SUPERMARIODOMAINS: GENERALIZING TO DOMAINS WITH EVOLVING GRAPHICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Domains in previous Domain Generalization (DG) benchmarks have been sampled from various image collections of different styles such as photographs, sketches, cartoons, paintings, product images, and etc. However, from these existing DG datasets, it is still difficult to quantify the magnitude of domain shift between different domains and relate that to the performance gap across domains. It is also unclear how to measure the overlap between different domains. Therefore, we present a new DG dataset, SuperMarioDomains, containing four domains that are derived from four chronological titles in the Mario video game franchise on four generations of video game hardware. The discrepancy between our domains is quantified in terms of image representation complexity that reflect the hardware evolution in image resolution, color palette, and presence of 3D rendering. We benchmark state-of-the-art DG algorithms under both Multi-Source and Single-Source DG settings on our dataset and find that they can only surpass the random average baseline in our dataset by at most 18.0% and 10.4% respectively. In addition, we show that adding our dataset as part of the pre-training process improves performance of existing DG algorithms on the PACS benchmark.

1 INTRODUCTION

Domain Generalization (DG) is a crucial task in deep learning that remains challenging despite the recent fast development and the mass effort poured into the field. Fundamentally, a robust DG approach must be capable of both capturing invariant representations from the known training environments and adapting to the unknown test environments. Ideally, having a strong DG model can bring huge improvement to practical applications. For example, a self-driving software would be able to seamlessly transfer what it learns in computer generated simulation scenarios onto real life situations with little risk of causing traffic accidents.

Researchers have long identified one of the biggest challenges in the DG task is the problem of domain shift, handling the representation gaps between the known training domains and the unknown test domains (Pan & Yang, 2009). To tackle this problem, many have created specific image DG datasets and benchmarks, from those as early as VLCS (Khosla et al., 2012) and RotatedMNIST Ghifary et al. (2015), to the more recent and larger-sized PACS (Li et al., 2017), Office-Home (Venkateswara et al., 2017), SVIRO (Cruz et al., 2020), and DomainNet (Peng et al., 2019). The domains featured in these DG datasets often collect samples from pre-existing image datasets categorized by image style, such as photographs, cartoons, paintings, hand sketches, infographs, and so on. These DG datasets also feature distinguishable image classes that span over their domains of choice, such as dogs and cats in forms of photograph, cartoon, painting, and such. Thanks to such increased availability of resources, state-of-the-art DG algorithms have been discovered and developed based on these datasets over the years, including ERM (Vapnik, 1991), DANN (Ganin et al., 2016), CDANN (Li et al., 2018b), SagNet (Nam et al., 2021), RandConv (Xu et al., 2020), and SWAD (Cha et al., 2021).

Despite the fast evolution of methods, there is still a lack of quantified understanding of the magnitude of domain shift. For human beings, after observing realistic photos of dogs, one takes little effort to identify dogs in oil paintings, while they would struggle tell dogs apart in highly abstracted hand sketches. However, we do not have a quantitative measure to implicate whether the domain shift gap from photos to sketches is many times larger than the gap from photos to paintings. We

