MIRANDA: MId-feature RANk-adversarial Domain Adaptation toward climate change-robust ecological forecasting with deep learning

Supplementary Material

1. Implementation details

We follow the implementation of PhenoFormer [15] as our backbone and use two transformer encoder layers (t_1 and t_2). All models are trained using the MSE loss and optimized with Adam [20] with a learning rate of 10^{-4}, batch size of 16, and up to 300 epochs. All domain adaptation methods adopt PhenoFormer as the backbone. For approaches involving adversarial training, the discriminator p is implemented as a three-layer MLP with ReLU activations. Model-specific implementation details are provided below.

- **DANN** [1]: We adopt Phenoformer as the backbone and attach a domain discriminator to the high-level features extracted from the last transformer encoder layer. The discriminator is trained with the standard binary domain classification objective, while the feature extractor is optimized adversarially via a gradient reversal layer, following the original DANN formulation.
- **ADDA** [37]: We initialize ADDA with the pretrained Phenoformer and follow the same experimental protocol as the other adaptation methods. Specifically, we fine-tune the encoder and train a domain discriminator to align feature representations in an adversarial manner.
- **CORAL** [35]: We apply the CORAL loss to the high-level features extracted from the last transformer encoder layer, prior to the regression head, using the original formulation of the CORAL objective. For a fair comparison, we set the CORAL weighting coefficient λ to be the same as that used for our $\mathcal{L}_{\text{rank}}$.
- **DANL** [30]: The original DANL method introduces domain-agnostic normalization for batch normalization layers in convolutional architectures under the DANN framework. Since Phenoformer is transformer-based and relies on layer normalization, we adapt DANL by replacing the standard layer normalization layers in the transformer encoder with our domain-agnostic layer normalization, while keeping the rest of the DANN training procedure unchanged.
- **AdaBN** [22]: AdaBN was originally proposed for models with batch normalization, where target-domain statistics are used during inference. To incorporate AdaBN into Phenoformer, we insert a BatchNorm1d layer before the linear decoder and update its running statistics using unlabeled target samples during adaptation.
- **M1** [4]: We use the implementation of M1 in the R package, Phenor [19] and follow the same implementation setup specified in [15].
- **MIRANDA** : We set $\tau = 0.1$ in $\mathcal{L}_{\text{rank}}$ and adopt the dynamic λ scheduler from [1]. For the embeddings used for rank loss, we set the dimension to 128.

2. Additional experiment results

2.1. Guidance for rank-based adversarial training

We investigate difference guidance for rank-based adversarial training: besides using year information as guidance, we explore the effect of using annual temperature and elevation as the guidance.

Table 3. Different guidance for rank-based adversarial training under the Structured Temporal split.

Guidance	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
Annual temperature	0.50	9.01	6.85
Elevation	0.50	9.01	6.89
Year (ours)	**0.51**	**8.97**	**6.80**

Table 4. Different guidance for rank-based adversarial training under the Annual Temperature split.

Guidance	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
Annual temperature	0.43	9.44	7.18
Elevation	0.42	9.44	7.19
Year (ours)	**0.43**	**9.43**	**7.16**

Table 5. Different guidance for rank-based adversarial training under the Elevation split.

Guidance	$R^2 \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
Annual temperature	0.22	11.36	8.67
Elevation	0.19	11.55	8.86
Year (ours)	**0.22**	**11.32**	**8.63**

2.2. Additional qualitative results

Detailed per-species scatter plots of PhenoFormer and our method of elevation data split are shown in Figure 1 to 5 .

