TOWARDS DEEP VISUAL LEARNING IN THE WILD: DATA-EFFICIENCY, ROBUSTNESS AND GENERALIZATION

Zhenlin Xu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2022

Approved by:

Marc Niethammer

Colin Raffel

Martin Styner

Junier Oliva

Shashank Srivastava

ABSTRACT

Zhenlin Xu: Towards Deep Visual Learning in the Wild: Data-efficiency, Robustness and Generalization
(Under the direction of Marc Niethammer and Colin Raffel)

Deep Learning approaches have achieved revolutionary performance improvement on many computer vision tasks from understanding natural images and videos to analyzing medical images. Besides building more complex deep neural networks (DNNs) and collecting giant annotated datasets to obtain performance gains, more attention has now been focused on the shortcomings of DNNs. As recent research has shown, even when trained on millions of labeled samples, deep neural networks may still lack robustness to domain shift, small perturbations, and adversarial examples. On the other hand, in many real-world scenarios, e.g. in clinical applications, the number of labeled training samples is significantly smaller than for large existing deep learning benchmarks. Moreover, current deep learning models cannot generalize to samples with novel combinations of seen elementary concepts.

Therefore, in this thesis, I focus on handling the critical needs to make modern deep learning approaches applicable in the real-world with a focus on computer vision tasks. Specifically, I focus on data efficiency, robustness, and generalization. I propose (1) `DeepAtlas`, a joint learning framework for image registration and segmentation that can learn DNNs for both tasks from unlabeled images and a few labeled images. (2) `RandConv`, a data augmentation technique that applies a random convolution layer on images during training to improve the generalization performance of a DNN in the presence of domain shift and robustness to image corruptions. (3) A comprehensive study of compositional generalization in unsupervised representation learning on disentanglement and emergent language models.

To my parents, friends and those who loved, helped, and supported me along the journey.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my Ph.D. advisor Dr. Marc Niethammer and Dr. Colin Raffel. Over the past five years, Marc has been supporting me to explore research that I am interested in and excited about even when the topic is outside his focus. The critical thinking in research I learned from him benefits me the most and are essential for me to grow into an independent researcher. Colin started to advise me since he joined UNC when I just started to explore more fundamental topics in computer vision and machine learning. His insights on various topics of deep learning greatly broadened my research thinking and inspired me to find new research perspectives. The patience and encouragement of Marc and Colin helped me go through many challenges along my Ph.D. journey. I feel grateful and honored to work with them. I cannot imagine I can accomplish what I did today without their guidance.

I would like to thank my other committee members (Martin Styner, Junier Oliva, Shashank Srivastava) for their constant support and for providing valuable feedback in completing this dissertation. I also thanks my research collaborators and close friends Deyi Liu and Junlin Yang for their contribution. I also thank those who hosted me for internships: Kyle Bradbury at Duke University; Eli Gibson, Siqi Liu and Sasa Grbic at Siemens Healthineers; Andriy Myronenko, Daguang Xu at Nvidia; Marya Khademi, Simon Knornblith, Ting Chen, Dilip Krishnan at Google. I am grateful for the opportunities and the great experience that improved my technique skills, introduced me to new topics and further inspired new ideas in my own research. I am very lucky to work in the field of machine learning with an open research community. I want to thank all the contributors who open-sourced their work and code and freely shared their knowledge.

I would like to thank all the UNC-BIAG group members (Xu Han, Heather D. Couture, Zhipeng Ding, Zhengyang Shen, Yifeng Shi, Peirong Liu, Lin Tian, Benjamin Levine, Sahin Olut, Boqi Chen, Hastings Greer, Chun-Hung Chao, Kenya Vazquez Martinez, Yining Jiao, Qin Liu, and Nurislam

Tursynbek) and the r-three group members (Nikhil Kandpal, Derek Tam, Michael Matena, Haokun Liu, Anisha Mascarenhas, Muqeeth Mohammed, Vishal Baskaran, Huang Tenghao, Yi-Lin Sung, Jay Mohta, Ellie Evans) for the brainstorming, research discussion, and hanging out, which made my Ph.D. experience joyful and interactive. I also express my gratitude to all the stuff members at UNC Computer Science Department especially Bill Hayes, Murrey Anderegg, and John Sopko for setting up and maintaining computing clusters, and Denise Kenney for the processing all the paper works and course management.

Ph.D. is such a long journey with many challenges and difficulties that I often felt struggling and upset. I would like to thank my parents (Wenwu Xu and Mei Li) for their supporting and being there for all my life. I want to thank the accompany of all my friends over the years: Qiuyu Xiao, Ming Yang, Rui Wang, Hao Tan, Zhen Wei, Ruibin Ma, Junhua Yan, Conny Lu, Hao Jiang, Jie Lei, Yixin Nie, Menyu Fu, Yubo Luo, Yuan Tian, Zipei Zhu, Yufei Hou, and too many to list. I want to thank Ruiyang Zhao who has been constantly supporting and listening to me remotely. I also want to express my special appreciation to my mental health coach Shahnaz Khawaja, who helped me through the hardest time in my Ph.D.

I am grateful to be supported by the Dissertation Completion Fellowship of Royster Society of Fellows as well as NIH grants to finish my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Ingelligence
CNN	Convolutional Neural Network
DA	DeepAtlas
DG	Domain Generalization
DR	Domain Randomization
DL	Deep Learning
DNN	Deep Neural Network
EL	Emergent Language
GBT	Gradient Boosting Tree
LSTM	Long Short-Term Memory
MI	Mutual Information
MLP	Multi-Layer Perceptron
MRI	Magnetic Resonance Imaging
NCC	Normalized Cross Correlation
SSD	Sum of Squared Differences
VAE	Variational Auto-Encoder

CHAPTER 1: Introduction

1.1 Motivation

In the recent decade, deep learning has achieved significant success for applications in speech and natural language processing, computer vision, and so on. Deep neural networks (DNNs) of a large number of parameters trained on large-scale labeled datasets often outperform alternative methods in most academic benchmarks and can sometimes even beat human performance. However, researchers also point out the issues and concerns when applying deep learning to real-world scenarios. For example, annotating a larger scale data for every new task is expensive and impossible in some domain e.g. medical image analysis; A trained deep neural network may encounter samples that are out of the distribution of training samples and the performance usually degrade significantly in this scenario; In supervised learning on data without enough diversity, the deep network is good at find the "short-cut" features irrelevant to the task in the training dataset and will generalize poorly to real world testing data.

Therefore, in this thesis, I focus on handling the critical needs to make modern deep learning approaches applicable in the real-world with a focus on computer vision tasks.

1.2 Thesis Statement and Contributions

Injecting prior-knowledge into deep neural networks by means of inductive biases on task relations, model biases, representation format, can improve data-efficiency, robustness and generalization for deep visual learning.

I present following contributions in this thesis:

1. I develop `DeepAtlas`, a joint learning framework for image registration and segmentation that can learn DNNs for both tasks from unlabeled images and a few labeled images.

2. I propose `RandConv`, a data augmentation technique that applies a random convolution layer on images during training to improve the generalization performance of a DNN in the presence of domain shift and robustness to image corruptions.

3. I present a comprehensive study of `compositional generalization` in unsupervised representation learning that includes disentanglement and emergent language models. I propose an evaluation protocol based on transfer learning and observe that disentangled representation may not generalize well as we expected, while representations induced by emergent language learning show consistent good generalization performance.

1.3 Overview of Chapters

This thesis is organized as follows:

Chapter 2 presents the joint learning framework for image registration and segmentation that can learn DNNs for both tasks from unlabeled images and a few labeled images. Chapter 3 propose the data augmentation technique that applies a random convolution layer on images during training a DNN to improve its domain generalization and robustness performance. Chapter 4 presents a study on evaluating the compositional generalization of unsupervised representation learning algorithms. Chapter 5 concludes this thesis with a discussion of the contributions and an outlook on future work.

CHAPTER 2: DeepAtlas: Joint Semi-Supervised Learning of Image Registration and Segmentation

This chapter presents a framework to address the need for data-efficiency for deep-learning based medical image analysis. Obtaining 3D segmentations of medical images for supervised training is difficult and labor intensive. Motivated by classical approaches for joint segmentation and registration we therefore propose a deep learning framework that jointly learns networks for image registration and image segmentation. In contrast to previous work on deep unsupervised image registration, which showed the benefit of weak supervision via image segmentations, our approach can use existing segmentations when available and computes them via the segmentation network otherwise, thereby providing the same registration benefit. Conversely, segmentation network training benefits from the registration, which essentially provides a realistic form of data augmentation. Experiments on knee and brain 3D magnetic resonance (MR) images show that our approach achieves large simultaneous improvements of segmentation and registration accuracy (over independently trained networks) and allows training high-quality models with very limited labeled data and even one-shot segmentation.

The remainder of this chapter is organized as follows: Section 2.1 introduces the motivation and contribution of our work and Section 2.3 describes the proposed approach. Section 2.4 provides the experimental setup and details of the knee and brain MRI datasets; Section 2.5 presents results. Section 2.6 concludes this chapter. The work presented in this chapter was published in the Proceedings of the 2019 International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2019) (Xu and Niethammer, 2019).

2.1 Introduction

Image segmentation and registration are two crucial tasks in medical image analysis. They are also highly related and can help each other. *E.g.*, labeled atlas images are used via image

registration for segmentation. Segmentations can also provide additional supervision (in addition to image intensities) for image registration and are used to evaluate registration results. Consequently, joint image registration and segmentation approaches have been proposed. *E.g.*, approaches based on active contours (Yezzi et al., 2001) and Bayesian (Pohl et al., 2006) or Markov random field formulations (Mahapatra and Sun, 2010). While these methods jointly estimate registration and segmentation results, they operate on *individual* image pairs (instead of a population of images) and require the computationally costly minimization of an energy function.

Deep learning (DL) has been widely and successfully applied to medical image analysis. For supervised image segmentation, CNN-based approaches are faster and better than classical methods when many labeled training samples are available (Litjens et al., 2017). DL-based registration achieves similar performance to optimization-based approaches but is much faster. As true transformations are not available, training either uses estimates from optimization-based methods (Yang et al., 2017) or is unsupervised (Balakrishnan et al., 2018). Recent work (Balakrishnan et al., 2019) shows that weak supervision via an additional image segmentation loss between registered images can improve results over unsupervised training, which is based on only images. In practice, obtaining segmentations for 3D medical images is difficult and labor intensive. Hence, manual segmentations will often not be available for a large fraction of image data.

We propose **DeepAtlas**, to jointly learn deep networks for weakly supervised registration and semi-supervised segmentation. Our contributions are as follows:

- *We propose the first approach to jointly learn two deep neural networks for image registration and segmentation*. Previous joint approaches require joint optimizations for each pair of images. Instead, we jointly learn from a population of images during training, but can independently use the resulting segmentation and registration networks at test time.

- *Our joint approach only requires few manual segmentations*. Our two networks mutually guide each other's training on unlabeled images via an anatomy similarity loss. This loss penalizes the dissimilarity between the warped segmentation of the moving image and the segmentation of the target image. When registering image pairs consisting of a manually

labeled image and the estimate of a labeled image (via its network-predicted segmentation), this loss provides anatomy consistency supervision for registration and forces the predicted segmentation to match the manual segmentation after registration.

- *We evaluate our approach on large 3D brain and knee MRI datasets*. Using few manual segmentations, our method outperforms separately learned registration and segmentation networks. In the extreme case, where only one manually segmented image is available, our approach facilitates one-shot segmentation and boosts registration performance at the same time.

2.2 Background

2.2.1 Image Registration

Image registration is a process to find the spatial correspondence between a pair of images [1]: a moving image I_m and a target image I_t, such that their semantics, e.g. anatomical structures, are spatially aligned. It is an essential step for longitudinal analysis of the same patient or for the comparison of different patients within a cohort. The spatial correspondence between the pair of images is established by estimating a spatial transformation Φ that maps the moving image I_m to the same coordinate system as the target image I_t, which can be solved as an optimization problem:

$$\Phi^\star = \arg\min_{\Phi} Sim(I_m \circ \Phi^{-1}, I_t) + \lambda \cdot Reg(\Phi^{-1}) \tag{2.1}$$

where $\circ$ is the sampling operator that interpolates local values according to the inverse transform Φ^{-1} from I_t to I_m, $Reg(\cdot)$ is the function measure the irregularity of the transformation, and $Sim(\cdot, \cdot)$ measures the dissimilarity between the warped moving image and the target image. The optimal transformation maximizes the similarity between the pair of images with as low irregularity as possible.

[1] Although image registration can be done for a group of images, we focus on the case of pair-wise registration.

Transformation Depending on the desired degree of freedom of the desired transformation, Φ can be parametric transformations such as rigid, affine and b-spline transformations that may not need the regularization term, as well as more free non-parametric deformation models like elastic (Broit, 1981) deformation and fluid-based deformation (Bro-Nielsen and Gramkow, 1996).

Similarity Metrics that measures how well the two images are aligned are important for image registration and are often studied to improve the registration performance. The most common similarity metrics are often based on image intensities, and their details are below.

- *Sum of squared differences (SSD)* metric simply compares two images I_1 and I_2 by computing the SSD between their intensities.

$$SSD(I_1, I_2) = \sum_{x_i \in \Omega} (I_1(x_i) - I_2(x_i))^2, \tag{2.2}$$

  where Ω is the image spatial domain parameterized by x. SSD is a popular measure when two images are of the same modality and have a similar intensity range.

- *Normalized cross correlation (NCC)* is defined as the ratio between the cross-covariance of intensities between two images and the multiplication of the standard deviation of intensities of each image:

$$NCC(I_1, I_2) = \frac{\sum_{x_i \in \Omega}(I_1(x_i) - \bar{I}_1)(I_2(x_i) - \bar{I}_2)}{\sqrt{\sum_{x_i \in \Omega}(I_1(x_i) - \bar{I}_1)^2}\sqrt{\sum_{x_i \in \Omega}(I_2(x_i) - \bar{I}_2)^2}}, \tag{2.3}$$

  where $\bar{I}$ is the mean intensity of image I. NCC between two images is 1 or -1 when they are statistically dependent, and is 0 when independent. Since maximizing the statistical dependence of the aligned image is desired, $1 - NCC(I_1, I_2)^2$ is typically used in the energy function. Compared to SSD, NCC is insensitive to multiplicative factors between the two images.

- *Mutual Information (MI)* is a more flexible similarity measure than NCC that emerges from information theory. Given two random variables X and Y, the mutual information is defined as

$$MI(X,Y) = \int \log \frac{p(x,y)}{p(x)p(y)}\, dxdy, \tag{2.4}$$

where $p(x)$ and $p(y)$ are probability density function of X and Y, and $p(x,y)$ is the joint probability density function. For image registration, image intensities are viewed as random variables. $p(x,y)$ is the normalized joint intensity histogram. When two images are independent, $p(x,y) = p(x)p(y)$ and therefore the MI is zero. MI measures how well one can predict the intensity of the other image given the intensity of one image. The more similar the two images are, the higher the value of MI is. MI does not rely on one-to-one intensity correspondence. Therefore, it is well suited for multi-modality registration.

The traditional image registration directly optimizes a transformation for a pair of image according to 2.1 while the iterative optimization process can be computationally expensive. As deep neural networks have been widely used for image analysis, the image registration task can fit into the deep learning framework: training a deep network to predict the registration transformations for any pair of image inputs. To predict non-parameterized transformation output, e.g. displacement fields, deep networks with encoder-decoder designs, e.g. U-Net (Ronneberger et al., 2015), are often used. The registration deep network can be trained in a supervised way. However, there are no ground truth for image registration. Therefore, transformations generated by traditional registration approaches (Yang et al., 2017) or synthetic transformations (Sokooti et al., 2017) can be used as pseudo-ground truth. Although supervised methods have been shown to substantially accelerate registration and achieve accuracy comparable to traditional methods, the fundamental challenge of obtaining plausible ground-truth transformations remains and limits the supervised approaches. Therefore, unsupervised image registration is proposed by turning (2.1) into the DL framework:

$$\theta^\star = \arg\min_\theta Sim(I_m \circ \Phi^{-1}, I_t) + \lambda \cdot Reg(\Phi^{-1}) \tag{2.5}$$

where θ represent the learnable paramters of the registration network and the transformation Φ^{-1} is predicted by the network and will warp the moving image to compute the similarity loss as well as the regularization loss. Since the warping operation is implemented as a differentiable spatial transform network module (Jaderberg et al., 2015), the registration network can be trained end-to-end by (2.5). When image segmentations are available, an additional similarity loss can be computed based on segmentation, which results in weakly supervised registration.

2.2.2 Image Segmentation

Image Segmentation is a fundamental visual understanding task that partitions images or videos into objects or semantically different segments. Segmentation is important in various application domains, e.g. biomedical image analysis, video surveillance, autonomous driving, photo/video editing, and augmented reality to name a few. Before deep learning, numerous segmentation algorithms have been developed from simple approaches based on thresholding (Otsu, 1979), region growing (Adams and Bischof, 1994), clustering (Dhanachandra et al., 2015), and watersheds (Beucher, 1982), to more advanced algorithms such as active contour (Kass et al., 1988), graph cuts (Boykov et al., 2001), and conditional random fields (Lafferty et al., 2001). Similar to other computer vision tasks, segmentation performance is greatly improved by DL-based approaches. Under the DL framework, the segmentation task can be formulated as a pixel-wise classification problem: a deep network takes an image as input and outputs a dense label map. Many research efforts are focused on the design of model architectures and training loss functions, and we refer to Minaee et al. (2021) for a detailed survey.

In some specific domains, such as medical image analysis, obtaining dense annotations for training a deep network for segmentation is difficult and expensive. Therefore, learning a DL-based segmentation model with limited supervision is a critical and challenging problem. Effective solutions include data augmentation, transfer learning, leveraging unlabeled images, and learning from weaker supervisions (e.g. image-level labels). Tajbakhsh et al. (2020) provides a detailed review for related works.

Image registration has been used for segmentation with the atlas-based segmentation approach. In the single atlas case, the atlas, an image template I_a (moving image) associated with a segmentation map S_a, is registered to an image I_t (target image) that needs to be segmented. The registration result Φ^{-1} can warp the atlas segmentation S_a to the coordinate system of the target image $S_a \circ \Phi^{-1}$ which can be used as the segmentation of I_t. Multiple atlas can work together to improve segmentation performance (Aljabar et al., 2009).

2.3 Method

Our goal is to improve registration and segmentation accuracy when few manual segmentations are available for a large set of images by jointly learning a segmentation and a registration network. Fig. 2.1 illustrates our approach consisting of two parts: weakly supervised registration learning (solid blue lines) and semi-supervised segmentation learning (dashed yellow lines). Our loss is the weighted sum of the registration regularization loss ($\mathcal{L}_r$), the image similarity loss ($\mathcal{L}_i$), the anatomy loss ($\mathcal{L}_a$) penalizing segmentation dissimilarity, and the supervised segmentation loss ($\mathcal{L}_{sp}$). The losses $\{\mathcal{L}_r, \mathcal{L}_i, \mathcal{L}_a\}$ drive the weakly supervised learning of registration (Sec. 2.3.1) and the losses $\{\mathcal{L}_a, \mathcal{L}_{sp}\}$ drive the semi-supervised learning of segmentation (Sec. 2.3.2). Sec. 2.3.3 details the implementation.

2.3.1 Weakly-supervised Registration Learning

Given a pair of moving and target images I_m and I_t, a registration network $\mathcal{F}_R$ with parameters θ_r predicts a displacement field $\mathbf{u} = \mathcal{F}_R(I_m, I_t; \theta_r)$. This then allows warping the moving image to the target image space, $I_m^w = I_m \circ \Phi^{-1}$, where $\Phi^{-1} = \mathbf{u} + \mathrm{id}$ is the deformation map and id is the identity transform. A good map, Φ, maps related anatomical positions to each other. Unsupervised registration learning optimizes θ_r over an intensity similarity loss $\mathcal{L}_i$ (penalizing appearance differences between I_t and I_m^w) and a regularization loss $\mathcal{L}_r$ on $\mathbf{u}$ to encourage smooth transformations. Adding weak supervision by also matching segmentations between the target image (S_t) and the warped moving image ($S_m^w = S_m \circ \Phi^{-1}$) via an anatomy similarity loss $\mathcal{L}_a$ can improve registrations (Balakrishnan et al., 2019). Weakly-supervised registration learning is then

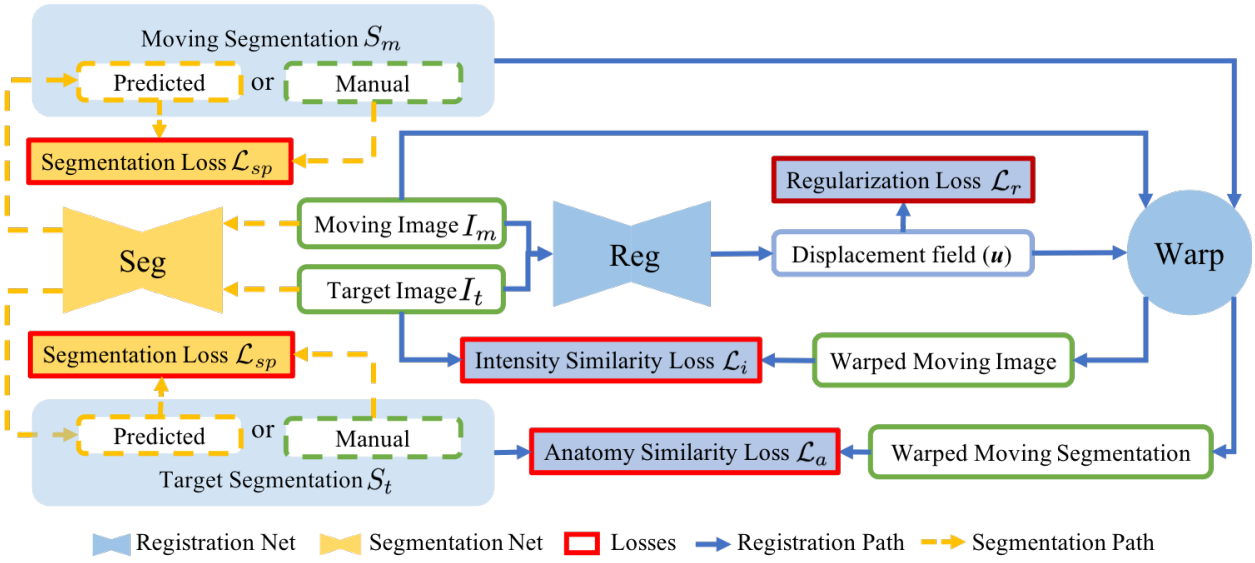


Figure 2.1: `DeepAtlas` for joint learning of weakly supervised registration and semi-supervised segmentation. Unlabeled moving/target images are segmented by the segmentation network so that every training registration pair has weak supervision via the anatomy similarity loss which also guides segmentation learning on unlabeled images.

formulated as:

$$\theta_r^\star = \underset{\theta_r}{\operatorname{argmin}}\{\mathcal{L}_i(I_m \circ \Phi^{-1}, I_t) + \lambda_r \mathcal{L}_r(\Phi^{-1}) + \lambda_a \mathcal{L}_a(S_m \circ \Phi^{-1}, S_t)\}, \tag{2.6}$$

with weights $\lambda_r, \lambda_a \geq 0$. In practice, while a large set of images are often available, few of them have manual segmentations. In contrast to existing work, we estimate missing moving or target segmentations via our segmentation network (see Fig. 2.1). Hence, we provide weak supervision for *every* training image pair.

2.3.2 Semi-supervised Segmentation Learning

The segmentation network $\mathcal{F}_S$ with parameters θ_s takes an image I as input and generates probabilistic segmentation maps for all semantic classes: $\hat{S} = \mathcal{F}_S(I; \theta_s)$. In addition to the typical supervised segmentation loss $\mathcal{L}_{sp}(\hat{S}, S)$ where S is a given manual segmentation, the anatomy similarity loss for registration $\mathcal{L}_a(S_m \circ \Phi^{-1}, S_t)$ also drives segmentation learning when S_m or S_t

are predicted via $\mathcal{F}_S$ for unlabeled images. Specifically, we define these losses as:

$$\mathcal{L}_{seg} = \begin{cases} \lambda_a \mathcal{L}_a(S_m \circ \Phi^{-1}, \mathcal{F}_{\mathcal{S}}(I_t)) + \lambda_{sp} \mathcal{L}_{sp}(\mathcal{F}_{\mathcal{S}}(I_m), S_m), \text{if } I_t \text{ is unlabeled;} \\ \lambda_a \mathcal{L}_a(\mathcal{F}_{\mathcal{S}}(I_m) \circ \Phi^{-1}, S_t) + \lambda_{sp} \mathcal{L}_{sp}(\mathcal{F}_{\mathcal{S}}(I_t), S_t), \text{if } I_m \text{ is unlabeled;} \\ \lambda_a \mathcal{L}_a(S_m \circ \Phi^{-1}, S_t) + \lambda_{sp} \mathcal{L}_{sp}(\mathcal{F}_{\mathcal{S}}(I_m), S_m), \text{if } I_m \text{ and } I_t \text{ are labeled;} \\ 0, \text{ if both } I_t \text{ and } I_m \text{ are unlabeled.} \end{cases} \tag{2.7}$$

with weights $\lambda_a, \lambda_{sp} \geq 0$. $\mathcal{L}_a$ teaches $\mathcal{F}_S$ to segment an unlabeled image such that the predicted segmentation matches the manual segmentation of a labeled image via $\mathcal{F}_R$. In the case where the target image I_t is unlabeled, $\mathcal{L}_a$ is equivalent to a supervised segmentation loss on I_t, in which the single-atlas segmentation $S_m \circ \Phi^{-1}$ is the noisy true label. Note that we do not use two unlabeled images for training and $\mathcal{L}_a$ does not train the segmentation network when both images are labeled. We then train our segmentation network in a semi-supervised manner as follows:

$$\theta_s^\star = \underset{\theta_s}{\operatorname{argmin}}\, \mathcal{L}_{seg}. \tag{2.8}$$

2.3.3 Implementation Details

Losses: Various choices are possible for the intensity/anatomy similarity, the segmentation, and the regularization losses. Our choices are as follows.

Anatomy similarity and supervised segmentation loss: A cross-entropy loss requires manually tuned class weights for imbalanced multi-class segmentations (Ronneberger et al., 2015). We use a soft multi-class Dice loss which addresses imbalances inherently:

$$\mathcal{L}_{dice}(S, S^\star) = 1 - \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_x S_k(x) S_k^\star(x)}{\sum_x S_k(x) + \sum_x S_k^\star(x)}, \tag{2.9}$$

where k indicates a segmentation label (out of K) and x is voxel location. S and $S^\star$ are two segmentations to be compared.

Intensity similarity loss: We use normalized cross correlation (NCC) as:

$$\mathcal{L}_i(I_m^w, I_t) = 1 - NCC^2(I_m^w, I_t), \tag{2.10}$$

which will be in $[0, 1]$ and hence will encourage maximal correlation.

Regularization loss: We use the bending energy (Rueckert et al., 1999):

$$\mathcal{L}_r(\mathbf{u}) = \frac{1}{N} \sum_{\mathbf{x}} \sum_{i=1}^{d} \|H(u_i(\mathbf{x}))\|_F^2 \tag{2.11}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $H(u_i(\mathbf{x}))$ is the Hessian of the i-th component of $\mathbf{u}(\mathbf{x})$, and d denotes the spatial dimension ($d = 3$ in our case). N denotes the number of voxels. Note that this is a second-order generalization of diffusion regularization, where one penalizes $\|\nabla u_i(\mathbf{x})\|_2^2$ instead of $\|H(u_i(\mathbf{x}))\|_F^2$.

