Counterfactual Generative Networks

Axel Sauer[1,2] Andreas Geiger[1,2]
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen
{firstname.lastname}@tue.mpg.de

Abstract

In this work, we take a step towards more robust and interpretable image classifiers that explicitly expose the task's causal structure. We propose to decompose the image generation process into independent causal mechanisms that we train without direct supervision. By exploiting appropriate inductive biases, these mechanisms disentangle object shape, object texture, and background; hence, they allow for generating counterfactual images. We demonstrate the ability of our model to generate such images on MNIST and ImageNet. Further, we show that counterfactual images can improve out-of-distribution robustness with a marginal drop in performance on the original classification task, despite being synthetic. We open-source our models and code. [1]

1. Introduction

Despite the considerable successes of deep neural networks (DNNs), they still struggle in many situations. e.g., classifying images perturbed by an adversary or failing to recognize known objects in unfamiliar contexts. Many of these failures can be attributed to shortcut learning [3]. The DNN learns the simplest correlations and tends to ignore more complex ones. This characteristic becomes problematic when the simple correlation is spurious, i.e., not present during inference. Consider the setting of a DNN that is trained to recognize cows in images [1]. A real-world dataset will typically depict cows on green pastures in most images. The most straightforward correlation a classifier can learn to predict the label "cow" is hence the connection to a green, grass-textured background. Generally, this is not a problem during inference as long as the test data follows the same distribution. However, if we provide the classifier an image depicting a purple cow on the moon, the classifier should still confidently assign the label "cow." Thus, if we want to achieve robust generalization beyond the training data, we need to disentangle possibly spurious correlations from causal relationships.

[1]Code and full-length paper can be found at https://sites.google.com/view/counterfactual-generation/home

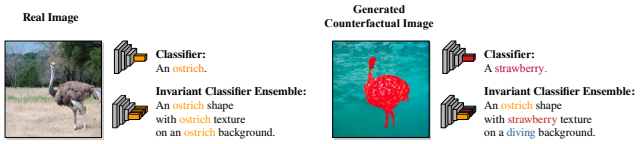

Figure 1: **Out-of-Domain (OOD) Classification.** A classifier focuses on all factors of variation (FoV). An FoV might be a spurious correlation, hence, impairing the classifier's performance on OOD data. An ensemble, e.g., a classifier with a common backbone and multiple heads, each head invariant to all but one FoV, increases OOD robustness.

Distinguishing between spurious and causal correlations is one of the core questions in causality research [8]. One central concept in causality is the assumption of *independent mechanisms* (IM), which states that a causal generative process is composed of autonomous modules that do not influence each other. In the context of image classification (e.g., on ImageNet), we can interpret the generation of an image as a causal process [6, 9]. We decompose this process into separate IMs, each controlling one factor of variation (FoV) of the image. Concretely, we consider three IMs: one generates the object's shape, the second generates the object's texture, and the third generates the background. With access to these IMs, we can produce *counterfactual images*, i.e., images of unseen combinations of FoVs. We can then train an ensemble of invariant classifiers on the generated counterfactual images, such that every classifier relies on only a single one of those factors. The main idea is illustrated in Figure 1. By exploiting concepts from causality, this paper links two previously distinct domains: disentangled generative models and robust classification. This allows us to scale our experiments beyond small toy datasets typically used in either domain.

2. SCMs for Image Generation

Our goals are two-fold: (i) We aim at generating counterfactual images with previously unseen combinations like a cat with elephant texture or the proverbial "bull in a china shop." (ii) We utilize these images to train a classifier invariant to chosen factors of variation.

Problem Setting Consider a dataset comprised of observations $\mathbf{x}$ (e.g. images), and corresponding labels y (e.g. classes). A common assumption is that each $\mathbf{x}$ can be described by lower-dimensional, semantically meaningful factors of variation $\mathbf{z}$ (e.g., color or shape of objects in the image). If we can *disentangle* these factors, we are able to control their influence on the classifier's decision. In the disentanglement literature, the factors are often assumed to be statistically independent, i.e., $\mathbf{z}$ is distributed according to $p(\mathbf{z}) = \Pi_{i=1}^{n}(z_i)$. However, assuming independence is problematic because certain factors might be correlated in the training data, or the combination of some factors may not exist. Consider colored MNIST [5], where both the digit's color and its shape correspond to the label. The simplest decision rule a classifier can learn is to count the number of pixels of a specific color value; no notion of the digit's shape is required. This kind of correlation is not limited to constructed datasets – classifiers trained on ImageNet strongly rely on texture for classification, significantly more than on the object's shape [3]. While texture or color is a powerful classification cue, we do not want the classifier to ignore shape information completely. Therefore, we advocate a generative viewpoint. However, simply training, e.g., a β-VAE [4] on this dataset, does not allow for generating data points of unseen combinations – the VAE cannot generate green zeros if all zeros in the training data are red.

