A Broad Study of Pre-training for Domain Generalization and Adaptation Supplementary Material

Donghyun Kim[2], Kaihong Wang[1], Stan Sclaroff[1], and Kate Saenko[1,2]

[1]Dept. CS, Boston University, [2]MIT-IBM Watson AI Lab
{donhk, kaiwkh, sclaroff, saenko}@bu.edu

Summary. We provide additional details and experimental results in this supplementary material. Future domain adaptation work can use our code[1] for a new baseline for domain transfer tasks.

I Appendix

I.1 Additional Training Details

Our implementation is based on the timm[2] library and the transfer learning library in [2]. We directly use the implementation of the the transfer learning library in [2], which supports domain adaptation baselines (DANN, CDAN, MCC, AFN, and MDD). For some pre-trained weights not available in the timm library, we directly use the publicly released pre-trained weights from the authors. In Fig. I, we show example images from different domains in each dataset, which show the type of domain shift in each benchmark.

I.2 Source Accuracy on Single Source Generalization

We provide a comparison of source accuracy on the source validation set on ConvNext and Swin Transformers in Fig. II. We compare the source accuracy between shallow models, ConvNeXt-S and Swin-S pre-trained on ImageNet-1K, and deep models, ConvNeXt-XL and Swin-L pre-trained on ImageNet-22K. We observe that deep models, ConvNeXt-XL and Swin-L, obtain higher source accuracy on all the benchmarks compared to the shallow models, ConvNeXt-S, Swin-S. According to the theory in in [1], the expected target error $\epsilon_T(h)$ can be bounded by the expected source error, the discrepancy between the source and target domains, and the shared error of the ideal joint hypothesis λ.

$$\epsilon_T(h) \leq \epsilon_S(h) + d_1\left(\mathcal{D}_S, \mathcal{D}_T\right) + \lambda \tag{1}$$

Prior domain adaptation methods assume that expected source error $\epsilon_S(h)$ is low since we assume many source labels. Therefore, prior domain adaptation methods focus on minimizing the discrepancy between the source and target

[1] https://github.com/VisionLearningGroup/Benchmark_Domain_Transfer
[2] https://github.com/rwightman/pytorch-image-models

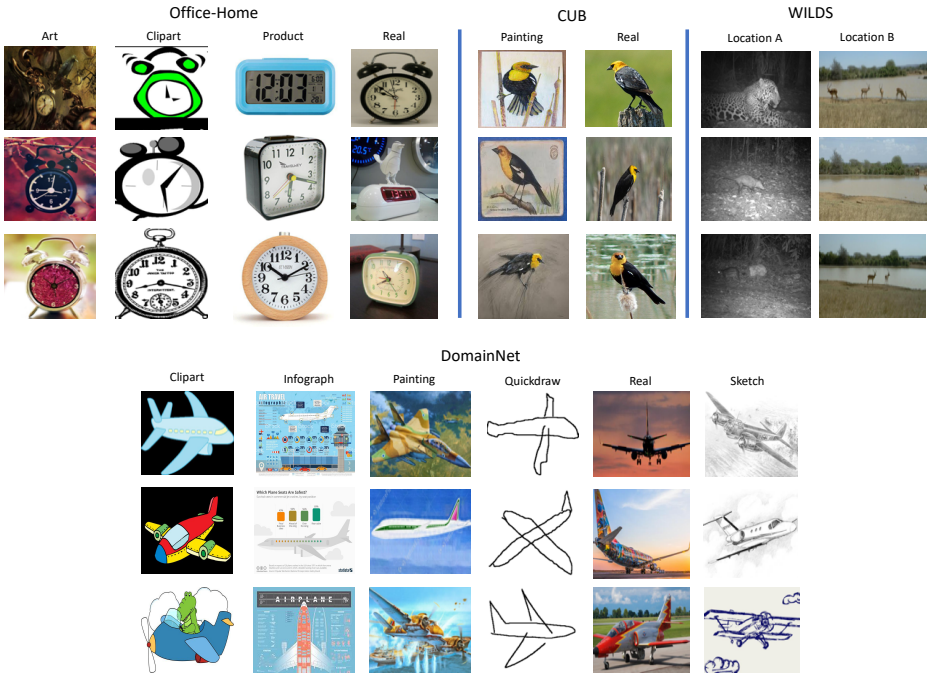


Fig. I: Examples from different domains in the benchmarks

domains. However, as shown in Fig. II, there is a gap in $\epsilon_S(h)$ between pre-trained models. Using modern pre-training can further reduce the upper bound of expected target error with a lower value of $\epsilon_S(h)$.

