ON-DEVICE DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a systematic study of domain generalization (DG) for tiny neural networks. This problem is critical to on-device machine learning applications but has been overlooked in the literature, where research has been focused on large models only. Tiny neural networks have much fewer parameters and lower complexity, and thus should not be trained the same way as their large counterparts for DG applications. We find that knowledge distillation is a strong candidate for solving the problem: it outperforms state-of-the-art DG methods that were developed using large models with a large margin. Moreover, we observe that the teacher-student performance gap on test data with domain shift is bigger than that on in-distribution data. To improve DG for tiny neural networks without increasing the deployment cost, we propose a simple idea called out-of-distribution knowledge distillation (OKD), which aims to teach the student how the teacher handles (synthetic) out-of-distribution data and is proved to be a promising framework for tackling the problem. We also contribute a scalable method for creating DG datasets, called DOmain Shift in COntext (DOSCO), which can be applied to broad data at scale without much human effort. Code and models will be released.

1 INTRODUCTION

Domain generalization (DG), also known as out-of-distribution (OOD) generalization, is a problem concerned with whether or not a model learned from source data can perform well on unseen target data with domain shift (Blanchard et al., 2011; Muandet et al., 2013). In the last decade, DG has been extensively studied in the literature (Zhou et al., 2022a), especially for neural networks that have become the mainstream approach in machine learning and pattern recognition. However, existing research mainly focuses on large models with colossal parameter sizes and heavy computations.

This paper presents a systematic study on approaches *to improve DG for tiny neural networks*. Tiny neural networks are critical to on-device machine learning applications (Cai et al., 2022), which have received increasing attention due to the rapid increase in low-cost mobile devices, such as mobile phones and IoT devices. Running neural networks on mobile devices is challenging because it has strict requirements for model size (storage) and latency. For example, IoT devices with microcontroller units (MCUs) typically have an SRAM smaller than 512KB, which is too small to fit most neural networks (Lin et al., 2021). See Table 1 for a comparison of model specifications between neural networks of different sizes.

DG is essential for tiny neural networks because mobile devices are often used in an unanticipated environment, suggesting that the model must have the ability to overcome domain shift of any kind. However, making tiny neural networks domain-generalizable is non-trivial since they have much fewer parameters than large neural networks, and hence smaller capacity. Table 1 shows that the performance declines significantly for tiny models—especially on OOD data—which prompts the need to study DG for them.

Since tiny neural networks differ from large neural networks, they should not be trained the same way as the latter for DG applications. We observe that for tiny neural networks, state-of-the-art DG methods only bring limited improvements, like one or two percent increases in accuracy, over the Empirical Risk Minimization (ERM) model—known to be a strong baseline method (Gulrajani & Lopez-Paz, 2020). See the results of RSC (Huang et al., 2020), MixStyle (Zhou et al., 2021) and EFDMix (Zhang et al., 2022b) in Figure 1a.

Table 1: Large vs tiny neural networks. Due to capacity mismatch, the three tiny neural networks perform much worse, both in- and out-of-distribution, than the large counterpart, ResNet50.

Model	Params	Size	MACs	ID Acc	OOD Acc
ResNet50	25.56M	97.70MB	4133.74M	77.4%	60.2%
MobileNetV3-Small	1.50M	5.79MB	61.44M	64.0%	42.9%
MobileNetV2-Tiny	0.75M	2.91MB	65.20M	63.6%	42.0%
MCUNet	0.74M	2.87MB	145.13M	66.8%	46.4%

ID means in-distribution. OOD means out-of-distribution. Acc means accuracy (average performance on the DOSCO-2k benchmark). The Params, Size and MACs columns are measured on the 1,000-class ImageNet.

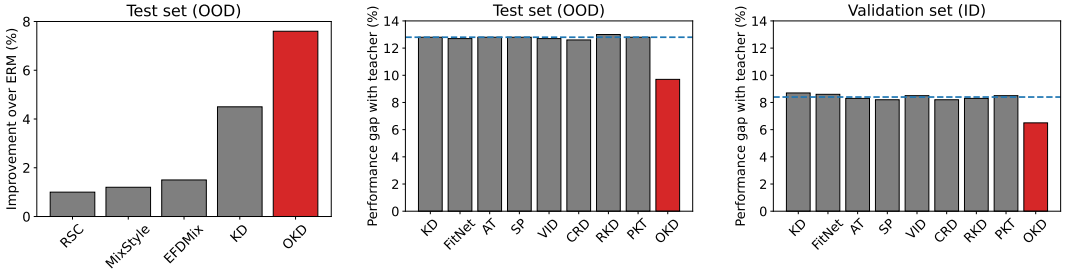
A straightforward solution is to use knowledge distillation (KD) (Bucilua et al., 2006; Hinton et al., 2015), which is an established technique for compressing large models by using their output as a supervision signal to train tiny models. We indeed find that KD boosts the DG performance more significantly than the DG methods, as shown in Figure 1a. Nevertheless, the teacher-student performance gap on OOD test data is still huge, which is about 12%, and such a gap exists for a wide range of KD-based methods, as evidenced in Figure 1b. Interestingly, we observe that the gap on OOD data is much bigger than that on in-distribution data (see Figure 1b and 1c). The results suggest that the capacity mismatch issue makes it harder to transfer generalizable knowledge from the teacher to the student.

We believe a student failed to match the teacher's OOD performance because the student was never taught how to handle OOD data—the conventional KD loss is computed on in-distribution data only. To mitigate the problem, we propose *out-of-distribution knowledge distillation* (OKD), a simple idea that extends KD by adding another distillation loss computed on OOD data. Since collecting OOD data for downstream datasets is challenging (and expensive), we resort to using image transformations to synthesize OOD data. Through a thorough investigation over a wide spectrum of image transformation methods, we identify the best match, i.e., combining CutMix (Yun et al., 2019) and Mixup (Zhang et al., 2017): the former twists local statistics while the latter perturbs global statistics, and together they make the synthetic images diverge from the support of the training data distribution but not completely disjoint. Despite the simplicity, OKD significantly improves upon KD without increasing the deployment cost (see Figure 1 for an overview of the main findings of this paper).

Furthermore, we contribute a new suite of DG datasets, constituting the *DOmain Shift in COntext* (DOSCO) benchmark. In particular, we observe that common visual domain shift is closely related to *contextual shift*, and can be automatically detected using a neural network trained on the Places dataset (Zhou et al., 2017). As suggested in the literature (Zhou et al., 2014), Places neural networks can extract meaningful patterns associated with the composition of scenes, and hence visual context. Unlike existing datasets that contain limited categories and types of domain shift, DOSCO is more diverse in both dimensions: it covers broader categories—e.g., generic objects, fine-grained categories like aircrafts and animals, and human actions—and a broader spectrum of domain shift in terms of image style, background, viewpoint and object pose. Therefore, DOSCO offers a more comprehensive toolkit to evaluate DG performance. More importantly, such an approach of synthesizing domain shift can be applied to broad data (any vision dataset) at scale without much human effort.