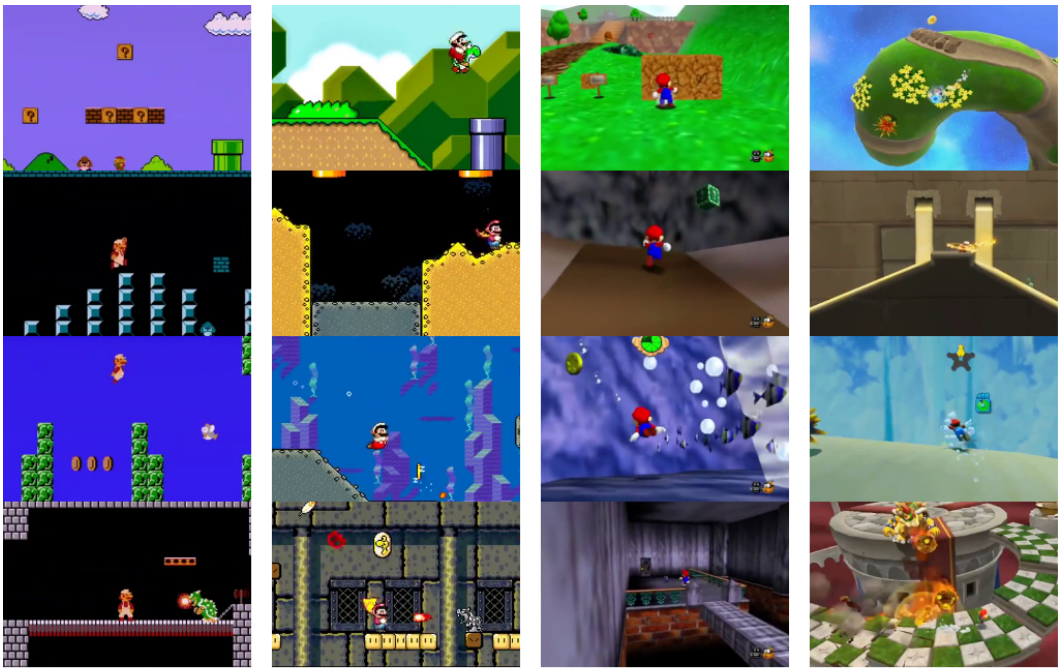

Figure 1: An overview of our dataset. Each column from left to right features actual gameplay footage from a certain Mario game released on a specific Nintendo game console in chronological order. We name our domains after these four consoles. On the other hand, each row shows how a certain class of in-game scenes is rendered in increasingly sophisticated graphics thanks to the hardware improvement over time. The scene classes, from top to bottom, are Overworld, Underground, Aquatic, and Castle, which we discuss in detail in Chapter 3.1.

also do not know a quantitative scale of the similarity between the domains, as naturally we need to learn to capture the common features so as to develop domain-invariant representations.

In this paper, we create and compile a new multi-domain DG dataset for classification benchmarks, dubbed as SuperMarioDomains, inspired by the evolving technology of video game graphics. We name our domains after the game consoles we sample from - NES, SNES, N64, and Wii. The domains in our dataset are dissimilar to each other following a chronological order, indicating the enhancement of video game graphics rendered on increasingly capable hardware. The first domain in order, NES, consists of gameplay images rendered in the lowest resolution, in 8-bit color palette, and in 2D only. In the following domains, we gradually add up the the resolution, the color complexity, and the presence of 3D graphics. As for the image contents, each domain features a game from the Mario franchise, which employs a consistent graphic level design over the generations. We can thus classify the gameplay images in the Mario games using a consistent set of labels that describe the scenes.

We then quantitatively explore the image style domain shift by introducing the JPEG compression rate metric onto our dataset, as well as the two most commonly used PACS and VLCS datasets. With this metric, we show that the domains of the same style remain nearly identical to each other. We also discover that a large domain style deviation exists in PACS's domains that adds to its diversity, while the domains in our dataset maintain a smaller domain style discrepancy in between.

For baseline evaluation, we showcase the performances from current state-of-the-ark DG methods on our dataset. The experiments show that our dataset, although with a built-in similarity in between domains by design, poses a decent challenge to existing approaches both in Multi-Source DG and Single-Source DG settings, where the best performances top at around 40%. We also find that by adding our dataset to the pretraining process, we can further improve the overall DG performance

on PACS induced by its large domain diversity without changing the method. We hope our work can serve as a good test bed to encourage better approaches for improving domain generalization.

2 Related Works

Image Domain Generalization Benchmarks. Early DG benchmarks such as Office (Saenko et al., 2010) or VLCS (Khosla et al., 2012) focus solely on photorealistic images. Since the single-style bias is first exposed by DeCAF (Donahue et al., 2014), more image styles have been introduced to the mix of image domains, and we have seen a steady increase in scale for DG datasets. Office-Home (Venkateswara et al., 2017) and PACS (Li et al., 2017) first introduce clipart, painting, and hand sketches, while both maintaining a modest size of around 10K samples and 4 categories. We also see the introductions of much specialized domains such as digital single lens reflex (DSLR) camera configurations by TerraIncognita (Beery et al., 2018), various car driving simulation environments by SVIRO (Cruz et al., 2020), or medical imaging of different tumor types by WILDS (Koh et al., 2021) and Camelyon17 (Bandi et al., 2018). As of now, the DomainNet dataset (Peng et al., 2019) tops at incorporating 569K images, featuring 6 image style domains and 345 categories.

Domain Generalization Approaches. Techniques to tackle the problem of domain shift in Domain Generalization have been rapidly developed over the years. Researchers are no longer restricted to straightforward approaches such as finding linear alignment (Hoffman et al., 2013) or non-linear alignment (Duan et al., 2012) in between domains. Current popular approaches on Domain Generalization can involve in adaptation and combination of deep neural network models by Ganin et al. (2016) and Li et al. (2018b), leveraging Meta-Learning (Li et al., 2018a) or Adversarial-Learning (Volpi et al., 2018; Qiao et al., 2020; Gokhale et al., 2023) to transfer models parameters, finding causality information in between domains, or using regularization methods to achieve optimized generalizability (Cha et al., 2021). Still, in many occasions, simple methods such as ERM (Vapnik, 1991) can yield high performance in popular Domain Generalization benchmarks Gulrajani & Lopez-Paz (2021).

