Back-to-Bones: Rediscovering the role of backbones in domain generalization

Pattern Recognition 156 (2024) 110762

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Back-to-Bones: Rediscovering the role of backbones in domain generalization

Simone Angarano [a,b,*], Mauro Martini [a,b], Francesco Salvetti [a,b,c], Vittorio Mazzia [a,b,c], Marcello Chiaberge [a,b]

[a] Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy
[b] PIC4SeR, Politecnico di Torino Interdepartmental Centre for Service Robotics, Turin, Italy
[c] SmartData@PoliTo, Big Data and Data Science Laboratory, Turin, Italy

ARTICLE INFO

Keywords:
Deep learning
Computer vision
Image classification
Domain generalization
Backbone

ABSTRACT

Domain Generalization (DG) studies the capability of a deep learning model to generalize to out-of-training distributions. In the last decade, literature has been massively filled with training methodologies that claim to obtain more abstract and robust data representations to tackle domain shifts. Recent research has provided a reproducible benchmark for DG, pointing out the effectiveness of naive empirical risk minimization (ERM) over existing algorithms. Nevertheless, researchers persist in using the same outdated feature extractors, and little to no attention has been given to the effects of different backbones yet. In this paper, we go "back to the backbones", proposing a comprehensive analysis of their intrinsic generalization capabilities, which so far have been overlooked by the research community. We evaluate a wide variety of feature extractors, from standard residual solutions to transformer-based architectures, finding an evident linear correlation between large-scale single-domain classification accuracy and DG capability. Our extensive experimentation shows that by adopting competitive backbones in conjunction with effective data augmentation, plain ERM outperforms recent DG solutions and achieves state-of-the-art accuracy. Moreover, our additional qualitative studies reveal that novel backbones give more similar representations to same-class samples, separating different domains in the feature space. This boost in generalization capabilities leaves marginal room for DG algorithms. It suggests a new paradigm for investigating the problem, placing backbones in the spotlight and encouraging the development of consistent algorithms on top of them. The code is available at https://github.com/PIC4SeR/Back-to-Bones.

1. Introduction

The problem of induction has a central role in the learning process. Without generalization, machine learning algorithms would be able to exhibit useful behaviors only in situations identical to the ones previously experienced [1]. Deep neural networks are powerful models capable of extracting subtle regularities from training data. Nevertheless, they often fail to generalize to out-of-training data. Even if supervised training methodologies have proved to produce neural networks with remarkable performances, their results are valid only in well-defined settings and do not generalize across tasks, domains, and categories [2]. For the specific object recognition task, several literature works have shown that, unlike humans, training frameworks commonly produce networks that are more prone to be biased towards textures and global image statistics in making decisions [3,4], prioritizing easier-to-fit spurious correlations in favor of invariant shape cues [5]. That prevents scaling on all samples showing a distribution shift and poses a concrete barrier to deploying models in all critical applications that require true generalization power. For instance, autonomous driving could face environments and circumstances not encountered during the training phase caused by light, weather, background, and nearby object dynamics. Indeed, disparate independent studies report how neural networks could easily fail without effective generalization capabilities, negatively affecting the behavior of the overall system [6,7]. Similarly, another realistic example of a domain gap is training neural networks in simulation, which has become a standard procedure in the robotics research community. Recently, researchers have faced the Simulation-to-Reality (Sim2Real) gap problem, trying to effectively transfer Deep Neural Networks from virtual scenarios to the real world [8,9].

Domain Generalization (DG) aims at training models that generalize to out-of-distribution (OOD) data. The access to a set of source datasets provides a predictor with the ability to extract and learn general invariant patterns, which are, hypothetically, also recognizable

* Corresponding author at: Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy.
E-mail addresses: simone.angarano@polito.it (S. Angarano), mauro.martini@polito.it (M. Martini), francesco.salvetti@polito.it (F. Salvetti), vittorio.mazzia@polito.it (V. Mazzia), marcello.chiaberge@polito.it (M. Chiaberge).

https://doi.org/10.1016/j.patcog.2024.110762
Received 20 July 2022; Received in revised form 24 October 2023; Accepted 4 July 2024
Available online 9 July 2024

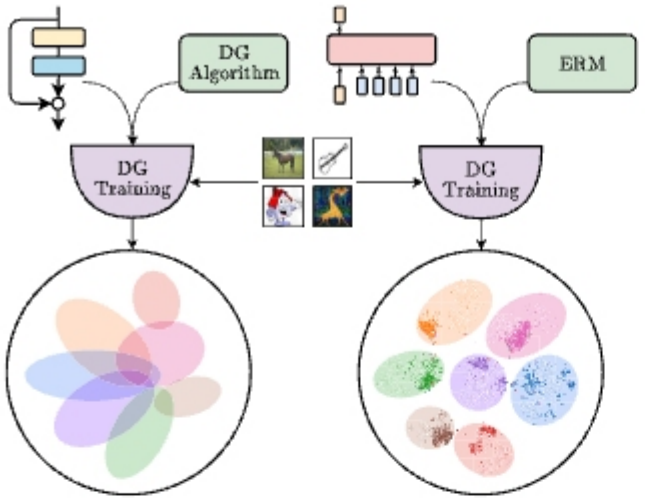


Fig. 1. Our experimentation proves the importance of backbones in Domain Generalization. We find that novel architectures, such as transformed-based models, lead to a better representation of data, outperforming outdated backbones, such as ResNets, and leaving marginal room for feature mapping improvement using DG algorithms.

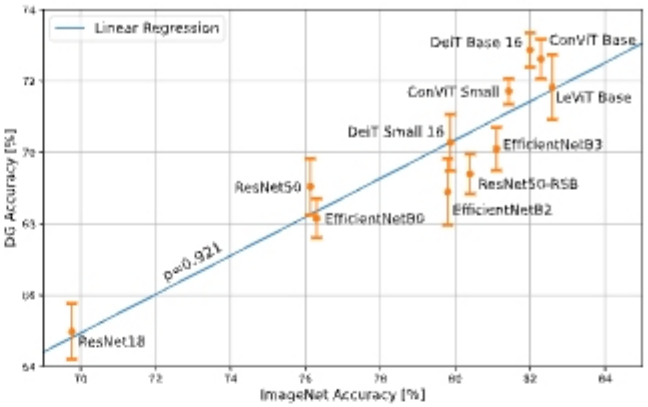


Fig. 2. DG accuracy achieved by tested backbones compared with their performance on ImageNet, with error bars. Regardless of different architectures and priors, we find a strong linear correlation between the two metrics ($\rho = 0.921$). In Section 3.1, we also compare DG accuracy with the number of parameters, finding a much weaker correlation.

in the target domain dataset [10,11]. As an extension of supervised learning, this approach aims to minimize empirical risk at training time to extrapolate an overall probability distribution from source datasets that enables accurate classification of OOD data. In the last decade, aware of the tremendous impact of generalization on computer vision applications, the DG research community has tackled the problem with algorithms that aim to find invariant features that hold with novel domains. Among the constellation of proposed approaches, we identify the principal broad strategies adopted for domain generalization in augmenting the source domain [12,13], aligning domain distributions [14–18], meta-learning [19–21], self-supervised learning [22–24], and regularization strategies [25–29].

Although methodologies have given meaningful insights about the nature of DG over the years, only recent research contributions have proposed a rigorous testing benchmark to evaluate and compare the advantages provided by DG algorithms fairly. With DomainBed [30], the results obtained by the most relevant solutions have been critically analyzed over DG datasets, unmasking the marginal positive or negative improvement obtained in most cases compared to naive empirical risk minimization (ERM). Nevertheless, the study has been carried out uniquely with ResNet50 [31] as a feature extractor. Thus, new DG algorithms are still proposed overlooking a fundamental aspect of practical deep learning applications: the importance of the backbone. In past years, several competitive deep learning architectures, characterized by different types of feature extractors, have been proposed to solve classification tasks [32] on popular datasets such as ImageNet [33]. Classical backbones are based on convolutional layers: AlexNet [34] is a network based on a small set of convolutions and max-pooling layers combined with ReLU activation. The VGG architecture [35], in its variations VGG-16 and VGG-19, further explores the convolution-pooling structure by stacking more layers and reaching a deeper design. ResNet [36] first adopted a residual approach to help gradient flow with skip-connections, and it is still a widely adopted backbone for various computer vision tasks. Similarly to VGG, ResNet has been proposed in different fashions, with variable depth, such as ResNet18, ResNet34, and ResNet50. Other architectures, such as DenseNet [37] or InceptionNet [38], focus on different mechanisms, like dense connections or parallelization of convolutional layers with different kernel sizes. MobileNet [39] and EfficientNet [40] have been proposed to increase model efficiency, reaching competitive classification results with lightweight architectures and fewer parameters. More recently, self-attention-based models have reached state-of-the-art image classification performance, inspired by the Transformer [41] architecture first proposed for language modeling. In particular, the Vision Transformer (ViT) [42] first adopted a Transformer encoder for vision tasks, while its training methodology has been refined by the Data Efficient ViT (DeiT) [43]; ConViT [44] combines convolutions with self-attention, and LeViT [45] focuses on a pyramidal architecture of self-attention layers that progressively shrinks spatial dimensions. This rich literature landscape offers a wide choice for researchers when selecting feature extractors for visual applications. However, among the different computer vision tasks, the DG community has substantially neglected the generalization power of existing backbones, promoting sophisticated algorithms combined with outdated feature extractors such as ResNet18 or even AlexNet. Only very few attempts have been made in this direction: Sultana et al. [46] proposed the first DG algorithm specifically for Transformer-based models; Guo et al. [47] studied how MLP-like models generalize better than CNN by incorporating more global-structure information and proposed a new Mixture-of-Experts architecture; a concurrent work by Li et al. [48] has brought useful insights on the intuition that multi-head attention is a low-pass filter with a shape bias, while convolution is a high-pass filter with a texture bias.

In this paper, we claim that the domain gaps existing in realistic scenarios should be tackled starting from accurately selecting the model architecture, which is undeniably central in most deep learning applications (Fig. 1). We come to similar experimental conclusions to the concurrent work of [48] on the generalization of transformers and the weaker effect of DG methods. However, we push it further by evaluating multiple backbones with different priors and several DG methodologies and find a strong correlation between ImageNet accuracy and generalization.

In particular, we conduct extensive experimentation on the principal DG datasets and assess a wide variety of backbone architectures, from novel vision transformers to standard convolutional models. Our results demonstrate an evident linear correlation between large-scale single-domain classification accuracy and domain generalization performance (Fig. 2). Moreover, we achieve state-of-the-art results in DG with naive ERM and simple data augmentation, remarking that, under fair testing conditions, the most promising algorithms presented so far give no substantial advantage.

We reinforce the experimentation with a visual analysis of the feature extractors. Using the t-SNE manifold learning technique [49] on extracted features, we show that novel backbones map same-class samples closer in the feature space and outperform older architectures

when trained in a DG framework. We propose a quantitative evaluation of this difference by fitting a k-NN classifier on the extracted features.

This study aims to promote a complete and meaningful approach to the domain generalization problem, avoiding isolated research efforts on DG algorithms and encouraging contributions that target the overall maximization of model generalization. Evidence in the literature shows that researchers from disparate application fields could significantly benefit from a shift of the DG paradigm towards realistic circumstances. For instance, data augmentation can automatically be exploited to generate a vast collection of artificial source domains. Domain Randomization fully exploits this principle [8], demonstrating its effectiveness in training agents in simulation for controlling manipulators accomplishing visual tasks [50] and autonomous racing drones [51]. That is further concrete proof that the success of domain generalization in real-world applications relies on simple ERM techniques, which offer an easy implementation together with a robust generalization boost.

The main contributions of this work can be, therefore, summarized as follows:

- We propose an extensive evaluation of backbones for domain generalization, showing remarkable improvements compared to literature results. We empirically find a linear correlation between large-scale single-domain classification accuracy and domain generalization performance (Fig. 2).
- We prove that adopting DG algorithms does not provide the expected generalization boost compared to naive ERM when using state-of-the-art feature extractors.
- We enrich the conducted experiments with an introspective study of the backbones, comparing the feature representations before and after the DG fine-tuning.

As an outcome of this work, we release Back-to-Bones,[1] a testbed to encourage the deep learning community to evaluate and compare the domain generalization performance of newly proposed backbones.

The rest of the paper is organized as follows. In Section 2, we briefly frame the DG theoretical background and introduce our backbone definition. In Section 3, we introduce our research outcome, describing the conducted approach and the criteria that guided the choice of backbones, model selection, hyperparameter optimization, and overall experimental framework; then, we report numerical results in conjunction with a visual introspection of the representations learned by the most relevant backbones under investigation. Section 4 discusses additional considerations about transformer-based backbones generalization and baseline selection in previous works. Finally, in Section 5, we present our conclusive remarks and suggestions for future works on DG.

2. The domain generalization framework

In this section, we first define necessary notations and concepts to frame the problem of domain generalization and empirical risk minimization. Secondly, we introduce a formal definition of a backbone and its constituents.

