BirdSAT: Supplementary Material

A. Dataset

Data Preparation. We downloaded the entire iNaturalist 2021 dataset from the iNAT Competition GitHub (`https://github.com/visipedia/inat_comp/tree/master/2021`). We sliced the dataset to include only birds using the value *Aves* in category *class* key. This resulted in 414,847 samples in training and 14,680 samples in testing. The maximum and minimum samples per species were 300 and 152 respectively. Then, we applied a *minimal filter* to remove entries with missing geolocations or timestamps. This resulted in dropping only 888 out of 414,847 (0.2%) observations in training. In testing, we dropped 29 out of 14,860 (0.1%).

Dataset Details. The total number of samples in training and testing were 413,959 and 14,831 respectively. The maximum and minimum samples per species after filtering remained the same as before. The distribution of samples per species and per month is shown in Figure **??**1. As is seen, the distribution of samples across the species did not change significantly after filtering. Further, the even distribution of samples across the species comes out-of-the-box from the original dataset. Finally, we collected corresponding satellite imagery for each bird sample using the geolocation present in the dataset. This was done by issuing WMS requests to the Sentinel-2 Cloudless server (`https://s2maps.eu/`).

B. Training

Training details. We use the *timm* package for creating all our models and *pytorch_lightning* (pl) package for training and inference. All the training is done on 4 NVIDIA A100_SXM4_40GB GPU's using pl's DistributedDataParellel (ddp) recipe. The experiments are run in parallel across 2 nodes (Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz).

Metadata. The iNAT-2021 Birds Dataset includes meta-information in the form of dictionary with keys: *id*, *width*, *height*, *file_name*, *license*, *rights_holder*, *date*, *latitude*, *longitude*, *location_uncertainty*. We extract the *latitude*, *longitude*, and *date* values for each image. For *date*, we extract the *month* and discard all other fields. The three values are mapped as follows:

$$lon \rightarrow (sin(\pi * lon/180), cos(\pi * lon/180)) \quad (1)$$

Table 1: Pre-training hyperparameters and settings.

Config	Value
optimizer	AdamW
weight decay	0.01
base learning rate	1e-4
batch size	308
optimizer momentum	β_1=0.9, β_2=0.95
learning rate scheduler	cosine decay
input normalization	μ = [0.485, 0.456, 0.406]
	σ = [0.229, 0.224, 0.225]
masking ratio	0.75
meta dropout	0.25
augmentation_ground	RandomResizedCrop(384)
	TrivialAugment
augmentation_satellite	RandomResizedCrop(224)
	ColorJitter(0.5, 0.5, 0.5)
	RandomHorizontalFlip(p=0.5)

$$lat \rightarrow (sin(\pi * lat/90), cos(\pi * lat/90)) \quad (2)$$

$$month \rightarrow (sin(\pi * month/12), cos(\pi * month/12)) \quad (3)$$

All the values are concatenated and passed to a linear layer which embeds them to a dimension of 768. This embedding is added after extracting features from the encoders. More specifically, we add it to the [cls] token's embedding from the encoders. This final [cls] embedding is used in pre-training and various downstream tasks. If meta-dropout is turned on with probability p, we simply add zeros to the [cls] embedding with probability p during training.

Pre-Training. We use a masking ratio of 75% for the masked reconstruction objective. We *do not* use momentum contrast for the contrastive learning objective as [1] only reported a minor improvement in performance. We use color jittering, RandomResizedCrop and RandomHorizontalFlip for the overhead satellite images. For the ground-level images, we only use RandomResizedCrop and TrivialAugment [2]. The specific details are presented in Table 1.