Alternating training: It is in principle straightforward to optimize two networks according to Eqs. 2.6 and 2.8. However, as we work with the whole 3D images, not cropped patches, GPU memory is insufficient to simultaneously optimize the two networks in one forward pass. Hence, we alternately train one of the two networks while keeping the other fixed. We use a 1:20 ratio between training steps for the segmentation and registration networks, as the segmentation network converges faster. Since it is difficult to jointly train from scratch with unlabeled images, we independently pretrain both networks. When only few manual segmentations are available, *e.g.*, only one, separately training the segmentation network is challenging. In this case, we train the segmentation network from scratch using a fixed registration network trained unsupervisedly. We start alternating training when the segmentation network achieves reasonable performance.

Networks: DeepAtlas can use any CNN architecture for registration and segmentation. We use the network design of (Balakrishnan et al., 2018) for registration; and a customized light 3D U-Net design for segmentation with LeakyReLU instead of ReLU, and smaller feature size due to GPU memory limitations. Fig. 2.2 shows the architectures of the segmentation and registration networks.

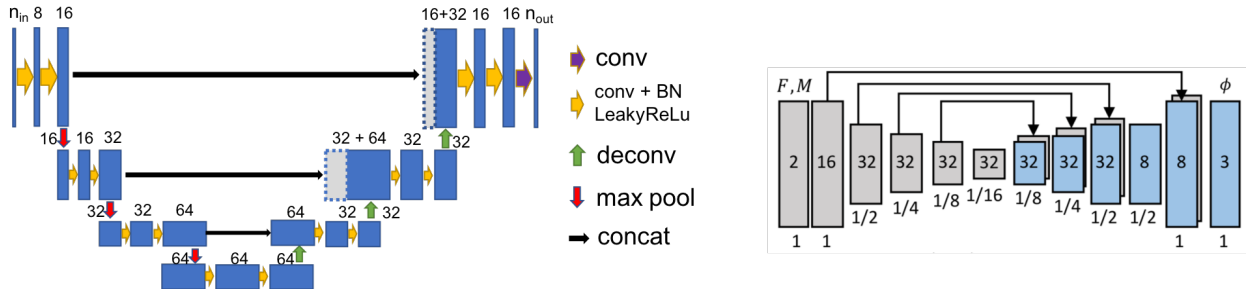


Figure 2.2: Architectures of the segmentation network (left) and the registration network (right). In the segmentation network, max-pooling is used for down-sampling for which 2-stride convolution is used in the registration network.

2.4 Experiments

We show on a 3D knee and a 3D brain MRI dataset that our framework improves both registration and segmentation when many images with few manual segmentations are available: i.e. N of M images are labeled ($N << M$).

Mono-networks: We train single segmentation/registration models as baselines. For segmentation, fully supervised networks are trained with N labeled images; the registration networks are trained via Eq. 2.6 using all M training images with N images labeled; the anatomy similarity loss, $\mathcal{L}_a$, is only used for training pairs where both images have manual segmentations. Models trained with $N = M$ manual segmentations (*i.e.*, with manual segmentations for all images) provide our upper performance bound. All mono-networks are trained for a sufficient number of epochs until they over-fit. The best models based on validation performance are evaluated.

Optimizer: We use the `Adam` optimizer. The initial learning rates are 1e-3 for the mono-networks. Initial learning rates are 5e-4 for the registration network and 1e-4 for the segmentation network for Semi-DA and DA. Learning rates decay by 0.2 at various epochs across experiments. We use PyTorch and run experiments on Nvidia V100 GPUs with 16GB memory.

DeepAtlas (DA): We initialize the joint model with the trained mono-networks. In addition to the alternately trained DA models, we hold one network fixed all through training, termed **Semi-DeepAtlas (Semi-DA)**.

In one-shot learning (N=1) experiments, training a supervised segmentation network based on a single labeled image is difficult; hence, we do not compute a segmentation mono-network in this case. For Semi-DA, we train a segmentation network from scratch with a fixed registration network that is trained unsupervised (N=0). The DA model is initialized using the Semi-DA segmentation network and the unsupervised registration network.

Knee MRI experiment: We test our method on 3D knee MRIs from the Osteoarthritis Initiative (OAI) [2] and corresponding segmentations of femur and tibia as well as femoral and tibial cartilage (Ambellan et al., 2019). From a total of 507 labeled images, we use 200 for training, 53 for validation, and 254 for testing. To test registration performance we use 10,000 random image pairs from the test set. All images are affinely registered to an atlas built from the training images, resampled to isotropic spacing of 1mm, cropped to $160 \times 160 \times 160$ and intensity normalized to [0,1]. Also, the images of the right knees are flipped to be consistent with the images of the left knees. For training, the loss weights are $\lambda_r = 20,000$, $\lambda_a = 3$, and $\lambda_{sp} = 3$ based on approximate hyper-parameter tuning. Note that when computing $\mathcal{L}_r$ from the displacements, the image coordinates are scaled to [-1, 1] for each dimension following the convention in the interpolation function of PyTorch.

N	Models	Segmentation Dice (%)			Registration Dice (%)		
		Bones	Cartilages	All	Bones	Cartilages	All
0	Mono	-	-	-	95.32(1.13)	65.71(5.86)	80.52(3.24)
1	Semi-DA	96.43(0.85)	76.67(3.24)	86.55(1.86)	-	-	-
	DA	**96.80(0.81)**	**77.63(3.22)**	**87.21(1.84)**	95.76(1.01)	70.77(5.68)	83.27(3.14)
5	Mono	96.51(1.69)	78.95(3.91)	87.73(2.37)	95.60(1.08)	68.13(5.98)	81.87(3.31)
	Semi-DA	96.97(1.26)	79.73(3.84)	88.35(2.22)	**96.38(0.81)**	73.48(5.26)	84.93(2.89)
	DA	**97.49(0.67)**	**80.35(3.64)**	**88.92(2.01)**	96.35(0.82)	**73.67(5.22)**	**85.01(2.86)**
10	Mono	97.29(1.03)	80.59(3.67)	88.94(2.07)	95.77(1.02)	69.45(5.93)	82.61(3.27)
	Semi-DA	97.60(0.76)	**81.21(3.58)**	89.40(1.99)	**96.66(0.72)**	74.67(5.01)	**85.66(2.73)**
	DA	**97.70(0.65)**	81.19(3.47)	**89.45(1.91)**	96.62(0.75)	**74.69(5.03)**	85.66(2.75)
200	Mono	98.24(0.34)	83.54(2.93)	90.89(1.56)	96.98(0.56)	77.33(4.34)	87.16(2.35)

Table 2.1: The performance of `DeepAtlas` on 3D knee MRI segmentation and registration. Average (standard deviation) of Dice scores (%) for bones (femur and tibia) and cartilages (femoral and tibial). N of 200 training images are manually labeled.

[2] https://nda.nih.gov/oai/

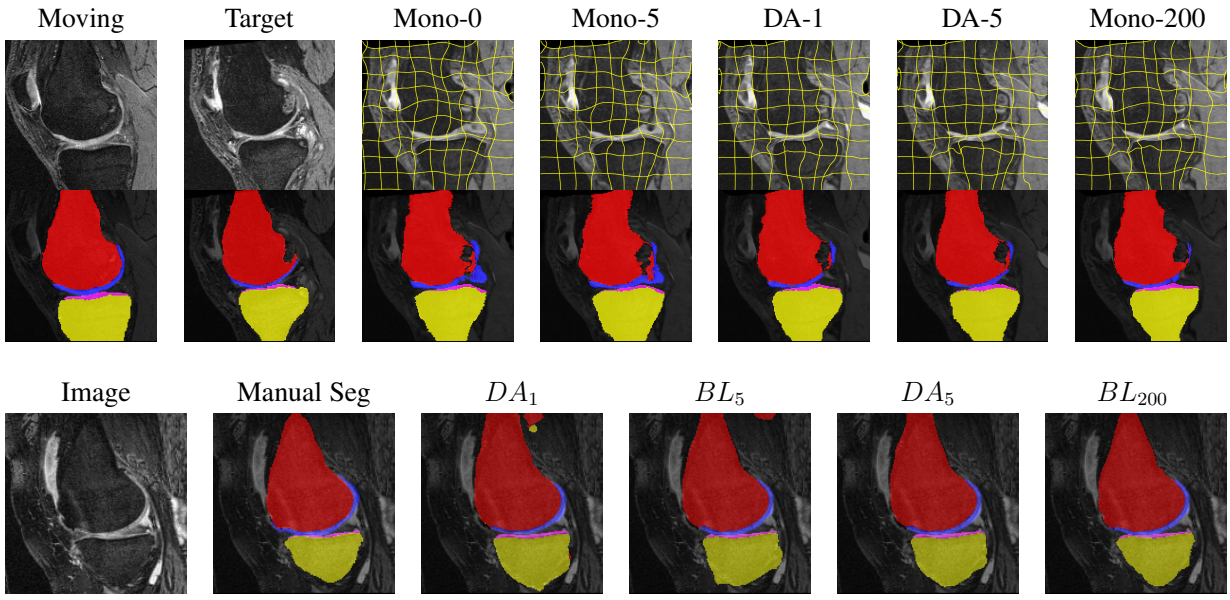


Figure 2.3: Examples of registration (top) and segmentation (bottom) results on OAI knee MRI data. **Top**: The first two columns are the moving image/segmentation and the target image/segmentation followed by the warped moving images (with deformation grids)/segmentations by different models. **Bottom left to right**: original image, manual segmentation and predictions of various models. Mono-i and DA-i represent the mono- and DA models with i manual segmentations respectively.

N	Models	Seg Dice (%)	Reg Dice (%)
0	Mono	-	54.75(2.37)
1	Semi-DA	61.19(1.49)	-
	DA	**61.22(1.40)**	56.54(2.32)
21	Mono	73.48(2.58)	59.47(2.34)
	Semi-DA	75.63(1.66)	62.92(2.14)
	DA	**76.06(1.50)**	**62.92(2.13)**
65	Mono	81.31(1.21)	63.25(2.07)

Table 2.2: The performance of DeepAtlas on 3D brain MRI segmentation and registration. Average(Standard deviation) of Dice scores (%) for 31 cortical regions. N of 65 training images are manually labeled.

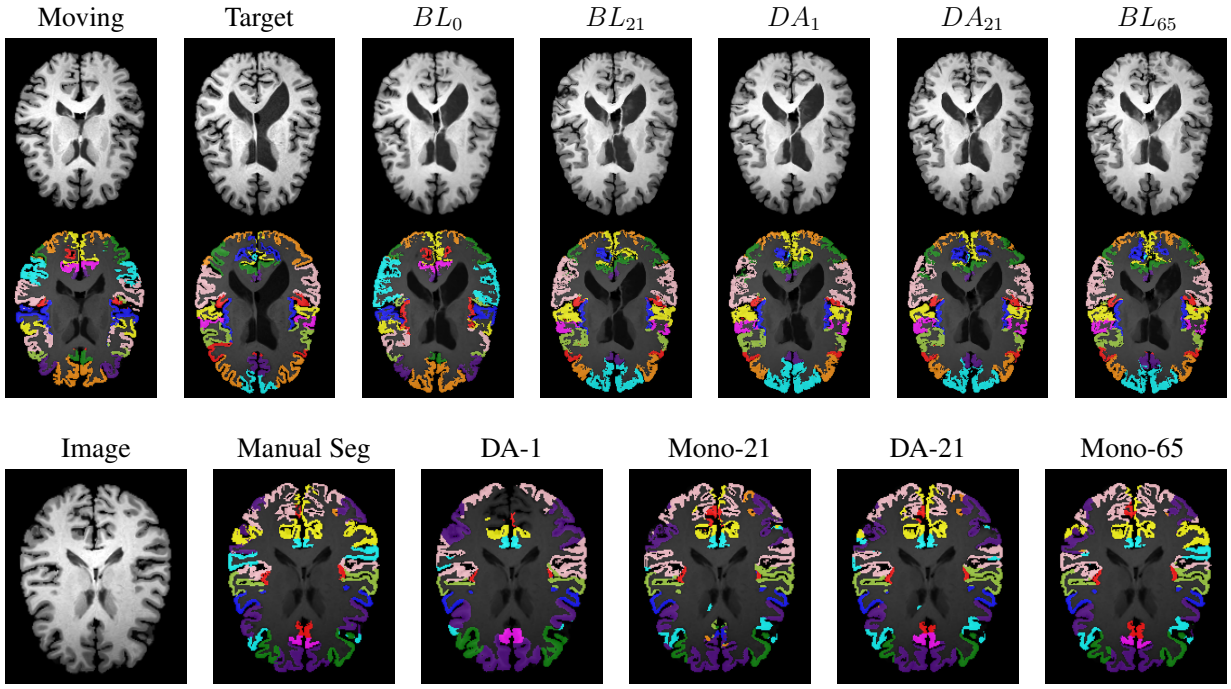


Figure 2.4: Examples of registration (top) and segmentation (bottom) results on OAI knee MRI data. **Top**: The first two columns are the moving image/segmentation and the target image/segmentation followed by the warped moving images (with deformation grids)/segmentations by different models. **Bottom left to right**: original image, manual segmentation and predictions of various models. Mono-i and DA-i represent the mono- and DA models with i manual segmentations respectively.

Brain MRI experiment: We also evaluate our method on the MindBooggle101 (Klein and Tourville, 2012) brain MRIs with 32 cortical regions. We fuse corresponding segmentation labels of the left and right brain hemispheres. MindBoogle101 consists of images from multiple datasets, *e.g.*, OASIS-TRT-20, MMRR-21 and HLN-12. After removing images with incorrect labels, we obtain a total of 85 images. We use 5 images from OASIS-TRT-20 as validation set and 15 as test set. We use the remaining 65 images for training. Manual segmentations in the N=1 and N=21 experiments are only from the MMRR-21 subset; this simulates a common practical use case, where we only have few manual segmentations for one dataset and additional unlabeled images from other datasets, but desire to process a different, new dataset. All images are 1mm isotropic, affinely-aligned, histogram-matched, and cropped to size $168 \times 200 \times 169$. We apply sagittal flipping for training data augmentation. We use the same loss weights as for the knee MRI experiment except for $\lambda_r = 5,000$, since cross-subject brain registrations require large deformations and hence less regularization.

2.5 Results

All trained networks are evaluated using Dice overlap scores between predictions and the manual segmentations for the segmentation network, or between the warped moving segmentations and the target segmentations for the registration network. Tabs. 2.1 and 2.2 show results for the knee and brain MRI experiments respectively in Dice scores (%). Fig. 2.3 and Fig. 2.4 shows examples of registration and segmentation results on knee and brain MRI.

General results: For both datasets across different numbers of manual segmentations, Semi-DA, which uses a fixed pre-trained network to help the training of the other network, boosts performance compared to separately trained mono-networks. DA, where both networks are alternately trained, achieves even better Dice scores in most cases. Based on a Mann-Whitney U-test with a significance level of 0.05 and a correction for multiple comparisons with a false discovery rate of 0.05, our models (DA/Semi-DA) result in significantly larger Dice scores than the mono-networks for all experiments. This demonstrates that segmentation and registration networks can indeed help each other by providing estimated supervision on unlabeled data.

Knee results: On knee MRIs, our method improves segmentation scores over separately learned networks by about 1.2% and 0.5%, and registration scores increase by about 3.1% and 3.0%, when training with 5 and 10 manual segmentation respectively. Especially for the challenging cartilage structures, our joint learning boosts segmentation by 1.4% and 0.7%, and registration by 5.5% and 5.2% for N=5 and N=10 respectively.

Brain results: Dice scores for segmentation and registration increase by about 2.6% and 3.5% respectively for the cortical structures of the brain MRIs.

One-shot learning: In the one-shot experiments on both datasets, reasonable segmentation performance is achieved; moreover, DA increases the Dice score over unsupervised registration by about 2.7% and 1.8% on the knee and brain data respectively. This demonstrates the effectiveness of our framework for one-shot learning.

Qualitative results: DA achieves more anatomically consistent registrations than the mono-networks on the knee (Fig. 2.3) and Brain MRI samples (Fig. 2.4).

2.6 Conclusion

We presented our DeepAtlas framework for joint learning of segmentation and registration networks using only few images with manual segmentations. By introducing an anatomical similarity loss, the learned registrations are more anatomically consistent. Furthermore, the segmentation network is guided by a form of data augmentation provided via the registration network on unlabeled images. For both bone/cartilage structures in knee MRIs and cortical structures in brain MRIs, our approach shows large improvements over separately learned networks. When only given one manual segmentation, our method provides one-shot segmentation learning and greatly improves registration. This demonstrates that one network can benefit from imperfect supervision on unlabeled data provided by the other network. Our approach provides a general solution to the lack of manual segmentations when training segmentation and registration networks. For future work, introducing uncertainty measures for the segmentation and registration networks may help alleviate the effect of poor predictions of one network on the other. It would also be of interest to investigate multitask

learning via layer sharing for the segmentation and registration networks. This may further improve performance and decrease model size.

CHAPTER 3: RandConv: Robust and Generalizable Visual Representation Learning via Random Convolutions

While successful for various computer vision tasks, deep neural networks have shown to be vulnerable to texture style shifts and small perturbations to which humans are robust. In this chapter, we propose to use random convolutions as data augmentation to improve the robustness of neural networks in the presence of distribution shift with respect to image "style". Random convolutions are approximately shape-preserving and may distort colors and local textures. Intuitively, randomized convolutions create an infinite number of new domains with similar global shapes but random local texture. Therefore, we explore using outputs of multi-scale random convolutions as new images or mixing them with the original images during training. When applying a network trained with our approach to unseen domains, our method consistently improves the performance on domain generalization benchmarks and is scalable to ImageNet. In particular, in the challenging scenario of generalizing to the sketch domain in PACS and to ImageNet-Sketch, our method outperforms state-of-art methods by a large margin. More interestingly, our method can benefit downstream tasks by providing a more robust pretrained visual representation.

This chapter is organized as follows: Section 3.1 introduces the motivation and contribution of our work and Section 3.2 introduces related previous works. Section 3.3.2 describes the details of proposed `RandConv`data augmentation pipeline. Section 3.4 presents the experimental setup and results on domain generalization and robustness benchmarks. Section 3.6 concludes this chapter. Section 3.5 provides further details and additional experiments. The work presented in this chapter was published in the Proceedings of the 2021 International Conference on Learning Representations (ICLR 2021) (Xu et al., 2021).

3.1 Introduction

Generalizability and robustness to out-of-distribution samples have been major pain points when applying deep neural networks (DNNs) in real world applications (Volpi et al., 2018). Though DNNs are typically trained on datasets with millions of training samples, they still lack robustness to domain shift, small perturbations, and adversarial examples (Luo et al., 2019). Recent research has shown that neural networks tend to use superficial features rather than global shape information for prediction even when trained on large-scale datasets such as ImageNet (Geirhos et al., 2019). These superficial features can be local textures or even patterns imperceptible to humans but detectable to DNNs, as is the case for adversarial examples (Ilyas et al., 2019). In contrast, image semantics often depend more on object shapes rather than local textures. For image data, local texture differences are one of the main sources of domain shift, e.g., between synthetic virtual images and real data (Sun and Saenko, 2014). Our goal is therefore to learn visual representations that are invariant to local texture and that generalize to unseen domains. While texture and color may be treated as different concepts, we follow the convention in (Geirhos et al., 2019) and include color when talking about texture.

We address the challenging setting of robust visual representation learning from *single domain data*. Limited work exists in this setting. Proposed methods include data augmentation (Volpi et al., 2018; Qiao et al., 2020; Geirhos et al., 2019), domain randomization (Tobin et al., 2017; Yue et al., 2019), self-supervised learning (Carlucci et al., 2019), and penalizing the predictive power of low-level network features (Wang et al., 2019a). Following the spirit of adding inductive bias towards global shape information over local textures, we propose using random convolutions to improve the robustness to domain shifts and small perturbations. While recently Lee et al. (2020) proposed a similar technique for improving the generalization of reinforcement learning agents in unseen environments, we focus on visual representation learning and examine our approach on visual domain generalization benchmarks. Our method also includes a multiscale design and a mixing variant. In addition, considering that many computer vision tasks rely on training deep networks based on ImageNet-pretrained weights (including some domain generalization benchmarks), we

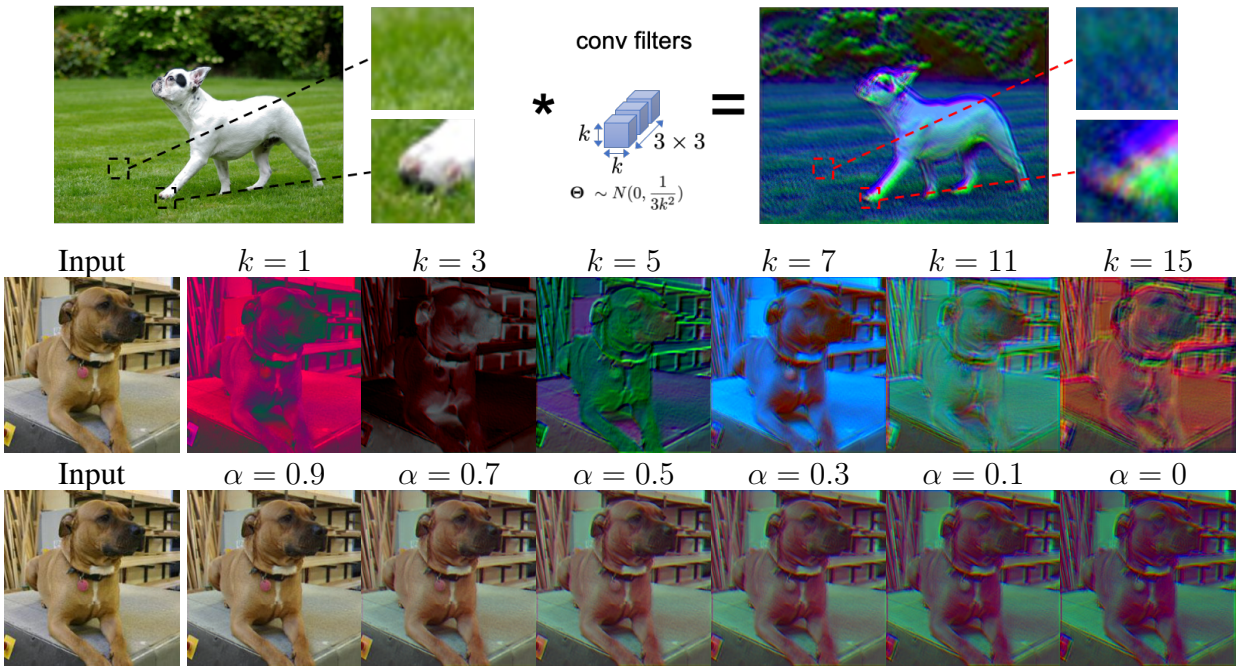


Figure 3.1: **Top**: Illustration that `RandConv` randomizes local texture but preserves shapes in the image. **Middle:** First column is the input image of size 224^2; following columns are convolutions results using random filters of different sizes k. **Bottom:** Mixing results between an image and one of its random convolution results with different mixing coefficients α.

ask *"Can a more robust pretrained model make the finetuned model more robust on downstream tasks?"* Different from (Kornblith et al., 2019; Salman et al., 2020) who studied the transferability of a pretrained ImageNet representation to new tasks while focusing on in-domain generalization, we explore generalization performance on *unseen domains* for new tasks.

We make the following contributions:

- We develop `RandConv`, a data augmentation technique *using multi-scale random-convolutions to generate images with random texture while preserving global shapes.* We explore using the `RandConv` output as training images or mixing it with the original images. We show that a consistency loss can further enforce invariance under texture changes.
- We provide insights and justifications on why `RandConv` augments images with different local texture but the same semantics with the shape-preserving property of random convolutions.

- We validate `RandConv` and its mixing variant in extensive experiments on synthetic and real-world benchmarks as well as on the large-scale ImageNet dataset. Our methods outperform single domain generalization approaches by a large margin on digit recognition datasets and for the challenging case of generalizing to the Sketch domain in PACS and to ImageNet-Sketch.
- We explore if the robustness/generalizability of a pretrained representation can transfer. We show that transferring a model pretrained with `RandConv` on ImageNet can further improve domain generalization performance on new downstream tasks on the PACS dataset.

3.2 Related Work

Domain Generalization (DG) aims at learning representations that perform well when transferred to unseen domains. Modern techniques range between feature fusion (Shen et al., 2019), meta-learning (Li et al., 2018a; Balaji et al., 2018), and adversarial training (Shao et al., 2019; Li et al., 2018b). Note that most current DG work (Ghifary et al., 2016; Li et al., 2018a,b) requires a multi-source training setting to work well. However, in practice, it might be difficult and expensive to collect data from multiple sources, such as collecting data from multiple medical centers (Raghupathi and Raghupathi, 2014). Instead, we consider the more strict single-domain generalization DG setting, where we train the model on source data from a single domain and generalize it to new unseen domains (Carlucci et al., 2019; Wang et al., 2019b).

Domain Randomization (DR) was first introduced as a DG technique by Tobin et al. (2017) to handle the domain gap between simulated and real data. As the training data in (Tobin et al., 2017) is synthesized in a virtual environment, it is possible to generate diverse training samples by randomly selecting background images, colors, lighting, and textures of foreground objects. When a simulation environment is not accessible, image stylization can be used to generate new domains (Yue et al., 2019; Geirhos et al., 2019). However, this requires extra effort to collect data and to train an additional model; further, the number of randomized domains is limited by the number of predefined styles.

Data Augmentation has been widely used to improve the generalization of machine learning models (Simard et al., 2003). DR approaches can be considered a type of synthetic data augmentation. To improve performance on unseen domains, Volpi et al. (2018) generate adversarial examples to augment the training data; Qiao et al. (2020) extend this approach via meta-learning. As with other adversarial training algorithms, significant extra computation is required to obtain adversarial examples.

Learning Representations Biased towards Global Shape Geirhos et al. (2019) demonstrated that convolutional neural networks (CNNs) tend to use superficial local features even when trained on large datasets. To counteract this effect, they proposed to train on stylized ImageNet, thereby forcing a network to rely on object shape instead of textures. Wang et al. improved out-of-domain performance by penalizing the correlation between a learned representation and superficial features such as the gray-level co-occurrence matrix (Wang et al., 2019b), or by penalizing the predictive power of local, low-level layer features in a neural network via an adversarial classifier (Wang et al., 2019a). Our approach shares the idea that learning representations invariant to local texture helps generalization to unseen domains. However, `RandConv` avoids searching over many hyper-parameters, collecting extra data, and training other networks. It also scales to large-scale datasets since it adds minimal computation overhead.

Random Mapping in Machine Learning Random projections have also been effective for dimensionality reduction based on the distance-preserving property of the Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984). (Vinh et al., 2016) applied random projections on entire images as data augmentation to make neural networks robust to adversarial examples. Lee et al. (2020) recently used random convolutions to help reinforcement learning (RL) agents generalize to new environments. Neural networks with *fixed* random weights can encode meaningful representations (Saxe et al., 2011) and are therefore useful for neural architecture search (Gaier and Ha, 2019), generative models (He et al., 2016a), natural language processing (Wieting and Kiela, 2019), and RL (Osband et al., 2018; Burda et al., 2019). In contrast, `RandConv` uses *non-fixed* randomly-sampled weights to generate images with different local texture.

3.3 RandConv: Randomize Local Texture at Different Scales

We propose using a convolution layer with non-fixed random weights as the first layer of a DNN during training. This strategy generates images with random local texture but consistent shapes, and is beneficial for robust visual representation learning. Sec. 3.3.1 justifies the shape-preserving property of a random convolution layer. Sec. 3.3.2 describes `RandConv`, our data augmentation algorithm using a multi-scale randomized convolution layer and input mixing.

3.3.1 A Random Convolution Layer Preserves Global Shapes

Convolution is the key building block for deep convolutional neural networks. Consider a convolution layer with filters $\mathbf{\Theta} \in \mathbb{R}^{h \times w \times C_{in} \times C_{out}}$ with an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times C_{in}}$, where H and W are the height and width of the input and C_{in} and C_{out} are the number of feature channels for the input and output, and h and w are the height and width of the layer's filter. The output (with appropriate input padding) will be $\mathbf{g} = \mathbf{I} * \mathbf{\Theta}$ with $\mathbf{g} \in \mathbb{R}^{H \times W \times C_{out}}$.