Problem Definition Given the input random variable X with values $x \in \mathcal{X}$, and the target random variable Y with values $y \in \mathcal{Y}$, the definition of *domain* is associated with the joint probability distribution P_{XY}, or $P(X,Y)$, over $\mathcal{X}\times\mathcal{Y}$. Supervised learning aims to train a classifier $f : \mathcal{X} \to \mathcal{Y}$ exploiting N available labeled examples of a dataset $D = (x_i, y_i)_{i=1}^{N}$ that are identically and independently distributed and sampled according to P_{XY}. The goal of the training process is to minimize the *empirical risk* associated with a loss function $l : \mathcal{Y}\times\mathcal{Y} \to [0, +\infty)$,

$$R_{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N} l(f(x_i), y_i) \tag{1}$$

by learning the classifier f. The dataset D is the only available source of knowledge to learn P_{XY}. We refer to this basic learning method as empirical risk minimization [52].

In domain generalization, a set of different K source domains $S = (S_k)_{k=1}^{K}$ is used to learn a classifier f that aims at generalizing well on an unknown target domain T. Each source domain is associated with its joint probability distribution P_{XY}^{k}, whereas P_{XY}^{S} indicates the overall source distribution learned by the classifier [53]. Indeed, DG aims to enable the classifier to predict well on out-of-distribution data, namely on the target domain distribution P_{XY}^{T}, by learning an overall domain invariant distribution from the source domains seen during training.

Backbone Definition We define a backbone $\mathcal{B} = f(\mathcal{A}, \mathcal{T}_{\mathcal{B}}, D)$ as a function of three elements: the model architecture $\mathcal{A}$, the training procedure $\mathcal{T}_{\mathcal{B}}$ (including optimization, regularization, and data augmentation), and the training data D. Consequently, all three factors introduce a certain degree of variability to the domain generalization accuracy:

$$\text{DG}_{\text{accuracy}}(S,T) = g(\mathcal{B}, \mathcal{T}_{DG}, \mathcal{N}_{exp}^{*})$$

where $\mathcal{T}_{DG}$ is the adopted DG training procedure and $\mathcal{N}_{exp}^{*}$ is the experimentation noise. $\mathcal{T}_{DG}$ usually includes a dedicated algorithm to cope with domain shifts. $\mathcal{N}_{exp}^{*}$ comprehends a systematic error due to the adopted model selection strategy and a random component caused by the stochasticity in the training process.

3. Back-to-bones

We set up our experimental benchmark to run a detailed analysis of the role of feature extractors in domain generalization. Besides choosing architectures, datasets, and DG algorithms to evaluate, particular attention is given to model selection strategy and statistical interpretation to obtain a fair and accurate benchmark. In the following subsections, we provide details on our experimental setup.

Backbones To be consistent with previous works, we include ResNet18 and ResNet50 [31] in the benchmark and compare them with some of the most successful architectures proposed in recent image classification research. We also consider the latest ResNet50 A1 [54], trained using the most recent practices in optimization and data augmentation and reaching a remarkable 80.4% top-1 accuracy on Imagenet1K. We include different sizes for each network to glimpse the effects of model dimension on DG accuracy. EfficientNet [55] demonstrated that systematical model scaling and dimension balancing yield remarkable results with fewer parameters. For this reason, we select three network versions, namely B0, B2, and B3. Finally, transformers [41] recently revolutionized deep learning by proving the effectiveness of self-attention for feature extraction; hence, four transformer-based architectures are included in the comparison. In particular, we choose DeiT (Small and Base) [43], ConViT [44] (both in its Small and Base configurations), and LeViT Base [45]. To provide further insights on the effect of additional pretraining data besides standard ImageNet [33], we also include Vision Transformer (ViT) [42] trained on ImageNet21K in its Small and Base versions. Regarding ViT Base, a configuration with a 32 × 32 patch size has been added to the standard 16 × 16 format to test the impact of patch size on DG. Further information on architectural details can be found in the cited papers. We report the number of parameters for each model in the last column of Table 1.

Datasets Among the various datasets created explicitly for DG in the last years, we use four of the most widely adopted ones for our primary experimentation. VLCS [56] considers four previous classification datasets as domains, while PACS [57] and Office–Home [58] focus more on style shifts (e.g. from photos to cartoons, sketches, and paintings). Terra Incognita [59] comprehends several animal photos taken with camera traps placed in different locations by day and night. To those, we add DomainNet [60], a bigger and more recent dataset that contains six domains divided by style and 345 classes. We use it

[1] https://github.com/PIC4SeR/Back-to-Bones

Table 1
Baselines comparison of different backbones for DG. We report the average accuracy over three runs and the associated standard deviation for each model. We include the results achieved by DomainBed with ResNet50 for reference. The models marked with * are pretrained on Imagenet21K instead of ImageNet1K. The rightmost column indicates the accuracy of the networks on ImageNet1K. In Appendix A, we report in detail the results obtained for all the domains.

Backbone	PACS	VLCS	Office–Home	Terra Incognita	Average	ImageNet	Parameters
ResNet18	80.51 ± 0.29	74.64 ± 0.61	63.87 ± 0.36	40.93 ± 1.85	64.99 ± 0.78	69.76	11.69M
ResNet50 [30]	85.50 ± 0.20	77.50 ± 0.40	66.50 ± 0.30	46.10 ± 1.80	68.90 ± 0.68	76.13	25.56M
ResNet50	83.85 ± 0.77	76.21 ± 1.20	68.79 ± 0.21	47.32 ± 0.97	69.04 ± 0.79	76.13	25.56M
ResNet50 A1	84.52 ± 0.68	78.37 ± 0.56	72.47 ± 0.13	42.23 ± 0.87	69.40 ± 0.56	80.40	25.56M
EfficientNetB0	85.46 ± 0.65	75.16 ± 0.34	67.27 ± 0.27	44.76 ± 0.94	68.16 ± 0.55	76.30	5.29M
EfficientNetB2	87.02 ± 1.37	75.44 ± 0.20	69.35 ± 0.24	43.80 ± 1.90	68.90 ± 0.93	79.80	9.11M
EfficientNetB3	86.71 ± 0.30	78.14 ± 0.18	69.84 ± 0.08	45.70 ± 1.84	70.10 ± 0.60	81.10	12.23M
DeiT Small 16	86.22 ± 1.33	79.47 ± 0.41	72.03 ± 0.33	43.40 ± 1.08	70.28 ± 0.79	79.87	22.05M
DeiT Base 16	**88.10 ± 0.48**	79.80 ± 0.32	76.35 ± 0.36	47.22 ± 0.75	72.87 ± 0.48	82.00	86.57M
ConViT Small	87.10 ± 0.33	80.00 ± 0.34	73.90 ± 0.17	45.83 ± 0.61	71.71 ± 0.36	81.43	27.78M
ConViT Base	87.27 ± 0.40	**80.31 ± 0.67**	76.51 ± 0.25	46.37 ± 0.89	72.62 ± 0.55	82.29	86.54M
LeViT Base	87.55 ± 1.50	78.91 ± 0.50	75.16 ± 0.13	45.68 ± 1.50	71.83 ± 0.91	82.59	39.13M
ViT Small 16*	83.59 ± 0.43	79.96 ± 0.60	77.25 ± 0.33	44.12 ± 1.07	71.23 ± 0.61	81.40	22.05M
ViT Base 32*	84.00 ± 1.17	78.46 ± 0.64	76.84 ± 0.17	36.71 ± 2.07	69.00 ± 1.01	80.72	88.22M
ViT Base 16*	**88.48 ± 1.22**	**80.05 ± 0.15**	**81.47 ± 0.21**	**49.77 ± 1.28**	**74.94 ± 0.72**	84.53	86.57M

to further stress the generalization capability of the best-performing backbones in the presence of more transfer learning data and fewer samples per class. We omit Rotated MNIST [61] and Colored MNIST [5] since we consider them too distant from any practical application. Moreover, from our perspective, simple rotation and colorization do not constitute actual domain shifts.

DG Algorithms We choose some of the most promising DG algorithms in recent research, particularly considering their performance on DomainBed [30]. Moreover, we select them to explore different approaches to the DG problem. CORAL [15] and MMD [17], indeed, focus on aligning the extracted features through second-order statistics (covariance). On the other hand, Mixup [62] works directly on input images, interpolating samples from different domains and considering the loss coming from both precursors. RSC [26], instead, introduces a heuristic that discards dominant features in the label determination, stimulating the model to rely on weaker data correlations. CausIRL [63] (used in combination with MMD or CORAL) builds from a causal analysis of generalization enforcing soft domain invariance to interventions on the source domain. CAD [64] introduces a contrastive adversarial domain bottleneck to guarantee convergence to target domains that preserve the Bayes predictor. ADDG [65] exploits a double mechanism (Intra-model and Inter-model) to diversify attention between features and suppress domain-related attention.

Data Augmentation Many research works prove that data augmentation plays a fundamental role in DG, as it can partially compensate for certain domain shifts [13]. That is particularly true in the presence of style changes, as popular data augmentation strategies involve the alteration of saturation, hue, and contrast. Since the effect of data augmentation on DG has already been investigated, in this paper, we use a standard setup to keep the focus on backbones. The de-facto standard augmentation strategy for DG, which we use in our benchmark, includes random cropping keeping at least 80% of the original image, horizontal flipping with 50% probability, image grayscaling with 10% chance, and random changes in color brightness, contrast, saturation, and hue, with a maximum of 40%. Since all the models are pretrained on ImageNet1K or ImageNet21K, input images are further normalized according to the mean and standard deviation of that datasets.

Model Selection To assess the DG capability of the considered pretrained networks, we fine-tune each of them on a set of K source domains S and test them on a target domain T. As pointed out by [30], "a domain generalization algorithm should be responsible for specifying a model selection method" and avoid improper comparisons between results obtained adopting different selection methods. In total agreement with their recommendations, we use the *training-domain validation set* strategy, which picks the model maximizing the accuracy on a validation split of the training set (in our case 10%, uniform across domains) at the end of each epoch. This selection method assumes that the average distribution of source domains is similar to that of the target domain on which the best model is tested.

Hyperparameter Search We conduct a random search for each backbone and dataset to determine the optimal training hyperparameters for the baselines. We define a range of values for continuous arguments and a set of choices for discrete ones, running approximately 32 iterations for each search and selecting the best combination via the previously defined model selection strategy. The learning rate is bounded in the range $[10^{-6}, 10^{-2}]$, choosing its scheduler among step (90% reduction after 80% of the epochs), exponential (with a decay in the range $[0.9, 1)$), and cosine annealing. The batch size and the number of training epochs are the same for all the experimentation, fixing their values at 32 and 30, respectively. Finally, we use cross-entropy loss and select the optimizer among SGD (with a momentum of 0.9) and Adam, keeping the weight decay to $5 \cdot 10^{-4}$.

Experimental Framework Our benchmarks are developed in Python 3 using the deep learning framework PyTorch. As the experimentation applies transfer learning to pretrained models, we use existing implementations of the considered backbones. Only the classification head is changed, adapting the network to the different number of classes. In particular, standard ResNets are taken from the PyTorch library *torchvision*,[2] EfficientNets from *EfficientNet-PyTorch*,[3] transformers and ResNet50 A1 from *timm*.[4] The implementations of DG algorithms are taken from DomainBed[5] and adapted to work with the architectures under test.

We repeat each training three times with different and randomly generated seeds to give more statistical information about accuracy results. In this way, both hyperparameter search and benchmarks cannot take advantage of the repeatability of trials, as data splitting, augmentation, and weight initialization change from one iteration to the next. Therefore, each of the results of our benchmark is reported as the mean over three repetitions, along with its standard deviation.

3.1. Baseline benchmark

The first analysis of our work consists of a precise and fair benchmark of the DG capabilities of recent deep learning architectures for image classification, trying to determine what solutions work best

[2] pytorch.org/vision/stable/models
[3] github.com/lukemelas/EfficientNet-PyTorch
[4] github.com/rwightman/pytorch-image-models
[5] github.com/facebookresearch/DomainBed

Table 2
Baseline comparison of a selection of the best backbones on DomainNet (*Clipart*, *Infograph*, *Painting*, *Quickdraw*, *Real*, and *Sketch* domains). We include the results achieved by DomainBed with ResNet50 for reference. The model marked with * is pretrained on Imagenet21K instead of ImageNet1K.

Backbone	C	I	P	Q	R	S	Avg
ResNet50 [30]	58.1	18.8	46.7	12.2	59.6	49.8	40.9
DeiT Base 16	69.1	25.0	55.8	17.1	69.3	57.0	48.9
ConViT Base	69.5	24.3	55.7	**17.7**	69.3	57.0	48.9
ViT Base 16*	**74.9**	**28.9**	**60.8**	17.5	**77.3**	**61.8**	**53.5**

and, possibly, why. Every pretrained backbone, after a hyperparameter search, is trained following the standard DG *leave-one-domain-out* procedure using the previously described model selection strategy. Our benchmark results are reported in Table 1 as the mean and standard deviation over three iterations.