Linear Probing. Following [3], we only use RandomResizedCrop during linear probing. All embeddings are

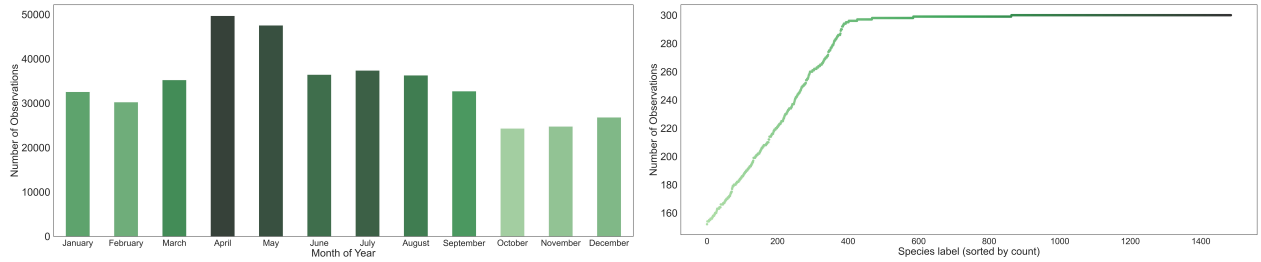

Figure 1: **Distribution of Training Samples**. We show the number of observations per month (left) and number of observations per species (right). The figures indicate that the training set is fairly balanced across the species.

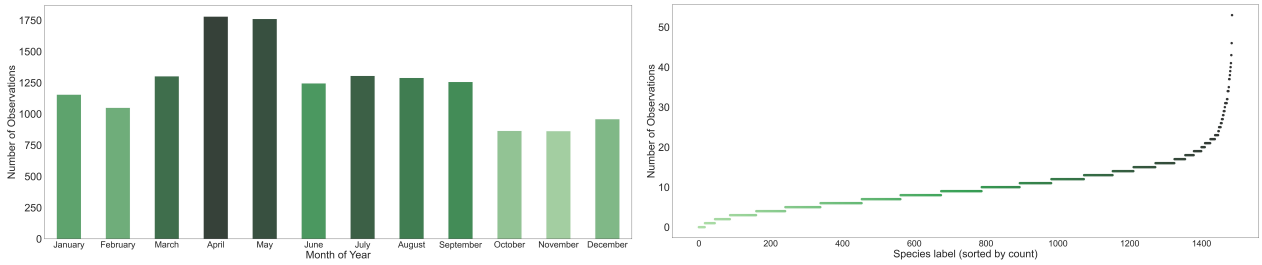

Figure 2: **Distribution of Testing Samples**. We show the number of observations per month (left) and number of observations per species (right). The figures indicate that the distribution of species across the months is fairly balanced while there is a slight imbalance in the number of observations per species.

Table 2: Linear probing hyperparameters and settings.

Config	Value
optimizer	AdamW
weight decay	1e-4
base learning rate	0.1
batch size	308
optimizer momentum	β_1=0.9, β_2=0.999
learning rate scheduler	cosine decay
input normalization	μ = [0.485, 0.456, 0.406]
	σ = [0.229, 0.224, 0.225]
meta dropout	0.25
augmentation_ground	RandomResizedCrop(384)
augmentation_satellite	RandomResizedCrop(224)

normalized before passing to linear layer for classification. Other details are illustrated in Table 2. As reported by [3], linear probing accuracy is uncorrleated from fine-tuning accuracy. This explains the fact that there is a large gap in the metrics on iNAT-2021 Birds Dataset. They also concluded that contrastive-based models were better than MAE at linear probing. The combination of contrastive and masked reconstruction objectives allows our model to learn robust features for a variety of downstream tasks and beat purely contrastively trained models.

Table 3: Downstream fine-grained classification hyperparameters and settings.

Config	Value
optimizer	AdamW
weight decay	0.1
base learning rate	5e-5
batch size	308
optimizer momentum	β_1=0.9, β_2=0.999
learning rate scheduler	cosine decay
input normalization	μ = [0.485, 0.456, 0.406]
	σ = [0.229, 0.224, 0.225]
meta dropout	0.25
augmentation_ground	RandomResizedCrop(384)
	RandAugment(10, 12)
	CutMix = 1.0
	mixup = 0.8
	LabelSmoothing = 0.1
augmentation_satellite	RandomResizedCrop(224)
	ColorJitter(0.5, 0.5, 0.5)
	RandomHorizontalFlip(p=0.5)

Fine-Tuning. For ViT's, fine-tuning (in general) requires severe data augmentations and higher weight decay. As a result, we use RandomResizedCrop, RandAugment,

Table 4: Comparison of F1 Score, Precision and Recall achieved by our proposed models and SotA approaches on the standard test set of iNAT-2021 Birds dataset. We report this for fine-tuned models.

