- Supplementary Materials -
Domain Generalization using Large Pretrained Models with Mixture-of-Adapters

Hyperparameter	LoRA	KAdaptation	LoRA-MoA	KMoA
# of Experts	N/A		4	4
Scale of $\mathcal{L}_{\text{aux}}$	N/A		0.01	0.01
Router	N/A		Cosine	
Router Top-k	N/A		Top-1	
Rank of adapter (r_i)	2	1	$[1, 2, 4, 8]$	
# of Kroneker products (t)	N/A	64	N/A	64
Batch size	160 (DomainNet), 96 (Otherwise)			
Learning rate	$5e-5$			
Optimizer	Adam			

Table 1. List of hyperparameters used in experiments on domain generalization benchmarks.

In this supplemental material, we provide additional analysis results and visualizations. We also include the code needed to reproduce our experimental results.

1. Additional implementation details

All adapters, except LoRA, are implemented from their official repositories; LoRA is implemented using an unofficial version [13]. By following the previous experimental settings, Adam [9] optimizer is used for model optimization along with a learning rate of $5e-5$. A batch size of 32 per domain is used for the ViT-Base model. We run 15,000 iterations on DomainNet and 5,000 for others, and evaluate at every 500 iteration steps for DomainNet, 200 steps for others. We perform all experiment on one machine with 8 NVIDIA RTX3090 GPUs.

Evaluation protocols and datasets. For a fair comparison, we employ DomainBed evaluation protocols [2,5]. The following five benchmark datasets: PACS [10], VLCS [4], OfficeHome [14], TerraIncognita [1], and DomainNet [12]. Using a *leave-one-out cross-validation*, all performance scores are evaluated by averaging all the cases that use a single domain as the target domain and the others as the source domains. Experiment is repeated three times and 20% percent of source domain data is left out for validation purposes. Lastly model selection (training-domain validation) and hyperparameter search follow DomainBed procedures. We perform three runs with different random seeds for each setting and report their mean and standard deviation to show the training randomness. In ablation studies, we keep all the random seeds fixed and conduct the experiment.

Test Env.	$\mathcal{L}_{\text{aux}}$	Expert 0	Expert 1	Expert 2	Expert 3	Std
Art	✗	0.03	0.14	0.23	0.60	0.213
	✓	0.22	0.26	0.32	0.21	**0.044**
Cartoon	✗	0.02	0.19	0.07	0.72	0.275
	✓	0.23	0.18	0.25	0.35	**0.062**
Photo	✗	0.07	0.17	0.56	0.21	0.184
	✓	0.23	0.21	0.29	0.26	**0.030**
Sketch	✗	0.10	0.17	0.16	0.57	0.185
	✓	0.32	0.31	0.17	0.20	**0.065**

Table 2. Analysis about the effectiveness of auxiliary loss on PACS dataset. Each number represents the relative allocation ratio, calculated by counting the number of tokens routed to each expert and dividing by the total number of tokens.

2. Additional analysis

In this section, we present an additional analysis of routed tokens, loss landscapes, and maximum Hessian eigenvalue spectra.

2.1. Comparisons of loss landscape visualizations

We show loss landscapes for all test environments in PACS dataset [10] in Fig. 1. Similar with the visualizations in main paper, the other test environments have a tendency that fully fine-tuned models show most sharp loss landscape. But trained models with LoRA and KAdaptation shows much more flatter loss landscapes, especially KAdaptation have most flat loss landscape.

2.2. Analysis about the effectiveness of auxiliary loss

In this section, we analyze how our model's router allocates each token according to $\mathcal{L}_{\text{aux}}$. As shown in Fig. 2, without the auxiliary loss, the router's token allocation to the experts is highly imbalanced. However, when the auxiliary loss is applied, the allocation becomes significantly more balanced. We show the standard deviation of the tokens in Table 2. The results indicate that training with $\mathcal{L}_{\text{aux}}$

leads to a more balanced distribution of tokens across the experts. This balance could play a crucial role when scaling up the model or applying it to downstream tasks.