2 Approach

Tiny neural networks are much harder than large neural networks to train for DG applications due to their small capacity. To mitigate the problem, we propose out-of-distribution knowledge distillation (OKD). This simple idea significantly improves DG for tiny models while adding no extra parameter to the model, keeping the deployment cost unchanged. Below we briefly review KD and then give the technical details of OKD.



(a) Improvement over ERM.

(b) The teacher-student gap on out-of-distribution (OOD) data.

(c) The teacher-student gap on in-distribution (ID) data.

Figure 1: Overview of the results of some DG and KD-based methods obtained using tiny neural networks (i.e., MobileNetV3-Small) on the proposed DOSCO-2k benchmark. Our approach, out-of-distribution knowledge distillation (OKD), shows significant gains over the other methods in terms of both ID and OOD performance. The blue dashed lines in (b) and (c) denote the average performance of the KD-based methods.

2.1 Background on Knowledge Distillation

The main idea of KD (Bucilua et al., 2006; Hinton et al., 2015) is to use one or multiple (ensemble) big models, called *teacher*, to guide the learning of a small model, called *student*—the latter is more suitable for model deployment on resource-constrained devices, such as mobile phones or IoT devices. From the technical perspective, KD adds an auxiliary loss function (typically a distance measure) that encourages the student's output to mimic the teacher's output, along with a task-related loss, such as the cross-entropy for classification.

Formally, let $\boldsymbol{x}$ denote an input (i.e., an image in our case) and y the label, the overall loss function for KD can be written as

$$L_{KD} = \lambda H(y, f_S(\boldsymbol{x})) + (1-\lambda) D_{KL}(f_S(\boldsymbol{x}), f_T(\boldsymbol{x})), \tag{1}$$

where the first term is the cross-entropy loss, and the second term is the KL divergence between the output of the student f_S and teacher f_T. λ is a balancing weight, which is often set to 0.1 in the KD literature (Tian et al., 2020).

The prediction probability $\boldsymbol{p}$ for the input $\boldsymbol{x}$ is computed as

$$\boldsymbol{p}_i = \frac{\exp(\boldsymbol{z}_i/\pi)}{\sum_j \exp(\boldsymbol{z}_j/\pi)}, \tag{2}$$

where $\boldsymbol{z}$ denotes logits, and π is a temperature parameter that softens the probability. The common practice is to set $\pi = 1$ for the first term in Eq. 1 and $\pi = 4$ for the second term.

2.2 Out-of-Distribution Knowledge Distillation

The idea of OKD is simple: to teach the student how the teacher handles OOD data. This is vital for distilling generalizable knowledge from the teacher to the student but is missing in the conventional KD formulation.

Formulation Let $A(\cdot)$ denotes a data augmentation function, called OOD data generator, which aims to make the input deviate from the support of the data distribution. The formulation of OKD is as follows:

$$L_{OKD} = \lambda H(y, f_S(\boldsymbol{x})) + (1-\lambda)(D_{KL}(f_S(\boldsymbol{x}), f_T(\boldsymbol{x})) + D_{KL}(f_S(A(\boldsymbol{x})), f_T(A(\boldsymbol{x})))). \tag{3}$$

Compared with KD in Eq. 1, OKD adds a third term in Eq. 3 for distilling the teacher's knowledge about OOD data, which only adds negligible overhead during training while leaving the inference unchanged because the model architecture remains the same. See Figure 2 for an illustration of the OKD framework.

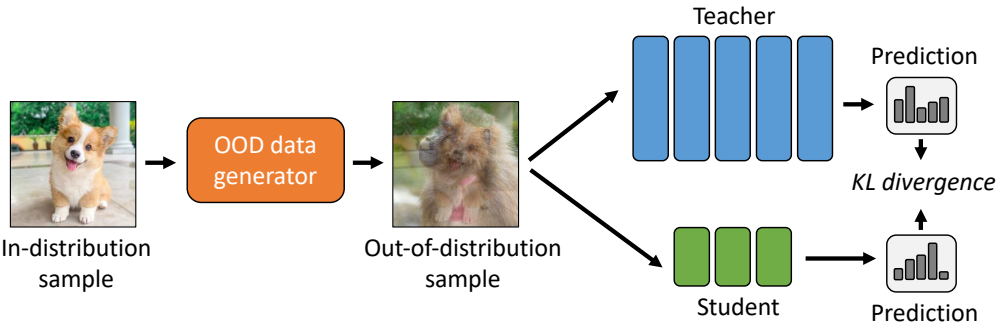


Figure 2: The main idea of out-of-distribution knowledge distillation (OKD) is to teach the student how the teacher handles out-of-distribution data, which is synthesized using image transformations.

OOD Data Generator Collecting extra OOD data for downstream datasets is impractical, so we use image transformations to synthesize OOD data. Specifically, we select a list of candidate transformations, as visualized in Figure 4a, and conduct a thorough investigation to identify the best one to be the OOD data generator $A(\cdot)$ in Eq. 3. It is worth noting that $A(\boldsymbol{x})$ does not necessarily have to maintain the semantics of $\boldsymbol{x}$, e.g., we can use CutMix or Mixup that may produce images not belonging to any existing class. The reason why this works is that the teacher's prediction (i.e., the soft probability distribution) on the OOD $A(\boldsymbol{x})$ also offers valuable insight for the student to learn. In the future, it would be interesting to explore more sophisticated formulations for the augmentation function, such as making it fully learnable.

3 Domain Shift in Context: The DOSCO Benchmark

Motivation Building DG datasets from scratch is non-trivial: one needs to first define domain labels—which are often difficult to describe using natural language—and then use them to collect data from particular sources. As a result, existing datasets are limited in diversity for both categories and types of domain shift—most are based on image style changes, such as PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017) and DomainNet (Peng et al., 2019).

Main Idea In computer vision, domain shift is often associated with visually perceptible changes in, for example, image style, background, viewpoint, contrast, object pose, and camera setups, to name a few. All these types of domain shifts can essentially be summarized as a shift in *visual context*. Based on this idea, we design a simple pipeline, called DOmain Shift in COntext (DOSCO), to automatically identify contextual information in images and thus create DG testbeds in an efficient way. Specifically, the DOSCO approach first trains a neural network on the Places dataset (Zhou et al., 2017) to extract contextual information in images, i.e., the composition of scenes that encapsulates all relevant image attributes mentioned above. Then, images are clustered using features extracted by the neural network to synthesize domain labels.