Method	Location	Date	Pre-training	F1 Score	Precision	Recall
MoCo-V2-Geo	✓	✗	InfoNCE+Geo-Clf.	0.507	0.511	0.503
MAE	✗	✗	Recons. Loss	0.488	0.482	0.495
MetaFormer-2	✓	✓	ImageNet Clf.	0.510	0.534	0.488
CVE-MAE	✗	✗	InfoNCE+Recons. Loss	0.520	0.519	0.522
CVE-MAE-Meta	✓	✓	InfoNCE+Recons. Loss	0.527	0.523	0.531
CVM-MAE	✗	✗	Matching+Recons. Loss	0.545	0.552	0.539
CVM-MAE-Meta	✓	✓	Matching+Recons. Loss	**0.553**	**0.561**	**0.546**

CutMix, mixup and LabelSmoothing. Other details are preseneted in Table 3. We also report additional metrics of our fine-tuned models in Table 4.

MoCo-V2-Geo. To make fair comparisons, we implemented a cross-view training routine for the MoCo-V2-Geo. Instead of utilizing temporal positives or data augmentation techniques to create positive and negative pairs, we use the corresponding satellite images. We cluster the geographic coordinates present in the meta-information into 20 classes using the KMeans clustering algorithm (Figure **??**). These labels are then used for computing the geo-classification loss using the [cls] embeddings obtained from the ground-level image encoder. We use a queue of size 10000 and the same data augmentations for the ground-level and satellite images as reported in Table 1.

C. Ablation Study

We conduct ablation on the meta-dropout rate which is a key component of our architecture. It tunes the dependence of models on metadata. Rest of the components are pre-trained SotA architectures which have been extensively studied in previous literatures. As is seen in Table 5, models severely overfit on metadata, when meta-dropout is turned off. Also, the performance of the models' decrease as meta-dropout rate increases more than 25%.

Table 5: Impact of meta-dropout rate on the classification performance of models on Cross-View iNAT-2021 dataset.

Model	Meta-Dropout Rate				
	0.00	0.25	0.50	0.75	1.00
CVE-MAE-Meta	82.23	**86.23**	85.02	84.79	83.78
CVM-MAE-Meta	83.55	**87.46**	86.22	85.97	85.89

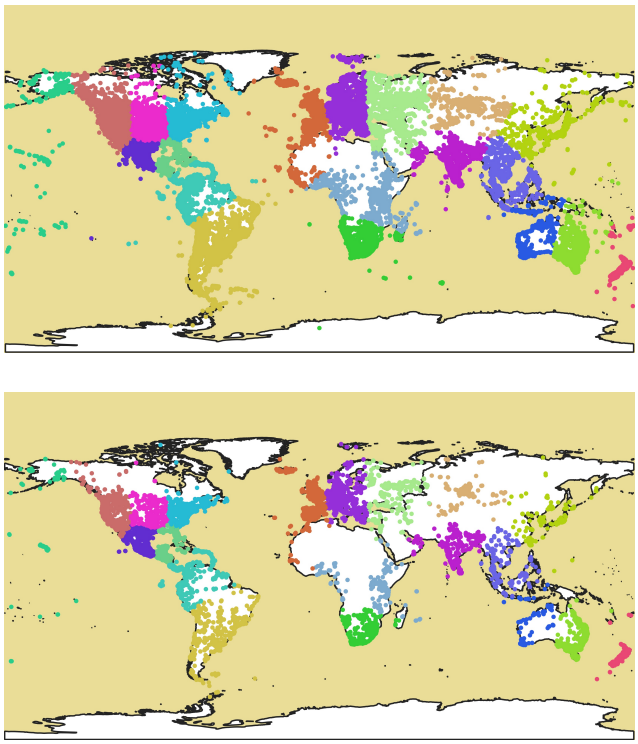

Figure 3: **Geo-Clusters**. Using KMeans Clustering, we cluster the geographic coordinates into 20 classes based on latitude and longitude values. These classes are then used for training on the geo-classification objective for the MoCo-V2-Geo method. Here, we show the training (top) and testing (bottom) geo-locations of images along with color representing their cluster label.