3 The Mario Dataset

3.1 Design Choices

We build our SuperMarioDomains dataset (hence referred to as **the Mario dataset**) featuring domains where synthesized scenes are rendered in an evolving manner. Unlike previous Domain Generalization datasets, we incorporate images starting from pixelated mosaic 2D graphics, then gradually transition towards high polygon 3D graphics via multiple stages. Our domains of choice are video game footage captured on 4 chronological generations of video game consoles, all manufactured by Nintendo: The Nintendo Entertainment System (NES) released in 1985, the Super Nintendo Entertainment System (SNES) in 1990, the Nintendo 64 (N64) in 1996, and the Wii in 2006. Game scenes performed on the older generations of consoles demonstrate more limited rendering capabilities at its time, while the newer games enjoy stronger graphics hardware and better ability to mimic complex real-life scenes.

Table 1 lists down the hardware specifications of the game consoles as our domains of choice. In general, the later hardware feature faster processing speed, dedicated GPUs, larger memory, higher output resolution, and more vibrant colors. Previous datasets for image Domain Generalization ignore such transitional graphic quality improvements, oftentimes addressing solely the overall domain shift such as synthetic versus photorealistic. Our dataset features a much fine-grained visual domain shift design that challenges the robustness of current high performing DG approaches.

The sources of data. For the actual image contents, we choose gameplay footages entirely from the Mario franchise, one title per our featured game consoles. The reason we decide to use the Mario games is that the Mario franchise employs similar level and scenery design shared across the generations, so that we may have consistent scene labels throughout the evolution of visual quality. Respectively, our dataset incorporates gameplay footage from Super Mario Bros on NES, Super Mario World on SNES, Mario 64 on N64, and Super Mario Galaxy on Wii - the older two games rendered in 2D graphics while the latter two in 3D.

Table 1: The graphic hardware specifications of the consoles we feature as the domains in our dataset. These figures are of the products released in North America.

Console	Release Year	CPU Freq.	GPU Freq.	Video RAM	Max Resolution	Color Palette	3D Rendering
NES	1985	1.79 MHz	-	2 KB	256x240	8-bit	No
SNES	1990	3.58 MHz	-	64 KB	512x478	16-bit	No
N64	1996	93.75 MHz	62.5 MHz	8 MB	704x480	24-bit	Yes
Wii	2006	729 MHz	243 MHz	88 MB	854x480	32-bit	Yes

The annotation procedure. We sample our images from multiple full walkthrough gameplay videos uploaded to YouTube and Twitch.tv, using video recordings that cover all the levels in games from start to end, so that we can cover as many diverse in-game scenes as possible. We then manually label video segments from the chosen gameplay footages using timestamps, so that all the video frames within one segment share one scene label. During labeling, we skip the segments that do not depict any level designs, such as those when narrative stories are played, the scoreboard is displayed, or the whole screen is blackened out for level transition. Since earlier games on NES or SNES in general take shorter time complete, we sample the frames from the NES and the SNES video games at a higher 10 fps, while the NES and the Wii videos at 3 fps. We also crop out the on-screen text, the user interface, and the level floor regions, so that our samples in the dataset introduce as few biases from scene-specific artifacts as possible. Example scenes from our domains are shown in Figure 2.

The scene classes. We feature 4 types of distinct but universal Mario scenes inspired by entries on the open-source Mario Wiki:

- **Overworld** scenes cover a wide range of outdoors levels with an open bright background and natural green vegetation, such as grassland or forest.
- **Underground** scenes feature closed-off dark interior background with sporadic weak lighting and mountain cave-like or sewage textures.
- **Aquatic** scenes appear with watery textures surrounded by aquatic creatures such as fish or squid, and oftentimes with swimming movements by the player controlled character.
- **Castle** or 'Boss Fight' scenes are stages within dungeon-like environments, normally indoors and/or with an architectural texture and man-made obstacles such as traps and spikes.

The samples in our dataset labeled in the above 4 classes are separable in-domain. As is shown in Table 2, when learning only the in-domain samples using basic image classification models, the trainings easily converge with regard to each individual domain and the test accuracies all reach above 90%.

Table 2: The in-domain test accuracy using basic classification models from scratch.

Model	NES	SNES	N64	Wii
Resnet18	99.1	96.3	97.9	98.9
Resnet50	99.0	96.2	98.3	98.6

3.2 STATISTICS

Table 3: The 4 classes of labeled scenes back in the original gameplay videos are distributed disproportionately. So we further down-sample them evenly and make sure each domain shares the same number of the samples per label.