Firstly, our benchmark highlights a strong correlation between DG accuracy and ImageNet performance. As depicted in Fig. 2, we find a direct proportionality between the two metrics (excluding the ViT models due to their different pretraining). We apply linear least-square regression and obtain a Pearson correlation coefficient $\rho = 0.921$. Indeed, a quick look at the results is sufficient to notice how newer and more performing backbones tend to achieve a higher DG accuracy on nearly all the datasets. That is primarily true for different sizes of the same architecture. ResNet50 reaches better results than ResNet18 for all the datasets, and the same happens for EfficientNet, ConViT, and ViT variants. For ResNet50, we also compare our results with those obtained by DomainBed and find comparable values. ResNet50 A1 benefits from its stronger pretraining, largely improving the accuracy obtained by the standard model on VLCS and Office–Home. However, Terra Incognita seems to penalize the network with its peculiar light conditions, resulting in a slight overall enhancement. Regarding different architectures, EfficientNetB2 performs very similarly to ResNet50 while the B3 version gains an additional 1% on them. Transformer-based models bring further improvements by exploiting their self-attention-based feature extraction, even in the case of DeiT Small and ConViT Small. In particular, they strongly outperform EfficientNet on Office-Home by over 4%, while Terra Incognita is the only dataset without any significant progress. That is probably due to the peculiarity of the domains, comprehending many night shots that can be challenging even for humans and rewarding less effective ImageNet pretraining. Among other transformers, DeiT Base 16 and ConViT Base prove to be the best, the latter being slightly more performing. Finally, the three ViT models show that pretraining on a more significant amount of data improves generalization. However, only ViT Base 16 registers a considerable step forward, suggesting that the abundance of data is fully exploited only by larger models. Nonetheless, ConViT Small performs similarly to the same-sized ViT Small 16, while larger patches demonstrate to degrade the accuracy of ViT Base 32. In conclusion, our results show how better DG comes from the union of a good feature extractor architecture and an optimal pretraining, as none of the two is sufficient alone. In Section 4, we further discuss the generalization capability of transformers. We stress the importance of adopting a good model selection strategy by comparing our ResNet18 baseline with various recent results obtained using the same backbone.

As an additional comparison, we plot the achieved DG accuracy compared to the number of parameters of the backbones (Fig. 3). Contrary to the graph of Fig. 2, in this case, the correlation between model dimension and generalization is much less marked, with a Pearson correlation coefficient (ρ) of 0.740. This confirms the central role of model architecture in DG tasks and our idea of backbone as the union of architecture, training procedure, and data.

Finally, we conduct an additional benchmark on the DomainNet dataset. Although representing a significant challenge for large-scale generalization, we choose to include DomainNet only in this second stage of the study due to its demanding computational nature and

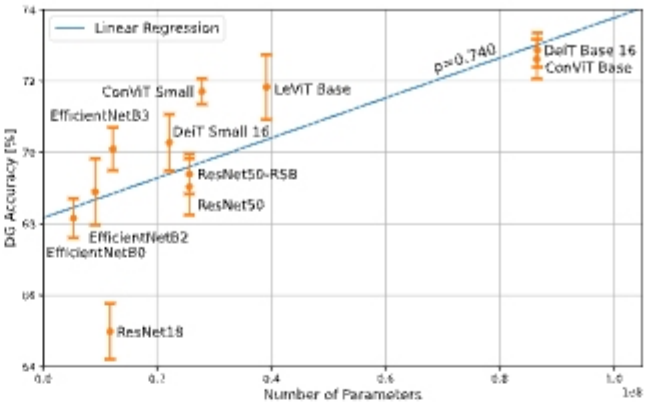


Fig. 3. DG accuracy achieved by tested backbones compared with their number of parameters, with error bars. We find a much weaker correlation between the two metrics ($\rho = 0.740$) than the one reported in Fig. 2.

strong class unbalancing. Indeed, our main intention is to promote a practical and accessible benchmark that aims to become a widespread reference for DG. We select only the best three models from the previous tests for this one (DeiT Base 16, ConViT Base, and ViT Base 16). In Table 2, we report the results achieved on each test domain, including those obtained by DomainBed on ResNet50 for reference. It is well evident that the feature extraction capabilities of modern backbones bring substantial improvement in all the domains, with an average increase in DG accuracy up to 12.6%. Moreover, ViT further enhances the results by exploiting its stronger pretraining.

3.2. Model introspection

After assessing the DG performance of different backbones, we propose a series of insights on how different architectures leverage training data to create their inner representation. First, we investigate the benefits of ImageNet pretraining for DG with a k-NN classifier, comparing ResNet50 and the best models from our benchmark. Then, we apply t-SNE [49] on the same extracted features to visualize how close same-class and same-domain samples are and the effect of fine-tuning on DG datasets. Finally, we inspect the attention maps of one of the transformer-based models to have a qualitative insight on the region of the images it focuses on.

K-NN Evaluation Firstly, we take ResNet50 and the best-performing models from our benchmark and evaluate their ability to tackle DG without fine-tuning. To do so, we use ImageNet weights to extract features from training domains and a k-NN (with $k = 5$) to fit that data. Then, we use test-domain images for the evaluation. To have a fair comparison with our benchmark, we use the same amount of training data, leaving out 10% of samples from source domains. The results in Table 3 show an overall difference of about 5% between ResNet50 and transformer-based models pretrained on ImageNet1k. This outcome is consistent with the generalization boost achieved in the standard DG framework (Table 1), although k-NN results tend to oscillate among different datasets. On the same trend, ViT Base 16 gains an additional 10% average accuracy, thanks to its pretraining on the larger ImageNet21K dataset. This outcome suggests that learning a wider overall source distribution P^S_{XY} is always needed to tackle a substantial domain gap effectively. That pretraining alone does not guarantee the ability to extract domain-invariant features.

Feature Mapping Visualization To further enlighten the role of backbones in extracting meaningful and invariant features to deal with DG,

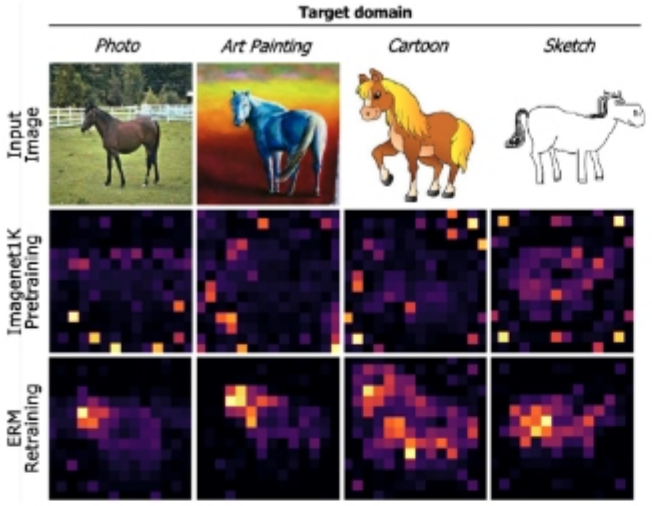


Fig. 4. DeiT Base attention maps when using the `[CLS]` token as a query for the different heads in the last layer. We select the same head for all examples. ERM encourages the backbone to focus on domain-invariant features, highly mitigating pretraining noise.

Table 3
Comparison of different feature extractors without fine-tuning, using a k-NN classifier ($k = 5$). The model marked with * is pretrained on Imagenet21K instead of ImageNet1K.

Backbone	PACS	VLCS	Office–Home	TerraInc.	Avg
ResNet50	56.04	69.57	56.26	14.75	49.16
DeiT Base 16	56.27	65.50	65.57	27.06	53.60
ConViT Base	56.83	64.50	66.63	**27.96**	53.98
ViT Base 16*	**75.14**	**75.14**	**82.72**	25.64	**64.66**

we can visualize the distributions in the feature space by projecting them in a two-dimensional space using t-SNE. Fig. 5 shows t-SNE visualization for ResNet50 and ConViT Base, pretrained on ImageNet1K and fine-tuned on PACS, targeting the *Art Painting* domain. For each model, we remove the classification head and extract the features for the whole dataset. The more clustered the same class features appear in the t-SNE, the more separable from other classes they are in the original space. We also include the silhouette score (S) as a quantitative metric of the separation of classes below each plot.

Fig. 5(c) shows how ResNet50 pretrained on ImageNet tends to map together same-domain samples and not same-class ones, being therefore unsuitable for DG without fine-tuning. After the fine-tuning process (Fig. 5(a)), the model achieves a better separation of source domain classes. However, many target domain samples are still mapped in the same space, far from the same-class source clusters (e.g. the *Art Painting* guitar example). Similarly to ResNet50, without fine-tuning, domains dominate the features space distribution of ConViT (Fig. 5(d)), causing several clusters of the same class but different domains to emerge in different locations (e.g. horse samples). However, some same-class samples of more similar domains, such as the guitars of *Cartoon* and *Art Painting*, are effectively clustered together. The fine-tuning process (Fig. 5(e)) distinctly pushes together same-class clusters, resulting in good generalization over the target domain. This analysis suggests that the ConViT backbone is more suited for DG than ResNet50 since it tends to give more similar representations to same-class samples from different domains. Additional feature mapping visualizations are presented in Appendix B.

Self-attention Visualization In literature, DG algorithms are often presented with a qualitative analysis, highlighting the regions the network focuses on using interpretation methods such as GradCAM [66]. Indeed, heat maps are brought as evidence of their capability to push attention toward more localized and domain-invariant features. Nevertheless, this section shows that competitive backbones with naive ERM can perfectly localize class-discriminative regions. In particular, Fig. 4 shows the attention maps extracted using the `[CLS]` token as a query for the different heads in the last layer of the DeiT Base architecture. We provide four random examples for different target domains of PACS showing the same attention head map before and after DG fine-tuning. It is remarkable how naive ERM is able to redirect attention towards more invariant features. Additional attention visualizations are reported in Appendix B.

3.3. Domain generalization algorithms

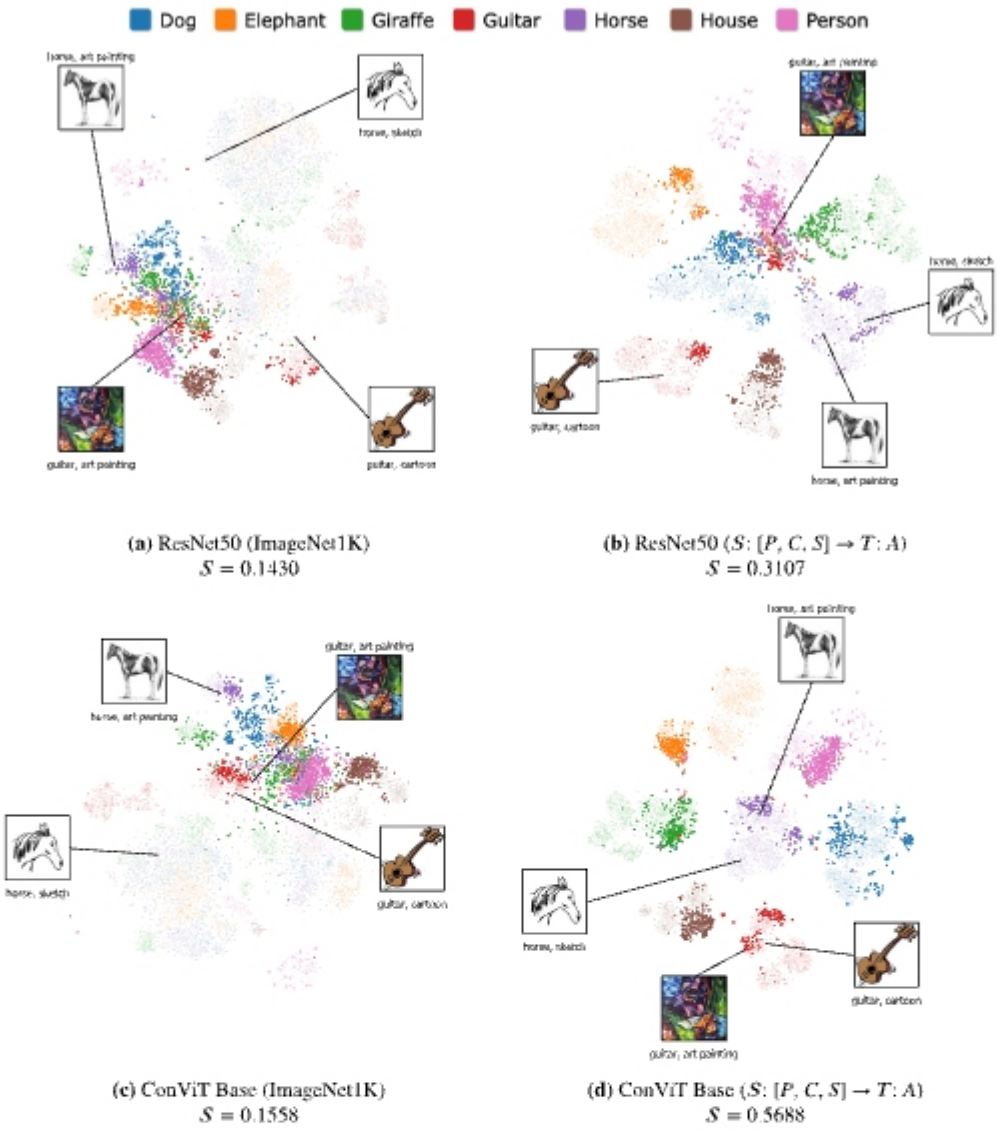
Domain generalization research mainly focuses on studying non-trivial algorithms to reduce the effect of domain shifts on classification accuracy. However, these algorithms are uniquely proposed in combination with outdated backbones such as ResNet50, ResNet18, or even AlexNet. According to the results in Table 1, recent backbones can provide significant improvements compared to ResNet50 with simple ERM. At this point, it is worth determining whether the application of DG algorithms brings a further boost in generalization to our baselines. To do so, we combine some of the most promising and recent algorithms available on DomainBed with three of our best baselines. We evaluate the methods introduced at the beginning of this Section (MMD, CORAL, Mixup, RSC, CAD, CausIRL CORAL, CausIRL MMD, and ADDG) using ViT Base 16, DeiT Base 16, and ConViT Base as backbones and repeating each training three times. Table 4 reports the obtained results, composed of average accuracy and associated standard deviation. Results obtained with ResNet50 are also reported directly from DomainBed for the same group of datasets as a reference. The only exception is the most recent ADDG, which the authors have not tested on VLCS and Terra Incognita and does not report standard errors.