Structural Causal Models In representation learning, it is commonly assumed that a potentially complex function f generates images from a small set of high-level semantic variables. Most previous work, e.g, [9], imposes no restrictions on f, i.e., a neural network is trained to map directly from a low-dimensional latent space to images. We follow the argument that rather than training a monolithic network to map from a latent space to images, the mapping should be decomposed into several functions. Each of these functions is autonomous, e.g., we can modify the background of an image while keeping all other aspects of the image unchanged. These demands coincide with the concept of structural causal models (SCMs) and independent mechanisms (IMs). An SCM $\mathfrak{C}$ is defined as a collection of d (structural) assignments

$$S_j := f_j\left(\mathbf{PA}_j, U_j\right), \quad j = 1, \ldots, d \tag{1}$$

where each random variable S_j is a function of its parents $\mathbf{PA}_j \subseteq \{S_1, \ldots, S_d\} \setminus \{S_j\}$ and a noise variable U_j. The noise variables $U_1, \ldots, U_d$ are jointly independent. The functions f_i are independent mechanisms, intervening on one mechanism f_j does not change the other mechanisms $\{f_1, \ldots, f_d\} \setminus \{f_j\}$. The SCM $\mathfrak{C}$ defines a unique distribution over the variables $\mathbf{S} = (S_1, \ldots, S_d)$ which is referred to as the *entailed distribution* $P_{\mathbf{S}}^{\mathfrak{C}}$. If one or more structural assignments are replaced, i.e., $S_k := \tilde{f}(\tilde{\mathbf{PA}}_k, \tilde{U}_k)$, this is called an intervention. We consider the case of *atomic interventions*, when $\tilde{f}(\tilde{\mathbf{PA}}_k, \tilde{U}_k)$ puts a point mass on a real value a. The entailed distribution then changes to the intervention distribution $P_{\mathbf{S}}^{\mathfrak{C};do(S_k:=a)}$, where the *do* refers to the intervention. If we learn a sensible set of IMs, we can intervene on a subset of them and generate *interventional images* $\mathbf{x}_{IV}$. These images were not part of the training data $\mathbf{x}$ as they are generated from the intervention distribution $P_{\mathbf{S}}^{\mathfrak{C};do(S_k:=a)}$. To generate a set of *counterfactual images* $\mathbf{x}_{CF}$, we fix the noise u and randomly draw a, hence answering counterfactual questions such as "How would *this* image look like with a different background?". In our case, a corresponds to a class label that we provide as input, denoted as y_{CF} in the following.

Training an Invariant Classifier To train an invariant classifier, we generate counterfactual images $\mathbf{x}_{CF}$, by intervening on all f_j simultaneously. Towards this goal, we draw labels uniformly from the set of possible labels $\mathcal{Y}$ for each f_j, i.e., each IM is conditioned on a different label. We denote the domain of images generated by all possible label permutations as $\mathcal{X}_{\mathcal{CF}}$. The task of the invariant classifier $r : \mathcal{X}_{\mathcal{CF}} \to \mathcal{Y}_{CF,k}$ is then to predict the label $\mathbf{y}_{CF,k}$ that was provided to one specific IM f_k – rendering r invariant wrt. all other IMs. This type of invariance is reminiscent of the idea of domain randomization (DR) [10]. In DR, we commonly assume access to the true generative model (the simulator). This assumption is not feasible without access to this model. It is also possible to train on interventional images $\mathbf{x}_{IV}$, i.e., generating a single image per sampled noise vector. Empirically, we find that counterfactual images improve performance over interventional ones.

3. Counterfactual Generative Networks

Our goal is to decompose the image generation process into several IMs. We assume the causal structure to be known. Concretely, we consider four IMs for the task of image classification: object shape, object texture, background, and image composition. The inherent structure of the model allows us to generate meaningful counterfactuals by construction. We refer to the entire generative model as Counterfactual Generative Network (CGN).

3.1. Independent Mechanisms

An overview of our CGN is shown in Figure 2. All IM-specific losses are optimized jointly end-to-end. For the experiments on ImageNet, we initialize each IM backbone with weights from a pre-trained BigGAN-deep-256 [2], the current state-of-the-art for conditional image generation.