In images, nearby pixels with similar color or texture can be grouped into primitive shapes that represent parts of objects or the background. A convolution layer linearly projects local image patches to features at corresponding locations on the output map using shared parameters. While a convolution with random filters can project local patches to arbitrary output features, the output of a random linear projection approximately preserves relative similarity between input patches, proved in Appendix 3.5.2. In other words, since any two locations within the same shape have similar local textures in the input image, they tend to be similar in the output feature map. Therefore, shapes that emerge in the output feature map are similar to shapes in the input image provided that the filter size is sufficiently small compared to the size of a typical shape.

In other words, the size of a convolution filter determines the smallest shape it can preserve. For example, 1x1 random convolutions preserve shapes at the single-pixel level and thus work as a random color mapping; large filters perturb shapes smaller than the filter size that are considered local texture of a shape at this larger scale. See Fig. 3.1 for examples. *More discussion and a formal proof are in Appendix 3.5.1 and 3.5.2*.

3.3.2 Multi-scale Image Augmentation with a Randomized Convolution Layer

Algorithm 1 Learning with Data Augmentation by Random Convolutions

Input: Model Φ, task loss $\mathcal{L}_{task}$, training images $\{I_i\}_{i=1}^N$ and their labels $\{y_i\}_{i=1}^N$, pool of filter sizes $\mathcal{K} = \{1, ..., n\}$, fraction of original data p, whether to `mix` with original images, consistency loss weight λ
function RANDCONV(I, $\mathcal{K}$, `mix`, p)
    Sample $p_0 \sim U(0, 1)$
    if p_0 ¡ p and `mix` is False **then**
        return I ▷ When not in `mix` mode, use the original image with probability p
    else
        Sample scale $k \sim \mathcal{K}$
        Sample convolution weights $\mathbf{\Theta} \in \mathbb{R}^{k \times k \times 3 \times 3} \sim N(0, \frac{1}{3k^2})$
        $I_{rc} = I * \mathbf{\Theta}$ ▷ Apply convolution on I
        if `mix` is True **then**
            Sample $\alpha \sim U(0, 1)$
            return $\alpha I + (1 - \alpha) I_{rc}$ ▷ Mix with original images
        else
            return I_{rc}
Learning Objective:
for $i = 1 \rightarrow N$ **do**
    for $j = 1 \rightarrow 3$ **do**
        $\hat{y}_i^j = \Phi(\texttt{RandConv}(I_i))$ ▷ Predict labels for three augmented variants of the same image
    $\mathcal{L}_{cons} = \lambda \sum_{j=1}^3 \text{KL}(\hat{y}_i^j || \bar{y}_i)$ where $\bar{y}_i = \sum_{j=1}^3 \hat{y}_i^j / 3$ ▷ Consistency Loss
    $\mathcal{L} = \mathcal{L}_{task}(\hat{y}_i^1, y_i) + \lambda \mathcal{L}_{cons}$ ▷ Learning with the task loss and the consistency loss

Sec. 3.3.1 discussed how outputs of randomized convolution layers approximately maintain shape information at a scale larger than their filter sizes. Here, we develop our `RandConv` data augmentation technique using a randomized convolution layer with $C_{out} = C_{in}$ to generate shape-consistent images with randomized texture (see Alg. 1). Our goal is not to use `RandConv` to parameterize or represent texture as in previous filter-bank based texture models (Heeger and Bergen, 1995; Portilla and Simoncelli, 2000). Instead, we only use the three-channel outputs of `RandConv` as new images with the same shape and different "style" (loosely referred to as "texture"). We also note that a convolution layer is different from a convolution operation in image filtering. Standard image filtering applies the same 2D filter on three color channels separately. In

contrast, our convolution layer applies three different *3D* filters and each takes all color channels as input and generates one channel of the output. Our proposed `RandConv` variants are as follows:

RC_{img}: Augmenting Images with Random Texture A simple approach is to use the randomized convolution layer outputs, $I * \mathbf{\Theta}$, as new images; where $\mathbf{\Theta}$ are the randomly sampled weights and I is a training image. If the original training data is in the domain D^0, a sampled weight $\mathbf{\Theta}_k$ generates images with consistent global shape but random texture forming the random domain D^k. Thus, by random weight sampling, we obtain an infinite number of random domains $D^1, D^1, \ldots, D^\infty$. Input image intensities are assumed to be a standard normal distribution $N(0, 1)$ (which is often true in practice thanks to data whitening). As the outputs of `RandConv` should follow the same distribution, we sample the convolution weights from $N(0, \sigma^2)$ where $\sigma = 1/\sqrt{C_{in} \times h \times w}$, which is commonly applied for network initialization (He et al., 2015). We include the original images for training at a ratio p as a hyperparameter.

RC_{mix}: Mixing Variant As shown in Fig. 3.1, outputs from RC_{img} can vary significantly from the appearance of the original images. Although generalizing to domains with significantly different local texture distributions is useful, we may not want to sacrifice much performance on domains similar to the training domain. Inspired by the AugMix (Hendrycks et al., 2020b) strategy, we propose to blend the original image with the outputs of the `RandConv` layer via linear convex combinations $\alpha I + (1 - \alpha)(I * \mathbf{\Theta})$, where α is the mixing weight uniformly sampled from $[0, 1]$.In RC_{mix}, the `RandConv` outputs provide shape-consistent perturbations of the original images. Varying α, we continuously interpolate between the training domain and the randomly sampled domains of $\texttt{RC}_{\texttt{img}}$.

Multi-scale Texture Corruption As discussed in Sec. 3.3.1, image shape information at a scale smaller than a filter's size will be corrupted by `RandConv`. Therefore, we can use filters of varying sizes to preserve shapes at various scales. We choose to uniformly randomly sample a filter size k from a pool $\mathcal{K} = 1, 3, ...n$ before sampling convolution weights $\mathbf{\Theta} \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$ from a Gaussian distribution $N(0, \frac{1}{k^2 C_{in}})$. Fig. 3.1 shows examples of multi-scale `RandConv` outputs.

Consistency Regularization To learn representations invariant to texture changes, we use a loss encouraging consistent network predictions for the same `RandConv`-augmented image for different random filter samples. Approaches for transform-invariant domain randomization (Yue et al., 2019), data augmentation (Hendrycks et al., 2020b), and semi-supervised learning (Berthelot et al., 2019) use similar strategies. We use Kullback-Leibler (KL) divergence to measure consistency. However, enforcing prediction similarity of two augmented variants may be too strong. Instead, following (Hendrycks et al., 2020b), we use `RandConv` to obtain 3 augmentation samples of image I: $G_j = \texttt{RandConv}^j(I)$ for $j = 1, 2, 3$ and obtain their predictions with a model Φ: $y^j = \Phi(G^j)$. We then compute the *relaxed* loss as $\lambda \sum_{j=1}^{3} \text{KL}(y^j || \bar{y})$, where $\bar{y} = \sum_{j=1}^{3} y^j/3$ is the sample average.

3.4 Experiments

Secs. 3.4.1 to 3.4.3 evaluate our methods on the following datasets: multiple digit recognition datasets, PACS, and ImageNet-sketch. Sec. 3.4.4 uses PACS to explore the out-of-domain generalization of a pretrained representation in transfer learning by checking if pretraining on ImageNet with our method improves the domain generalization performance in downstream tasks. All experiments are in the single-domain generalization setting where training and validation sets are drawn from one domain. *Additional experiments with ResNet18 as the backbone are given in the Appendix*.

3.4.1 Digit Recognition

The five digit recognition datasets (MNIST (LeCun et al., 1998), MNIST-M (Ganin et al., 2016), SVHN (Netzer et al., 2011), SYNTH (Ganin and Lempitsky, 2014) and USPS (Denker et al., 1989)) have been widely used for domain adaptation and generalization research (Peng et al., 2019a,b; Qiao et al., 2020). Following the setups in (Volpi et al., 2018) and (Qiao et al., 2020), we train a simple CNN with *10,000* MNIST samples and evaluate the accuracy on the test sets of the other four datasets. We also test on MNIST-C (Mu and Gilmer, 2019), a robustness benchmark with *15 common corruptions* of MNIST and report the average accuracy over all corruptions.

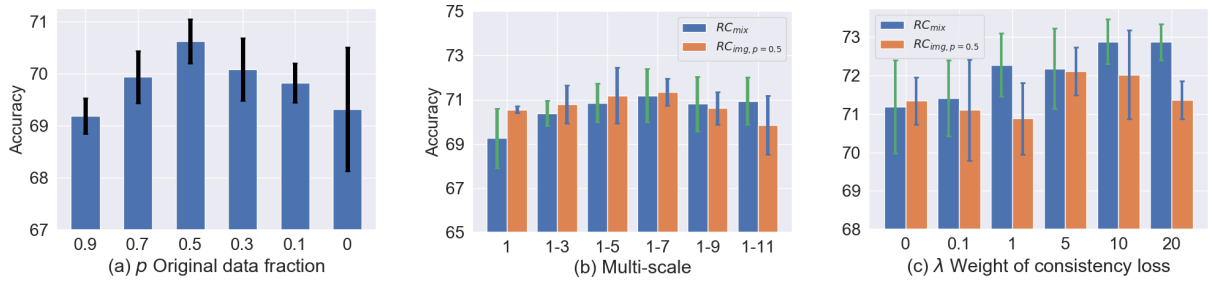


Figure 3.2: Average accuracy and 5-run variance of MNIST model on MNIST-M, SVHN, SYNTH and USPS. Studies for: (a) original data fraction p for RC_{img}; (b) multiscale design (1-n refers to using scales 1,3,..,n) for $\text{RC}_{\text{img},p=0.5}$ (orange) and RC_{mix} (blue); (c) consistency loss weight λ for $\text{RC}_{\text{img1}-7,p=0.5}$ (orange) and $\text{RC}_{\text{mix1}-7}$ (blue).

Selecting Hyperparameters and Ablation Study. Fig. 3.2(a) shows the effect of the hyperparameter p on RC_{img} with filter size 1. We see that adding only 10% `RandConv` data ($p = 0.9$) immediately improves the average performance (DG-Avg) on MNIST-M, SVHN, SYNTH and USPS performance from 53.53 to 69.19, outperforming all other approaches (see Tab. 3.1) for every dataset. We choose $p = 0.5$, which obtains the best DG-Avg. Fig. 3.2(b) shows results for a multiscale ablation study. Increasing the pool of filter sizes up to 7 improves DG-Avg performance. Therefore we use multi-scale 1-7 to study the consistency loss weight λ, shown in Fig. 3.2(c). Adding the consistency loss improves both `RandConv` variants on DG-avg: $\text{RC}_{\text{mix1}-7}$ favors $\lambda = 10$ while $\text{RC}_{\text{img1}-7,p=0.5}$ performs similarly for $\lambda = 5$ and $\lambda = 10$. We choose $\lambda = 10$ for all subsequent experiments.

Results. Tab. 3.1 compares the performance of $\text{RC}_{\text{img1}-7,p=0.5,\lambda=10}$ and $\text{RC}_{\text{mix1}-7,\lambda=10}$ with other state-of-the-art approaches. We show results of the adversarial training based methods GUD (Volpi et al., 2018), M-ADA (Qiao et al., 2020), and PAR (Wang et al., 2019a). The baseline model is trained only on the standard classification loss.

3.4.2 PACS Experiments

The PACS dataset (Li et al., 2018b) considers 7-class classification on 4 domains: photo, art painting, cartoon, and sketch, with very different texture styles. Most recent domain generalization work studies the multi-source domain setting on PACS and uses domain labels of the training data.

	MNIST	MNIST-M	SVHN	USPS	SYNTH	DG-Avg	MNIST-C
Baseline	98.40(0.84)	58.87(3.73)	33.41(5.28)	79.27(2.70)	42.43(5.46)	53.50(4.23)	88.20(2.10)
GreyScale	98.82(0.02)	58.41(0.99)	36.06(1.48)	80.45(1.00)	45.00(0.80)	54.98(0.86)	89.15(0.44)
ColorJitter	98.72(0.05)	62.72(0.66)	39.61(0.88)	79.18(0.60)	46.40(0.34)	56.98(0.39)	89.48(0.18)
BandPass	98.65(0.11)	70.22(2.73)	48.34(2.56)	78.60(0.82)	57.17(2.01)	63.58(1.89)	87.89(0.68)
MultiAug	98.80(0.05)	62.32(0.66)	39.07(0.68)	79.31(1.02)	46.48(0.80)	56.79(0.34)	89.54(0.11)
PAR (our imp)	98.79(0.05)	61.16(0.21)	36.08(1.27)	79.95(1.18)	45.48(0.35)	55.67(0.33)	89.34(0.45)
GUD	-	60.41	35.51	77.26	45.32	54.62	-
M-ADA	-	67.94	42.55	78.53	48.95	59.49	-
$RC_{img1\text{-}7,\ p=0.5,\ \lambda=5}$	98.86(0.05)	87.67(0.37)	54.95(1.90)	82.08(1.46)	63.37(1.58)	72.02(1.15)	90.94(0.51)
$RC_{mix1\text{-}7,\lambda=10}$	98.85(0.04)	87.76(0.83)	57.52(2.09)	83.36(0.96)	62.88(0.78)	72.88(0.58)	**91.62(0.77)**
$RC_{mix1\text{-}7,\lambda=10}$ + MultiAug	98.82(0.06)	**87.89(0.29)**	**62.07(0.62)**	**84.39(1.02)**	**63.90(0.63)**	**74.56(0.46)**	91.40(0.93)

Table 3.1: Average accuracy and 5-run standard deviation (in parenthesis) of MNIST10K model on MNIST-M, SVHN, SYNTH, USPS and their average (DG-avg); and average accuracy of 15 types of corruptions in MNIST-C. Both `RandConv` variants significantly outperform all other methods.

Although we follow the convention to train on 3 domains and to test on the fourth, we simply pool the data from the 3 training domains as in (Wang et al., 2019a), without using domain labels during the training.

Baseline and State-of-the-Art. Following (Li et al., 2017), we use Deep-All as the baseline, which finetunes an ImageNet-pretrained AlexNet on 3 domains using only the classification loss and tests on the fourth domain. We test our `RandConv` variants $RC_{img1\text{-}7,p=0.5}$ and $RC_{mix1\text{-}7}$ with and without consistency loss, and ColorJitter/GreyScale/BandPass/MultiAug data augmentation as in the digit datasets. We also implemented PAR (Wang et al., 2019a) using our baseline model. $RC_{mix1\text{-}7}$ combined with MultiAug is also tested. Further, we compare to the following state-of-the-art approaches: Jigen (Carlucci et al., 2019) using self-supervision, MLDG (Li et al., 2018a) using meta-learning, and the conditional invariant deep domain generalization method CIDDG (Li et al., 2018c). Note that previous methods used different Deep-All baselines which make the final accuracy not directly comparable, and MLDG and CIDDG use domain labels for training.

Results. Tab. 3.2 shows *significant improvements on Sketch* for both `RandConv` variants. Sketch is the most challenging domain with no color and much less texture compared to the other 3 domains. The success on Sketch demonstrates that our methods can guide the DNN to learn

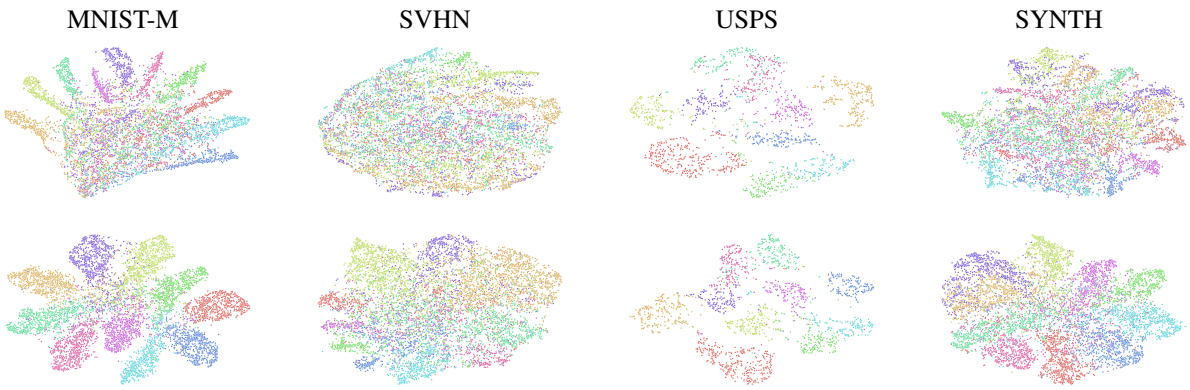


Figure 3.3: t-SNE feature embedding visualization for digit datasets for models trained on MNIST without (top) and with our $\text{RC}_{\text{mix1-7},\lambda=10}$ approach (bottom). Different colors denote different classes.

global representations focusing on shapes that are robust to texture changes. Without using the consistency loss, $\text{RC}_{\text{mix1-7}}$ achieves the best overall result improving over Deep-All by $\sim$4% but adding MultiAug does not further improve the performance. Adding the consistency loss with $\lambda = 10$, $\text{RC}_{\text{mix1-7}}$ and $\text{RC}_{\text{img1-7},p=0.5}$ performs better on Sketch but degrades performance on the other 3 domains, so do GreyScale and ColorJitter. *This observation will be discussed in Sec 3.4.4.*

3.4.3 Generalizing an ImageNet Model to ImageNet-Sketch

ImageNet-Sketch (Wang et al., 2019a) is an out-of-domain test set for models trained on ImageNet. We trained AlexNet from scratch with $\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$ and $\text{RC}_{\text{mix1-7},\lambda=10}$. We evaluate their performance on ImageNet-Sketch. We use the AlexNet model trained without `RandConv` as our baseline. Tab. 3.3 compares `PAR` and its baseline model and AlexNet trained with Stylized ImageNet (SIN) (Geirhos et al., 2019) on ImageNet-Sketch. Although `PAR` uses a stronger baseline, `RandConv` achieves significant improvements over our baseline and outperforms `PAR` by a large margin. Our methods achieve more than a 7% accuracy improvement over the baseline and surpass PAR by 5%. SIN as an image stylization approach that can modify image texture in a hierarchical and realistic way. However, albeit its complexity, it still performs on par with RandConv. Note that image stylization techniques require additional data and heavy precomputation. Further, the images for the style source also need to be chosen. In contrast, RandConv is much easier to use:

Base	Method	Photo	Art	Cartoon	Sketch	Average
Ours	Deep-All	86.77(0.42)	60.11(1.33)	64.12(0.32)	55.28(4.71)	66.57(1.36)
	GreyScale	83.93(1.47)	61.60(1.18)	62.12(0.61)	60.07(2.47)	66.93(0.83)
	ColorJitter	84.61(0.83)	59.01(0.24)	61.43(0.68)	62.44(1.68)	66.88(0.33)
	BandPass	87.08(0.57)	59.46(0.27)	**64.39(0.51)**	55.39(2.95)	66.58(0.73)
	MultiAug	85.21(0.47)	59.51(0.38)	62.88(1.01)	61.67(0.76)	67.32(0.23)
	PAR (our imp.)	**87.21(0.42)**	60.17(0.95)	63.63(0.88)	55.83(2.57)	66.71(0.58)
	$RC_{img1\text{-}7,\ p=0.5}$	86.50(0.72)	61.10(0.38)	64.24(0.62)	68.50(1.83)	70.09(0.43)
	$RC_{mix1\text{-}7}$	86.60(0.67)	61.74(0.90)	64.05(0.66)	69.74(0.66)	**70.53(0.25)**
	$RC_{mix1\text{-}7}$ + MultiAug	86.23(0.74)	**61.91(0.76)**	62.69(0.76)	67.74(1.21)	69.64(0.49)
	$RC_{img1\text{-}7,\ p=0.5,\ \lambda=10}$	81.15(0.76)	59.56(0.79)	62.42(0.59)	71.74(0.43)	68.72(0.58)
	$RC_{mix1\text{-}7,\lambda=10}$	81.78(1.11)	61.14(0.51)	63.57(0.29)	**71.97(0.38)**	69.62(0.24)
Results below are not directly comparable due to different Deep-All implementations.						
Wang et al. (2019a)	Deep-All (our run)	88.40	66.26	66.58	59.40	70.16
	PAR (our run)	88.40	65.19	68.58	61.86	71.10
	PAR (reported)	89.6	66.3	68.3	64.1	72.08
Carlucci et al. (2019)	Deep-All	89.98	66.68	69.41	60.02	71.52
	Jigen	89.00	67.63	71.71	65.18	73.38
Li et al. (2018a)	Deep-All	86.67	64.91	64.28	53.08	67.24
	MLDG (use domain labels)	88.00	66.23	66.88	58.96	70.01
Li et al. (2018c)	Deep-All	77.98	57.55	67.04	58.52	65.27
	CIDDG (use domain labels)	78.65	62.70	69.73	64.45	68.88

Table 3.2: Mean and 5-run standard deviation (in parenthesis) results for domain generalization on PACS. Best results with our Deep-All baseline are in **bold**. The domain name in each column represents the target domain. Base column indicates different baselines and results under different baselines are not directly comparable. MLDG and CIDDF used domain labels for training.

it can be applied to any dataset via a simple convolution layer. We also measure the shape-bias metric proposed by (Geirhos et al., 2019) for the `RandConv` trained AlexNet. $RC_{img1\text{-}7,p=0.5,\lambda=10}$ and $RC_{mix1\text{-}7,\lambda=10}$ improve the baseline from 25.36% to 48.24% and 54.85% respectively.

3.4.4 Revisiting PACS with more Robust Pretrained Representations

A common practice for many computer vision tasks (including the PACS benchmark) is transfer learning, i.e. finetuning a backbone model pretrained on ImageNet. Recently, how the accuracy on ImageNet (Kornblith et al., 2019) and adversial robustness (Salman et al., 2020) of the pretrained model affect transfer learning has been studied in the context of domain generalization. Instead, we

	Baseline (Wang et al., 2019a)	PAR (Wang et al., 2019a)	Baseline	$\text{RC}_{\text{img1-7,}}$ p=0.5,λ=10	$\text{RC}_{\text{mix1-7,}}$ λ=10	SIN (Geirhos et al., 2019)
Top1	12.04	13.06	10.28	18.09	16.91	17.62
Top5	25.60	26.27	21.60	35.40	33.99	36.22

Table 3.3: Accuracy of ImageNet-trained AlexNet on ImageNet-Sketch (IN-S) data. Our methods outperform PAR by 5% and are on par with a Stylized-ImageNet (SIN) trained model. Note that PAR was built on top of a stronger baseline than our model, and both PAR and SIN fine-tuned the baseline model which helped the performance, while we train `RandConv` model from scratch.

study how out-of-domain generalizability transfers from pretraining to downstream tasks and shed light on how to better use pretrained models.

Impact of ImageNet Pretraining A model trained on ImageNet may be biased towards textures (Geirhos et al., 2019). Finetuning ImageNet pretrained models on PACS may inherit this texture bias, thereby benefitting generalization on the Photo domain (which is similar to ImageNet), but hurting performance on the Sketch domain. Therefore, as shown in Sec. 3.4.2, using `RandConv` to correct this texture bias improves results on Sketch, but degrades them on the Photo domain. Since pretraining has such a strong impact on transfer performance to new tasks, we ask: *"Can the generalizability of a pretrained model transfer to downstream tasks? I.e., does a pretrained model with better generalizability improve performance on unseen domains on downstream tasks?"* To answer this, we revisit the PACS tasks based on ImageNet-pretrained weights where our two `RandConv` variants of Sec. 3.4.3 are used during ImageNet training. We study if this results in performance changes for the Deep-All baseline and for finetuning with `RandConv`.

Better Performance via RandConv pretrained model We start by testing the Deep-All baselines using the two `RandConv`-trained ImageNet models of Sec. 3.4.3 as initialization. Tab. 3.4 shows significant improvements on Sketch. Results are comparable to finetuning with `RandConv` on a normal pretrained model. Art is also consistently improved. Performance drops slightly on Photo as expected, since we reduced the texture bias in the pretrained model, which is helpful for the Photo domain. A similar performance improvement is observed when using the SIN-trained AlexNet as initialization. Using `RandConv` for *both* ImageNet training and PACS finetuning, we achieve 76.11% accuracy on Sketch. As far as we know, this is the best performance using an

PACS	ImageNet	Photo	Art	Cartoon	Sketch	Avg
Deep-All	Baseline	**86.77(0.42)**	60.11(1.33)	64.12(0.32)	55.28(4.71)	66.57(1.36)
	$\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$	84.48(0.52)	62.61(1.23)	66.13(0.80)	69.24(0.80)	70.61(0.53)
	$\text{RC}_{\text{mix1-7},\lambda=10}$	85.59(0.40)	63.30(0.99)	63.83(0.85)	68.29(1.27)	70.25(0.45)
	SIN	85.33(0.66)	**65.85(0.87)**	65.39(0.62)	65.75(0.59)	70.58(0.21)
$\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$	Baseline	81.15(0.76)	59.56(0.79)	62.42(0.59)	71.74(0.43)	68.72(0.58)
	$\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$	84.36(0.36)	63.73(0.91)	**68.07(0.55)**	<u>75.41(0.57)</u>	<u>72.89(0.33)</u>
	$\text{RC}_{\text{mix1-7},\lambda=10}$	84.63(0.97)	63.41(1.22)	66.36(0.43)	74.59(0.84)	72.25(0.54)
$\text{RC}_{\text{mix1-7}\ \lambda=10}$	Baseline	81.78(1.11)	61.14(0.51)	63.57(0.29)	71.97(0.38)	69.62(0.24)
	$\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$	85.16(1.03)	63.17(0.38)	<u>67.68(0.60)</u>	**76.11(0.43)**	**73.03(0.46)**
	$\text{RC}_{\text{mix1-7},\lambda=10}$	<u>86.17(0.56)</u>	<u>65.33(1.05)</u>	65.52(1.13)	73.21(1.03)	72.56(0.50)

Table 3.4: Generalization results on PACS with `RandConv` and SIN pretrained AlexNet. ImageNet column shows how the pretrained model is trained on ImageNet (baseline represents training the ImageNet model using only the classification loss); PACS column indicates the methods used for finetuning on PACS. **Best** and <u>second best</u> accuracy for each target domain are highlighted in bold and underlined.

AlexNet baseline. This approach even outperforms Jigen (Carlucci et al., 2019) (71.35%) with a stronger ResNet18 baseline model. Cartoon and Art are also improved. The best average domain generalization accuracy is 73.03%, with a more than 6% improvement over our initial Deep-All baseline.

This experiment confirms that generalizability may transfer: removing texture bias may not only make a pretrained model more generalizable, but it may help generalization on downstream tasks. For similar target and pretraining domains like Photo and ImageNet, where learning texture bias may actually be beneficial, performance may degrade slightly.

3.5 Theoretical Justification and More Experimental Details

This section provides additional details. Specifically, in Sec. 3.5.1 and 3.5.2, we discuss definitions of shapes and textures in images and justify why random convolution preserves global shapes and disrupts local texture formally by proving Theorem 3.1. This theorem shows that random linear projections are approximately distance preserving. We also discuss our simulation-based bound based on 80% distance rescaling on real image data. Sec. 3.5.3 provides more experimental details for the different datasets. Sec. 3.5.4 shows experimental results with a stronger backbone

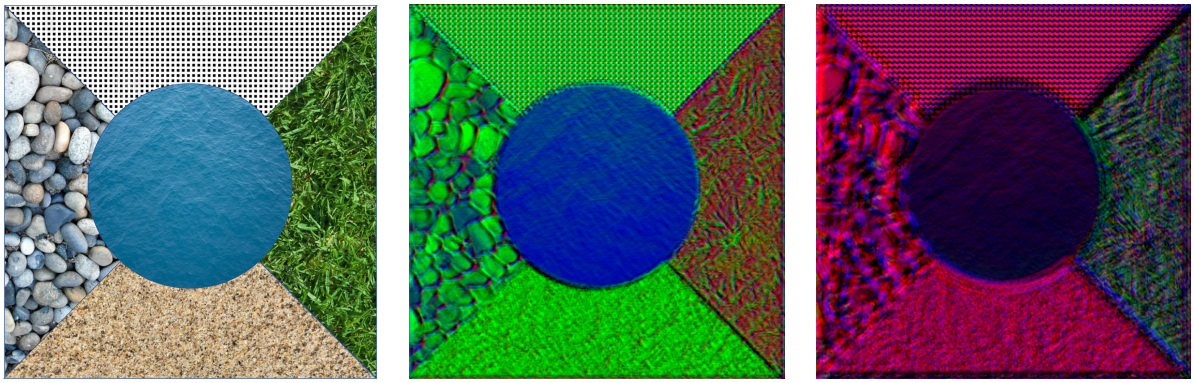


Figure 3.4: **Left:** An image with texture and shapes at different scales; **Middle:** The output of `RandConv` with a small filter size which largely preserves the shapes of the stones. **Right:** The output of `RandConv` with a large filter size distorts the shape of the stones as well.

architecture and on a new benchmark ImageNet-R (Hendrycks et al., 2020a). Sec. 3.5.5 provides more detailed results regarding hyperparameter selection and ablation studies. Lastly, Sec. 3.5.6 shows example visualizations of `RandConv` outputs and for its mixing variants.

