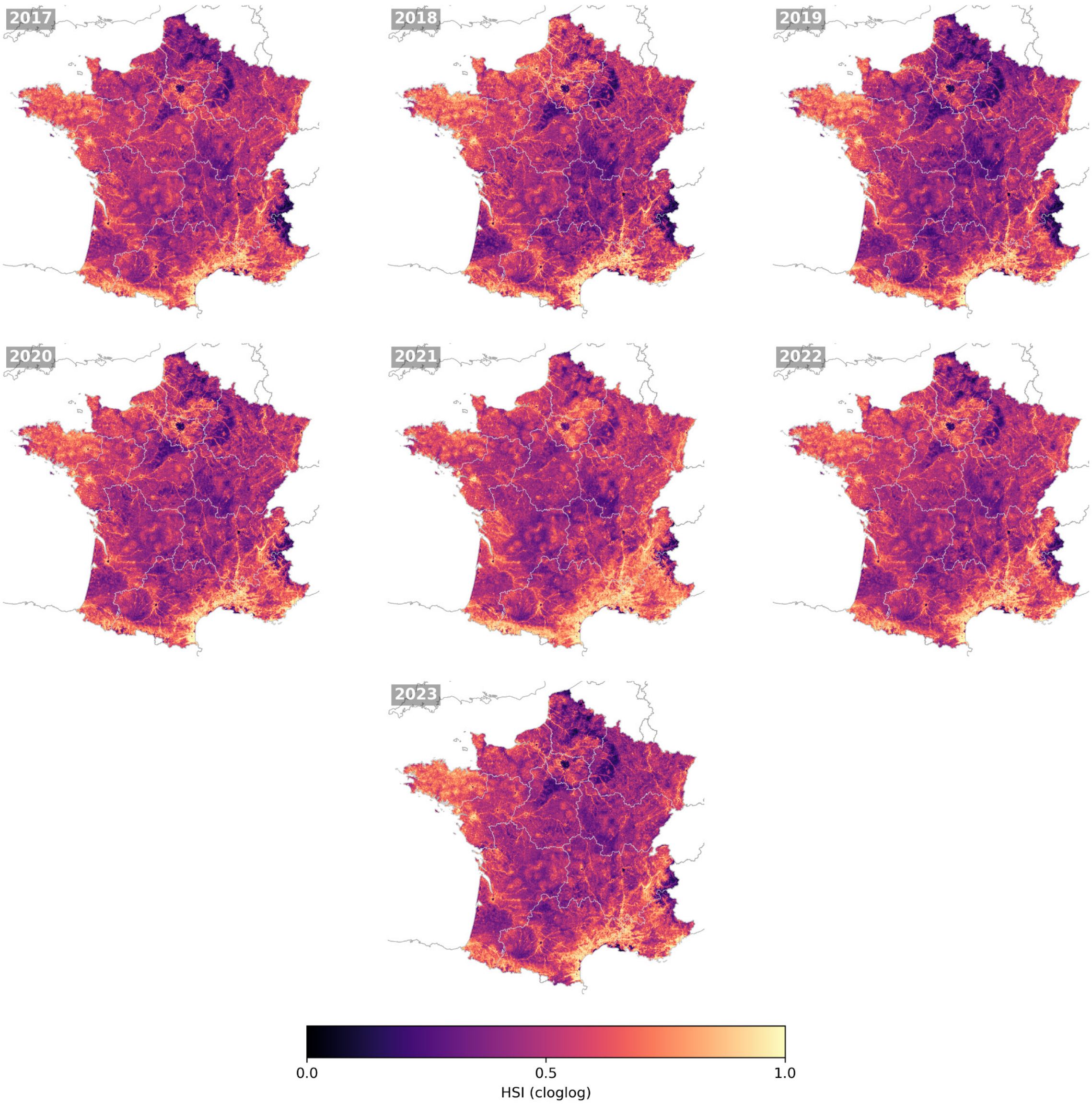


Fig. S8.2 – Winter habitat suitability (cloglog output) from monotemporal MaxEnt calibrated on all occurrence records, projected to independent test data of the annotated year.

Winter Multitemporal GBIF only MaxEnt

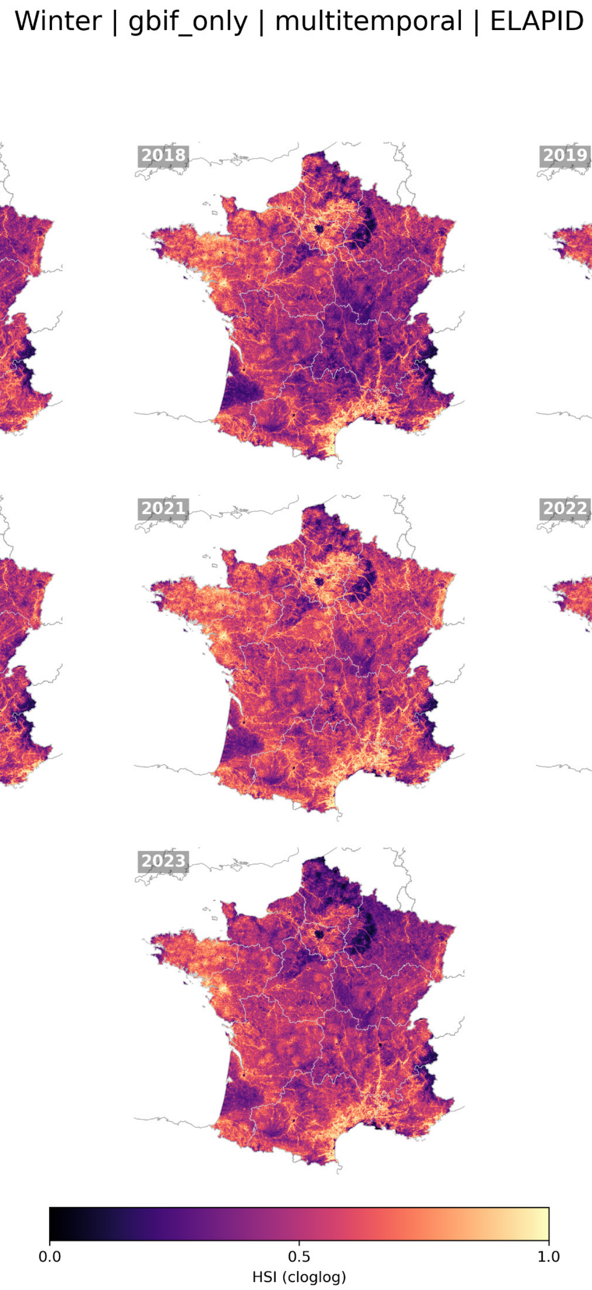


Fig. S8.3 – Winter habitat suitability (cloglog output) from multitemporal MaxEnt calibrated on GBIF occurrences only, projected to independent test data of the annotated year.