Domain	Overworld	Underground	Aquatic	Castle	Raw Footage Length	Dataset Size
NES	10K	3.4K	5K	3.6K	33 min	7K
SNES	17K	8K	4K	6K	2 hr 38 min	20K
N64	30K	7K	6K	7K	4 hr 55 min	25K
Wii	61K	19K	23K	18K	9 hr 25 min	30K

Our dataset consists of debiased gameplay video frames of 4 domains and 4 scene classes. Within each domain, each class has the same number of samples. The source video length for each domain increases by generation: the most aged NES domain has the fewest possible unique frames from less than 1 hour of gameplay, while the latest Wii domain are down-sampled from 10 hours of recordings.

Since we label an entire video segment with one label during annotation, we further select evenly spaced frames for every video segment, so as to ensure all variable scenes even under the same scene class are considered. After further manual inspection by removing any mislabeled, duplicate, or pure-color images, we eventually yield 7K samples for NES, 20K for SNES, 25K for N64, and 30K for Wii as are shown in Table 3.

4 Quantitative Analyses of Domain Style Shifts

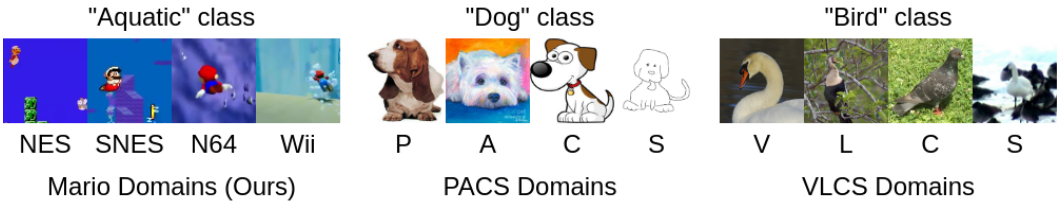

Figure 2: A qualitative overview of image domain shift under a certain class in multiple DG datasets. Our Mario domains are stratified by the evolving hardware capabilities of multiple game consoles referred in Table 1 over time. In PACS, the domains are differed by image styles: **P**hoto, **A**rt painting, **C**artoon, and **S**ketch. In VLCS, all images are real life photographs and the domains are simply the names of its 4 source collections: **V**OC2007 (Everingham et al., 2010), **L**abelMe (Russell et al., 2008), **C**altech101 (Fei-Fei et al., 2004), and **S**UN09 (Choi et al., 2010).

The purpose of Domain Generalization task is to encourage better methods at learning about domain invariant representations via the designated DG datasets. In theory, the domains need to be similar enough that common features can be learned. However, previous DG datasets adopted as benchmarks on this task often resort to measuring the domain shift by feature space distances using metrics such as KL-Divergence or t-SNE (Li et al., 2017). Such metrics do not provide quantified magnitude of domain shift in between the domains and does not show potential domain overlaps. Instead, it only shows that the domains are separable somehow, reflected in the performance difference across the domains.

The domains in our dataset, on the other hand, represent the chronological improvement of hardware rendering capabilities spanning from 1980s to 2000s. As is specified in Table 1, early video game consoles such as NES and SNES are only capable of rendering 2D pixelated spirits that are composed of highly regular color trunks and shapes, whereas the later generations of consoles obtain the power to render more complex 3D graphics and show higher volumes of simultaneous polygons on screen and thus can generate more realistic-looking scenes.

The compression metric for domain style shift. Therefore, to better quantify the shift in domain style complexity in Domain Generalization datasets, we choose to calculate the averaged JPEG compression rates with regard to individual domains. First proposed by Wallace (1992), the lossy JPEG compression algorithm leverages Discrete Cosine Transform and quantization to reduce the file size of an image. We specifically leverage the characteristic of JPEG compression that, given the same compression parameters, the more diverse irregular patterns and more vibrant colors the input image has, the smaller output file size JPEG compression may yield. The compression rate is also controllable by a pre-set quality parameter Q ranging from 1 to 100. Compressing an image at a lower Q value or lower JPEG quality produces more artifact patterns in the final output and results in a smaller file size in return.

For the domain shift analyses, we set up our experiments on the domains in our Mario dataset, PACS, and VLCS in the following steps. We first unify the sizes of all involved images into 224 by 224 pixels and convert all of them into the PNG lossless format to retain their as-is color information. Then, from the lossless format, we use 3 different levels of lossy JPEG compression qualities - Best (Q=95), Medium (Q=50), and Low (Q=10) - to compress all images of all three datasets. We calculate the means and the standard deviations of the lossless-to-lossy file size rate per dataset domain. We also take the average of the means and standard deviations across all the domains in each dataset at a certain compression quality setting.

