Exploiting and Transferring Generalizable Knowledge for 2D/3D Object Recognition

by **Jiachen Kang**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Wenjing Jia

School of Electrical and Data Engineering
Faculty of Engineering and IT
University of Technology Sydney

March 18, 2024

Certificate of Authorship / Originality

I, Jiachen Kang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: March 18, 2024

Abstract

In recent years, deep neural networks have significantly advanced the field of computer vision. However, these advancements have largely relied on the assumption of independent and identical training and test data distribution. In real-world scenarios, violations of this assumption due to covariate shift can trigger performance degradation, thereby highlighting the challenge of out-of-distribution (*o.o.d.*) generalization. In contrast, humans excel in *o.o.d.* generalizability based on their acquired generalizable knowledge. However, current deep learning models struggle with biased dataset confounders, hindering their acquisition of such knowledge. Therefore, in this research, experiments are conducted to explore the mechanisms and principles behind the acquisition and exploitation of generalizable knowledge, in order to address the challenge of *o.o.d.* generalization.

Our initial explorations focus on the learnability of generalizable knowledge using 2D transformation estimation tasks. Results demonstrate that utilizing a convolution neural networks that accept image pairs as inputs, along with causal synthetic datasets, enables machines to acquire knowledge about 2D transformations that can be generalized to unrelated semantic domains.

Based on this insight, this research introduces InterpretNet, a novel architecture to explicitly exploit generalizable knowledge of 2D transformations, which achieves enhanced test accuracy and explainability in hand-written digit classification. Expanding the scope, we integrate the learning methodology into a contrastive learning paradigm to implicitly exploit the generalizable knowledge. The results demonstrate enhanced model representation capability and classification accuracy in point cloud understanding tasks.

Finally, to further validate the potential of disentangling more confounding mechanisms in real-world tasks, we propose PCExpert, a self-supervised representation learning approach to transfer knowledge learned from a pre-trained image-text model to 3D point cloud understanding. Our results show that PCExpert outperforms state-of-the-art models across various tasks with enhanced representation capability, while substantially reducing trainable parameters.

In summary, this research investigates knowledge acquisition of target concepts based on causal theory, and introduces InterpretNet and regression loss to explicitly and implicitly exploit the acquired knowledge, respectively. This methodology is further validated through the PCExpert architecture in 3D understanding tasks. The findings in this research offers new insights and methodologies for future studies on $o.o.d$ generalization.

Dedication

To my beloved mom and dad.

Acknowledgements

To my mom and dad. This journey was fueled by your unconditional love and support. Words can scarcely express my gratitude. This thesis is dedicated to you.

To Prof. Sean He and Dr. Gray Peng from Intergenepharm. My incredible doctoral journey and adventure in academia would not even start without you. I am so grateful for the opportunity you provided.

To Associate Professor Wenjing Jia. Having you as my supervisor has been both a privilege and a stroke of fortune. You always offered support in every possible way, prioritizing the personal development and interests of your students. With so many challenges faced during my Ph.D. study, your patient guidance and insightful advice always helped me make a crucial step forward. It is the accumulation of these small but significant steps that have brought me to where I am today. I have gradually come to realize that the way you guide your students, and how you approach everything else, how you tackle every challenge, is the same – it's about taking a small step at a time, steadily moving forward. This is one of the most invaluable gifts from you, which has profoundly shaped my approach in both research and life.

To my candidature assessment panel – Associate Professor Qiang Wu, Associate Professor Min Xu and Professor Stuart Perry, for your insightful challenges and the constructive suggestions you provided for my work and thesis. Your rigorous and valuable feedback have greatly contributed to the refinement and depth of my research.

To my relatives in Sydney – Aunt Nan, Xinru and Beibei. Thank you for your invitations for meals and gatherings, and all the familial warmth in a land far from home. Aunt Nan, thank you for your help and those city walks we shared during my early days in Sydney.

To my friends in Australia. You all have painted my life here with vibrant colors of friendship, making everything, including my research journey, so meaningful. Thanks to Lucas, Wenbo and Kevin, for each moment shared with you, from the belly-busting hotpot gatherings to the joyful ping pang contests. A special thanks to Lucas for consistently calling me out, and for introducing me to new friends. My days in Sydney would have been dull and uninteresting without you. To Sam and Zack for offering your apartment at a ridiculously low rate. To Gerry, Ethan, Scott and all my board game buddies, for the endless conversations and the unforgettable board game nights.

To the people who created various online learning resources and information, which have constantly inspired me and broadened my perspective. Among these are: Andrew Ng, Yannic Kilcher, 3Blue1Brown, Mu Li, bryanyzhu, PaperWeekly, BigDataDigest, AI_era, and many others.

And thanks to everybody else who has supported me throughout this journey. A special mention to Ke Ding, Rokey Zhang, Ye Huang, Leizhong Zhang, Yanjing Liu, Xiang Su, Lu Zhang, all my aunts and uncles in Lanzhou, Zequn Jia, Peng He, Zhenhuan Li, Xiaochen Fan, Prof. Guan, Jinglin Lv, Ming Qin, Jia, Xiaoyang Lu, Xudong Song, Chengpei Xu and Si Wang, Axuan Sun, Jiangxin Xie, Xian Li, Yanhui Su, Yue Yang, Xiguang Yang, Byron, Xiao He, Mengfan Lv, Feng Ge, Yining Hou, Bo He, Liangyan Wang, Yan Zhang, Xiao Tan, Xiao Li, Qijian Deng, Jianhua Li, Yi Li and Jiarong Zhang, Yang Wang and Yihui Lv, Lynn Wu and Jojo, Hongen Wang, Zhanzhong Gu, Weiwei Du, Zijie Zhang, and Chenguang Hu.

Jiachen Kang
March 18, 2024
Sydney, Australia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Deep learning algorithms based on neural networks (NN) have made rapid progress in the past two decades. These algorithms have impressive performance in challenging tasks such as image recognition, image generation, machine translation, robotics, *etc.* However, one of the main problems current deep learning community encounters with is out-of-distribution (*o.o.d.*) generalization. Most of these algorithms are based on statistical learning principles, and thus strongly rely on the assumption of *i.i.d.*, *i.e.*, the training data and test data are independently and identically distributed. In practice, however, this assumption could be violated when the distribution of the training dataset is not representative to that in the real scenario, which leads to performance drop on the actual production data (*i.e.*, test data) comparing with the training data. This situation is also described as generalization performance drop under covariate shift [1], which has become one of the main challenges that the deep learning community encounters nowadays.

To tackle the problem of *o.o.d.* generalization performance drop, extensive studies have been carried out recently. One obvious category of solutions is data-centric. Under the assumption of *i.i.d.*, the more precisely the distribution of the training dataset can approximate to that in real production environment, the better performance we can obtain. This is why the current deep learning algorithms typically require extensive training datasets. But the question is: what constitutes an adequately extensive dataset? ImageNet [2], a standard dataset extensively used for training classification and detection algorithms, contains more than 14 million images, with around one thousand images in each synonym set. However, even models trained on this vast dataset have suffered from $40 - 45\%$ performance drop when evaluated on ObjectNet [3]. ObjectNet is a bias-controlled dataset that generates thousands of images through 600 combinations of parameters, only intervening on *three* factors of variation during the photo generation process, including viewpoint, object rotation and background. This implies that should we attempt to construct a sufficiently large dataset to approximate the distribution of real-world data, by considering all possible combinations of parameters of factors of variation, the number of necessary data points would approach infinite. Similar generalization problems in various sensory domains have been reported in deep learning literature, such as 3D object modeling [4], [5], natural language processing [6], [7], time series signal processing [8]–[10], *etc.*

Furthermore, in the paradigm of supervised learning, each training data point is required to be labeled. Take ImageNet [2] again as an example, each image is manually annotated with one or some of more than twenty thousand labels. This is merely the tip of the iceberg, though. Compared with ImageNet where every image is only attached with some word labels and a few of bounding box coordinates, datasets suitable for training semantic segmentation require pixel-level metadata which need orders of magnitude longer time to annotate. It is reported in [11], [12] that for images with high-quality semantic annotations in CamVid [11] and Cityscapes [12] datasets, it took 60 and 90 minutes on average to label a single image, respectively. The time-consuming and therefore prohibitively expensive human labor work usually result in insufficient dataset size [13]. Besides, sometimes annotation is not able to provide complete information of content in images due to occlusion, lighting conditions, *etc.* [14]. In extreme scenarios, correct annotation on real world images is impossible [15].

In summary, despite the remarkable advancements in deep learning, generalization performance drop remains a challenging problem. Although enlarging the training dataset may alleviate this problem, particularly with the advancements of multimodal generative algorithms empowered by large language models (LLMs), the feasibility and cost associated with unlimited expanding of datasets make this approach an arduous and potentially unsustainable path. Perhaps this is an inevitable path? Or not? Before further discussion, let us reflect on ourselves, and examine how *humans* perform in terms of generalizability.

1.1.1 Human Generalizability

To begin with, let us simply take image recognition task as an example. When we have seen elephants, whether in documentaries or in real life, the concept of elephants has left an impression in our minds. Later, when we saw a photo of an elephant, we could naturally associate this photo with the concept of an elephant. This is also what most current state-of-the-art neural networks can do with a high accuracy rate. But what if the image is a top view photo taken with a drone, or a close-up one of an elephant nose? And how about an elephant in a refrigerator, or a purple elephant drawn by a kindergarten kid? As human beings, there is a high probability that we can still recognize the content in the image. Note that, the data required for human learning a new concept is also inevitably limited. In other words, we can generalize the concept of an elephant to novel scenarios with efficiently learned knowledge. It's quite questionable, though, whether neural networks can do the same, given the reports of failures caused by various unintentional or intentional interference [16], [17].

The ability of infants to efficiently acquire knowledge and flexibly reuse them in novel scenarios has been extensively studied [18]–[21]. Some researchers refer to this generalizability as systematicity [22] in language learning. A more easily grasped description appears in Gary Marcus's discussion in [23]. He describes human cognition activities as the "Algebraic Mind", indicating human mind as a "computer-like manipulator" of symbolic variables. One of the properties of algebraic operations is symmetry, resembling human cognition ability. That is, if human learn the knowledge of a concept or mechanism, they can generalize this knowledge to other related or even unrelated domains. For instance, in the aforementioned example of elephant images, it's quite unlikely in reality for an elephant to appear in a refrigerator, yet this does not hinder human from recognizing. It is even a bold assertion to say that humans are capable of recognizing an elephant regardless of its location. The concept of an elephant is disentangled from other factors of variation and exhibits symmetry.

In mathematics, symmetry can be described as an intrinsic property of a mathematical object which causes it to remain invariant under certain classes of transformations [24]. Consequently, symmetry in deep learning algorithms can be concretely defined as follows.

Definition 1.1.1 (Symmetry)**.** Consider a machine learning algorithm designed to approximate a function $f : \mathbf{X} \rightarrow \mathbf{Y}$, where $\mathbf{X}$ and $\mathbf{Y}$ represent the input and output spaces, respectively. The function f is said to exhibit symmetry with respect to a set of transformations $\mathcal{T}$, if for every transformation $t \in \mathcal{T}$ applied to $\mathbf{X}$, the function's output remains invariant. Mathematically, this symmetry property is defined as:

$$f(\,\mathbf{X}\,) = f(\,t(\mathbf{X})\,), \quad \forall\, t \in \mathcal{T}$$

In this context, symmetry refers to the invariance of the learned information in the dataset $\{\mathbf{X}, \mathbf{Y}\}$ under the specified transformations $\mathcal{T}$.

Based on this definition, it can also be expressed that the learned information (parameterized in f) is *disentangled* from the transformations $\mathcal{T}$. In this thesis, the learned information that is symmetric or disentangled is also referred to as "*generalizable knowledge*". Based on the findings and conclusions drawn from the aforementioned research, we assume that it is precisely the acquisition of generalizable knowledge that accounts for humans' remarkable generalizability. This leads to another plausible hypothesis: should machines also acquire generalizable knowledge, their generalizability could also be enhanced. We now discuss why acquisition of generalizable knowledge can be challenging for machines.

1.1.2 A Causal Perspective

Let us return to the example of elephant images. From the causal perspective, we can think of image generation process as a result of the interaction of various mechanisms. In elephant images, the mechanisms include those intrinsically related to the concept of elephants, such as the shape, color and size of various body parts, characteristics of the elephant's skeleton (which determines the positional relationship of these body parts), the characteristics of materials of the skin (which determines the way the light reflects), *etc.* In addition, there are some indirectly relevant mechanisms, such as the environment, the intensity of sunlight when taking the photo, the position and angle of the camera, *etc.* These mechanisms are the cause by which a photo presents its content.

If a causal graph is created according to the generation process of a photo (Figure 1.1), it can be found that:

- Classification tasks infer along the anti-causal direction, and the label (cause) variable is not the direct cause of the image (effect) variable.
- Environment variable U_1 acts as the confounder.
- Image capturing equipment (*e.g.* a camera) variable U_2 is another source of distribution shift in popular computer vision datasets.

Therefore, in order to make robust prediction on $\mathbb{E}(Y|X)$ in cases of distribution shift, we should also condition on variable U_1 at least, which is impossible though, because image X is the only observable variable that can be conditioned on. This may explain how performance drop happens with distribution shift, and why improving *o.o.d.* generalization is a very challenging task.

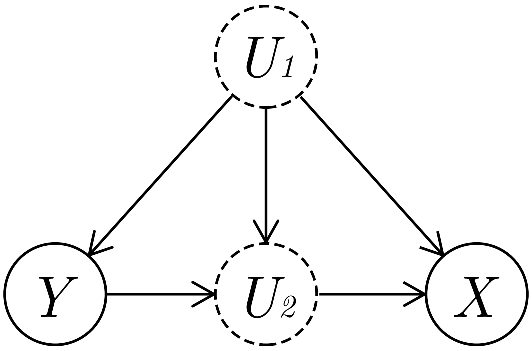


Figure 1.1: A simplified causal graph of the image generation process. X: image variable, Y: ground truth (label of the image), U_1: variables of objects in the environments, such as foreground subject, background objects, the sunlight, *etc.*, U_2: variables of the image capturing equipment, such as position and view angle of the camera, exposure parameters, *etc.* Variables contained in U_1 and U_2 are all unobservable, *i.e.*, they are not available in the dataset.

While children gain generalizable knowledge and understand physical mechanisms through extensive observations and experimentation over time [25]–[27], building foundations for object perception and future knowledge acquisition [20], [28], [29], such opportunities are rarely available to existing deep learning models, which can only learn from samples drawn from the joint distribution of these mechanisms. Current datasets for visual learning inevitably introduce confounding mechanisms, which make it difficult for models to learn unbiased representations and to acquire disentangled knowledge [30]–[32].

Thus, this leads us to the key questions in this thesis:

- Is there a methodology that allows deep learning models to acquire generalizable knowledge? If yes,
- How to exploit the knowledge in computer vision tasks?

These questions serve as the core motivation in this thesis.

1.2 Overview

In order to address the research questions mentioned above, the following objectives and contributions are accomplished in my PhD studies.

- Through specifically designed synthetic datasets and learning tasks, generalizable knowledge of 2D transformation can be acquired utilizing deep neural networks (DNN).
- We propose InterpretNet, a novel architecture that can explicitly exploit the generalizable knowledge acquired by DNNs in handwritten digit classification tasks. InterpretNet exhibits remarkable *o.o.d.* performance under covariate shift.
- We integrate the methodology of generalizable knowledge learning into self-supervised learning, thereby enhancing the generalizability of DNN models in encoding 3D point clouds.
- We devise a novel transfer learning architecture "PCExpert" to efficiently exploit and transfer generalizable knowledge from pre-trained image-text models

for 3D point cloud understanding.

The above findings are discussed in depth in Chapters 3 to 5 in this thesis. These findings also resulted in the following publications:

- J. Kang, W. Jia, and X. He, "Toward Extracting and Exploiting Generalizable Knowledge of Deep 2D Transformations in Computer Vision," *Neurocomputing*, 2023.
  DOI: https://doi.org/10.1016/j.neucom.2023.126882

- J. Kang, W. Jia, X. He, and K. M. Lam, "Point Clouds Are Specialized Images: A Knowledge Transfer Approach for 3D Understanding," *IEEE Transactions on Multimedia*, 2023. Under review.

- J. Kang, W. Jia, and X. He, "Disentangled Knowledge is Generalizable: A Cognitive Perspective," *Under submission.*

The rest of the thesis is organized as follows:

- In Chapter 2, we conduct a comprehensive literature review to lay a solid foundation for the investigations and discussions that follow in subsequent chapters. We first delve into the definition of generalization and the significance of this concept. We also explore existing works and challenges in improving model generalization, which is crucial to understand the context of the research in this thesis. Furthermore, we provide a comparative analysis of various methodologies on transfer learning, identifying gaps that our research aims to fill. By integrating the insights from previous studies, Chapter 2 establishes the stage for our original contributions presented in the later chapters.

- We begin our work with Chapter 3 by exploring the learnability of generalizable knowledge, through a series of experiments focusing on 2D transformations such as rotation, scaling, and translation. The aim of the exploration is to establish the methodology enabling DNNs to estimate the parameters of transformations applied to an image, regardless of the image's semantic domain, like human beings. Our results demonstrates that utilizing a convolution neural network that accept image pairs as inputs alongside causal synthetic datasets enables the machine to acquire knowledge about 2D transformations that is generalizable to unrelated semantic domains. This finding has provided us with the toolkit to disentangle specific knowledge, and laid the foundation for further exploiting this knowledge.

- In Chapter 4, we investigate potential ways to exploit generalizable knowledge in downstream tasks. Inspired by the "hypothesis-verification" interpretation process of humans, we propose InterpretNet architecture for handwritten digit classification. InterpretNet consisting of two distinct modules: an ESTIMATOR and an IDENTIFIER, each of which equips generalizable knowledge acquired with the methodology introduced in Chapter 3. The results show that InterpretNet not only classifies images under covariate shift with significantly higher test accuracy, but also provides underlying explanations, which closely resembles human perception.

  Furthermore, we also explore a more effective and efficient way to exploit generalizable knowledge in real-world tasks, *i.e.*, through the integration of a regression loss into the framework of self-supervised learning. This approach allows us to incorporate generalizable knowledge into a DNN model in a more implicit manner, and facilitate the learning of more descriptive representations. We also discuss the relationship between image-text contrastive learning and generalizable knowledge learning, which forms our hypothesis that models pre-trained in this manner exhibit enhanced generalizability. To validate this hypothesis, we conduct experiments on complex and realistic datasets such as 3D point clouds. Our results indicate a significant improvement in the model's representation capability and accuracy in point cloud classification tasks.

- In Chapter 5, we take a further step in examining the benefits of generalizable knowledge that is acquired through large-scale text-image contrastive learning. We introduce PCExpert, a novel architecture that exploits and transfers the knowledge of 2D images to 3D point cloud understanding. Our results reveal that transferring generalizable knowledge from a pre-trained text-image model significantly enhances the model's representation capability for point clouds, which further strengthens our understanding of the learning and exploitation of generalizable knowledge.

- In Chapter 6, we conclude this thesis by integrating the key findings of our work and implications our contributions have for the field of study. Moreover, we briefly discuss some of the challenges ahead in future research, and how the insights gained through our research can be potentially useful addressing these challenges.

Chapter 2

Literature Review

This chapter conducts an in-depth review of literature, aiming to understand two core topics that are critically relevant to this thesis. The first topic investigates the foundations and development of generalization, a core concept that drives the field towards more robust and versatile models. The second topic examines the various methodologies developed for transfer learning, a learning paradigm that utilizes knowledge acquired by a model on one problem and applies it to a different problem.

In Section 2.1, a comprehensive analysis of the first topic is conducted, focusing on the concept of generalization and its related aspects. We begin with Section 2.1.1, by exploring the definition and paradigms that have contributed to shaping our understanding of this concept. This is followed by Section 2.1.2, which provides a review of various factors that could affect generalizability of deep learning models. Section 2.1.3 concludes this part by discussing the potential gaps and challenges for future exploration.

To address the second topic, Section 2.2 presents an in-depth review of current works on transfer learning. We first review the technique of domain adaptation in Section 2.2.1, which is one of the most popular methodologies that facilitate the transfer of learned knowledge across different but related domains. Then, Section 2.2.2 discusses the technique of fine-tuning as a strategy for adapting pre-trained models to new tasks or domains.

The body of literature reviewed in this chapter predominantly consists of articles in the studies of deep learning that were published within the last five years. The majority of these publications appear in academic conferences and journals that hold a high level of scholarly reputation. A smaller portion comprises articles that, while published on Arxiv and not peer-reviewed, have had considerable citations from the community.