Winter Multitemporal GBM

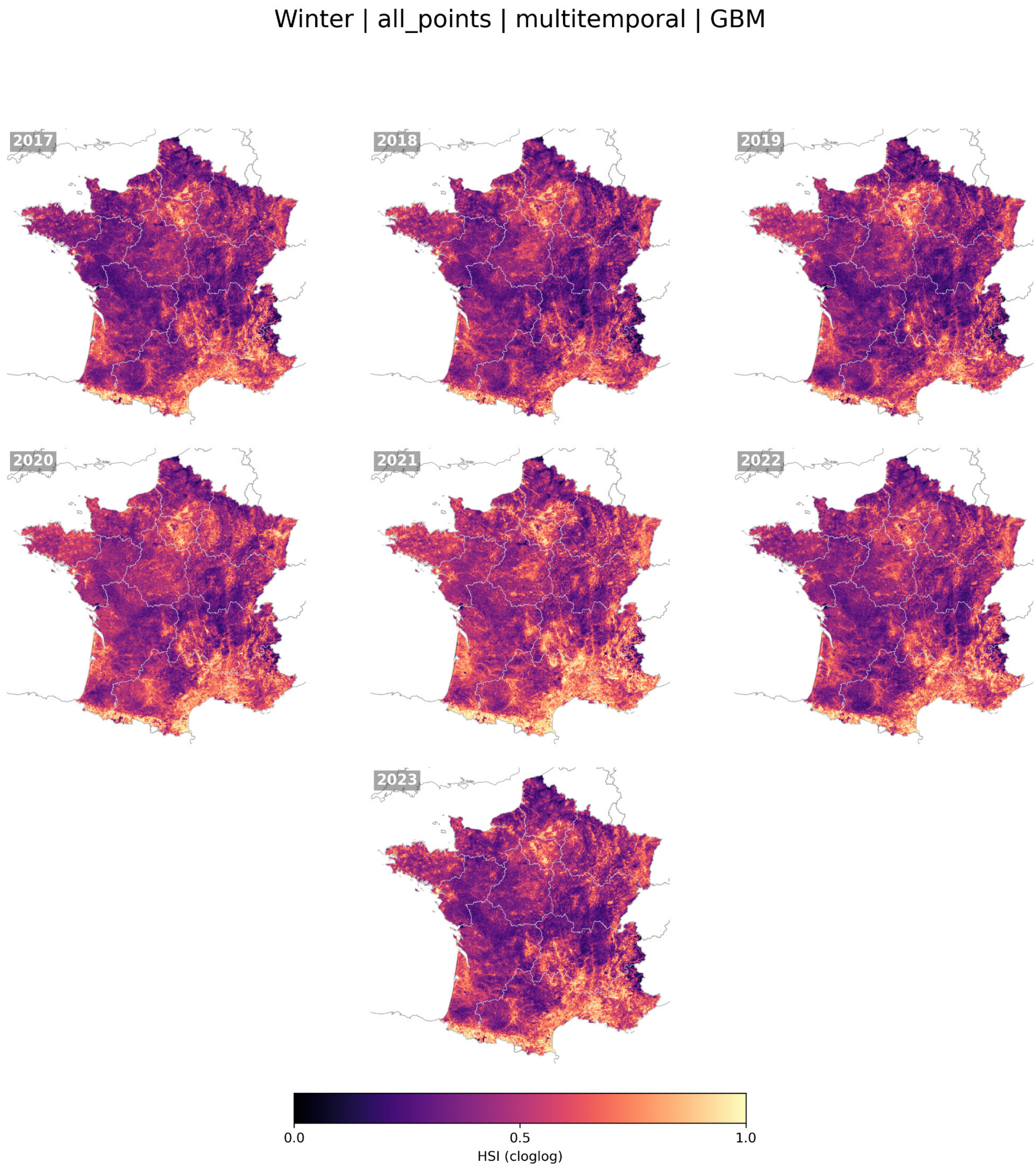


Fig. S8.4 – Winter habitat suitability predicted by multitemporal gradient boosting machine (GBM), projected to independent test data of the annotated year; transposed into MaxEnt style cloglog HSI.