Implementation Details We follow He et al. (2022) to fine-tune a ViT-Large model (Dosovitskiy et al., 2020) on Places365 (Zhou et al., 2017), which was pre-trained on ImageNet (Deng et al., 2009) using the masked autoencoders approach (He et al., 2022). The model is dubbed *PlacesViT* hereafter. The DOSCO approach is then applied to seven image datasets widely used in transfer learning (Zhai et al., 2019; Zhou et al., 2022b) and covering a diverse set of visual recognition problems: (1) FGVCAircraft (Maji et al., 2013), (2) StanfordCars (Krause et al., 2013), (3) Caltech101 (Fei-Fei et al., 2004), (4) Omni-Instrumentality (Zhang et al., 2022c), (5) Omni-Mammal (Zhang et al., 2022c), (6) OxfordPets (Parkhi et al., 2012), and (7) UCF101 (Soomro et al., 2012). For each dataset, the images within each class are clustered using K-means ($K=10$) based on the PlacesViT features. For each class, a random 50/50 split on the domain labels is then performed to produce a training set and a test set. A held-out validation set is randomly sampled from within the training set using a $2:8$ ratio. Each dataset has *three* random splits so average results over them will be used for comparison. Example images can be found in Appendix A.1.

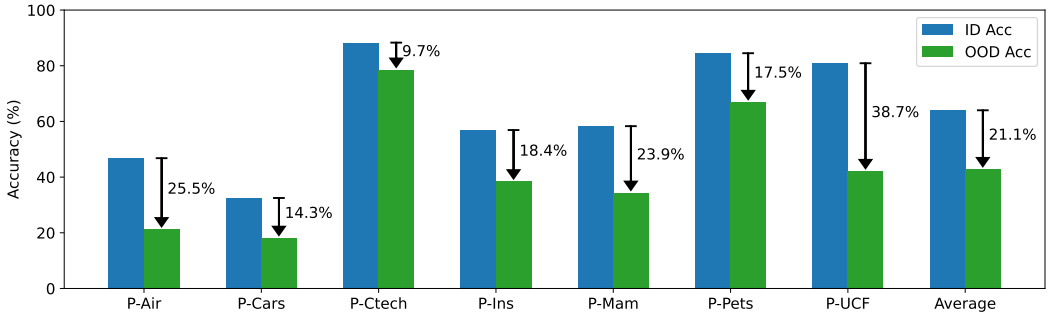


Figure 3: Comparison of performance on in-distribution (ID) and out-of-distribution (OOD) data on the DOSCO-2k benchmark. The results are obtained by an ERM model based on MobileNetV3-Small.

The DOSCO Benchmark For brevity, we call the seven datasets P-Air, P-Cars, P-Ctech, P-Ins, P-Mam, P-Pets, and P-UCF, respectively. "P" means these datasets are created using a Places neural network. The results shown in Figure 3 illustrate that the synthesized train-test domain gaps significantly challenge the generalization performance of neural networks. Following Zhai et al. (2019), we focus on transfer learning and thus create a 2k version of the datasets where the training and validation data for each dataset consists of 2,000 images in total (1,600 for training and 400 for validation).[1] All experiments conducted in this paper use **DOSCO-2k**. Note that domain labels are unavailable during training, which simulates a challenging yet realistic scenario.

4 Experiments

Model Architectures We use MobileNetV3-Small (Howard et al., 2019) as the primary tiny model throughout the paper, unless otherwise specified. ResNet50 (He et al., 2016) is used as the teacher model for KD-based methods. We also evaluate using two other tiny models: MobileNetV2-Tiny (Lin et al., 2020) and MCUNet (Lin et al., 2020), both specifically designed for MCUs. The specifications of these models can be found in Table 1. All models (both teacher and student) are pre-trained on ImageNet—since we focus on downstream transfer learning performance.

Training Details The batch size is set to 32. SGD with momentum is used as the optimizer. The learning rate starts from 0.01 and decays with the cosine annealing strategy. The maximum epoch is set to 100. Generic data augmentation methods are used, including random crop and flip. As mentioned before, the balancing weight λ in Eq. 3 is fixed to 0.1, which has been used in most KD methods (Wang & Yoon, 2021).

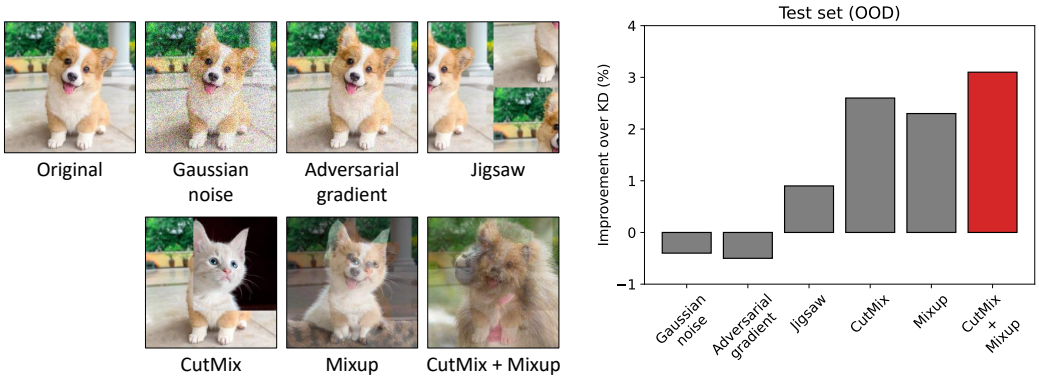
Baseline Methods Since DOSCO-2k does not provide training domain labels, we choose top-performing DG methods that do not need such labels for training to compare, including (1) ERM, (2) RSC (Huang et al., 2020), (3) MixStyle (Zhou et al., 2021), and (4) EFDMix (Zhang et al., 2022b). We also compare with the classic logit-based KD method (Hinton et al., 2015).[2]

Model Selection Model selection is a critical step when evaluating DG algorithms (Gulrajani & Lopez-Paz, 2020). We assume the model can only see source data during training, and use the source-domain validation data for model selection. Specifically, the model with the best validation performance achieved within the 100 training epochs is deployed on the test data.[3]

[1] We suggest that only the validation data should be used for parameter tuning for future work.

[2] Most other KD-based methods achieve similar performance with KD on DOSCO-2k (see Figure 1), so they are excluded from the main tables for comparison.