The scope of this review is deliberately focused on emerging trends and methodologies in the field of deep learning, particularly with regard to generalizable knowledge and its application in 2D and 3D understanding tasks. While this focus allows for a detailed discussion, it also inherently poses limitations. For instance, this scope may neglect machine learning studies in general and interdisciplinary approaches which might contribute to the broader understanding of these topics. Moreover, the exclusion of older, foundational works could potentially overlook their enduring relevance.

2.1 Generalizable Knowledge

2.1.1 Definition and Significance of Generalization

Generalization in the context of deep learning refers to the model's ability to apply learned knowledge to unseen data effectively. The performance gap between training data and test data is a measure of generalization. A narrower gap generally indicates better generalization. This characteristic is central to the utility of deep learning models, which represents the quality of predictions or decisions in novel situations. Goodfellow *et al.* [33] defined generalization as "the ability to perform well on previously unobserved inputs". A contrastive concept is overfitting, which "occurs when the gap between the training error and test error is too large" [33]. Generalization addresses the challenge of overfitting by emphasizing the understanding of patterns that are universally applicable rather than those that are dataset-specific [33]. In this thesis, the usage and definition of generalization (or generalizable knowledge) are aligned with the above definition.

Because generalization in machine learning determines the ability of a model to perform on new, unseen data, which can be effectively quantified in practical tasks, the concept becomes the cornerstone of a model's utility in practical applications. Without this, a model would fail to deliver reliable results in real-world scenarios, regardless of its accuracy on training data. This concept is particularly significant in the context of *deep* learning due to the complexity and high dimensionality of the deep neural networks involved. Deep neural networks, known for their capacity to learn intricate patterns in large datasets, can easily overfit to training data, capturing noise and specificity that do not generalize to broader contexts. Therefore, the pursuit of generalization continuously pushes the boundaries of model architecture design and training methodologies, and makes deep neural networks a transformative force across a wide range of academic and industrial fields.

2.1.2 Factors Affecting Generalization

In earlier deep learning research, researchers have explored how generalization correlates with various factors, including the size of datasets [34], regularization techniques [35], [36], training dynamics [37], *etc.* This section provides an in-depth review focusing on current studies that explore the factors affecting model generalization, including dataset diversity, regularization, and model complexity.

Dataset Diversity

The diversity and size of the dataset play a significant role in model generalization. Diverse datasets covering a wide range of data variations provide more information for models to learn from and help them to generalize better.

One technique to diversify a dataset is data augmentation. Data augmentation is a common practice to improve model generalization by enlarging the training dataset using various transformations on the original data. Techniques like rotation, scaling, and cropping are employed to create a more diversified dataset that helps in making the model more robust to unseen data [38]–[42].

Besides traditional image transformation techniques such as rotation, flipping, color intensity variations, *etc.*, some studies aim at generating abundant variations beyond the support of training data by randomizing certain parameters during image generation process. These studies can be categorized into domain randomization. In object detection tasks [43]–[45], various objects with random pose, scale, color, texture and position were placed in 3D scenes. Illustration parameters, camera angle, *etc.*, were adjusted randomly within predefined ranges. Prakash *et al.* [46] developed this technique by introducing structured randomization. With a hierarchical conceptual model, the probability distribution of each parameter is conditioned on its parent parameter, so that more realistic images could be generated.

Another direction to augment data is through adversarial gradients. In study [47], being guided by the loss gradients from a label and a domain classifier, the distribution of training data can be expanded across different domains without decreasing label classification accuracy. Volpi *et al.* [48] proposed to generate "worst-case" samples iteratively along the adversarial gradient, so that the model is allowed to be trained to generalize to "hard" samples.

Generative models such as auto-encoder and GANs are also utilized to generate new data by stylizing source domain images with styles from other source domains [49], ImageNet [50], a specific style set [51], or instances from other domains [52].

However, in many scenarios it is very difficult and costly to obtain a real-world dataset that is of abundant amount of data and with accurate labels. To address the issue of data (or rather annotation) insufficiency, an effective way is to use synthetic data as substitute for or complement to real data. Through the development of computer graphics and electronic games, one can now easily create a 3D virtual scene from scratch and render realistic images. Thanks to the structural information architecture in graphic engines, theoretically unlimited synthetic images with pixel-accurate annotations can be produced efficiently in the terms of time and money. This section focuses on reviewing research related to synthetic datasets and their role in enhancing model performance.

LINEMOD [53] is one of the earliest developed datasets that provide RGBD video sequences and 3D models of 15 3D objects in cluttered indoor scenes with ground truth class labels and 3D pose. It has become a popular benchmark for object classification and 3D pose estimation tasks.

The dataset vKITTI [54], initially consisting of 35 synthetic videos with about

$17,000$ frames was built with Unity upon real world traffic video dataset KITTI [55]. To resemble the real world traffic scenarios as closely as possible, researchers constructed 3D virtual worlds according to the real traffic information recorded and annotated in KITTI, which includes positions, sizes, rotation angles of surrounding cars and the movement information of the recording car itself. Five different scenes in KITTI were used to cover most of driving situations. Annotations in each frame is accurate and adequate for downstream visual learning tasks, which include detailed properties of surrounding cars and the recording camera, depth map, instance-level segmentation map, dense optical flow between frames, and multi-instance bounding boxes.

Because of its rich API, Grand Theft Auto V (GTA5) by Rockerstar Games [56] is widely used in computer vision community. The game is basically a driving game in a city scenario with realistic rendering, so it is adopted in many deep visual learning tasks such as detection [57], crowd counting [15], segmentation [58], [59], *etc.* The game was initially introduced to computer vision community by Richter *et al.* [60]. The authors created $24,996$ pixel-level accurately annotated images for semantic segmentation tasks covering 19 classes commonly seen on urban streets. Benefit from the APIs released to the public, resources of elements appeared in a scene including mesh, texture and shader can be tracked. Based on predetermined association rules, related resources can be grouped and annotated to be one of the 19 classes, so that the semantic information in that scene can be extracted. Most importantly, these rules can be automatically constructed and propagated in other scenes, which dramatically accelerate the annotation efficiency. The authors reported that the annotation process for 25 thousand images completed in only 49 hours. In comparison, annotation of similar quality in Cityscapes [12] would require 90 minutes per image. Sample images rendered with GTA V and corresponding annotations are shown in Figure 2.1.

VisDA is one of the widely used datasets constructed by Peng *et al.* [61] dedicated for synthetic-to-real domain adaptation studies. VisDA can be exploited as the off-the-shelf datasets for training deep visual domain adaptation algorithms, as the validation and test sets are chosen from widely used public datasets, *e.g.* Microsoft COCO [62] and YouTube Bounding Boxes [63] according to the object categories

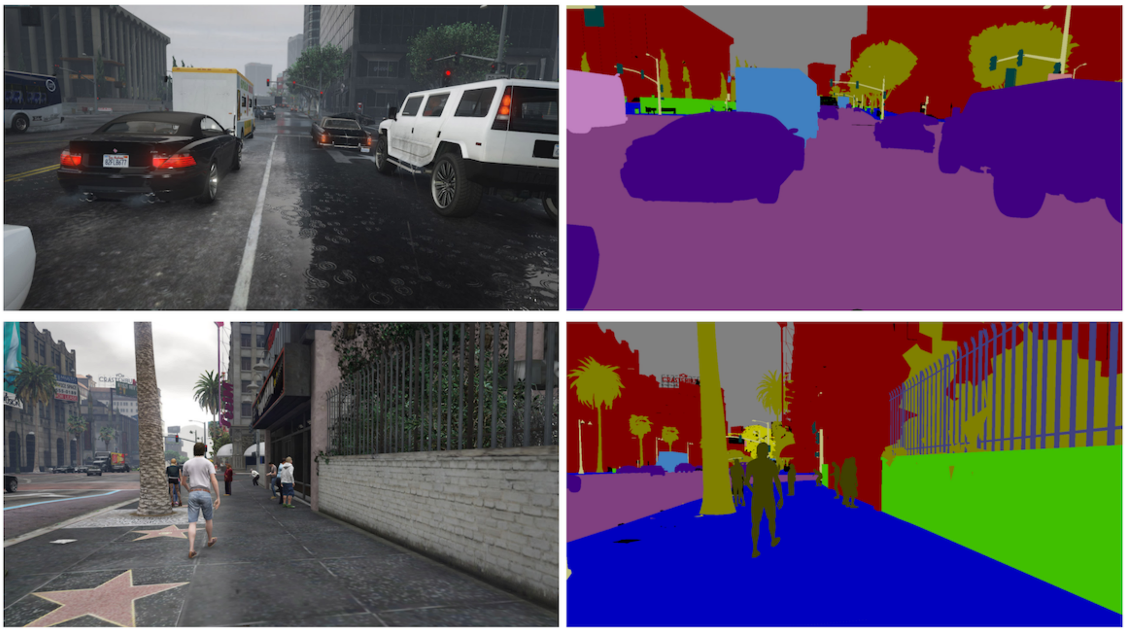


Figure 2.1: Sample in-game images (left column) rendered with GTA V [60] and their semantic map annotations (right column).

in the source domain without further manual filtering. The datasets were updated in [64] with the addition of two datasets, *i.e.*, Syn2Real-D for detection task and Syn2Real-O for open-set classification task. Syn2Real-D and Syn2Real-O adopt 3D models of 33 more classes from ShapeNet [65] and are rendered with objects of 20 categories semi-randomly scattered in the scene.

Vehicle X is a synthetic dataset created by Yao *et al.* [66] dedicated for vehicle re-identification tasks. To render images of cars, 272 3D vehicle models and $1,362$ vehicle identities were created. Unity [67] was adopted for rendering and 5 parameters were adjusted to control the appearance of rendered images, including pose of 3D model, direction and intensity of light, and height and distance of camera.

In addition to the advantages of efficient synthesis and accurate annotation, another benefit of utilizing synthetic virtual world is the possibility of controlled analysis.

Being powered by modern game engines such as Unity [67] and Unreal Engine [68], it is possible to control nearly every single mechanism in the virtual world, while keeping all the others unchanged. This feature is especially useful in scenarios such as *ceteris paribus* analysis or “what-if analysis” where rare cases or “hard samples” can be intentionally created and investigated, which, however, is nearly impossible for real world datasets.

From a causal perspective, being able to control every single mechanism means that intervention and treatment randomization becomes possible, which will make each mechanism detach from its causal parent, and thus get rid of confounding association. In this way, a virtual world provides an excellent laboratory for us to experiment on deep learning algorithms with regard to the generalization ability in a causal theoretical framework.

Regularization

Regularization is crucial in preventing overfitting and thereby improving model generalization. Techniques such as L1 and L2 regularization, dropout, and early stopping have been employed to achieve better generalization by introducing some form of constraint or penalty on the complexity of the model.

Huang *et al.* [69] proposed a novel regularization technique, called Feature Variance Regularization (FVR), focusing on penalizing the empirical variance of features during the training process. The concept behind FVR is to induce a form of confusion in feature extraction, preventing models from learning features overly specific to training samples. The authors theoretically and empirically justified FVR, demonstrating its effectiveness across multiple visual tasks such as classification and semantic segmentation.

In study [70], the authors proposed Guillotine Regularization (GR), a technique that involves training a deep neural network with self-supervised learning and subsequently removing its last few projector layers for downstream tasks. This proposed approach is based on the hypothesis that the performances across layers are different which can be affected by optimization methods and data distribution during training. The research highlights that when the positive views are more aligned with the downstream task, the optimal layer to use is closer to the last layer. This method has shown to significantly boost performance in applications like ImageNet classification, where more than 30 percentage improvement in accuracy can be gained.

Gao *et al.* [71] proposed "coupled tensor norm regularization" to reduce overfitting by ensuring that both the model's output feature matrix and the input data lie in a low-dimensional manifold. In the context of DNNs, the coupled tensor norm regularization presents as non-convex and non-differentiable. To address this, the study introduces an auxiliary variable leading to a quadratic penalty formulation. An alternating minimization method is then employed to manage the non-separability of the optimization problem. The results indicate that the technique outperforms traditional methods such as the L1, L2 and Tikhonov regularizations, particularly in scenarios with limited data availability.

Bacanin *et al.* [72] presented a novel approach to optimize the dropout regularization in convolutional neural networks. This method employs an automated framework that is based on a hybridized version of the Sine Cosine Algorithm and Firefly Algorithm swarm meta-heuristics to determine the optimal dropout rate. The experimental results on four benchmark datasets indicate that the proposed method outperforms other state-of-the-art methods in terms of classification accuracy.

The technique of batch normalization, which helps in stabilizing the training of deep neural networks, has also shown to improve model generalization significantly [73]–[76]. Ioffe *et al.* [75] argued that batch normalization regularizes the model, reducing the need for dropout. Specifically, in experiments on the MNIST dataset, the authors found that batch normalization made the distribution of inputs more stable and reduced internal covariate shift. This stability translated into faster training and higher accuracy of the network. Furthermore, the authors applied batch normalization to an ImageNet classification network and demonstrated that it can match the network's performance using only 7% of the training steps and exceed its accuracy by a substantial margin.

Santurkar *et al.* [74] challenged the common belief that the effectiveness of batch normalization stems from the reduction of internal covariate shift during training. Instead, they emphasized its role in creating models that depend less on single directions in activation space, which was first discussed by Morcos *et al.* [77]. Additionally, the study highlights the impact of batch normalization in decoupling the length and direction of weights in a network, which has been shown to enhance training efficiency through faster convergence.

Cakaj *et al.* [76] introduced a novel method called spectral batch normalization

(SBN), which aims to improve generalization by normalizing feature maps in the frequency domain. They argued that this approach prevents large values in the feature maps throughout the training process, thereby reducing overfitting. The experimental results show a notable increase in network accuracy when SBN is employed alongside the traditional batch normalization.

Model Complexity and Capacity

The capacity and complexity of a deep learning model are crucial factors affecting its generalization. It was believed that models with high complexity might fit the training data very well but fail to generalize on unseen data. Various studies have delved into understanding the balance between model complexity and generalization, some of which present findings that contradict traditional perspectives.

Kawaguchi *et al.* [78] explored the intricacies of model generalization, addressing a central question in the field: How do deep learning models generalize well despite their large capacity and complexity. The paper challenges the traditional belief that models with large hypothesis-space complexity inherently lead to poor generalization. It illustrates that even models with overwhelming capacity can achieve small test errors and expected risks, thus maintaining good generalization. Specifically, the paper suggests that conventional wisdom about the norm of parameters and over-parameterization may not fully explain the generalization of some models, such as over-parameterized linear models. The paper presents a theorem indicating that over-parameterized linear models can memorize any training data and reduce training and test errors to near zero, regardless of the norm of parameters and their distance from ground-truth parameters.

In their survey, Hu *et al.* [79] examined the current research on the measurement of model complexity in deep learning. They argued that the general proposition from [80], which suggests that "a learned model with lower complexity generalizes better", may still be valid if model complexity can be analyzed in a more detailed and systematic manner. The authors categorized model complexity into two facets: expressive capacity and effective model complexity. The paper identifies four critical factors that influence model complexity, including model framework, model size, optimization process and data complexity. Expressive capacity has been explored from four aspects, which are depth efficiency, width efficiency, expressible functional space and VC dimension and Rademacher complexity. These tools provides us a nuanced perspective on how to systematically analyze deep learning model complexity and its influence on the generalization and performance.

In their book titled “Geometry of Deep Learning: A Signal Processing Perspective”, Ye [81] reviewed and extended the above discussions. A central finding presented is the “double descent” curve, which revises the traditional bias-variance trade-off. The curve demonstrates that as model capacity increases beyond a certain point, known as the interpolation threshold, test performance can actually improve. This phenomenon also contradicts the expected outcome of overfitting in over-parameterized models. The author also examined the role of optimization algorithms in influencing generalization. It is found that stochastic gradient descent (SGD) introduces a bias towards simpler models, which results in better generalization even in over-parameterized models.

2.1.3 Challenges in Achieving Generalization

As the core problem in deep learning, the fundamental challenge in model generalization originates from our incomplete understanding of how deep learning models work. Zhang *et al.* [82] highlighted in their study that the conventionally believed factors contributing to improving generalization are continuously being challenged. The authors conducted experiments where labels of the training data were replaced randomly. Surprisingly, the convolutional neural networks for image classification trained with stochastic gradient methods easily fit this randomized data. The paper also theoretically demonstrates that even simple two-layer ReLU networks can express any labeling of the training data, given sufficient parameters. These findings challenge the traditional belief that over-parameterization leads to overfitting, and suggest that the effective capacity of neural networks might be larger than previously thought. Furthermore, this study shows that most regularization techniques in deep learning are not necessary for generalization. These findings [82] that contradict empirical and intuitive expectations unmistakably indicate a substantial gap in our existing understanding about generalization.

The study by Zhang *et al.* [82] poses a conceptual challenge to traditional statistical learning theory. As discussed by Kawaguchi *et al.* in [78], theoretical frameworks such as Vapnik–Chervonenkis Bounds, Rademacher Complexity, *etc.*, while useful, often fail to provide tight generalization bounds for deep neural networks that align with empirical results. Hence, the gap between theoretical predictions and practical outcomes in deep learning generalization is still an open challenge requiring further research [78], [81].

It is also pointed out in [82] that the existing measures of model complexity struggle to explain the generalization ability of large neural networks. Hu *et al.* [79] also highlighted several promising research directions for effective measures of model complexity and expressive capacity. These include cross-model complexity comparisons, exploration of model size bottleneck, among others. These identified directions represent challenging aspects of generalization that await further exploration and resolution in the field.

2.2 Transfer Learning

In the ever-evolving landscape of machine learning, transfer learning has emerged as an important technique, especially in the realm of deep learning. This approach, fundamentally different from traditional machine learning methodologies, involving the core principle of leveraging pre-learned patterns or knowledge from one domain to enhance learning in another, different, but related domain. Its growing significance is not merely a byproduct of theoretical interest but is deeply rooted in its practical efficacy across various applications, from computer vision to natural language processing.

In traditional deep learning approaches, models are designed and trained for specific tasks, requiring a substantial amount of data that is often expensive and time-consuming to collect. However, transfer learning circumvents this limitation by adapting pre-trained models, which have been exposed to vast and diverse datasets, to new tasks with relatively limited data. This method not only accelerates the training process but also often leads to improvements in model performance, particularly in scenarios where data availability is a constraint.

The conceptual foundation of transfer learning shares a connection with human learning processes [83]. Just as humans apply knowledge gained from previous experiences to new situations, transfer learning enables artificial intelligence systems to replicate this aspect of learning efficiency. The practicality of this concept is evident in the utilization of pre-trained neural networks. For example, models trained on large image datasets, such as ImageNet, have been successfully adapted to medical imaging tasks, demonstrating the versatility and power of transfer learning.

This section will delve into the seminal works and recent advancements in transfer learning, exploring their theoretical foundations and practical implementations. Two major areas in transfer learning will be focused on. In Section 2.2.1, we explore the techniques and theories behind domain adaptation. Then, typical works of different approaches in fine-tuning are reviewed in Section 2.2.2.

2.2.1 Domain Adaptation

In this study, we mainly focus on the *o.o.d.* generalization problem that is caused by domain shift. In the context of causality, domain shift can be considered as consequence of distribution shift of parameter value(s) of one or more mechanisms, when comparing source domain(s) (*i.e.*, training data) with the target domain (*i.e.* test data).

Formally, it is assumed that there exist K ($K \geqslant 1$) source domains, $\mathcal{D}_{\mathcal{S}} = \{D_S^{(k)}\}_{k=1}^{K}$, and each domain $D_S^{(k)} = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{N_S^{(k)}}$, where $X^{(k)}$, $Y^{(k)}$ denote input dataset and corresponding label drawn from the domain probability distribution $P_S^{(k)}(x, y)$, and $N_S^{(k)}$ denotes the size of the dataset. If $K = 1$ or domain label is not of concern in the context, data points and distribution of source domain can also be denoted as $X^S, Y^S \sim P_S(x, y)$. Similarly, the target domain is $D_T = \{(X_i^T, Y_i^T)\}_{i=1}^{N_T}$, where $X^T, Y^T \sim P_T(x, y)$. In the setting of domain shift, $P_S^{(k)}(x, y) \neq P_S^{(k')}(x, y)$ for $k \neq k'$ and $k, k' \in \{1, ..., K\}$. Moreover, $P^T(x, y) \neq P_S^{(k)}(x, y)$, $\forall k \in \{1, ..., K\}$.

If Y^T is unavailable and only X^S, Y^S and X^T is accessible during training, it is usually considered as a domain adaptation task. If any data from D_T is unavailable during training, then it is generally classified as a domain generalization problem.

In both settings, the objective is to find a function $f : x \Rightarrow y$, so that after reducing the source risk $\epsilon_S(f) = \mathbb{E}_{(x^S,y^S)\sim P_S}[f(x^S) \neq y^S]$ by training the model on the source data, the target risk $\epsilon_T(f) = \mathbb{E}_{(x^T,y^T)\sim P_T}[f(x^T) \neq y^T]$ in the test stage can also be reduced and has $\epsilon_S(f)$ as the lower-bound.