Table 4: The means and the standard deviations of JPEG compression rates per dataset domain in 3 different qualities. For example, a compression rate of 5 means the size of the output JPEG image is one fifth of the original image. PACS is shown to possess the highest deviations in the compression metric largely thanks to its Sketch domain.

Mario (Ours)	NES	SNES	N64	Wii	Average
JPEG Best	2.54±0.21	3.08 ±0.45	3.81 ±0.24	3.52 ±0.33	3.24 ±0.55
JPEG Medium	6.99 ±1.11	9.41 ±1.84	12.04 ±1.04	11.65 ±1.62	10.02 ±2.33
JPEG Low	12.96 ±3.21	20.50 ±5.77	25.72 ±2.99	25.10 ±4.12	21.07 ±5.89
PACS	Photo	Art painting	Cartoon	Sketch	Average
JPEG Best	3.49 ±0.24	3.43 ±0.31	3.09 ±0.34	1.36 ±0.15	2.84 ±**1.00**
JPEG Medium	12.18 ±1.26	11.48 ±1.17	9.23 ±1.72	2.93 ±0.40	8.96 ±**4.21**
JPEG Low	28.98 ±3.76	27.67 ±3.43	18.75 ±3.90	4.78 ±0.82	20.05 ±**11.15**
VLCS	VOC2007	LabelMe	Caltech101	SUN09	Average
JPEG Best	3.52 ±0.37	3.97 ±0.47	3.42 ±0.65	3.47 ±0.41	3.60 ±0.48
JPEG Medium	12.11 ±1.75	13.41 ±1.89	11.31±2.38	11.51±1.79	12.09 ±1.95
JPEG Low	29.68 ±4.14	32.33 ±3.39	26.87 ±5.88	27.87±3.85	29.19 ±4.32

Table 4 reveals several observations of domain style shifts out of the three datasets. Since all four domains in VLCS consist of real-life photos only, they share consistent compression rates with small deviations altogether with the Photo domain in PACS under the same JPEG compression quality. However, the Sketch domain in PACS deviates far from its other three peers in the dataset in terms of our compression rate metric; the gap starts at from 1.36 to 3.49 in the Best setting, then jumps all the way to a wider range from 4.78 to 28.98 in the Low setting. As is indicated in the qualitative overview Figure 2, the Sketch domain contains way less complex patterns and can be hardly compressed. This low rate of compression about the Sketch domain also contributes to PACS's overall highest compression rate deviations amongst the three datasets considered.

Table 4 also shows that our Mario domains provides a smoother domain shift with smaller gaps in between the domains. The image styles out of the Mario domains shift not as volatilely as those of PACS do, but still provide a higher diversity compared to the single-style VLCS. Most specifically, the least diverse NES domain in our dataset is still more vibrant than the Sketch domain in PACS. In the following experiment Chapter 5.2, based on our observation, we argue that it is possible that training on more diverse styles representing on the same concept (such as in PACS) is not the best approach for improving domain generalization. Our dataset thus is a good test bed for this concept.

5 Baseline Evaluation

In this baseline evaluation section, we show experiments that demonstrate the unique challenges within our new dataset as means to analyze the robustness of common Domain Generalization / Domain Adaptation methods on evolving video game image domains. We show the performances from widely used models and approaches under the Out-of-domain classification metrics, where the domains selected for training do not overlap with the unseen domains for testing. Specifically, we show that the current top-ranking methods can only generalize modestly across our evolving video game domains.

For the experiments mentioned in the section, we convert all images into grayscale to minimize the difference of color variations across the video game domains. We resize all images to 224 by 224 in pixels. We randomly partition the domains in our dataset by 9:1 for training and testing, while within the test set each domain has the same number of samples per class. We also apply random horizontal flipping to all training samples as data augmentation. All the deep learning methods are implemented using PyTorch 1.11 and Torchvision 0.12.0, as well as the DomainBed code framework by Gulrajani & Lopez-Paz (2020). The backbone models are using pretrained weights from ImageNet1K (Deng et al., 2009). The experiments are conducted using a single Nvidia Tesla V100 GPU cluster.

Table 5: Multi-Source Domain Generation on our Mario dataset. Each domain column e.g. NES, means this domain is the one target domain unknown to the training samples.