Summer Multitemporal All Points MaxEnt

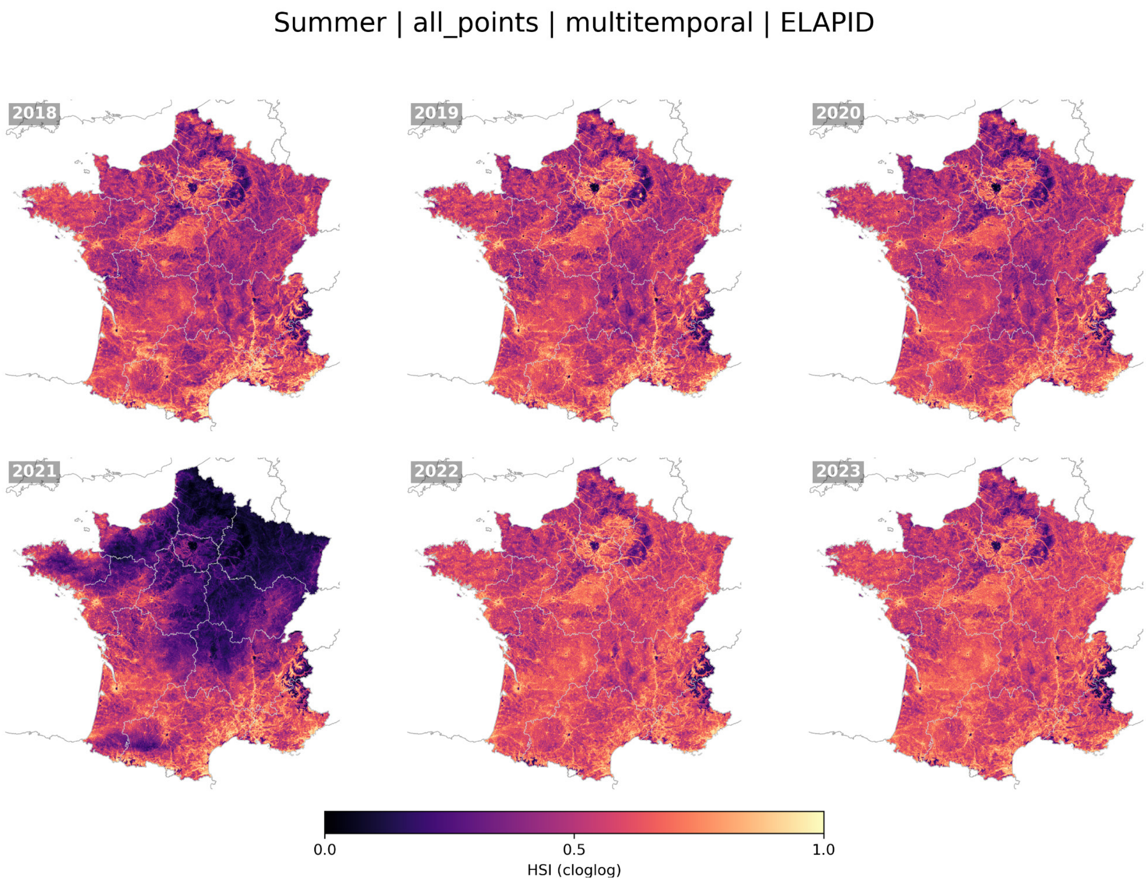


Fig. S8.5 – Summer habitat suitability (cloglog output) from multitemporal MaxEnt calibrated on all occurrence records, projected to independent test data of the annotated year.

Summer Monotemporal All Points MaxEnt

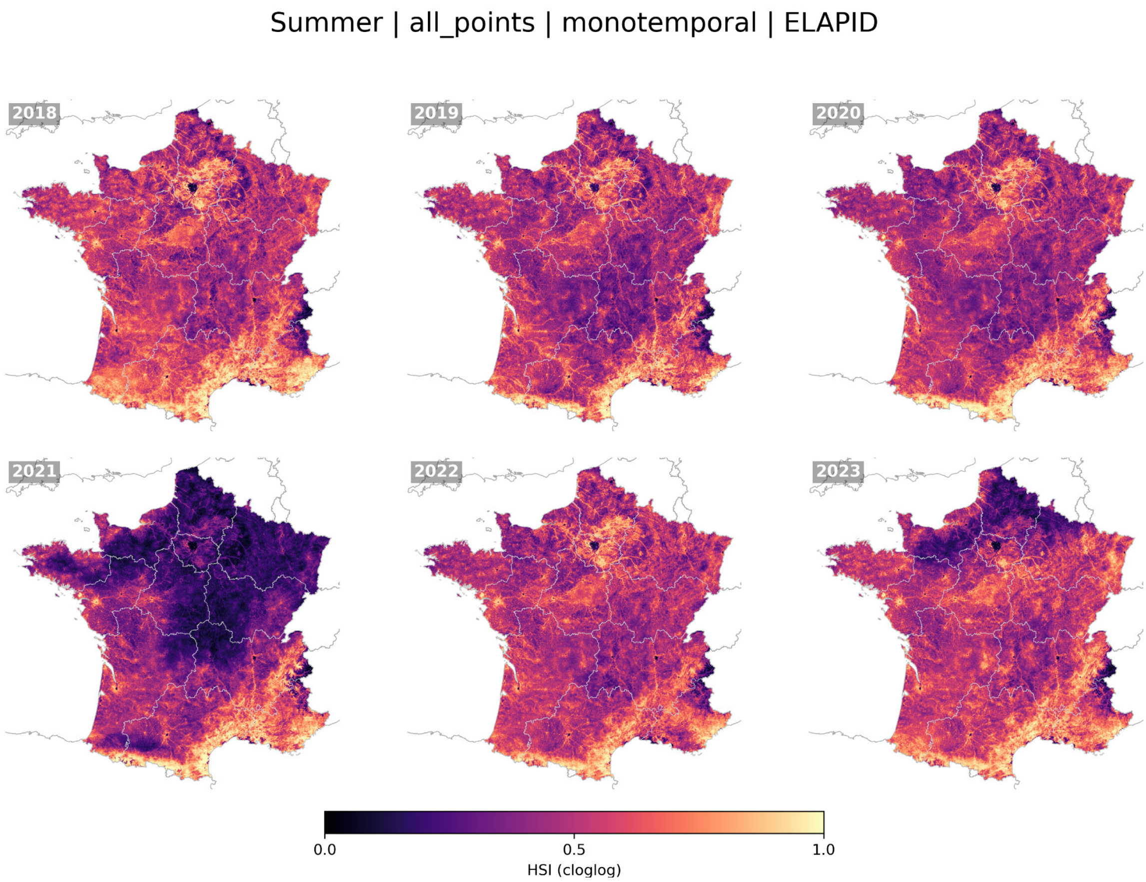


Fig. S8.6 – Summer habitat suitability (cloglog output) from monotemporal MaxEnt calibrated on all occurrence records, projected to independent test data of the annotated year.