3.5.1 Shapes and Texture in Images

As discussed in the main text, we define shapes in images that are preserved by a random convolution layer as primitive shapes: spatial clusters of pixels with similar local texture. An object in a image can be a single primitive shape alone but in most cases it is the composition of multiple primitive shapes e.g. a car includes wheels, body frames, windshields. Note that the definition of texture is not necessarily opposite to shapes, since the texture of a larger shape can includes smaller shapes. For example, in Fig.3.4, the left occluded triangle shape has texture composed by shapes of cobble stones while cobble stones have their own texture. Random convolution can preserve those large shapes that usually define the image semantics while distorting the small shapes as local texture.

To formally define the shape-preserving property, we assume (x_1, y_1), (x_2, y_2) and (x_3, y_3) are three locations on a image and (x_1, y_1) has closer color and local texture with (x_2, y_2) than (x_3, y_3). For example, (x_1, y_1) and (x_2, y_2) are within the same shape while (x_3, y_3) is located at a neighboring shape. Then we have $\|\mathbf{r}(x_1, y_1) - \mathbf{r}(x_2, y_2)\| < \|\mathbf{r}(x_1, y_1) - \mathbf{r}(x_3, y_3)\|$, where $\mathbf{r}(x_i, y_i)$

is the image patch at location (x_i, y_i). A transformation f is *shape-preserving* if it *maintains* such relative distance relations for most location triplets, i.e.

$$\|f(\mathbf{r}(x_i, y_i)) - f(\mathbf{r}(x_j, y_j))\| / \|\mathbf{r}(x_i, y_i) - \mathbf{r}(x_j, y_j)\| \approx r \tag{3.1}$$

for any two spatial location (x_i, y_i) and (x_j, y_j); $r \geq 0$ is a constant.

3.5.2 Random Convolution is Shape-preserving as a Random Linear Projection is Distance Preserving

We can express a convolution layer as a local linear projection:

$$\mathbf{g}(x, y) = \mathbf{U}\mathbf{r}(x, y)\,, \tag{3.2}$$

where $\mathbf{r}(x, y) \in \mathbb{R}^d$ ($d = h \times w \times C_{in}$) is the vectorized image patch centerized at location (x, y), $\mathbf{g}(x, y) \in \mathbb{R}^{C_{out}}$ is the output feature at location (x, y), and $\mathbf{U} \in \mathbb{R}^{C_{out} \times d}$ is the matrix expressing the convolution layer filters $\mathbf{\Theta}$. I.e., for each sliding window centered at (x, y), a convolution layer applies a linear transform $f : \mathbb{R}^d \to \mathbb{R}^{C_{out}}$ projecting the d dimensional local image patch $\mathbf{r}(x, y)$ to its C_{out} dimensional feature $\mathbf{g}(x, y)$. When $\mathbf{\Theta}$ is independently randomly sampled, e.g. from a Gaussian distribution, the convolution layer preserves global shapes since a random linear projection is *approximately* distance-preserving by bounding the range of r in Eq. 3.1 in Theorem 3.1.

Theorem 3.1. *Suppose we have N data points $\mathbf{z}_1, \cdots, \mathbf{z}_N \in \mathbb{R}^d$. Let $f(\mathbf{z}) = \mathbf{U}\mathbf{z}$ be a random linear projection $f : \mathbb{R}^d \to \mathbb{R}^m$ such that $\mathbf{U} \in \mathbb{R}^{m \times d}$ and $\mathbf{U}_{i,j} \sim N(0, \sigma^2)$. Then we have:*

$$\begin{aligned} P\Big(\sup_{i \neq j; i,j \in [N]} \Big\{ r_{i,j} := \frac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|}{\|\mathbf{z}_i - \mathbf{z}_j\|} \Big\} > \delta_1 \Big) \leq \epsilon, \\ P\Big(\inf_{i \neq j; i,j \in [N]} \Big\{ r_{i,j} := \frac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|}{\|\mathbf{z}_i - \mathbf{z}_j\|} \Big\} < \delta_2 \Big) \leq \epsilon, \end{aligned} \tag{3.3}$$

where $\delta_1 := \sigma\sqrt{\chi^2_{\frac{2\epsilon}{N(N-1)}}(m)}$ and $\delta_2 := \sigma\sqrt{\chi^2_{1-\frac{2\epsilon}{N(N-1)}}(m)}$. Here, $\chi^2_\alpha(m)$ denotes the α-upper quantile of the χ^2 distribution with m degrees of freedom.

Thm. 3.1 tells us that for any data pair $(\mathbf{z}_i, \mathbf{z}_j)$ in a set of N points, the distance rescaling ratio $r_{i,j}$ after a random linear projection is bounded by δ_1 and δ_2 with probability $1-\epsilon$. A Smaller N and a larger output dimension m give better bounds. E.g., when $m = 3$, $N = 1,000$, $\sigma = 1$ and $\epsilon = 0.1$, $\delta_1 = 5.8$ and $\delta_2 = 0.01$. Thm. 3.1 gives a theoretical bound for *all* the $N(N-1)/2$ pairs. However, in practice, preserving distances for a majority of $N(N-1)/2$ pairs is sufficient. To empirically verify this, we test the range of central 80% of $\{r_{i,j}\}$ on real image data. Using the same (m, N, σ, ϵ), 80% of the pairs lie in $[0.56, 2.87]$, which is significantly better than the strict bound: $[0.01, 5.8]$. A proof of the theorem and simulation details are given in the following.

Proof. Let $\mathbf{U}_k$ represent to the k-th row of $\mathbf{U}$. It is easy to check that $\mathbf{v}_k := \langle \mathbf{U}_k, \mathbf{z}_i - \mathbf{z}_j \rangle / \|\mathbf{z}_i - \mathbf{z}_j\| \sim N(0, \sigma^2)$. Therefore,

$$\frac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2} = \frac{1}{\sigma^2} \frac{(\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{U}^\top \mathbf{U} (\mathbf{z}_i - \mathbf{z}_j)}{\|\mathbf{z}_i - \mathbf{z}_j\|^2} = \sum_{k=1}^{m} \frac{\mathbf{v}_k^2}{\sigma^2} \sim \chi^2(m).$$

Therefore, for $0 < \epsilon < 1$, we have

$$P\Big(\frac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2} > \chi^2_{\frac{2\epsilon}{N(N-1)}}(m) \Big) \leq \frac{2\epsilon}{N(N-1)}.$$

From the above inequality, we have

$$\begin{aligned}
&P\Big(\sup_{i \neq j; i,j \in [N]} \Big\{ \tfrac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\|\mathbf{z}_i - \mathbf{z}_j\|^2} \Big\} > \sigma^2 \chi^2_{\frac{2\epsilon}{N(N-1)}}(m) \Big) \\
&= P\Big(\sup_{i \neq j; i,j \in [N]} \Big\{ \tfrac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2} \Big\} > \chi^2_{\frac{2\epsilon}{N(N-1)}}(m) \Big) \\
&= P\Big(\bigcup_{i \neq j; i,j \in [N]} \Big\{ \tfrac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2} > \chi^2_{\frac{2\epsilon}{N(N-1)}}(m) \Big\} \Big) \\
&\leq \sum_{i \neq j; i,j \in [N]} P\Big(\tfrac{\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2}{\sigma^2 \|\mathbf{z}_i - \mathbf{z}_j\|^2} > \chi^2_{\frac{2\epsilon}{N(N-1)}}(m) \Big) \\
&\leq \epsilon,
\end{aligned}$$

which is equivalent to

$$P\Big(\sup_{i\neq j;i,j\in[N]}\Big\{\frac{\|f(\mathbf{z}_i)-f(\mathbf{z}_j)\|}{\|\mathbf{z}_i-\mathbf{z}_j\|}\Big\}>\sigma\sqrt{\chi^2_{\frac{2\epsilon}{N(N-1)}}(m)}\Big)\leq\epsilon.$$

Similarly, we have

$$P\Big(\inf_{i\neq j;i,j\in[N]}\Big\{\frac{\|f(\mathbf{z}_i)-f(\mathbf{z}_j)\|}{\|\mathbf{z}_i-\mathbf{z}_j\|}\Big\}<\sigma\sqrt{\chi^2_{1-\frac{2\epsilon}{N(N-1)}}(m)}\Big)\leq\epsilon.$$

□

Simulation on Real Image Data To better understand the relative distance preservation property of random linear projections in practice, we use Algorithm 2 to empirically obtain a bound for real image data. We choose $m=3$, $N=1,000$, $\sigma=1$ and $\epsilon=0.1$ as in computing our theoretical bounds. We use $M=1,000$ real images from the PACS dataset for this simulation. Note that the image patch size or d does not affect the bound. We use a patch size of 3×3 resulting in $d=27$. This simulation tell us that applying linear projections with a randomly sampled U on N local images patches in every image, we have a $1-\epsilon$ chance that 80% of $r_{i,j}$ is in the range $[\delta_{10\%},\delta_{90\%}]$.

Algorithm 2 Simulate the range of central 80% of $r_{i,j}$ on real image data

Input: M images $\{I_i\}_{i=1}^M$, number of data points N, projection output dimension m, standard deviation σ of normal distribution, confidence level ϵ.
for $m=1\rightarrow M$ **do**
    Sample images patches in I_m at 1,000 locations and vectorize them as $\{\mathbf{z}_l^m\}_{l=1}^N$
    Sample a projection matrix $\mathbf{U}\in\mathbb{R}^{m\times d}$ and $\mathbf{U}_{i,j}\sim N(0,\sigma^2)$
    for $i=1\rightarrow N$ **do**
        for $j=i+1\rightarrow N$ **do**
            Compute $r_{i,j}^m=\frac{\|f(\mathbf{z}_i^m)-f(\mathbf{z}_j^m)\|}{\|\mathbf{z}_i^m-\mathbf{z}_j^m\|}$, where $f(\mathbf{z})=\mathbf{U}\mathbf{z}$
    $q_{10\%}^m$ = 10% quantile of $r_{i,j}^m$ for I_m
    $q_{90\%}^m$ = 90% quantile of $r_{i,j}^m$ for I_m ▷ Get the central 80% of $r_{i,j}$ in each image
$\delta_{10\%}$ = ϵ quantile of all $q_{10\%}^m$
$\delta_{90\%}$ = $(1-\epsilon)$ quantile of all $q_{90\%}^m$ ▷ Get the ϵ confident bound for $q_{10\%}^m$ and $q_{90\%}^m$
return $\delta_{10\%}$, $\delta_{90\%}$

3.5.3 Experimental Details

Digits Recognition The network for our digits recognition experiments is composed of two *Conv5×5-ReLU-MaxPool2×2* blocks with 64/128 output channels and three fully connected layer with 1024/1024/10 output channels. We train the network with batch size 32 for 10,000 iterations. During training, the model is validated every 250 iterations and saved with the best validation score for testing. We apply the `Adam` optimizer with an initial learning rate of 0.0001.

PACS We use the official data splits for training/validation/testing; no extra data augmentation is applied. We use the official `PyTorch` implementation and the pretrained weights of AlexNet for our PACS experiments. AlextNet is finetuned for 50,000 iterations with a batch size of 128. Samples are randomly selected from the training data mixed between the three domains. We use the validation data of the source domains only at every 100 iterations. We use the `SGD` optimizer for training with an initial learning rate of 0.001, Nesterov momentum, and weight decay set to 0.0005. We let the learning rate decay by a factor of 0.1 after finishing 80% of the iterations.

ImageNet Following the `PyTorch` example [1] on training ImageNet models, we set the batch size to 256 and train AlexNet from scratch for 90 epochs. We apply the `SGD` optimizer with an initial learning rate of 0.01, momentum 0.9, and weight decay 0.0001. We reduce the learning rate via a factor of 0.1 every 30 epochs.

3.5.4 More Experiments with ResNet-18

In this section, we demonstrate that `RandConv` also works on other stronger backbone architectures, e.g. for a Residual Network He et al. (2016b). Specifically, we run the PACS and ImageNet experiments with ResNet-18 as the baseline and `RandConv`. As Table 3.5 shows, `RandConv` improves the baseline using ResNet18 on ImageNet-sketch by 10.5% accuracy. When using a `RandConv` pretrained ResNet-18 on PACS, the performance of finetuning with DeepAll and `RandConv` are both improved as shown in Table 3.7. The best average domain generalization accuracy is 84.09%, with a more than 8% improvement over our initial Deep-All baseline. A model

[1] https://github.com/pytorch/examples/tree/master/imagenet

pretrained with $\text{RC}_{\text{mix1-7},\lambda=10}$ generally performs better than when pretrained with $\text{RC}_{\text{img1-7},p=0.5,\lambda=10}$. We also provide the ResNet-18 performance of JiGen (Carlucci et al., 2019) on PACS as a reference. Note that JiGen uses extra data augmentation and a different data split than our approach and it only improves over its own baseline by 1.5%. In addition, we test `RandConv` trained ResNet-18 on ImageNet-R (Hendrycks et al., 2020a), a domain generalization benchmark that contains images of artistic renditions of 200 object classes from the original ImageNet dataset. As Table 3.6 shows, `RandConv` also improves the generalization performance on ImageNet-R and reduces the gap between the in-domain (ImageNet-200) and out-of-domain (ImageNet-R) performance.

	Baseline	$\text{RC}_{\text{img1-7}}$, p=0.5, λ=10	$\text{RC}_{\text{mix1-7}}$, λ=10
Top1	20.23	28.79	30.70
Top5	37.26	49.02	51.80

Table 3.5: Accuracy of ImageNet-trained ResNet-18 on ImageNet-Sketch data.

	Baseline	$\text{RC}_{\text{img1-7}}$, p=0.5, λ=10	$\text{RC}_{\text{mix1-7}}$, λ=10
ImageNet-200 (%)	88.15	83.72	72.7
ImageNet-R (%)	33.06	37.38	35.75
Gap	55.09	46.34	36.95

Table 3.6: Top 1 Accuracy of ImageNet-trained ResNet-18 on ImageNet-R data. ImageNet-200 are the original ImageNet data with the same 200 classes as ImageNet-R.

3.5.5 Hyperparameter Selections and Ablation Studies on Digits Recognition Benchmarks

We provide detailed experimental results for the digits recognition datasets. Table 3.8 shows results for different hyperparameters p for RC_{img1}. Table 3.9 shows results for an ablation study on the multi-scale design for RC_{mix} and $\text{RC}_{\text{img},p=0.5}$. Table 3.10 shows results for studying the consistency loss weight λ for $\text{RC}_{\text{mix1-7}}$ and $\text{RC}_{\text{img1-7},p=0.5}$. Tables 3.8, 3.9, and 3.10 correspond to Fig. 3.2 (a)(b)(c) in the main text respectively.

PACS	ImageNet	Photo	Art	Cartoon	Sketch	Avg
Deep-All	Baseline	**95.45(0.43)**	74.96(0.99)	71.48(1.22)	62.09(1.12)	76.00(0.37)
	$RC_{\text{img1-7},p=0.5,\lambda=10}$	94.65(0.16)	73.85(0.97)	74.78(0.58)	73.51(1.16)	79.20(0.40)
	$RC_{\text{mix1-7},\lambda=10}$	94.10(0.43)	76.72(1.43)	73.41(1.29)	77.60(0.55)	80.46(0.74)
$RC_{\text{img1-7}, p=0.5,\lambda=10}$	Baseline	92.37(0.54)	76.50(0.55)	71.33(0.29)	79.65(1.32)	79.96(0.53)
	$RC_{\text{img1-7},p=0.5,\lambda=10}$	94.43(0.22)	79.80(1.03)	73.40(0.37)	81.51(0.85)	82.28(0.38)
	$RC_{\text{mix1-7},\lambda=10}$	94.57(0.45)	**81.32(1.00)**	**76.28(0.82)**	**84.18(0.94)**	**84.09(0.61)**
$RC_{\text{mix1-7}}$ $\lambda=10$	Baseline	93.57(0.40)	77.73(0.91)	71.24(0.91)	75.53(2.17)	79.52(0.61)
	$RC_{\text{img1-7},p=0.5,\lambda=10}$	<u>95.23(0.30)</u>	80.56(0.82)	74.18(0.53)	80.70(1.43)	82.67(0.46)
	$RC_{\text{mix1-7},\lambda=10}$	95.01(0.32)	<u>81.09(1.24)</u>	<u>76.04(0.92)</u>	<u>83.02(0.93)</u>	<u>83.79(0.60)</u>
Deep-All	Baseline	95.73	77.85	74.86	67.74	79.05
JiGen		96.03	79.42	75.25	71.35	80.51

Table 3.7: Generalization results on PACS with `RandConv` pretrained model using ResNet-18. ImageNet column shows how the pretrained model is trained on ImageNet (baseline represents training using only the classification loss); PACS column indicates the methods used for finetuning on PACS. **Best** and <u>second best</u> accuracy for each target domain are highlighted in bold and underlined. The performance of JiGen (Carlucci et al., 2019) and its baseline using ResNet-18 is also given.

	MNIST-10k	MNIST-M	SVHN	USPS	SYNTH	DG Avg	MNIST-C
Baseline	98.40(0.84)	58.87(3.73)	33.41(5.28)	79.27(2.70)	42.43(5.46)	53.50(4.23)	88.20(2.10)
$RC_{\text{img1}, p=0.9}$	98.68(0.06)	83.53(0.37)	53.67(1.54)	80.38(1.41)	59.19(0.85)	69.19(0.34)	**89.79(0.44)**
$RC_{\text{img1}, p=0.7}$	98.64(0.07)	84.17(0.61)	54.50(1.55)	**80.85(0.91)**	60.25(0.85)	69.94(0.50)	89.20(0.60)
$RC_{\text{img1}, p=0.5}$	98.72(0.08)	85.17(1.12)	**55.97(0.54)**	80.31(0.85)	**61.07(0.47)**	**70.63(0.42)**	88.66(0.62)
$RC_{\text{img1}, p=0.3}$	98.71(0.12)	85.45(0.87)	54.62(1.52)	79.78(1.40)	60.51(0.41)	70.09(0.60)	89.02(0.32)
$RC_{\text{img1}, p=0.1}$	98.66(0.06)	85.57(0.79)	54.34(1.52)	79.21(0.44)	60.18(0.63)	69.83(0.38)	88.53(0.38)
$RC_{\text{img1}, p=0}$	98.55(0.13)	**86.27(0.42)**	52.48(3.00)	79.01(1.11)	59.53(1.14)	69.32(1.19)	88.01(0.36)

Table 3.8: Ablation study of hyperparameter p for RC_{img1} on digits recognition benchmarks. DG-Avg is the average performance on MNIST-M, SVHN, SYNTH and USPS. Best results are **bold**.

	MNIST-10k	MNIST-M	SVHN	USPS	SYNTH	DG Avg	MNIST-C
RC_{mix1}	98.62(0.06)	83.98(0.98)	53.26(2.59)	80.57(1.09)	59.25(1.38)	69.26(1.35)	88.59(0.38)
$\text{RC}_{\text{mix1-3}}$	98.76(0.02)	84.66(1.67)	55.89(0.83)	80.95(1.15)	60.07(1.05)	70.39(0.58)	89.80(0.94)
$\text{RC}_{\text{mix1-5}}$	98.76(0.06)	84.32(0.43)	**56.50(2.68)**	81.85(1.05)	60.76(1.02)	70.86(0.86)	90.06(0.80)
$\text{RC}_{\text{mix1-7}}$	98.82(0.06)	84.91(0.68)	55.61(2.63)	**82.09(1.00)**	**62.15(1.30)**	**71.19(1.21)**	90.30(0.44)
$\text{RC}_{\text{mix1-9}}$	98.81(0.12)	**85.13(0.72)**	54.18(3.36)	82.07(1.28)	61.85(1.41)	70.81(1.24)	**90.83(0.52)**
$\text{RC}_{\text{img1}, p=0.5}$	98.66(0.05)	85.12(0.96)	55.59(0.29)	80.65(0.71)	60.85(0.48)	70.55(0.15)	89.00(0.45)
$\text{RC}_{\text{img1-3}, p=0.5}$	98.79(0.07)	85.36(1.04)	**55.60(1.09)**	80.99(0.99)	61.26(0.80)	70.80(0.86)	89.84(0.70)
$\text{RC}_{\text{img1-5}, p=0.5}$	98.83(0.07)	**86.33(0.47)**	54.99(2.48)	80.82(1.83)	62.61(0.75)	71.19(1.25)	90.70(0.43)
$\text{RC}_{\text{img1-7}, p=0.5}$	98.83(0.07)	86.08(0.27)	54.93(1.27)	**81.58(0.74)**	**62.78(0.86)**	**71.34(0.61)**	**91.18(0.38)**
$\text{RC}_{\text{img1-9}, p=0.5}$	98.80(0.12)	85.63(0.70)	52.82(2.01)	81.48(1.22)	62.55(0.74)	70.62(0.73)	90.79(0.48)

Table 3.9: Ablation study of multi-scale `RandConv` on digits recognition benchmarks for RC_{mix} and $\text{RC}_{\text{img},p=0.5}$. Best entries for each variant are **bold**.

	λ	MNIST-10k	MNIST-M	SVHN	USPS	SYNTH	DG Avg	MNIST-C
$\text{RC}_{\text{mix1-7}}$	20	98.90 (0.05)	87.18 (0.81)	**57.68 (1.64)**	**83.55 (0.83)**	63.08 (0.50)	72.87 (0.47)	91.14 (0.53)
	10	98.85 (0.04)	**87.76 (0.83)**	57.52 (2.09)	83.36 (0.96)	62.88 (0.78)	**72.88 (0.58)**	**91.62 (0.77)**
	5	98.94 (0.09)	87.53 (0.51)	55.70 (2.22)	83.12 (1.08)	62.37 (0.98)	72.18 (1.04)	91.46 (0.50)
	1	98.95 (0.05)	86.77 (0.79)	56.00 (2.39)	83.13 (0.71)	**63.18 (0.97)**	72.27 (0.82)	91.15 (0.42)
	0.1	98.84 (0.07)	85.41 (1.02)	56.51 (1.58)	81.84 (1.14)	61.86 (1.44)	71.41 (0.98)	90.72 (0.60)
	0	98.82 (0.06)	84.91 (0.68)	55.61 (2.63)	82.09 (1.00)	62.15 (1.30)	71.19 (1.21)	90.30 (0.44)
$\text{RC}_{\text{img1-7}}$ p=0.5	20	98.79 (0.04)	87.53 (0.79)	53.92 (1.59)	81.83 (0.70)	62.16 (0.37)	71.36 (0.49)	**91.20 (0.53)**
	10	98.86 (0.05)	87.67 (0.37)	54.95 (1.90)	82.08 (1.46)	63.37 (1.58)	72.02 (1.15)	90.94 (0.51)
	5	98.90 (0.04)	**87.77 (0.72)**	**55.00 (1.40)**	**82.10 (0.55)**	**63.58 (1.33)**	**72.11 (0.62)**	90.83 (0.71)
	1	98.86 (0.04)	86.74 (0.32)	53.26 (2.99)	81.51 (0.48)	62.00 (1.15)	70.88 (0.93)	91.11 (0.62)
	0.1	98.85 (0.14)	86.85 (0.31)	53.55 (3.63)	81.23 (1.02)	62.77 (0.80)	71.10 (1.31)	91.13 (0.69)
	0	98.83 (0.07)	86.08 (0.27)	54.93 (1.27)	81.58 (0.74)	62.78 (0.86)	71.34 (0.61)	91.18 (0.38)

Table 3.10: Ablation study of consistency loss weight λ on digits recognition benchmarks for $\text{RC}_{\text{mix1-7}}$ and $\text{RC}_{\text{img1-7},p=0.5}$. DG-Avg is the average performance on MNIST-M, SVHN, SYNTH and USPS. Best results for each variant are **bold**.

3.5.6 More Examples of RandConv Data Augmentation

We provide additional examples of RandConv outputs for different convolution filter sizes in Fig. 3.6 and for its mixing variants at scale $k = 7$ with different mixing coefficients in Fig. 3.5. We observe that RandConv with different filter sizes retains shapes at different scales. The mixing strategy can continuously interpolate between the training domain and a randomly sampled domain.

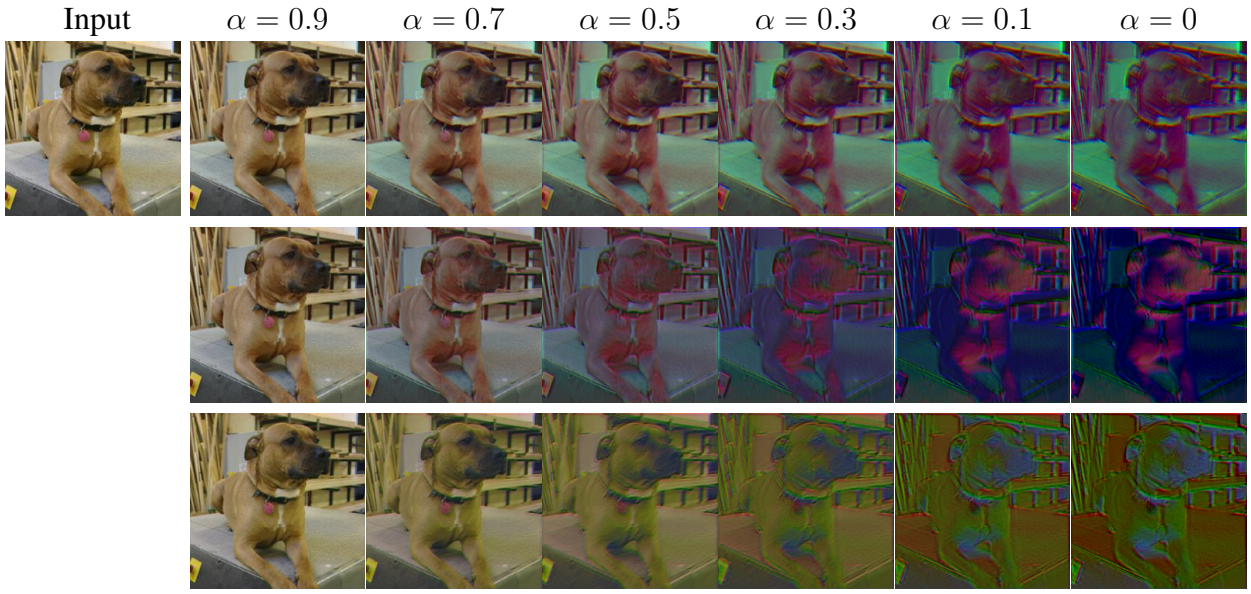


Figure 3.5: Examples of the RandConv mixing variant RC_{mix7} on images of size 224^2 with different mixing coefficients α. When $\alpha = 1$, the output is just the original image input; when $\alpha = 0$, we use the output of the random convolution layer as the augmented image.