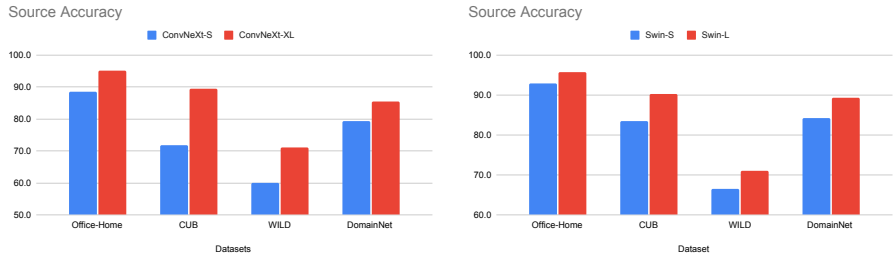
Additionally, we observe that there are noticeable inconsistencies between the source validation accuracy and target accuracy across different optimizers and learning rates in shallow models trained on ImageNet-1K (*e.g.*, ResNet-50, Tiny, Small models of ConvNeXtand Swin) when fine-tuned on CUB and WILD. That means using the higher source validation accuracy for model selection can obtain a very lower target accuracy. We choose the best optimizer and learning rate on target accuracy but use early stopping with source validation accuracy only on these cases for proper comparisons. However, these gaps become smaller in deeper models trained ImageNet-22K and we observe similar trends between source validation and target accuracy. This also suggests that deep models trained on a larger dataset can obtain better domain invariant features than shallow models trained on a smaller dataset.

I.3 Additional Results on Single Source Generalization

In our main paper, we only use the image encoder in ALBEF. We also try to fine-tune the text encoder in ALBEF. Table I shows that the results of fine-tuning text and image encoder in ALBEF. We observe that fine-tuning the text encoder

Table I: The effect of fine-tuning the text encoder in ALBEF on Office-Home

Backbone	Fine-tune text encoder	Al	Cl	Pr	AVG
ALBEF	✗	81.7	72.5	87.2	80.4
ALBEF	✓	79.8	71.4	86.8	79.3



(a) ConvNeXt (b) Swin Transformers

Fig. II: (a) Source accuracy comparison on the variants of ConvNeXt (b) Source accuracy comparison on the variants of Swin Transformers

is not helpful on Office-Home. In Table II, we provide an accuracy comparison on different pre-training datasets and network architectures. In Tables III and IV, we provide additional results trained on other source domains on Office-Home and DomainNet.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains, vol. 79. Springer (2010)
2. Jiang, J., Chen, B., Fu, B., Long, M.: Transfer-learning-library. `https://github.com/thuml/Transfer-Learning-Library` (2020)

Table II: Additional accuracy comparison on different network architectures and datasets

Backbone	Pre-train. Data	Params	Office-Home			CUB	WILD	DomainNet					AVG
			Ar	Cl	Pr	Pa	-	Cl	In	Pa	Qu	Sk	
ResNet-50	ImageNet-1K	23	66.1	49.0	77.2	42.3	70.7	46.6	17.3	45.2	6.5	35.3	50.0
ResNet-101	ImageNet-1K	42	68.5	52.4	79.9	46.1	74.0	49.3	19.2	48.6	8.7	38.5	52.9
DeiT-B	ImageNet-1K	85	73.4	54.7	83.2	60.6	74.8	54.3	20.7	51.0	7.5	39.5	56.9
Swin-B	ImageNet-1K	86	75.7	54.7	84.8	57.1	76.6	56.7	22.8	52.6	8.8	41.9	58.1
ConvNeXt-B	ImageNet-1K	86	73.7	54.7	82.4	45.4	78.1	58.2	24.8	55.5	8.5	45.9	57.6
ViT-S	ImageNet-22K	21	74.1	52.7	83.9	64.0	75.7	53.3	20.3	51.2	7.4	37.3	56.9
ViT-B	ImageNet-22K	85	78.4	57.4	86.5	69.9	76.5	58.6	25.0	57.8	8.1	45.3	61.7
ViT-L	ImageNet-22K	303	84.0	73.0	89.9	76.5	77.7	65.5	27.3	61.3	10.2	52.1	67.5
Swin-B	ImageNet-22K	86	82.6	69.1	90.4	70.3	79.7	63.2	28.2	60.0	10.2	50.1	65.9
Swin-L	ImageNet-22K	194	83.4	74.3	90.9	73.0	81.4	67.2	30.6	62.5	11.2	54.1	68.6
ConvNeXt-B	ImageNet-22K	87	81.5	68.0	89.9	65.2	81.1	62.7	26.9	59.9	9.4	52.1	65.2
ConvNeXt-L	ImageNet-22K	196	84.6	73.2	90.5	70.7	81.1	66.6	28.8	61.6	10.0	54.3	67.9
ConvNeXt-XL	ImageNet-22K	348	85.1	74.0	91.4	71.9	81.5	67.7	29.7	62.2	11.4	55.5	68.8