[3] To better track the progress on our benchmark, we suggest future work should conduct hyper-parameter tuning using the same model selection method, i.e., choosing parameters that give the best in-distribution validation performance.



(a) A list of candidates for the OOD data generator.

(b) Results of the candidates on DOSCO-2k.

Figure 4: Investigating the best OOD data generator for OKD.

4.1 Choosing OOD Data Generator

A thorough experiment is conducted to identify the best OOD data generator for OKD. Specifically, we compare some commonly-used image transformation methods, as listed in Figure 4a. These methods meet our requirements on the OOD data generator as they can make images deviate from the support of the data distribution but not completely disjoint—we show later that using completely disjoint data does not help. The results, specifically the absolute improvements over KD, are shown in Figure 4b. Most image transformation methods can bring some improvements except the two noise-based methods, i.e., Gaussian noise and adversarial gradient. Adversarial gradient is worse than Gaussian noise, suggesting that the teacher's "knowledge" about samples close to the decision boundary is not helpful—even the teacher itself would be confused by these examples. CutMix and Mixup give the best performance and the combination of them further improves the performance.[4] Compared with jigsaw, which destroys the global structure, Mixup maintains the global structure to some extent while CutMix only twists the local statistics (like simulating occlusion), which are probably why CutMix and Mixup are best for synthesizing "realistic" OOD examples.

4.2 Main Results on DOSCO-2k

The results on DOSCO-2k are reported in Table 2. The first block contains DG methods, which were previously developed using large models only. In most cases, the two feature-based data augmentation methods, i.e., MixStyle and EFDMix, achieve better performance than the regularization method, RSC, which mutes most predictive subsets of neurons during training. Overall, our observations on the DG methods here echo those reported in a recent work that performed a fine-grained analysis over various large models in DG (Wiles et al., 2022): (i) Not a single DG method can consistently beat ERM, e.g., none of the DG methods outperforms ERM on P-Ins; (ii) Augmentation-based methods are generally a better option to use. However, it is worth mentioning that when using smaller models (see Section 4.4), the conclusions made here should be adjusted.

When it comes to the second block, which contains KD-based methods, the margins over the DG methods and ERM are significant, demonstrating the potential of KD-based methods for solving on-device DG. Compared with KD, OKD clearly shows more potential in tackling the problem, as evidenced by the large margin. In sum, the results justify the effectiveness of the idea of teaching the student how the teacher handles OOD data.

Nonetheless, there is still a 10% gap on average with the large model (i.e., KD's teacher), which means the problem of making tiny neural networks domain-generalizable is yet to be solved.

[4] CutMix+Mixup is implemented as $\alpha\,\text{Mixup}(\boldsymbol{x}) + (1-\alpha)\,\text{CutMix}(\boldsymbol{x})$ where α is sampled from a Beta distribution. Unlike CutMix or Mixup, CutMix+Mixup may produce a mixture of three examples (see the CutMix+Mixup example in Figure 4a).

Table 2: Domain generalization results on DOSCO-2k using MobileNetV3-Small.

	P-Air	P-Cars	P-Ctech	P-Ins	P-Mam	P-Pets	P-UCF	*Avg*
ERM	21.3	18.2	78.6	38.5	34.4	67.0	42.2	42.9
RSC	23.0	20.8	79.0	38.0	34.9	68.4	42.9	43.9
MixStyle	24.6	22.2	80.9	37.5	32.3	67.3	43.7	44.1
EFDMix	27.3	23.4	80.4	37.0	32.3	67.3	42.8	44.4
KD	29.7	26.2	82.4	39.8	37.5	69.3	46.6	47.4
OKD	**32.1**	**30.4**	**84.4**	**42.0**	**40.8**	**73.1**	**50.4**	**50.5**
KD's teacher	39.8	43.6	90.5	51.3	53.1	83.0	59.8	60.2

Bold denotes the best result in each column. OKD uses CutMix+Mixup as the OOD data generator.

Table 3: Domain generalization results on PACS and OfficeHome using MobileNetV3-Small.

	PACS					OfficeHome				
	A	C	P	S	*Avg*	A	C	P	R	*Avg*
ERM	63.3	73.4	86.0	66.5	72.3	42.4	43.5	66.1	63.7	53.9
RSC	63.3	72.7	85.7	64.4	71.5	41.8	42.8	65.2	63.0	53.2
MixStyle	64.6	73.2	85.7	68.6	73.0	42.9	46.4	64.7	61.7	53.9
EFDMix	67.3	73.7	85.2	72.6	74.7	41.5	47.4	64.1	62.1	53.8
KD	64.4	73.8	82.9	73.2	73.6	43.6	46.7	67.9	66.6	56.2
OKD	**70.3**	**76.4**	**87.0**	**73.6**	**76.8**	**49.1**	**48.5**	**71.5**	**70.1**	**59.8**
KD's teacher	77.4	79.9	92.9	78.1	82.1	57.7	53.3	73.9	74.7	64.9

Bold denotes the best result in each column. OKD uses CutMix+Mixup as the OOD data generator.

4.3 Main Results on PACS and OfficeHome

Besides the new DOSCO-2k benchmark, we also conduct experiments on two commonly-used DG datasets, i.e., PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017), using the leave-one-domain-out evaluation protocol. The results are reported in Table 3, where the findings are similar to those on DOSCO-2k.

4.4 Analyses

Image Transformation vs. External Data Sources We have mentioned earlier that a key to making OKD work is to not produce OOD samples that are completely disjoint from the data distribution. To validate this design choice, we conduct large-scale experiments by training the OKD model in the following way: for each dataset on DOSCO-2k, we use a different dataset as the OOD data source, and repeat such an experiment for all available OOD data sources. Figure 5 shows the results, which confirm our assumption that the synthetic OOD samples for OKD should fall within the vicinity distribution of the training samples (Zhang et al., 2017). We further verify this assumption on the Jigsaw version of OKD by varying the number of patches to shuffle (see Table 4): the more patches we shuffle, the farther away the synthetic data is from the training data distribution (and hence worse performance).

Table 4: Ablation study of OKD w/ Jigsaw.

	$k=4$	$k=16$	$k=64$
Accuracy	48.3	46.9	40.5

k denotes the total number of patches to shuffle.