In this section, mainstream methodologies and crucial techniques in domain adaptation and domain generalization are reviewed. In general, there exists two main directions to adapt models from source domain to target domain with domain adaptation techniques, *i.e.*, domain mapping and distribution alignment.

Domain Mapping

One of the directions involves manipulation of images in the source domain in the term of style or texture, as if they were sampled from the distribution of the target domain. More specifically, after domain mapping, source domain images will be indiscriminate to domain discriminator *w.r.t.* the statistical information of image style, while the semantic information remains unchanged.

One successful application of domain mapping technique is conducted by Shrivastava *et al.* [84]. To tackle gaze estimation problems, the researchers proposed a refinement network using SimGAN in order to adapt source dataset of UnityEyes [85] to target dataset of MPIIGaze [86]. A gaze estimation network is then trained on refined images and achieved state-of-the-art results.

In their research, several techniques and tricks are notable to improve domain adaptation performance. To obtain better refined images that look more realistic and at the same time preserve the semantic information of gazing direction, local adversarial loss and self-regularization loss are introduced. To make the adversarial training more stable and reduce artifacts in generated images, a pool of refined images produced at previous stages are used so that the discriminator can be trained on samples not only in current mini-batch (generated by the current version of GAN), but also in the whole training process.

After this work, there have been increasing approaches on domain mapping that continuously improve the adaptation performance and apply the technique to various applications. Bousmalis *et al.* [87] trained the generator by conditioning on not only source images but also a noise vector, so theoretically the size of training dataset mapped to the target domain could be unlimited. Zheng *et al.* [14] proposed T^2Net for single-image depth estimation. Stein *et al.* [88] utilized CycleGAN [89] for their synthetic data mapping module and report performance improvement in tasks of semantic segmentation and obstacle avoidance. In the setting of person re-identification (re-id) tasks, where each person could be captured by multiple cameras along his/her route of travel, it is assumed that the domain gap is caused by camera condition *e.g.* locations/backgrounds, lighting conditions, quality, *etc.* This means there exist multiple domains in person re-id datasets. Bak *et al.* [90] tackled this problem with a multi-domain mapping approach, where one of synthetic domains is selected and translated to target (test) domain for fine-tuning the re-identification network. Domain mapping is conducted with CycleGAN, with addition of a regulatory loss similar to [84] and a foreground mask which ensure the semantic identity information could be preserved during style translation.

Distribution Alignment

The other direction of domain adaptation is devoted to, instead of mapping one domain to the other, aligning representational distributions of these two domains. In other words, the feature representation for each data should be domain-invariant.

One way to do so is to directly measure and penalize the divergence between the two representation distributions. Rozantsev *et al.* [91] used Maximum Mean Discrepancy (MMD) [92], [93] for divergence criterion. As a contrast, Zhang *et al.* [58] suggested that we should "avoid the assumption" that there exists the function shared by source and target domain mapping to the same representation distribution in the context of semantic segmentation. Instead, a constraint based on label distribution is exploited, which is based on a simpler assumption that, in cityscape images, the proportion of pixels belonging to the same class *e.g.*, cars, roads, *etc.* does not vary too much, even across two domains. In the result, misclassifications that may be caused by similar proportions of pixels (*e.g.* rider vs person) may suggest defect in their criterion, as similar label distribution can not guarantee pixels being assigned correct label, but it provides complementary ideas in regularizing domain shift.

An algorithm to learn domain-invariant representations in an adversarial manner is originally proposed by Ganin *et al.* [94], [95]. Their algorithm, *i.e.*, domain-adversarial neural network (DANN) introduces a domain classifier with a gradient reversal layer in order to confuse the classifier which domain the input data is from. Together with the domain classifier, a task specific network is jointly trained through normal back-propagation process, *i.e.*, gradient descent to improve its performance on target domain data in the task.

This whole process can be described as a mini-max game as the parameters of deep feature extractor tries to minimize the task specific loss while maximizing the domain classification loss, during which the task discriminative and domain-invariant feature representations are allowed to arise. Lee *et al.* [96] applied this idea in vehicle re-identification task and obtained improved performance compared with the baseline. Saito *et al.* [97] developed this idea by proposing an algorithm called "Maximum Classifier Discrepancy" (MCD), where the two players in the minimax game become to a feature generator and two task-specific classifiers (with no domain classifier). Lee *et al.* [98] further enhanced the MCD performance by introducing Wasserstein Distance to measure the discrepancy instead of absolute values of output difference in the initial MCD. The approach of Conditional Domain Adversarial Networks (CDAN) by Long *et al.* [99] replaces the representation used to train the discriminator in DANN with a combination of the representation and the classification prediction via the randomized multilinear map, to encode class information into domain information. To reduce the domain-specific information encoded in feature representation, Luo *et al.* [100] proposed a constraint on latent space by applying penalty on KL divergence between the marginal distribution of latent representation and the standard Gaussian.

Domain Generalization The most significant difference between domain adaption and domain generalization is that the latter does not rely on the assumption that data from target domain is available during training. Therefore, domain generalization has more commonalities with real-world tasks, where the generation process and distribution of the test data are not predictable, and it is very likely that it is not *i.i.d.* with the training data.

Nevertheless, methods used in domain adaptation can still be exploited in domain generalization approaches. Similar to the approaches of distribution alignment in domain adaptation, extensive studies [101]–[107] proposed to align representations across multiple source domains $D_S^{(k)}$, for $k \in \{1, ..., K\}$ and $K > 1$, with the objective to learn (source-)domain-invariant representations. However, without the support of data from target domain, representations learned with these methods can not be theoretically guaranteed to be invariant to unseen domains. It is also reported that with a proper data augmentation technique, models can outperform the distribution alignment methods [108].

2.2.2 Fine-tuning

Fine-tuning in transfer learning is an approach designed to tailor pre-trained models to specific tasks or datasets. This process addresses a fundamental challenge in machine learning: the need for large, diverse datasets for training robust models.

Initially, a model is pre-trained on a large, general dataset (source domain), where it learns a broad range of knowledge. These knowledge represented as learned weights serve as the starting point. During fine-tuning, the model is further trained (usually with a reduced learning rate) on a smaller, target dataset (target domain). This process involves selectively retraining (*i.e.*, fine-tuning) some layers of the network while keeping others fixed, in order to adjusts the model's parameters to optimize performance on a smaller, task-specific dataset. The degree to which layers are fine-tuned can vary; in some cases, only the final layers are updated, while in others, more extensive fine-tuning is conducted.

The goal is to leverage the generic knowledge learned during the initial training while adapting the model to capture the nuances of the new task. This balancing between retaining learned knowledge and adapting to new information is at the heart of fine-tuning. By doing so, fine-tuning solves the problem of data scarcity and specificity, allowing for the efficient application of powerful, pre-trained models to a wide array of specialized tasks. In this section, we conduct literature review in two areas: computer vision (CV) and natural language processing (NLP).

Computer Vision

In the context of computer vision, fine-tuning has proven particularly advantageous and effective due to the hierarchical nature of learned features in convolutional neural networks (CNNs). In the layered structure of CNNs, lower layers process basic patterns and often capture low-level features (*e.g.*, edges and textures in images), which are applicable across various tasks, while higher layers tend to handle high-level concepts and learn task-specific features. Therefore, the higher layers are more adaptable through fine-tuning and are usually tailored to specific vision tasks [109]. In this way, by leveraging existing knowledge of visual features, the general features in pre-trained models can be repurposed to enhance learning efficiency and accuracy for specific tasks which share similar visual properties.

In this section, we review in literature the role of fine-tuning in enhancing the performance of deep convolutional neural networks. We classify the existing studies into four categories, based on the degree of fine-tuning involved. Then we choose a representative study from each category for an in-depth review.

Head Fine-tuning In their study, Wan *et al.* [110] focused on improving radar signal sorting using a deep transfer learning framework. The authors deployed unmanned aerial vehicle swarms to collect data from various areas (source domain), which is then used to pre-train deep learning models. Subsequently, these models were fine-tuned using data from specific target areas (target domain). The fine-tuning strategy involved using the lower layers of convolutional neural networks, which tend to capture more abstract and universal features, and adapting the top fully-connected layers to specific problems. By effectively mitigating issues like interference and missing pulses in main sorting processing, this approach shows higher signal sorting accuracy compared to baseline methods. The mean Average Precision (mAP) scores for models pre-trained with radar signal are higher than both the baseline (without pre-training) and ImageNet pre-trained models, demonstrating the effectiveness of transfer learning. The results also indicate that using data closely resembling the target domain as the source domain significantly facilitates knowledge transfer and improves model accuracy.

Change-Head Fine-tuning In the study [111], the researchers removed the fully-connected head layers and employ a new head layer during fine-tuning based on a modified ResNet50 [112] architecture for leaf classification. Additionally, this study used a two-phase fine-tuning approach. In the first phase, the initial layers of the pre-trained ResNet50 model are frozen to preserve the generic features they have learned. The newly added layers are trained using a different learning rate obtained by the one-cycle policy. In the second phase, the entire model is unfrozen and trained, during which, the image size is progressively increased from 80 to 180 pixels. This method allows the model to leverage both generic features learned from ImageNet and specific features relevant to leaf classification.

Last-Layer Fine-tuning Ay *et al.* [113] focused on leveraging deep transfer learning for classifying stages of pressure injuries using various convolutional neural networks. Six different CNN models pre-trained on ImageNet were employed in this study, including DenseNet121, InceptionV3, MobilNetV2, ResNet50, ResNet152, and VGG16. During fine-tuning, only the parameters of the last layer are updated through back propagation, while the parameters of all the other layers are kept frozen. This approach, combined with early stopping and regularization techniques (dropout and L2 regularization), effectively adapts these pre-trained networks for the specialized classification task in medical imaging. Among the six CNN models, the ResNets consistently demonstrate superior performance over the other architectures.

Full-Model Fine-tuning The study in [114] proposed a novel transfer learning approach to address the challenge of limited training data in medical imaging tasks. This approach involved training a convolutional neural network on a large number of unlabeled medical images and then fine-tuning it on a smaller, labeled dataset for specific tasks like skin and breast cancer classification. The study employed parallel convolutional layers with various filter sizes and multiple skipping connections in the CNN architecture for effective feature extraction. During training, techniques including batch normalization, dropout, and global average pooling were also incorporated to optimize training and reduce over-fitting. The model, with its hybrid architecture, is subjected to a two-phase training procedure. It is first pre-trained from scratch on the unlabeled source domain dataset, and then fine-tuned on the labeled target domain dataset for the specific classification task. The experimental results demonstrate that models trained with this two-phase fine-tuning approach consistently exhibit an approximate 10% improvement in F1-score for challenging classification tasks, compared to models trained from scratch.

Natural Language Processing

In the area of NLP, due to its advantage in scalability, the transformer architecture has become the predominant framework. Transformers are typically pre-trained on vast textual corpora data and undergo fine-tuning tailored to enhance performance on specific downstream tasks. Several popular fine-tuning strategies are discussed in this section.

Full-Model Fine-tuning The study in [115] focused on the variance in performance during the fine-tuning of the entire BERT (340 million parameters) [116], with the final layer being randomly initialized. Extensive experiments were conducted on the GLUE benchmark, modifying only random seeds controlling weight initialization and training data order. The authors observed significant performance differences due to these two factors, and some seeds consistently perform better than others across different tasks. In certain cases, the fine-tuned BERT even demonstrates comparable performance with more newer models. Based on these observations, the authors recommended best practices for early stopping using their algorithm that monitors the training curves and correlation plots. This early stopping strategy help optimize the fine-tuning efficiency and save computational resources.

Prompt and Model Fine-tuning In addition to full model fine-tuning, Ben *et al.* [117] proposed to train prompts that are crucial parameters in down-stream tasks. This study introduced PADA, an innovative "Prompt learning for Any-Domain Adaptation" in NLP. The core of PADA's methodology involves generating prompts based on Domain Related Features (DRFs), which are semantically significant tokens associated with source domains. These prompts are then used to predict task labels for new domains. The training process of PADA is a two-step framework. Initially, the model is fine-tuned to generate prompts based on an example's domain. Subsequently, it is trained on the example's label using these prompts. Tested across various NLP tasks, PADA demonstrates significant improvements over existing methods, especially in scenarios involving multi-source adaptation.

Parameter Efficient Fine-tuning Lester *et al.* [118] presented an innovative approach to fine-tune pre-trained language models with the use of "prompt tuning" technique for specific NLP tasks. This technique introduces a small set of trainable parameters act as the prompt and prepend them to the input while keeping the parameters of the original language model frozen during fine-tuning. The experiments show that prompt tuning can achieve comparable or improved performance to full model fine-tuning, especially in scenarios where generalization to different domains is crucial. The approach effectively leverages the pre-trained knowledge in large language models like T5 while updating a significantly smaller number of parameters, making it a more efficient method for adapting large language models to various tasks.

Hu *et al.* [119] introduced an alternative approach of parameter efficient fine-tuning. They introduced the "Low-Rank Adaptation" (LoRA) of large language models, like GPT-3, significantly reducing computational requirements. Specifically, LoRA injects trainable low-rank matrices A and B into the self-attention modules in each transformer block, in order to approximate the weight update of parameters of the original transformer during fine-tuning. Formally, this can be expressed as $W_0 + \Delta W = W_0 + BA$, where W_0 represents the original parameters. In this way, W_0 can be kept frozen during fine-tuning and only parameters of A and B are updated instead. This method reduces the number of trainable parameters by up to $10,000$ times and GPU memory requirement by three times, compared to full model fine-tuning. Remarkably, it maintains or even improves performance across various benchmarks and backbone architectures.

2.3 Summary

In summary, this chapter provides a detailed review of two key areas relevant to the thesis: the concept of generalization in deep learning and the methodologies of transfer learning. Generalization is firstly explored, including its definition, factors affecting it, and existing challenges. Then we examine various techniques of transfer learning, focusing on domain adaptation and fine-tuning of pre-trained models for new tasks or domains.

It is shown in the literature review that our understanding of model generalization, particularly in the *o.o.d.* setting, remains limited, with significant gaps in research. Moreover, the in-depth investigation into various transfer learning methodologies indicates that domain adaptation is not very effective in addressing *o.o.d.* generalization challenges, given its reliance on the knowledge in target domain. This constraint limits its adoption in our study. Instead, our approaches predominantly focus on leveraging knowledge by explicit exploitation with InterpretNet (Chapter 4), and implicit transferring from pre-trained models via fine-tuning (Chapters 4 and 5).

Chapter 3

Towards Learning Generalizable Knowledge of 2D Transformations

The first question we focus on is the learnability of generalizable knowledge. In this chapter, we devise a synthetic dataset and corresponding learning tasks based on causal theory, aiming to investigate the methodology for learning knowledge of 2D transformations, including rotation, scaling, and translation. Our objective is to enable DNN models to possess a human-like understanding of 2D transformations, such that the machine should be capable of determining whether, and to what extent, an image has been transformed, regardless of the image's semantic domain. The experimental results indicate that through our designed methodology, the knowledge acquired by DNNs exhibits a degree of generalizability. Interestingly, DNNs can even learn patterns of 2D transformations from meaningless black-and-white noise. This finding has provided us with the potential tools to disentangle specific knowledge, and laid the foundation for further exploiting this knowledge.

3.1 Introduction

As introduced in Chapter 1, we can think of image generation process as a result of the interaction of various mechanisms. To begin our exploration into the learnability of generalizable knowledge of a mechanism, it is essential to first define what we mean by *the knowledge of a mechanism.* We take the 2D rotation of images as an example of such mechanisms. As human beings, if we have gained the knowledge of 2D rotation, it means that for any image, with a proper tool, (a) we can rotate the image at will, and (b) we are able to determine whether (and even how many degrees) the image has been rotated. Obviously, the knowledge that we know about 2D rotation generalizes systematically and is independent of the domain of images. For transformation mechanisms studied in this work, the affine transformation functions are in accord with the description in (a), and are used as a tool to make precise operations[1]. Therefore, our main purpose is the learning of the latter aspect (b), *i.e.*, the estimation of transformation parameters. To achieve this, we devise a new training methodology and use synthetic datasets generated with the target transformation mechanisms for training.

It has been found that with this training methodology, the transformation parameters can be estimated accurately and stably, even when networks are trained on random noise and tested on images of semantically different domains. Additionally, when the transformation is matched with the inductive bias of the model, it exhibits some interesting properties as a by-product, with which, we can actually restore (to some extent) the transformed images from the originals using only gradient descent.

To the best of our knowledge, this is the first work that attempts to learn generalizable knowledge about a specific mechanism. The main contribution of this chapter lies in our development of a learning methodology based on causal theory, through transformation estimation tasks exploiting synthetic datasets. Using this methodology, DNNs are able to disentangle the concept of 2D transformation mechanisms from confounding factors, thereby robustly acquire generalizable knowledge of these mechanisms.

Real-world images can be considered as the result of the interactions between mechanisms, such as foreground and background objects, lighting conditions, camera attributes, *etc.* Additionally, with the rapid development of computer graphics, photo-realistic synthetic datasets with 1) controlled interventions on target factors of variation, and 2) automatic pixel-accurate annotations, can be efficiently created

[1] It does not imply that transformation operations cannot be learned from data. Generative models, which are beyond the scope of this study, have been studied in various tasks [120], [121].

with 3D rendering engines. Therefore, the proposed methodology offers a potential toolkit for learning generalizable knowledge of diverse mechanisms in real-world data.

In the following sections, we first review research works related to our study in Section 3.2. We then propose the methodology to learn generalizable knowledge of 2D Transformations in Section 3.3. Details of experiments and results are presented in Section 3.4. Finally, the chapter concludes in Section 3.5.

3.2 Related Works

In this section, we provide a brief overview of research works and techniques that are relevant to this study, showing how these various research areas provide the theoretical and technical foundations for our study. However, our work also distinguishes itself in many aspects, which are illustrated in the respective sub-sections.

Data Augmentation and Domain Randomization

To tackle the potential performance drop in *o.o.d*, commonly used effective techniques include data augmentation [122]–[126] and domain randomization [127], [128]. Data augmentation plays a crucial role in computer vision by expanding the size and diversity of training datasets, reducing overfitting, and enhancing the accuracy of machine learning models. In this section, we briefly review recent works in computer vision to illustrate various data augmentation techniques.

Geometric and color transformations such as rotation, shearing, translation, contrast, brightness, and color jittering, are widely used techniques. Researchers often combine these transformations to improve performance. Therefore, Cubuk *et al.* [123] proposed a search space for automated augmentation strategies that control all operations jointly. This technique has led to reduced computational expense and improved performance across various tasks (*e.g.,* 1.0 - 1.3% accuracy improvement on object detection tasks). Noise injection is another commonly used technique. Kar *et al.* [125] developed an approach that generates noise and corruption by incorporating 3D information consistent with the scene geometry. This approach includes corruptions such as motion blur, fog, *etc.*, which better represents distribution shifts occurring in the real world, leading to a lower error rate across various tasks (*e.g.,* 1.56% l_1 error reduction on the AE benchmark). Synthetic image generation is gaining attention in computer vision. Hao *et al.* [126] proposed MixGen, a technique that generates new image-text pairs preserving their semantic relationships, thus enhancing data efficiency. This technique achieved significant performance improvements (a 6.2% accuracy boost on the COCO fine-tuned image-text retrieval task).

In particular circumstances, each of these techniques has its own advantages and disadvantages. For example, in the case of multi-modal pre-training which is growing in influence in computer vision, geometric and color transformations may result in mismatching of image-text pairs, thereby leading to unnecessary data pollution within multi-modal datasets. Synthetic data generation may be more suitable in such cases, even though they may require additional computational resources.

The technique of domain randomization aligns with the underlying principles of data augmentation. Data augmentation is primarily associated with 2D transformations, whereas domain randomization is typically employed for parameter manipulations in 3D environments. From a causal perspective, both techniques use treatment randomization to eliminate confounders and to enhance the learning of invariance. Based on this principle, our work also produces synthetic datasets through treatment randomization, but for a different purpose. Instead of randomizing *out* the mechanisms of variation, we aim to take them *into* consideration for downstream tasks.