As highlighted by the values in bold, the overall performance of ERM is equal to or better than other DG algorithms for all the considered datasets and backbones. Indeed, even where another methodology slightly outperforms ERM, the accuracy results mostly fall in the same confidence interval and hence differ very little statistically. We can then conclude from our experimentation that DG algorithms improve generalization properties marginally or even negatively for transformer-based backbones. This outcome extends the recent findings of DomainBed to other baselines and strongly reinforces the belief that choosing an effective backbone is the first step towards filling domain gaps. Adopting an outdated or poorly trained baseline is not the correct way to demonstrate the improvement derived from a DG algorithm. In the next section, we briefly ask ourselves what the reason behind this result is. Moreover, in Appendix A, we detail the results obtained for all the domains.

4. Additional considerations

4.1. Are transformer-based backbones better at generalizing?

Reflecting on experimental evidence and visual introspection from previous sections, we discuss whether transformer-based backbones are more robust to domain shifts in this paragraph. Undoubtedly, all baseline comparisons of Section 3.1 and features visualizations shown in Fig. 5 would suggest a positive answer to this interesting question. In all results and visual representations, self-attention-based models tend



(a) ResNet50 (ImageNet1K)
$S = 0.1430$

(b) ResNet50 (S: $[P, C, S] \rightarrow T$: A)
$S = 0.3107$

(c) ConViT Base (ImageNet1K)
$S = 0.1558$

(d) ConViT Base (S: $[P, C, S] \rightarrow T$: A)
$S = 0.5688$

Fig. 5. Backbone features visualization with t-SNE on PACS (Photo (P), Art Painting (A), Cartoon (C) and Sketch (S) domains). Target domain samples are highlighted. Some image examples from different domains and classes are visualized for better interpretability. After the fine-tuning, the ConViT Base architecture achieves a better class separation than ResNet50, clustering together same-class samples of different domains.

to generalize better to unseen domains. This result enforces the finding of [48] that multi-head attention acts as a low-pass filter with a shape bias thanks to its milder prior, while convolution is a high-pass filter with a texture bias.

Nevertheless, exercising caution and critically analyzing all the variables involved in the process is important. Indeed, such a conclusion only holds leveraging our backbone definition as a function of architecture $\mathcal{A}$, training procedure $\mathcal{T}_R$, and data D (as presented in Section 2). Architecture and training procedure are difficult to disentangle, and there is no guarantee that a training procedure optimal for a specific architecture remains the best for another. Therefore, that implies it is impossible to compare two different architectures directly. Some recent experimentation on residual architectures with current state-of-the-art training procedures has shed some light on the contribution of $\mathcal{A}$ to the generalization process. Indeed, in [54], a vanilla ResNet50 is trained with the approach developed by [43], reaching 80.4% top-1 accuracy on ImageNet without extra data or distillation. However, ResNet50 A1 performs only slightly better than the original model on our Back-to-Bones testbed, even if there is a difference of over 4% in ImageNet accuracy. That deviates slightly from the linear correlation described in Section 3.1 and suggests that a transformer-based architecture brings a significant generalization contribution. As further evidence of this trend, ConViT Small has comparable parameters with [54] and a similar training procedure but outperforms it by more than 4% on some datasets. Nonetheless, further experimentation can yield more comprehensive results on this interesting aspect of vision transformers.

4.2. On baseline selection in previous works

As already stated in Section 1, in the last decade, a plethora of algorithms for domain generalization (DG) has been proposed in

Table 4
Comparison between ERM and three promising DG algorithms on the best-performing backbones of our benchmark. We report the average accuracy over three runs and the associated standard deviation for each model. We highlight in bold the best result for each dataset, including ERM, when its accuracy is in the same confidence interval. We include the results achieved by DomainBed with ResNet50 for reference. The model marked with * is pretrained on Imagenet21K instead of ImageNet1K. In Appendix A, we report in detail the results obtained for all the domains.

Backbone	Algorithm	PACS	VLCS	Office-Home	Terra Incognita	Overall
ResNet50 [30]	ERM [52]	85.50 ± 0.20	77.50 ± 0.40	66.50 ± 0.30	46.10 ± 1.80	68.90 ± 0.68
	RSC [26]	85.20 ± 0.90	77.10 ± 0.50	65.50 ± 0.90	46.60 ± 1.00	68.60 ± 0.83
	Mixup [62]	84.60 ± 0.60	77.40 ± 0.60	68.10 ± 0.30	**47.90 ± 0.80**	69.50 ± 0.58
	CORAL [15]	**86.20 ± 0.30**	**78.80 ± 0.60**	**68.70 ± 0.30**	47.60 ± 1.00	**70.33 ± 0.55**
	MMD [17]	84.60 ± 0.50	77.50 ± 0.90	66.30 ± 0.10	42.20 ± 1.60	67.65 ± 0.78
	CausIRL CORAL [63]	85.80 ± 0.10	77.50 ± 0.60	68.60 ± 0.30	47.30 ± 0.80	69.80 ± 0.45
	CausIRL MMD [63]	84.00 ± 0.80	77.60 ± 0.40	65.70 ± 0.60	46.30 ± 0.90	68.40 ± 0.68
	CAD [64]	85.20 ± 0.90	78.00 ± 0.50	67.40 ± 0.20	47.30 ± 2.20	69.48 ± 0.95
	ADDG [65]	89.2	–	72.5	–	–
DeiT Base 16	ERM [52]	**88.10 ± 0.48**	**79.80 ± 0.32**	76.35 ± 0.36	47.22 ± 0.75	**72.87 ± 0.48**
	RSC [26]	85.37 ± 1.30	77.27 ± 0.51	76.47 ± 0.28	45.41 ± 1.50	70.97 ± 0.90
	Mixup [62]	85.67 ± 0.61	78.25 ± 0.60	75.96 ± 0.11	46.63 ± 0.49	71.32 ± 0.48
	CORAL [15]	85.13 ± 0.82	78.34 ± 0.86	76.48 ± 0.14	46.33 ± 1.83	71.38 ± 0.93
	MMD [17]	87.22 ± 0.28	78.71 ± 0.22	**77.03 ± 0.10**	**49.35 ± 1.42**	**73.08 ± 0.50**
	CausIRL CORAL [63]	83.86 ± 0.75	77.80 ± 0.40	76.12 ± 0.04	46.73 ± 0.81	71.13 ± 0.50
	CausIRL MMD [63]	85.46 ± 0.68	77.27 ± 0.42	76.53 ± 0.42	45.77 ± 1.66	71.26 ± 0.79
	CAD [64]	**87.74 ± 0.62**	79.28 ± 0.36	76.61 ± 0.15	47.46 ± 0.64	**72.77 ± 0.44**
	ADDG [65]	75.30 ± 0.34	78.28 ± 0.77	**77.58 ± 0.30**	29.14 ± 2.24	65.07 ± 0.91
ConViT Base	ERM [52]	**87.27 ± 0.40**	**80.31 ± 0.67**	76.51 ± 0.25	**46.37 ± 0.89**	**72.62 ± 0.55**
	RSC [26]	85.73 ± 0.81	79.05 ± 0.61	76.77 ± 0.26	44.94 ± 1.47	71.62 ± 0.79
	Mixup [62]	86.00 ± 0.45	80.00 ± 0.76	76.48 ± 0.16	43.95 ± 0.18	71.61 ± 0.39
	CORAL [15]	86.24 ± 0.24	79.62 ± 0.38	75.33 ± 0.22	44.41 ± 1.33	71.40 ± 0.54
	MMD [17]	86.84 ± 0.63	**80.72 ± 0.55**	**77.94 ± 0.31**	**46.78 ± 1.22**	**73.07 ± 0.68**
	CausIRL CORAL [63]	84.71 ± 0.31	79.14 ± 0.69	77.05 ± 0.16	45.63 ± 2.03	71.63 ± 0.80
	CausIRL MMD [63]	86.59 ± 0.96	**80.30 ± 0.56**	**77.92 ± 0.35**	**46.85 ± 0.59**	**72.92 ± 0.61**
	CAD [64]	**87.42 ± 0.66**	79.99 ± 0.41	**77.71 ± 0.09**	**46.77 ± 3.31**	**72.97 ± 1.12**
	ADDG [65]	86.34 ± 0.76	79.79 ± 0.30	76.29 ± 0.33	43.97 ± 1.75	71.60 ± 0.78
ViT Base 16*	ERM [52]	**88.48 ± 1.22**	80.05 ± 0.15	81.47 ± 0.21	49.77 ± 1.28	**74.94 ± 0.72**
	RSC [26]	86.58 ± 2.14	79.59 ± 0.63	78.74 ± 0.64	40.79 ± 1.41	71.42 ± 1.20
	Mixup [62]	**88.62 ± 0.54**	**80.77 ± 1.28**	**82.93 ± 0.07**	48.59 ± 0.92	**75.23 ± 0.70**
	CORAL [15]	84.60 ± 1.31	**80.89 ± 0.49**	80.92 ± 0.25	**50.58 ± 0.26**	74.25 ± 0.58
	MMD [17]	87.99 ± 0.08	79.54 ± 0.37	81.71 ± 0.28	49.40 ± 2.45	**74.66 ± 0.79**
	CausIRL CORAL [63]	**88.26 ± 1.09**	80.10 ± 0.91	81.73 ± 0.13	47.29 ± 2.64	74.35 ± 1.19
	CausIRL MMD [63]	86.57 ± 1.13	79.48 ± 1.12	81.62 ± 0.22	49.52 ± 0.58	74.30 ± 0.76
	CAD [64]	87.44 ± 0.53	78.79 ± 2.43	79.80 ± 0.36	39.45 ± 4.15	71.37 ± 1.87
	ADDG [65]	75.33 ± 0.54	77.77 ± 0.32	77.72 ± 0.09	25.60 ± 0.64	64.11 ± 0.40

Table 5
Comparison between the ResNet18 baseline obtained in our BacktoBones benchmark on PACS and those reported by popular DG works. Our accuracy result (without any extra component) outperforms all previous ones, which rarely include statistical information from multiple training iterations. Moreover, these works seldom discuss hyperparameter search procedures and model selection strategies.

Baseline	Average	Std Deviation
TRM [67]	77.13	1.53
MMLD [68]	78.70	–
JiGen [69]	79.05	–
Epi-FCR [70]	79.05	–
MASF [71]	79.23	0.15
SagNet [72]	79.26	–
DDAIG [73]	79.53	0.48
D-SAM [74]	79.55	–
PAdaIN [75]	79.72	–
MetaReg [19]	79.93	0.28
RSC [26]	79.94	–
BacktoBones	**80.51**	**0.29**

the literature, trying to tackle the problem with a wide variety of sophisticated methodologies. Nevertheless, our experimentation highlights that presented baselines often lack proper optimization. Table 5 compares the accuracy result obtained in our BacktoBones benchmark with those reported by several recent works. We evaluate ResNet18 on PACS as this is the most common setup, and the baseline we obtain with fair hyperparameter search and validation outperforms all those reported in the latest research works without adding any extra component. Moreover, statistical information is often absent in past DG works, overlooking proper hyperparameter search and model selection strategy discussion. In accordance with the outcomes of DomainBed, we hope to encourage the adoption of rigorous testing procedures, in conjunction with a standard model selection strategy, for transparent research results. With this study, we suggest that new DG algorithms should be analyzed based on adopting well-trained backbones. As a matter of fact, an advantage brought to underpowered baselines can be considered meaningless.

5. Conclusion and future work

In this paper, we deeply investigate the role of backbones in domain generalization, bringing back to light the fundamental contribution neglected by the community that a competitive feature extractor provides for generalizing to out-of-distribution data. According to our suggested backbone definition, novel architectural solutions such as DeiT, ConViT, and LeViT show remarkable improvements in reducing domain

Table 6
Baselines comparison of different backbones on the four considered DG datasets. We report the average accuracy over three runs and the associated standard deviation for each model. The models marked with * are pretrained on Imagenet21K instead of ImageNet1K.