Results of Other Tiny Architectures We further study DG for two other tiny neural networks, i.e., MobileNetV2-Tiny and MCUNet, which are specifically designed for MCUs (Lin et al., 2020). As shown in Table 1, these two architectures are half the size of MobileNetV3-Small, meaning that their capacity is further shrunk down, and as a consequence, improving their DG performance

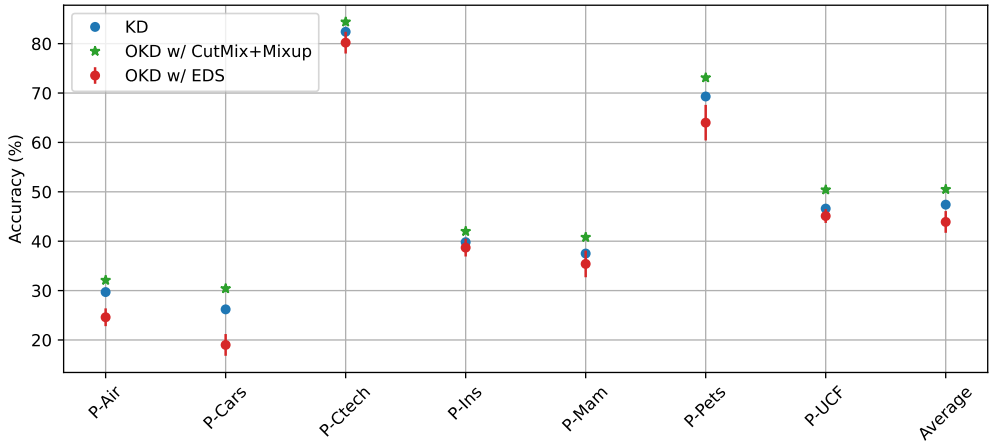
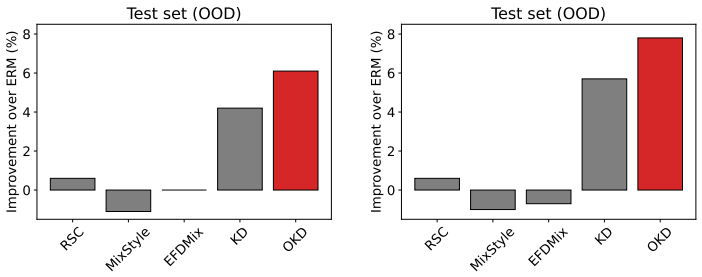


Figure 5: Image transformation vs. external data sources (EDS). The vertical bars of OKD w/ EDS correspond to standard deviations. *Better viewed by zoom-in.*



(a) MobileNetV2-Tiny. (b) MCUNet.

Figure 6: Domain generalization results of (a) MobileNetV2-Tiny and (b) MCUNet.

would be much more challenging. We repeat the same experiments using these two architectures on DOSCO-2k. The improvements of the DG and KD-based methods over ERM are shown in Figure 6. Interestingly, the two feature-based data augmentation methods, which could beat ERM and RSC when MobileNetV3-Small is used, are no longer competitive in this challenging setting. RSC, on the other hand, is able to gain some improvements over ERM, but the improvements are only marginal (less than 1%) and much smaller than those obtained with MobileNetV3-Small (cf. Figure 1a). Our approach OKD still maintains its dominance and achieves non-trivial improvements over the strong KD model for both architectures—this again strongly justifies the design of OKD and indicates its potential.

5 Related Work

We briefly review two areas closely related to our research, namely domain generalization (DG) and knowledge distillation (KD). See Zhou et al. (2022a); Wang & Yoon (2021) for more comprehensive surveys in these two areas.

Domain Generalization The majority of DG methods can be grouped into the following three categories: (i) domain alignment, (ii) meta-learning, and (iii) data augmentation. Domain alignment methods often employ a distance measure like Maximum Mean Discrepancy (Li et al., 2018b) or adversarial learning (Zhang et al., 2022a) to reduce the feature distributions between two or multiple source domains. Meta-learning methods, on the other hand, adopt the notion of learning-to-learn and typically perform model learning using pseudo-source and pseudo-target data, which simulates domain shift (Li et al., 2018a; Balaji et al., 2018; Dou et al., 2019; Shi et al., 2022). Data augmen-

tation methods aim to diversify the training data, which is often achieved by learning a generative model (Zhou et al., 2020b;a) or mixing data at the input (Xu et al., 2020b; Yao et al., 2022) or feature-level (Zhou et al., 2021; Zhang et al., 2022b). More recent research has explored test-time adaptation (Wang et al., 2020; Zhang et al., 2021; Iwasawa & Matsuo, 2021), which updates the model on-the-fly using a test datapoint or minibatch, and multimodal learning, such as learning a joint embedding space for image and language (Min et al., 2022).

Our research significantly differs from existing ones in that we focus on mobile DG applications, and for the first time study DG for tiny neural networks. It is worth mentioning that existing DG methods are mainly developed using large models, so it was unclear whether they can be applied to tiny models, which have a much smaller capacity and hence are more difficult to train.

Knowledge Distillation KD is a popular technique used in model compression (Bucilua et al., 2006) as well as other areas like semi-supervised learning (Chen et al., 2020). The most basic form of KD is to minimize the KL divergence between the student and teacher's outputs (Hinton et al., 2015), as reviewed in Section 2.1. Several follow-ups extended KD by re-designing representations (to be transferred) based on, for example, hints (Romero et al., 2014), probabilistic distributions (Passalis & Tefas, 2018), attention maps (Zagoruyko & Komodakis, 2017), and mutual information (Ahn et al., 2019). Some variants sought other useful knowledge sources like inter-instance correlations (Tian et al., 2020; Park et al., 2019; Tung & Mori, 2019) or self-supervision (Xu et al., 2020a).

More related to our work are data-free KD methods (Fang et al., 2021; Binici et al., 2022), which typically use generative modeling to synthesize training data. Differently, our work addresses on-device DG, a new problem that unifies DG and efficient model deployment. Our results unveil an interesting find that has not been brought up in the KD literature: the teacher-student gap on out-of-distribution data is larger than that on in-distribution data, which leads to concerns over mobile DG applications and is thus worth further investigating.

6 Conclusion, Limitation and Future Work

The paper presents a novel study on how to improve DG for tiny neural networks, which have been overlooked by existing research. We show that current state-of-the-art DG methods do not work consistently well for different tiny network architectures, e.g., MixStyle and EFDMix can improve upon ERM when using MobileNetV3-Small, but their performance plunges below ERM's when it comes to two tinier architectures specifically designed for MCUs, i.e., MobileNetV2-Tiny and MCUNet. Overall, the results suggest that tiny neural networks, which have small capacity and low complexity, should be trained differently than their large counterparts. Our OKD framework, despite having an extremely simple design that leverages well-known data augmentation methods and not adding any additional parameter to the model, demonstrates great potential for solving on-device DG. We believe our approach can serve as a strong baseline to build upon for future work.