3.6 Conclusion and Discussion

Randomized convolution (RandConv) is a simple but powerful data augmentation technique for randomizing local image texture. RandConv helps focus visual representations on global shape information rather than local texture. We theoretically justified the approximate shape-preserving property of RandConv and developed RandConv techniques using multi-scale and mixing designs. We also make use of a consistency loss to encourage texture invariance. RandConv outperforms state-of-the-art approaches on the digit recognition benchmark and on the sketch domain of PACS and on ImageNet-Sketch by a large margin. By finetuning a model pretrained with RandConv on

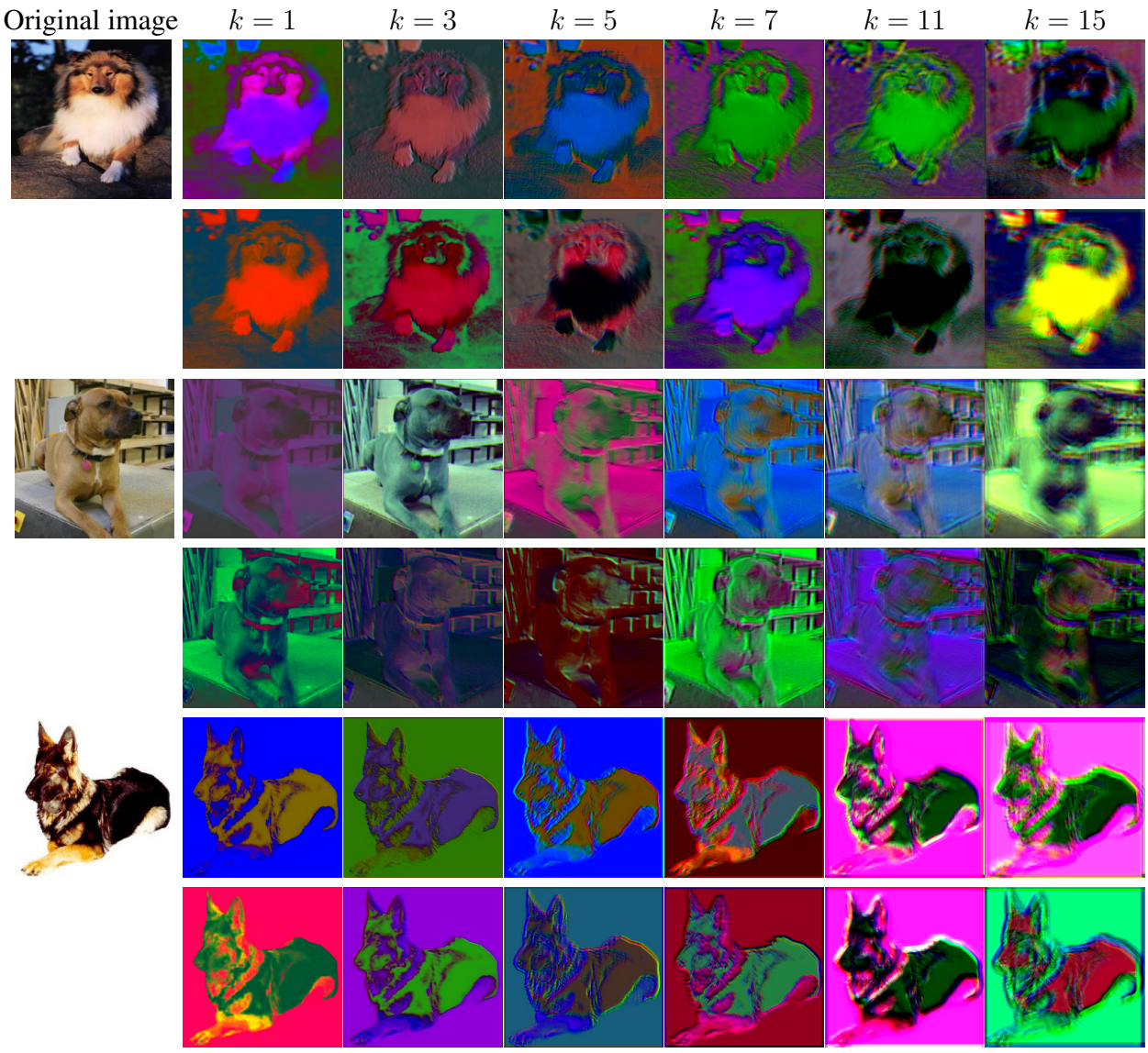


Figure 3.6: `RandConv`data augmentation examples on images of size 224^2. First column is the input image; following columns are convolution results using random filters of different sizes k. We can see that the smaller filter sizes help maintain the finer shapes.

PACS, we showed that the generalizability of a pretrained model may transfer to and benefit a new downstream task. This resulted in a new state-of-art performance on PACS in the Sketch domain.

`RandConv` can help computer vision tasks when a shape-biased model is helpful e.g. for object detection. `RandConv` can also provide a shape-biased pretrained model to improve performance on downstream tasks when generalizing to unseen domains. However, local texture features can be useful for many computer vision tasks, especially for fixed-domain fine-grained visual recognition. In such cases, visual representations that are invariant to local texture may hurt in-domain performance. Therefore, important future work includes learning representations that disentangle shape and texture features and building models to use such representations in an explainable way.

Adversarial robustness of deep neural networks has received significant recent attention. Interestingly, (Zhang and Zhu, 2019) find that adversarially-trained models are more shape biased and less texture/color-biased; (Shi et al., 2020) show that their method for increasing shape bias also helps adversarial robustness, especially when combined with adversarial training. Therefore, exploring how `RandConv` affects the adversarial robustness of models could be interesting future work. Moreover, recent biologically inspired models for improving adversarial robustness (Dapello et al., 2020) use Gabor filters with fixed random configurations followed by a stochastic layer to add Gaussian noise to the network input, which may explain the importance of randomness in `RandConv`. Exploring connections between `RandConv` and biological mechanisms in the human visual system would be interesting future work.

CHAPTER 4: Compositional Generalization in Unsupervised Compositional Representation Learning: A Study on Disentanglement and Emergent Language

Deep learning models struggle with compositional generalization, i.e., the ability to recognize or generate novel combinations of observed elementary concepts. In hopes of enabling compositional generalization, various unsupervised learning algorithms have been proposed with inductive biases that aim to induce compositional structure in learned representations (e.g. disentangled representation and emergent language learning). In this chapter, we evaluate these unsupervised learning algorithms in terms of how well they enable *compositional generalization*. Specifically, our evaluation protocol focuses on whether or not it is easy to train a simple model on top of the learned representation that generalizes to new combinations of compositional factors.

- We systematically study three unsupervised representation learning algorithms – β-VAE, β-TCVAE, and emergent language (EL) autoencoders – on two datasets that allow directly testing compositional generalization. We find that directly using the bottleneck representation with simple models and few labels may lead to worse generalization than using representations from layers before or after the learned representation itself.

- Additionally, we find that the previously proposed metrics for evaluating the levels of compositionality are not correlated with the actual compositional generalization in our framework. Surprisingly, we find that increasing pressure to produce a disentangled representation (e.g., increasing β in the β-VAE) produces representations with *worse* generalization, whereas representations from EL models show strong compositional generalization.

- Motivated by this observation, we further investigate the advantages of using EL to induce compositional structure in unsupervised representation learning, finding that it shows consistently stronger generalization than disentanglement models, especially when using less unlabeled data for unsupervised learning and fewer labels for downstream tasks.

Taken together, our results shed new light onto the compositional generalization behavior of different unsupervised learning algorithms with a new setting to rigorously test this behavior, and suggest the potential benefits of developing EL learning algorithms for more generalizable representations.

The remainder of this chapter is organized as follows: Section 4.1 introduces the motivation and contribution of our work. Section 4.2 describes the details of unsupervised disentanglement learning and emergent language learning algorithms. Section 4.3 presents the experimental design including data, evaluation protocal, and implementation details for evaluation the compositional generalization. Section 4.4 illustrate the key studies, interesting findings and the their implications. Section 4.6 discusses the limitation of our study, Section 4.5 introduces related works, and Section 4.9 concludes this chapter. Section 4.7 and Section 4.8 provides further details and additional studies and results. The work presented in this chapter has been accepted for publishing in Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022) (Xu et al., 2022).

4.1 Introduction

A human's ability to recognize or generate novel combinations of seen elementary concepts, also known as *compositional generalization*, is desirable for building general artificial ingelligence (AI) systems (Hoffman and Richards, 1984; Fodor and Pylyshyn, 1988; Biederman, 1987). The Recognition-By-Components theory by Biederman (Biederman, 1987) influenced the early development of computer vision models that are inherently compositional, e.g., hierarchical features (Felzenszwalb et al., 2009; Fidler and Leonardis, 2007) and part-based models (Ott and Everingham, 2011; Pandey and Lazebnik, 2011). However, modern deep learning systems still struggle with this key capability of human intelligence (Marcus, 2018). A few works studied specific spatial and object-wise compositionality (Stone et al., 2017; Lin et al., 2020) or more general compositionality in the space of pre-defined attributes (Tokmakov et al., 2019; Purushwalkam et al., 2019) or the semantics of human language descriptions (Yun et al., 2022; Thrush et al., 2022).

Humans often express complex meaning in a compositional manner: we combine elementary representations to describe observations. For example, an object with simple geometry is described by separate and independent properties such as color (red, blue, ...), position (left, right, close, far

away,...), and shape (circle, triangle, ...). Therefore, *compositional representations* are thought as helpful or even essential to achieve compositional generalization (Hoffman and Richards, 1984; Fodor, 1975; Biederman, 1987; Lake et al., 2017). However, considering that there are exponentially many possible combinations of a given set of elementary concepts, we need to deal with this combinatorial explosion for real-world visual observations. It is unrealistic to annotate enough data to learn the fine-grained compositionality. Therefore, unsupervised compositional representation learning is appealing because it does not require comprehensive labeling. However, unsupervised representation learning heavily relies on the design of an effective inductive bias (e.g. on the representation formulation) to induce the emergence of compositional representations.

A widely explored representation formulation with explicit compositionality is *disentanglement*. The most common formulation of disentanglement is that the generative factors of observations should be encoded into different factors of low-dimensional representations, and a change of a single factor in an observation leads to a change in a single factor of the representation. State-of-the-art unsupervised disentanglement models (Higgins et al., 2017; Kim and Mnih, 2018; Kumar et al., 2018; Chen et al., 2018; Mathieu et al., 2019) are largely built on top of variational generative models (Kingma and Welling, 2013). To measure the level of disentanglement, various quantitative metrics have been proposed that are defined based on the statistical relations between the learned representations and ground truth factors with an emphasis on the separation of factors. A summary of disentanglement metrics and methods is provided in (Locatello et al., 2019). Another representation learning approach with an inductive bias towards compositionality is *emergent language learning*. Natural language allows us to describe novel composite concepts by combining expressions of their elementary concepts according to grammar. Therefore, linguists have been interested in studying the compositionality of discrete codes evolving during multi-agent communication when agents learn to complete a task cooperatively (Chrupała et al., 2015; Lazaridou et al., 2016; Havrylov and Titov, 2017; Ren et al., 2020). Compositionality metrics with language structure assumptions, e.g. topographic similarity (Brighton and Kirby, 2006), were used to evaluate the learned language. However, for the above two types of methods, relatively few studies (Montero et al., 2021; Schott

et al., 2022; Chaabouni et al., 2020; Andreas, 2019) have directly evaluated how well the learned representations generalize to novel combinations on downstream tasks, which is the main motivation for compositional representation learning in the first place.

In this work, we study the *compositional generalization* performance of representations learned from unsupervised learning algorithms with two types of inductive biases for compositionality: disentanglement and emergent languages (EL). Instead of measuring the compositionality and disentanglement metrics defined based on various assumptions, we directly measure the generalization performance on novel input combinations with a two-stage protocol. Specifically, with a dataset divided into train and test sets ensuring that the test set contains *novel combinations of concepts* that never appear in the train set, we first learn an unsupervised representation model from unlabeled images in the train set. With very few labeled samples from the train set and the frozen unsupervised representation model, we train simple (e.g. linear) models on top of learned representations to predict the ground truth value for each generative factor of the dataset and evaluate these simple models on the test set. These choices are aligned with common practices in recent deep representation learning works, for example, self-supervised representation learning (Dittadi et al., 2021) and semi-supervised learning with generative models (Kingma et al., 2014). Different from previous studies (e.g. (Chaabouni et al., 2020; Montero et al., 2021)) that measure unsupervised learning performance (e.g. image reconstruction), we evaluate the performance on downstream tasks. We also emphasize how easily we can obtain downstream task models with the learned representation, e.g. when using very few labels and simple linear models. These designs highlight the generalization performance of the unsupervised learning stage, different from a setup that uses many more or even all labeled samples of the train set in the downstream task learning stage (Dittadi et al., 2021; Schott et al., 2022) or performs unsupervised learning on the entire dataset (Locatello et al., 2019). More importantly, we study not only the compositional generalization of intermediate representations at the model bottleneck, e.g. where the disentangled latent variables are formulated, but also the representations from layers before or after it.

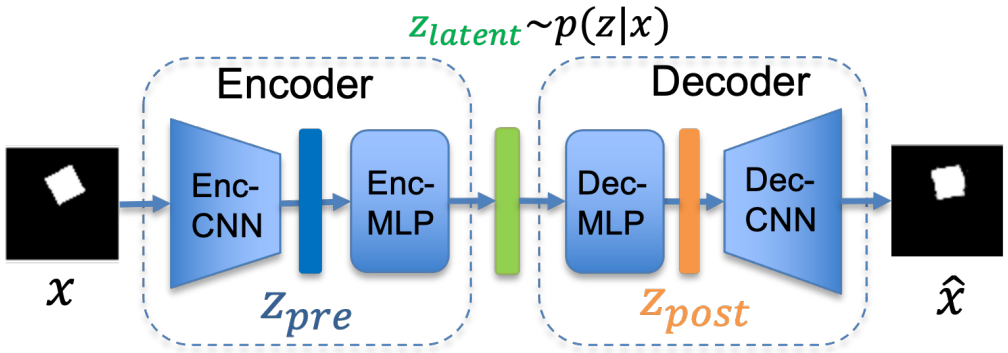


Figure 4.1: The architecture of VAE-based unsupervised disentanglement models.

With the above evaluation protocol for compositional generalization, we explore selected unsupervised learning algorithms by varying (1) the hyperparameters of each algorithm that may control the levels of compositionality; (2) image datasets and the amount of data for both unsupervised and supervised learning stages; and (3) design choices of EL learning. First, we find that, compared to the low-dimensional latent variables from the model bottleneck, representations from the layers before or after the model bottleneck enable better compositional generalization. Second, we find that attaining higher scores on previously proposed compositionality/disentanglement metrics does not always correlate with better generalization performance. Finally, the representations learned from EL models show stronger generalization performance than disentanglement models. These findings reveal the divergence between the efforts to achieve better compositionality / disentanglement metric scores and the initial motivation to obtain better generalization performance. To our best knowledge, this is the first study to compare representations in disentanglement models and emergent language learning through the lens of compositional generalization with a unified evaluation setup. We also discuss the advantage of the emergent language representation format versus disentanglement and connect it with recent related research in representation learning.

4.2 Unsupervised Learning with Compositional Representation Inductive Bias

In this section, we introduce more details on unsupervised learning algorithms with two different compositionality-seeking inductive biases on the representation formulation: disentangled representations in Section 4.2.1 and emergent languages in Section 4.2.2.

4.2.1 Learning Disentangled Representations

The concept of disentanglement assumes that high-dimensional observations of the real world $\mathbf{x}$ can be represented by low-dimensional latent variables $\mathbf{z}$, where each dimension of $\mathbf{z}$ encodes independent factors of variations in $\mathbf{x}$. For unsupervised disentanglement learning algorithms, we select the popular β-VAE and β-TCVAE methods which both modify the evidence lower bound (ELBO) objective in the variational autoencoder (VAE).

β-VAE use a hyperparameter β for the Kullback-Leibler (KL) regularization term of the vanilla VAE loss to control the bandwidth of the VAE bottleneck:

$$\mathbb{E}_{p(\mathbf{x})}[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta \mathbf{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \,, \tag{4.1}$$

where $p(\mathbf{z})$ is the assumed prior distribution of the latent variables and its conditional distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is parameterized by a neural network (encoder) whose parameters are ϕ and the posterior $p_\theta(\mathbf{x}|\mathbf{z})$ is parameterized by a decoder whose parameters are θ. $\beta = 1$ corresponds to the VAE loss.

β-TCVAE further decomposes the KL term in Eq. (4.1) into mutual information, total correlation, and dimension-wise KL terms, and penalizes the total correlation with the hyperparameter β:

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \alpha I_q(\mathbf{x};\mathbf{z}) - \beta \mathbf{KL}(q_\phi(\mathbf{z})||\prod_j q_\phi(z_j)) - \gamma \sum_j \mathbf{KL}(q(z_j)||p(z_j)) \,, \tag{4.2}$$

where $I_q(\mathbf{x};\mathbf{z})$, $\mathbf{KL}(q_\phi(\mathbf{z})||\prod_j q_\phi(z_j))$ and $\sum_j \mathbf{KL}(q(z_j)||p(z_j))$ are the mutual information term, total correlation term and the dimension-wise KL term respectively. The proposed β-TCVAE uses $\alpha = \gamma = 1$ and tunes β only.

4.2.2 Learning Emergent Language

An alternative compositional representation learning method is emergent language (EL) learning, which aims to learn a representation that mimics the properties of natural language. The emergent language consists of sequences of discrete symbols from a vocabulary. Since the model

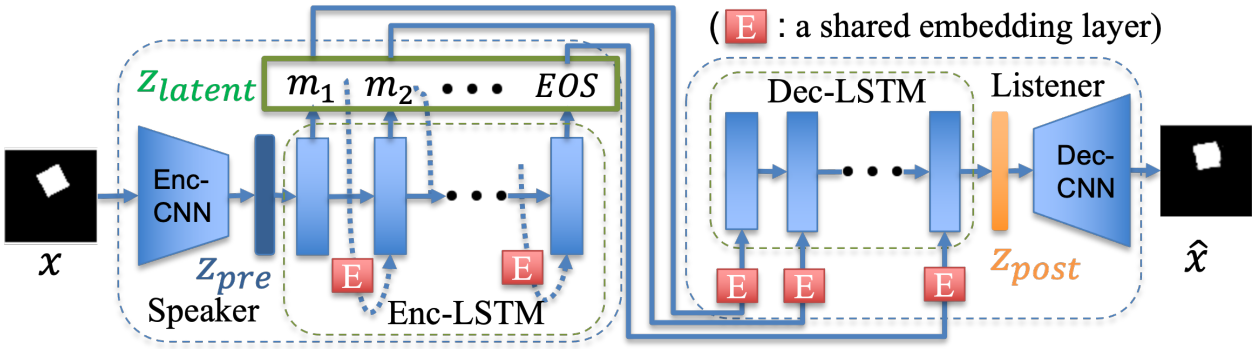


Figure 4.2: The architecture of emergent language learning models.

combines discrete symbols in the vocabulary to represent complex semantics in observations, one expects that meaningful compositionality might naturally emerge in communication between multiple agents using the emergent language to solve tasks. We consider the typical speaker-listener model (two-agent communication) for EL learning, as shown in Figure 4.2. We apply EL learning to the image reconstruction task to be consistent with the reconstruction objective used by variational auto-encoder-based models. Note that in our setting the terminology "speaker-listener" is equivalent to the more common "encoder-decoder" terminology. The task is as follows.

1. The speaker receives an input $\mathbf{x}$ and encodes it as a message $\mathbf{m} = \{m^1, m^2, ...\}$, a sequence of discrete symbols from the vocabulary $V = \{c_1, c_2, c_{n_V}\}$ of size n_V. The maximum length of $\mathbf{m}$ is n_{msg}.

2. The listener model receives the message $\mathbf{m}$ and outputs $\hat{\mathbf{x}}$, aiming to accurately reconstruct the encoder input $\mathbf{x}$.

In this work, we use a speaker and listener that are both hybrids of a convolutional neural network and an LSTM recurrent neural network. The flattened convolutional embedding of the input image, $\text{EncConv}(\mathbf{x})$, is used as the initial cell state of an LSTM encoding module (EncLSTM). EncLSTM generates a discrete distribution $q(m^t|x)$ over V at each time step t autoregressively (with the embedding of the discrete token sampled at the previous step $t-1$ as input):

$$q(m^t|\mathbf{x}) = \text{EncLSTM}(m^{t-1}|\text{EncConv}(\mathbf{x}), \text{emb}(m^1), .., \text{emb}(m^{t-2}))\,, \tag{4.3}$$

where emb$(\cdot)$ is the learnable layer that projects a discrete token into a high-dimensional embedding. We use the Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to sample from the discrete distribution $q(m^t|x)$ and the "straight-through" (ST) gradient estimator (Bengio et al., 2013) for quantization (from soft-distribution to one-hot vector). This allow us to estimate the gradients from the discrete sampling process.

$$m^t = \text{ST}(\text{GumbelSoftmax}(q(m^t|\mathbf{x}))) \,. \tag{4.4}$$

To allow the message to be of variable length, which better mimics natural language, we set one token in V to be the end-of-sequence (EOS) token that indicates the message end.

When the listener decodes the message $\mathbf{m}$, it first maps each discrete symbol into an embedding based on a learnable embedding layer and uses a decoding LSTM layer (DecLSTM) to process the sequence of embeddings recurrently.

$$\text{Emb}^t(\mathbf{m}) = \text{DecLSTM}(\text{emb}(m^t)|\text{emb}(m^1), \text{emb}(m^2), .., \text{emb}(m^{t-1})) \,. \tag{4.5}$$

We use the output of DecLSTM at the ending step T as the embedding of the message to be the input of a convolutional decoder for image reconstruction:

$$\text{Emb}(\mathbf{m}) = \text{Emb}^T(\mathbf{m}), \text{where } T = \min(n_{msg}, \arg\min_i \{m^i == EOS, i \in \{1..N\}\}) \,. \tag{4.6}$$

Finally, the convolutional decoder (DecConv) reconstructs the input by:

$$\hat{\mathbf{x}} = \text{DecConv}(\text{Emb}(\mathbf{m})) \,. \tag{4.7}$$

4.3 Experimental Design

4.3.1 Datasets

We are interested in whether representations can generalize to novel combinations of seen concepts. Therefore, we need datasets that provide ground truth labels of elementary concepts for (1) creating train/test splits and (2) downstream task evaluation. We consider datasets with n_{gen} independent generative factors $F = \{f_1, ..., f_{n_{\text{gen}}}\}$ where the space of factor f_i is S_i. For example, if f_1 is color, then S_1 could be $\{yellow, red, blue, ...\}$. The data space D is defined by the Cartesian product of the spaces of each factor, and therefore the cardinality of the dataset is $|D| = \prod_i^{n_{\text{gen}}} |S_i|$.

In our study we use two public image datasets studied in the disentanglement literature: dSprites (Matthey et al., 2017) and MPI3D (Gondal et al., 2019). The dSprites dataset contains images of 2D shapes generated from 5 factors $F = \{$shape, scale, rotation, x and y position$\}$. To avoid label ambiguity due to the rotational symmetry of the square and ellipse shapes, we limit the range of orientations to be within [0, $\pi/2$). Then the cardinality of each factor's space is $\{3, 6, 10, 32, 32\}$ respectively, which makes $|D| = 183,320$. MPI3D is a set of 3D datasets synthesized or recorded in a controlled environment with an object held by a robotic arm. In our evaluation, the challenging real-world version (MPI3D-Real) is used. It has 7 factors F = $\{$object-color(6), object-shape(6), object-size(2), camera-height(3), background-color(3), horizontal-axis(40), vertical-axis(40)$\}$ with the corresponding cardinalities of the space in parentheses, leading to a total of 1,036,800 images.

4.3.2 Compositional Generalization Evaluation Protocol

Our evaluation protocol emphasizes the compositional generalization that models can achieve on downstream tasks. We aim to measure how *easily* an unsupervised representation learning method can produce compositional generalization by using a simple model on top of the learned representation. Figure4.3 shows the designs of our evaluation protocol. (1) *Data splits.* We first split a dataset into train/test sets randomly while ensuring that all samples in the test-set are novel combinations of elementary factors seen in the train-set. (2) *Unsupervised representation learning.*

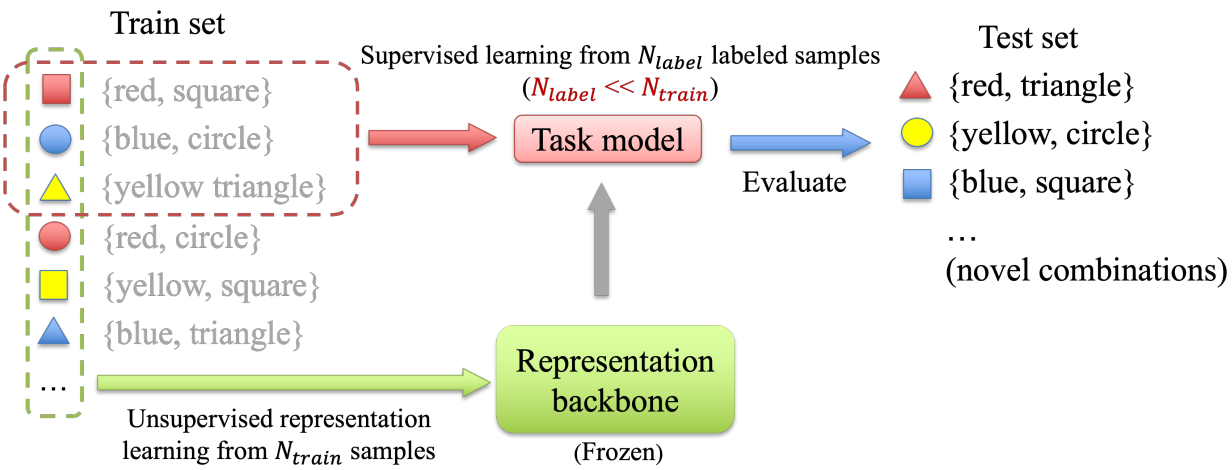


Figure 4.3: Our compositional generalization evaluation protocol.

We learn unsupervised representations from the unlabeled images of the train-set of size N_{train} with a selected algorithm. (3) *Learning for downstream tasks.* Then, we freeze the learned representation model and use the N_{label} labeled samples from the train set ($N_{label} << N_{train}$), to train a simple classifier/regressor to predict the ground truth value for each factor f_i of the dataset. (4) *Testing generalization performance.* Lastly, we test the performance of the downstream task models on novel combinations of seen values of elementary factors.

Readout model. We argue that it is important to use simple read-out models and a limited number of labeled training samples to evaluate downstream tasks. Otherwise, the performance gain from the downstream task learning stage is mixed with that of the unsupervised learning stage. For example, if the unsupervised representation model is an identity mapping, we can still get great performance with a powerful read-out model and enough labeled samples. In our main article, we use linear models for downstream tasks. However, perfectly disentangled but linearly inseparable representations may still show poor performance with a linear readout model. Through sanity checking experiments (discussed in Section 4.8.1), the linear readout model can still generalize well on nonlinear oracle representations possibly due to the limited value range of attributes in our datasets. In addition, extra results using a non-linear readout model (Gradient Boosting Trees) are in Section 4.8 and the main observations are consistent. The linear models we use to predict the values of the generative factors are ridge regression, using the R^2 score as the evaluation metric,

and logistic regression, using classification accuracy as the evaluation metric. Since the R^2 score can be negative while $R^2 = 0$ indicates random guessing, we clip all negative R^2 scores to zero.

