DISENTANGLING CAUSAL MECHANISMS BY OBSTRUCTING CLASSIFIERS

Suraj Anand, Neil Xu
Brown University
{suraj_anand}, {neil_xu}@brown.edu

ABSTRACT

Deep neural networks (DNNs) often entangle features in latent representations leads to a reliance on spurious correlations, black-box behavior causing uninterpretable predictions, and non-compositional representations. We build on the idea of isolating independent mechanisms to build disentangled representations. We propose a novel method for learning to disentangle generative factors in the latent space by pressuring an explicitly mapped partition of causal mechanisms. We separate counterfactual image generation into independent causal mechanisms trained with supervision, creating a Disentangled Autoencoder. We do this by applying a projection-based classifier obstruction algorithm called R-LACE on latent space representations of the Colored MNIST dataset during training to separate the encoding of distinct features. In experiments testing our method, we were able to achieve disentanglement by partitioning the latent space into one subspace that encodes for digit and one subspace that encodes for color. [1]

1 Introduction

Deep neural networks (DNNs) have become a universal solution to modern statistical challenges. However, these models suffer from poor data sample efficiency. Moreover, DNNs are neither robust nor generalizable in comparison to biological intelligence [16, 19]. We attribute these shortcomings with DNNs to failures in learning a faithful representation of the underlying data structure [1, 3]. The tendency of DNNs to entangle features in latent representations leads to a reliance on spurious correlations (i.e. correlations that may not be present in out-of-sample data), black-box behavior causing uninterpretable predictions, and non-compositional representations [21, 23]. This is the motivation for an increasing body of work on disentangled representation learning, which explicitly learn latent representations where generative factors of the data are orthogonal and axis-aligned [6]. Disentangled representations have been shown to be more semantically interpretable [8, 15], more generalizable to out-of-distribution data [22], robust to adversarial attacks [2], and useful to numerous tasks including reinforcement learning [13], transfer learning [18], and sequential data generation [17].

To further concretize the problem, we provide an example. In the image classification domain, models may employ features such as background color or background texture to classify foreground objects [5, 10]. For instance, Beery et al. provides the example of a classifier trained to recognize cows in certain images [5]. Since images of cows in a real-world dataset will typically depict the animal against a green, grass textured background, the model may learn to associate the presence of such a background with the label "cow". As a result, a situation in which the model is given an image of a car parked on grassy field may yield the predicted label of "cow" as well. In this example, the model failed to learn a meaningful latent representation of the cow itself, but rather simply learned the simple correlational relationship of green, grassy background and the label "cow". A disentangled representation that compositionally represents the background and the foreground would not be susceptible to such spurious correlations [21].

Past research into disentangled representation learning includes InfoGAN, which maximizes the mutual information between a fixed subset of noise variables and the observations [8], and β-VAEs, which introduces a tunable constraint for implicit independence pressure on the learned posterior [12]. We build upon this

[1] Code available at https://github.com/surajK610/interventions-autoencoder

idea of isolating *independent mechanisms* to build disentangled representations. This approach considers a generative process to be a composition of a number of independent, naturally-arising features. For instance, a common perspective for engineering image classification systems represents images as a composition of the independently generated concepts of object, texture, and background. Recent work by Saur & Geiger in Counterfactual Generative Networks isolates these mechanism by exploiting inductive biases specific to shape, background, and texture [21]. In constrast, we aim to create a more general framework for isolating independent mechanisms by employing linear projection to obstruct the superposition of specified features. We pressure the formation of independent mechanisms within generative models through Relaxed Linear Adversarial Concept Erasure (R-LACE). R-LACE removes a linear subspace of rank r from the embedding vector space at a certain layer in the network by employing an adversarial minimax game. This method has shown promising results when used in applications involving related to debiasing, such as removing gender information from certain words and phrases [20].