Although OKD's improvements are significant, the performance gap with large models is still huge—about 10% differences compared with ResNet50 on DOSCO-2k. Therefore, on-device DG is yet to be solved and many interesting questions can be explored in the future: Can we design more advanced OOD data generators or even make them fully learnable? Is it possible to combine KD with DG methods while pursuing efficacy? Will state-of-the-art generative models such as GAN or diffusion models help? How about making model learning fully on-device, i.e., allowing model updates to be performed on tiny devices? Or can we try some parameter-efficient designs that can offer a good trade-off between performance and efficiency? Just to name a few.

Broader Impact

The scaling of neural networks-based AI models often brings significant improvements in performance but meanwhile can cause huge costs, both economically and environmentally. Our research can help alleviate the above issue by making tiny AI models more generalizable, and hence more comparable to their large counterparts so tiny models can be adopted more often in practice—to achieve *Green AI*. Furthermore, our research not only opens new challenges to the domain generalization community but also provides new insight to the area of knowledge distillation, and more broadly, model compression.

REFERENCES

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Kuluhan Binici, Nam Trung Pham, Tulika Mitra, and Karianto Leman. Preventing catastrophic forgetting and distribution mismatch in knowledge distillation via synthetic data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 663–671, 2022.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.

Han Cai, Ji Lin, Yujun Lin, Zhijian Liu, Haotian Tang, Hanrui Wang, Ligeng Zhu, and Song Han. Enable deep learning on mobile devices: Methods, systems, and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 27(3):1–50, 2022.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019.

Gongfan Fang, Yifan Bao, Jie Song, Xinchao Wang, Donglin Xie, Chengchao Shen, and Mingli Song. Mosaicking to distill: Knowledge distillation from out-of-domain data. *Advances in Neural Information Processing Systems*, 34:11920–11932, 2021.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*, pp. 124–140. Springer, 2020.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.

Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. Mcunetv2: Memory-efficient patch-based inference for tiny deep learning. arxiv 2021. In *Advances in Neural Information Processing Systems*, volume 34, pp. 2346–2358, 2021.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Seonwoo Min, Nokyung Park, Siwon Kim, Seunghyun Park, and Jinkyu Kim. Grounding visual representations with texts for domain generalization. *arXiv preprint arXiv:2207.10285*, 2022.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018.

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

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations*, 2022.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1365–1374, 2019.

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

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2022.

Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pp. 588–604. Springer, 2020a.

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6502–6509, 2020b.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7277–7286, 2022a.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021.

Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8035–8045, 2022b.

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. *arXiv preprint arXiv:2207.07106*, 2022c.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 13025–13032, 2020a.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pp. 561–578. Springer, 2020b.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, pp. 1–12, 2022b.

A Appendix

A.1 Example Images of the DOSCO Benchmark

Here we show example images of each dataset from the DOSCO benchmark:

- Figure 7: P-Air
- Figure 8: P-Cars
- Figure 9: P-Ctech
- Figure 10: P-Ins
- Figure 11: P-Mam
- Figure 12: P-Pets
- Figure 13: P-UCF

It is clear that different groups of images contain distinct visual contexts associated with background, object pose, image style, viewpoint, etc.

(a) 707-320. (b) F-16A_B. (c) Model_B200.

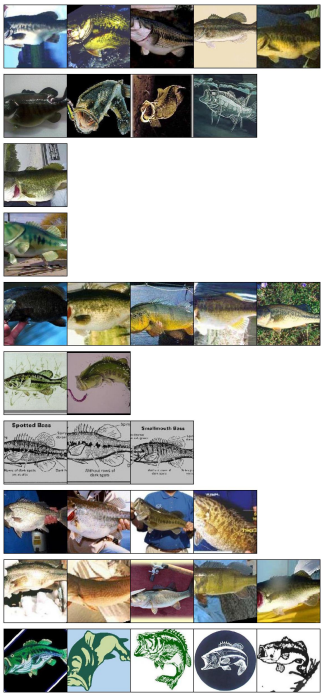
(d) BAE_146-200. (e) Cessna_172. (f) Fokker_50.

Figure 7: Example images from P-Air. Each row contains images with the same domain label (some domains have less than five images).

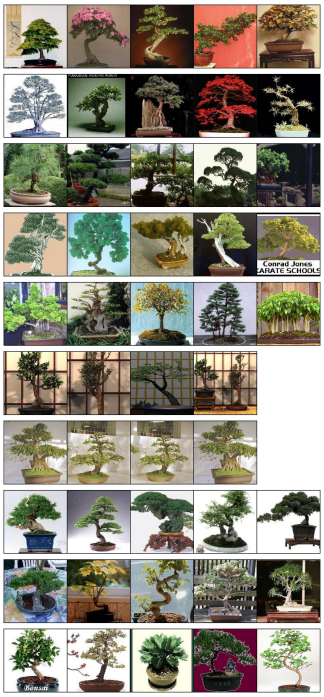
(a) 1999 Plymouth Neon Coupe. (b) 2009 Bentley Arnage Sedan. (c) 2012 Acura TSX Sedan.

(d) 2012 Mitsubishi Lancer Sedan. (e) 2012 Nissan NV Passenger Van. (f) 2012 Volvo C30 Hatchback.

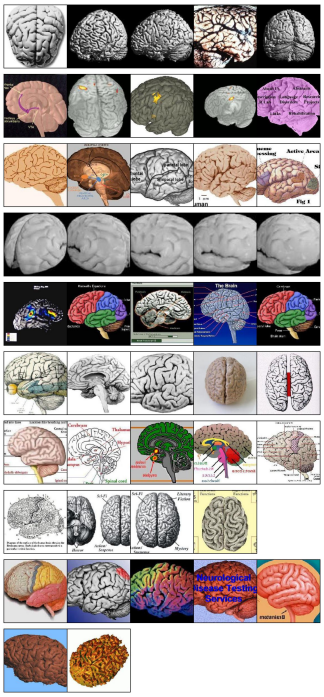
Figure 8: Example images from P-Cars. Each row contains images with the same domain label (some domains have less than five images).



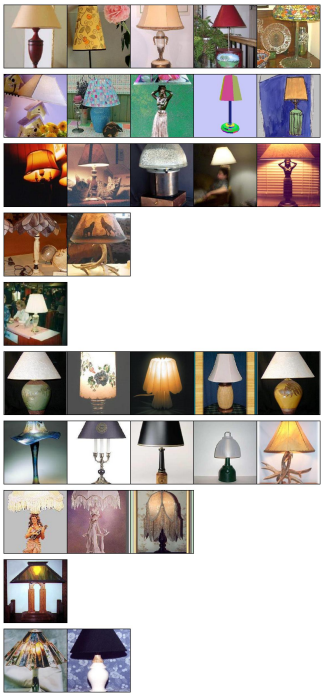
(a) Bass.