Representation Mode. In unsupervised disentanglement learning, low-dimensional latent variables with explicit disentanglement regularization are used as the representation for downstream tasks, e.g. the mean values of Gaussian distributions of the latent variables in the VAE. If the disentangled latent variables each represent a single ground truth factor, only a simple mapping between factors and the corresponding variables must be learned to achieve good performance. However, it is questionable if the learned latent variables disentangle in the assumed structure and therefore improve generalization. On the other hand, the simple linear models in our evaluation protocol may not be capable to process the latent variables (discrete messages) in emergent language (EL) learning because we would not expect discrete sequential latent variables to be easily linearly separable. Therefore, we also evaluate the intermediate features of the layers before or after the model bottleneck. Specifically, in addition to the latent variables ($\mathbf{z}_{\text{latent}}$), we also use the features immediately after the convolutional encoder ($\mathbf{z}_{\text{post}}$) or before the convolutional decoder ($\mathbf{z}_{\text{pre}}$) as representations of images, shown in Figure 4.1 and 4.2. We evaluate the use of $\mathbf{z}_{\text{post}}$ and $\mathbf{z}_{\text{pre}}$ in both disentanglement and emergent language models.

4.3.3 Implementation details

Data Splits For the main experiments, we use a 1:9 train/test split. The train set size (10%) is smaller than what common machine learning setups and previous studies have used (Montero et al., 2021; Schott et al., 2022). However, considering the number of possible combinations increases exponentially for real data with an increased number of generative factors, using fewer training samples even at the unsupervised learning stage can help to obtain more meaningful conclusions for real-world scenarios.

Model architectures. Figure 4.1 and 4.2 show the architectures for disentanglement and emergent language learning models, respectively. The encoder and decoder in disentanglement learning models are symmetric architectures with a convolutional network and a multi-layer perceptron (MLP) similar to the design in (Burgess et al., 2018). We scale up the size of the model by

doubling the width of all layers, which improves the performance of all models. Since our emergent language learning model uses the same autoencoding task as disentanglement learning, it also uses a symmetric encoder-decoder architecture. Instead of MLPs, the EL model uses LSTM modules to encode and decode latent variables. The sizes of the MLP module in the disentanglement models and the LSTM module in the EL models are matched for fair comparison. We use a Bernoulli decoder for the autoencoding task, which uses pixel values normalized to [0, 1] as probabilities.

Hyperparameters. We follow previous studies (Locatello et al., 2019; Montero et al., 2021) for disentanglement models to set our hyperparameters. We set the number of disentangled latent variables, $|\mathbf{z}_{latent}|$, and the maximum length of the emergent latent message, n_{msg}, to 10, unless specifically mentioned. For the pre-training stage, we use batch size 64 and train the model for 500,000 steps for dSprites and 1,000,000 steps for MPI3D-Real using the Adam optimizer with learning rate 0.0001. For each specific model, we collect results from three different runs with different random seeds.

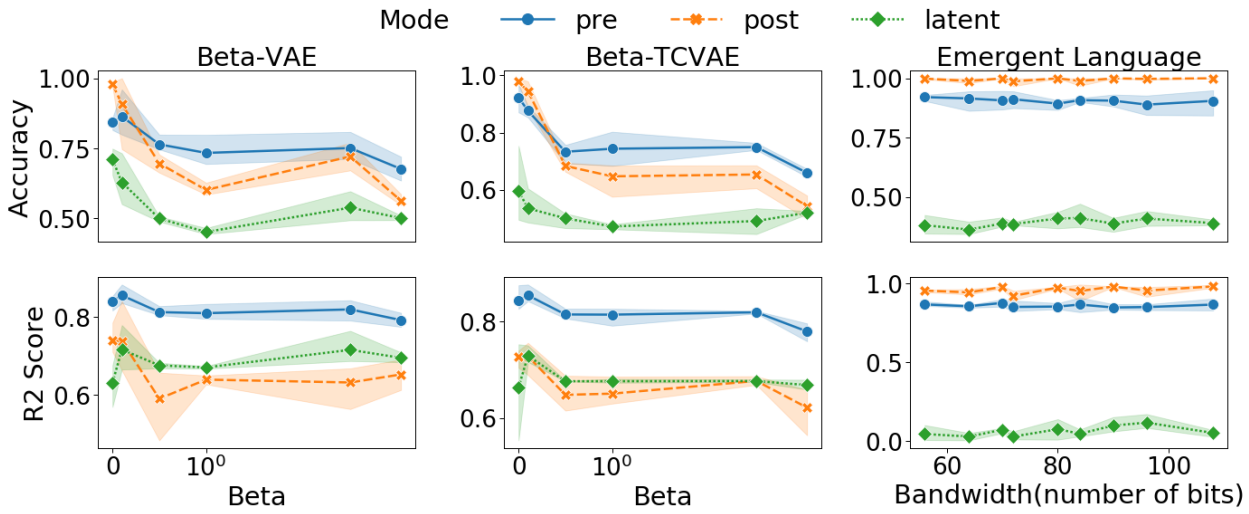


Figure 4.4: Generalization performance (accuracy for classification tasks and R2 score for regression tasks) of three representation models: β-VAE, β-TCVAE, and emergent language on dSprites.

4.4 Key Studies and Results

4.4.1 Compositional latent variables may not be the best representations for downstream tasks

We first study the generalization behaviors of different representation modes. For each unsupervised learning algorithm, we vary the hyperparameters that were designed to control the compositionality of representations. For β-VAE and β-TCVAE, we simply vary β. For emergent language models, we use the communication bandwidth, defined as the number of bits in the message and calculated by $\log_2(n_V^{n_{msg}})$, $n_{msg} \in \{8, 10, 12\}$ and $n_V \in \{128, 256, 512\}$. Figure4.4 shows the results for accuracy and R^2-score vs β and the number of bits when $N_{label} = 500$ for the dSprites dataset. We highlight the following observations:

Emergent language learning As expected, linear models do not work well on $\mathbf{z}_{\text{latent}}$ for EL models. $\mathbf{z}_{\text{post}}$ and $\mathbf{z}_{\text{pre}}$ are more suitable. $\mathbf{z}_{\text{post}}$ consistently outperforms $\mathbf{z}_{\text{pre}}$, which reveals that the emergent language bottleneck induces compositional latent structures useful for downstream tasks after being processed by the DecLSTM module.

Disentanglement learning Surprisingly, mismatched with the goal of disentanglement, $\mathbf{z}_{\text{latent}}$ in β-VAE and β-TCVAE is not the optimal choice for downstream tasks. For β -VAE and β -TCVAE, increasing β to decrease the bandwidth of the bottleneck (which was thought of as the source of disentanglement (Burgess et al., 2018)) *reduces* generalization performance. When β is larger, $\mathbf{z}_{\text{pre}}$ works best. For most cases, the representation of $\beta = 0$ is best. When $\beta = 0$, the regression task seems to favor $\mathbf{z}_{\text{pre}}$ while $\mathbf{z}_{\text{post}}$ performs better for classification tasks.

Implications. Both disentangled representation learning and emergent language learning try to induce compositionality through the inductive bias on the representation's compositional structure. The $\mathbf{z}_{\text{latent}}$ representation itself does not perform well as its structure may be too complex to be processed by simple linear models. But if $\mathbf{z}_{\text{latent}}$ can generalize well, its decoded version $\mathbf{z}_{\text{post}}$ will perform well. In EL models, $\mathbf{z}_{\text{post}}$ consistently performs better than $\mathbf{z}_{\text{pre}}$, which demonstrates the improved generalization with the language-like bottleneck. However, disentanglement models may learn useless disentangled representations since increasing the penalty on either the KL-term of the

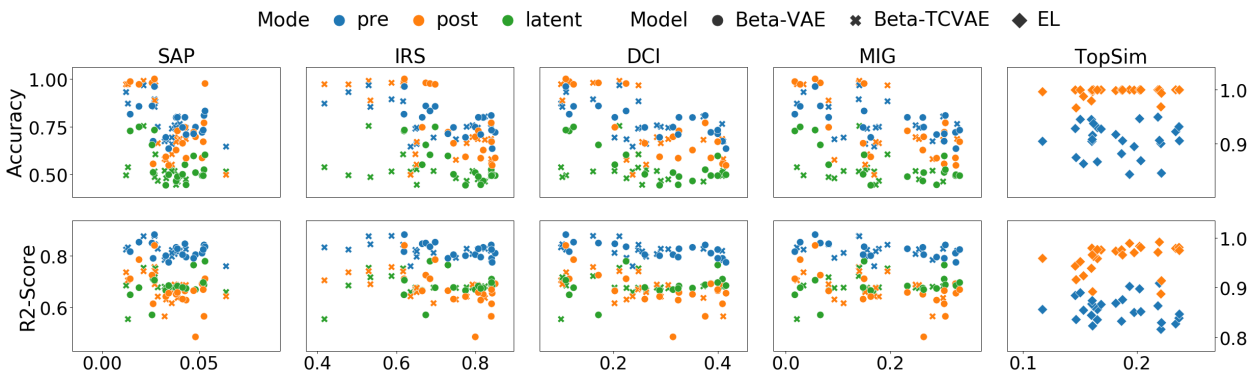


Figure 4.5: Compositionality metrics vs generalization performance in the dSprites dataset. The disentanglement metrics (SAP, IRS, DCI, MIG) of the β-VAE (dots) and β-TCVAE (crosses) models are not postively correlated with generalization performance in all the three representation modes. The compositionality metric for emergent language (EL), topographical similarity (TopSim), shows no strong correlation with generalization performance.

ELBO or the total correlation consistently lead to worse generalization performance. To further confirm this conclusion, we also measure the disentanglement metrics on the learned β-VAE and β-TCVAE models in the next section.

4.4.2 Compositionality Metrics May Not Represent Generalization Performance

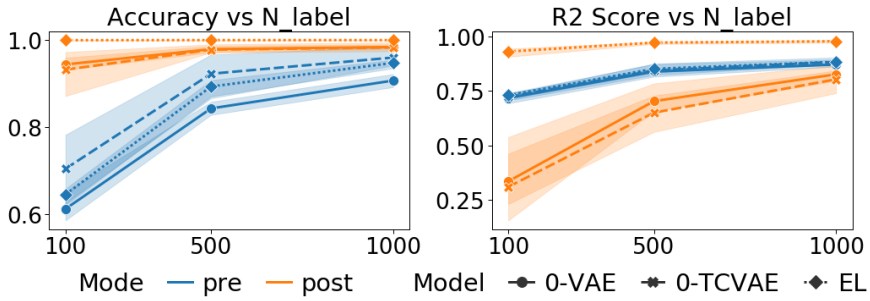


Figure 4.6: Generalization performance vs N_{label}. Three representation modes of β-VAE with β=0, β-TCVAE with β=0, and emergent language (EL) with n_V=256 are evaluated.

We further evaluate existing compositionality metrics on learned representations. For β-VAE, β-TCVAE with various β, we measure four common disetanglement metrics: Separated Attribute Predictability (SAP) Score (Kumar et al., 2018), Interventional Robustness Score (IRS) (Suter et al., 2019), Disentanglement-Completeness-Informativeness (DCI) (Eastwood and Williams, 2018), and Mutual Information Gap (MIG) (Chen et al., 2018) based on the implementation in disentanglement-

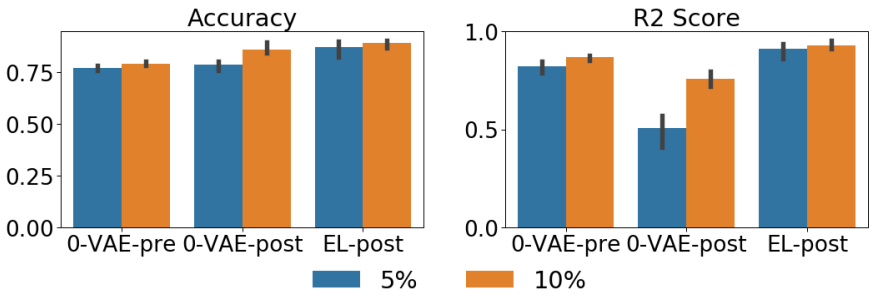


Figure 4.7: Generalization performance (with $N_{label} = 500$) of β-VAE with β=0, and Emergent language (EL) with n_V=256 on MPI3D-Real when using (5%) and (10%) unlabeled data.

lib (Locatello et al., 2019). The large-scale studies used by Locatello et al. 2019 (Locatello et al., 2019) broadly measured these metrics on a few datasets. For emergent language learning models, we vary the number of bits in the discrete messages and measure topographical similarity, a common compositionality metric for emergent languages that uses the (Spearman) correlation $\rho_{Spearman}$ between the pairwise distances of the inputs and the distances of the corresponding representations. We compute the cosine distance for input attributes and the editing distance for messages. Since we have the train/test data split for the unsupervised learning stage, we evaluate these metrics on both splits. The metrics on these two splits are similar, and we only show the ones for the train split.

We find that both β-VAE and β-TCVAE show positive correlations between β and the disentanglement metric scores, stronger in dSprites and weaker in MPI3D-Real where the highest scores usually occur for moderate β values. However, neither of the two datasets show the same pattern as in Figure 4.4 where generalization performance is negatively correlated with β. Therefore, for β-VAE and β-TCVAE, models with higher disentanglement scores do not achieve a better generalization performance. Figure 4.5 confirms the non-correlation or even negative correlation between compositionality metrics and generalization performance on the dSprites dataset. The topographical similarity of emergent language models does not show a strong correlation, e.g. $\rho_{Spearman} < 0.5$ with compositional generalization for both dSprites and MPI3D-Real datasets. The observations are consistent with some previous studies on emergent language (Chaabouni et al., 2020) and disentanglement (Locatello et al., 2019).

Implications. Existing compositionality/disentanglement metrics do not measure the generalization performance of learned representations. However, we should be careful about interpreting the results as compositionality in representation learning does not necessarily help compositional generalization. As these metrics were defined more or less based on the compositional annotations by humans, they can only measure particular types of compositionality. However, the compositionality exhibited in representation learning, especially with unsupervised algorithms, may not match with human definitions, e.g., a language where editing distance is a poor measure of similarity, and therefore cannot be captured by those metrics. Since designing generic compositionality is challenging or maybe even impossible, one should directly evaluate the compositional generalization if it is the ultimate goal.

4.4.3 Representations Learned by Emergent Language Models Generalize Better

In Fig 4.4, we observed some clues that "post" representations of EL models generalize consistently well on both tasks. In this section, we take a closer look at the generalization performance of representations learned by EL by comparing them with disentanglement models.

Varying the number of labeled samples for downstream learning. We evaluate the performance of downstream models trained with different numbers of labeled samples $N_{label} \in \{100, 500, 1000\}$. We compare three models: β-VAE and β-TCVAE with $\beta = 0$ (since $\beta = 0$ consistently performs best, as shown in Figure 4.4) and the EL model with $n_V = 256$. Figure 4.6 shows results for dSprites. When N_{label} is low, e.g. $N_{label} = 100$ and 500 for dSprites, z_{post} of EL consistently beats all other models in all representation modes for both classification (accuracy metric) and regression (R^2 score metric). More importantly, even when N_{label} is large, 0-VAE/0-TCVAE/AE models do not have a consistent representation mode that shows similar performance as z_{post} of the EL models: their z_{post} is worse at regression and z_{pre} is worse at classification.

Using less unlabeled data for unsupervised learning. Furthermore, we reduced the train-split ratio to 5% to test unsupervised learning algorithms with *less unlabeled data*. We compared the performance of 0-VAE and EL models on MPI3D-Real in Figure 4.7. With 5% unlabeled training data, the post representation of EL models outperforms the pre/post 0-VAE model at

both classfication or regression tasks and all N_{label} with an even larger margin. In other words, the generalization performance of representations of 0-VAE models degrades faster with reduced unlabeled data.

Implications. Representations learned by emergent language models generalize well even with limited unlabeled data for unsupervised learning and labeled samples for downstream learning.

4.4.4 Ablations on Emergent Language Models

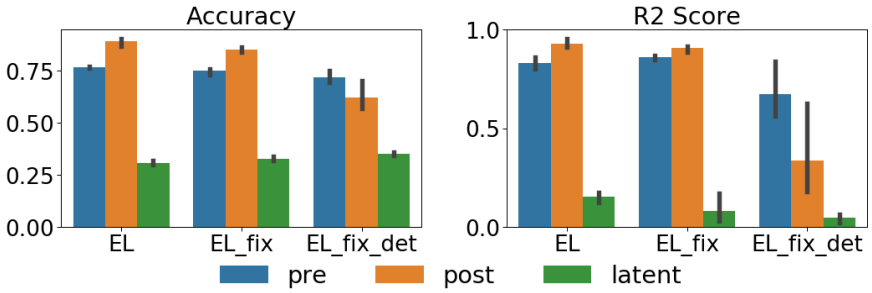


Figure 4.8: Ablation study of Emergent Language (EL) with n_V = 256 and $N_{label} = 500$ in MPI3D-Real using fixed-length messages (EL-fix) and greedy sampling (EL-fix-det).

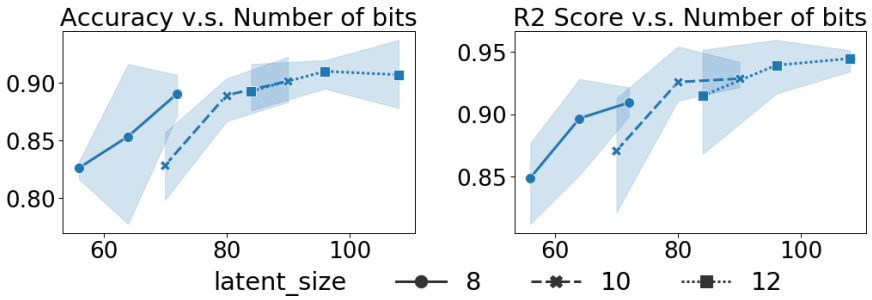


Figure 4.9: Generalization performance of EL models with z_{post} and $N_{label} = 500$ on MPI3D-Real dataset, for different bandwidths by varying message sizes $n_{msg} \in \{8, 10, 12\}$ and vocabulary sizes $n_V \in \{128, 256, 512\}$. Three n_{msg} are plotted as segments of different line styles with increasing n_V/bits.

Since EL models showed superior compositional generalization performance over disentanglement models, we conduct ablation studies on important hyperparameters and design choices of EL models on the MPI3D-real dataset. We first study the impact of bandwidth controlled by either length of the messages or the size of dictionary. We can see that increasing the bandwidth generally improves compositional generalization. In some cases, for similar bandwidth, shorter

sequences/a larger vocabulary is favored, e.g. 512^8 vs 128^{10}. Since the MPI3D-Real dataset only has $6 + 6 + 2 + 3 + 3 + 40 + 40 = 100$ distinct values for all generative factors, a simple choice of language can express the values of factors in a fixed order using $n_{\text{msg}} = 7$ and $n_{\text{V}} = 100$. Apparently this behavior is not learned by the current design of EL models possibly because of optimization issues due to poor gradient estimation for discretization, regardless of the good generalization performance. However, EL models can learn their own language with additional redundancies that require more bandwidth.

We also ablate the two characteristics of the EL models we have considered thus far: allowing variable-length messages and using stochastic sampling. We can turn an EL model into one with fixed-length messages by always generating n_{msg} tokens (EL-fix) and additionally with deterministic messages by using greedy sampling (EL-fix-det). We can see that as we remove these two designs, the generalization performance is worse, especially after making the model deterministic. Notably, the EL model with fixed-length messages and using deterministic sampling (EL-fix-det) is very close to the popular discrete representation learning model VQ-VAE (Van Den Oord et al., 2017) that uses a shared vocabulary over spatial locations of the feature map. Therefore, our study may indicate that the use of discrete latent variables of variable length and stochastic sampling can enable unsupervised learning models to learn better representations for downstream tasks. We also found that while the EL model allows variable-length codes, at convergence it still almost always uses the maximum message length on both the training and testing set. This is not surprising given that the reconstruction task drives the discrete message to be longer so that more information can pass through the discrete bottleneck.

4.5 Related Work

Compositional generalization was shown in previous work on disentangled representation learning (Esmaeili et al., 2019; Higgins et al., 2018) to generate images of novel combinations of concepts. Zhao et al. (Zhao et al., 2018) systematically evaluated generative models in a few compositional generalization tasks without exploring whether disentanglement is correlated with generalization performance. Recently, Montero et al. (Montero et al., 2021) studied the

compositional generalization along with the interpolation and extrapolation generalization of two VAE-based disentanglement models. Similar to (Zhao et al., 2018), the performance is evaluated on the unsupervised learning task, e.g. image reconstruction/generation. However, we evaluate the compositional generalization directly. Different from (Montero et al., 2021) that manually selects the train-test splits with various difficulty, we use random splits which is earlier for studying different split ratios.

The comprehensive study on unsupervised learning of disentangled representation by Locatello et al. (Locatello et al., 2019) included an evaluation on downstream tasks. However, they train unsupervised models on the whole dataset and therefore are not able to evaluate the generalization as we do. (Dittadi et al., 2021) evaluated the downstream out-of-distribution (OOD) generalization that includes interpolation and extrapolation generalization of unsupervised and weakly supervised disentanglement models, while we focus on compositional generalization. A very recent work (Schott et al., 2022) included compositional generalization behaviors of unsupervised and weakly supervised disentanglement models in their broad study on generalization. Unlike our evaluation that trains simple downstream models with a small amount of labeled samples, they use all labels in the train set to learn MLP models. Furthermore, none of these works evaluated the representations beyond the latent variables at the bottleneck layer as we did.

In emergent language (EL) learning, Chaabouni et al. (Chaabouni et al., 2020) found that regardless of the degree of compositionality, measured by topographic similarity, EL can generalize to novel combinations of concepts when the input space is rich enough. Andreas (Andreas, 2019) proposed a new compositionality metric, tree reconstruction error (TRE), measuring how well a representation model can be approximated by a compositional operator and learnable primitive representations. While claimed as a general compositionality metric with learnable compositional operator, some pre-commitment to a restricted composition function is essentially inevitable. Similar to (Chaabouni et al., 2020), (Andreas, 2019) also studied the relationship between TRE and generalization. While they work on attribute data and evaluate on tasks for learning emergent language, we used image data and are motivated by learning useful representations for downstream

tasks. Furthermore, different from all previous works, we studied the compositional generalization of representations from both disentanglement and emergent language models with a unified setup.

4.6 Limitations of Our Study

Our goal was to provide a thorough study of learning algorithms including their hyperparameters. However, our study is limited in the variety of other design choices to restrict the experimental complexity. While we studied both synthetic and realistic image datasets, both these datasets are relatively simple with the same small number of generative factors and each of the factor follows a uniform distribution. For learning algorithms, we focus on studying the inductive bias on the representation format while fixing the model architecture design which can impact the results. Moreover, we did not study hyperparameters beyond those related to the latent representations. Specifically, we did not study how the type and configurations of the optimizer, and the batch size would change the results, instead, we followed common setups in previous studies.

4.7 More Experimental Details

Model architectures. In our experiments, disentanglement models and emergent language (EL) models use the same architectures for the convolutional encoder and decoder. Disentanglement models use an MLP to encode the convolutional features into the disentangled latent variables and another MLP to decode them, while EL models use two LSTMs to encode and decode the discrete messages. Table 4.1 provides architecture details for all these modules.

Table 4.1: The encoder module architectures which are the symmetric reflection of the decoder layers.

EncConv	**EncMLP**	**EncLSTM**
4x4 Conv, 64 ReLU, stride 2	FC 512, ReLU	
4x4 Conv, 128 ReLU, stride 2	FC 1024, ReLU	LSTM 512
4x4 Conv, 128 ReLU, stride 2	FC 1024, ReLU	Linear n_{msg}
4x4 Conv, 128 ReLU, stride 2	FC 512, ReLU	
Flatten layer	Linear $\lvert z_{latent} \rvert$	

Readout Model We use the implementations in scikit-learn[1] (version 0.22) for the readout models of our downstream evaluation. The specific linear or gradient boosting tree (GBT) models for classification or regression tasks are configured as specified in Table 4.2.

Table 4.2: The implementation details of the readout models.

	Scikit-learn function	Configuration
Linear	linear_model.LogisticRegressionCV	default
	linear_model.RidgeCV	alphas=[0, 0.01, 0.1, 1.0, 10]
GBT	ensemble.GradientBoostingClassifier	default
	ensemble.GradientBoostingRegressor	default

Computational Cost. We use an Nvidia RTX A6000 to benchmark the computational cost. For unsupervised representation learning, training a β-VAE or a β-TCVAE model takes about 3 hours and 10 hours for dSprites and MPI3D-real respectively, and an EL model with $n_{msg} = 10$ takes about 11 and 15 hours respectively for dSprites and MPI3D-real .

Dataset License. The two datasets used in our experiments, dSprites[2] and MPI3D[3], are publicly available under an Apache License and the Creative Commons Public License respectively.

Reproducibility. Our code is publicly available.[4].

4.8 More Experimental Results

4.8.1 Sanity Check Experiments with Oracle Representations

We test the oracle representations using the ground truth value of all attributes or the squared value of each attribute (which would not be perfectly linearly fittable). On the dSprites dataset, we use 500 samples to train linear or GBT readout models for classification and regressions tasks. Their generalization performance is given in Table 4.3. The attribute values are expected to generalize perfectly with linear or GBT readout models. However, linear readout models can still fit the

[1] https://scikit-learn.org/

[2] https://github.com/deepmind/dsprites-dataset

[3] https://github.com/rr-learning/disentanglement_dataset

[4] https://github.com/wildphoton/Compositional-Generalization

non-linear attributes^2 well. We think this may be due to the limited value range of attributes of our datasets. This experiment shows that if the learned representation is disentangled into attributes (as is often the goal), the linear head should not be a major issue to constrain the generalization performance.

Table 4.3: Sanity check with oracle linear and non-linear representations. The accuracy/R2-score are reported for classfification and regression tasks respectively.

Representations	Linear	GBT
attributes	100% /100%	100% / 100%
attributes^2	94.7% / 100%	100% / 100%

4.8.2 A Closer Look of Generalization.

Our evaluation protocol includes two training stages: unsupervised pre-training of a representation model and supervised training of a read-out model with a small amount of labeled data. To better understand the generalization results, we further evaluate performance on the whole unsupervised training data (US-train) that is only visible to the representation model, and the supervised training data (S-train) that is part of the pre-training data and were seen by both the representation model and the readout model. The results on the held-out test set (Test) unseen by both training stages are given as a reference. In Table 4.4, we show the results of the evaluation of the pre / latent / post representations of EL and β-VAE(beta=0) with linear or GBT readout models on the dSprites dataset. The classification accuracy or regression R2 score is given in each entry.

- On all representation models/modes and read-out models, the performance of Unsup-train and test is very close. This tells us that an example seen by the unusupervised pretraining stage does not necessarily have good performance in a downstream task in our setting.

- In the S-train set, linear readout models under-fit the latent representations of both VAE and EL models. Bad performance is expected for EL-latent representations since a linear readout model is not a good choice for language-like messages as discussed in the paper.

Combined with sanity-checking experiment 4.8.1, it indicates that VAE models do not produce representations that disentangle attributes.

- GBT readout models fit EL-latent and VAE-latent well on S-train but generalize poorly to US-train and Test. This further indicates that latent representations perform worse in providing good generalization when compared to pre/post representations.

Table 4.4: Performance on three subsets of dSprites data. Pre/latent/post representations of EL and β-VAE(beta=0) are evaluated with linear and GBT readout models. The classification accuracy / regression R2 score is given in each entry.

Data + Readout	VAE-Pre	VAE-Latent	VAE-Post	EL-Pre	EL-Latent	EL-Post
S-train + Linear	99.93/94.48	73.27/64.83	100/95.99	100/92.55	46.13/13.35	100/99.39
US-train + Linear	84.83/84.15	70.89/63.13	98.74/71.29	91.02/84.7	39.11/9.92	99.99/98.04
Test + Linear	84.3/83.9	70.98/63.1	97.88/73.98	90.56/84.6	38.8/9.5	99.94/97.85
S-train + GBT	100/96.44	99.33/93.64	100/98.81	100/95.86	96.93/83.63	100/99.83
US-train + GBT	77.67/79.44	76.8/77.57	97.73/88.34	81.58/81.02	56.19/58.53	99.53/95.83
Test + GBT	76.76/78.95	76.09/77.26	96.83/87.82	80.86/90.56	54.84/57.36	99.52/95.68

4.8.3 Detailed Results with GBT read-out models

In Section 4.4, we only present representative results to support our key findings. In this section, we provide detailed results for different algorithms and datasets with additional gradient-boosting-tree (GBT) read-out models that are able to model non-linear mappings.