We hypothesize that a DNN can learn independent mechanisms by explicitly pressuring independent generative factors to occupy axis-aligned, orthogonal vector spaces. We accomplish this by continual projection, obstructing classification in a linear probe of all but one independent generative factors in an axis-aligned subspace of the latent space. Intuitively, this may be thought of as imposing an information bottleneck for each independent mechanism that only enables flow through a certain region. This approach has the advantage of a known explicit mapping to generation factors. However, the training procedure is not always stable and causal mechanisms must be specified.

To demonstrate the viability of this approach, we show that we can learn independent mechanisms in the latent space of a simple Autoencoder. We employ R-LACE to enforce independent mechanisms by partitioning the latent space into discrete axis-aligned dimensions that are mapped to distinct features. Through this constraint, we remove spurious inter-feature correlations and can generate. Our work makes the following contributions:

- We propose a generalizable framework for learning completely disentangled representations by pressuring independent generative factors to occupy orthogonal (and axis-aligned) vector spaces.
- We present a procedure for training a simple image Autoencoder and that utilizes the projection-based method of R-LACE to disentangle independent mechanisms within the latent space.
- We demonstrate the viability of our method by partitioning the latent space representation of a Colored MNIST dataset into two disjoint orthogononal sets of dimensions that encode for color and digit. We prove the use of these disentangled representations in warding against spurious correlations and interpretable generation.

2 Methods

2.1 Problem Setting

The process of image generation and classification involves a dataset that consists of labeled images. Using DNNs, we assume that the high dimensional image data can be broken down into a set of lower dimensional features and concepts that convey some meaning. We consider the Colored-MNIST dataset [14], which consists of labeled handwritten digits from 0-9 and colored either red, green, or blue. We construct colored MNIST such that color and number are independent generated. We chose this scheme to verify that our disentanglement methodology works. We aim to partition color and number into orthogonal, axis-aligned

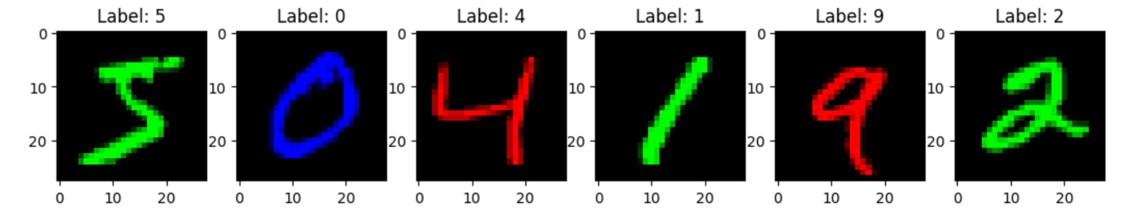


Figure 1: Examples from the Colored-MNIST dataset

dimensions. Using R-LACE, we seek to train a model that is invariant to learning such correlations by disentangling the latent space into dimensions that represent color encodings, and dimensions that represent digit encodings.

2.2 Relaxed Linear Adversarial Concept Erasure (R-LACE)

We approach the problem of disentangling the latent space representation of the Colored-MNIST dataset through projection-based classifier obstruction. R-LACE formulates concept identification and removal as a constrained, linear minimax game [20]. The method seeks to discover a low dimensional subspace representing a particular feature. Upon removal of this subspace through an orthogonal projection of the vector representation, we can eliminate the information regarding this concept (2).

The linear minimax game searches for an orthogonal projection matrix P that projects onto $B\ \perp$, the orthogonal complement of the bias subspace B. Let $\mathcal{P}_k$ be the set of all $D \times D$ orthogonal projection matrices that neutralize a rank k subspace. Therefore, we have that

$$P \in \mathcal{P}_k \iff \underbrace{P = I_D - W^\top W}_{\text{Projects to null(W)}}, \underbrace{W \in \mathbb{R}^{K \times D}}_{\text{rank(W)}\leq k}, \underbrace{WW^\top = I_k}_{\text{orthogonal}}$$

Where I_k represents the $k \times k$ identity matrix and I_D is the $D \times D$ identity matrix. Thus P neutralizes the k dimensional subspace $B = span(W)$. The minimax game thus takes the following form:

$$\min_{\theta \in \Theta} \max_{P \in \mathcal{P}_k} \sum_{n=1}^{N} \ell(y_n, g^{-1}(\theta^\top P x_n))$$

where k is a hyperparameter that represents the rank of the neutralized subspace, l is the loss function, and g^{-1} is the inverse of a generalized linear model link function.