D. Species Distribution Mapping

Species distribution maps are constructed by first collecting satellite images over a dense rectangular grid draped on

the area of interest. Then, similarity scores are computed between a query bird image and the satellite images. More specifically, we precompute the embeddings for all the image pairs on GPU and then compute their similarity on CPU. The grid of scores is then interpolated at a desired spatial resolution. We use the IDW interpolation and a spatial resolution of 0.01^o for The Netherlands. Further, we clamp negative similarity scores to zero before visualizing. A similar procedure can be followed if one wants to use land cover maps, digital elevation models (DEM), and so on for creating species distribution maps. In the future, one could also incorporate text descriptions as query for generating these maps.

E. Reconstruction Results

The results of our reconstructions are not fully imperative for species classification and mapping, since our models are trained with a contrastive objective. In the wild, animals come in different poses and sizes, making it challenging to reconstruct them perfectly. However, our models have effectively learned the structures of different bird species. The large scale pre-training along-with satellite imagery and metadata such as month of year and location helps our model learn robust fine-grained features. The zero-shot reconstruction results on CUB-200-2011 and NABirds (Figure 7 and Figure 8) confirm that the features learned by our models are highly transferable.

References

[1] Z. Huang, X. Jin, C. Lu, Q. Hou, M.-M. Cheng, D. Fu, X. Shen, and J. Feng, "Contrastive masked autoencoders are stronger vision learners," *arXiv preprint arXiv:2207.13532*, 2022.

[2] S. G. Müller and F. Hutter, "Trivialaugment: Tuning-free yet state-of-the-art data augmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 774–782, 2021.