Parameter Estimation

As introduced above, the purpose of learning mechanisms of 2D transformations is to estimate the transformation parameters. This task has been extensively studied in various computer vision topics, such as 2D spatial invariance learning [129], object detection [130], [131], and 3D pose estimations [132], [133], among many others. However, in most existing studies, parameter estimation is restricted to object categories that appear in the training sets. An important reason is that single-image parameter estimation is an ill-defined problem, in the sense that parameters of transformations are actually procedural variables, whose values are determined by both of the pre- and post-transformation states. The analysis and results in Sections 3.3.1 and 3.4.3 show that models trained with methodologies based on single images, are not able to generalize to unseen categories. In this work, we seek to develop a parameter estimation ability that should display a certain degree of generalizability, similar to that exhibited by humans. Another series of works [134], [135] and the study in [136] conducted representation learning based on pairs of images that are related through mechanisms, by using a single encoder for multiple mechanisms. In this work, to eliminate the potential confounding between multiple mechanisms, we try to isolate knowledge of single mechanisms and reuse them in downstream tasks.

Time Series Analysis

Deep learning studies on time series cover almost every field of real-world applications, because of its inherent connection with the temporal dimension of the world. These applications include geophysical processes modeling [137], human physical [138], [139] and mental [140]–[142] activity analysis, cybersecurity [143], to name a few. If we consider the transformations of images as sequential processes, and focus on the most critical time slices which are those before and after the transformations, we can see this study as related to a time series analysis. Architectures such as Convolutional Neural Networks (CNNs) [139]–[141], [144], Long-Short-Term-Memory (LSTM) [140], [145], Extreme Learning Machine (ELM) [144], [146], *etc.,* are widely used in research on time series. CNN is adopted in this study to better model 2D image transformations.

Program Induction

Knowledge learning in this work is essentially a program induction problem. Active deep learning topics in this area include program synthesis [147], [148], image generation [149], [150], *etc.* Program induction aims for a more effective generation of programs, whereas this work focuses more on the interpretation of images. Therefore, domain-specific languages in this work are fundamentally different, being more semantically relevant to the downstream tasks.

Based on the above overview of related works, it is clear that our study is distinctive in its unique motivation of exploring the learnability of generalizable knowledge.

This motivation has guided us to develop a novel approach to disentangle a target mechanism using synthetic datasets in specifically designed tasks, which underscores our unique contributions to the existing body of knowledge.

3.3 Methodology

As introduced in Section 3.1, our objective in this chapter is to learn the generalizable knowledge of 2D transformations. To achieve this, we train the models in parameter estimation tasks using synthetic datasets that are generated through target transformations. In the following sections, we first describe how the datasets are constructed in Section 3.3.1 and then describe the training method in Section 3.3.2.

3.3.1 Synthetic Datasets

To facilitate the learning of generalizable knowledge about a mechanism with DNNs, the underlying principle guiding the synthesis of a training set is described below. Generally, let us denote by $\mathbf{x}$ and $\mathbf{x}_f$, respectively, the images before and after transformation f (parameterized with $\boldsymbol{\theta}$). Thus, we have

$$\mathbf{x}_f = f(\mathbf{x}; \boldsymbol{\theta}). \tag{3.1}$$

Note that, $\boldsymbol{\theta}$ here can be a vector, representing any transformation parameters. In this study, $\boldsymbol{\theta}$ represents the rotation angle, the scaling factor, the translation offsets, or the combination of these. As explained in the Introduction, the goal of the knowledge learning is to estimate the value of transformation parameter $\boldsymbol{\theta}$. Let X, X_f and Θ be the variables from which $\mathbf{x}$, $\mathbf{x}_f$ and $\boldsymbol{\theta}$ are instantiated, respectively. According to the causal graph in Figure 3.1, if the estimation is made based only on the image *after* transformation, *i.e.*, $\mathbb{E}(\Theta|X_f)$, given that X_f is a collider, conditioning on it will inevitably cause the information flow from U to Θ, which will hinder us from learning stable and thus generalizable knowledge of f (via Θ). Therefore, in order to remove confounding caused by U, thus making the prediction of Θ generalize better in test domains, we have to condition on both X and X_f, *i.e.*, the Markov blanket of Θ. [2]

[2] This is also intuitively true, because it is pointless to ask how a picture has been transformed when no reference is provided.

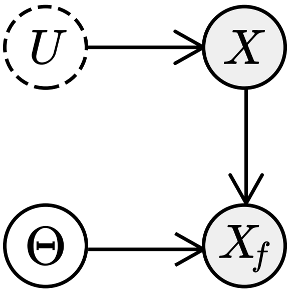


Figure 3.1: The causal graph of image transformation. X: Image before the transformation. X_f: Image after the transformation. Θ: Parameter(s) of the transformation in this study, as the variable is randomly sampled, this "treatment randomization" operation removes all arrows pointing to Θ. U: Other unobservable variables that cause the generation of X.

Concretely, in knowledge learning we aim to compute $\mathbb{E}_{P_{test}}(\Theta|X, X_f)$ given only access to $P_{train}(\mathbf{x}, \mathbf{x}_f, \boldsymbol{\theta})$. The Covariate Shift Assumption and Same Support Assumption, *i.e.*,

$$P_{train}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{x}_f) = P_{test}(\boldsymbol{\theta}|\mathbf{x}, \mathbf{x}_f) \text{ and,} \tag{3.2}$$

$$supp_{train}(\mathbf{x}, \mathbf{x}_f) = supp_{test}(\mathbf{x}, \mathbf{x}_f), \tag{3.3}$$

are required for the causal model to work, where P_{train} and P_{test} are distributions of data in training and test sets, and $P_{train}(\mathbf{x}, \mathbf{x}_f, \boldsymbol{\theta}) \neq P_{test}(\mathbf{x}, \mathbf{x}_f, \boldsymbol{\theta})$.

In this work, synthetic datasets for knowledge learning are constructed according to the above causal framework. Each data point is composed of a pair of images $\mathbf{x}$ and $\mathbf{x}_f$ that are before and after the transformation, and the transformation parameter $\boldsymbol{\theta}$. Since the labels are automatically generated and no manual annotation is needed, this can be viewed as a self-supervised learning problem.

3.3.2 Knowledge Learning

To explore DNNs that are capable of learning generalizable knowledge, we investigate a less studied Convolutional Neural Network (CNN) model, which takes *concatenated* image pairs as input (shown in Figure 3.2(a)). This model is referred to as "CNN_pair" in this thesis. Additionally, we select two commonly investigated CNN models as our baselines because of their relevance to this research, namely, the Siamese networks [151] (Figure 3.2(b)) and a vanilla CNN (Figure 3.2(c)). The Siamese networks, extensively studied on datasets with intrinsic relations in metric learning and representation learning, also take image pairs as input during training. Vanilla CNN, which is another common method for numerical regression tasks, takes single-images as input and is denoted as "CNN_single" in this thesis.

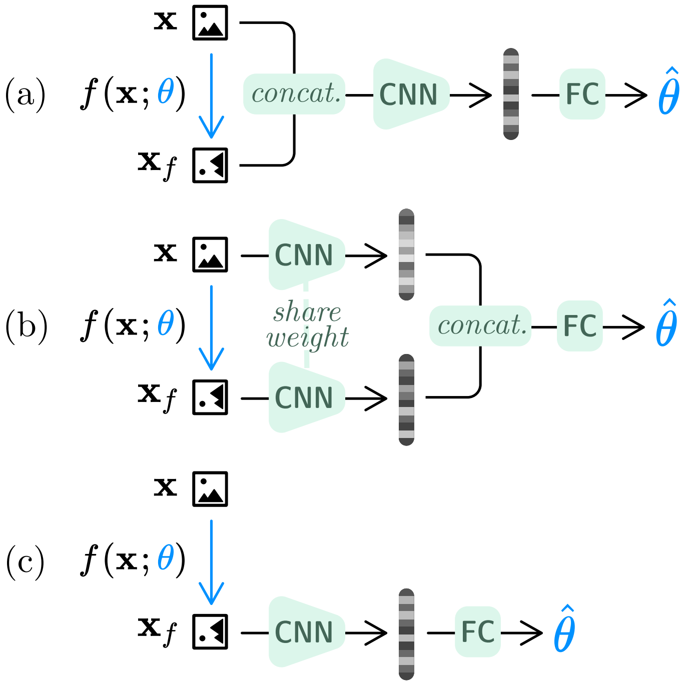


Figure 3.2: The forward process of three CNN models used for knowledge learning. (**a**) CNN_pair: paired images $\mathbf{x}$ and $\mathbf{x}_f$ are concatenated in channel dimension before being fed into CNN. The transformation information is encoded as representations in the latent space, which are then sent to the fully connected (FC) layer; (**b**) Siamese network: $\mathbf{x}$ and $\mathbf{x}_f$ are fed into CNN separately. The representations are then concatenated and fed into the FC layer; (**c**) CNN_single: Only the transformed images $\mathbf{x}_f$ are fed into CNN and encoded. The representations are then linearly transformed through the FC layer, and the 2D transformation parameters are predicted as output.

In parameter estimation tasks, the mean squared error is used as the loss function, *i.e.*,

$$\mathcal{L}_{MSE}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \frac{1}{Nk} \sum^{N} \sum_{i}^{k} (\hat{\boldsymbol{\theta}}_i - \boldsymbol{\theta}_i)^2, \tag{3.4}$$

where N is the batch size, and $\hat{\boldsymbol{\theta}}_i$ denotes the i-th dimension of $\hat{\boldsymbol{\theta}} \in \mathbb{R}^k$.

Therefore, the objective of knowledge learning is

$$\arg\min_{\alpha} \mathcal{L}_{MSE}(f_{\alpha}(\mathbf{x}, \mathbf{x}_f), \boldsymbol{\theta}) \tag{3.5}$$

where f_α is a deep neural network for knowledge learning parameterized with α.

3.4 Experiments

In this section, experiments are conducted to answer the question raised in the Introduction, *i.e.*, *whether the learned knowledge of transformation mechanisms can exhibit some degree of generalizability?*

In order to study the robustness of the estimations on $\boldsymbol{\theta}$ of f, synthetic datasets are constructed according to the procedure described in Section 3.3.1. Three DNN models are trained and tested based on the methodology illustrated in Section 3.3.2, with the training details described in Section 3.4.1. Next, we examine the generalizability of learned knowledge in Sections 3.4.2. Further discussions are conducted in Section 3.4.3.

3.4.1 Training

Datasets

In the experiments, the original images in MNIST, EMNIST [152] and CIFAR-10 [153] are randomly transformed before being used as $\mathbf{x}$ to alleviate the potential overfitting. We obtain $\mathbf{x}_f = f(\mathbf{x}; \boldsymbol{\theta})$, where the transformation parameters $\boldsymbol{\theta}$ are randomly sampled in a uniform distribution (see Table 3.1).

In this work, we conduct learning on four types of f, including individual transformations of rotation, scaling and translation, and the joint transformation of the above three. For learning individual transformations, only one of the three transformations is applied at a time, while in the joint case, all three transformations are applied simultaneously.

Furthermore, to demonstrate that the generalizable knowledge is independent of the domain of images, a synthetic dataset composed of black/white noises (of a Bernoulli distribution) is randomly generated and used as $\mathbf{x}$. To better test generalizability, all

Table 3.1: The parameters of 2D transformations investigated in experiments. Each parameter is uniformly sampled within its range.

Parameter	Range
Rotation angle	$[-90^\circ, 90^\circ]$
Translation (horizontal)	$[-5, 5]$ pixels
Translation (vertical)	$[-5, 5]$ pixels
Scale factor	$[0.7, 1.3]$

test data are sampled from datasets that are semantically different from the training sets. Three groups of experiments are conducted, whose detailed schemes are listed in Table 3.2.

Model Settings for Knowledge Learning

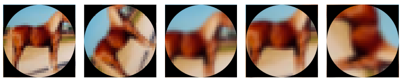
The CNN model in [134] is used as the backbone in CNN_pair and the two baselines (*i.e.*, the Siamese network and CNN_single). All input pairs of $\mathbf{x}$ and $\mathbf{x}_f$ are concatenated along the channel dimension before being fed into the CNN_pair. Thus, the input dimension is $N_{batch} \times 2 \times 28 \times 28$ in Exp_MNIST and Exp_NOISE, and $N_{batch} \times 6 \times 32 \times 32$ in Exp_CIFAR, where N_{batch} is the batch size. We keep the default settings for the baselines.

We follow the implementation in [134] to construct the three models (in Figure 3.2) for knowledge learning experiments. The architectures for individual mechanism learning are shown in Table 3.3. The models for joint learning are different only in channel sizes, which are all doubled in Exp_MNIST and Exp_NOISE, and 50% larger in Exp_CIFAR.

Training Details

The CNN models are trained using Adam optimizer [154] with a batch size of 512 and the weight decay set to 5.0×10^{-4}. In Exp_MNIST and Exp_CIFAR, the models are trained for 500 epochs in each experiment, with the learning rate initialized to 0.03 and decaying by a factor of 0.6 for every 50 epochs. In Exp_NOISE, since the noise images are generated on-the-go, the models are trained for 1.0×10^5 steps with the same batch size of 512. The initial learning rate is also set to 0.03 with a

Table 3.2: The training and test data used in the three groups of experiments for knowledge learning. Five example images are provided for each dataset to demonstrate the 2D transformations in each experiment. These transformations, shown from left to right, include the original image, rotation, translation, scaling and a combination of the three. To prevent potential artifacts being generated during transformations, such as slanted image edges, a circular mask is applied to CIFAR-10 and Noise images.

Experiment	Training set	Test set
Exp_MNIST	MNIST (training) 	EMNIST (test, ‘letter’ division)
Exp_CIFAR	CIFAR-10 (training, 9 classes) 	CIFAR-10 (training, the remaining class)
Exp_NOISE	black/white noise 	MNIST (test)

decaying factor of 0.5, and a decaying cadence of 1.0×10^4 steps.

The codes of our methodology are publicly available. [3]

3.4.2 Learning of 2D Transformation Mechanisms

Individual Learning

The performance of CNN_pair in learning the knowledge of the three individual transformations is presented in Figs. 3.3 and 3.4. It can be observed in Figure 3.3 that the most of the absolute percentage errors (APE) (*e.g.* the third quartile in the distributions) are below 20% in most experiments for CNN_pair. Moreover, because of the domain shift between the training and test sets, varying degrees of distribution shifts of the APE can be observed in Figure 3.4. However, the shift is significantly smaller for model CNN_pair compared to the other two models. When considering the shift of median in APE distribution, averaged across all three transformations, the CNN_pair exhibits a significantly lower shift of 2.5% APE between training and

[3] Our codes are released at `https://github.com/jiachenkang/InterpretNet`.

Table 3.3: The architecture of models for knowledge learning.

Models in Exp_MNIST and Exp_NOISE	Models in Exp_CIFAR
5×5 Conv 96, BatchNorm, ReLU	5×5 Conv 192, BatchNorm, ReLU
1×1 Conv 64, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 64, BatchNorm, ReLU
3×3 MaxPooling stride 2	3×3 MaxPooling stride 2
3×3 Conv 32, BatchNorm, ReLU	3×3 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
3×3 MaxPooling stride 2	3×3 MaxPooling stride 2
3×3 Conv 32, BatchNorm, ReLU	3×3 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
3×3 MaxPooling stride 2	3×3 MaxPooling stride 2
2×2 Conv 32, BatchNorm, ReLU	2×2 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
1×1 Conv 32, BatchNorm, ReLU	1×1 Conv 128, BatchNorm, ReLU
3×3 MaxPooling stride 2	3×3 MaxPooling stride 2
FC	FC
FC (Siamese networks only)	FC (Siamese networks only)

test sets. In contrast, the Siamese network and the CNN_single present shifts of 9.2% and 76.8% APE, respectively.

The minor distributional difference of APE in above results indicates the robust generalizability of 2D transformation knowledge learned by CNN_pair. This is a noteworthy finding, considering that the data in the training and test sets differ completely in terms of semantics.

Further results demonstrating the performance of CNN_pair in learning individual 2D transformation across the three dataset settings are presented in Figure 3.5. Similar to the result shown in Figure 3.3, it is observed that the majority of APE can be controlled below 20% for each learning cases of individual transformation. Moreover, the distributions of APE between the training and test sets consistently

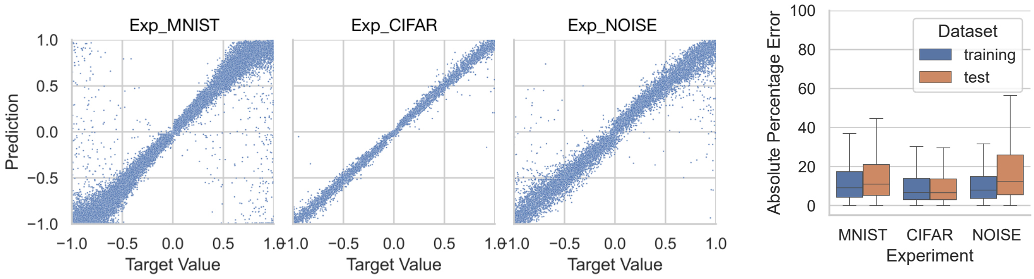


Figure 3.3: The performance of CNN_pair for individual rotation learning. (**left**) Predictions of rotation angle *vs.* the ground truth (normalized to $[-1, 1]$) in test set. (**right**) Distributions of absolute percentage errors (in %) of all data points in the dataset.

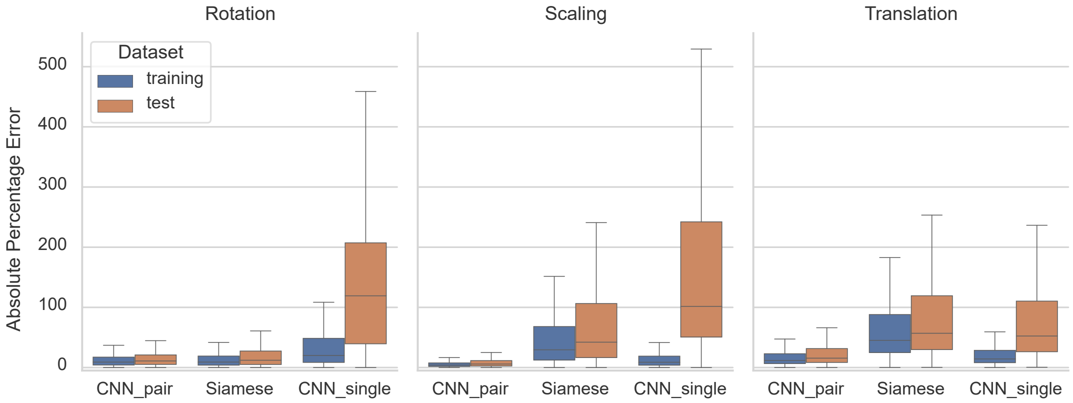


Figure 3.4: The performance of learning individual transformations across different models.

exhibit minor variations across all experiments, indicating robust generalizability. These findings collectively indicates the effectiveness of our methodology for learning generalizable knowledge of 2D transformation.

Joint Learning

In the case of joint transformation learning, a noticeable decline in performance for both training and test sets can be observed in Figure 3.6, compared to individual learning, despite the fact that the number of parameters in the CNN_pair used for joint learning is four times that of models used for individual learning. Similar results of decreased performance are reported in study [155], where more

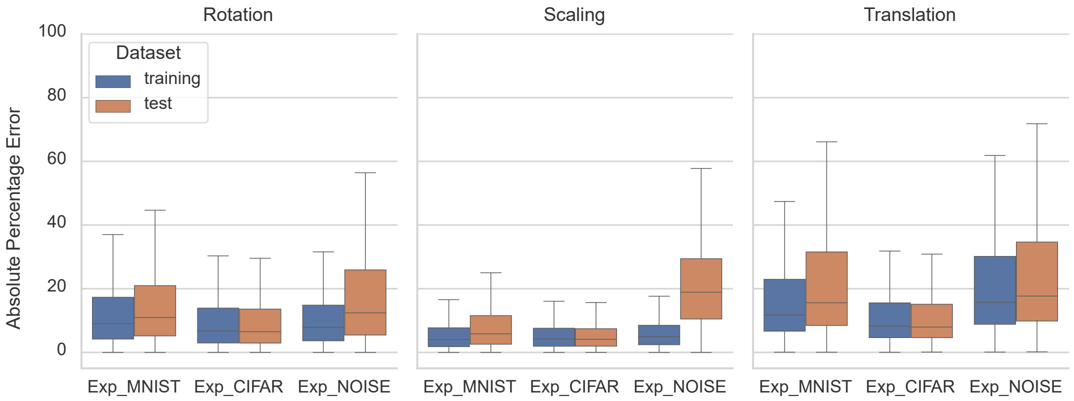


Figure 3.5: Performance of CNN_pair for individual 2D transformation learning. **(left)** Rotation. **(center)** Scaling. **(right)** Translation.