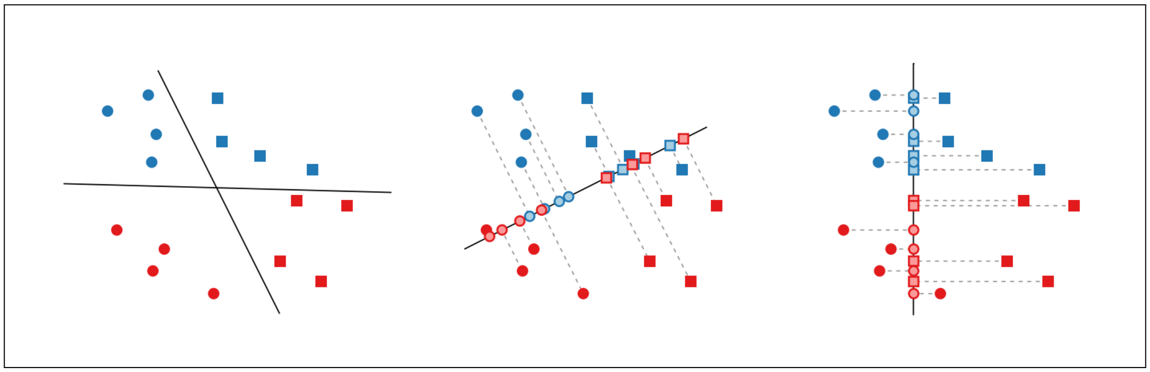


Figure 2: Left: a set of data with two linearly-separable features (ex. shape and color). Middle: taking the orthogonal projection of the color boundary removes color information from the vector representations of the data and obstructs color classification. Right: taking the orthogonal projection of the shape boundary has the opposite effect [11]

2.3 Model Details

The full architecture of our autoencoder model is visualized in Figure 3. Built around a vanilla autoencoder with a Mean Squared Error (MSE) objective function, our model encodes Colored-MNIST images into a low dimensional latent space $(d = 2n, n = 1, 2, 3, 5)$. To disentangle the latent space, we first apply R-LACE to the first n dimensions to remove all information regarding color. We then apply R-LACE to the remaining n dimensions to remove all information regarding digit. This effectively constrains the first n latent dimensions to exclusively encode for the digit and the remaining n dimensions to exclusively encode for color.

The encoder network consists of several convolution, batch norm, and ReLU layers that encode input images into a low dimensional latent space of size $2n$, which in experiments was 6, 8, or 10 $(n = 3, 4, 5)$ dimensions. This latent space is then partitioned into two vectors, each with n dimensions. Next, we run R-LACE on the first vector in order to remove the information in the vector regarding the color of the digit and only retain