Table III: Additional results trained on a different source domain on Office-Home across architectures

Backbone	Pre-train. Data	Params	Source: Ar			Source: Cl			Source: Pr			Source: Re			
			Cl	Pr	Re	Ar	Pr	Re	Ar	Cl	Re	Ar	Cl	Pr	Avg
ResNet-50	ImageNet-1K	23	46.8	64.4	71.2	52.5	62.5	63.6	49.5	42.5	72.3	66.1	49.0	77.2	58.4
ConvNeXt-T	ImageNet-1K	27	46.2	65.0	74.0	59.3	68.3	71.8	52.1	43.3	74.8	67.4	48.7	77.9	61.6
Swin-T	ImageNet-1K	27	52.3	68.8	75.5	57.9	65.9	69.3	62.1	47.6	78.8	71.3	49.4	81.1	64.2
ResNet-101	ImageNet-1K	42	53.3	69.5	76.1	56.5	66.2	67.0	55.2	47.1	75.1	68.5	52.4	79.9	62.9
Swin-S	ImageNet-1K	48	55.0	76.8	81.9	63.3	71.6	73.7	56.5	47.6	77.9	73.8	54.5	84.2	67.1
ConvNeXt-S	ImageNet-1K	49	53.4	72.7	78.6	67.5	72.9	75.4	61.8	49.0	80.0	72.2	52.7	80.9	67.9
Swin-B	ImageNet-22K	86	70.7	86.1	88.5	80.6	84.3	86.7	77.9	66.1	88.3	82.6	69.1	90.4	81.0
ConvNeXt-B	ImageNet-22K	87	69.7	83.0	86.9	80.5	85.9	86.5	73.8	63.2	87.6	81.5	68.0	89.9	79.7
Swin-L	ImageNet-22K	194	72.6	85.2	89.5	84.0	86.9	89.4	80.6	72.9	91.0	83.4	74.3	90.9	83.6
ConvNeXt-L	ImageNet-22K	196	71.9	86.5	89.6	83.7	85.9	88.0	80.8	67.7	89.3	84.6	73.2	90.5	82.6
ConvNeXt-XL	ImageNet-22K	348	74.1	88.2	90.9	83.6	86.7	89.2	77.5	66.9	89.9	85.1	74.0	91.4	83.0

Table IV: Additional results trained on a different source domain on DomainNet across architectures

Backbone	Pre-train. Data	Params	Source: Cl						Source: In					
			In	Pa	Qu	Re	Sk	AVG	Cl	Pa	Qu	Re	Sk	AVG
ResNet-50	ImageNet-1K	23	12.8	30.8	11.7	45.8	42.6	28.8	35.2	32.0	3.9	47.6	30.5	29.9
Swin-T	ImageNet-1K	27	16.5	37.5	14.0	55.4	44.7	33.6	39.9	35.7	4.6	53.4	32.9	33.3
ConvNeXt-T	ImageNet-1K	27	16.7	39.5	13.3	57.6	46.8	34.8	40.0	37.7	4.4	54.4	34.9	34.3
ResNet-101	ImageNet-1K	42	15.7	36.2	13.2	52.6	45.3	32.6	37.9	33.4	4.9	49.6	32.1	31.6
Swin-S	ImageNet-1K	48	18.9	41.5	14.6	59.8	48.3	36.6	44.6	39.4	5.9	57.2	37.8	37.0
ConvNeXt-S	ImageNet-1K	49	19.1	43.9	13.7	61.3	50.5	37.7	46.3	42.3	5.6	58.8	40.3	38.7
Swin-B	ImageNet-22K	86	26.2	52.8	16.7	70.5	56.8	44.6	59.2	53.0	9.9	70.9	49.7	48.5
ConvNeXt-B	ImageNet-22K	87	23.8	53.2	14.8	71.6	58.1	44.3	58.8	53.0	7.5	71.4	51.2	48.4
Swin-L	ImageNet-22K	194	29.5	56.5	16.9	72.9	60.1	47.2	63.5	56.1	9.7	73.6	53.3	51.2
ConvNeXt-L	ImageNet-22K	196	26.0	54.6	15.8	72.9	60.1	45.9	62.0	55.2	7.5	73.4	53.7	50.3
ConvNeXt-XL	ImageNet-22K	348	29.2	58.3	14.6	75.5	62.2	48.0	63.7	57.2	8.1	74.7	55.1	51.8