Backbone	Photo	Art painting	Cartoon	Sketch
ResNet18	94.03 ± 0.49	79.65 ± 1.74	75.71 ± 0.75	72.63 ± 1.53
ResNet50	94.53 ± 0.54	82.86 ± 2.42	76.83 ± 4.70	81.18 ± 0.68
ResNet50 A1	97.84 ± 0.66	85.87 ± 0.60	74.43 ± 1.33	79.94 ± 0.86
EfficientNetB0	95.43 ± 0.23	82.29 ± 1.00	80.69 ± 1.55	83.45 ± 2.14
EfficientNetB2	96.61 ± 0.35	85.30 ± 1.44	82.75 ± 1.09	83.44 ± 3.21
EfficientNetB3	95.89 ± 0.60	83.82 ± 0.54	81.93 ± 1.23	85.20 ± 1.18
DeiT Small 16	98.38 ± 1.25	87.58 ± 2.13	81.32 ± 1.96	77.60 ± 0.88
DeiT Base 16	99.38 ± 0.15	90.74 ± 0.75	82.75 ± 1.07	79.52 ± 1.54
ConViT Small	99.16 ± 0.16	91.23 ± 0.57	81.58 ± 1.45	76.43 ± 1.38
ConViT Base	99.18 ± 0.18	91.05 ± 0.75	81.39 ± 1.61	77.47 ± 1.45
LeViT Base	98.04 ± 0.55	85.20 ± 2.67	86.21 ± 1.59	80.76 ± 2.15
ViT Small 16*	99.40 ± 0.12	89.94 ± 0.52	80.60 ± 2.46	64.40 ± 1.20
ViT Base 32*	99.30 ± 0.33	89.63 ± 0.28	80.22 ± 0.91	66.84 ± 4.60
ViT Base 16*	99.50 ± 0.17	93.25 ± 1.09	85.52 ± 2.64	75.67 ± 2.29

(a) PACS

Backbone	Caltech	Labelme	Pascal	Sun
ResNet18	95.60 ± 0.18	62.55 ± 1.29	72.80 ± 1.90	67.60 ± 1.63
ResNet50	96.09 ± 1.36	64.47 ± 1.72	73.43 ± 3.34	70.83 ± 3.25
ResNet50 A1	98.89 ± 0.16	63.23 ± 0.77	77.64 ± 1.84	73.72 ± 0.55
EfficientNetB0	96.84 ± 0.99	61.76 ± 0.40	70.43 ± 1.16	71.62 ± 1.01
EfficientNetB2	97.97 ± 1.27	63.94 ± 0.95	71.11 ± 1.52	68.73 ± 0.62
EfficientNetB3	97.36 ± 0.41	63.68 ± 0.99	76.32 ± 1.51	75.20 ± 2.09
DeiT Small 16	97.53 ± 0.38	64.91 ± 0.52	79.58 ± 1.21	75.85 ± 1.02
DeiT Base 16	97.79 ± 0.18	65.24 ± 0.29	78.06 ± 2.17	78.11 ± 1.41
ConViT Small	97.95 ± 0.31	64.98 ± 0.51	80.33 ± 1.07	76.72 ± 0.67
ConViT Base	97.95 ± 0.37	65.78 ± 0.36	79.22 ± 2.63	78.28 ± 2.01
LeViT Base	98.19 ± 0.30	64.52 ± 1.13	76.57 ± 0.74	76.38 ± 1.30
ViT Small 16*	97.90 ± 0.41	64.86 ± 1.99	79.41 ± 2.72	77.67 ± 0.96
ViT Base 32*	98.78 ± 0.71	64.66 ± 0.24	75.97 ± 1.40	74.46 ± 2.79
ViT Base 16*	97.69 ± 0.59	65.42 ± 2.28	79.55 ± 2.64	77.53 ± 0.41

(b) VLCS

Backbone	Product	Art	Clipart	Real world
ResNet18	73.88 ± 0.36	55.42 ± 0.91	52.73 ± 0.39	73.46 ± 0.20
ResNet50	77.16 ± 0.28	63.23 ± 0.46	56.26 ± 0.20	78.51 ± 1.00
ResNet50 A1	79.88 ± 0.57	69.28 ± 0.44	57.63 ± 0.68	83.11 ± 0.32
EfficientNetB0	75.96 ± 0.28	60.91 ± 0.72	54.56 ± 1.39	77.65 ± 0.06
EfficientNetB2	78.52 ± 0.67	62.99 ± 0.66	56.27 ± 0.45	79.62 ± 0.51
EfficientNetB3	79.37 ± 0.43	64.50 ± 0.83	55.13 ± 1.04	80.38 ± 0.41
DeiT Small 16	79.91 ± 0.55	69.30 ± 0.93	56.74 ± 0.63	82.16 ± 0.21
DeiT Base 16	83.55 ± 0.37	75.22 ± 0.55	61.07 ± 0.52	85.55 ± 0.64
ConViT Small	80.69 ± 0.37	72.45 ± 0.83	58.69 ± 0.17	83.79 ± 0.44
ConViT Base	83.21 ± 0.58	74.49 ± 0.50	62.58 ± 1.44	85.77 ± 0.31
LeViT Base	83.23 ± 0.14	72.85 ± 0.91	60.55 ± 0.56	84.02 ± 0.66
ViT Small 16*	84.79 ± 0.55	76.32 ± 0.82	60.50 ± 0.72	87.38 ± 0.58
ViT Base 32*	83.92 ± 0.37	75.28 ± 0.33	60.95 ± 0.71	87.21 ± 0.15
ViT Base 16*	88.39 ± 0.35	79.93 ± 0.87	67.71 ± 0.36	89.85 ± 0.89

(c) Office-Home

Backbone	L100	L38	L43	L46
ResNet18	44.48 ± 2.71	35.95 ± 2.87	49.26 ± 2.08	34.03 ± 1.50
ResNet50	53.19 ± 5.66	41.57 ± 2.48	54.10 ± 1.33	40.43 ± 1.42
ResNet50 A1	48.22 ± 1.90	39.41 ± 2.77	45.50 ± 2.49	35.77 ± 0.84
EfficientNetB0	44.73 ± 3.15	41.40 ± 4.13	54.14 ± 0.83	38.76 ± 0.99
EfficientNetB2	41.92 ± 0.75	41.42 ± 4.02	55.06 ± 1.74	36.80 ± 3.08
EfficientNetB3	48.93 ± 2.76	37.92 ± 3.69	58.27 ± 1.08	37.67 ± 2.40
DeiT Small 16	53.11 ± 2.52	30.64 ± 5.40	50.79 ± 2.44	39.07 ± 1.76
DeiT Base 16	58.53 ± 1.07	35.93 ± 1.42	52.57 ± 2.82	41.83 ± 2.12
ConViT Small	53.55 ± 1.06	36.41 ± 0.44	53.93 ± 2.09	39.43 ± 0.65
ConViT Base	52.17 ± 4.05	32.54 ± 3.26	57.30 ± 0.27	43.50 ± 2.44
LeViT Base	55.02 ± 3.48	36.23 ± 2.34	55.37 ± 0.92	36.11 ± 1.52
ViT Small 16*	54.10 ± 4.17	33.70 ± 1.04	50.15 ± 2.23	38.52 ± 2.25
ViT Base 32*	33.33 ± 3.86	28.84 ± 2.64	52.57 ± 2.82	32.09 ± 0.94
ViT Base 16*	58.65 ± 4.18	41.14 ± 2.12	56.83 ± 1.39	42.47 ± 2.62

(d) Terra Incognita

gaps with their intrinsic feature mapping mechanisms. They achieve state-of-the-art results in DG with naive ERM and data augmentation only. Hence, we point out that a complete domain generalization study should consider the choice of the backbone as the first step. Moreover, we claim that the advantage of adopting generalization algorithms should be proved using recent and effectively trained feature extractors.

The enhancement of architectures represents the main road to guide future research on DG. Mixture-of-Experts, for example, have recently shown remarkable generalization capabilities [47,48] and deserve further study. For this reason, we encourage the research community to evaluate novel backbones on the proposed testbed Back-to-Bones to maintain a benchmark dedicated to backbones in the DG problem. With this work, we do not add a methodology to the long list but ponder the current situation surrounding DG works, trying to shift towards more effective research. From a broader perspective, our research points out the fundamental role of backbones in DG. However, we did not thoroughly examine the backbone components responsible for the observed correlation between source and target accuracy. Indeed, architecture, backbone training procedure, and data could contribute differently to the measured generalization capabilities. Therefore, we believe that besides collecting additional results with the proposed benchmark, further studies aim to evaluate the role of backbone components in DG.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the data used in this work comes from existing publicly available datasets. The code will be open sourced.

Acknowledgments

This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics (PIC4SeR)[6] and SmartData@Polito.[7]

Appendix A. Additional benchmark results

A.1. Baseline benchmark

This section includes additional results from our benchmark of different backbones, described in Section 3.1. In particular, Table 6 reports accuracy results for each target domain, further highlighting the differences between different models. Besides some oscillations given by peculiar domains, the trend of novel and more performing backbones overcoming outdated ones is present for each target split. Each value is recorded as the mean over three independent runs, along with its standard deviation.

A.2. Domain generalization algorithms

As for the baseline benchmark, we report more detailed results for the experimentation described in Section 3.3. In particular, Table 7 includes accuracy results for each target domain to highlight the strengths and the weaknesses of the examined algorithms. The evaluated methodologies generally do not significantly benefit DG accuracy, performing similarly or even worse than naive ERM for nearly all target domains. Moreover, the unreliability of DG algorithms highlighted by our experimentation is further compounded by the fact that target

[6] https://pic4ser.polito.it
[7] https://smartdata.polito.it

Table 7
Comparison of different DG algorithms on the three best backbones from our benchmark, covering each domain of the four considered datasets. We report the average accuracy over three runs and the associated standard deviation for each model. The model marked with * is pretrained on Imagenet21K instead of ImageNet1K.

Backbone	Algorithm	Photo	Art painting	Cartoon	Sketch
DeiT Base 16	ERM [52] [52]	99.38 ± 0.15	90.74 ± 0.75	82.75 ± 1.07	79.52 ± 1.54
	RSC [26]	99.20 ± 0.12	87.65 ± 0.51	78.46 ± 3.18	76.19 ± 1.77
	Mixup [62]	99.28 ± 0.16	87.52 ± 0.81	77.20 ± 1.16	78.66 ± 2.36
	CORAL [15]	99.44 ± 0.15	87.97 ± 0.32	75.74 ± 3.67	77.38 ± 0.92
	MMD [17]	99.48 ± 0.03	89.44 ± 0.85	79.75 ± 0.65	80.20 ± 1.15
	CausIRL CORAL [63]	99.28 ± 0.26	86.56 ± 0.79	72.94 ± 1.35	76.65 ± 1.82
	CausIRL MMD [63]	99.04 ± 0.36	87.75 ± 1.02	77.69 ± 1.11	77.36 ± 2.70
	CAD [64]	99.54 ± 0.19	90.10 ± 0.44	81.08 ± 1.58	80.23 ± 1.42
	ADDG [65]	96.87 ± 0.88	87.50 ± 1.18	72.65 ± 2.22	44.16 ± 2.15
ConViT Base	ERM [52]	99.18 ± 0.18	91.05 ± 0.75	81.39 ± 1.61	77.47 ± 1.45
	RSC [26]	98.36 ± 1.33	89.96 ± 0.93	78.37 ± 2.05	76.22 ± 2.06
	Mixup [62]	99.32 ± 0.30	90.79 ± 0.71	78.94 ± 1.05	74.94 ± 0.55
	CORAL [15]	99.50 ± 0.09	90.67 ± 0.64	78.06 ± 0.64	76.73 ± 2.12
	MMD [17]	99.54 ± 0.12	90.84 ± 0.62	80.19 ± 1.10	76.77 ± 1.82
	CausIRL CORAL [63]	99.18 ± 0.30	88.92 ± 1.14	75.41 ± 0.60	75.32 ± 1.70
	CausIRL MMD [63]	99.50 ± 0.09	90.67 ± 1.15	79.45 ± 0.33	76.74 ± 2.43
	CAD [64]	99.48 ± 0.17	90.63 ± 0.83	82.76 ± 0.58	76.80 ± 1.36
	ADDG [65]	99.34 ± 0.12	90.14 ± 1.16	80.79 ± 0.86	75.11 ± 1.92
ViT Base 16*	ERM [52]	99.50 ± 0.17	93.25 ± 1.09	85.52 ± 2.64	75.67 ± 2.29
	RSC [26]	98.94 ± 0.40	92.07 ± 2.34	86.03 ± 1.18	69.26 ± 7.69
	Mixup [62]	99.60 ± 0.12	95.04 ± 0.32	87.17 ± 0.87	72.67 ± 0.94
	CORAL [15]	98.38 ± 1.00	93.08 ± 0.73	81.20 ± 1.39	65.75 ± 4.21
	MMD [17]	99.36 ± 0.18	94.56 ± 0.41	84.88 ± 2.18	73.14 ± 1.58
	CausIRL CORAL [63]	99.50 ± 0.12	94.07 ± 0.42	86.36 ± 0.14	73.11 ± 4.03
	CausIRL MMD [63]	99.42 ± 0.09	93.57 ± 1.27	83.96 ± 2.10	69.34 ± 2.53
	CAD [64]	99.52 ± 0.06	93.85 ± 0.91	84.91 ± 0.45	71.47 ± 1.65
	ADDG [65]	97.80 ± 0.54	88.35 ± 0.57	72.22 ± 0.62	42.96 ± 1.58