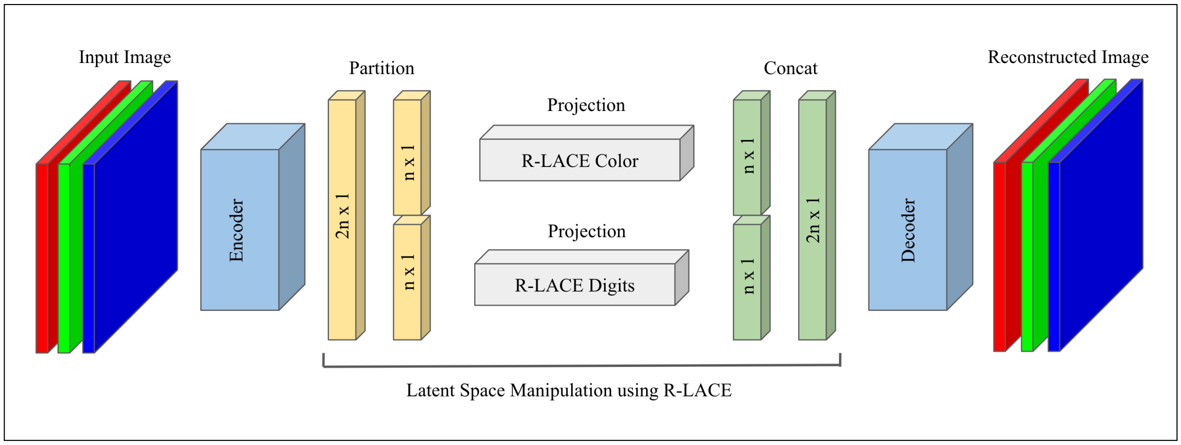


Figure 3: This autoencoder network makes use of R-LACE to disentangle the latent space and encourage Independent Mechanisms in representing the training data. The network consists of a vanilla autoencoder and a latent space modification step in which the encoded latent space is partitioned into two vectors. Each vector is fed into R-LACE, which attempts to remove the information regarding one feature. Afterwards, the transformed vectors are concatenated back into the latent space dimension and fed through the decoder.

information regarding the digit in the image. A similar process is done for the second vector such that it only retains information regarding image color. Afterwards, these two augmented vectors are concatenated back to a $2n$ dimensional “disentangled" latent space. Finally, the disentangled latent space is passed through the decoder network which reconstructs the input image.

2.4 Disentangled Autoencoder Training and Evaluating

The training process of this modified Autoencoder architecture involves alternating between training the Autoencoder to generate accurate reconstructions of the input images, and training R-LACE to learn the orthogonal projection matrices that will remove color and shape information from the two latent vector partitions. In practice, we employ a scheme that alternates between training the Autoencoder for n epochs, and training R-LACE to remove a rank 1 subspace of information for 1 epoch.

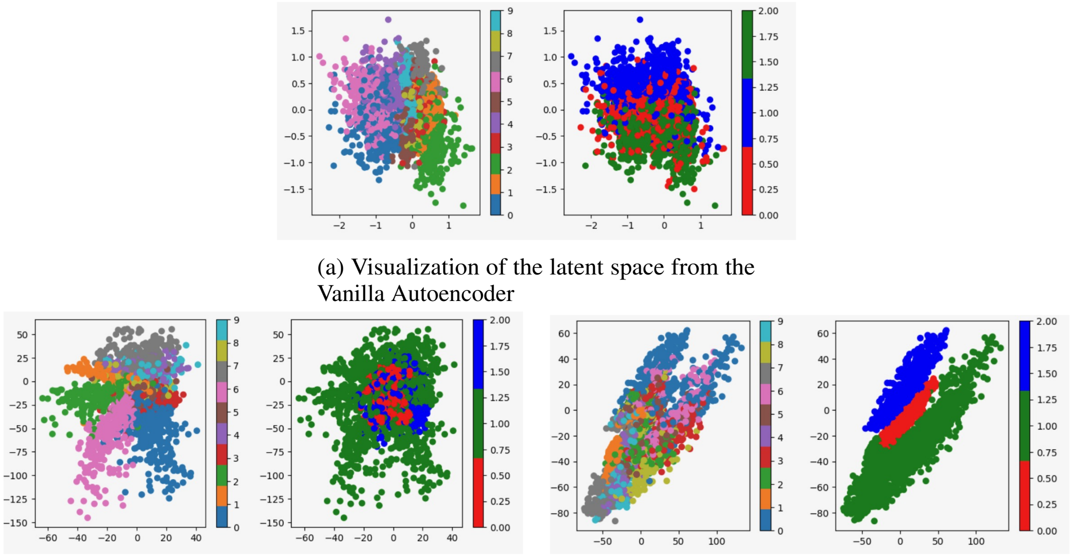
2.5 Experiments

As a baseline, we first train a Vanilla Autoencoder with 4 latent dimensions and the same encoder and decoder architecture as our Disentangled Autoencoder. We also train a logistic classifier on the latent space of the Vanilla Autoencoder to determine a baseline level of digit and color classification accuracy.