Algorithm	Backbone	NES	SNES	N64	Wii	Average
Random	-	25.0	25.0	25.0	25.0	25.0
ERM	Resnet18	50.8	41.2	28.6	29.5	37.5
	Resnet50	46.7	36.4	23.5	26.0	33.2
	AlexNet	46.7	44.7	26.2	29.9	36.9
	VGG11	43.1	37.4	25.0	25.2	32.7
	ViT	44.7	44.9	32.4	**30.1**	38.0
	RegNetY	44.3	48.4	**34.4**	25.2	38.1
RandConv	Resnet18	36.2	34.8	26.0	25.8	30.7
	Resnet50	40.5	34.0	22.2	25.0	30.4
SWAD	Resnet18	43.0	36.6	21.4	32.6	33.4
	Resnet50	43.8	40.7	21.5	28.3	33.6
SWAD+Dither	Resnet18	57.8	44.4	21.4	24.1	37.0
	Resnet50	**58.4**	**53.9**	23.9	27.9	**41.0**

Table 6: Single-Source Domain Generation on our Mario dataset. Domain-wise, we use N for NES, S for SNES, 64 for N64, and W for Wii. N→S denotes the source domain is NES and the target domain is SNES.

Algorithm	Backbone	N→S	N→64	N→W	S→N	S→64	S→W	64→N	64→S	64→W	W→N	W→S	W→64	Ave.
Random	-	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
ERM	Resnet18	51.5	36.1	32.6	44.6	40.9	30.4	35.8	29.4	26.5	25.6	30.5	34.2	34.8
	Resnet50	42.7	29.7	35.2	41.7	**44.2**	28.4	32.8	30.1	25.3	25.4	31.2	25.7	32.7
	AlexNet	56.2	32.6	**40.0**	34.5	33.5	**37.5**	34.4	30.7	**30.0**	29.3	33.5	34.3	35.5
	VGG11	57.0	39.8	39.0	42.8	44.2	29.0	34.1	32.5	24.9	21.5	30.8	29.7	35.4
	ViT	**67.6**	**46.4**	25.0	45.2	43.4	33.4	28.5	35.0	25.8	**29.4**	31.4	**35.1**	**37.2**
RandConv	Resnet18	33.2	25.0	25.9	25.0	25.5	30.3	24.6	26.5	21.9	25.3	30.0	25.0	26.5
	Resnet50	30.7	24.8	27.3	27.3	26.8	29.7	23.6	29.7	23.5	25.4	29.7	23.6	26.8
SWAD	Resnet18	39.9	28.6	25.8	49.9	28.5	29.6	28.3	25.2	26.1	19.1	35.8	30.4	30.6
	Resnet50	33.2	29.3	25.6	59.5	32.1	31.5	34.5	30.3	27.1	18.7	34.6	21.1	31.5
SWAD+Dither	Resnet18	38.9	28.9	24.2	**64.0**	29.4	27.7	**54.4**	**47.4**	24.4	23.5	**38.9**	21.0	35.2
	Resnet50	41.7	32.6	25.1	60.4	28.1	27.6	50.2	44.7	26.2	27.2	37.2	25.1	35.5

5.1 Out-of-domain Classification

We evaluate existing Domain Generalization approaches on our Mario domains in out-of-domain classification accuracy. We set up the experiments in two major configurations - **Multi-Source Domain Generalization (MSDG)** where we train our model with all-but-one domains and test on the remaining one domain, and **Single-Source Domain Generalization (SSDG)** where we train our model using one domain and then test the accuracy with every other domains individually.

To guarantee a fair comparison, we follow the same domain generalization protocols by Gulrajani & Lopez-Paz (2020), and we report the highest performances per algorithm-backbone model combination out of its best hyper-parameter configuration. For each method, we apply sufficient training iterations to make sure the out-of-domain evaluation is performed on a converged model. We run each method 3 times with different random seeds and take the average accuracy per source-target domain setting.

As for the training details, we use the following configurations. We use 40 epochs for Resnet18 and Resnet50 backbones (He et al., 2016), while we use 20 epochs for all other backbones such as AlexNet (Krizhevsky et al., 2017) and VGG11 (Simonyan & Zisserman, 2014). We use a mini-batch size of 8 for large-scale backbone architectures Vision Transformer (ViT) (Dosovitskiy et al., 2020) and RegNetY (Xu et al., 2022), while for all other backbones we use 32. While under the ERM algorithm (Vapnik, 1991) we have more options to substitute various vision processing network as backbone, we use Resnet18 and Resnet50 since the more advanced RandConv (Xu et al., 2020) and SWAD (Cha et al., 2021) algorithms currently only support Resnet variants as backbone. For our experiments, the RandConv algorithm uses MultiAug, RC_{mix} with consistency loss, and λ=10. The SWAD algorithm uses a tolerence ratio of 0.3. We also add Floyd–Steinberg color dithering (Wellner, 1993) on top of SWAD as additional initial random noises, denoted as the SWAD+Dither

Table 7: Muti-Source Domain Generation on PACS with and without adding the Mario dataset samples to the pre-training process. Δ_{rel} denotes the relative improvement in the Average metric under the same algorithm-backbone setting.