2.3. Visualizations of routed patches in PACS and TerraIncognita dataset.

We additionally show the visualizations of routed patch indices in Fig. 3, 4 on PACS dataset [10], and Fig. 5, 6 on TerraIncognita dataset [1]. All images are visualizations from the last adapter-attached transformer layer, layer 10. Similar with the findings from main paper, we can observe that same indices are clustered at the regions where having semantic meanings.

2.4. Limitations

Our method heavily relies on the performance of large pretrained models, hence using a better pretrained model can lead to improved performance. But, such models are limited and require a substantial amount of time and cost for training. These weakness also exist in methods like MIRO [3] or SIMPLE [11], and the availability of high-performance open-source models like OpenCLIP [8] can alleviate these drawbacks. Our approach may not significantly outperform on datasets more challenging than TerraIncognita due to fewer trainable parameters compared to fully fine-tuned DG algorithms. However, it offers flexibility by adjusting trainable parameters via the inner rank r_i, and optimal rank can be obtained through hyperparameter search, effectively addressing this limitation.

References

[1] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1, 2, 7, 8

[2] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021. 1

[3] Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 440–457. Springer, 2022. 2

[4] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013. 1

[5] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020. 1

[6] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient fine-tuning for vision transformers. *arXiv preprint arXiv:2203.16329*, 2022. 3

[7] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 3

[8] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

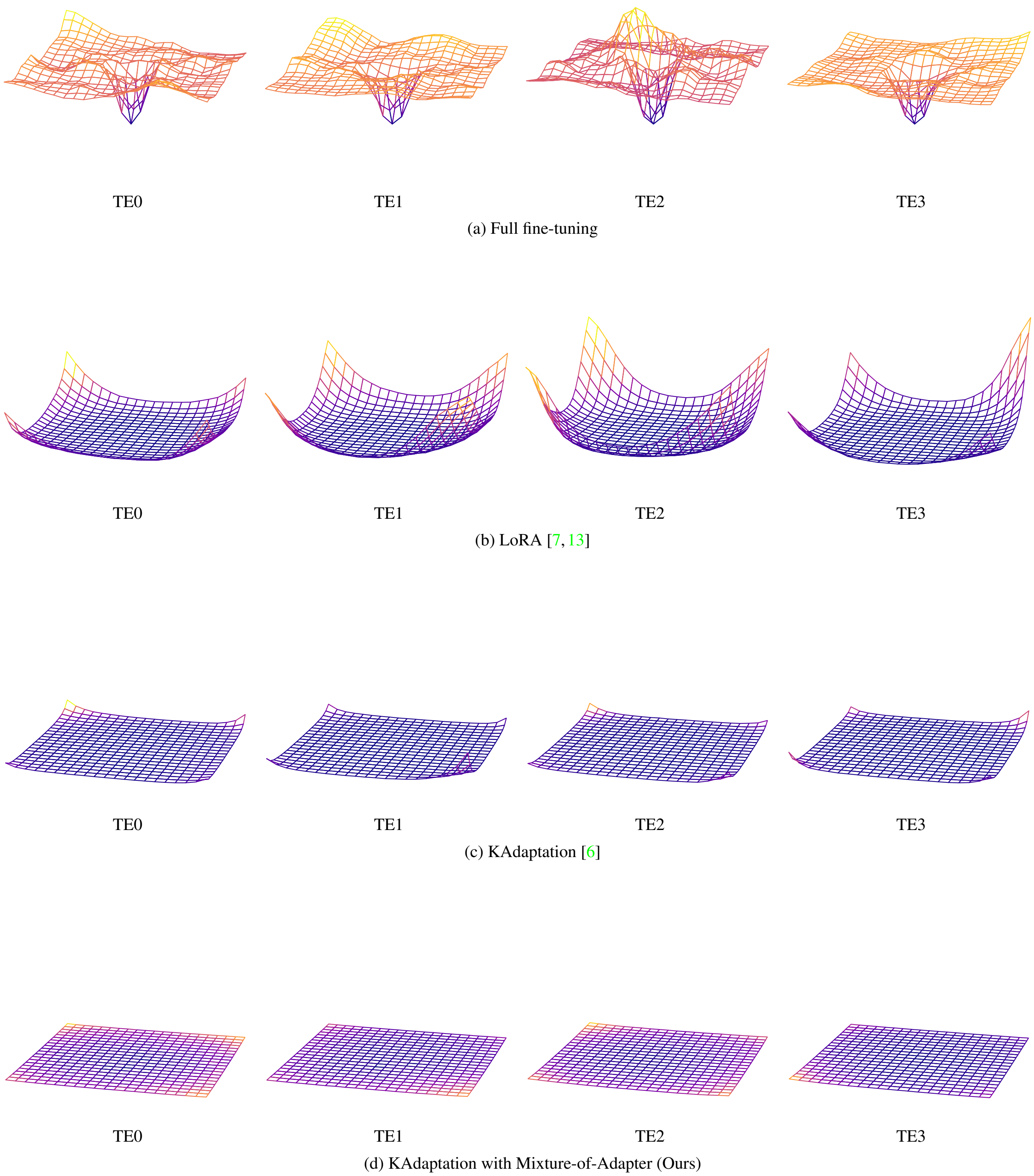
[10] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1, 2, 3, 5, 6