(a) PACS

Backbone	Algorithm	Caltech	Labelme	Pascal	Sun
DeiT Base 16	ERM [52]	97.79 ± 0.18	65.24 ± 0.29	78.06 ± 2.17	78.11 ± 1.41
	RSC [26]	97.95 ± 0.96	64.95 ± 0.26	71.42 ± 0.58	74.77 ± 0.79
	Mixup [62]	98.42 ± 0.30	63.72 ± 0.60	74.59 ± 2.19	76.25 ± 0.52
	CORAL [15]	97.74 ± 0.55	65.32 ± 0.95	73.55 ± 1.80	76.74 ± 0.43
	MMD [17]	98.09 ± 0.88	64.29 ± 1.13	74.74 ± 0.90	77.71 ± 0.08
	CausIRL CORAL [63]	98.87 ± 0.35	62.64 ± 0.33	73.95 ± 1.31	75.72 ± 0.96
	CausIRL MMD [63]	97.36 ± 0.59	63.67 ± 2.00	71.97 ± 0.48	76.09 ± 0.31
	CAD [64]	96.89 ± 0.77	65.30 ± 0.91	77.80 ± 1.56	77.14 ± 1.25
	ADDG [65]	99.22 ± 0.12	65.19 ± 0.89	77.17 ± 1.06	71.54 ± 1.99
ConViT Base	ERM [52]	97.95 ± 0.37	65.78 ± 0.36	79.22 ± 2.63	78.28 ± 2.01
	RSC [26]	98.19 ± 0.29	65.50 ± 1.21	77.89 ± 0.86	74.63 ± 1.54
	Mixup [62]	99.22 ± 0.14	64.57 ± 0.85	79.17 ± 2.81	77.04 ± 0.38
	CORAL [15]	98.35 ± 0.25	66.49 ± 0.75	77.14 ± 1.06	76.51 ± 0.41
	MMD [17]	97.92 ± 0.55	68.03 ± 0.23	79.44 ± 1.54	77.47 ± 0.17
	CausIRL CORAL [63]	99.20 ± 0.29	64.59 ± 1.07	77.53 ± 2.78	75.25 ± 0.31
	CausIRL MMD [63]	98.61 ± 0.34	65.86 ± 0.58	79.55 ± 1.40	77.16 ± 0.68
	CAD [64]	97.67 ± 0.12	65.49 ± 0.58	80.01 ± 1.44	76.80 ± 0.61
	ADDG [65]	98.45 ± 0.92	65.28 ± 1.00	79.02 ± 1.21	76.43 ± 1.38
ViT Base 16*	ERM [52]	97.69 ± 0.59	65.42 ± 2.28	79.55 ± 2.64	77.53 ± 0.41
	RSC [26]	98.23 ± 1.84	65.21 ± 0.81	78.02 ± 1.64	76.89 ± 1.38
	Mixup [62]	97.46 ± 0.46	66.87 ± 0.31	80.26 ± 4.70	78.51 ± 0.64
	CORAL [15]	98.78 ± 0.46	67.24 ± 0.47	78.33 ± 0.38	79.20 ± 1.20
	MMD [17]	98.07 ± 0.55	64.92 ± 0.64	77.98 ± 2.01	77.20 ± 0.15
	CausIRL CORAL [63]	97.03 ± 1.10	65.89 ± 0.73	79.93 ± 2.10	77.54 ± 3.33
	CausIRL MMD [63]	97.67 ± 0.40	65.59 ± 0.59	79.16 ± 3.52	75.50 ± 1.19
	CAD [64]	98.09 ± 0.12	60.30 ± 8.14	79.38 ± 2.61	77.37 ± 1.25
	ADDG [65]	99.43 ± 0.08	63.30 ± 2.83	76.41 ± 1.20	71.93 ± 1.89

(b) VLCS

Backbone	Algorithm	Art	Clipart	Product	Real world
DeiT Base 16	ERM [52]	75.22 ± 0.55	61.07 ± 0.52	83.55 ± 0.37	85.55 ± 0.64
	RSC [26]	75.11 ± 0.11	61.70 ± 1.26	83.07 ± 0.40	83.41 ± 0.22
	Mixup [62]	75.10 ± 0.75	60.83 ± 0.93	81.62 ± 0.20	81.37 ± 0.25
	CORAL [15]	74.60 ± 0.77	61.55 ± 0.06	83.31 ± 0.47	83.45 ± 0.28
	MMD [17]	75.53 ± 0.18	62.44 ± 0.91	83.36 ± 0.31	86.78 ± 0.14
	CausIRL CORAL [63]	74.69 ± 0.22	61.02 ± 0.25	82.56 ± 0.07	86.22 ± 0.22
	CausIRL MMD [63]	74.50 ± 0.73	61.79 ± 0.91	83.43 ± 0.15	86.40 ± 0.25
	CAD [64]	75.20 ± 0.40	61.76 ± 0.94	83.52 ± 0.15	85.95 ± 0.05
	ADDG [65]	76.62 ± 0.80	60.62 ± 1.58	85.82 ± 0.38	87.25 ± 0.27
ConViT Base	ERM [52]	74.49 ± 0.50	62.58 ± 1.44	83.21 ± 0.58	85.77 ± 0.31
	RSC [26]	75.95 ± 0.17	63.97 ± 0.32	83.49 ± 0.64	83.68 ± 0.36
	Mixup [62]	75.79 ± 0.60	62.95 ± 0.11	81.67 ± 0.13	85.52 ± 0.18
	CORAL [15]	75.71 ± 0.54	63.37 ± 0.24	81.18 ± 0.19	81.05 ± 0.52
	MMD [17]	76.46 ± 0.68	64.71 ± 0.64	83.79 ± 0.56	86.80 ± 0.24
	CausIRL CORAL [63]	75.16 ± 0.54	63.54 ± 0.73	82.86 ± 0.12	86.20 ± 0.24
	CausIRL MMD [63]	76.47 ± 0.25	64.74 ± 0.68	83.63 ± 0.28	86.84 ± 0.34
	CAD [64]	76.43 ± 0.37	64.31 ± 0.25	83.83 ± 0.08	86.26 ± 0.35
	ADDG [65]	74.42 ± 0.63	62.86 ± 0.54	82.41 ± 0.44	85.46 ± 0.30
ViT Base 16*	ERM [52]	79.93 ± 0.87	67.71 ± 0.36	88.39 ± 0.35	89.85 ± 0.89
	RSC [26]	76.99 ± 0.54	66.32 ± 0.34	85.42 ± 2.33	86.21 ± 0.53
	Mixup [62]	82.05 ± 0.31	69.70 ± 0.28	89.17 ± 0.34	90.77 ± 0.34
	CORAL [15]	79.95 ± 0.34	67.14 ± 0.44	87.98 ± 0.56	89.63 ± 0.23
	MMD [17]	80.48 ± 0.28	68.33 ± 0.59	88.12 ± 0.53	89.89 ± 0.26
	CausIRL CORAL [63]	80.16 ± 0.56	68.54 ± 1.48	88.30 ± 0.35	89.94 ± 0.29
	CausIRL MMD [63]	80.44 ± 0.17	68.22 ± 0.65	88.19 ± 0.10	89.65 ± 0.32
	CAD [64]	78.13 ± 0.60	66.89 ± 0.64	86.19 ± 0.24	87.99 ± 0.57
	ADDG [65]	76.93 ± 0.33	60.44 ± 0.34	85.87 ± 0.45	87.62 ± 0.30

(c) Office–Home

Backbone	Algorithm	L100	L38	L43	L46
DeiT Base 16	ERM [52]	58.53 ± 1.07	35.93 ± 1.42	52.57 ± 2.82	41.83 ± 2.12
	RSC [26]	56.55 ± 2.31	29.77 ± 5.20	51.91 ± 2.01	43.42 ± 0.73
	Mixup [62]	48.40 ± 1.09	35.62 ± 0.43	54.73 ± 1.02	47.76 ± 1.46
	CORAL [15]	52.28 ± 3.75	35.06 ± 3.15	52.32 ± 1.04	45.65 ± 2.33
	MMD [17]	57.11 ± 4.33	38.27 ± 2.38	56.38 ± 2.71	45.63 ± 2.58
	CausIRL CORAL [63]	48.73 ± 3.87	37.55 ± 2.86	54.17 ± 1.15	46.46 ± 1.32
	CausIRL MMD [63]	53.38 ± 3.61	33.51 ± 3.77	51.95 ± 1.46	44.23 ± 0.97
	CAD [64]	53.94 ± 1.92	38.07 ± 1.90	53.69 ± 1.79	44.11 ± 0.86
	ADDG [65]	23.78 ± 2.17	39.85 ± 7.85	29.76 ± 3.00	23.18 ± 3.28
ConViT Base	ERM [52]	52.17 ± 4.05	32.54 ± 3.26	57.30 ± 0.27	43.50 ± 2.44
	RSC [26]	48.27 ± 4.20	31.87 ± 1.45	55.82 ± 0.72	43.80 ± 1.03
	Mixup [62]	44.45 ± 1.81	29.82 ± 0.88	55.82 ± 0.28	45.73 ± 2.19
	CORAL [15]	45.91 ± 5.60	31.17 ± 4.07	56.03 ± 2.38	44.52 ± 1.37
	MMD [17]	48.39 ± 0.30	33.57 ± 3.10	58.02 ± 1.15	47.13 ± 2.89
	CausIRL CORAL [63]	43.63 ± 1.91	35.67 ± 6.21	57.33 ± 1.03	45.88 ± 0.57
	CausIRL MMD [63]	47.04 ± 3.78	36.39 ± 0.84	58.44 ± 0.65	45.54 ± 1.22
	CAD [64]	50.88 ± 4.26	34.43 ± 6.64	57.78 ± 1.78	44.01 ± 1.44
	ADDG [65]	43.57 ± 0.67	30.47 ± 4.60	56.32 ± 0.94	45.51 ± 2.66
ViT Base 16*	ERM [52]	58.65 ± 4.18	41.14 ± 2.12	56.83 ± 1.39	42.47 ± 2.62
	RSC [26]	48.83 ± 5.00	30.84 ± 6.25	52.39 ± 1.01	31.10 ± 3.84
	Mixup [62]	58.50 ± 2.43	38.26 ± 1.16	57.12 ± 2.32	40.49 ± 0.92
	CORAL [15]	59.77 ± 2.14	44.58 ± 0.92	56.31 ± 4.21	41.65 ± 1.70
	MMD [17]	60.50 ± 2.22	42.88 ± 4.99	54.90 ± 1.69	39.31 ± 5.26
	CausIRL CORAL [63]	55.03 ± 8.22	39.82 ± 2.94	53.98 ± 1.72	40.33 ± 1.73
	CausIRL MMD [63]	56.71 ± 2.52	42.57 ± 1.45	55.53 ± 1.40	43.28 ± 2.85
	CAD [64]	47.53 ± 11.7	32.31 ± 1.90	46.65 ± 6.57	31.29 ± 0.85
	ADDG [65]	23.91 ± 1.88	29.30 ± 2.84	26.73 ± 3.86	22.48 ± 3.86

(d) Terra Incognita

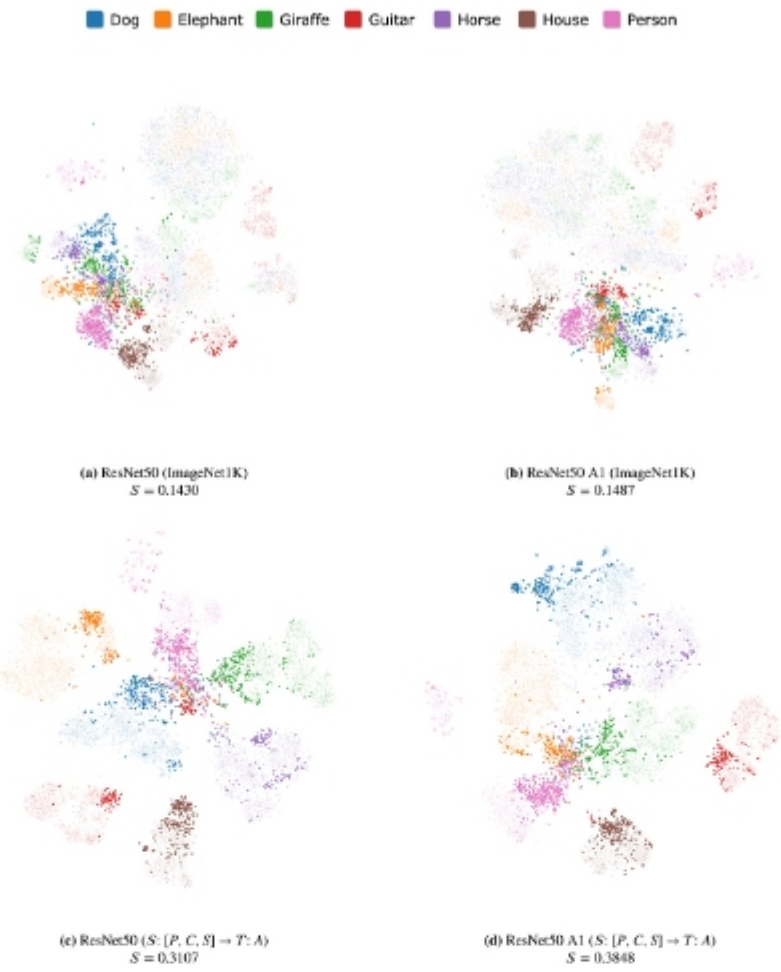


Fig. 6. PACS features extracted with ResNet50 and ResNet50 A1 projected on 2D space with t-SNE. Target domain *Art Painting* samples are highlighted. Even though ResNet50 A1 has a higher starting accuracy on ImageNet1K, the two backbones have comparable feature space distributions.

labels are rarely available in a real scenario. That prevents a direct check of the effectiveness of the procedure, leading to the adoption of more robust and trustworthy solutions. Indeed, ERM is easy to implement and consistently achieves remarkable accuracy results. Each value is recorded as the mean over three independent runs, along with its standard deviation.