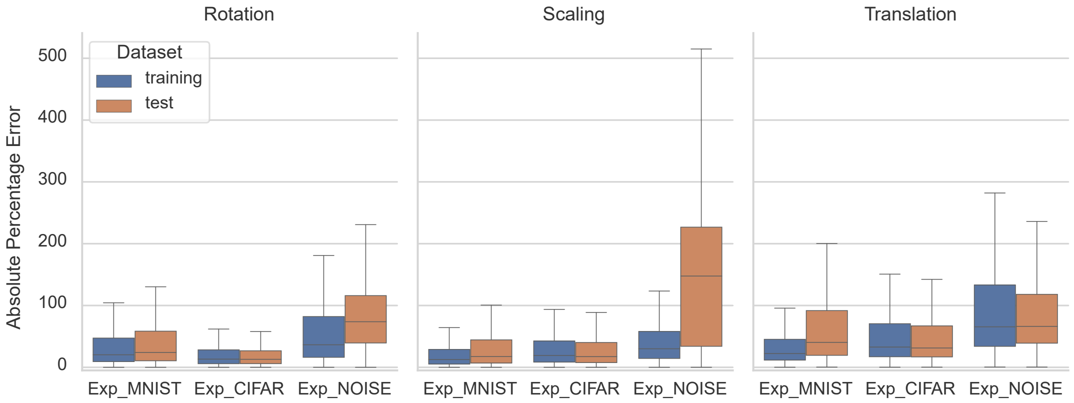


Figure 3.6: Performance of CNN_pair for joint 2D transformation learning. **(left)** Rotation. **(center)** Scaling. **(right)** Translation.

accurate estimates of variables are made by separately trained models, because of the improved “selectivity and invariance at the individual neuronal level”.

Nevertheless, the distributions differences of APE between the training and test sets continue to present negligible variations in most experiments, indicating robust generalizability. The notable exception is the EXP_NOISE case, which will be discussed in further detail below.

CNN_pair Trained in Exp_NOISE

While CNN_pair exhibits strong generalizability, its performance shows a relative decline when there is a significant pattern difference between the training and test sets. For instance, in the Exp_NOISE experiment, a more pronounced performance gap between the training and test set is observable, as compared to the other two experiments in Figures 3.5 and 3.6. The most apparent characteristic in the Exp_NOISE experiment is the distinct difference in patterns between noises and hand-written digits, suggesting a potential variation in exploitation of patterns during the training process.

To prove this hypothesis, an ablation study was conducted by altering the black-to-white pixels ratio in the training data of Exp_NOISE. As shown in Figure 3.7, the most best-performing model for rotation learning is trained with a $7:3$ ratio of black-to-white noises. However, when the pixel values in MNIST are swapped (*i.e.* black digits on white background), the best performance is observed around a $4:6$ ratio. The different ratios present distinct patterns that can be exploited in training. The ideal ratio for individual learning of translation and rotation is around $7:3$, while for scaling it is closer to $3:7$. This observation also explain the suboptimal *o.o.d.* generalization performance in joint learning in Exp_NOISE, since it is challenging for the model to equally well learn the three transformations with a single fixed ratio.

Validation on ImageNet

To further assess the effectiveness and adaptability of our proposed algorithm across different domains and data complexities, we extended our evaluation to include the ImageNet dataset [156], a popular benchmark in the field of computer vision.

The experimental setup for ImageNet closely follows that described in 3.4.1, but with a few dataset-specific modifications. We adopt a ResNet-50 architecture [112] as the backbone of CNN_pair model, modifying only the final fully connected layer to output the estimated transformation parameters. From the dataset, we selected images from 900 classes (with label indices under 900) for training, reserving images

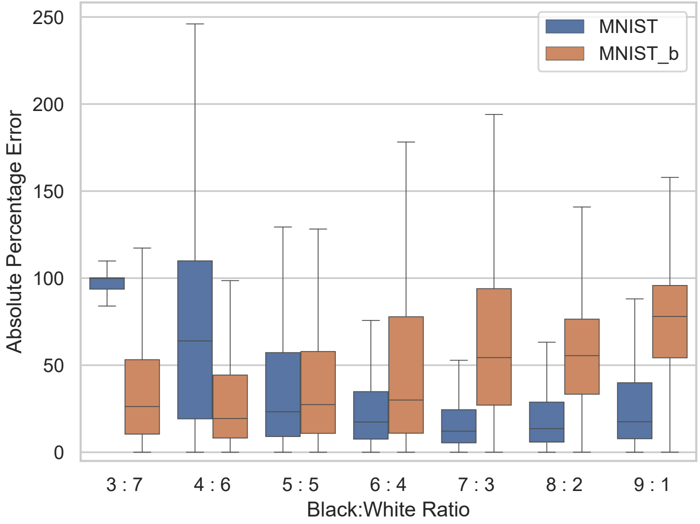


Figure 3.7: Performance of CNN_pair in rotation learning with controlled black-/white pixel ratios in EXP_NOISE. Pixel values are swapped in MNIST_b.

from the remaining 100 classes for testing. Each image undergoes standard preprocessing before being center-cropped to 224 pixels, resulting in an input dimension of $N_{batch} \times 6 \times 224 \times 224$, where N_{batch} represents the batch size. The rotation and scaling parameters are uniformly sampled within the ranges specified in Table 3.1, while translation offsets are set to the range of $[-40, 40]$ pixels. Training was conducted using the Adam optimizer [154] with a batch size of 256 and a weight decay factor of 5.0×10^{-4}. The models are trained for 100 training epochs, starting with an initial learning rate of 0.1, which is decayed every 10 epochs with a decaying factor of 0.5.

The experimental results, detailed in Table 3.4, reveal that the CNN_pair model maintains consistent performance across all conducted experiments, even if the semantic diversity and image size has increased in the ImageNet experiment. The negligible discrepancy between training and testing results indicates the model's excellent generalizability, especially considering the test images are from entirely unseen categories. These findings strongly support the effectiveness of our methodology in the acquisition of generalizable knowledge.

Table 3.4: Performance metrics of the CNN_pair across various experiments, reporting the mean and median Absolute Percentage Error (APE, %) for transformations of rotation, scaling and translation.

Experiment	Split	Rotation		Scaling		Translation	
		mean	median	mean	median	mean	median
ImageNet	training	40.58	11.56	11.27	5.32	42.78	19.92
	test	64.82	11.63	16.21	5.33	45.53	22.87
Exp_MNIST	training	27.89	9.04	20.75	4.02	67.85	11.78
	test	29.10	10.95	33.10	5.83	94.56	15.59
Exp_CIFAR	training	22.07	6.71	15.85	4.21	34.30	8.24
	test	19.33	6.50	21.48	4.11	32.26	7.95
Exp_NOISE	training	27.16	7.88	24.19	4.97	47.42	15.75
	test	82.96	12.42	165.39	18.90	78.26	17.68

3.4.3 Key Elements in Knowledge Learning

In this section, ablation results are discussed to examine elements crucial for generalizable knowledge learning.

Firstly, as analyzed based on the causal graph in Figure 3.1, if there exists a causal relationship from U to X, it is necessary to condition on both X and X_f in order to predict Θ robustly. As shown in Figure 3.4, the generalization degradation of CNN_single is much more severe in all learning cases, compared with CNN_pair and Siamese networks that both take paired images X and X_f as inputs. The translation learning of CNN_single generalizes relatively better than its learning of rotation or scaling, because the position of X (the original images in this case) is always in the center and independent of U. However, while being able to estimate rotation angles accurately in the training set, CNN_single completely fails in the test set. This is primarily because rotation angle estimation is heavily dependent on the pattern of images, which is determined by U. This finding provides valuable insights into numerical regression tasks in contemporary computer vision studies, such as object pose estimation. It suggests that when training is conducted solely with images after transformation, expecting robust generalization performance is unrealistic.

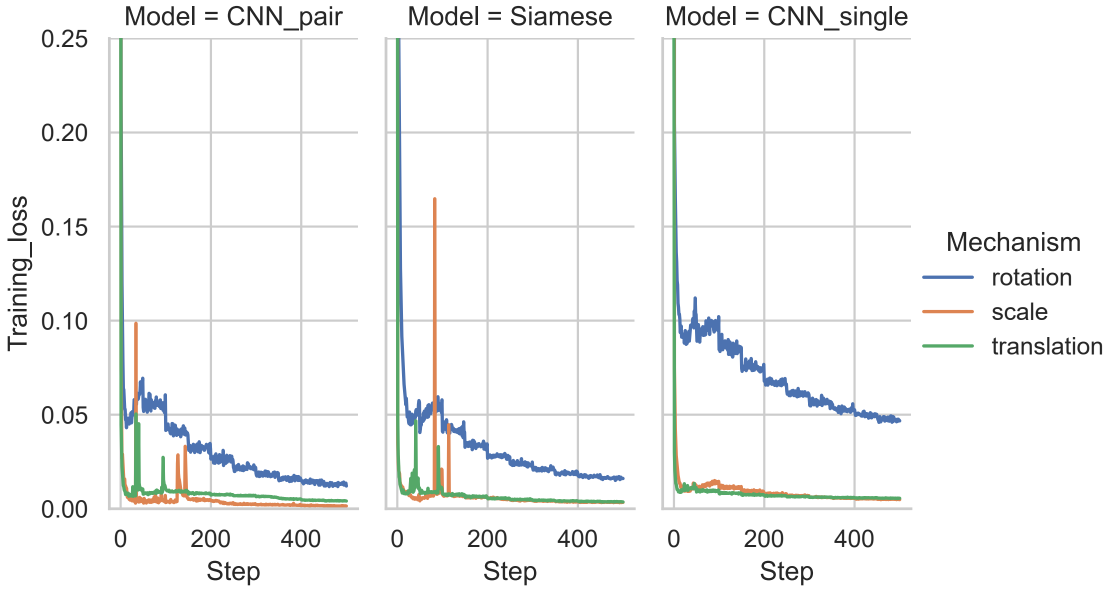


Figure 3.8: The learning curves in transformation learning across different models. Fast learning on translation and scaling and slower learning on rotation can be observed for all models.

Secondly, for CNN backbones, computation based on image-level concatenation (instead of feature-level) is beneficial for making more accurate estimates. Figure 3.4 shows inferior performance of Siamese networks, in comparison to CNN_pair across all learning tasks. For the Siamese networks, substantial information regarding transformations is lost through the application of convolutional and max pooling operations. In contrast, the CNN_pair retains a greater amount of information from the beginning of data processing.

Additionally, we speculate that the inductive bias of CNNs fundamentally affects the effectiveness of knowledge learning. This is based on the observation of the learning curves of the three mechanisms (in Figure 3.8). Across all three models, a rapid learning pace is evident in translation and scaling, contrasted with a slower one in rotation. This indicates that CNN models have a greater challenge in learning the mechanism of rotation.

Another interesting property of CNN_pair and Siamese networks can be found (only) in learning translations. Given two images $\mathbf{x}$ and $\mathbf{x}_T$ both with a small square in the center, and the target value of translation $\boldsymbol{\theta}_T$, we can obtain a (coarse) translated

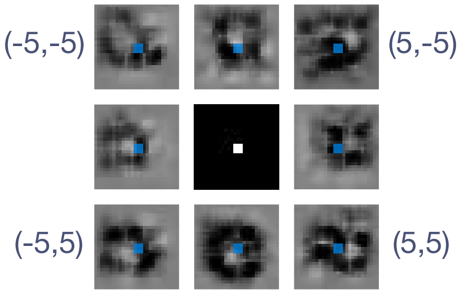


Figure 3.9: Images obtained with the Translation CNN_pair through gradient decent. The image in the center is the original one $\mathbf{x}$. According to the values of $\boldsymbol{\theta}$ (four of them are marked in the corners), $\mathbf{x}_T$ are generated through gradient descent. In each of $\mathbf{x}_T$, an obvious offset of the light area from the original position (the blue dot) to the target position can be observed.

version of $\mathbf{x}_T$ by optimizing $\mathbf{x}_T$ with gradient decent according to:

$$\mathbf{x}_T \leftarrow \mathbf{x}_T - \alpha \nabla_{\mathbf{x}_T} L_{MSE}(E(\mathbf{x}, \mathbf{x}_T), \boldsymbol{\theta}_T), \tag{3.6}$$

where α is the learning rate. As shown in Figure 3.9, this operation can be viewed as an approximation of the translation function f_T. Although this reversed generation of images is by no means accurate and only limited to very simple patterns, the phenomenon clearly shows what the models have learned.

Considering CNN's properties of translation-equivariance, positional information can be encoded and operated with CNN at higher efficiency. An extensive investigation into other inductive biases should be made in the future, to provide support for any more solid claim.

Impact of Masking

Initial experiments indicated a tendency for the model to overfit during training. A closer inspection of the training samples revealed that affine transformations could introduce artifacts, such as skewed edges, as shown in the grey regions around the corners in Figure 3.10 column (a). These artifacts were readily exploited by the model, negatively impacting its performance. To mitigate this issue, a circular mask was applied to the images, obscuring peripheral regions and compelling the

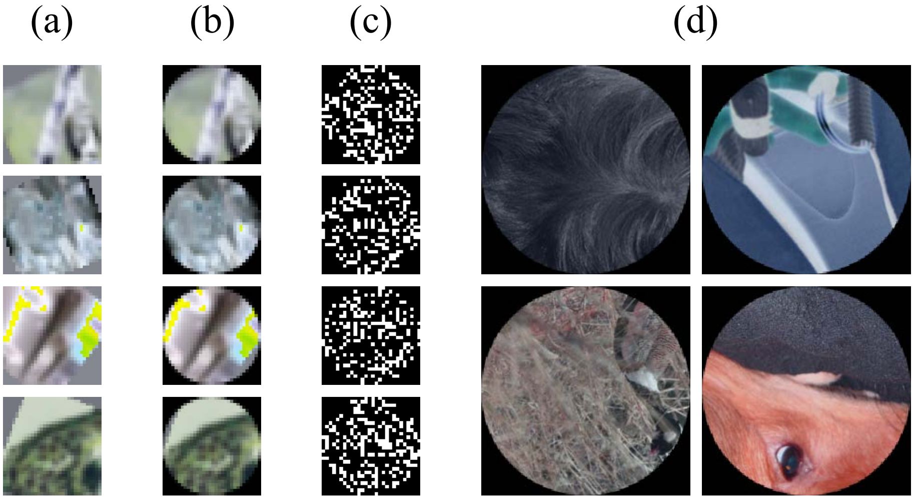


Figure 3.10: Examples of training and test data. **(a)** Transformed CIFAR-10 images without masks. **(b)** Transformed CIFAR-10 images with applied masks. **(c)** Masked black/white noise images. **(d)** Masked ImageNet samples.

model to focus on features in the center. Therefore, instead of simplifying the task, this masking strategy actually makes knowledge learning more robust.

Furthermore, the image preprocessing steps, including random affine transformations, color adjustments, and center cropping, *etc.*, often displace or obscure the primary subjects of the images, leaving behind non-informative textures, as shown in Figure 3.10 columns (b) and (d). Particularly in the Exp_NOISE, the dataset intrinsically lacks semantic content or any "main bodies" of subjects (Figure 3.10 columns (c)), further challenging the model's ability to discern meaningful patterns.

The experimental results, discussed in Section 3.4.2, confirm that models trained on masked noise images exhibit comparable performance to those trained on authentic imagery. This highlights the effectiveness of the masking technique in not merely raising the training challenge, but also in fostering more robust model performance.

3.5 Summary

In this chapter, we have conducted comprehensive experiments to address the learnability of generalizable knowledge of 2D image transformations. The experimental results have demonstrated that learning such knowledge is possible if the CNN_pair model is trained on synthetic images that are intrinsically related through the transformation. The CNN_pair model has exhibited a notably lower shift in the average of median APE, as low as 2.5%. This performance is markedly better compared to 9.2% and 76.8% observed in the Siamese and the CNN_single models, respectively. This result indicates robust generalizability of the learned knowledge, irrespective of the semantic domain difference of images. Therefore, this study introduces a potential toolkit for learning other forms of generalizable knowledge, that is by disentangling the concepts from confounding factors using parameter estimation tasks based on causal datasets.

Chapter 4

Improving 2D/3D Classification with Learned Generalizable Knowledge

In the previous chapter, we have gained some insights into how to learn knowledge with a degree of generalizability, which involves disentangling the target concept from confounding factors. These insights naturally prompt a further question: *Given that generalizable knowledge is learnable, how can we exploit the learned knowledge in real tasks?* In this chapter, we endeavor to explore two distinct paradigms (*i.e.* the InterpretNet and regression-loss-integrated self-supervised learning) to address this question.

4.1 Introduction

In this chapter, two distinct paradigms for generalizable knowledge exploitation are investigated, *i.e.*, explicit and implicit exploitation. For the first paradigm, we introduce "InterpretNet", an innovative architecture that emulates the hypothesis-verification process observed in human perception. To tackle the limitations brought by "InterpretNet", we propose a second paradigm, which integrate the acquisition of generalizable knowledge into the framework of self-supervised representation learning. We provide an overview of these two paradigms in the following subsections.

4.1.1 The Hypothesis-Verification Process

The process of "hypothesis-verification" in human conscious perception is initially discussed by Marcel (1983) [157], and is often referred to as predictive coding in current research [158]. This process is a significant model in understanding cortical activity and perception. It implies that perception is a hierarchical process wherein the brain continuously updates and adapts its processing of sensory inputs based on predictions. These predictions are formulated from an internal model of the world and are constantly compared to incoming sensory information to minimize "prediction errors" — the difference between the prediction and the actual observation [158], [159].

There is emerging evidence connecting the hypothesis-verification process with high-level symbolic manipulation in the human brain [160]. As we introduced in Chapter 1, it is suggested that the human "algebraic mind" could provide a generalizable model of the world, which offers a crucial computational foundation for the hypothesis-verification process. In the previous chapter, we have gained some insights into the acquisition of knowledge that exhibits a degree of generalizability. Consequently, the next question we investigate is whether applying such generalizable knowledge to real tasks can enhance the *o.o.d.* generalizability of deep learning models. To address this, we propose "InterpretNet", an innovative architecture that emulates the *hypothesis-verification* process in human perception, and apply it to the classification of hand-written digits.