Summer Multitemporal GBIF only MaxEnt

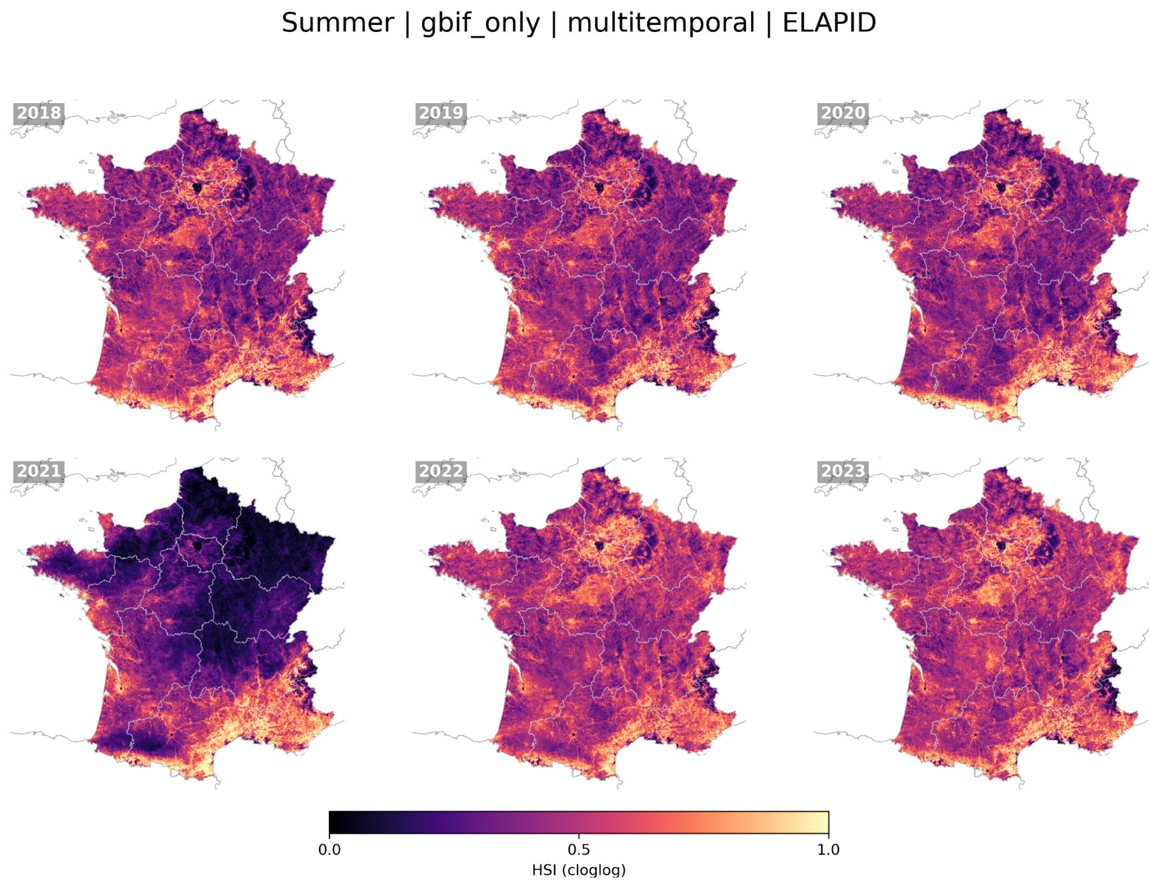


Fig. S8.7 – Summer habitat suitability (cloglog output) from multitemporal MaxEnt calibrated on GBIF occurrences only, projected to independent test data of the annotated year.

Summer Multitemporal GBM

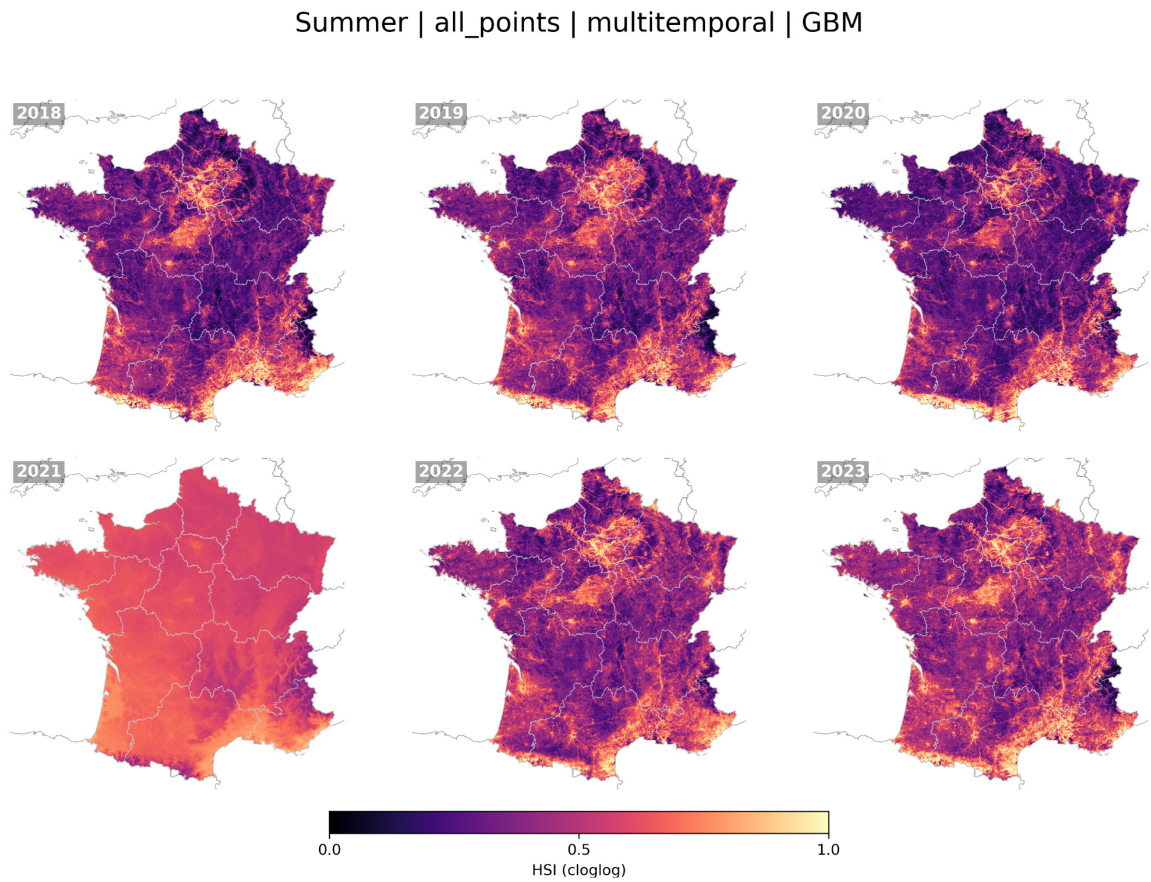


Fig. S8.8 – Summer habitat suitability predicted by multitemporal gradient boosting machine (GBM), projected to independent test data of the annotated year; transposed into MaxEnt style cloglog HSI.

S9 – Extrapolation Novelty

Table S9.1. Summer novelty summary.

Year	NT1 mono (%)	NT1 multi (%)	NT2 mono (%)	NT2 multi (%)	ΔNT1 (pp)	ΔNT2 (pp)
2018	3.71	0.05	0.80	0.92	-3.66	+0.12
2019	3.58	0.04	0.76	1.02	-3.54	+0.26
2020	2.53	0.03	0.72	0.88	-2.49	+0.16
2021	32.31	5.49	0.84	1.04	-26.83	+0.21
2022	5.20	1.24	0.75	0.90	-3.97	+0.15
2023	9.20	0.09	0.76	1.06	-9.11	+0.30

NT1 = fraction of AoI with MESS < 0; NT2 = fraction of AoI classified as NT2-only or NT1+NT2. Δ = multitemporal minus monotemporal (percentage points). AoI = 544,521 km².