Composition Mechanism. The function of the composer is not learned but defined analytically. Given the generated masks, textures and backgrounds, we composite the

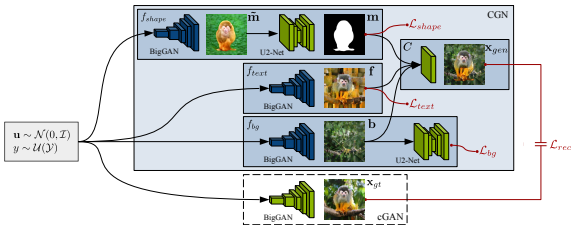

Figure 2: **CGN.** Here, we illustrate the architecture used for the ImageNet experiments. The CGN is split into four mechanisms, the shape mechanism f_{shape}, the texture mechanism f_{text}, the background mechanism f_{bg}, and the composer C. Components with trainable parameters are blue, components with fixed parameters are green.

image $\mathbf{x}_{gen}$ using alpha blending, denoted as C:

$$\mathbf{x}_{gen} = C(\mathbf{m}, \mathbf{f}, \mathbf{b}) = \mathbf{m} \odot \mathbf{f} + (1 - \mathbf{m}) \odot \mathbf{b} \quad (2)$$

where $\mathbf{m}$ is the mask (or alpha map), $\mathbf{f}$ is the foreground, and $\mathbf{b}$ is the background. The operator $\odot$ denotes element-wise multiplication. While, in general, IMs may be stochastic (Eq. 1), we did not find this to be necessary for the composer; therefore, we leave this mechanism deterministic. This fixed composition is a strong inductive bias in itself – the generator needs to generate realistic images through this bottleneck. To get a strong supervisory signal, we use an unconstrained, conditional GAN (cGAN) to generate pseudo-ground-truth images $\mathbf{x}_{gt}$ from noise $\mathbf{u}$ and label y. We feed the same $\mathbf{u}$ and y into the IMs to generate $\mathbf{x}_{gen}$ and minimize a reconstruction loss $\mathcal{L}_{rec}(\mathbf{x}_{gt}, \mathbf{x}_{gen})$.

Shape Mechanism. We model the shape using a binary mask predicted by shape IM f_{shape}, where 0 corresponds to the background and 1 to the object. The shape loss $\mathcal{L}_{shape}$ prohibits trivial solutions, i.e., masks with all 0's or 1's that are outside of a defined interval while at the same time forcing the output to be close to either 0 or 1. As we utilize a BigGAN backbone for our ImageNet-Experiments, we need to extract a binary mask from the backbone's output. Therefore, we add a pre-trained U2-Net [7] as a head on top of the BigGAN backbone. The U2-Net was trained for salient object detection, hence, it is class agnostic. While the U2-Net presents a strong bias towards binary object masks, it does not fully solve the task at hand as it captures non-class specific parts. By fine-tuning the BigGAN backbone, the IM learns to generate images of the relevant part with exaggerated features to increase saliency.

Texture Mechanism. The texture mechanism f_{text} is responsible for generating the foreground object's appearance, while not capturing any object shape or background cues. For MNIST, we use an architectural bias – an additional layer before the final output. This layer spatially divides its input into patches and randomly rearranges them, similar to a shuffled sliding puzzle. This conceptually simple idea does not work on ImageNet, as we want to preserve local object structure, e.g., the position of an eye. We, therefore, sample patches from the full composite image and concatenate them into a grid $\mathbf{pg}$. We then minimize a perceptual loss between the foreground $\mathbf{f}$ (the output of f_{text}) and the patchgrid: $\mathcal{L}_{text}(\mathbf{f}, \mathbf{pg})$. Over training, the background gradually transforms into object texture.

Background Mechanism. The background mechanism f_{bg} needs to capture the background's global structure while the object must be removed and inpainted realistically. However, we found that we cannot use standard inpainting techniques, as they are either too slow or do not work well on synthetic data because of the domain shift. Instead, we exploit the same U2-Net as used for the shape mechanism f_{shape}. Again, we feed the output of the BigGAN backbone through the U2-Net with fixed weights. However, this time, we *minimize* the predicted saliency. Over the progress of training, this leads to the object shrinking and finally disappearing, while the model learns to inpaint the object region.

3.2. Generating Counterfactuals to Train Invariant Classifiers

After training the CGN, each IM network has learned a class-conditional distribution over shapes, textures, or backgrounds. By randomizing the label input y and noise $\mathbf{u}$ of each network, we can generate counterfactual images. The number of possible combinations is the number of classes to the power of the number of IM's. For ImageNet, this is 1000^3. The amount of possible images is even larger since we learn distributions, i.e., we can generate a nearly unlimited variety of shapes, textures, and backgrounds, per class. We train on both real and counterfactual images.

4. Experiments

Our experiments aim to answer (i) if our approach can reliably learn the disentangled IMs, and (ii) if counterfactual images enable the training of invariant papers. We report selected results of the full-length paper.