The proposed InterpretNet, depicted in Figure 4.1, is composed of two distinct modules: an ESTIMATOR and an IDENTIFIER. These modules, trained offline separately, are equipped with generalizable knowledge of target mechanisms, including 2D transformations. With the acquired knowledge, InterpretNet is able to provide additional explainability when classifying images with covariate shifts. Specifically, InterpretNet's functionality extends beyond the basic classification questions such as "Is there a '5' in the image? " It is also capable to answer interpretative questions such as "Why do you think it is a '5'? "

However, two limitations of the explicit knowledge exploitation approach in InterpretNet have been identified. Firstly, a covariate shift is introduced in test set by in-

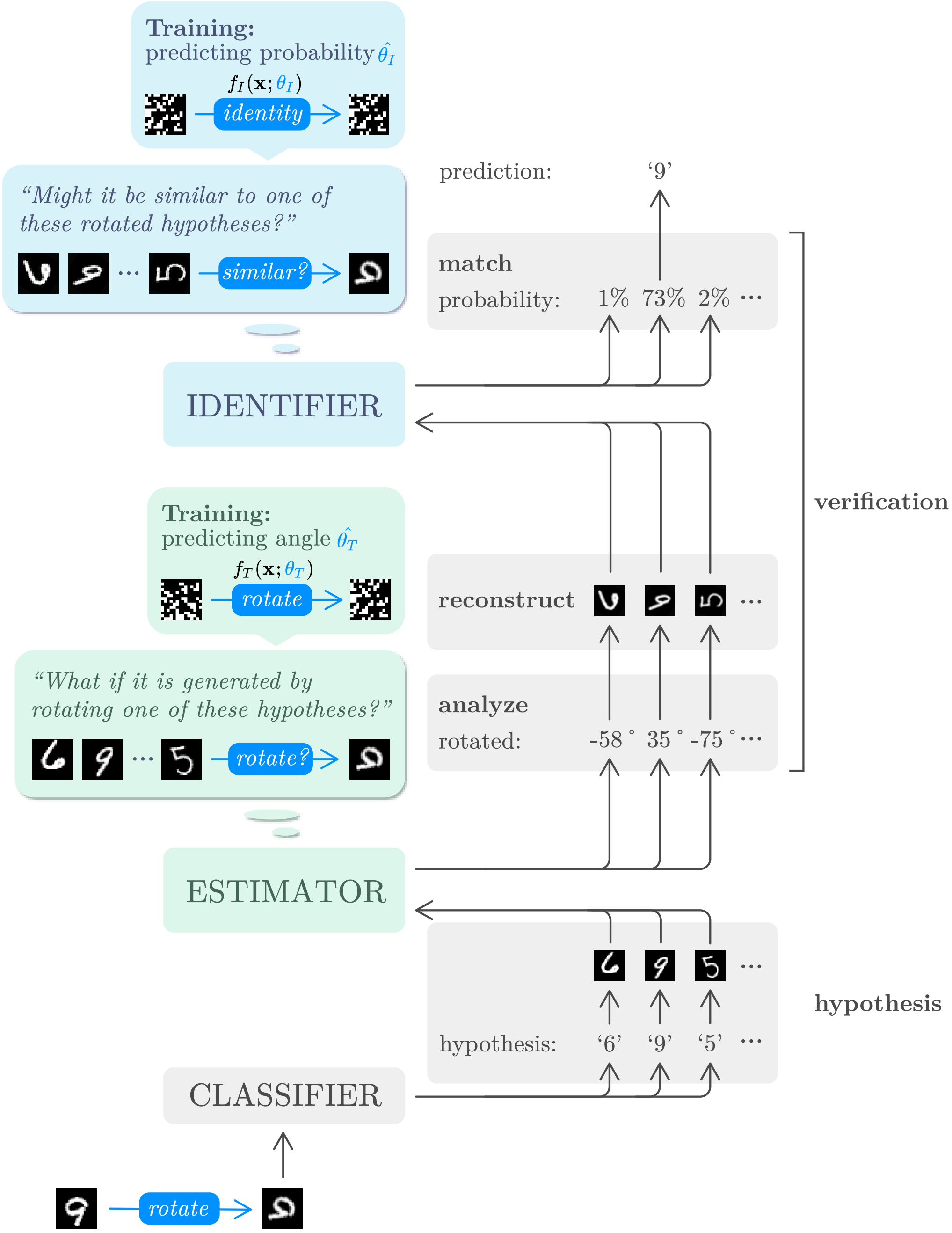

Training:
predicting probability $\hat{\theta}_I$
$f_I(\mathbf{x};\theta_I)$
identity
"Might it be similar to one of these rotated hypotheses?"
similar?
IDENTIFIER
Training:
predicting angle $\hat{\theta}_T$
$f_T(\mathbf{x};\theta_T)$
rotate
"What if it is generated by rotating one of these hypotheses?"
rotate?
ESTIMATOR
CLASSIFIER
rotate
prediction: '9'
match
probability: 1% 73% 2% ···
reconstruct ···
analyze
rotated: -58 ° 35 ° -75 ° ···
verification
hypothesis: '6' '9' '5' ···
hypothesis

Figure 4.1: The InterpretNet architecture. Potential classes are hypothesized by the CLASSIFIER C, and verification on these classes is made by the ESTIMATOR E and the IDENTIFIER I through the pipeline of (1) analyzing possible transformations, (2) reconstructing from candidates and (3) matching them with the sample.

tervening on a single mechanism (*i.e.* rotation), which is an over-simplified scenario. In real-world tasks, however, there always exists covariate shift caused by various mechanism simultaneously. It is still challenging for InterpretNet to leverage multiple ESTIMATORS and address this problem. Secondly, the hypothesis-verification process in InterpretNet utilizes a greedy algorithm, which requires a thorough comparison between the target sample and a large pool of candidate samples. This method is quite straightforward, but is time-intensive and computationally demanding. Moreover, the operations for reconstruction and comparison are conducted at the image level rather than the more efficient vector level, further escalating computational expenses. These two drawbacks limit its applicability in real-world scenarios.

4.1.2 Enhancing Real-World Applicability

In order to exploit the acquired generalizable knowledge more effectively in real-world tasks, we investigate more methodologies in this chapter.

Integration into Self-supervised Learning

From the findings of the previous chapter, we know that generalizable knowledge can be learned by conditioning on samples before and after transformations, which is similar to the paradigm of contrastive learning. The difference of the two lies in the target of the learning processes, while acquisition of generalizable knowledge focuses on *equivariance* learning, contrastive objectives are aiming at *invariance* learning. Thus, our hypothesis is that by incorporating transformation parameter estimation as a pretext task into self-supervised learning based on contrastive objectives, neural networks can *implicitly* learn generalizable knowledge through disentanglement and obtain enhanced representational capabilities.

Therefore, in this work, we introduce an innovative "regression loss" function specifically designed for transformation parameter estimation in self-supervised learning, as illustrated in Figure 4.2.

It is found in the results that this function is crucial for learning more descriptive representations, which facilitate the differentiation of individual samples by lever-

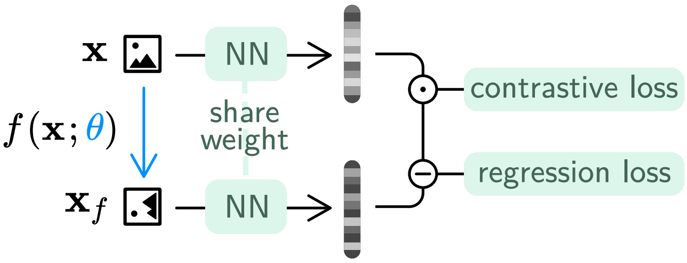


Figure 4.2: The integration of transformation parameter estimation task through regression loss into contrastive-based self-supervised learning.

aging knowledge about their intrinsic relationships. This approach offers a more feasible pathway for exploiting generalizable knowledge to real-world tasks.

Exploiting Knowledge in Pre-trained Models

The above findings have inspired us to take a further step in exploiting knowledge more effectively. In the process of self-supervised learning, we can further enhance the model's representational capability by increasing the number of transformation parameters to estimate. This increasing could presumably disentangle more concepts, and thus improve the generalizability of model representation by leveraging the knowledge about these concepts.

However, the recent advancements in multi-modal contrastive learning have provided us a novel pathway to acquire generalizable knowledge. Take the CLIP model [161] as an example, the model is trained with contrastive methodology based on image-text data pairs and has achieved impressive performance in downstream tasks, particularly demonstrating exceptional capability in zero-shot classification. Under this learning paradigm, each word in the textual descriptions can be considered as providing semantic grounding for the corresponding image content. If we regard certain content within the image as the concept we aim to learn, then the words in the description can be seen as parameters for the concept to show in the image. Consequently, this contrastive learning objective can be abstracted as to predict the probability of consistency between the transformation results and transformation parameters, as depicted in Figure 4.3.

Therefore, we hypothesize that models trained with this methodology can achieve better disentanglement and enhanced generalizability, thereby possessing superior

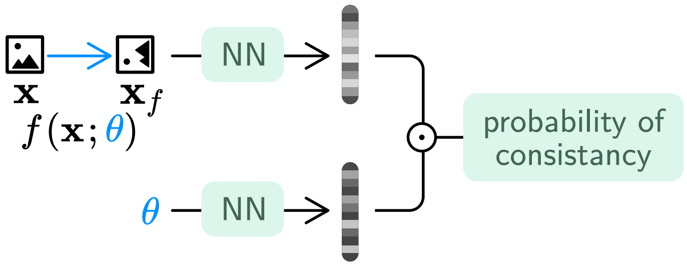


Figure 4.3: The abstraction of image-text contrastive learning. The purpose of learning is to predict the probability of consistency between the content in an image (result of transformation) and the textual description (transformation parameters).

representational capabilities. Based on this hypothesis, instead of training the model from scratch, we propose employing the pre-trained image-text models such as CLIP to transfer their knowledge to downstream tasks, potentially enhancing performance through established generalizability. This proposition gains verification from recent studies [162]–[164], which utilized the CLIP model to positive effect. Our experimental results also corroborate this hypothesis, revealing that the integration of pre-trained CLIP embeddings can significantly improve task performance even across another modality (*e.g.* point clouds).

The main contributions in this chapter are:

- We introduce a novel architecture "InterpretNet" to explicitly exploit learned knowledge in image classification. To the best of our knowledge, InterpretNet is the first work that emulates the human hypothesis-verification cognitive process and provides enhanced *o.o.d.* generalizability and extra explainability in hand-written digits classification.

- We devise a novel regression loss function for the integration of transformation parameter estimation into contrastive self-supervised learning. This approach implicitly enables neural networks to learn generalizable knowledge via disentanglement, leading to more descriptive and generalizable representations for downstream real-world tasks.

- We show that image-text contrastive learning enhances the acquisition of generalizable knowledge, as demonstrated by successful transfer learning applications employing pre-trained models.

In the following sections, we first describe methodologies for the explicit and implicit knowledge exploitation, respectively, in Section 4.2. Section 4.3 provides a detailed description of our experimental procedures and results. This is followed by in-depth discussions in Section 4.4. The chapter concludes in Section 4.5.

4.2 Methodology

4.2.1 InterpretNet

In the previous chapter, we have gained some insights regarding the acquisition of knowledge with a measurable degree of generalizability. Building on this understanding, this study introduces InterpretNet, comprising two deep neural network (DNN) modules: an ESTIMATOR E and an IDENTIFIER I. The effectiveness of exploiting generalizable knowledge is examined by evaluating the model's performance in image classification. This evaluation is particularly focused on how the model handles potential covariate shifts in the test set, which arise due to a specific target mechanism.

In our experimental setting, the target mechanism remains unaddressed by data augmentation techniques. This scenario is frequently encountered in real-world applications. To simulate this condition, we apply random 2D transformations (*e.g.*, rotation) to the MNIST test set, while deliberately avoiding any form of data augmentation during the training phase.

In the following subsections, we first propose the training methodology for modules E and I in Section 4.2.1. InterpretNet makes predictions in classification by raising hypotheses with a vanilla classifier and verifying them with E and I. The details about the architecture are described in Section 4.2.1.

Training

Dataset To train the E and I modules of InterpretNet, a synthetic dataset is created based on the methodology detailed in Section 3.3.1. This dataset comprises pairs of images, $\mathbf{x}$ and $\mathbf{x}_T$, representing the pre-transformation and post-transformation states, along with the associated transformation parameters $\boldsymbol{\theta}_T$).

InterpretNet leverages two types of generalizable knowledge in the image classification task. Firstly, it utilizes knowledge about 2D transformation for estimating transformation parameters. Secondly, it employs knowledge of identity matching to determine the identity of an image pair defined by an identity function f_I. An image $\mathbf{x}_T$ generated through 2D transformation f_T can be represented as:

$$\mathbf{x}_T = f_T(\mathbf{x}; \boldsymbol{\theta}_T). \tag{4.1}$$

In this study, transformation is implemented using affine transformation functions. For the identity function $f_I(\mathbf{x}; \boldsymbol{\theta}_I)$, when $\boldsymbol{\theta}_I = 1$, the function returns a same-identity but transformed image $\hat{\mathbf{x}_T}$, and any random sample other than $\hat{\mathbf{x}_T}$ when $\boldsymbol{\theta}_I = 0$. Concretely, the identity function is defined by:

$$\mathbf{x}_I = f_I(\mathbf{x}; \boldsymbol{\theta}_I) = \begin{cases} \hat{\mathbf{x}_T} & \text{if } \boldsymbol{\theta}_I = 1; \\ \hat{\mathbf{x}'_T} & \text{if } \boldsymbol{\theta}_I = 0, \end{cases} \tag{4.2}$$

where

$$\begin{aligned} \hat{\mathbf{x}_T} &= f_T(\mathbf{x}; \hat{\boldsymbol{\theta}_T}), \\ \hat{\mathbf{x}'_T} &= f_T(\mathbf{x}'; \hat{\boldsymbol{\theta}_T}). \end{aligned}$$

Here, $\mathbf{x}'$ is a random sample other than $\mathbf{x}$, and $\hat{\boldsymbol{\theta}_T}$ is the 2D transformation parameter estimated by the ESTIMATOR E (see Sections 4.2.1 and 4.2.1 for the details).

Learning Objectives Based on the above synthetic dataset, the ESTIMATOR E and the IDENTIFIER I are trained to learn knowledge of 2D transformation f_T and the identity function f_I, respectively. Specifically, we employ E which takes as the input paired images $\mathbf{x}$ and $\mathbf{x}_T$ generated from f_T, to predict the transformation parameters $\hat{\boldsymbol{\theta}_T}$. The role of I, on the other hand, is to learn from f_I and to predict the probability that a pair of images are of the same identity. Note that, in practice, the inputs of I are $\mathbf{x}_T$ and $\mathbf{x}_I$ (instead of $\mathbf{x}$ and $\mathbf{x}_I$).

The mechanism of f_T is independent of f_I, and thus E is optimized first, by minimizing the mean squared error loss L_{MSE} on $\boldsymbol{\theta}_T$. I is then trained based on datasets generated with f_I and E, and optimized by minimizing the binary cross entropy loss L_{BCE} on $\boldsymbol{\theta}_I$. Therefore, the objectives of knowledge learning in this study can be represented as:

$$\arg\min_{E} L_{MSE}(E(\mathbf{x}, f_T(\mathbf{x}; \boldsymbol{\theta}_T)), \boldsymbol{\theta}_T), \tag{4.3}$$

$$\arg\min_{I} L_{BCE}(I(f_T(\mathbf{x}; \boldsymbol{\theta}_T), f_I(\mathbf{x}; \boldsymbol{\theta}_I)), \boldsymbol{\theta}_I). \tag{4.4}$$

Architecture

Subsequently, we illustrate the exploitation of acquired knowledge through InterpretNet in image classification tasks. InterpretNet is composed of two DNN modules: an ESTIMATOR (E) and an IDENTIFIER (I). The architecture functions by integrating hypotheses generated by a basic CLASSIFIER (C) and verifying these hypotheses using the E and I modules (Figure 4.1). This section delves into a detailed description of each module and their respective contributions to emulating the human cognitive process of hypothesis and verification.

Classifier C To establish an out-of-distribution (*o.o.d.*) task, images from the MNIST test set are transformed prior to testing, indicated as X_T^{test}. In contrast, images in the training set, X^{train}, remain unaltered. For a given test sample $\mathbf{x}_T^{test} \in X_T^{test}$, the basic CLASSIFIER C generates a probability distribution reflecting the likelihood of the sample belonging to each class. This distribution is utilized to derive confidence scores. If the highest confidence score among all classes falls below a predetermined threshold, C refrains from making an immediate classification. Instead, it proposes a hypothesis $H(\mathbf{x}_T^{test}) = \{y_i\}_{i=1}^k$, comprising a list of the top k class labels based on confidence scores, subjected to further verification.

Estimator E The ESTIMATOR E samples $N(N \geqslant 1)$ random candidates from X^{train} for each class identified in the hypothesis $H(\mathbf{x}_T^{test})$. The set of all candidates corresponding to the test sample $\mathbf{x}_T^{test}$ is represented as $X_c \subset X^{train}$, where $X_c = \{X_c^{(y_i)}\}_{i=1}^k$, and each $X_c^{(y_i)}$ consists of N candidates: $\{\mathbf{x}^{(y_i),j}\}_{j=1}^N$. Under the presumption that $\mathbf{x}_T^{test}$ could be a transformed version of any candidate in X_c, E examines the relationship between $\mathbf{x}_T^{test}$ and each candidate *w.r.t.* the 2D transformation. This analysis is based on the previously acquired knowledge. For each candidate, E calculates $\widehat{\boldsymbol{\theta}_T^{i,j}} = E(\mathbf{x}^{(y_i),j}, \mathbf{x}_T^{test})$, which represents the estimated transformation parameters between the test sample and the candidate.

Identifier I Given that E is a deterministic function and will yield an output irrespective of the actual relationship between two images, the IDENTIFIER I is tasked with determining which candidate exhibits the greatest similarity to $\mathbf{x}_T^{test}$. This process begins with reconstructing each candidate using the parameters $\widehat{\boldsymbol{\theta}_T^{i,j}}$ predicted by E. The reconstruction produces $\widehat{\mathbf{x}_T^{(y_i),j}} = f_T(\mathbf{x}^{(y_i),j}; \widehat{\boldsymbol{\theta}_T^{i,j}})$. Subsequently, $\widehat{\mathbf{x}_T^{(y_i),j}}$ is evaluated by I to assess its likelihood of matching $\mathbf{x}_T^{test}$. This evaluation is based on $I(\mathbf{x}_T^{test}, \widehat{\mathbf{x}_T^{(y_i),j}})$, which is trained using the identity function f_I. The label of the candidate with the highest likelihood score is then selected as the final prediction, represented by $\hat{y} = \arg\max_{y_i} I(\mathbf{x}_T^{test}, \widehat{\mathbf{x}_T^{(y_i),j}})$.

In the described methodology, potential classes are initially proposed by the CLASSIFIER C. Subsequently, the modules E and I verify these classes through a sequence of steps: (a) analyzing potential transformations, (b) reconstructing images from the candidate pool, and (c) assessing the match between these reconstructions and the test sample.

It is noteworthy that the pre-trained modules E and I are designed to operate independently of the MNIST training data and do not necessarily depend on the CLASSIFIER C for their functionality. Given that the training and test sets of MNIST share the same class label space, we also investigate a classification approach devoid of the basic CLASSIFIER C, referred to as "InterpretNet_noC". The primary distinction with InterpretNet_noC is its approach to hypotheses generation: it considers all classes by default ($k = 10$), thus eliminating the initial class prediction step by the CLASSIFIER C.

4.2.2 Methods Based on Self-supervised Learning

In the following subsections, we first describe the design of the "regression loss" function and its integration into the framework of contrastive self-supervised learning in Section "Regression Loss". Section "Knowledge Transfer Learning" illustrates the learning methodology employed for transferring knowledge from image-based models to point cloud understanding tasks.

Regression Loss

The regression loss function is designed for acquisition of transformation knowledge through predicting transformation parameters. To facilitate this, we have converted the task from a numerical regression problem into a logistic regression one. This conversion involves a redefinition of how we handle parameter values. The parameter data range $[\theta_a, \theta_b)$ is first segmented into C equal parts. Each of these segments represents a distinct category within the total C categories. Then, instead of predicting a specific value for the parameters, our approach assigns each parameter to one of these categories. This categorization process results in the target y_i being classified within the discrete range of $[0, C)$.

In order to effectively integrate the regression loss into the framework of contrastive self-supervised learning, we adopt the Siamese Network, as depicted in Figure 3.2(**b**) and Figure 4.2. The input to the Siamese Network consists of a batch of N data pairs, represented as $\{\mathbf{x}_i, \mathbf{x'}_i\}_{i=1}^N$. Each data point comprises two elements: $\mathbf{x}_i$ and $\mathbf{x'}_i$, which correspond to the pre-transformation and post-transformation states, respectively. The network's output hidden representations are denoted as $\{\mathbf{h}_i, \mathbf{h'}_i\}_{i=1}^N$.

Subsequently, we calculate the difference between $\mathbf{h}_i$ and $\mathbf{h'}_i$, and project the resultant difference vector into a $\mathbb{R}^C$ space using a linear projection f_{FC}, *i.e.*,

$$\hat{y_i} = norm[f_{FC}(\mathbf{h}_i - \mathbf{h'}_i)] \tag{4.5}$$

where *norm* signifies the normalization operation. Consequently, the regression loss $\mathcal{L}_{reg}$ can be expressed as:

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{i=1}^{N} (1 - \hat{y}_{i,y_i}) \tag{4.6}$$

For the contrastive objective, we adopt a widely-used loss function, as outlined in [161]. This function is designed to distinguish positive pairs from a total of N^2 potential pairs in a batch of size N. The loss function is represented as:

$$\mathcal{L}_i^{O2T} = -\log \frac{\exp(\mathbf{h}_i{}^\top \mathbf{h'}_i/\tau)}{\sum_{j=1}^N \exp(\mathbf{h}_i{}^\top \mathbf{h'}_j/\tau)} \tag{4.7}$$

$$\mathcal{L}_{ctr} = \frac{1}{2N} \sum_{i=1}^{N} (\, \mathcal{L}_i^{O2T} + \mathcal{L}_i^{T2O} \,) \tag{4.8}$$

where τ stands for the temperature co-efficient, and the superscripts $O2T$ and $T2O$ denote the matching of original-to-transformed and transformed-to-original sample pairs, respectively. The formula $\mathcal{L}_i^{O2T}$ represents the loss for discerning the correct transformed match for an original sample, while $\mathcal{L}_i^{T2O}$ accounts for identifying the original sample for a transformed one. The overall contrastive loss $\mathcal{L}_{ctr}$ is then calculated as the average of these losses across all samples in the batch.

Finally, the overall objective of the regression-loss-integrated self-supervised learning is to optimize the model with the combination of the above two losses:

$$\mathcal{L} = \lambda_{ctr}\mathcal{L}_{ctr} + \lambda_{reg}\mathcal{L}_{reg} \tag{4.9}$$

Here, λ_{ctr} and λ_{reg} represent the coefficients that adjust the influence of $\mathcal{L}_{ctr}$ and $\mathcal{L}_{reg}$, respectively. Both coefficients are set to 1.0 in our experiments to ensure an equal contribution from each loss.

Intuitively, this approach minimize the similarity across distinct semantic categories on a large scale through the regularization of contrastive loss, while concurrently preserving a microstructure within each semantic category by employing the equal-variance principle governed by the regression loss, as depicted in Figure 4.4.