[11] Ziyue Li, Kan Ren, Xinyang Jiang, Yifei Shen, Haipeng Zhang, and Dongsheng Li. Simple: Specialized model-sample matching for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 1

[13] Simo Ryu. lora. https://github.com/cloneofsimo/lora, 2023. 1, 3

[14] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1



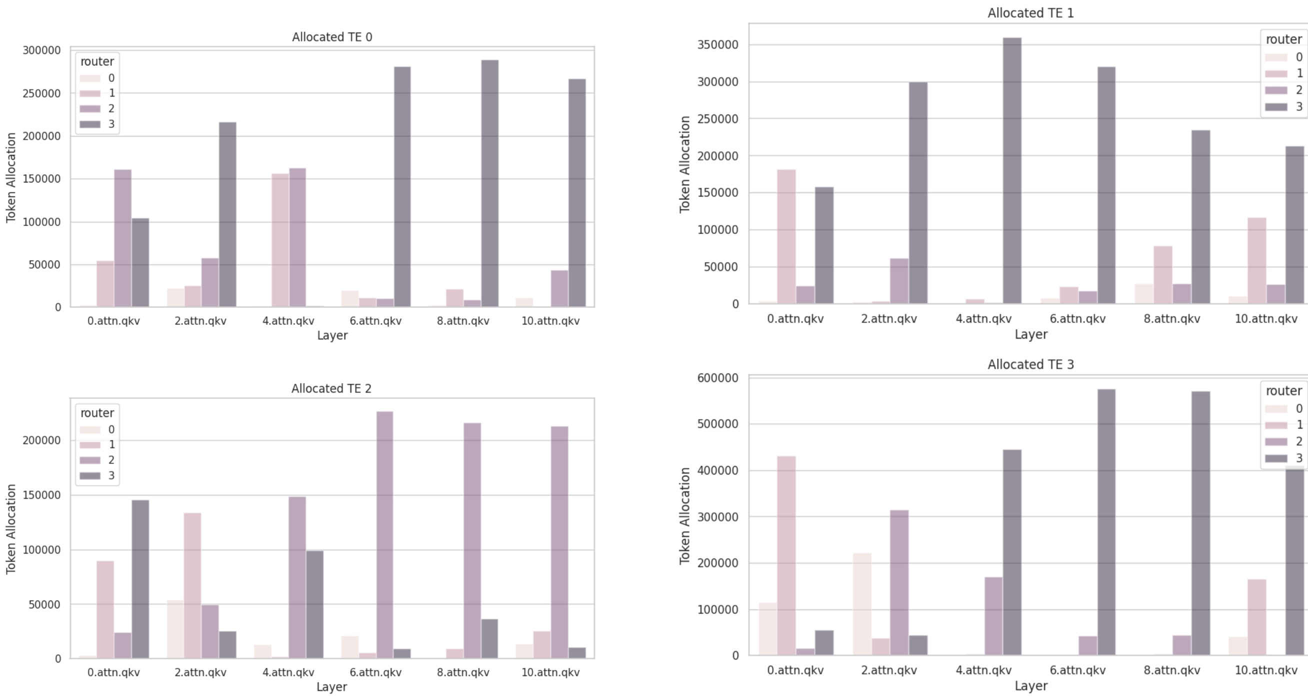
(a) Full fine-tuning

(b) LoRA [7, 13]