4.1. Does our approach learn the disentangled independent mechanisms?

Standard metrics like FID are not applicable since the counterfactual images are outside of the natural image domain. We thus focus on qualitative results in this section. As shown in Figure 3, our CGN generates counterfactuals of high visual fidelity on ImageNet. We also find an unexpected benefit of our approach. In some instances, the composite images eliminate structural artifacts of the original BigGAN images, such as surplus legs.We hypothesize that f_{shape} learns a general shape concept per class, resulting in outliers, like elephants with eight legs, being smoothed out. The CGN can fail to produce high-quality texture maps for very small objects, e.g., for a bird high up in the sky, the

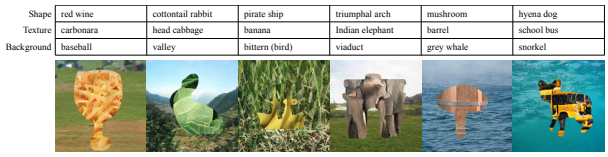

Figure 3: **Counterfactuals.** The CGN successfully learns the disentangled shape, texture, and background mechanisms, and enables the generation of permutations thereof.

	colored MNIST		double-colored MNIST		Wildlife MNIST	
	Train Acc ⇑	Test Acc ⇑	Train Acc ⇑	Test Acc ⇑	Train Acc ⇑	Test Acc ⇑
Original	99.5 %	35.9 %	**100.0** %	10.3 %	**100.0** %	10.1 %
IRM (2 Envs)	99.6 %	59.8 %	**100.0** %	67.7 %	99.9 %	11.3 %
IRM (5 Envs)	-	-	99.9 %	78.9 %	99.8 %	76.8 %
LNTL	99.3 %	81.8 %	98.7 %	69.9 %	99.9 %	11.5 %
Original + GAN	**99.8** %	40.7 %	**100.0** %	10.8 %	**100.0** %	10.4 %
Original + CGN	99.7 %	**95.1** %	97.4 %	**89.0** %	99.2 %	**85.7** %

Table 1: **MNISTs Classification**. In the test set, colors and textures are randomized.

texture map will still show large portions of the sky. Also, in some instances, a residue of the object is left on the background, e.g., a dog snout. For generating counterfactual images, this is not a problem as a different object will cover the residue. Lastly, the enforced constraints can lead to a reduction in realism of the composite images $\mathbf{x_{gen}}$ compared to the original BigGAN samples.

4.2. Do counterfactual images enable training of invariant classifiers?

In the following, we describe our results on the different MNIST variants. The label is encoded in the digit shape, foreground color or texture, and the background color or texture. In the training domain, all FoVs are correlated with the class label. In the test domain, only the shapes correspond to the correct class. We compare to current approaches for training invariant classifiers: IRM [1] and Learning-not-to-learn (LNTL) [5]. *Original + CGN* is additionally trained on counterfactual data to predict the input labels of the shape IM. *Original + GAN* is a baseline that is trained on real and generated, non-counterfactual samples.

IRM considers a signal to be causal if it is stable across several environments. We train IRM on 2 environments (90 % and 100 % correlation) or 5 environments (90 %, 92.5 %, 95 %, 97.5 %, and 100% correlation). LNTL considers color to be spurious, whereas we assume (complementary) that shapes are causal. The results in Table 1 confirm that training on counterfactual data leads to classifiers that are invariant to the spurious signals. We hypothesize that the difference between environments may be hard to pick up for IRM, especially if only a few are available. We find that we can further improve IRM's performance by adding more environments. However, continually increasing the number of environments is an unrealistic premise and only feasible in simulated environments. Our results indicate that LNTL and IRM have trouble scaling to more complex data.

5. Discussion

We assume that an image can be neatly distinguished into a class foreground and background throughout this work. This assumption breaks once we consider more complex scenes with different object instances or for tasks without a clear foreground-background distinction, e.g., in medical images. An exciting research direction is to explore different configurations of IMs to tackle this challenge. Further, in our experiments, we assume the causal structure to be known. This assumption is substantially stronger than in more general standard disentanglement frameworks [4]. A possible extension to our work could leverage causal discovery to isolate IMs in a domain-agnostic manner.

Acknowledgements. We acknowledge the financial support by the BMWi in the project KI Delta Learning (project number 19A19013O). Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533).

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 4

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018. 2

[3] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020. 1, 2

[4] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 2, 4

[5] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, 2019. 2, 4

[6] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. In *ICLR*, 2018. 1

[7] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 2020. 3

[8] Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019. 1

[9] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019. 1, 2

[10] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017. 2