Table S9.2. Winter novelty summary.

Year	NT1 mono (%)	NT1 multi (%)	NT2 mono (%)	NT2 multi (%)	ΔNT1 (pp)	ΔNT2 (pp)
2017	5.84	0.03	0.75	0.79	-5.81	+0.04
2018	1.69	0.23	0.74	0.69	-1.46	-0.06
2019	3.57	0.06	0.73	0.78	-3.52	+0.04
2020	0.18	0.06	0.69	0.71	-0.12	+0.02
2021	1.36	0.04	0.72	0.75	-1.32	+0.03
2022	1.67	0.03	0.71	0.74	-1.63	+0.03
2023	9.89	2.27	0.75	0.77	-7.61	+0.03

NT1 = fraction of AoI with MESS < 0; NT2 = fraction of AoI classified as NT2-only or NT1+NT2. Δ = multitemporal minus monotemporal (percentage points). AoI = 544,521 km².

Each figure below shows the spatial distribution of NT1 (MESS) and NT2 (ExDet) novelty for a given LOYO test year, comparing monotemporal and multitemporal training schemes.

Summer

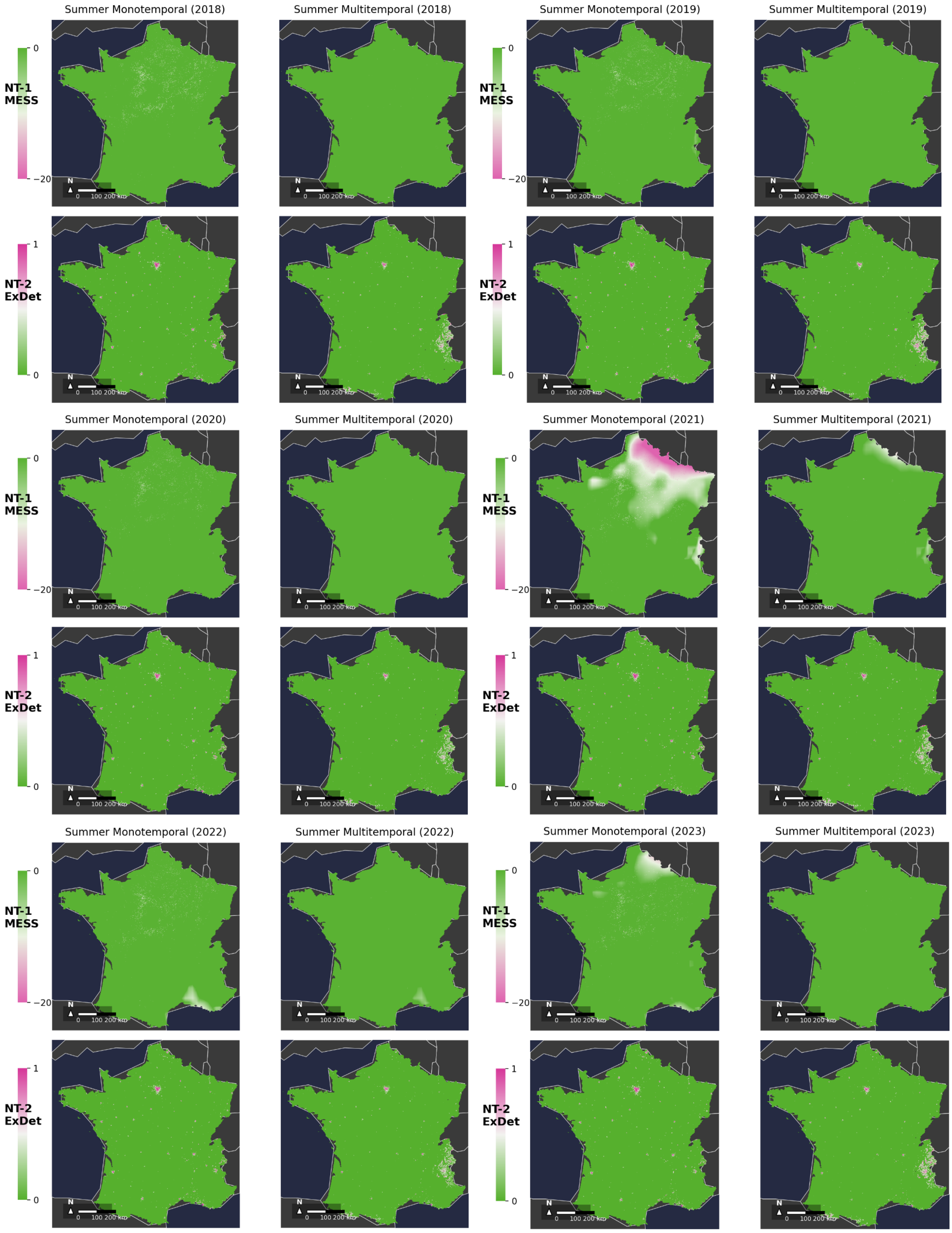


Figure S9.3. Summer LOYO test years.