Knowledge Transfer Learning

In this study, our objective is to transfer generalizable knowledge from the image encoder of a large-scale image-text model to a point cloud encoder. This process is achieved through a self-supervised learning approach, as illustrated in Figure 4.5. For the dataset, we construct each mini-batch to contain N pairs of point clouds and images, with each pair representing the same object. The point and image data are processed through their respective encoders, which yield output representations denoted as $\{\mathbf{h}_i^{\mathcal{P}}, \mathbf{h}_i^{\mathcal{I}}\}_{i=1}^N$, respectively. Contrastive loss is conducted to maximize the cosine similarity between the corresponding output representations:

$$\mathcal{L}_i^{\mathcal{P}2\mathcal{I}} = -\log \frac{\exp(\mathbf{h}_i^{\mathcal{P}\,\top}\mathbf{h}_i^{\mathcal{I}}/\tau)}{\sum_{j=1}^N \exp(\mathbf{h}_i^{\mathcal{P}\,\top}\mathbf{h}_j^{\mathcal{I}}/\tau)} \tag{4.10}$$

$$\mathcal{L}_{ctr} = \frac{1}{2N}\sum_{i=1}^{N}(\,\mathcal{L}_i^{\mathcal{P}2\mathcal{I}} + \mathcal{L}_i^{\mathcal{I}2\mathcal{P}}\,) \tag{4.11}$$

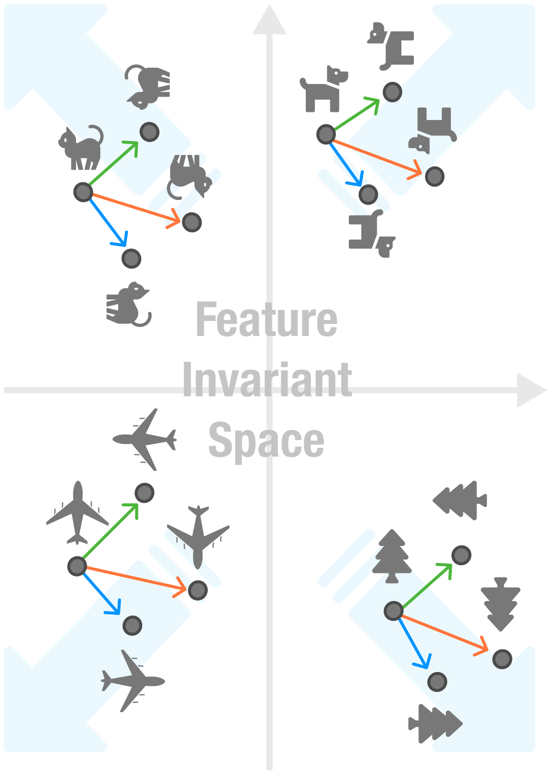


Figure 4.4: The intuition behind the integration of regression loss into the contrastive objective.

where the superscript $\mathcal{P}2\mathcal{I}$ and $\mathcal{I}2\mathcal{P}$ indicate the point-to-image and image-to-point sample pair matching, respectively. A key aspect of the training methodology is that only the parameters belonging to the point cloud encoder are updated via back-propagation, while the parameters of the pre-trained image encoder are kept frozen.

4.3 Experiments

This section begins with an analysis of the InterpretNet's performance in image classification under covariate shift, as detailed in Section 4.3.1. Subsequently, the impact of implicit knowledge exploitation on the model's performance in point cloud classification is explored in Section 4.3.2.

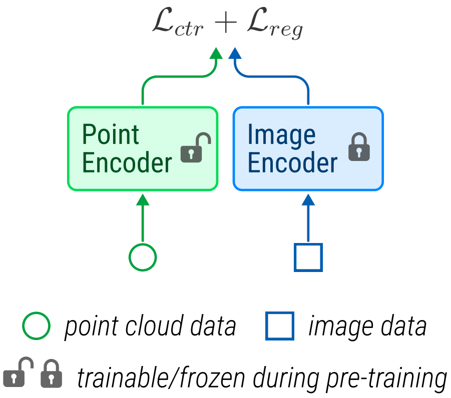

$\mathcal{L}_{ctr} + \mathcal{L}_{reg}$
Point Encoder
Image Encoder
point cloud data
image data
trainable/frozen during pre-training

Figure 4.5: The architecture for knowledge transfer learning.

4.3.1 InterpretNet

Model Training

To construct InterpretNet, the modules E and I are instantiated using the CNN_pair models, which is described in Section 3.3.2. Specifically, the CNN_pair models consist of four sequential CNN blocks. Each block is composed of three convolutional layers, featuring a range of 32 to 96 channels. Following each block, there is a 3×3 max pooling layer. The output head is a fully connected layer, which outputs the estimated transformation parameter and the probability of identity matching, for modules E and I, respectively. The detailed architecture is listed in Table 3.3

A black-white noise dataset is created based on the methodology used for Exp_NOISE detailed in Section 3.3.1 of Chapter 3. Specifically, the input dimension is $N_{batch} \times 2 \times 28 \times 28$, where N_{batch} is the batch size. The black/white ratio for the noise images is set to $7:3$. 2D rotation is chosen as the covariate, which is applied to the dataset and leads to the $o.o.d.$ data scenario. The rotation angle is randomly sampled in a uniform distribution in the range of $[-90°, 90°]$.

The modules E and I are trained using Adam optimizer [154] with a batch size of 512 and the weight decay set to 5.0×10^{-4}. The models are trained for 1.0×10^5 steps. The initial learning rate is set to 0.03 with a decaying factor of 0.5, and a decaying cadence of 1.0×10^4 steps.

Classification Performance

We follow common practices in the computer vision community [165], [166], by conducting the investigation with the most popular and fundamental MNIST dataset [167]. The CLASSIFIER C is trained with original samples X^{train} in MNIST without any data augmentations. The length k of hypothesis $H(\mathbf{x}_T^{test})$ is set to 5 and 10. The number of candidates N for E is set to 200 for each class. The confidence threshold of C is set to 0.9999.

The classification accuracy obtained on the MNIST test set, with or without rotations, is shown in Figure 4.6. The first observation is that, in the case of rotated test set, the basic classifier has experienced nearly a 40% performance drop. However, the accuracy of InterpretNet has increased to 77% when $k = 5$ (InterpretNet_5) and even further to 82% when $k = 10$ (InterpretNet_10), with a minimal impact on performance for test sets with no rotation applied. In InterpretNet, E and I are introduced for further interpretation when C is not very confident in its prediction. They provide extra explanations about why the sample is classified as such and how it is rotated, by leveraging the knowledge of rotation with E. Specifically, when posed with a question such as "What number is in the image, and why do you think it is that number?", InterpretNet might respond, "It appears to be a '5', primarily *because* it looks *similar* to a reference image of a '5' (from the candidate pool), but *rotated* by 24 degrees." In this scenario, the first half of the answer is predicted by C, whereas the latter half, the explanation of identity and transformation, is generated through modules E and I.

Secondly, InterpretNet_noC is studied by removing C from InterpretNet. Because of the absence of C and thus the length of label space is unknown, the value of k is set to 10. It is found that InterpretNet_noC outperforms the basic classifier by +13%, with a classification accuracy of 75% (in Figure 4.6). It is worth noting that the performance is achieved without *any* knowledge of the handwritten digits (since both E and I are trained in Exp_NOISE), but only through the processes of analyzing, reconstructing and matching. Furthermore, only 4% ($200 \times 10/50000$) of the training data are accessed during inference. This result indicates that InterpretNet_noC is capable of classifying characters, even those it has never encountered or had any

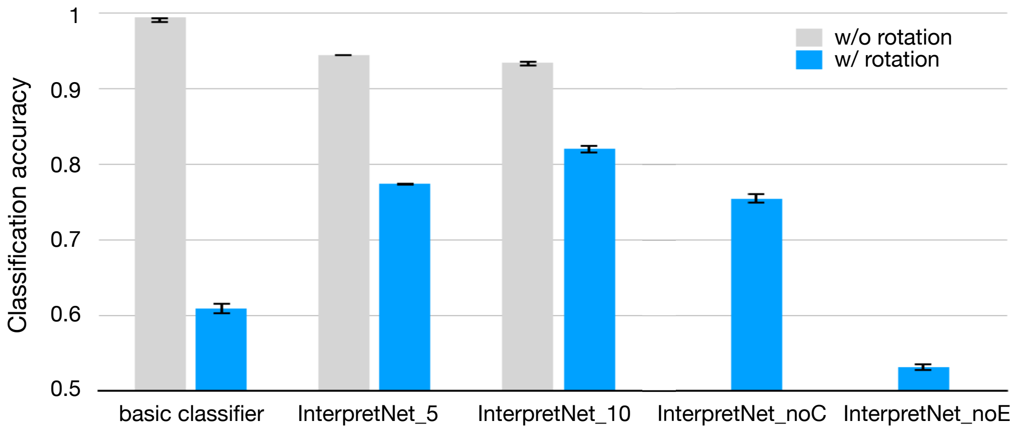


Figure 4.6: The performance of classification. InterpretNet_5 and InterpretNet_10 denote InterpretNet with hypothesis $k = 5$ and $k = 10$, respectively.

prior knowledge of, given the condition that some necessary references are provided. This capability is behaviorally similar to that of human beings.

To investigate the role of E with its knowledge about rotation, an ablation study was conducted on InterpretNet_noE by removing E from the InterpretNet. As shown in Figure 4.6, the InterpretNet_noE loses the ability to interpret rotation information and the performance on recognising rotated test set has dropped from 82% to less than 60%. On the one hand, this indicates the importance of rotation knowledge to I, which requires instructions for reconstruction. On the other hand, since the rotated samples look very different from the candidates, it also indirectly demonstrates the effectiveness of I.

Number of Candidates. As shown in Figure 4.7, classification accuracy is greatly affected by the number of candidates. Given that I is trained on noise, the module is really sensitive to subtle differences. Therefore, in order to find a candidate that is very similar to a sample, a candidate pool of a proper size is required. In addition, the generation of digits can also be viewed as a mechanism. Unlike 2D transformations, the parameterization of digit generation is much more complicated [168]. While the integration of an estimation module for digit generation (as a new E) into the existing InterpretNet would presumably reduce the required number of candidates significantly, this will, at the same time, introduce new challenges in

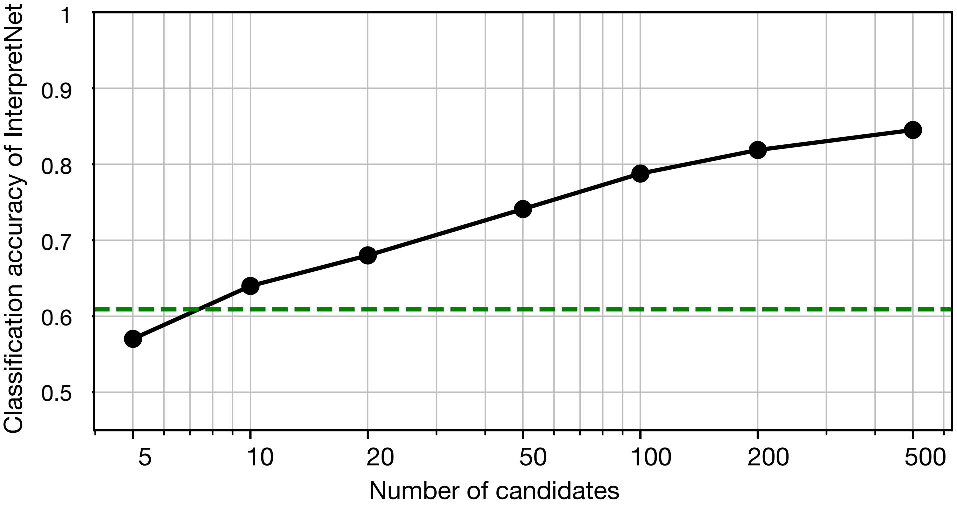


Figure 4.7: The classification accuracy of InterpretNet with different numbers of candidates. Performance exceeds the basic classifier (the green dash line) when $N \geqslant 10$.

compositionality, which involves the collaboration between multiple Es.

4.3.2 Regression Loss and Transfer Learning

In this section, we aim to further validate that our methodology facilitates the acquisition of generalizable knowledge, which enhances the model's performance, across broader domains. To strengthen the reliability of this validation, we have chosen a modality that is distinct yet related to 2D images: 3D point clouds, and we also utilize model architectures other than CNNs. To apply the acquired knowledge in real-world tasks, we use the methodology outlined in Section 4.2.2 for its implicit exploitation. The specifics of the training process are detailed in Section "Training Setup", followed by analysis of the experimental results in Section "Point Clouds Classification".

Training Setup

In order to ensure alignment with existing work and facilitate fair comparisons, the training setup and protocol in this study aligns with the framework established in CrossPoint [169]. The loss functions for cross-modal and inter-modal instance discrimination, as detailed in CrossPoint [169], are adopted as the foundational contrastive learning objectives. DGCNN [170] and CNN are employed as the default encoders for point clouds and images, respectively, unless stated otherwise.

The following variations are introduced to compare with the baseline established by CrossPoint:

- **Reg**: This variant incorporates the regression loss into the contrastive framework, as described in Section 4.2.2 “Regression Loss”. The objective is to assess the impact of transformation parameter estimation tasks on the model’s representational capabilities.
- **Reg+CLIP**: In addition to the integration of regression loss, this variation substitutes the CNN image encoder with a pre-trained CLIP. This experiment aims to transfer knowledge from large-scale text-image contrastive learning models, and evaluate the effect of the transferred knowledge on the representational capabilities of point cloud encoder. As described in Section 4.2.2 “Knowledge Transfer Learning”, the parameters of CLIP are kept frozen during training.
- **ViT**: This variant replaces both the point and image encoders with a ViT-based [171] architecture PCExpert [172], thereby broadening the spectrum of models under examination. The image tower utilizes the CLIP, with its parameters remaining fixed throughout the training phase.

Point Cloud Classification

In evaluating model performance in point cloud classification, we freeze parameters of the point cloud encoder after pretraining, and fit and test a linear SVM classifier on two point cloud datasets: ModelNet40 [173] and ScanObjectNN [174]. In line with standard procedures, we sample $1,024$ points from each instance in ModelNet40 for SVM fitting and testing, and the results are denoted by ModelNet40-1k. All classification results are reported in terms of overall accuracy (OA), unless specified otherwise.

The classification results on ModelNet40 and ScanObjectNN are reported in Table 4.1. Firstly, it is shown that the introduction of our proposed regression loss

Table 4.1: Effect of the regression loss (Reg) and knowledge transferred from CLIP (Reg+CLIP) on classification accuracy on ModelNet40 and ScanObjectNN.

Method	ModelNet40-1k	ScanObjectNN
Multi-Task [175]	89.1	-
Self-Contrast [176]	89.6	-
Jigsaw [177]	90.6	59.5
STRL [178]	90.9	77.9
Rotation [179]	90.8	-
OcCo [174]	89.2	78.3
CrossPoint [169]	91.2	81.7
Reg (Ours)	**91.4** ($\uparrow$ 0.2)	**84.0** ($\uparrow$ 2.3)
Reg+CLIP (Ours)	**91.7** ($\uparrow$ 0.5)	**87.8** ($\uparrow$ 6.1)

consistently enhances the model performance over CrossPoint and other methods. On ModelNet40-1k, there's a marginal increase of 0.2%, while on ScanObjectNN, the improvement is more substantial at 2.3%. This result suggests that the supervisory signal provided by the regression loss can further improve the model's ability to discriminate between instances that are intrinsically related by transformations. This improved ability is implicitly integrated into the model's representational capability through self-supervised representation learning, and is reflected in the model's enhanced generalizability when applied on novel datasets.

Secondly, it can be noticed that the addition of knowledge transferred from CLIP (Reg+CLIP) leads to even more pronounced improvements. For ModelNet40-1k, the increase is 0.5% over the baseline CrossPoint, while on ScanObjectNN, there is a significant leap of 6.1%. These results suggest that incorporating knowledge from a large-scale text-image contrastive learning model like CLIP substantially enhances the model's ability to generalize and accurately classify point clouds, especially in the more complex and challenging ScanObjectNN dataset. This result validates our hypothesis to some extent that text-image contrastive learning fundamentally involves the acquisition of generalizable knowledge. Even if the knowledge is transferred from image models and applied to point cloud tasks, it can still enhance the

Table 4.2: Comparison between two loss functions for the regression loss.

Method	ModelNet40-1k	ScanObjectNN
Baseline (CrossPoint [169])	91.2	81.7
CrossEntropy (Ours)	90.8	83.1
Cosine (Ours)	91.4	84.0

model's generalization performance.

Loss Functions In the study, we explore various loss functions for implementing the regression loss. Along with the method described in Equation 4.6, we examine another approach using cross-entropy as the loss function, represented as:

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{i=1}^{N} - \log \frac{\exp\left(\hat{y}_{i,y_i}\right)}{\sum_{c=1}^{C} \exp\left(\hat{y}_{i,c}\right)} \tag{4.12}$$

where C denotes the number of categories. This approach essentially redefines the parameter estimation task as a classification problem. The effect of these two loss functions on model performance is shown in Table 4.2. The results reveal a marginal decrease in performance with the CrosssEntropy loss compared to the Cosine loss. This suggests that the Cosine loss function aligns better with the intrinsic properties of the model's learning process for transformation parameter estimation task, and thereby it is more effective in enhancing the model's representational capabilities.

Model Architecture We investigate the effect of the regression loss on different model architectures, *i.e.*, DGCNN [170] and ViT [171], with the results shown in Table 4.3. It can be observed that incorporating the regression loss consistently improves model performance across both architectures and datasets, which further validate the effectiveness of our methodology for generalizable knowledge learning. Furthermore, in the case of the ViT model, the introduction of the regression loss yields a even more substantial improvement on ScanObjectNN, with an accuracy increase of 7.0%. This significant increase could imply that ViT, being a more complex and capable architecture, is better able to leverage the additional information provided by the regression loss. We will investigate the reasons behind this findings further in Chapter 5 and explore further the potential of transformer-based models in point cloud understanding tasks.

Table 4.3: The effect of the regression loss incorporation on different model architecture.

Backbone	Benchmark	w/o $\mathcal{L}_{reg}$	w/ $\mathcal{L}_{reg}$
DGCNN [170]	ModelNet40-1k	91.2 [169]	**91.4** (↑ 0.2)
	ScanObjectNN	81.7 [169]	**84.0** (↑ 2.3)
ViT [171]	ModelNet40-1k	91.8	**92.7** (↑ 0.9)
	ScanObjectNN	83.0	**90.0** (↑ 7.0)

4.4 Discussion

In this section, we delve into several relevant questions that emerge from the content of this chapter. Our discussion will first address the similarities and distinctions between InterpretNet and human visual perception in Section 4.4.1. We then review related work on parameter estimation, and discuss the distinction and innovation of our regression loss. Finally, we examine the relationship between human language and visual perception, by presenting hypotheses about the impact of language on knowledge learning and its correlation with our experimental findings, in Section 4.4.3. These examinations are aimed at providing insights about human cognitive behavior that could inspire and contribute to the advancement of artificial general intelligence in future research.

4.4.1 Simulation of Human's Visual Perception

In this work, we propose InterpretNet as an exploratory simulation of a human hypothesis-verification process in visual perception. Although the simulation is not reverse engineering of the human brain, based on psychological studies about cognition and behaviors, both humans and InterpretNet share similarities in how information is processed.

To elucidate the hypothesis-verification process more clearly, let us consider Figure 4.8, and examine the interpretations derived from Figure 4.8 (a) [180]. Observation of the same figure may yield at least two distinct interpretations, as depicted in Figs. 4.8(b) and (c). This simple example demonstrates a typical human perceptual process, where causal inference (in the anti-causal direction) is performed by

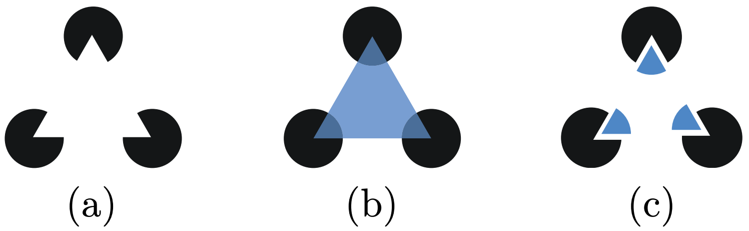


Figure 4.8: What is in image (a)? There are at least two ways to interpret it, *i.e.*, (b) three black circles partly *covered* by a white triangle, or (c) three black circles with a *notch* on each of them. (The former interpretation may have a stronger tendency in perception, according to the Gestalt principles [181].)

exploiting the knowledge of concepts like occlusion or notching upon variables of circles and/or triangles.

Specifically, this cognitive process involves formulating a hypothesis about the cause of the image — in this case, three circles being *occluded* by a triangle — and verifying the hypothesis through a simulation that reconstructs the process. Alternatively, if we propose a different hypothesis, such as three circles each having a *notch*, the image can still "make sense" with verification made by simulating the *notching* process.