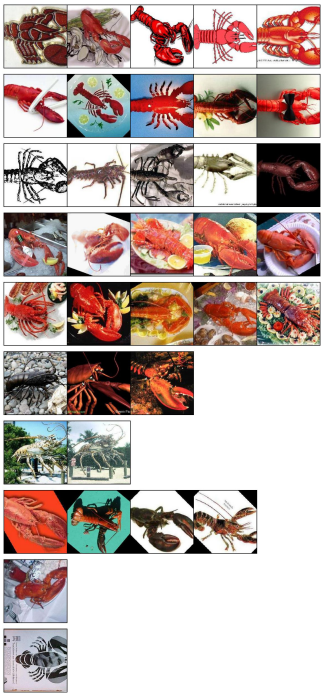
(b) Bonsai.



(c) Brain.



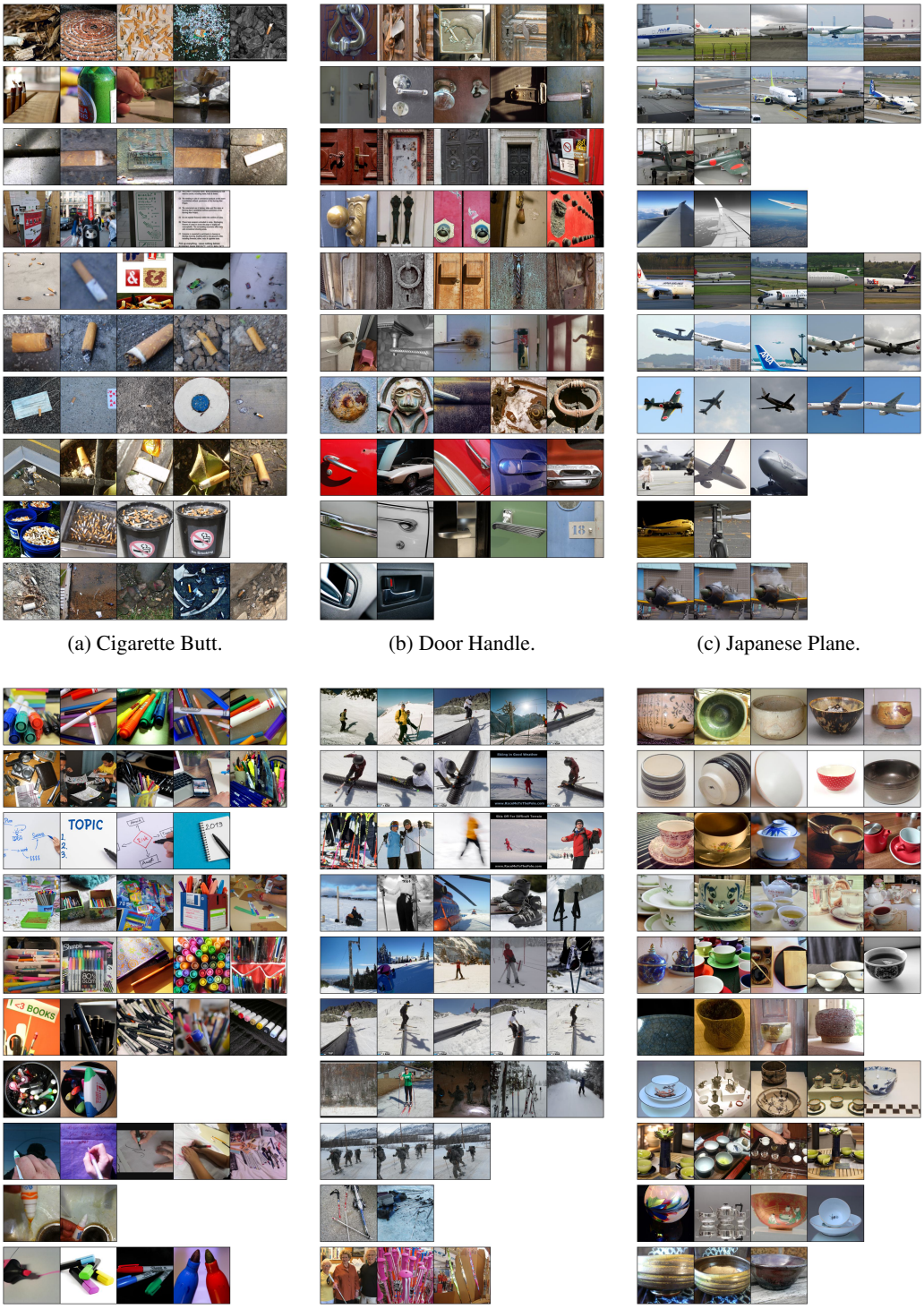
(d) Lamp.



(e) Lobster.

(f) Okapi.

Figure 9: Example images from P-Ctech. Each row contains images with the same domain label (some domains have less than five images).