Algorithm	Backbone	Pre-Training Set	Photo	Art	Cartoon	Sketch	Average	Δ_{rel}
(PACS baseline)	CNN	N/A	89.5	62.9	67.0	57.2	69.2	-
ERM	Resnet18	ImageNet1K	94.5	78.8	72.0	67.7	78.2	-
	Resnet50	ImageNet1K	95.4	83.5	79.8	76.7	83.8	-
RandConv	Resnet18	ImageNet1K	93.4	67.4	72.6	70.3	75.9	-
	Resnet50	ImageNet1K	92.8	65.6	75.7	73.1	76.8	-
SWAD	Resnet18	ImageNet1K	93.9	84.2	75.2	70.3	80.9	-
	Resnet50	ImageNet1K	96.0	86.4	78.9	76.5	84.5	-
ERM	Resnet18	ImageNet1K + Mario	93.4	77.1	72.6	70.3	78.3	+0.2
	Resnet50	ImageNet1K + Mario	96.3	82.3	80.1	**77.3**	84.0	+0.2
RandConv	Resnet18	ImageNet1K + Mario	89.7	75.1	75.3	73.0	78.3	+2.4
	Resnet50	ImageNet1K + Mario	92.0	75.9	77.0	73.1	79.5	+2.7
SWAD	Resnet18	ImageNet1K + Mario	93.5	81.2	76.9	72.8	81.0	+0.1
	Resnet50	ImageNet1K + Mario	**97.8**	**88.1**	**80.7**	76.9	**85.9**	+1.4

method. Universally, we deploy Adam (Kingma & Ba, 2014) as the optimizer and a learning rate of $5e^{-5}$.

Our results show that our dataset presents a challenge to current top performing DG approaches. In the MSDG setting in Table 5 where we use 3 source domains and 1 target domain, sophisticated backbones such as ViT or RegNetY can only surpass the random baseline by at most 13.1% in average. Also, generalizing towards the more complex 3D domains such as N64 and Wii is shown to be more difficult, where the best improvements are topped at 9.4% and 5.1% for N64 and Wii respectively. Algorithms such as RandConv or SWAD built with domain agnostic designs perform in average even lower than the basic ERM algorithm under the same backbone models. While color dithering provides a big boost on generalizing towards the 2D domains of simpler graphic styles, increasing up to 33.4% and 28.9% respectively for targeting NES and SNES, it often comes at a sacrifice of lowering the performances even below the random baseline when targeting 3D domains.

In the SSDG experiments, from Table 6, we likewise observe similar improvements over the random baseline, the best of all coming from ERM+ViT at in average +12.2%. Individually, the use of advanced backbones or advanced algorithm may achieve well above others in single categories, such as ERM+ViT's +42.6% at NES→SNES or SWAD+Dither+Resnet18's +39.0% at SNES→NES. Still, there is no dominant method that generalizes universally well in between any two domains in our dataset. We also notice that targeting the more complex 3D domains such as N64 and Wii in the SSDG setting is generally harder than targeting 2D domains.

5.2 Using Mario Domains as Additional Pre-Training Samples for PACS

In this section, we further show the merits of using our dataset to help improving DG performance on existing tasks. We demonstrate that by incorporating the samples from our dataset as part of training data alone, without changing the underlying algorithm or backbone model, the out-of-domain classification performances on PACS can be further improved. Under the same multi-source out-of-domain experiment settings, we conduct comparisons with a selected set of existing DG approaches, where per approach we only differentiate the involved pre-training data. The baselines only use ImageNet1K, while ours pre-train with ImageNet1K combined with our domains of different generations of video game graphics. We choose Resnet18 and Resnet50 as the backbones to study since they serve the best performances on PACS across multiple DG methods universally.

The results in Table 7 shows that using additional data from our Mario dataset to the pre-training part generically improves the Multi-Source Domain Generalization performance on PACS. Most noticeably, the Mario additive oftentimes contributes to larger improvement in generalizing over the Sketch domain, improving as far as 2.7% in the RandConv+Resnet18 method. The Sketch domain, as has been discussed in Chapter 4, has the largest domain shift in representation complexity in PACS.

Table 8: Ablation study of how the size of the Mario addition may contribute to the performance on PACS. The entries in bold text are the ones we report in the lower half of Table 7.