[3] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

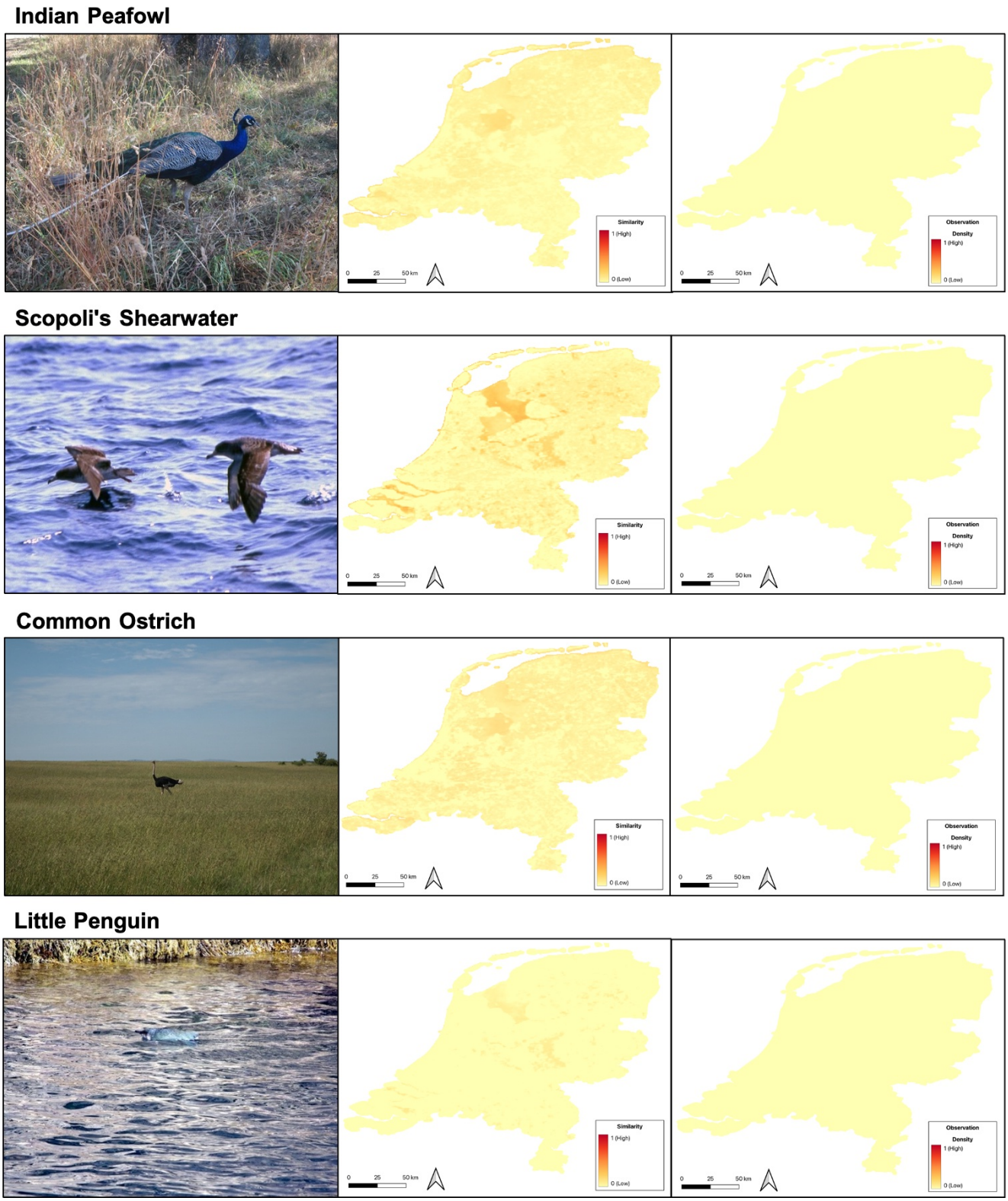

Figure 4: **Bird Maps for Negative Queries**. We select four bird species which are not typically found in The Netherlands. We show ground-level to satellite image similarity scores for those bird species over The Netherlands. Clearly, the maps show little to no activations.