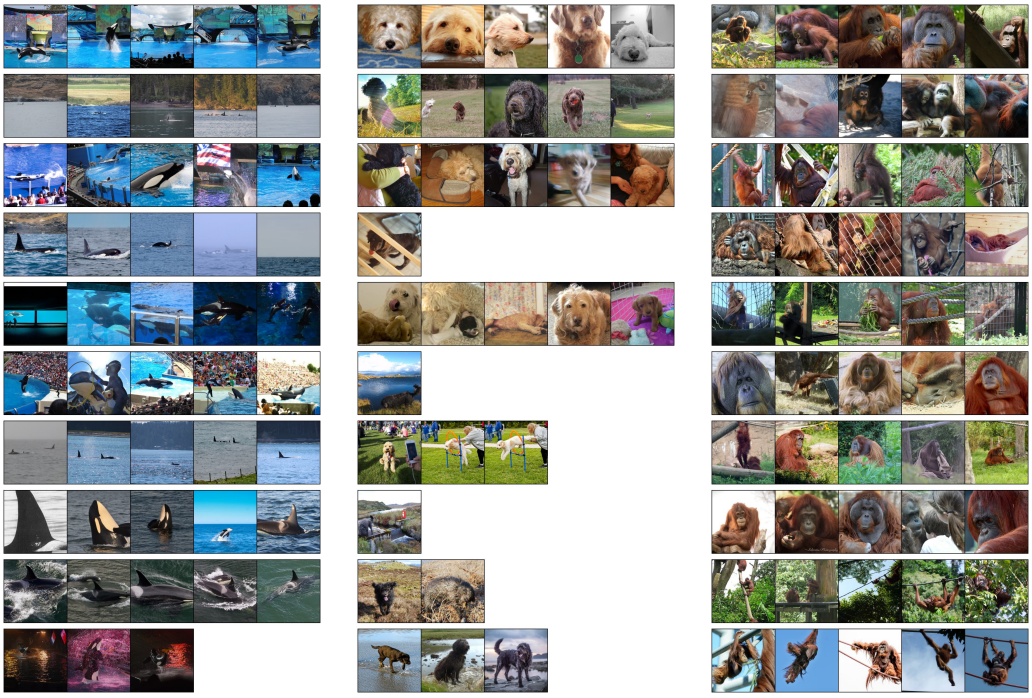


(a) Cigarette Butt. (b) Door Handle. (c) Japanese Plane.

(d) Marker Pen. (e) Ski Pole. (f) Tea Bowl.

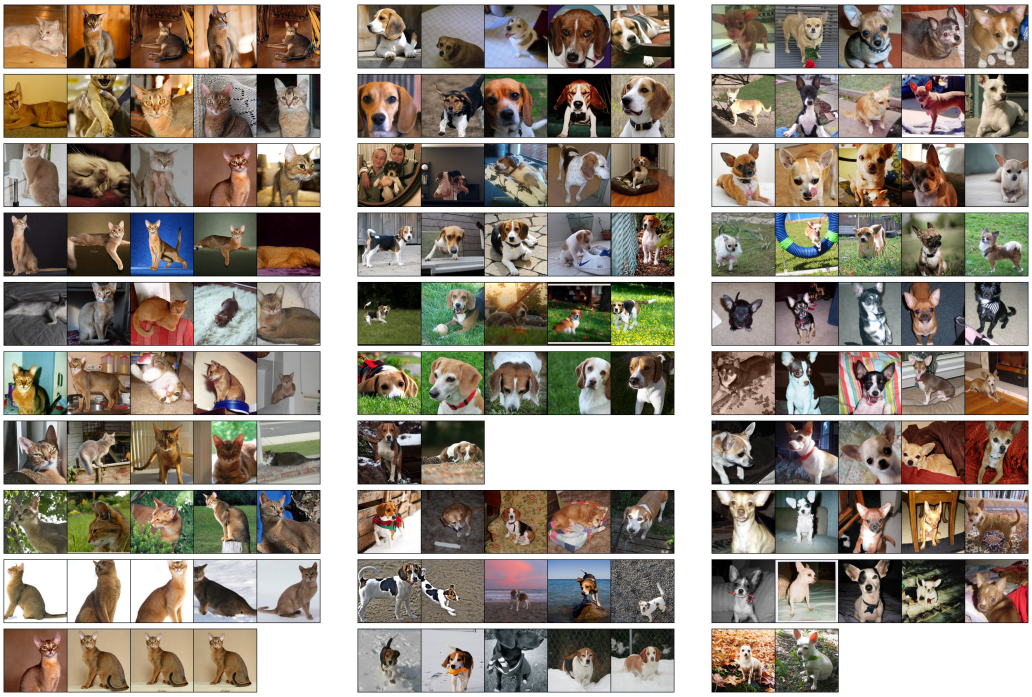
Figure 10: Example images from P-Ins. Each row contains images with the same domain label (some domains have less than five images).

(a) Border Terrier. (b) European Hare. (c) Japanese Cat.



(d) Killer Whale. (e) Labradoodle. (f) Orangutan.

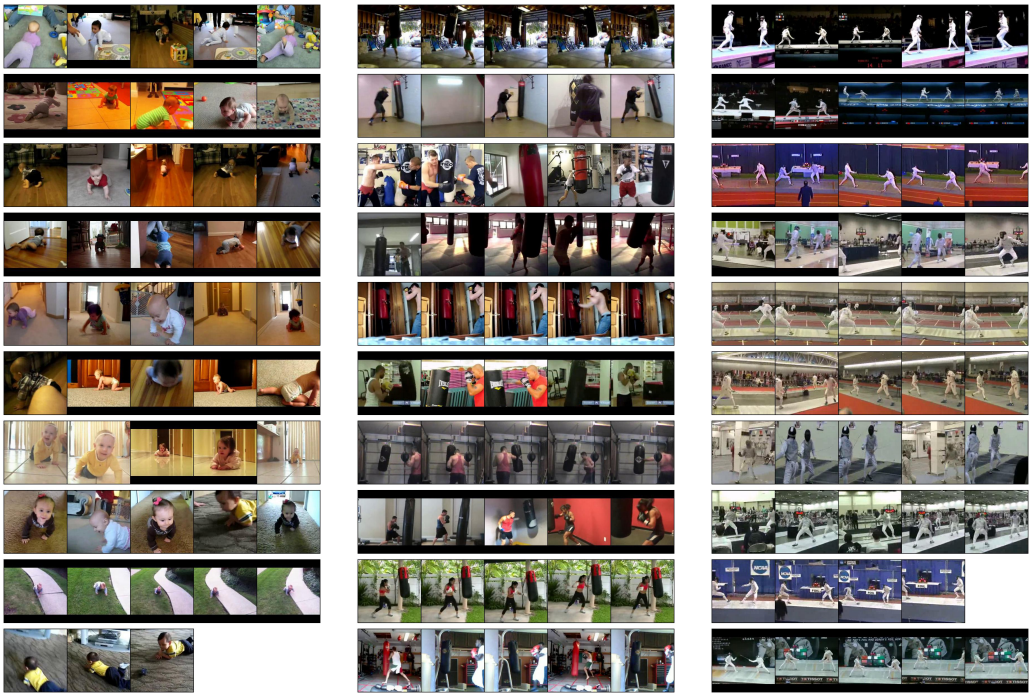
Figure 11: Example images from P-Mam. Each row contains images with the same domain label (some domains have less than five images).



(a) Abyssinian. (b) Beagle. (c) Chihuahua.

(d) Miniature Pinscher. (e) Pomeranian. (f) Scottish Terrier.

Figure 12: Example images from P-Pets. Each row contains images with the same domain label (some domains have less than five images).



(a) Baby Crawling. (b) Boxing Punching Bag. (c) Fencing.

(d) Floor Gymnastics. (e) Jump Rope. (f) Playing Cello.

Figure 13: Example images from P-UCF. Each row contains images with the same domain label (some domains have less than five images).