Compositional latent variables may not be the best representations for downstream tasks. Figure 4.10 compares pre, $latent$, and $post$ representation modes for three different learning models (β-VAE, β-TCVAE, and emergent language (EL)) using different read-out models. For emergent language (EL) models, when using GBT instead of linear models for downstream tasks, $\mathbf{z}_{\text{latent}}$ performance improves, but still underperforms $\mathbf{z}_{\text{pre}}$ and $\mathbf{z}_{\text{post}}$. For disentanglement models, increasing the regularization (β) still decreases the performance when using GBT read-out models. However, when $\beta = 0$, the regression task no longer favors $\mathbf{z}_{\text{pre}}$ and $\mathbf{z}_{\text{post}}$ performs well for both the regression and classification tasks.

Compositionality Metrics May Not Represent Generalization Performance. Figure 4.11 shows the disentanglement/compositionality metrics vs generalization performance on the dSprites and MPI3D-Real datasets using linear and GBT read-out models. Consistently to our results in the main article, we do not observe strong correlations between these metrics and generalization performance. Figs. 4.12 and 4.13 show quantitative measures of ranking correlation. We see that for disentanglement models, all metrics show no or negative correlations with generalization performance except for a weak correlation between the DCI score and generalization on the MPI3D-Real dataset. For EL models, $post$ representations show stronger, although still weak, correlations than pre representations.

Representations Learned by Emergent Language Models Generalize Better. Figure 4.14 compares EL models with β-VAE and β-TCVAE with $\beta = 0$. We see that $\mathbf{z}_{\text{pre}}$ of EL using linear read-out models gives the best performance overall especially when N_{label} is small. While applying GBT read-out models improves the performance of $\mathbf{z}_{\text{post}}$/$\mathbf{z}_{\text{pre}}$ over the 0-VAE and 0-TCVAE models and $\mathbf{z}_{\text{pre}}$ from the EL models in regression tasks, GBT read-out reduces performance in other cases, especially for classification tasks. The reason may be that GBT models need more labeled samples than linear models for training to work well. When we reduce the train-split ratio in Figure 4.15, the EL model learns more generalizable representations than the 0-VAE and 0-TCVAE models which degrade faster with reduced unlabeled data.

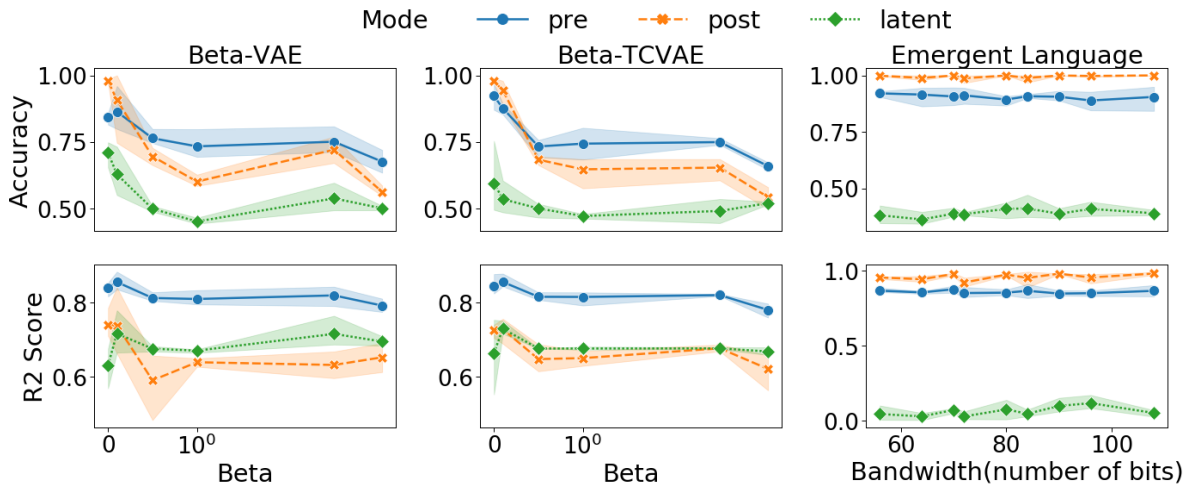
Ablations on Emergent Language Models. From the ablation study of EL models in Figs. 4.16 and 4.17, we observe consistent patterns: using a shorter sequence with a larger vocabulary size works better; using greedy sampling significantly reduces the performance of EL models.

4.9 Conclusions and Discussions

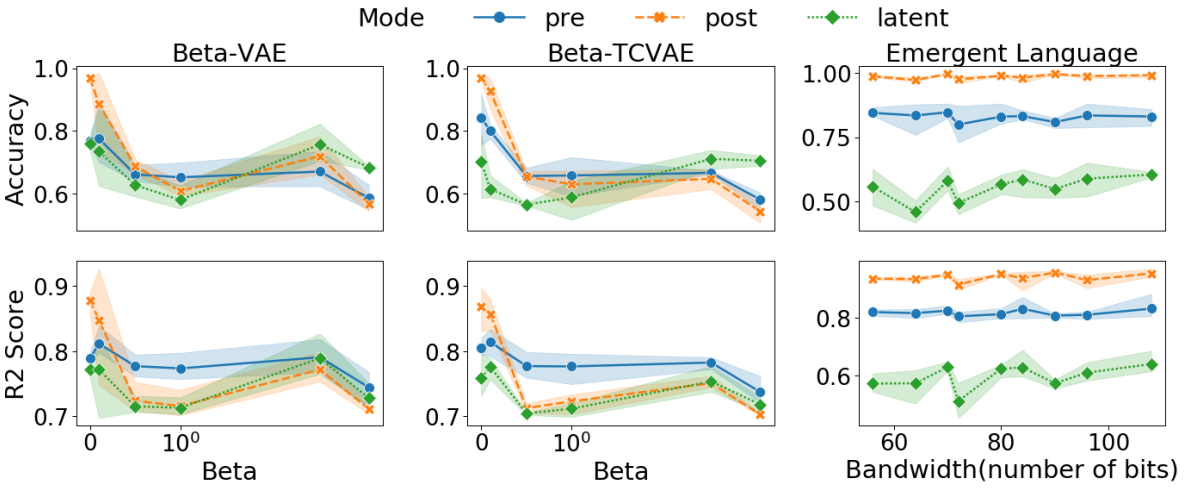
We proposed a protocol to evaluate the compositional generalization of unsupervised representation learning models that have a built-in inductive bias for compositionality: disentanglement models and emergent language (EL) learning. Our evaluation emphasizes using a small number of labeled samples to train simple models for downstream tasks. The interesting finding that latent

variables at the bottleneck do not work as well as "pre" and "post" representations reminds us to be careful when concluding that a model performs poorly when its latent representation does not perform well. For disentanglement models, we observe that generalization performance is not well correlated with existing disentanglement metrics. This finding demonstrates the gap between pursuing better disentanglement metrics and more generalizable representations in previous studies. Similarly, the existing compositionality metrics for EL, e.g. topographical similarity, cannot represent the generalization of learned representations. However, under the same setup, EL models which were not initially proposed for unsupervised representation learning induce representations with surprisingly strong compositional generalization and are robust to the change of dataset and hyperparameters.

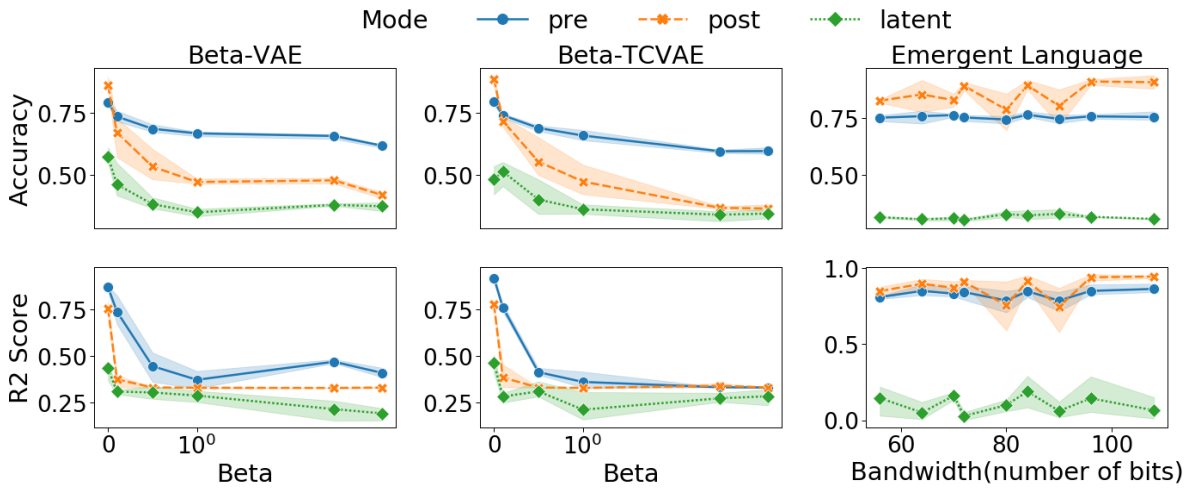
We focus on unsupervised learning algorithms that are specifically designed for compositional representation to answer whether they lead to better generalization (which was an assumption made by a great deal of prior work). A broader evaluation on other representation learning approaches e.g. self-supervised learning (Chen et al., 2020; Grill et al., 2020) would be interesting for future work. We hope that our study draws more attention to the study of compositional generalization and emergent language models as a way of learning representations. Interesting future directions include working on more challenging datasets, e.g. images of multiple objects; testing on other downstream tasks, e.g. visual question answering; testing different model architectures, e.g. Transformers (Vaswani et al., 2017); and developing better pre-training tasks beyond auto-encoding. Recent work showing that emergent language (z_{latent}) can be used as a corpus to pre-train a language model (Yao et al., 2022) also suggests that emergent language may be a better visual representation choice in vision-language models.



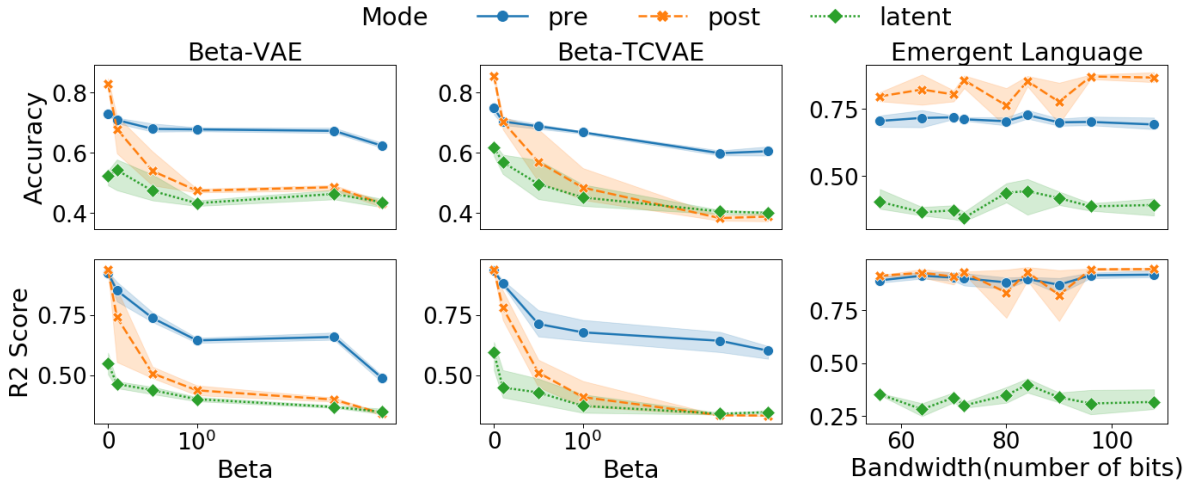
(a) Results on dSprites with linear read-out models.



(b) Results on dSprites with GBT read-out models.

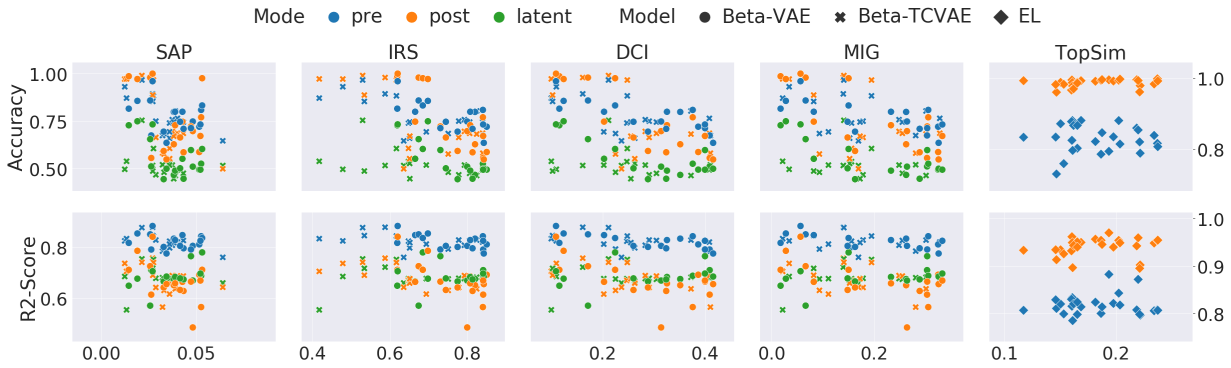


(c) Results on MPI3D-Real with linear read-out models.

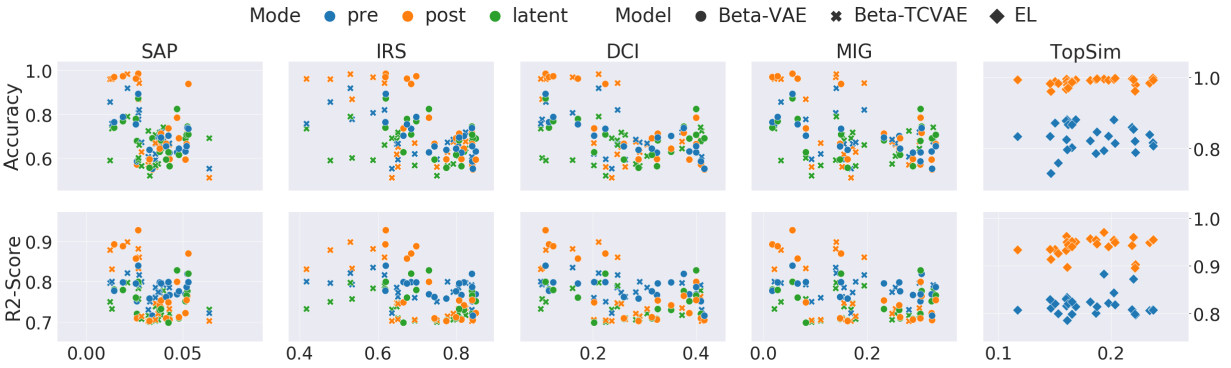


(d) Results on MPI3D-Real with GBT read-out models.

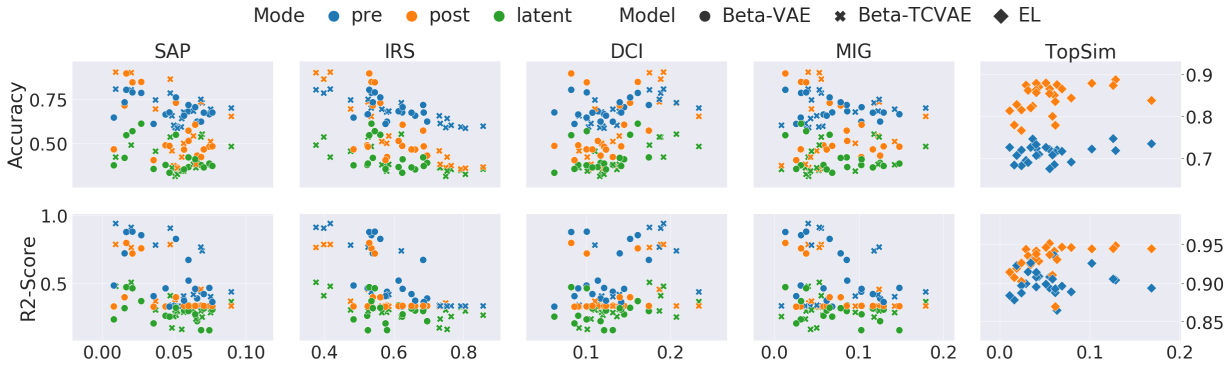
Figure 4.10: Generalization performance (accuracy for classification and R2 score for regression) with $N_{label} = 500$ of three representation models: β-VAE, β-TCVAE, and emergent language (EL) varying hyper-parameters (β or bandwidth), datasets (dSprites and MPI3D-Real) and read-out model (linear and GPT).



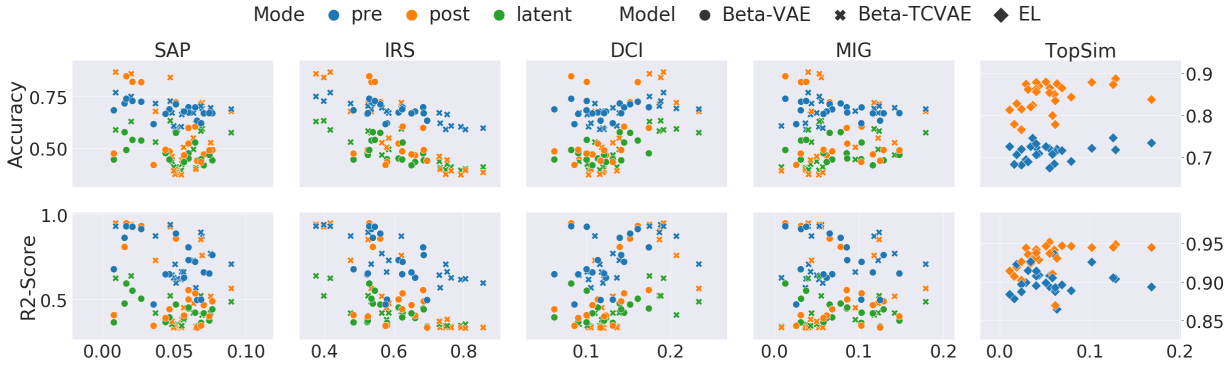
(a) Results on dSprites with linear read-out models.



(b) Results on dSprites with GBT read-out models.



(c) Results on MPI3D-Real with linear read-out models.



(d) Results on MPI3D-Real with GBT read-out models.

Figure 4.11: Compositionality metrics vs generalization performance on dSprites and MPI3D-Real datasets, and linear and GPT read-out models. The disentanglement metrics (SAP, IRS, DCI, MIG) of the β-VAE (dots) and β-TCVAE (crosses) models are not positively correlated with generalization performance in all the three representation modes. The compositionality metric for emergent language (EL), topographical similarity (TopSim), shows no strong correlation with generalization performance.

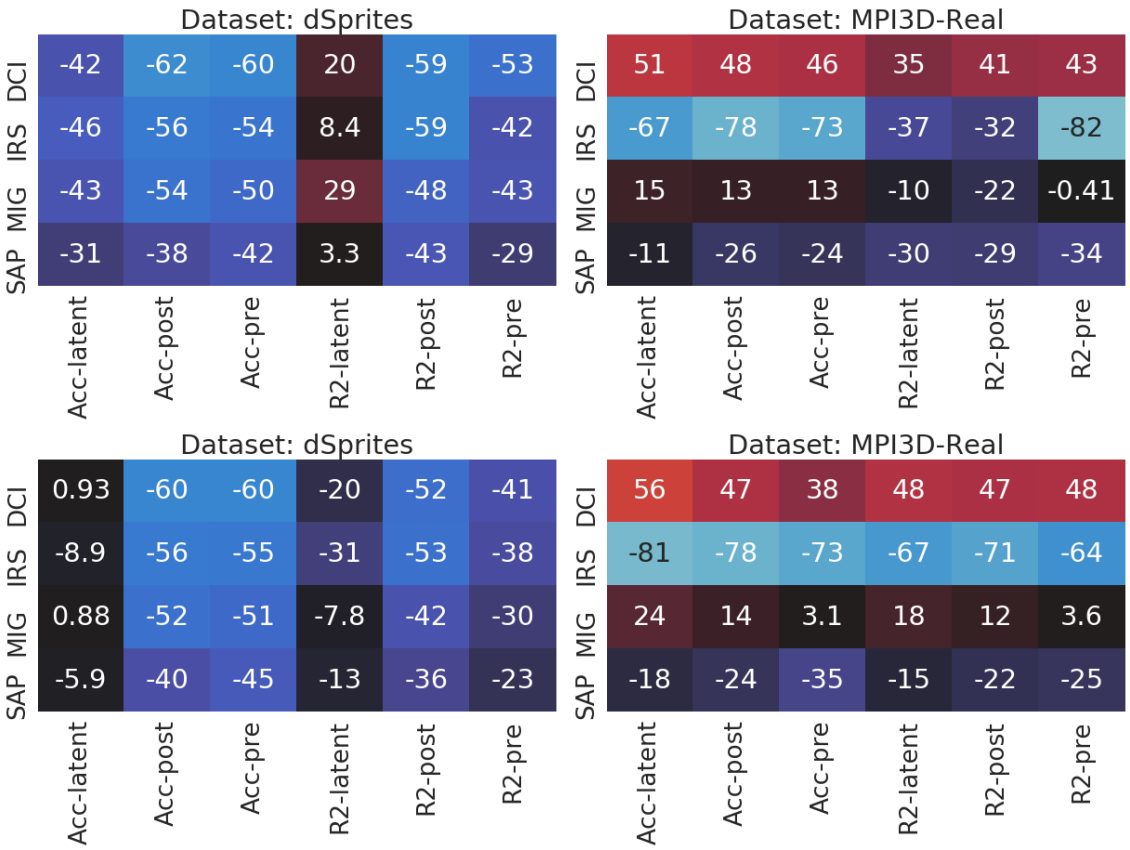


Figure 4.12: Ranking correlation between disentanglement scores (SAP, IRS, DCI, MIG) and the generalization performance of three representation modes (pre, latent, post) using linear (the first row) and GBT (the second row) read-out models on dSprites (the left column) and MPI3D-Real (the right column) datasets. Except for the DCI metric, which shows weak correlations with generalization performance on MPI3D-Real, all other metrics show no or even negative correlations.

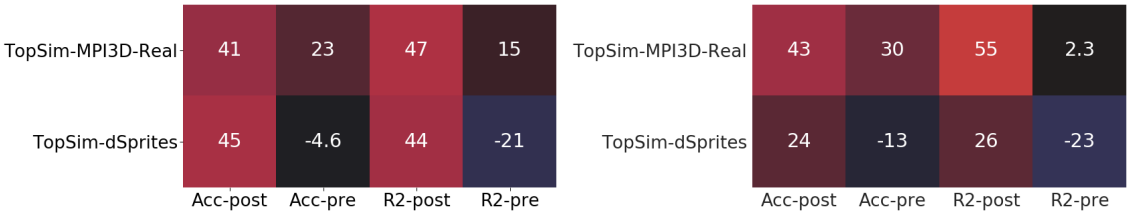
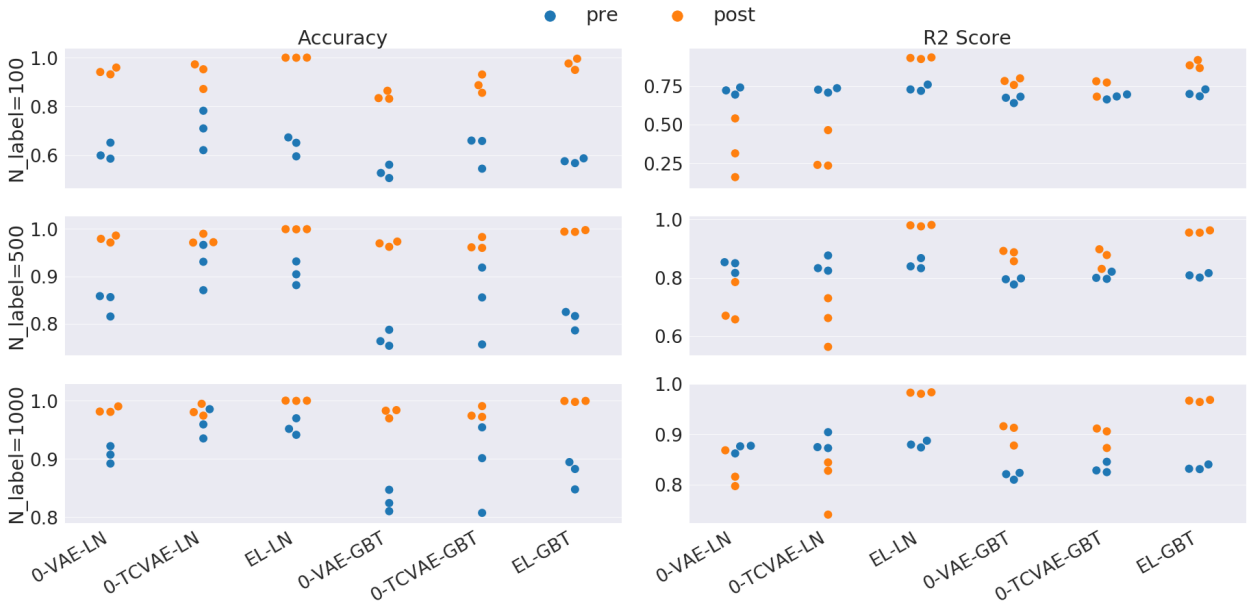
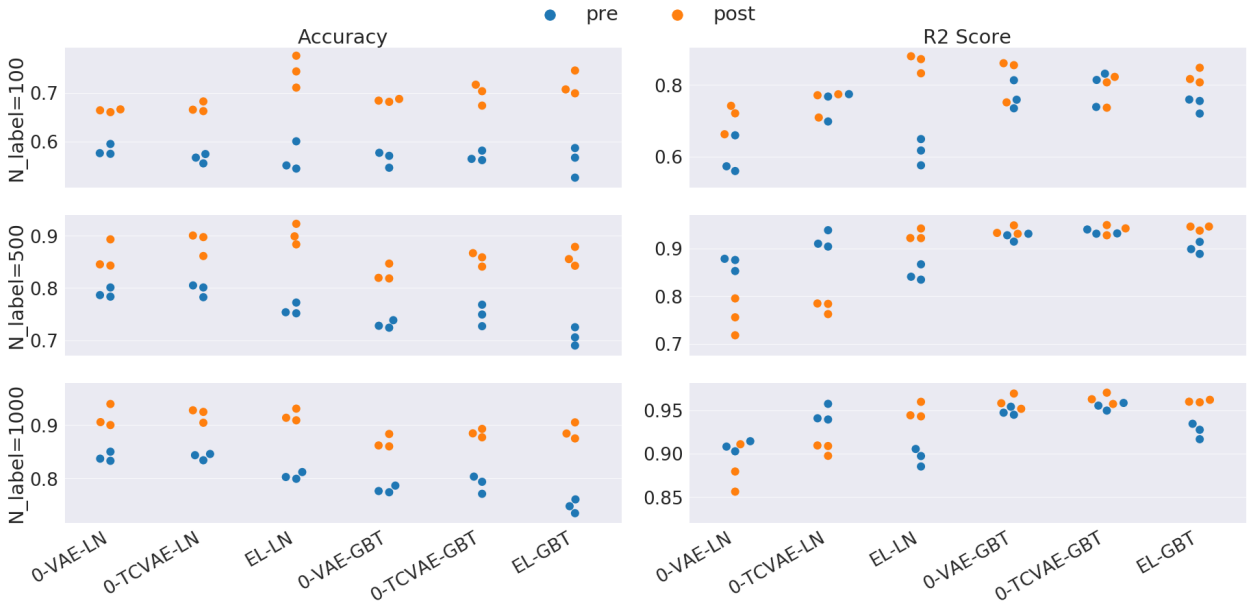


Figure 4.13: Ranking correlation between topographical similarity (TopSim) and the generalization performance of three representation modes (pre, latent, post) using linear (the left column) and GBT (the right column) read-out models on dSprites (the second row) and MPI3D-Real (the first row) datasets. The correlations between TopSim and generalization are stronger on $post$ representations than on the pre representations.