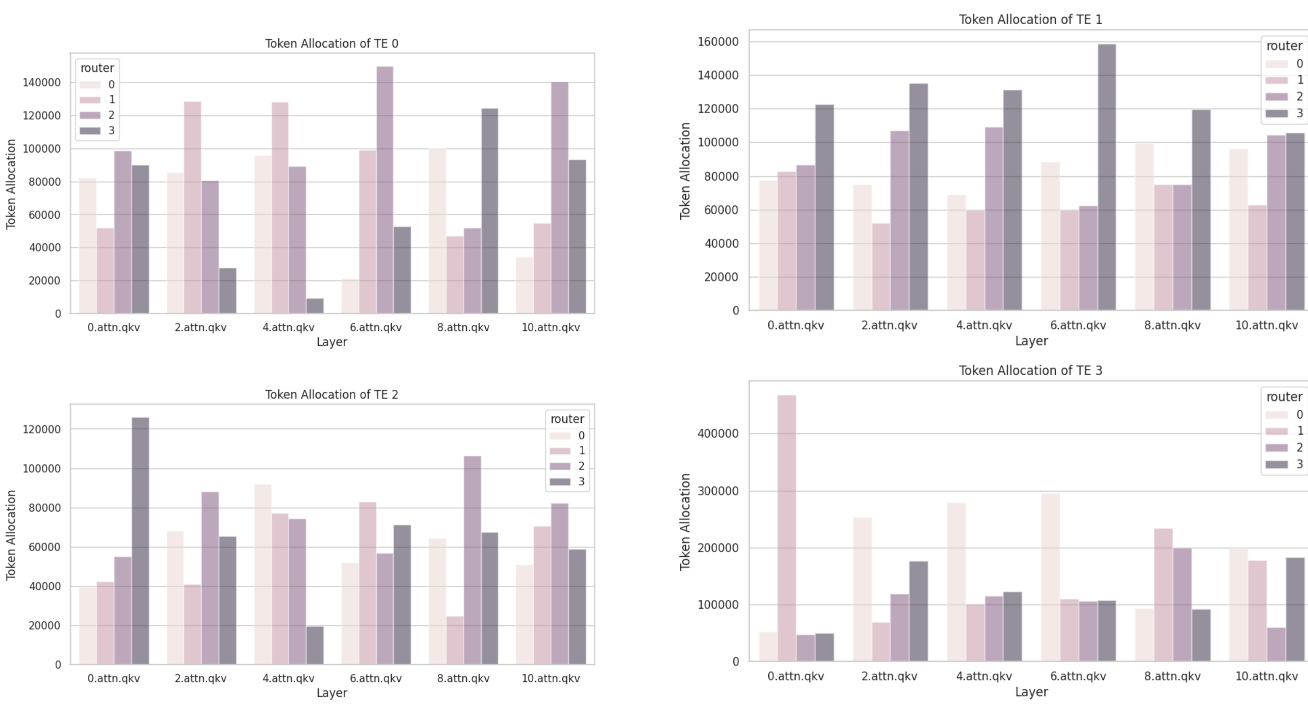
(c) KAdaptation [6]

(d) KAdaptation with Mixture-of-Adapter (Ours)

Figure 1. Flatness comparison of loss surfaces trained with full fine-tuning, LoRA, KAdaptation, and KAdaptation with mixture-of-expert on the PACS dataset [10].



Without $\mathcal{L}_{\text{aux}}$



With $\mathcal{L}_{\text{aux}}$

Figure 2. Visualizations of token routing tendencies with and without the auxiliary loss on PACS dataset. TE0 to TE3 correspond to the domains in the PACS dataset: Art_painting, Cartoon, Photo, and Sketch. The x-axis represents the layer names containing the router and experts, while the y-axis shows the number of tokens allocated to each expert.