Winter

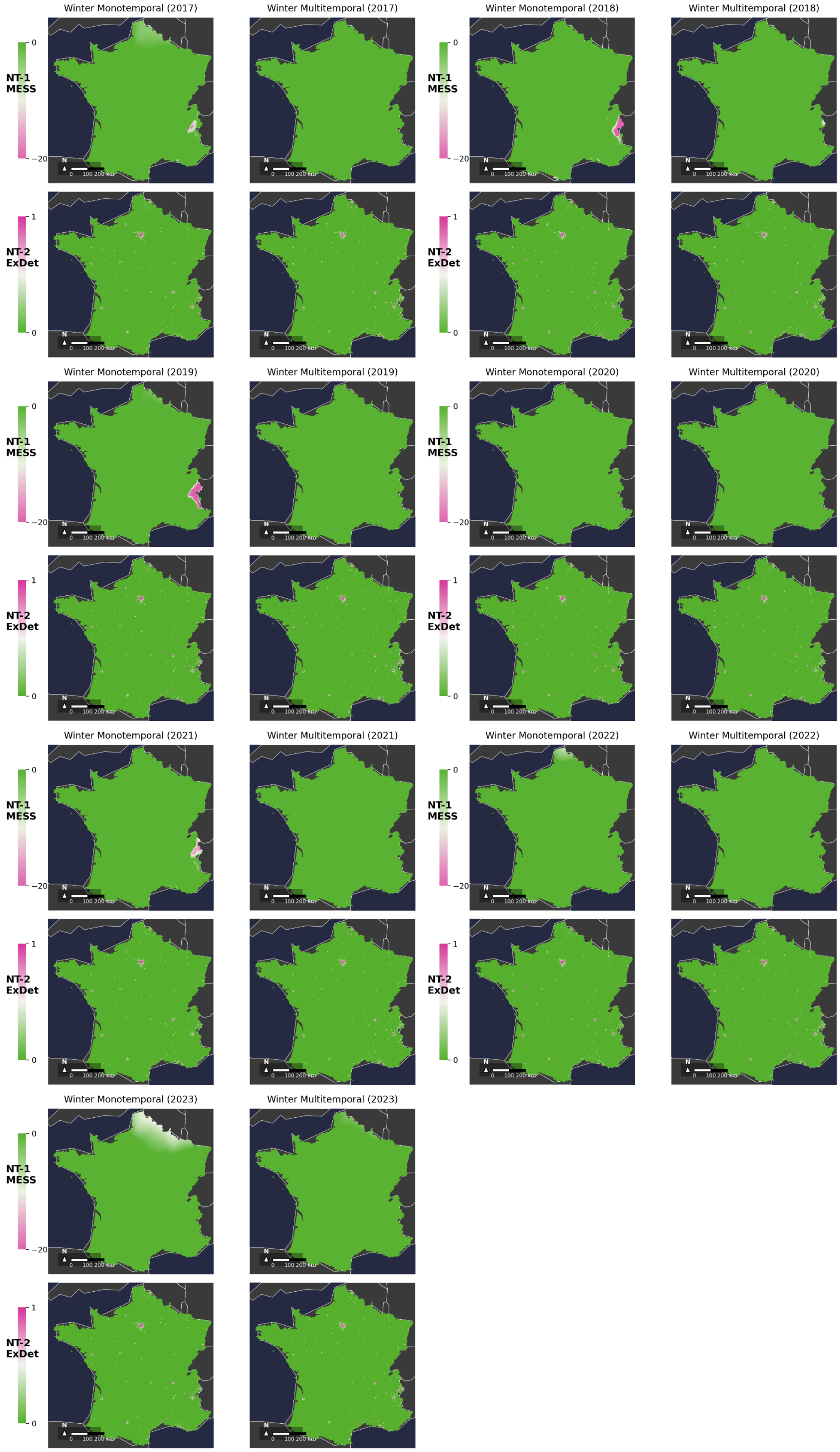


Figure S9.4. Winter LOYO test years.

Supplementary References

Abaigar T, Barrio GD, Vericad JR (1994) Habitat preference of wild boar (Sus scrofa L., 1758) in a mediterranean environment. Indirect evaluation by signs. 58:201–210. https://doi.org/10.1515/mamm.1994.58.2.201

Acevedo P, Escudero MA, Muńoz R, Gortázar C (2006) Factors affecting wild boar abundance across an environmental gradient in Spain. Acta Theriol (Warsz) 51:327–336. https://doi.org/10.1007/BF03192685

Aiello-Lammens M, Boria R, Radosavljevic A, et al (2015) spThin: An R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography 38:541–545. https://doi.org/10.1111/ecog.01132

Baker DJ, Maclean IMD, Gaston KJ (2024) Effective strategies for correcting spatial sampling bias in species distribution models without independent test data. Divers Distrib 30:e13802. https://doi.org/10.1111/ddi.13802

Baker DJ, Maclean IMD, Goodall M, Gaston KJ (2022) Correlations between spatial sampling biases and environmental niches affect species distribution models. Glob Ecol Biogeogr 31:1038–1050. https://doi.org/10.1111/geb.13491

Ballari SA, Barrios-García MN (2014) A review of wild boar Sus scrofa diet and factors affecting food selection in native and introduced ranges. Mammal Rev 44:124–134. https://doi.org/10.1111/mam.12015

Barasona JA, Carpio A, Boadella M, et al (2021) Expansion of native wild boar populations is a new threat for semi-arid wetland areas. Ecol Indic 125:107563. https://doi.org/10.1016/j.ecolind.2021.107563

Barber RA, Ball SG, Morris RKA, Gilbert F (2022) Target-group backgrounds prove effective at correcting sampling bias in Maxent models. Divers Distrib 28:128–141. https://doi.org/10.1111/ddi.13442

Barbet-Massin M, Jiguet F, Albert CH, Thuiller W (2012) Selecting pseudo-absences for species distribution models: how, where and how many? Methods Ecol Evol 3:327–338. https://doi.org/10.1111/j.2041-210X.2011.00172.x

Beaugrand G, Lenoir S, Ibañez F, Manté C (2011) A new model to assess the probability of occurrence of a species, based on presence-only data. Mar Ecol Prog Ser 424:175–190. https://doi.org/10.3354/meps08939

Beck J, Böller M, Erhardt A, Schwanghart W (2014) Spatial bias in the GBIF database and its effect on modeling species' geographic distributions. Ecol Inform 19:10–15. https://doi.org/10.1016/j.ecoinf.2013.11.002

Benard A (2023) Road ecology: toward a predictive model of wildlife-vehicle collisions. Univ Claude Bernard Lyon 1

Bergqvist G, Paulson S, Elmhagen B (2018) Effects of female body mass and climate on reproduction in northern wild boar. Wildl Biol 2018:wlb.00421. https://doi.org/10.2981/wlb.00421

Bisi F, Chirichella R, Chianucci F, et al (2018) Climate, tree masting and spatial behaviour in wild boar (Sus scrofa L.): insight from a long-term study. Ann For Sci 75:46. https://doi.org/10.1007/s13595-018-0726-6

Bosch J, Iglesias I, Martínez M, Torre A de la (2020) Climatic and topographic tolerance limits of wild boar in Eurasia: implications for their expansion. Geogr Environ Sustain 13:107–114. https://doi.org/10.24057/2071-9388-2019-52

Brandt S, Baubet E, Vassant J, Servanty S (2006) Régime alimentaire du sanglier en milieu forestier de plaine agricole. Faune Sauvage 273:20–27

Bruinderink GWTAG, Hazebroek E (1995) Modelling carrying capacity for wild boar Sus scrofa scrofa in a forest/heathland ecosystem. Wildl Biol 1:81–87. https://doi.org/10.2981/wlb.1995.0013