Appendix B. Additional model introspection

B.1. ResNet50 vs ResNet50 A1

ResNet50 A1 [54] is a retrained version of the popular ResNet50, exploiting the most recent techniques in data augmentation and hyperparameter search, leading to an increased top-1 accuracy on ImageNet1K test set of 80.4%. Fig. 6 compares the t-SNE visualization of the two models including the silhouette score (S) as a quantitative metric of the separation of classes. Even though ResNet50 A1 starts with a remarkable advantage in terms of ImageNet accuracy, the two backbones generate comparable feature distributions. In particular, both models tend to separate samples by domain and not by class without fine-tuning (see Fig. 6(a)), which does not favor DG. After retraining on three source domains (*Photo*, *Cartoon* and *Sketch*), same-class clusters emerge, still with a certain overlapping over the *Art Painting* target domain.

B.2. Feature mapping visualization

Section 3.2 reports and discusses the visualization of the PACS domain *Art Painting* with ResNet50 and ConViT, highlighting the advantage of using transformer-based networks. In this section, we propose an additional t-SNE single-domain representation of features extracted from all PACS domains, with ResNet50 and our three best backbones (Fig. 7). According to the higher distance between source and target distributions, more challenging target domains result in more agglomerate clusters of domain samples. From this representation, the competitive advantage offered by transformer-based backbones is especially evident for *Art Painting*. ConViT shows more separated class features for the *Cartoon* domain. These findings confirm the baseline results reported in Section 3.1, in which transformers show valuable improvements on every target domain. We also include the silhouette score (S) in Table 8 as a quantitative metric of the separation of classes.

B.3. Self-attention visualization

We provide more self-attention visualizations for randomly selected PACS images: *Photo* and *Art Painting* domains in Fig. 8, *Cartoon* and *Sketch* in Fig. 9. We show the four most active heads of DeiT Base using the `[CLS]` token as a query for the different heads of the last layer. It is clear how ERM maps present more localized attention regions,

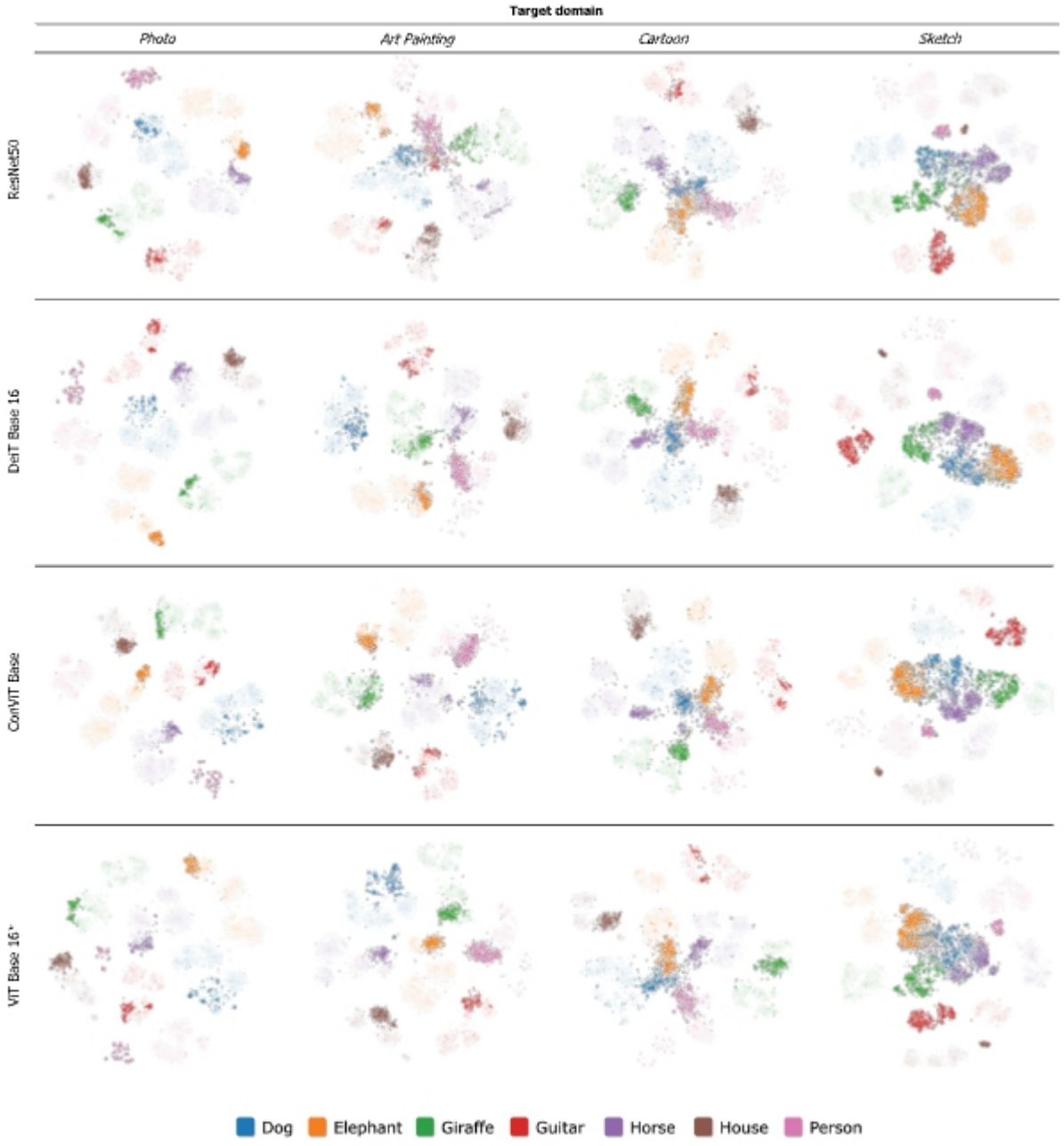


Fig. 7. The t-SNE representation of features extracted from all PACS target domains, with ResNet50 and several transformer-based networks, shows how better the same domain samples are divided into easier domains such as *Photo*. The *Sketch* distribution is affected by a more consistent domain gap, resulting in a more agglomerate domain cluster of samples. From this representation, the competitive advantage of transformer-based backbones is especially evident for *Art Painting*, although valuable in the classification accuracy on every target domain. The model marked with * is pretrained on Imagenet21K instead of ImageNet1K.

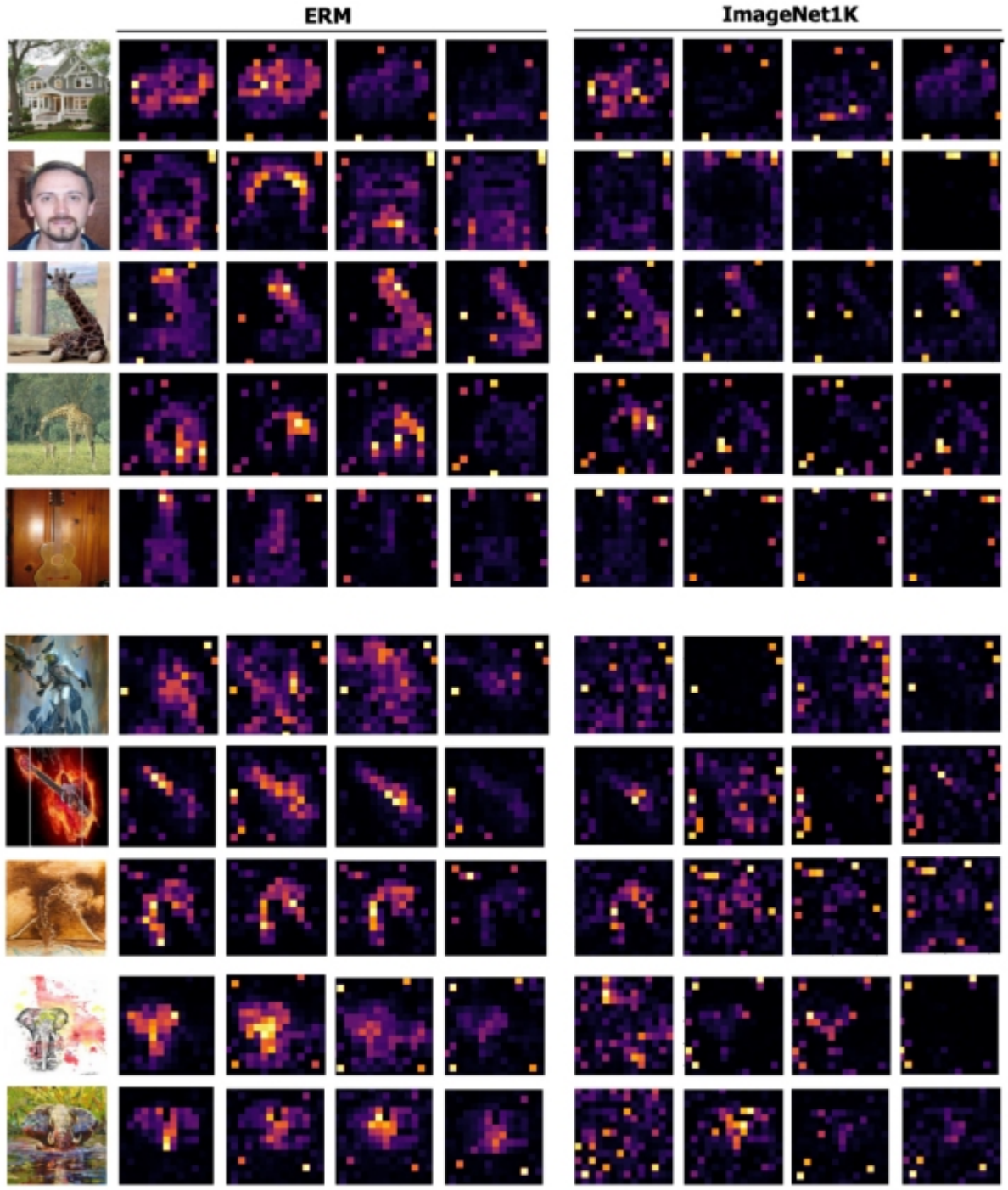


Fig. 8. Self-attention DeiT Base of most active heads of the last layer for some samples of the *Photo* and *Art Painting* PACS domains. We look at the attention map when using the `[CLS]` token as a query for the different heads in the last layer. It is clear how ERM is very effective at effectively redirecting attention toward more meaningful regions and mitigating pretraining noise.

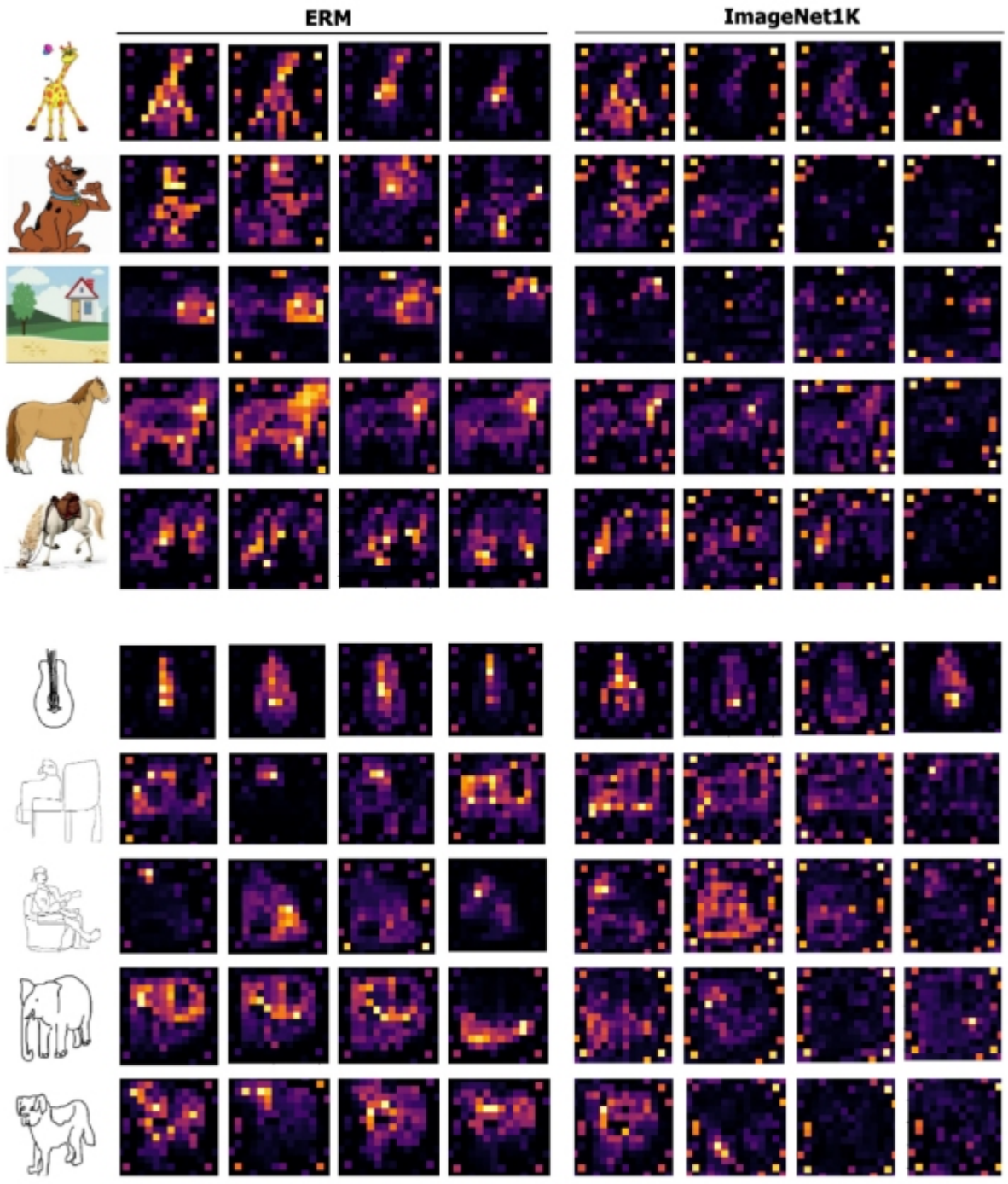


Fig. 9. Self-attention DeiT Base of most active heads of the last layer for some samples of the *Cartoon* and *Sketch* PACS domains. We look at the attention map when using the `[CLS]` token as a query for the different heads in the last layer. It is clear how ERM is very effective at effectively redirecting attention toward more meaningful regions and mitigating pretraining noise.