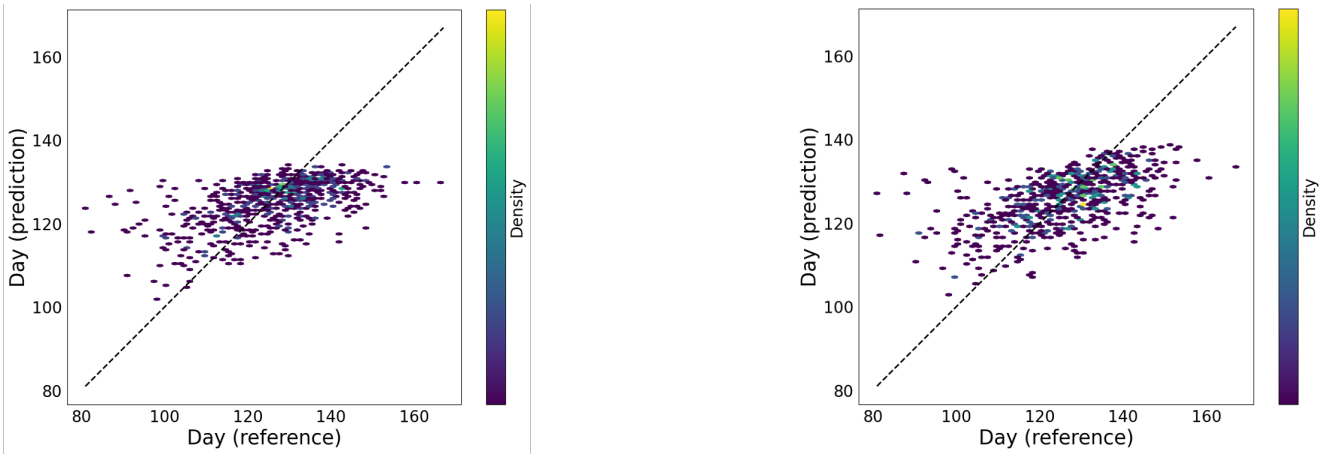

Figure 1. Scatter plot of predicted (y-axis) vs true dates (x-axis) **before (left, with R^2 = 24%) and after (right, with R^2 = 26%) adaptation** for phenology task leaf unholding of horse chestnut in elevation data split.

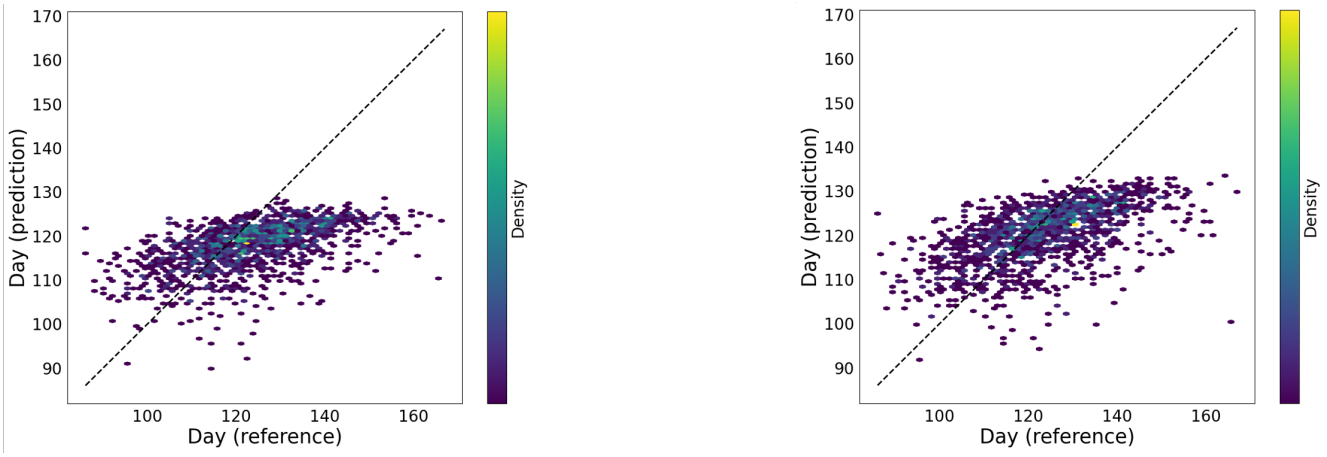

Figure 2. Scatter plot of predicted (y-axis) vs true dates (x-axis) **before (left, with R^2 = 12%) and after (right, with R^2 = 23%) adaptation** for phenology task needle emergence of European larch in elevation data split.

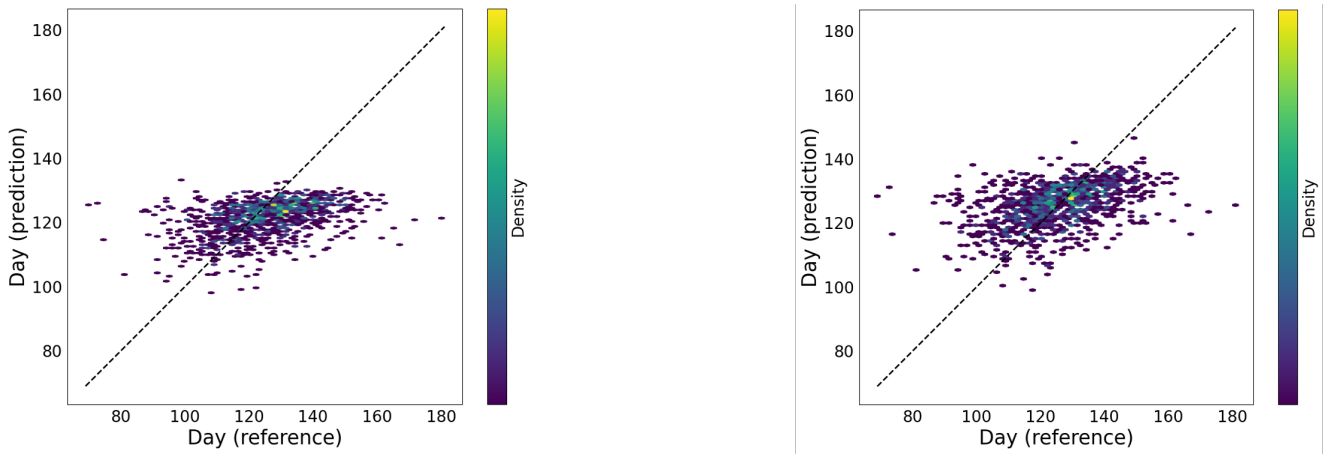

Figure 3. Scatter plot of predicted (y-axis) vs true dates (x-axis) **before (left, with R^2 = 12%) and after (right, with R^2 = 15%) adaptation** for phenology task leaf unholding of hazel in elevation data split.