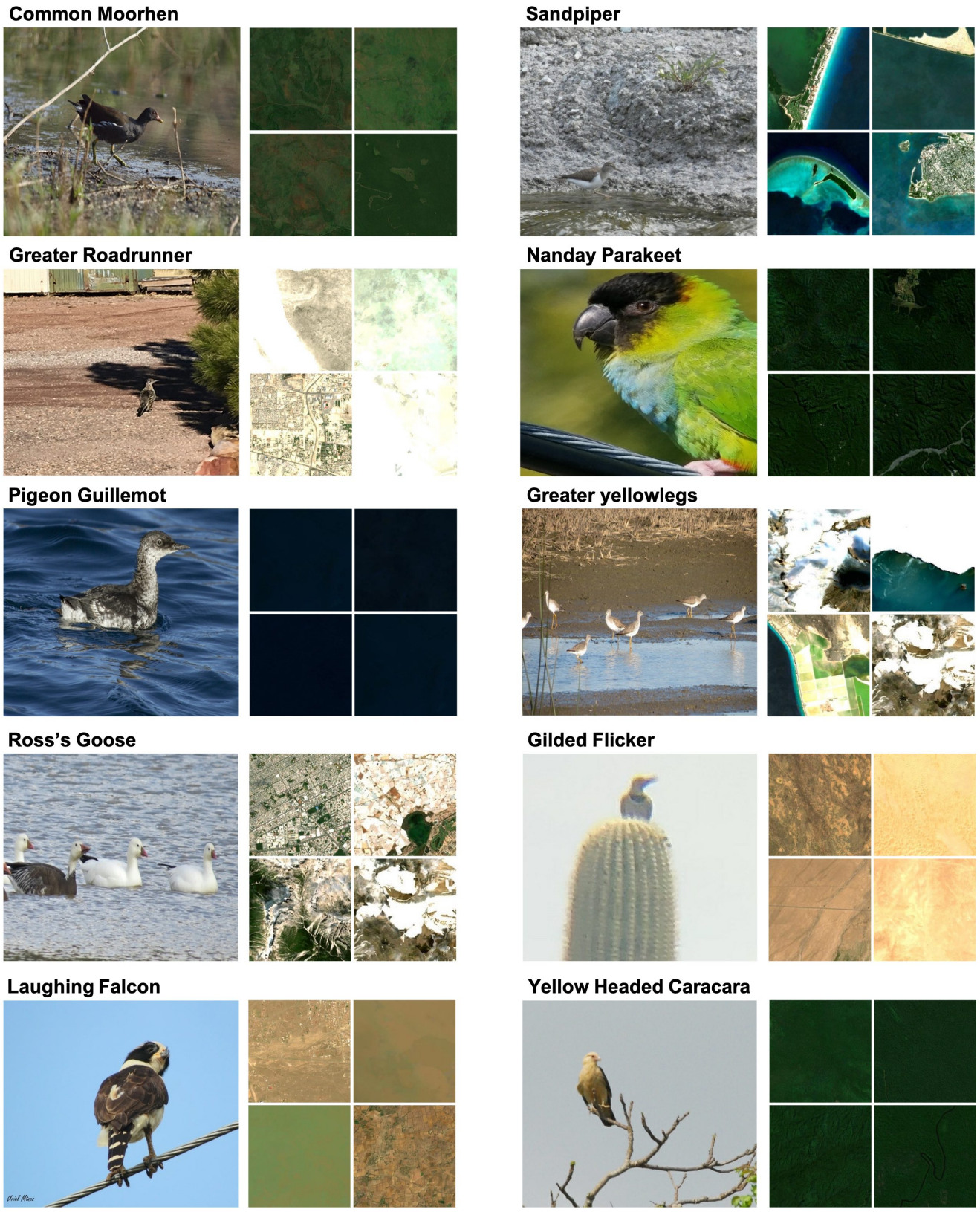

Figure 5: **Bird to Satellite Image Retrieval**. We show additional examples of uni-modal bird to satellite image retrieval. Clearly, our modal is able to associate bird species with their expected habitat and location.