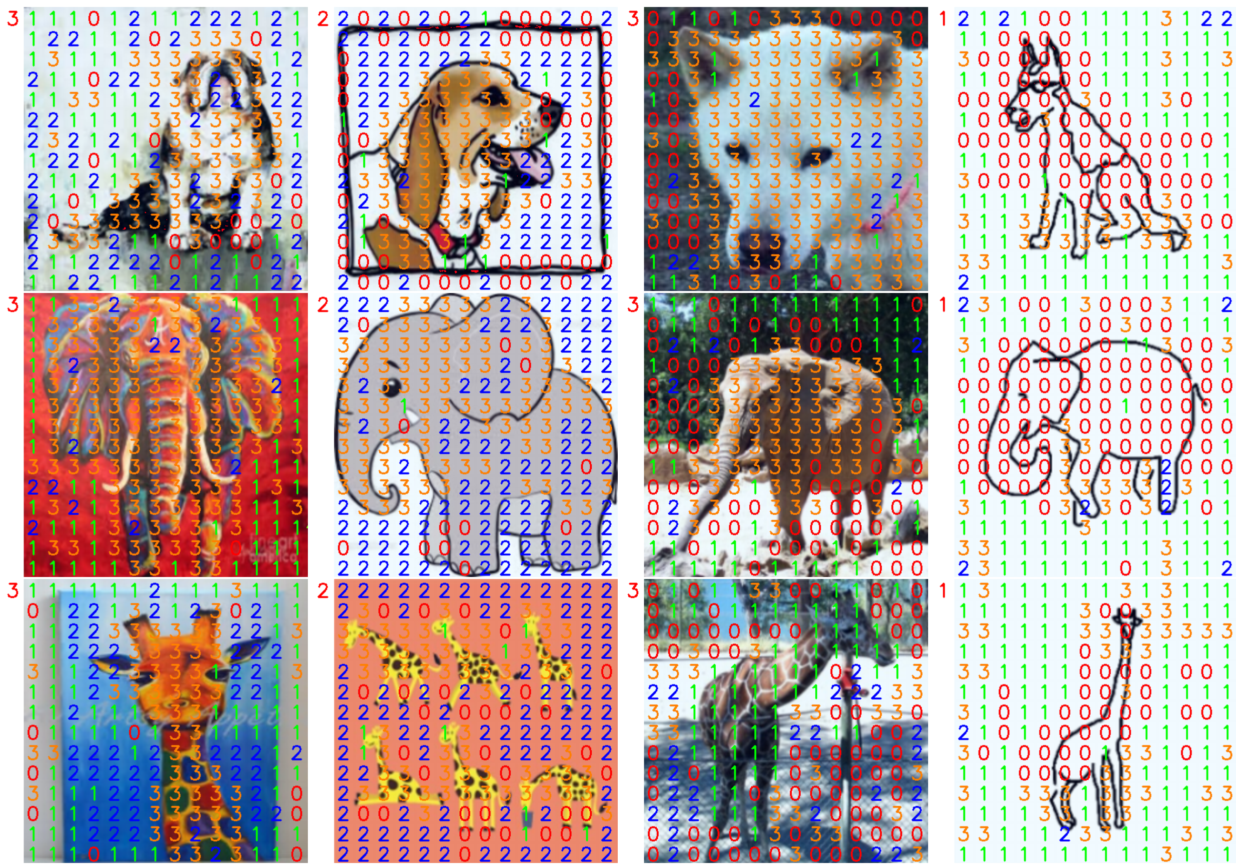


Figure 3. Visualizations of routed indices of each patch. We show a total of seven classes in PACS dataset [10], with one class per row in the order of 'Dog', 'Elephant', 'Giraffe'. Also, in each column, the same domains are located in the order of 'Art Painting', 'Cartoon', 'Photo', and 'Sketch'.