Cahill S, Llimona F, Cabañeros L, Calomardo F (2012) Characteristics of wild boar (Sus scrofa) habituation to urban areas in the Collserola Natural Park (Barcelona) and comparison with other locations. Anim Biodivers Conserv 35:221–233. https://doi.org/10.32800/abc.2012.35.0221

Calenge C, Maillard D, Fournier P, Fouque C (2004) Efficiency of spreading maize in the garrigues to reduce wild boar (Sus scrofa) damage to Mediterranean vineyards. Eur J Wildl Res 50:112–120

Calenge C, Maillard D, Vassant J, Brandt S (2002) Summer and hunting season home ranges of wild boar (Sus scrofa) in two habitats in France. Game Wildl Sci 19 (4):281–301

Cantelaube P, Carles M (2014) Le registre parcellaire graphique : des données géographiques pour décrire la couverture du sol agricole. NOVAE - Ingénierie Savoir-Faire Innovants 58–64

Castillo-Contreras R, Carvalho J, Serrano E, et al (2018) Urban wild boars prefer fragmented areas with food resources near natural corridors. Sci Total Environ 615:282–288. https://doi.org/10.1016/j.scitotenv.2017.09.277

Chamon LF, Ribeiro A (2017) Greedy sampling of graph signals. IEEE Trans Signal Process 66:34–47

Claverie M, Ju J, Masek JG, et al (2018) The Harmonized Landsat and Sentinel-2 surface reflectance data set. Remote Sens Environ 219:145–161. https://doi.org/10.1016/j.rse.2018.09.002

Couriot OH, Cameron MD, Joly K, et al (2023) Continental synchrony and local responses: Climatic effects on spatiotemporal patterns of calving in a social ungulate. Ecosphere 14:e4399. https://doi.org/10.1002/ecs2.4399

Croft S, Warren D, Blanco-Aguiar J, Smith G (2025) Predicting the distribution of common wild mammal species across Europe - are there sufficient occurrence data? Eur J Wildl Res 71:133. https://doi.org/10.1007/s10344-025-02014-2

Cutini A, Chianucci F, Chirichella R, et al (2013) Mast seeding in deciduous forests of the northern Apennines (Italy) and its influence on wild boar population dynamics. Ann For Sci 70:493–502. https://doi.org/10.1007/s13595-013-0282-z

de Oliveira G, Rangel TF, Lima-Ribeiro MS, et al (2014) Evaluating, partitioning, and mapping the spatial autocorrelation component in ecological niche modeling: a new approach based on environmentally equidistant records. Ecography 37:637–647. https://doi.org/10.1111/j.1600-0587.2013.00564.x

Dubos N, Préau C, Lenormand M, et al (2022) Assessing the effect of sample bias correction in species distribution models. Ecol Indic 145:109487. https://doi.org/10.1016/j.ecolind.2022.109487

EEA/FOEN (2011) Landscape fragmentation in Europe. European Environment Agency (EEA), Swiss Federal Office for the Environment (FOEN), Copenhagen, Denmark

Eldar Y, Lindenbaum M, Porat M, Zeevi YY (1997) The farthest point strategy for progressive image sampling. IEEE Trans Image Process 6:1305–1315

Enetwild Consortium, Croft S, Smith G, et al (2019) Wild boar in focus: initial model outputs of wild boar distribution based on occurrence data and identification of priority areas for data collection. EFSA Support Publ 16:1533E

Etherington TR (2019) Mahalanobis distances and ecological niche modelling: correcting a chi-squared probability error. PeerJ 7:e6678. https://doi.org/10.7717/peerj.6678

Faucqueur L, Morin N, Masse A, et al (2019) A new Copernicus high resolution layer at pan-European scale: small woody features. In: Remote Sensing for Agriculture, Ecosystems, and Hydrology XXI. SPIE, pp 268–278

Ficetola GF, Bonardi A, Mairota P, et al (2014) Predicting wild boar damages to croplands in a mosaic of agricultural and natural areas. Curr Zool 60:170–179. https://doi.org/10.1093/czoolo/60.2.170

Fourcade Y, Engler JO, Rödder D, Secondi J (2014) Mapping Species Distributions with MAXENT Using a Geographically Biased Sample of Presence Data: A Performance Assessment of Methods for Correcting Sampling Bias. PLOS ONE 9:e97122. https://doi.org/10.1371/journal.pone.0097122

Fournier-Chambrillon Ch, Maillard D, Fournier P (1995) Diet of the wild boar (Sus scrofa L.) inhabiting the Montpellier garrigue. J Mt Ecol 3:174–179

Gallardo B, Zieritz A, Aldridge DC (2015) The Importance of the Human Footprint in Shaping the Global Distribution of Terrestrial, Freshwater and Marine Invaders. PLoS ONE 10:e0125801. https://doi.org/10.1371/journal.pone.0125801

Giménez-Anaya A, Herrero J, Rosell C, et al (2008) Food habits of wild boars (Sus Scrofa) in a mediterranean coastal wetland. Wetlands 28:197–203. https://doi.org/10.1672/07-18.1

Hansen MC, Potapov PV, Moore R, et al (2013) High-Resolution Global Maps of 21st-Century Forest Cover Change. Science 342:850–853. https://doi.org/10.1126/science.1244693

Hughes AC, Orr MC, Ma K, et al (2021) Sampling biases shape our view of the natural world. Ecography 44:1259–1269. https://doi.org/10.1111/ecog.05926

Inglada J, Vincent A, Arias M, et al (2017) Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series. Remote Sens 9:95. https://doi.org/10.3390/rs9010095

James CN, Coverdale B, Shivambu TC, Downs CT (2023) Potential climatic suitability distribution model for wild boar (Sus scrofa) in South Africa. Afr J Wildl Res 53:. https://doi.org/10.3957/056.053.0185

Jiménez L, Soberón J (2022) Estimating the fundamental niche: Accounting for the uneven availability of existing climates in the calibration area. Ecol Model 464:109823. https://doi.org/10.1016/j.ecolmodel.2021.109823

Keuling O, Stier N, Roth M (2008) How does hunting influence activity and spatial usage in wild boar Sus scrofa L.? Eur J Wildl Res 54:729–737. https://doi.org/10.1007/s10344-008-0204-9