Next, we train the Disentangled Autoencoder using the scheme describe in 2.4. We train a logistic classifier on the concatenated latent space and compare the digit and color classification accuracy against the baseline. To evaluate disentanglement, we train logistic classifiers on the partitioned latent subspaces that have been orthogonally projected. We let a classifier for the “opposite" encoding refer to a classifier for a concept that was trained on the latent partition that has had that concept removed through orthogonal projection (e.x classifying digit using the latent partition encoding for color). Likewise, a “corresponding" encoding is a classifier for a feature that has been trained on the partitioned latent subspace that encodes for that feature (i.e classifying digit using the latent partition encoding for digit). We train an opposite and corresponding logistic classifier for the both projected latent space partitions. These classification accuracy values inform the degree of disentanglement achieved through R-LACE.

3 Results

Our experiments aim to determine whether using R-LACE to construct a Disentangled Autoencoder is able to successfully disentangle the latent space into a set of dimensions that encode for digit and a set that encodes for color, while preserving reconstruction accuracy compared to the Vanilla Autoencoder.



(a) Visualization of the latent space from the Vanilla Autoencoder

(b) The disentangled latent space dimensions representing the digit. Here, color information seems to have been removed through R-LACE as there no longer appears to be linear separability for color.

(c) The disentangled latent space dimensions representing image color. Here, information regarding the digit appears to have been removed, as the digit classes are highly entangled in this visualization.

Figure 4: Visualization of the disentangled latent dimensions from the Disentangled Autoencoder

After training the Vanilla Autoencoder for 100 epochs, we observe a reconstruction loss of 0.009. Next, we train the Disentangled Autoencoder using the scheme described in 2.4 where the partitioned latent dimension, n, is $n = 1, 2, 3, 5$. The reconstruction accuracy values and classification accuracy values are shown in 1. These experiments indicated that a partitioned latent dimension of at least $n = 3$ is required to successfully encode and reconstruct the images. When the partitioned latent dimension is too low $(n = 1, 2)$, the model struggles to adequately encode both color and digit. Since we remove information regarding a particular feature from each latent partition, if the latent dimension is $2n$, we are effectively allocating only n dimensions for color encoding and n for digit encoding. Thus given a four dimensional latent space in the Vanilla Autoencoder, we can expect digit reconstruction to behave roughly as a Vanilla Autoencoder with only two latent dimensions. The poor reconstruction loss when $n = 1, 2$ is thus somewhat expected, as prior work has shown that the MNIST dataset requires at least two latent dimensions to properly reconstruct the images [4]. Moreover, when $n = 1$, we see random accuracy for both classes as RLACE removes a rank 1 subspace which contains all of the information.

As the size of the latent dimension partitions increases, we observe that reconstruction loss converges to 0.009 with a partition size of $n = 3$ andge 0.006 with a partition size of $n = 5$. However, digit and color classification accuracy when measured using a classifier trained on the concatenated disentangled latent space of size $2n$ appears to be slightly lower than that of the Vanilla Autoencoder, but still within acceptable ranges 2. These results validate the effectiveness of the Disentangled Autoencoder in reconstructing the Colored MNIST dataset, thus retaining its functionality as an Autoencoder.

Next, we want to confirm whether R-LACE successfully remove information regarding color from the dimensions encoding for digit and vice versa. In order to verify this condition, we train two linear classifiers for each latent space partitions: one for digit and another for color. A successfully trained Disentangled Autoencoder should allow for the classifier of the feature represented by the latent space partition, for instance, digit, to perform as well as a baseline classifier on MNIST, while the classifier for color should perform about as well as guessing. The results of these four classifiers on the various latent space dimensions are show in 3.

Based on these results, we see that when the latent partition size is only 1, the digit and color classification accuracy is very poor, with a logistic classifier only capable of achieving random accuracy. For a latent partition size of $n = 2$, we seem to achieve partial disentanglement of color and digit within the latent space.