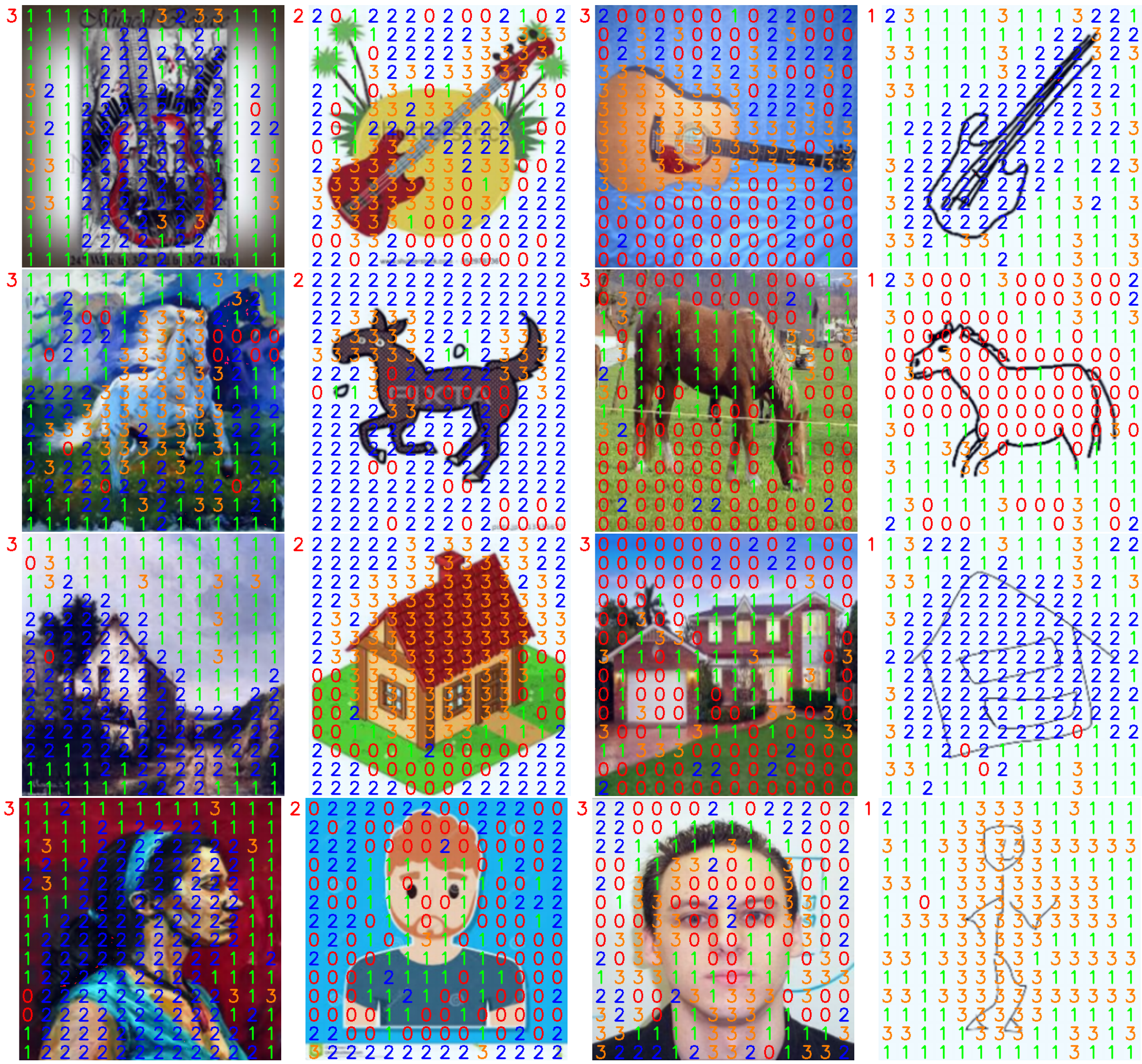
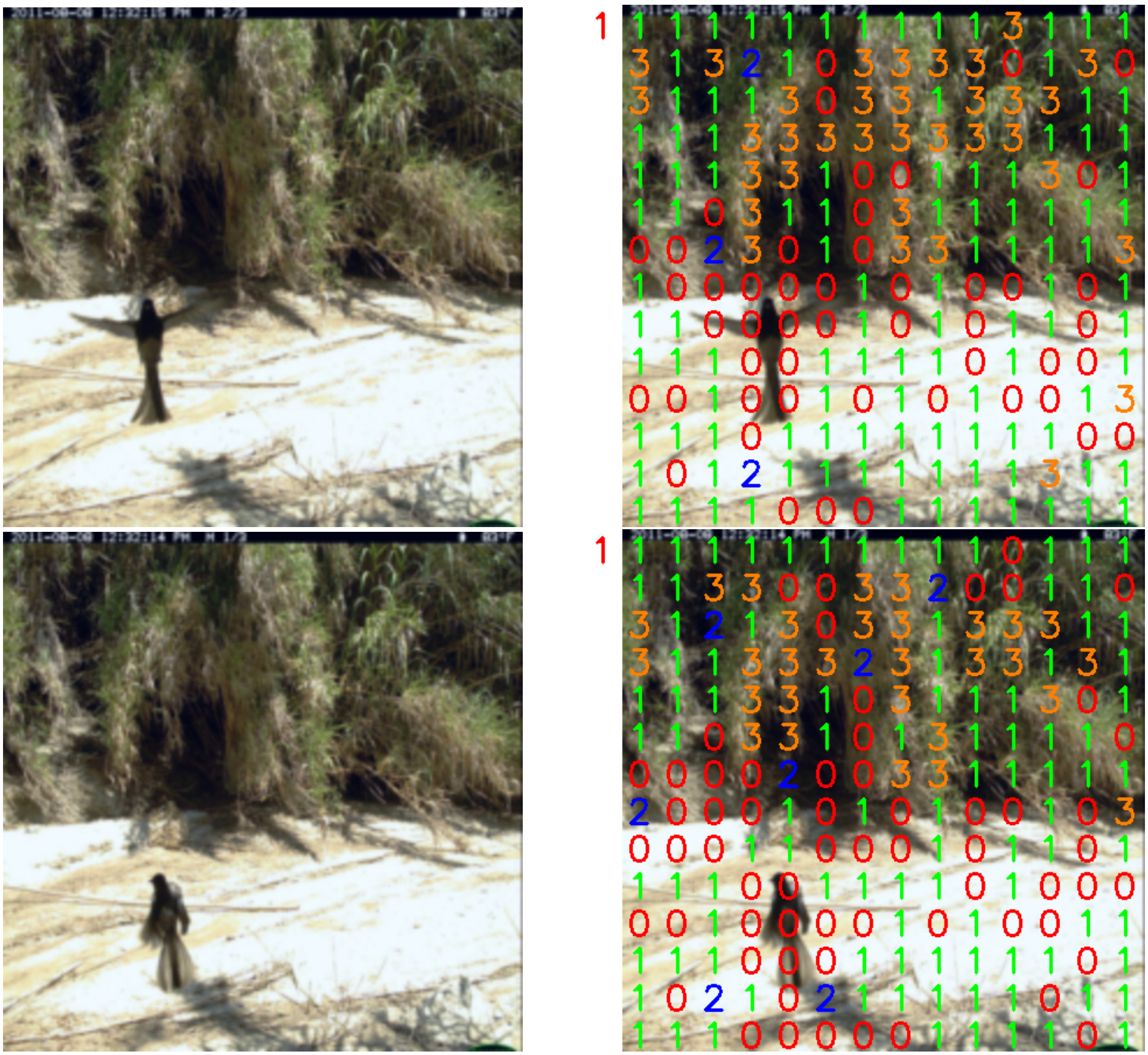
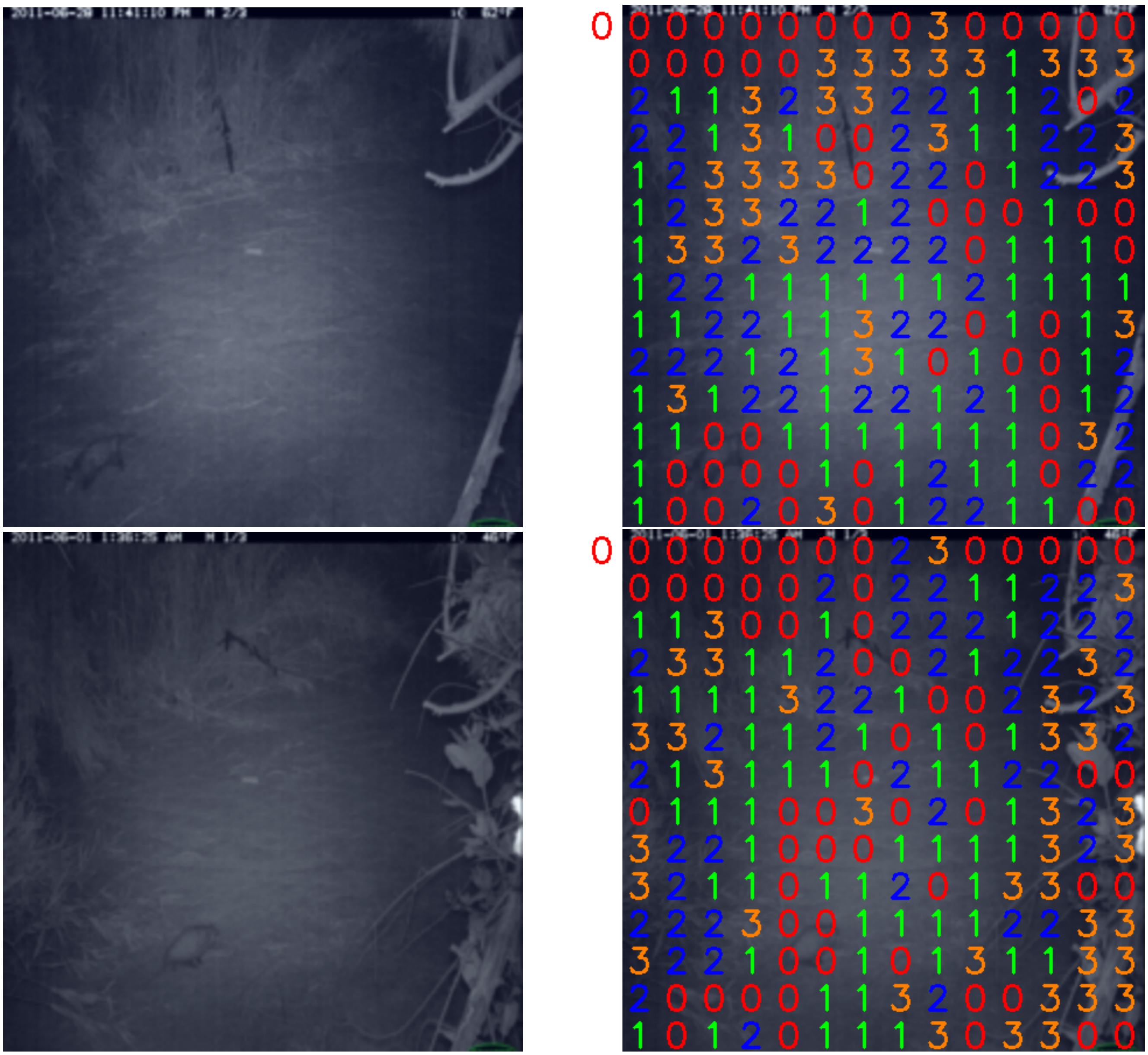


Figure 4. Visualizations of routed indices of each patch. We show a total of seven classes in PACS dataset [10], with one class per row in the order of 'Guitar', 'Horse', 'House', 'Person'. Also, in each column, the same domains are located in the order of 'Art Painting', 'Cartoon', 'Photo', and 'Sketch'.



Location 43

Figure 5. Visualizations of routed indices for each patch in the TerraIncognita [1] dataset. The left column displays the original image, while in the right column, we indicate where each patch is routed. The upper and lower images were taken at the same location but different times, therefore they share the same background but feature different object (bird) in terms of shape and location.



Location 46

Figure 6. Visualizations of routed indices for each patch in the TerraIncognita [1] dataset. The left column displays the original image, while in the right column, we indicate where each patch is routed. The upper and lower images were taken at the same location but different times, therefore they share the same background but feature different object (opossum) in terms of shape and location.