(a) Results on dSprites dataset.



(b) Results on MPI3D-Real dataset.

Figure 4.14: Generalization performance of two representation modes (pre, post) of β-VAE with β=0, β-TCVAE with β=0, and emergent language (EL) with n_V=512 when evaluated with linear (LN) and gradient boosting tree (GBT) read-out models.

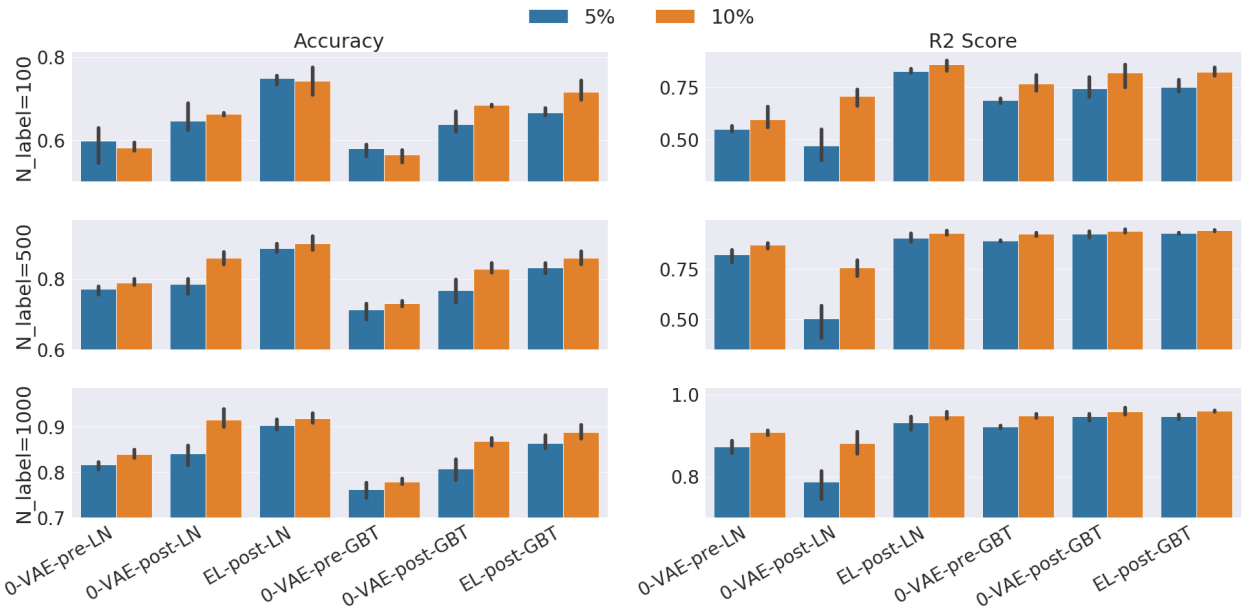
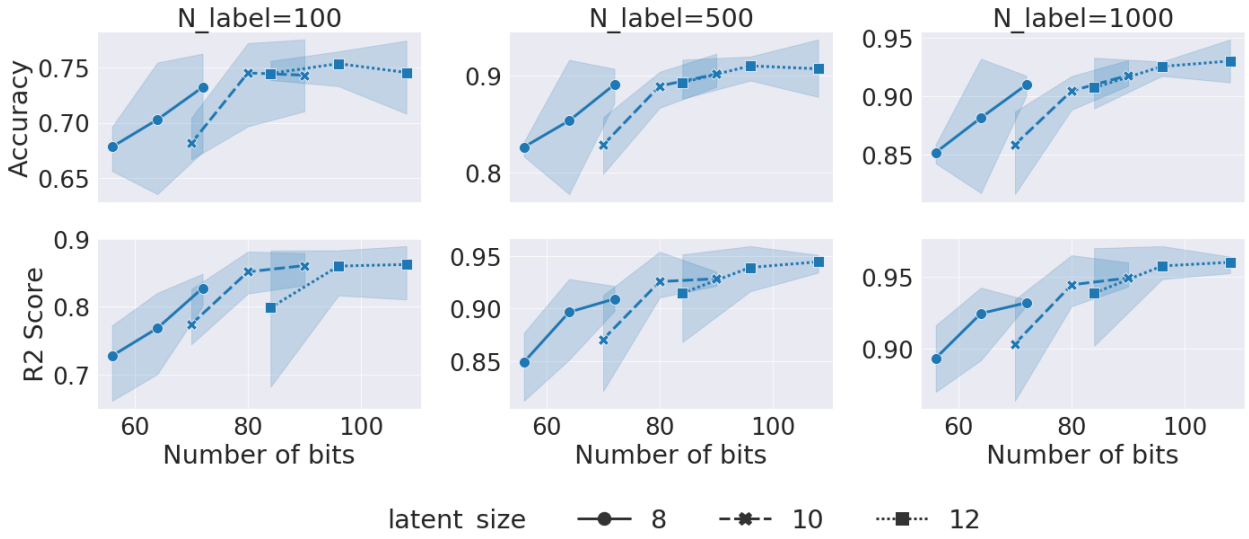
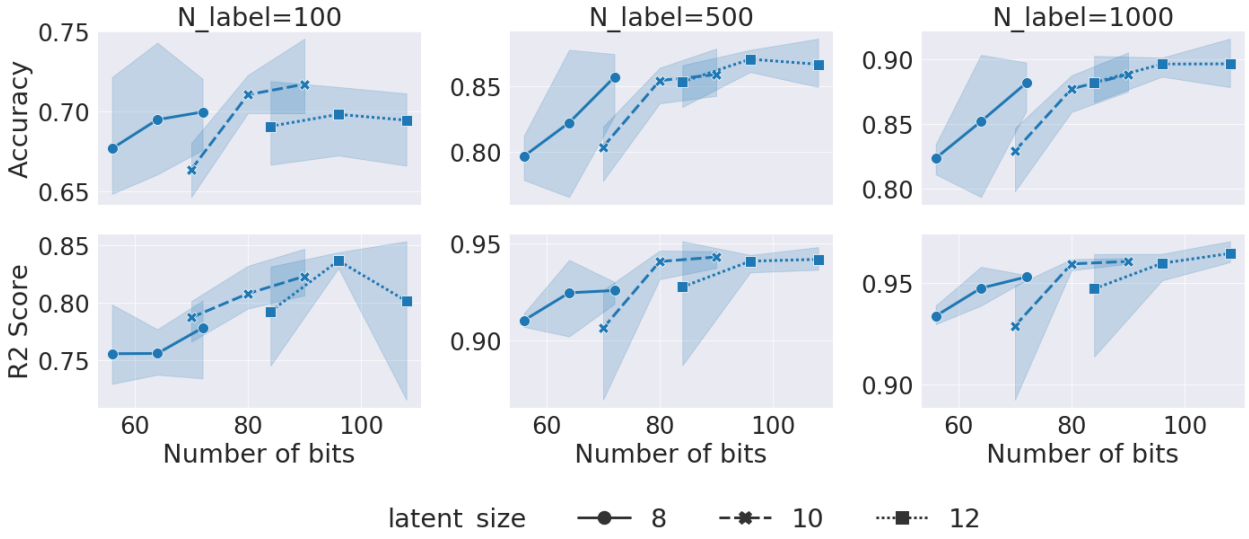


Figure 4.15: Generalization performance of pre and $post$ representations of β-VAE with β=0 (0-β-VAE) and β-TCVAE with β=0 (0-β-TCVAE), and $post$ representations of emergent language (EL) with n_V=512 when evaluated with linear (LN) and gradient boosting tree (GBT) read-out models on MPI3D-Real *when using (5%) and (10%) unlabeled data.*

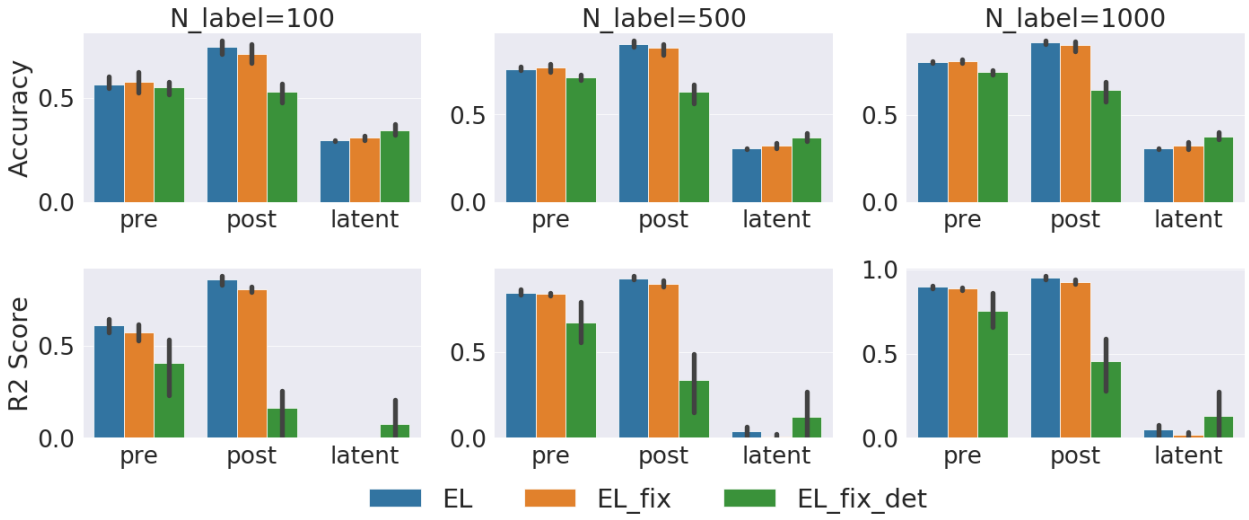


(a) Results using linear read-out models.

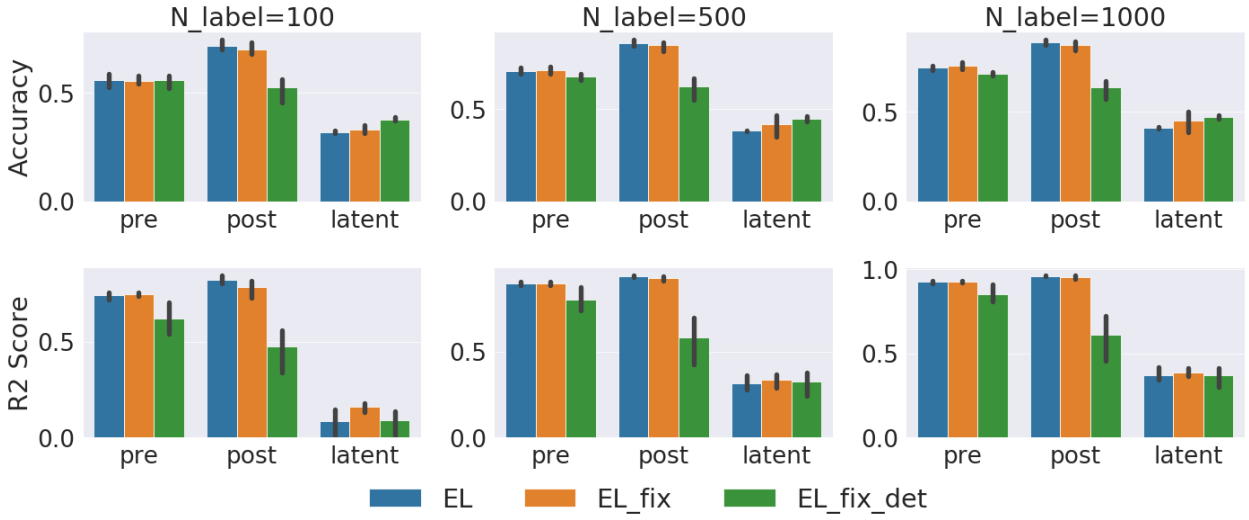


(b) Results using GBT read-out models.

Figure 4.16: Ablation study of Emergent Language (EL) models of *different bandwidths* by varying message sizes $n_{msg} \in \{8, 10, 12\}$ and vocabulary sizes $n_V \in \{128, 256, 512\}$ with z_{post} and $N_{label} = 100/500/1000$ on MPI3D-Real dataset evaluated with (a) linear and (b) gradient boosting tree read-out models, The three n_{msg} results are plotted as segments of different line styles with increasing n_V/bits.



(a) Results using linear read-out models.



(b) Results using GBT read-out models.

Figure 4.17: Ablation study of Emergent Language (EL) when *using fixed-length messages (EL-fix) and greedy sampling (EL-fix-det)*, with n_V = 512 and $N_{label} = 100/500/1000$ for MPI3D-Real dataset.

CHAPTER 5: Summary and Future Work

5.1 Summary of Contributions

To summarize, let me first revisit my thesis statement here:

Injecting prior knowledge into deep neural networks by means of inductive biases on task relations, model biases, representation format, can improve data efficiency, robustness, and generalization for deep visual learning.

I present following contributions in this thesis:

1. *I developed* `DeepAtlas`*, a joint learning framework for image registration and segmentation that can learn DNNs for both tasks from unlabeled images and a few labeled images.*

   The two networks mutually guide each other's training on unlabeled images via an anatomy similarity loss which penalizes the dissimilarity between the warped segmentation of the moving image and the segmentation of the target image. When registering image pairs consisting of a manually labeled image and a image pseudo-labeled by the prediction of the segmentation network, this loss provides anatomy consistency supervision for registration and forces the predicted segmentation to match the manual segmentation after registration. For both bone/cartilage structures in knee MRIs and cortical structures in brain MRIs, `DeepAtlas` shows large improvements over separately learned networks. `DeepAtlas` provides one-shot segmentation learning and greatly improves registration. This demonstrates that one task can benefit from imperfect supervision on unlabeled data provided by the other task.

2. *I proposed* `RandConv`*, a data augmentation technique that applies a random convolution layer on images during training to improve the generalization performance of a DNN in the presence of domain shift and robustness to image corruptions.*

`RandConv` is based on the observation that the bias towards local texture may hurt the generalization performance of DNN models. I theoretically justified the approximate shape-preserving property of `RandConv` and develop `RandConv` techniques with multi-scale and mixing designs, as well as a consistency loss to further enforce texture invariance. I validated `RandConv` and its mixing variant in extensive experiments on synthetic and real-world benchmarks as well as on the large-scale ImageNet dataset. `RandConv` outperforms single domain generalization approaches by a large margin on digit recognition datasets and for the challenging case of generalizing to the Sketch domain in PACS and to ImageNet-Sketch. I also explored if the robustness/generalizability of a pretrained representation can transfer. By finetuning a model pretrained with RandConv on PACS, the generalizability of a pretrained model may transfer to and benefit a new downstream task.

3. *I presented a comprehensive study of `compositional generalization` in unsupervised representation learning that includes disentanglement and emergent language models.* I proposed an evaluation protocol that focuses on whether or not it is easy to train a simple model on top of the learned representation that generalizes to new combinations of compositional factors. I systematically studied three unsupervised representation learning algorithms with compositional representation inductive biases. I find that directly using the bottleneck representation may lead to worse generalization than using representations from layers before or after the learned representation itself. Additionally, the previously proposed metrics to evaluate the levels of compositionality are not correlated with the actual compositional generalization in our framework. Surprisingly, I find that increasing pressure to produce a disentangled representation (e.g., increasing β in the β-VAE) produces representations with worse generalization, whereas representations from EL models show strong compositional generalization. My results shed new light onto the compositional generalization behavior of different unsupervised learning algorithms with a new setting to rigorously test this behavior, and suggest the potential benefits of developing EL learning algorithms for more generalizable representations.

5.2 Discussion and Future Work

Finally, in this section, I present an outlook on possible future work.

5.2.1 Joint Image Registration and Segmentation with Limited Labeled data

To further improve `DeepAtlas`, introducing uncertainty measures for segmentation and registration networks can help alleviate the effect of poor predictions of one network on the other. It would also be of interest to investigate multitask learning via layer sharing for the segmentation and registration networks, e.g. sharing the image encoder. This may further improve performance and decrease the size of the model. Furthermore, scaling up the training data by merging datasets of different organs and imaging modality may improve the performance and generalization for both tasks.

5.2.2 Domain Generalization and Robustness

`RandConv` can help computer vision tasks when a shape-biased model is helpful e.g. for object detection. `RandConv` can also provide a shape-biased pretrained model to improve performance in downstream tasks when generalizing to unseen domains. However, local texture features can be useful for many computer vision tasks, especially for fine-grained visual recognition in fixed domain. In such cases, visual representations that are invariant to local texture may hurt in-domain performance. Therefore, important future work includes learning representations that disentangle shape and texture features and building models to use such representations in an explainable way. Recent work on vision and language learning (Radford et al., 2021) that allows one to explicitly define a target domain with natural language shows promising results along this direction.

5.2.3 Compositionality Generalization

My study focused on unsupervised learning algorithms that are specifically designed for compositional representation to answer whether they lead to better generalization (which was an assumption made by a great deal of prior work). A broader evaluation on other representation learning approaches e.g. self-supervised learning (Chen et al., 2020; Grill et al., 2020) would be

interesting for future work. I hope that my study draws more attention to the study of compositional generalization and emergent language models as a way of learning representations due to the observed superior generalization behavior. Interesting future directions include working on more challenging datasets, e.g. images of multiple objects; testing on other downstream tasks, e.g. visual question answering; testing different model architectures, e.g. Transformers (Vaswani et al., 2017); and developing better pre-training tasks beyond auto-encoding. Recent work showing that emergent language can be used as a corpus to pretrain a language model (Yao et al., 2022) also suggests that emergent language may be a better visual representation choice in vision-language models. Inspired by recent advance on large scale vision-language model (Radford et al., 2021) and the shown compositional behavior (Goh et al., 2021), using language supervision to help to learn compositional representation with both good generalization and interpretability would be exciting future direction.

BIBLIOGRAPHY

Adams, R. and Bischof, L. (1994). Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647.

Aljabar, P., Heckemann, R. A., Hammers, A., Hajnal, J. V., and Rueckert, D. (2009). Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. *Neuroimage*, 46(3):726–738.

Ambellan, F., Tack, A., Ehlke, M., and Zachow, S. (2019). Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative. *MedIA*, 52(2):109 – 118.

Andreas, J. (2019). Measuring compositionality in representation learning. In *International Conference on Learning Representations*.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 998–1008.

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. (2018). An unsupervised learning model for deformable medical image registration. In *CVPR*, pages 9252–9260.

Balakrishnan, G., Zhao, A., Sabuncu, M. R., Guttag, J., and Dalca, A. V. (2019). Voxelmorph: a learning framework for deformable medical image registration. *IEEE TMI*.

Bengio, Y., Léonard, N., and Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059.

Beucher, S. (1982). Watersheds of functions and picture segmentation. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 1928–1931. IEEE.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239.

Brighton, H. and Kirby, S. (2006). Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242.

Bro-Nielsen, M. and Gramkow, C. (1996). Fast fluid registration of medical images. In *International conference on visualization in biomedical computing*, pages 265–276. Springer.

Broit, C. (1981). *Optimal registration of deformed images*. University of Pennsylvania.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. (2019). Exploration by random network distillation. In *International Conference on Learning Representations*.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*.

Carlucci, F. M., D'Innocente, A., Bucci, S., Caputo, B., and Tommasi, T. (2019). Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., and Baroni, M. (2020). Compositionality and generalization in emergent languages. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4427–4442. Association for Computational Linguistics.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31, pages 2610–2620. Curran Associates, Inc.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Chrupała, G., Kádár, Á., and Alishahi, A. (2015). Learning language through pictures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 112–118.

Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D., and DiCarlo, J. J. (2020). Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *Advances in Neural Information Processing Systems*, 33.

Denker, J. S., Gardner, W., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., Baird, H. S., and Guyon, I. (1989). Neural network recognizer for hand-written zip code digits. In *Advances in neural information processing systems*, pages 323–331.

Dhanachandra, N., Manglem, K., and Chanu, Y. J. (2015). Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771.

Dittadi, A., Träuble, F., Locatello, F., Wuthrich, M., Agrawal, V., Winther, O., Bauer, S., and Schölkopf, B. (2021). On the transfer of disentangled representations in realistic settings. In *International Conference on Learning Representations*.

Eastwood, C. and Williams, C. K. I. (2018). A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and Meent, J.-W. (2019). Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Fidler, S. and Leonardis, A. (2007). Towards scalable representations of object categories: Learning a hierarchy of parts. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Fodor, J. A. (1975). *The language of thought*, volume 5. Harvard university press.

Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Gaier, A. and Ha, D. (2019). Weight agnostic neural networks. In *Advances in Neural Information Processing Systems*, pages 5364–5378.

Ganin, Y. and Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M. (2016). Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430.

Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. (2021). Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30.

Gondal, M. W., Wuthrich, M., Miladinovic, D., Locatello, F., Breidt, M., Volchkov, V., Akpo, J., Bachem, O., Schölkopf, B., and Bauer, S. (2019). On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Havrylov, S. and Titov, I. (2017). Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in neural information processing systems*, pages 2149–2159.

He, K., Wang, Y., and Hopcroft, J. (2016a). A powerful generative model using random weights for the deep image representation. In *Advances in Neural Information Processing Systems*, pages 631–639.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Heeger, D. J. and Bergen, J. R. (1995). Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. (2020a). The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2020b). Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). β-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Higgins, I., Sonnerat, N., Matthey, L., Pal, A., Burgess, C. P., Bošnjak, M., Shanahan, M., Botvinick, M., Hassabis, D., and Lerchner, A. (2018). Scan: Learning hierarchical compositional visual concepts. In *International Conference on Learning Representations*.

Hoffman, D. D. and Richards, W. A. (1984). Parts of recognition. *Cognition*, 18(1-3):65–96.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in neural information processing systems*, 28.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Klein, A. and Tourville, J. (2012). 101 labeled brain images and a consistent human cortical labeling protocol. *Frontiers in Neuroscience*, 6:171.

Kornblith, S., Shlens, J., and Le, Q. V. (2019). Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671.

Kumar, A., Sattigeri, P., and Balakrishnan, A. (2018). VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS. In *International Conference on Learning Representations*.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282?289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Lazaridou, A., Pham, N. T., and Baroni, M. (2016). Towards multi-agent communication-based language learning. *arXiv preprint arXiv:1605.07133*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, K., Lee, K., Shin, J., and Lee, H. (2020). Network randomization: A simple technique for generalization in deep reinforcement learning. In *International Conference on Learning Representations. https://openreview. net/forum*.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, H., Jialin Pan, S., Wang, S., and Kot, A. C. (2018b). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., and Tao, D. (2018c). Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639.

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *MedIA*, 42:60–88.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124.

Luo, Y., Zheng, L., Guan, T., Yu, J., and Yang, Y. (2019). Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Mahapatra, D. and Sun, Y. (2010). Joint registration and segmentation of dynamic cardiac perfusion images using MRFs. In *MICCAI*, pages 493–501. Springer.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. (2019). Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. (2017). dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*.

Montero, M. L., Ludwig, C. J., Costa, R. P., Malhotra, G., and Bowers, J. (2021). The role of disentanglement in generalisation. In *International Conference on Learning Representations*.

Mu, N. and Gilmer, J. (2019). Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8617–8629.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

Ott, P. and Everingham, M. (2011). Shared parts for deformable part-based models. In *CVPR 2011*, pages 1513–1520. IEEE.

Pandey, M. and Lazebnik, S. (2011). Scene recognition and weakly supervised object localization with deformable part-based models. In *2011 International Conference on Computer Vision*, pages 1307–1314. IEEE.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019a). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415.

Peng, X., Huang, Z., Sun, X., and Saenko, K. (2019b). Domain agnostic learning with disentangled representations. In *ICML*.

Pohl, K. M., Fisher, J., Grimson, W. E. L., Kikinis, R., and Wells, W. M. (2006). A Bayesian model for joint segmentation and registration. *NeuroImage*, 31(1):228–239.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70.

Purushwalkam, S., Nickel, M., Gupta, A., and Ranzato, M. (2019). Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602.

Qiao, F., Zhao, L., and Peng, X. (2020). Learning to learn single domain generalization. *arXiv preprint arXiv:2003.13216*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Raghupathi, W. and Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health information science and systems*, 2(1):3.

Ren, Y., Guo, S., Labeau, M., Cohen, S. B., and Kirby, S. (2020). Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer.

Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L. G., Leach, M. O., and Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: application to breast mr images. *IEEE TMI*, 18(8):712–721.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. (2020). Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*.

Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B., and Ng, A. Y. (2011). On random weights and unsupervised feature learning. In *ICML*.

Schott, L., Kügelgen, J. V., Träuble, F., Gehler, P. V., Russell, C., Bethge, M., Schölkopf, B., Locatello, F., and Brendel, W. (2022). Visual representation learning does not generalize strongly within the same domain. In *International Conference on Learning Representations*.

Shao, R., Lan, X., Li, J., and Yuen, P. C. (2019). Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10023–10031.

Shen, W. B., Xu, D., Zhu, Y., Guibas, L. J., Fei-Fei, L., and Savarese, S. (2019). Situational fusion of visual representation for visual navigation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2881–2890.

Shi, B., Zhang, D., Dai, Q., Zhu, Z., Mu, Y., and Wang, J. (2020). Informative dropout for robust representation learning: A shape-bias perspective. In *International Conference on Machine Learning*.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3.

Sokooti, H., Vos, B. d., Berendsen, F., Lelieveldt, B. P., Išgum, I., and Staring, M. (2017). Nonrigid image registration using multi-scale 3d convolutional neural networks. In *International conference on medical image computing and computer-assisted intervention*, pages 232–239. Springer.

Stone, A., Wang, H., Stark, M., Liu, Y., Scott Phoenix, D., and George, D. (2017). Teaching compositionality to cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5058–5067.

Sun, B. and Saenko, K. (2014). From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Suter, R., Miladinovic, D., Schölkopf, B., and Bauer, S. (2019). Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR.

Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J. N., Wu, Z., and Ding, X. (2020). Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693.

Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., and Ross, C. (2022). Winoground: Probing vision and language models for visio-linguistic compositionality. *arXiv preprint arXiv:2204.03162*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE.

Tokmakov, P., Wang, Y.-X., and Hebert, M. (2019). Learning compositional representations for few-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6372–6381.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*.

Vinh, N. X., Erfani, S., Paisitkriangkrai, S., Bailey, J., Leckie, C., and Ramamohanarao, K. (2016). Training robust models using random projection. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 531–536. IEEE.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pages 5334–5344.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. (2019a). Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518.

Wang, H., He, Z., and Xing, E. P. (2019b). Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*.

Wieting, J. and Kiela, D. (2019). No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations*.

Xu, Z., Liu, D., Yang, J., Raffel, C., and Niethammer, M. (2021). Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*.

Xu, Z. and Niethammer, M. (2019). Deepatlas: Joint semi-supervised learning of image registration and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 420–429. Springer.

Xu, Z., Niethammer, M., and Raffel, C. (2022). Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. *Advances in Neural Information Processing Systems*.

Yang, X., Kwitt, R., Styner, M., and Niethammer, M. (2017). Quicksilver: Fast predictive image registration–a deep learning approach. *NeuroImage*, 158:378–396.

Yao, S., Yu, M., Zhang, Y., Narasimhan, K. R., Tenenbaum, J. B., and Gan, C. (2022). Linking emergent and natural languages via corpus transfer. In *International Conference on Learning Representations*.

Yezzi, A., Zollei, L., and Kapur, T. (2001). A variational framework for joint segmentation and registration. In *MMBIA*, pages 44–51.

Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., and Gong, B. (2019). Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2100–2110.

Yun, T., Bhalla, U., Pavlick, E., and Sun, C. (2022). Do vision-language pretrained models learn primitive concepts? *arXiv preprint arXiv:2203.17271*.

Zhang, T. and Zhu, Z. (2019). Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning*, pages 7502–7511.

Zhao, S., Ren, H., Yuan, A., Song, J., Goodman, N., and Ermon, S. (2018). Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31.