Reconstruction Loss	Digit Classification Accuracy	Color Classification Accuracy
.009	0.93	1.00

Table 1: Reconstruction and Classification Accuracy of Vanilla Autoencoder Trained on Randomly Colored Digits

Latent Partition Size	Reconstruction Loss	Digit Classification Accuracy	Color Classification Accuracy
1	.03	0.55	0.33
2	.02	0.51	0.95
3	.009	0.89	0.92
5	.006	0.89	0.97

Table 2: Reconstruction and Classification Accuracy of Disentangled Autoencoder.

Latent Partition Size	Latent Dimension	Digit Classification Accurracy	Color Classification Accuracy
1	Digit Encoding	0.12	0.32
1	Color Encoding	0.12	0.32
2	Digit Encoding	0.42	0.31
2	Color Encoding	0.10	0.66
3	Digit Encoding	0.42	0.38
3	Color Encoding	0.32	0.94
5	Digit Encoding	0.74	0.48
5	Color Encoding	0.39	0.99

Table 3: Reconstruction and classification accuracy of Disentangled Autoencoder trained on randomly colored digits

Within the latent subspace that has been modified by the orthogonal projection matrix for the vector subspace representing color (i.e Digit Encoding), color classification accuracy is approximately random. These results indicate that information regarding color has largely been removed from these dimensions as a logistic classifier trained on this latent partition is unable to learn a means of identifying color. When we consider the Color Encoding, digit classification accuracy is random, indicating full removal of digit information from the Color Encoding subspace. However, we observe that the logistic classifier achieves poor performance on the features encoded by the latent subspaces, with digit classification accuracy of only 0.42, indicating that though information regarding features are being removed from the corresponding subspaces, the encoder may be encoding information across latent subspaces that, when R-LACE is run, does not identify this information as a rank-1 subspace that can be orthogonally projected.

At the highest latent subspace dimension tested $(n = 5)$, we observe that digit classification accuracy is much higher in the latent dimensions representing Digit Encoding, at 0.74. Likewise, color classification accuracy using a logistic classifier on the Color Encoding yields nearly perfect accuracy at 0.99. These results indicate that as the latent dimension partition size increases, the encoder seems to be able to learn to more discretely encode a single feature into each latent partition. However, it is also apparent that using R-LACE to remove a rank-1 vector subspace from each latent partition no longer perfectly removes the information regarding the other feature. Namely, color classification accuracy in the digit encoding is 0.48, and digit classification accuracy in the color encoding is 0.39, which are both significantly higher than random accuracy. These values may indicate that the classifier is still able to learn some small amount of information regarding the feature we are attempting to remove after we have performed the orthogonal projection. Despite this possibility, the low accuracy achieved by the classifiers on the opposing encodings, along with the high accuracy of the classifiers on the corresponding encodings, serves as evidence for a high degree of latent space disentanglement.

Our results indicate that for a latent partition size of 5 dimensions, we are able to achieve effective disentanglement of the latent space into two vector subspaces that exclusively encode information regarding either color or digit. We therefore show that our Disentangled Autoencoder is able to achieve reconstruction loss equivalent to that of the Vanilla Autoencoder while maintaining a disentangled latent space.

4 Discussion

In this research, we explore the hypothesis that a DNN can learn to disentangle generative factors in the latent space by pressuring an explicitly mapped partition of causal mechanisms. In the context of current disentangled representation learning, we believe that this is an altogether novel method and have not found any similar methods in our literature search. Our method is unique in that it allows an explicit mapping of known generative factors to the latent space. We find strong evidence that by continually obstructing classifiers on the partitioned latent space, we may create an information bottleneck that forces disentangled generative factors. We believe that this is an important pioneering effort into creating disentangled representations by projection-based methods, and are excited about future work that builds upon this effort.