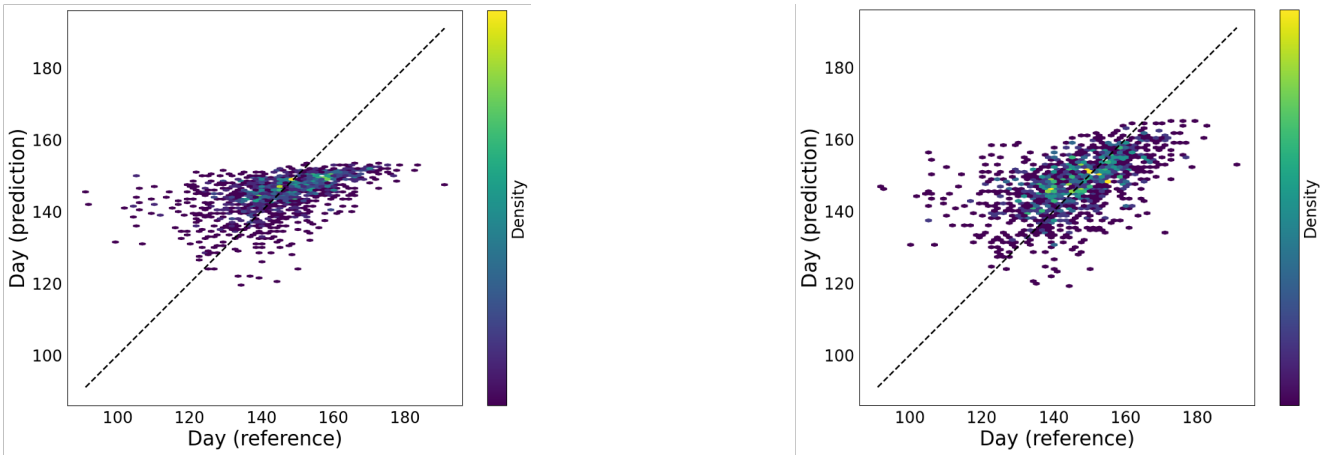

Figure 4. Scatter plot of predicted (y-axis) vs true dates (x-axis) **before (left, with R^2 = 25%) and after (right, with R^2 = 28%) adaptation** for phenology task needle emergence of common spruce in elevation data split.

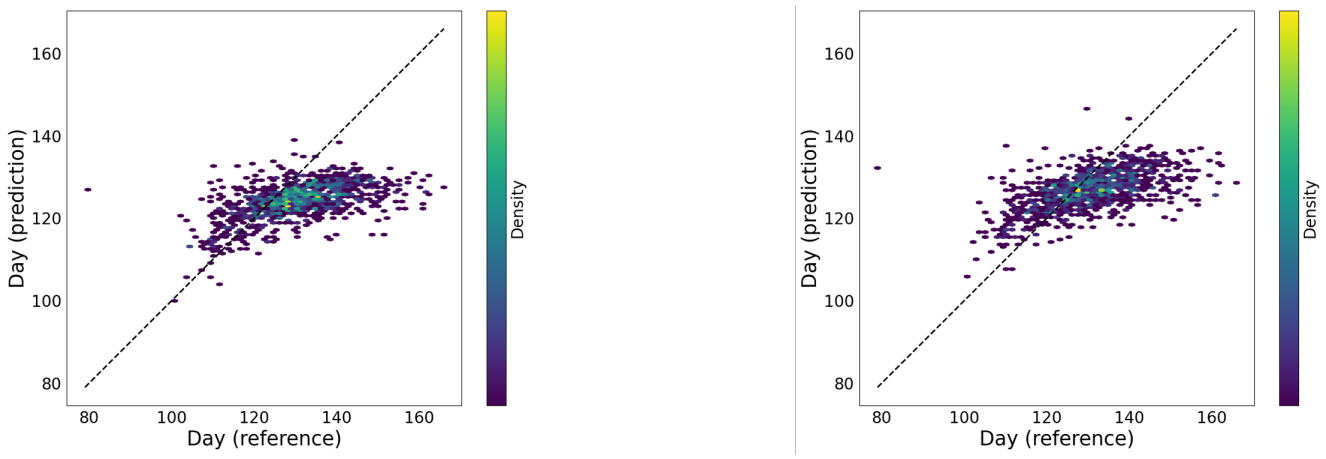

Figure 5. Scatter plot of predicted (y-axis) vs true dates (x-axis) **before (left, with R^2 = 13%) and after (right, with R^2 = 20%) adaptation** for phenology task leaf unfolding of European beech in elevation data split.