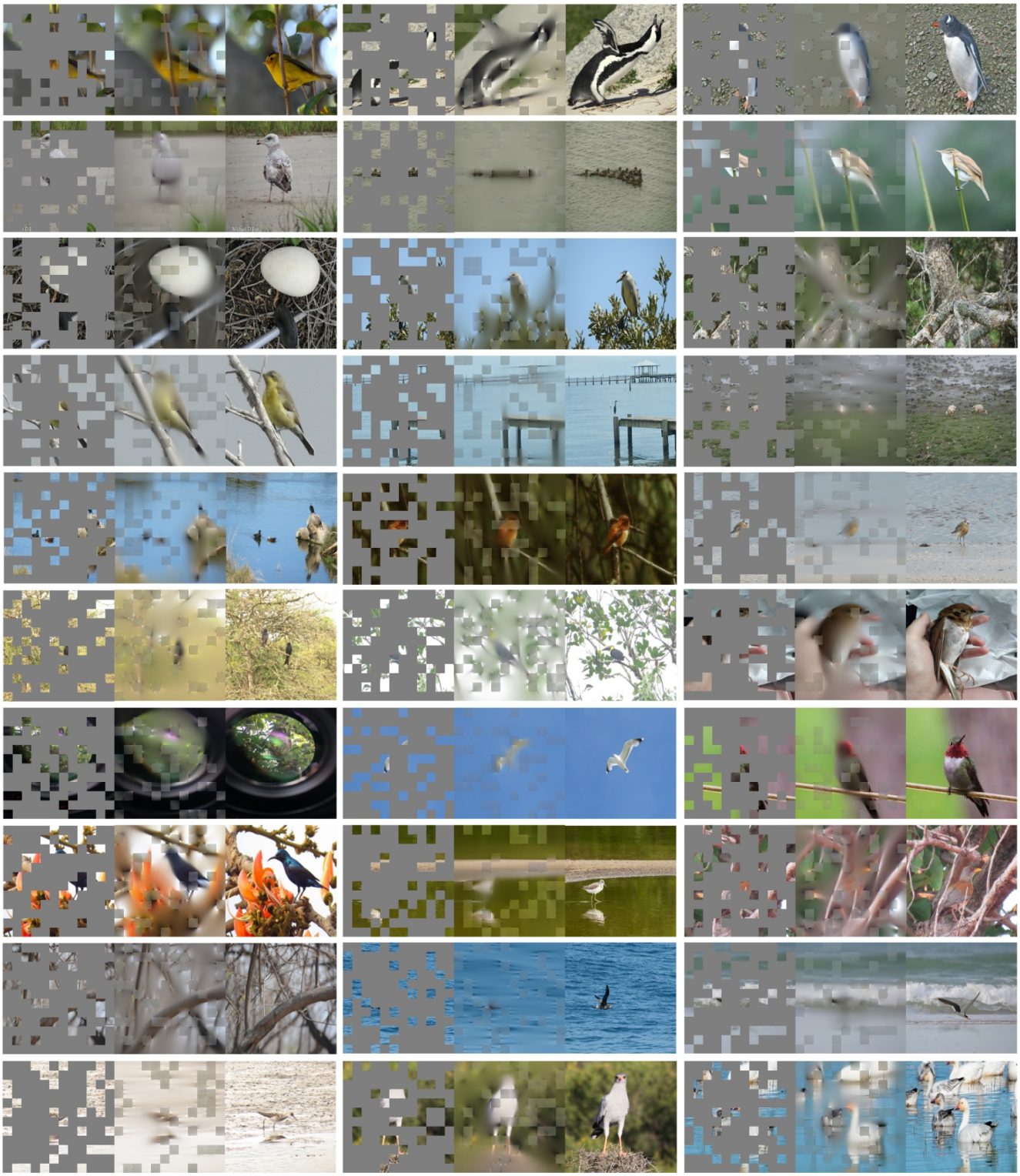

Figure 6: **Reconstruction Results**. Using pre-trained cross-view metric MAE model, we show reconstruction results on randomly selected images from the standard test set of Cross-View iNAT-2021 Birds Daataset. We illustrate masked (left), predicted (middle) and ground truth (right) images. The masking ratio is fixed at 75% during the inference.

Figure 7: **Zero-shot reconstruction on CUB-200-2011**. Using pretrained CVE-MAE-Meta, we show zero-shot reconsturction results on randomly selected images from testing set of CUB-200-2011. We show masked (left), predicted (middle) and ground truth (right) images.

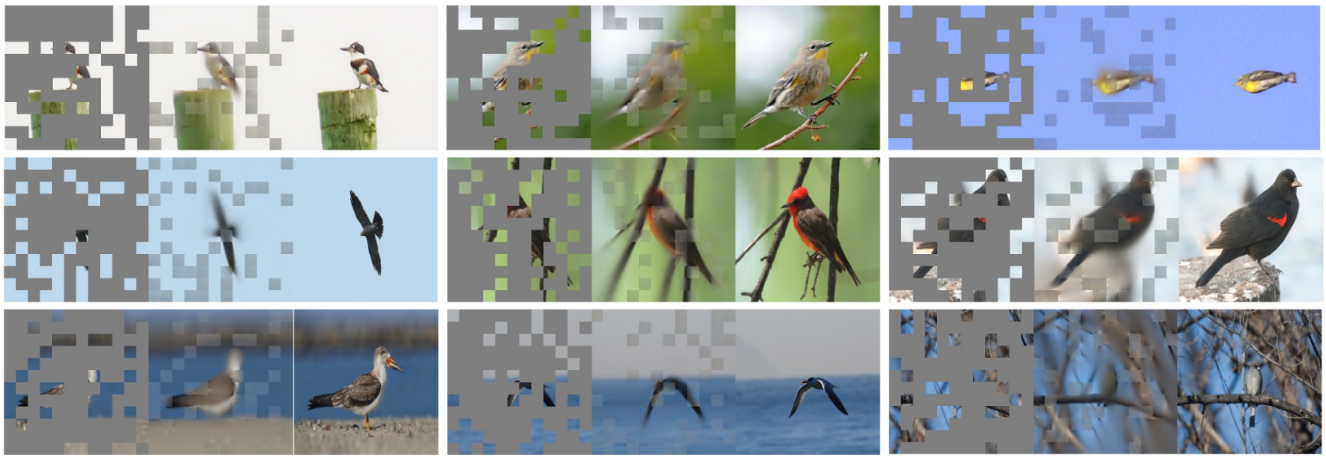

Figure 8: **Zero-shot reconstruction on NABirds**. Results on randomly selected images from testing set of NABirds. We illustrate masked (left), predicted (middle) and ground truth (right) images.