It is worth noting that our method strongly relies on the assumption that causal mechanisms are independent. Without this assumption, we would not be able to obstruct classification of one generative factor without also obstructing classification of other dependent factors. Moreover, we also assume the causal structure to be known and assume the generative factors of our dataset to be known. One potential extension to our research could utilize causal discovery to isolate independent mechanisms using techniques such as meta-learning [7]. However, if causal structure is known, our method enables machine learning practitioners to enforce this structure in a principled, yet flexible way, increasing sample efficiency, internal compositional representations, and domain knowledge integration [9]. Furthermore, in contrast to traditional feature engineering domain knowledge integration, our method maintains the functional approximation flexibility of an end-to-end neural system. This marriage of causal structure and end-to-end neural function approximation can help us build safer, better-aligned medical diagnoses systems, autonomous transportation, and chatbots.

References

[1] Alessandro Achille and Stefano Soatto. *Emergence of Invariance and Disentanglement in Deep Representations*. 2018. arXiv: 1706.01350 [cs.LG].

[2] Alexander A. Alemi et al. *Deep Variational Information Bottleneck*. 2019. arXiv: 1612.00410 [cs.LG].

[3] Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. *On Invariance and Selectivity in Representation Learning*. 2015. arXiv: 1503.05938 [cs.LG].

[4] Dor Bank, Noam Koenigstein, and Raja Giryes. *Autoencoders*. 2021. arXiv: 2003.05991 [cs.LG].

[5] Sara Beery, Grant van Horn, and Pietro Perona. *Recognition in Terra Incognita*. 2018. arXiv: 1807.04975 [cs.CV].

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[7] Yoshua Bengio et al. *A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms*. 2019. arXiv: 1901.10912 [cs.LG].

[8] Xi Chen et al. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*. 2016. arXiv: 1606.03657 [cs.LG].

[9] Alexander D’Amour et al. *Underspecification Presents Challenges for Credibility in Modern Machine Learning*. 2020. arXiv: 2011.03395 [cs.LG].

[10] Robert Geirhos et al. *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. 2022. arXiv: 1811.12231 [cs.CV].

[11] Pantea Haghighatkhah et al. *Obstructing Classification via Projection*. 2021. arXiv: 2105.09047 [cs.CG].

[12] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations*. 2017. URL: `https://openreview.net/forum?id=Sy2fzU9gl`.

[13] Irina Higgins et al. *DARLA: Improving Zero-Shot Transfer in Reinforcement Learning*. 2018. arXiv: `1707.08475 [stat.ML]`.

[14] Byungju Kim et al. *Learning Not to Learn: Training Deep Neural Networks with Biased Data*. 2019. arXiv: `1812.10352 [cs.CV]`.

[15] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. *Variational Inference of Disentangled Latent Concepts from Unlabeled Observations*. 2018. arXiv: `1711.00848 [cs.LG]`.

[16] Brenden M. Lake et al. *Building Machines That Learn and Think Like People*. 2016. arXiv: `1604.00289 [cs.AI]`.

[17] Yingzhen Li and Stephan Mandt. *Disentangled Sequential Autoencoder*. 2018. arXiv: `1803.02991 [cs.LG]`.

[18] Alexander H. Liu et al. "A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper_files/paper/2018/file/84438b7aae55a0638073ef798e50b4ef-Paper.pdf`.

[19] Gary Marcus. *Deep Learning: A Critical Appraisal*. 2018. arXiv: `1801.00631 [cs.AI]`.

[20] Shauli Ravfogel et al. *Linear Adversarial Concept Erasure*. 2022. arXiv: `2201.12091 [cs.LG]`.

[21] Axel Sauer and Andreas Geiger. *Counterfactual Generative Networks*. 2021. arXiv: `2101.06046 [cs.LG]`.

[22] Xander Steenbrugge et al. *Improving Generalization for Abstract Reasoning Tasks Using Disentangled Feature Representations*. 2018. arXiv: `1811.04784 [cs.LG]`.

[23] Hao Zheng and Mirella Lapata. *Disentangled Sequence to Sequence Learning for Compositional Generalization*. 2022. arXiv: `2110.04655 [cs.CL]`.