Backbone	Pre-train. Data	Params	Source: Pa						Source: Qu					
			Cl	In	Qu	Re	Sk	AVG	Cl	In	Pa	Qu	Sk	AVG
ResNet-50	ImageNet-1K	23	40.7	14.5	3.0	56.2	37.3	30.4	9.4	0.7	1.2	4.1	9.1	4.9
Swin-T	ImageNet-1K	27	47.5	16.7	6.8	60.4	40.6	34.4	24.9	1.9	6.2	13.3	13.8	12.0
ConvNeXt-T	ImageNet-1K	27	46.6	17.1	5.7	61.6	42.5	34.7	29.1	3.1	11.9	22.1	17.7	16.8
ResNet-101	ImageNet-1K	42	44.5	16.1	4.9	57.7	39.5	32.5	9.4	0.7	1.2	4.1	9.1	4.9
Swin-S	ImageNet-1K	48	50.3	19.1	6.6	62.9	43.3	36.4	28.6	2.6	8.5	17.1	16.0	14.6
ConvNeXt-S	ImageNet-1K	49	50.7	18.8	6.0	64.3	45.2	37.0	29.1	3.3	10.3	18.9	17.8	15.9
Swin-B	ImageNet-22K	86	60.8	25.8	7.7	72.5	51.6	43.7	32.1	3.1	9.3	18.8	19.3	16.5
ConvNeXt-B	ImageNet-22K	87	62.3	24.9	7.7	72.9	53.7	44.3	40.3	6.3	21.2	37.3	27.1	26.4
Swin-L	ImageNet-22K	194	64.5	28.7	9.0	75.1	53.8	46.2	40.3	6.3	20.5	35.6	26.2	25.8
ConvNeXt-L	ImageNet-22K	196	64.5	28.6	8.9	75.0	53.4	46.1	42.0	7.8	24.0	39.5	28.0	28.3
ConvNeXt-XL	ImageNet-22K	348	64.1	27.5	7.9	75.7	56.3	46.3	43.2	8.2	27.1	43.0	29.7	30.3

Backbone	Pre-train. Data	Params	Source: Re						Source: Sk					
			Cl	In	Pa	Qu	Sk	AVG	Cl	In	Pa	Qu	Re	AVG
ResNet-50	ImageNet-1K	23	46.6	17.3	45.2	6.5	35.3	30.2	52.4	13.3	36.3	12.3	47.5	32.3
Swin-T	ImageNet-1K	27	50.8	20.6	50.8	7.8	41.2	34.2	56.8	15.0	39.3	14.5	50.7	35.3
ConvNeXt-T	ImageNet-1K	27	49.3	19.8	37.3	7.3	37.3	30.2	56.6	15.1	41.5	13.7	53.5	36.1
ResNet-101	ImageNet-1K	42	49.3	19.2	48.6	8.7	38.5	21.8	53.8	12.9	35.5	13.7	43.2	31.8
Swin-S	ImageNet-1K	48	55.9	22.5	51.8	8.6	41.4	36.0	60.2	16.9	42.5	15.5	54.4	37.9
ConvNeXt-S	ImageNet-1K	49	54.9	22.2	52.8	8.1	43.0	36.2	60.7	17.6	44.6	14.1	57.6	38.9
Swin-B	ImageNet-22K	86	63.2	28.2	60.0	10.2	50.1	42.3	68.4	24.1	53.9	17.6	68.4	46.5
ConvNeXt-B	ImageNet-22K	87	62.7	26.9	59.9	9.4	52.1	42.2	68.9	22.3	54.4	15.0	68.4	45.8
Swin-L	ImageNet-22K	194	67.2	30.6	62.5	11.2	54.1	45.1	68.4	26.6	54.9	17.5	69.8	47.5
ConvNeXt-L	ImageNet-22K	196	66.6	28.8	61.6	10.0	54.3	44.3	69.9	24.2	56.4	16.6	69.8	47.4
ConvNeXt-XL	ImageNet-22K	348	67.7	29.7	62.2	11.4	55.5	45.3	70.4	26.0	58.2	16.4	73.5	48.9