Algorithm	Backbone	Pre-Training Set	Photo	Art	Cartoon	Sketch	Average	Δ_{rel}
ERM	Resnet18	ImageNet1K + Mario (Full)	93.1	75.0	72.2	72.9	78.3	+0.1
	Resnet18	ImageNet1K + Mario (Fair)	**93.4**	**77.1**	**72.6**	**70.3**	**78.3**	**+0.2**
	Resnet50	ImageNet1K + Mario (Full)	**96.3**	**82.3**	**80.1**	**77.3**	**84.0**	**+0.2**
	Resnet50	ImageNet1K + Mario (Fair)	96.2	82.0	79.9	76.4	83.6	-0.2
RandConv	Resnet18	ImageNet1K + Mario (Full)	**89.7**	**75.1**	**75.3**	**73.0**	**78.3**	**+2.4**
	Resnet18	ImageNet1K + Mario (Fair)	87.7	76.4	73.8	73.0	77.7	+1.8
	Resnet50	ImageNet1K + Mario (Full)	91.1	77.2	75.7	73.1	79.3	+2.5
	Resnet50	ImageNet1K + Mario (Fair)	**92.0**	**75.9**	**77.0**	**73.1**	**79.5**	**+2.7**
SWAD	Resnet18	ImageNet1K + Mario (Full)	94.3	82.1	77.2	70.3	81.0	+0.1
	Resnet18	ImageNet1K + Mario (Fair)	**93.5**	**81.2**	**76.9**	**72.8**	**81.0**	**+0.1**
	Resnet50	ImageNet1K + Mario (Full)	97.7	88.4	80.8	76.1	85.7	+1.2
	Resnet50	ImageNet1K + Mario (Fair)	**97.8**	**88.1**	**80.7**	**76.9**	**85.9**	**+1.4**

As an ablation study, we also explore if the volume of the additional Mario samples to pretraining makes any difference to the performance on PACS. We test with two settings: **Mario (Full)** where we pre-train with the original unbalanced sizes of our Mario domains; **Mario (Fair)** where we evenly downsample the SNES, N64, and Wii domains to the same size of the NES domain of 7K, then we feed this downsampled balanced Mario dataset to the pre-training process. Table 8 shows that the size of the Mario addition makes only minor difference to the final results, up to a gap of 0.6% in the average Multi-Source performance.

6 Conclusion

We present the SuperMarioDomains dataset, a novel benchmark for Domain Generalization composed of multiple chronological domains of video game graphics. We construct our dataset by sampling gameplay video recordings, and make sure that the magnitude of domain shift in between our domains is in a much quantified manner compared to existing datasets. Through experiments in Multi-Source and Single-Source Domain Generalization, we show that state-of-the-art Domain Generalization approaches can only perform modestly on our dataset. We further demonstrate that by adding our dataset to pre-training, we may help Domain Generalization methods perform better on the existing PACS benchmark. We sincerely hope this work offers a new angle to invite more robust approaches in the Domain Generalization field.

References

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 3

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018. 3

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. volume 34, pp. 22405–22418. Curran Associates, Inc., 2021. 1, 3, 7

Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 129–136. IEEE, 2010. 5

Steve Dias Da Cruz, Oliver Wasenmuller, Hans-Peter Beise, Thomas Stifter, and Didier Stricker. Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 973–982, 2020. 1, 3

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. 6

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014. 3

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. 7

Lixin Duan, Ivor W Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012. 3

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010. 5

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. 5

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1, 3

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015. 1

Tejas Gokhale, Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, Chitta Baral, and Yezhou Yang. Improving diversity with adversarially learned transformations for domain generalization. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. 3

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 6, 7

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI. 3

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. 7

Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013. 3

Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pp. 158–171. Springer, 2012. 1, 3

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 7

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. 1, 3, 5

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. 3

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018b. 1, 3

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8690–8699, 2021. 1

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 1

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. 1, 3

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 12553–12562. IEEE, 2020. doi: 10.1109/CVPR42600.2020.01257. URL https://doi.org/10.1109/CVPR42600.2020.01257. 3

Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008. 5

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pp. 213–226. Springer, 2010. 3

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 1, 3, 7

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. 1, 3

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5339–5349, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/1d94108e907bb8311d8802b48fd54b4a-Abstract.html. 3

G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. doi: 10.1109/30.125072. 5

Pierre D Wellner. Adaptive thresholding for the digitaldesk. *Xerox, EPC1993-110*, pp. 1–19, 1993. 7

Jing Xu, Yu Pan, Xinglin Pan, Steven Hoi, Zhang Yi, and Zenglin Xu. Regnet: self-regulated network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 7

Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. *arXiv preprint arXiv:2007.13003*, 2020. 1, 7