Table 8
Silhouette scores for the t-SNE representations in Fig. 7. The model marked with * is pretrained on Imagenet21K instead of ImageNet1K.

Backbone	Photo	Art painting	Cartoon	Sketch
ResNet50	0.7191	0.3107	0.4064	0.2930
DeiT Base 16	0.7689	0.5745	0.4720	0.4037
ConViT Base	0.7304	0.5688	0.4639	0.3921
*ViT Base 16**	0.5523	0.6778	0.5197	0.3093

focusing on more meaningful features. Finally, highly active isolated patches are learned during ImageNet training due to overfitting; even if some pretraining noise remains, ERM strongly attenuates this problem, further focalizing the attention of the network and reducing biased predictions.

References

[1] L. Valiant, Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World, Basic Books (AZ), 2013.

[2] G. Csurka, Domain Adaptation in Computer Vision Applications, Springer, 2017.

[3] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F.A. Wichmann, W. Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, in: International Conference on Learning Representations, 2019.

[4] L. Gatys, A.S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), in: Advances in Neural Information Processing Systems, vol. 28, Curran Associates, Inc., 2015.

[5] M. Arjovsky, L. Bottou, I. Gulrajani, D. Lopez-Paz, Invariant risk minimization, 2019, arXiv preprint arXiv:1907.02893.

[6] D. Dai, L. Van Gool, Dark model adaptation: Semantic image segmentation from daytime to nighttime, in: 2018 21st International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2018, pp. 3819–3824.

[7] G. Volk, S. Müller, A. von Bernuth, D. Hospach, O. Bringmann, Towards robust CNN-based object detection through augmentation with synthetic rain variations, in: 2019 IEEE Intelligent Transportation Systems Conference, ITSC, IEEE, 2019, pp. 285–292.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2017, pp. 23–30.

[9] M. Mozifian, A. Zhang, J. Pineau, D. Meger, Intervention design for effective Sim2Real transfer, 2020, arXiv preprint arXiv:2012.02055.

[10] G. Blanchard, G. Lee, C. Scott, Generalizing from several related classification tasks to a new unlabeled sample, in: Advances in Neural Information Processing Systems, vol. 24, 2011, pp. 2178–2186.

[11] K. Muandet, D. Balduzzi, B. Schölkopf, Domain generalization via invariant feature representation, in: International Conference on Machine Learning, PMLR, 2013, pp. 10–18.

[12] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, S. Sarawagi, Generalizing across domains via cross-gradient training, in: International Conference on Learning Representations, 2018.

[13] R. Volpi, H. Namkoong, O. Sener, J.C. Duchi, V. Murino, S. Savarese, Generalizing to unseen domains via adversarial data augmentation, in: Advances in Neural Information Processing Systems, vol. 31, 2018.

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016) 2030–2096.

[15] B. Sun, K. Saenko, Deep CORAL: Correlation alignment for deep domain adaptation, in: European Conference on Computer Vision, Springer, 2016, pp. 443–450.

[16] S. Motiian, M. Piccirilli, D.A. Adjeroh, G. Doretto, Unified deep supervised domain adaptation and generalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5715–5725.

[17] H. Li, S.J. Pan, S. Wang, A.C. Kot, Domain generalization with adversarial feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5400–5409.

[18] S. Chen, L. Wang, Z. Hong, X. Yang, Domain generalization by joint-product distribution alignment, Pattern Recognit. (ISSN: 0031-3203) 134 (2023) 109086.

[19] Y. Balaji, S. Sankaranarayanan, R. Chellappa, Metareg: Towards domain generalization using meta-regularization, Adv. Neural Inf. Process. Syst. 31 (2018) 998–1008.

[20] D. Li, Y. Yang, Y.Z. Song, T.M. Hospedales, Learning to generalize: Meta-learning for domain generalization, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[21] M.M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, C. Finn, Adaptive risk minimization: A meta-learning approach for tackling group shift, in: International Conference on Learning Representations, 2020.

[22] S. Bucci, A. D'Innocente, Y. Liao, F.M. Carlucci, B. Caputo, T. Tommasi, Self-supervised learning across domains, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2022) 5516–5528.

[23] I. Albuquerque, N. Naik, J. Li, N. Keskar, R. Socher, Improving out-of-distribution generalization via multi-task self-supervised pretraining, 2020, arXiv preprint arXiv:2003.13525.

[24] M.M. Rahman, C. Fookes, M. Baktashmotlagh, S. Sridharan, Correlation-aware adversarial domain adaptation and generalization, Pattern Recognit. (ISSN: 0031-3203) 100 (2020) 107124.

[25] S. Sagawa, P.W. Koh, T.B. Hashimoto, P. Liang, Distributionally robust neural networks, in: International Conference on Learning Representations, 2020.

[26] Z. Huang, H. Wang, E.P. Xing, D. Huang, Self-challenging improves cross-domain generalization, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020, pp. 124–140.

[27] S. Shahtalebi, J.C. Gagnon-Audet, T. Laleh, M. Faramarzi, K. Ahuja, I. Rish, SAND-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization, 2021, arXiv preprint arXiv:2106.02266.

[28] D. Kim, Y. Yoo, S. Park, J. Kim, J. Lee, Selfreg: Self-supervised contrastive regularization for domain generalization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9619–9628.

[29] M. Segu, A. Tonioni, F. Tombari, Batch normalization embeddings for deep domain generalization, Pattern Recognit. (ISSN: 0031-3203) 135 (2023) 109115.

[30] I. Gulrajani, D. Lopez-Paz, In search of lost domain generalization, in: International Conference on Learning Representations, Computer Vision Foundation, 2021.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.

[32] O. Elharrouss, Y. Akbari, N. Almaadeed, S. Al-Maadeed, Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches, 2022, arXiv preprint arXiv:2206.08016.

[33] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[34] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, vol. 25 (2012) 1097–1105.

[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[37] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[39] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.

[40] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.

[43] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 10347–10357.

[44] S. D'Ascoli, H. Touvron, M.L. Leavitt, A.S. Morcos, G. Biroli, L. Sagun, ConViT: Improving vision transformers with soft convolutional inductive biases, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 2286–2296.

[45] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, M. Douze, Levit: A vision transformer in convnet's clothing for faster inference, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12259–12269.

[46] M. Sultana, M. Naseer, M.H. Khan, S. Khan, F.S. Khan, Self-distilled vision transformer for domain generalization, in: Proceedings of the Asian Conference on Computer Vision, 2022, pp. 3068–3085.
[47] J. Guo, N. Wang, L. Qi, Y. Shi, ALOFT: A lightweight MLP-like architecture with dynamic low-frequency transform for domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24132–24141.
[48] B. Li, Y. Shen, J. Yang, Y. Wang, J. Ren, T. Che, J. Zhang, Z. Liu, Sparse mixture-of-experts are domain generalizable learners, in: The Eleventh International Conference on Learning Representations, 2022.
[49] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[50] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, et al., Domain randomization and generative models for robotic grasping, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 3482–3489.
[51] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, D. Scaramuzza, Deep drone racing: From simulation to reality with domain randomization, IEEE Trans. Robot. 36 (1) (2019) 1–14.
[52] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Netw. 10 (5) (1999) 988–999.
[53] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, C.C. Loy, Domain generalization: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2023) 4396–4415.
[54] R. Wightman, H. Touvron, H. Jegou, ResNet strikes back: An improved training procedure in timm, in: NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future, 2021.
[55] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 6105–6114.
[56] C. Fang, Y. Xu, D.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, ICCV, 2013.
[57] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017.
[58] H. Venkateswara, J. Eusebio, S. Chakraborty, S. Panchanathan, Deep hashing network for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017.
[59] S. Beery, G. Van Horn, P. Perona, Recognition in terra incognita, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 456–473.
[60] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, Computer Vision Foundation, 2019.
[61] M. Ghifary, W.B. Kleijn, M. Zhang, D. Balduzzi, Domain generalization for object recognition with multi-task autoencoders, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2551–2559.
[62] S. Yan, H. Song, N. Li, L. Zou, L. Ren, Improve unsupervised domain adaptation with mixup training, 2020, arXiv preprint arXiv:2001.00677.
[63] M. Chevalley, C. Bunne, A. Krause, S. Bauer, Invariant causal mechanisms through distribution matching, 2022, arXiv preprint arXiv:2206.11646.
[64] Y. Ruan, Y. Dubois, C.J. Maddison, Optimal representations for covariate shift, in: International Conference on Learning Representations, 2022.
[65] R. Meng, X. Li, W. Chen, S. Yang, J. Song, X. Wang, L. Zhang, M. Song, D. Xie, S. Pu, Attention diversification for domain generalization, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV, Springer, 2022, pp. 322–340.
[66] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[67] Y. Xu, T. Jaakkola, Learning representations that support robust transfer of predictors, 2021, arXiv preprint arXiv:2110.09940.
[68] T. Matsuura, T. Harada, Domain generalization using a mixture of multiple latent domains, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 11749–11756.
[69] F.M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, T. Tommasi, Domain generalization by solving jigsaw puzzles, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2229–2238.
[70] D. Li, J. Zhang, Y. Yang, C. Liu, Y.Z. Song, T.M. Hospedales, Episodic training for domain generalization, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2019.
[71] Q. Dou, D. Coelho de Castro, K. Kamnitsas, B. Glocker, Domain generalization via model-agnostic learning of semantic features, Adv. Neural Inf. Process. Syst. 32 (2019) 6450–6461.
[72] H. Nam, H. Lee, J. Park, W. Yoon, D. Yoo, Reducing domain gap by reducing style bias, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8690–8699.
[73] K. Zhou, Y. Yang, T. Hospedales, T. Xiang, Deep domain-adversarial image generation for domain generalisation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 13025–13032.
[74] A. D'Innocente, B. Caputo, Domain generalization with domain-specific aggregation modules, in: German Conference on Pattern Recognition, Springer, 2018, pp. 187–198.
[75] O. Nuriel, S. Benaim, L. Wolf, Permuted adain: Reducing the bias towards global statistics in image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9482–9491.

Simone Angarano is a Ph.D. candidate in Machine Learning at Politecnico di Torino. He is a member of the AI section of the Interdepartmental Centre for Service Robotics PIC4SeR (https://pic4ser.polito.it), where he focuses on creating efficient deep learning models for robot perception and control. In his research, particular attention is given to key aspects of real-world applications like generalization and robustness. He spent part of his Ph.D. at versity of Texas at Austin to work on efficient foundation vision models.

Mauro Martini is a Ph.D. student in Electrical, Electronics and Communication Engineering at Politecnico di Torino. He received from the Politecnico di Torino a Master's Degree with laude in Mechatronic Engineering in 2020, with the thesis "Visual based local motion planner with Deep Reinforcement Learning". He is now carrying out his research activity in collaboration with the Interdepartmental Centre for Service Robotics (PIC4SeR, https://pic4ser.polito.it). His research interests currently involve machine learning for autonomous navigation in service robotics, with a particular focus on perception and deep reinforcement learning based planners.

Francesco Salvetti is currently a Ph.D. student in Electrical, Electronics and Communications Engineering in collaboration with the interdepartmental centers PIC4SeR (https://pic4ser.polito.it) and Smart Data (https://smartdata.polito.it) at Politecnico di Torino, Italy. He received his Bachelor's Degree in Electronic Engineering in 2017 and his Master's Degree in Mechatronic Engineering in 2019 at Politecnico di Torino. He is currently working on Machine Learning applied to Computer Vision and Image Processing in robotics applications.

Vittorio Mazzia is a Ph.D. student in Electrical, Electronics, and Communications Engineering working with the two Interdepartmental Centres PIC4SeR (https://pic4ser.polito.it) and SmartData (https://smartdata.polito.it). He received a master's degree in Mechatronic Engineering from Politecnico di Torino, presenting a thesis entitled "Use of deep learning for low-cost automatic detection of cracks in tunnels", developed in collaboration with the California State University. His current research interests involve deep learning applied to different computer vision tasks, autonomous navigation for service robotics, and reinforcement learning. Moreover, he is currently working on machine learning algorithms and their embedded implementation for AI at the edge using neural compute devices (like Jetson Xavier, Jetson Nano, Movidius Neural Stick) for hardware acceleration.

Marcello Chiaberge is currently an Associate Professor within the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy. He is also the Co-Director of the Mechatronics Lab, Politecnico di Torino (www.lim.polito.it), Turin, and the Director and the Principal Investigator of the Interdepartmental Centre for Service Robotics (PIC4SeR, https://pic4ser.polito.it), Turin. He has authored more than 100 articles accepted in international conferences and journals, and he is the co-author of nine international patents. His research interests include hardware implementation of neural networks and fuzzy systems and the design and implementation of reconfigurable real-time computing architectures.