Leaper R, Massei G, Gorman ML, Aspinall R (1999) The feasibility of reintroducing Wild Boar (Sus scrofa) to Scotland. Mammal Rev 29:239–258. https://doi.org/10.1046/j.1365-2907.1999.2940239.x

Lee O, Schlichting PE, Jo Y-S (2022) Habitat model for wild boar (Sus scrofa) in Bukhansan National Park, Seoul. J Urban Ecol 8:juac027. https://doi.org/10.1093/jue/juac027

Lobo JM, Jiménez-Valverde A, Hortal J (2010) The uncertain nature of absences and their importance in species distribution modelling. Ecography 33:103–114. https://doi.org/10.1111/j.1600-0587.2009.06039.x

Marin C, Couderchet L, Le Campion G, Werno J (2024) Wildlife and the city. Modelling wild boar use of urban nature: Empirical contribution, methodological proposal. Urban Ecosyst 27:1291–1312. https://doi.org/10.1007/s11252-024-01510-8

Meyer C, Weigelt P, Kreft H (2016) Multidimensional biases, gaps and uncertainties in global plant occurrence information. Ecol Lett 19:992–1006. https://doi.org/10.1111/ele.12624

Morelle K, Lehaire F, Lejeune P (2013) Spatio-temporal patterns of wildlife-vehicle collisions in a region with a high-density road network. Nat Conserv 5:53–73. https://doi.org/10.3897/natureconservation.5.4634

Muñoz-Sabater J, Dutra E, Agustí-Panareda A, et al (2021) ERA5-Land: a state-of-the-art global reanalysis dataset for land applications. Earth Syst Sci Data 13:4349–4383. https://doi.org/10.5194/essd-13-4349-2021

Phillips SJ, Dudík M (2008) Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. Ecography 31:161–175. https://doi.org/10.1111/j.0906-7590.2008.5203.x

Preparata FP, Shamos MI (1985) Convex Hulls: Basic Algorithms. In: Preparata FP, Shamos MI (eds) Computational Geometry: An Introduction. Springer, New York, NY, pp 95–149

Puissant A, Sellé A, Baghdadi N, et al (2019) The ‘urban’ component of the French Land Data and Services Centre (THEIA). In: 2019 Joint Urban Remote Sensing Event (JURSE). pp 1–4

Rapinel S, Panhelleux L, Lemercier B, et al (2023) Projet de cartographie nationale des milieux humides – Rapport méthodologique de la prélocalisation des milieux et des zones humides, de la cartographie des habitats et de l’évaluation des fonctions. Patrinat (OFB-MNHN-CNRS-IRD)

Rosell C, Navàs F, Romero S (2012) Reproduction of wild boar in a cropland and coastal wetland area: implications for management. Anim Biodivers Conserv 35:209–217. https://doi.org/10.32800/abc.2012.35.0209

Rosell C, Navàs F, Romero S, de Dalmases I (2004) Activity patterns and social organization of wild boar (Sus scrofa, L.) in a wetland environment: preliminary data on the effects of shooting individuals. Galemys 16:157–166. https://doi.org/10.7325/Galemys.2004.NE.A13

Sabrina S, Jean-Michel G, Carole T, et al (2009) Pulsed resources and climate-induced variation in the reproductive traits of wild boar under high hunting pressure. J Anim Ecol 78:1278–1290. https://doi.org/10.1111/j.1365-2656.2009.01579.x

Sanderson EW, Jaiteh M, Levy MA, et al (2002) The Human Footprint and the Last of the Wild: The human footprint is a global map of human influence on the land surface, which suggests that human beings are stewards of nature, whether we like it or not. BioScience 52:891–904. https://doi.org/10.1641/0006-3568(2002)052%5B0891:THFATL%5D2.0.CO;2

Schley L, Roper TJ (2003) Diet of wild boar Sus scrofa in Western Europe, with particular reference to consumption of agricultural crops. Mammal Rev 33:43–56. https://doi.org/10.1046/j.1365-2907.2003.00010.x

Schwartz ALW, Shilling FM, Perkins SE (2020) The value of monitoring wildlife roadkill. Eur J Wildl Res 66:18. https://doi.org/10.1007/s10344-019-1357-4

Sicacha-Parada J, Steinsland I, Cretois B, Borgelt J (2019) Accounting for spatial varying sampling effort due to accessibility in Citizen Science data: A case study of moose in Norway

Sillero N, Barbosa AM (2021) Common mistakes in ecological niche models. Int J Geogr Inf Sci 35:213–226. https://doi.org/10.1080/13658816.2020.1798968

Stillfried M, Gras P, Börner K, et al (2017) Secrets of Success in a Landscape of Fear: Urban Wild Boar Adjust Risk Perception and Tolerate Disturbance. Front Ecol Evol 5:. https://doi.org/10.3389/fevo.2017.00157

Syfert MM, Smith MJ, Coomes DA (2013) The Effects of Sampling Bias and Model Complexity on the Predictive Performance of MaxEnt Species Distribution Models. PLOS ONE 8:e55158. https://doi.org/10.1371/journal.pone.0055158

Thurfjell H, Ball JP, Åhlén P-A, et al (2009) Habitat use and spatial patterns of wild boar Sus scrofa (L.): agricultural fields and edges. Eur J Wildl Res 55:517–523. https://doi.org/10.1007/s10344-009-0268-1

Thurfjell H, Spong G, Ericsson G (2013) Effects of hunting on wild boar Sus scrofa behaviour. Wildl Biol 19:87–93. https://doi.org/10.2981/12-027

Tong R, Yesson C, Yu J, et al (2023) Key factors for species distribution modeling in benthic marine environments. Front Mar Sci 10:. https://doi.org/10.3389/fmars.2023.1222382

USGS (2018) SRTM 1 Arc-Second Global

Vanschoenwinkel B, Waterkeyn A, Vandecaetsbeek T, et al (2008) Dispersal of freshwater invertebrates by large terrestrial mammals: a case study with wild boar (Sus scrofa) in Mediterranean wetlands. Freshw Biol 53:2264–2273. https://doi.org/10.1111/j.1365-2427.2008.02071.x

Vetter SG, Puskas Z, Bieber C, Ruf T (2020) How climate change and wildlife management affect population structure in wild boars. Sci Rep 10:7298. https://doi.org/10.1038/s41598-020-64216-9

Welander J (2000) Spatial and temporal dynamics of wild boar (Sus scrofa) rooting in a mosaic landscape. J Zool 252:263–271. https://doi.org/10.1111/j.1469-7998.2000.tb00621.x