Our ability to "make sense" of an object through simulating using mechanisms happens not only in visual perception, but also in other aspects of behaviors [182], [183], where individuals attempt to rationalize or explain their behaviors with convincing (but sometimes incorrect) reasons. In the context of InterpretNet, the role of E and I is actually to provide explanations. This functionality enables machines, to some extent, to "make sense" of the visual data they process.

Furthermore, the simulation and imagination in brains have been studied in various works, and are proposed as the key elements in the understanding of physical scenes and counterfactual reasoning [19], [184]. Based on the model of the world in their minds, humans can make predictions about the future (in a causal direction) and infer the causes of things that have happened (in an anti-causal direction). In the architecture of InterpretNet, the module E and the affine transformation functions enable the simulations of 2D transformations in anti-causal and causal directions, respectively. This architecture effectively equips the system with an imagination space.

Limitations of InterpretNet

Despite the advancements and capabilities demonstrated by InterpretNet, this section offers insights into the challenges and constraints that currently exist in the architecture.

Firstly, a covariate shift is introduced in test set by intervening on a single mechanism (*i.e.* rotation), which is an over-simplified scenario. In real-world tasks, there always exists covariate shift caused by various mechanism simultaneously. Ideally, in these scenarios, it would be beneficial if multiple Es could by utilized leveraging the knowledge learned separately and cooperate with each other. However, preliminary findings suggest that while Es exhibit good generalizability on the mechanisms they are specifically trained for, they struggle to generalize across other mechanisms. This is in line with [155], where the generalization improves only if more combinations of mechanisms (category and pose) are exposed during training. Therefore, addressing covariate shift caused by multiple mechanisms could involve either a stochastic training strategy or an intricate architectural design to handle interactions between modules (especially Es). This significantly increases the training cost and/or architectural complexity, and poses a substantial challenge in effectively leveraging the acquired knowledge.

Secondly, the hypothesis-verification process in InterpretNet utilizes a greedy algorithm, which, while straightforward, is proved to be time-intensive and computationally demanding. The method requires a thorough comparison between the target sample and a large pool of candidate samples, leading to significant consumption of computational resources and time. Moreover, the operations for reconstruction and comparison are conducted at the image level rather than the more efficient vector level, further escalating computational expenses. Such inefficiencies make the approach less suitable for real-time or large-scale applications where computational efficiency are crucial.

By identifying these limitations, we can better set realistic expectations for its performance, as well as lay the groundwork for future enhancements and research directions.

4.4.2 Related Work on Parameter Estimation

Parameter estimation has been extensively studied in existing research, with various studies employing supervised learning to predict parameter values [129]–[133], while others integrate parameter estimation with self-supervised learning for the enhancement of representation learning [134]–[136]. This research generally aligns with the latter approach, but with notable innovations.

Firstly, most existing studies employ single-image datasets for parameter estimation. According to causal theory, these approaches are restricted to object categories present in training datasets and are not able to generalize to unseen categories. In other words, these models can not acquire generalizable knowledge about relevant transformations. In contrast, this study aims to develop a parameter estimation capability with a level of generalizability regardless of categorical differences, akin to human cognition. This is achieved through the utilization of image pairs, and removal of confounders using random treatment. Furthermore, this work propose regression loss for parameter estimation. Instead of striving for precise parameter value prediction via MSE loss [134], we aim to maintain a microstructure within each semantic category in the representation space, by employing the equalvariance principle regulated under the regression loss. Therefore, this study adopts the idea of integrating parameter estimation into SSL, but it is distinctive and innovative in methodology.

4.4.3 Language and Mechanisms

As previously introduced in Chapter 1, humans demonstrate remarkable *o.o.d.* generalization capabilities through the “algebraic mind”, as conceptualized by Marcus (2003) [23]. This concept refers to the human ability to manipulate symbolic variables across various domains, enabling them to re-apply previously acquired concepts to novel scenarios [18]–[21].

Recalling the visual perception example depicted in Figure 4.8, we can more clearly illustrate the aspect of "algebraic mind" in cognitive process. During the process of verifying the hypothesis that "three circles are *occluded* by a triangle", we simulate the reconstruction process, as if we manipulate variables in a function, which can be algebraically modeled as:

```
figure = occlude(
                circle(3, black, [positional args]),
                triangle(1, white, [positional args])
               )
```

Alternatively, for the hypothesis of "three circles with a *notch* on each of them", the verification is made by simulating the `make_notch()` function:

```
figure = make_notch(
                  circle(3, black, [positional args]),
                  [positional args]
                 )
```

This showcases how cognitive processes are analogous to algebraic operations. It can be noticed that mechanisms in image generation processes (*e.g.* occlusion or notching) play a crucial role in human visual perception. Our interpretation of images is based on our understanding of these mechanisms, which is fundamentally different from the current deep learning models that largely rely on recognition of patterns from past visual experiences. It is also noteworthy that our knowledge of these mechanisms, like occlusion and notching, is systematic [22], and independent of the domain of the involved variables.

If we take a step further, it becomes evident that human language also shares significant similarities with algebraic operations. Human language can be considered as a multi-level nested and recursive system of symbol manipulation [185]. We can effortlessly convert the above algebraic operations into linguistic expressions, as exemplified below:

```
The figure shows (
    3 (
        black (
            circles (
                [evenly spaced]
                )))
    that are occluded (
        by a (
            white (
                tringle (
                    [in the center]
                    ))))).
```

When analyzing sentences using the above multi-layer nested structure, it is apparent that many words, in addition to verbs and nouns, can act like algebraic operations, manipulating variables. This observation leads to a bold hypothesis that potentially each word in a sentence could function in this manner, which opens up a new perspective to view the functionality of words. Considering each word as a function or mechanism also leads to the connection between the term of generalizability in deep learning and the term of compositionality [186] in Natural Language Processing (NLP). In computer vision, treating words as mechanisms allows us to consider text-image contrastive learning fundamentally the same as the learning of generalizable knowledge. This conceptual connection justifies the use of a pre-trained CLIP model for knowledge transfer in this study, which is validated by the findings detailed in Section4.3.2 demonstrating the impressive generalizability of CLIP.

4.5 Conclusion

In conclusion, this chapter has effectively explored two paradigms for exploiting generalizable knowledge in image classification tasks: the explicit and implicit exploitation. Our novel architecture, InterpretNet, which emulate human perception based on the process of hypothesis-verification, demonstrates that explicit knowledge exploitation can significantly enhance o.o.d. performance in hand-written digit classification. However, the limitations of InterpretNet suggest a need for alternative methods. This leads to our exploration of implicit knowledge exploitation, particularly through integration of the regression loss into self-supervised learning and leveraging pre-trained models like CLIP. Our results indicate that such methodologies not only improve the representational capabilities of neural networks, but also offer a more efficient way to exploit generalizable knowledge in diverse and complex real-world scenarios.

Chapter 5

3D Classification with 2D Generalizable Knowledge Transfer

In the preceding chapter, the experimental results demonstrate the effectiveness of regression loss in enhancing the generalizability of models, within a self-supervised learning paradigm. Furthermore, the discussion on the relationship of language and transformation mechanisms leads to the hypothesis that image-text contrastive learning, as a variant of regression loss-based self-supervised learning, can facilitate the acquisition of more generalizable knowledge. Consequently, a primary objective of this chapter is to empirically validate this hypothesis. This involves exploiting generalizable knowledge in a pre-trained image-text model (*e.g.*, CLIP [161]), and applying it via transfer learning to tasks in a different data domain. Moreover, this chapter also aims to integrate our understandings about the learning and exploiting generalizable knowledge from previous chapters, and apply these insights to challenging real-world tasks. To this end, we utilize regression loss in our proposed PCExpert architecture, thereby establishing a new benchmark across various 3D understanding tasks.

5.1 Introduction

To assess the generalizability of the image-text model, we have selected tasks related to 3D understanding as our focal point. Essentially, this involves leveraging the knowledge derived from 2D representations of objects to infer their characteristics in a 3D context. Such tasks significantly challenge the model's capacity to generalize across different levels of concepts. For the representative of 3D data, we have employed point cloud, which utilizes coordinates and various attributes to represent objects in three-dimensional space. This data format has demonstrated significant potential in deep learning and found wide-ranging applications. However, the acquisition of point cloud data is still inconvenient, because scanning equipment's design is usually aimed toward professional needs, and the scanning process is more complex than 2D photo capturing [187]. Furthermore, annotating the labels (ground truth) of 3D data for supervised learning tasks is typically more complex and time-consuming than 2D image data [188]. As a result, point cloud datasets tend to be smaller in terms of the number of individual samples, and only using annotated data may not be sufficient for point cloud understanding and applications. In order to better comprehend point cloud data while circumventing time-consuming data annotation, point cloud self-supervised representation learning (SSRL) has gained growing attention in recent years. This paradigm sidesteps the need for data annotation and, with properly designed models and pretext tasks, can yield performance comparable to supervised approaches.

Current SSRL encompasses two popular approaches: contrastive-based [169], [189], [190] and reconstruction-based [162], [191]–[193]. Since reconstruction-based approaches do not require positive or negative samples, they are more feasible to implement and thus have recently received prominence in point cloud understanding studies.

However, with the current substantial advancements in multi-modal learning [161], [194], [195], we identify novel opportunities to enhance SSRL's effectiveness using contrastive objectives. Various studies [161], [196] have shown that the representational capacity of image models can be significantly enhanced when they are aligned with large volumes of textual data. The alignment has even led to impressive performance in zero-shot classification scenarios. In these explorations, image data is studied as if it were a “foreign language” [194]. This naturally provokes a question: can point clouds be regarded as specialized images? Motivated by this question, the present study pursues a point-image contrastive-based approach to point cloud

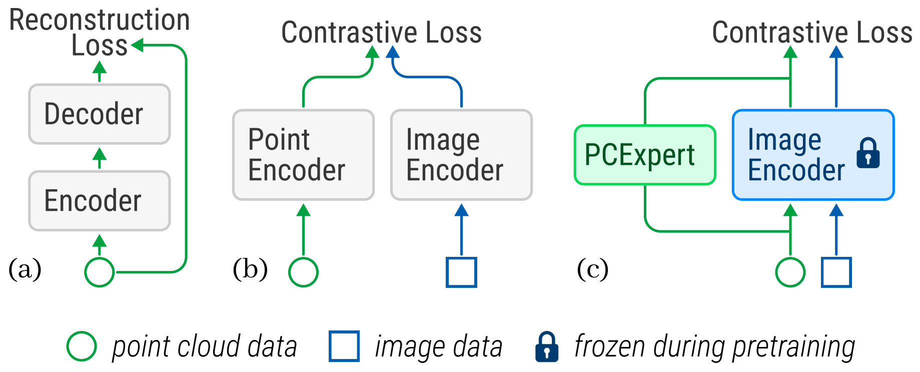


Figure 5.1: Comparison with current SSRL methods. **(a)**: Reconstruction-based methods. **(b)**: Contrastive-base methods. **(c)**: Our approach employs a pre-trained image model to encode both point and image data, and a modular network (PCExpert) for point cloud-specific knowledge acquisition.

understanding.

The standpoint of considering point cloud data as "specialized images" brings a paradigm shift in our mindset towards the design of architectures. Firstly, to address the aforementioned issue of the scarcity of point cloud datasets, we propose that models pre-trained on large-scale image datasets, instead of point datasets, can also serve a crucial role in guiding point cloud learning. This proposition is supported by recent research [163], [164], where the CLIP model [161] was employed as guidance. Secondly, in order to transfer knowledge more effectively between modalities, we assume that a substantial degree of parameter sharing between the image and point cloud encoders can be beneficial. Previous studies on point-image contrastive learning generally utilize separate encoders for each modality [163], [164], [169], [197]–[199](Figure 5.1). This separated encoding facilitates the adaptation of inductive biases to each modality. However, these methods miss the potential to apply knowledge acquired from large-scale image to point data at a deeper level through parameter sharing.

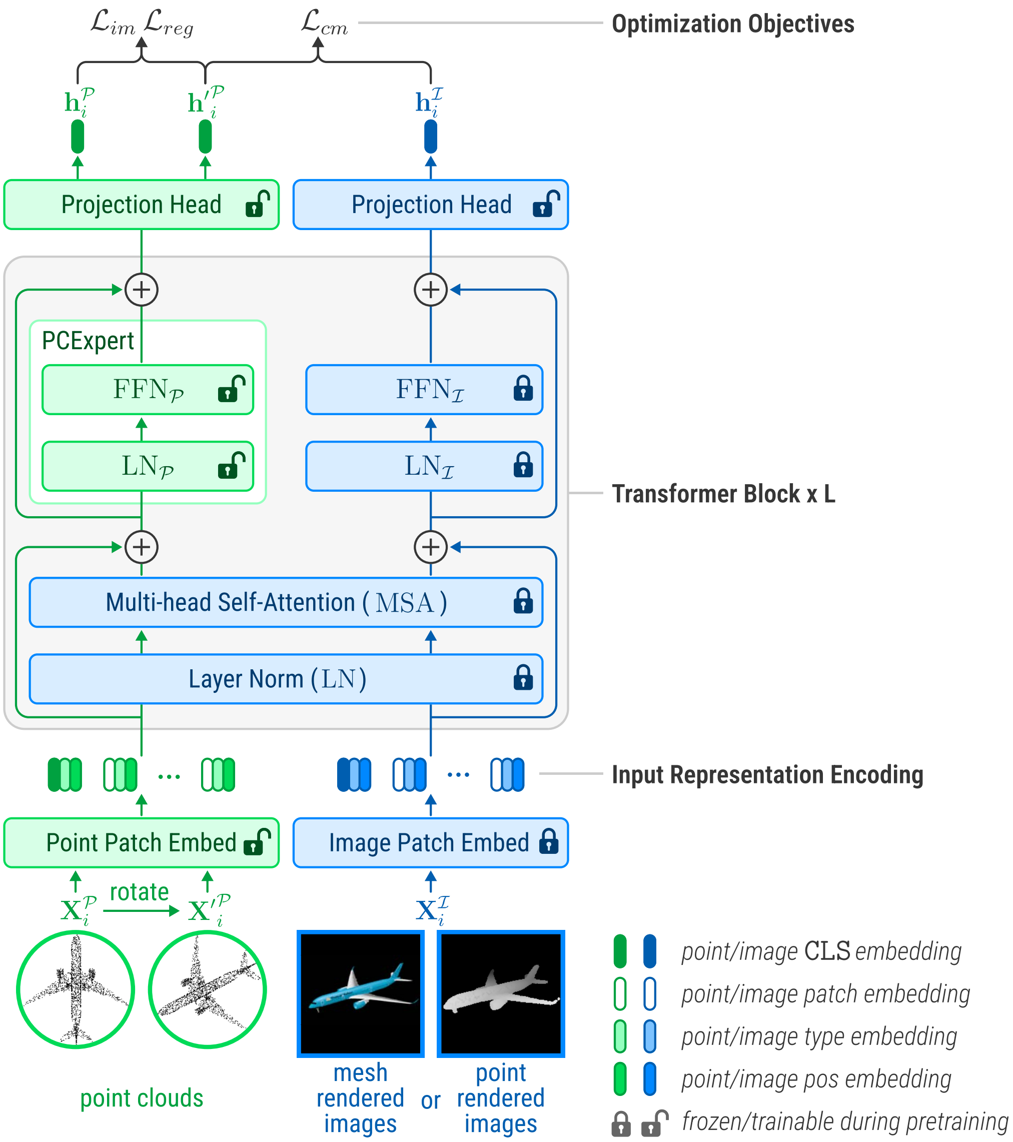
In this study, a multi-way Transformer [196] is adopted for point-image contrastive learning. Throughout the encoding of image and point data, this architecture enables an extensive sharing of parameters belonging to the image encoder, while providing a modular network for the acquisition of point cloud-specific knowledge. As this modular network is solely dedicated to the processing of point cloud data, we call it "PCExpert" (Point Cloud Expert) in this study. Figure 5.2 illustrates the pipeline of our proposed PCExpert architecture, detailing three key components: 1) the process of input representations, 2) the integration of PCExpert within Transformer blocks, and 3) the employed learning objectives for SSRL. Furthermore, PCExpert can also be conceptualized as a plug-in system for pre-trained Transformers. This system extends the network's functionality to a new modality with only a marginal increment in the number of parameters, while preserving the performance of the original model.

In addition to the proposed PCExpert architecture, this study also introduces a novel pretext task for point-image contrastive learning. Drawing from the insights in study [200] that learning factors of variation can enhance invariance learning, we hypothesize that the task of estimating transformation parameters can be a good complement to the conventional contrastive learning objectives. Therefore, we propose to minimize "regression loss" during the estimation of the transformation parameters, and reinforce the learning of more descriptive representations which are capable of differentiating point clouds by leveraging their intrinsic relationships.

In experimental results, PCExpert exhibits robust representational capacity, with a much lower number of parameters in comparison to the current SSRL methods. Combined with the regression loss, the model has achieved state-of-the-art (SOTA) results across several benchmarks. For instance, in the real-world dataset ScanObjectNN, PCExpert achieved an overall accuracy (OA) of 90.02% in the `LINEAR` finetuning protocol, with a 5% improvement over the SOTA performance. Furthermore, we have also taken into account the circumstance where the dataset lacks contrastive images. We conducted a parallel series of pre-training that is solely based on the 3D modality, using images rendered directly from point clouds. Even in this scenario, PCExpert still achieves performance on par with the established benchmarks.

Our main contributions in this study are as follows:

- We propose PCExpert for point cloud SSRL. To the best of our knowledge, it is the first architecture that exploits both image knowledge guidance and extensive parameter sharing in image-point contrastive learning. PCExpert provides evidence that Transformer blocks for image encoding are also capable



$\mathcal{L}_{im}$ $\mathcal{L}_{reg}$
$\mathcal{L}_{cm}$
Optimization Objectives
$\mathbf{h}_i^{\mathcal{P}}$
$\mathbf{h}_i'^{\mathcal{P}}$
$\mathbf{h}_i^{\mathcal{I}}$
Projection Head
Projection Head
PCExpert
$\mathrm{FFN}_{\mathcal{P}}$
$\mathrm{LN}_{\mathcal{P}}$
$\mathrm{FFN}_{\mathcal{I}}$
$\mathrm{LN}_{\mathcal{I}}$
Transformer Block x L
Multi-head Self-Attention (MSA)
Layer Norm (LN)
Input Representation Encoding
Point Patch Embed
Image Patch Embed
$\mathbf{X}_i^{\mathcal{P}}$
rotate
$\mathbf{X}_i'^{\mathcal{P}}$
$\mathbf{X}_i^{\mathcal{I}}$
point clouds
mesh rendered images
or
point rendered images
point/image CLS embedding
point/image patch embedding
point/image type embedding
point/image pos embedding
frozen/trainable during pretraining

Figure 5.2: The pipeline of PCExpert. **Left**: The input representations consist of sequences of embeddings, which are the summation of the patch/`CLS` tokens, the type embeddings and the position embeddings for the respective point and image data. **Middle**: The point and image input representations are then fed into a series of Transformer blocks. In each block, the representations are first processed by a shared Vision Transformer (ViT) Multi-head Self-Attention (MSA) module, and then processed by separate Feed Forward Networks (FFNs), according to their modality. **Right**: During the pre-training process, the parameters in ViT are kept frozen, while only the parameters related to point processing and projection heads are optimized, via three objectives: cross-modal contrastive ($\mathcal{L}_{cm}$), intra-modal contrastive ($\mathcal{L}_{im}$) and rotation angle regression ($\mathcal{L}_{reg}$).

of directly encoding point clouds, thus allowing knowledge of large-scale image data to be utilized for point cloud understanding, in a more intricate manner.

- We develop an alternative approach for rendering images directly from point clouds for image-point contrastive learning, which reduces the cost and difficulty associated with data collection. Our research indicates that for the positive sample pair, mesh-rendered images are not essential. Instead, images directly rendered by point clouds can be used as positive samples, with only a minimal impact on performance.

- In the pre-training phase, we introduce transformation parameter estimation as an extra pretext task, leveraging on the regression loss. In synergy with the contrastive objectives, this task further enhances model performance.

In summary, our research demonstrates that point cloud understanding can be reconceptualized and realized as the understanding of "specialized images". More importantly, the substantial advancements in current multi-modal learning are significantly driven by 1) the exploitation of large-scale datasets and 2) the scalability and versatility of Transformers. With this perspective, our work presents a promising pathway towards more effective self-supervised point cloud understanding through image-assisted cross-modal learning, leveraging the potential of large-scale, low-cost datasets and pre-trained multi-modal Transformers.

The remainder of this chapter is organized as follows. Section 5.2 provides a comprehensive review of related work, focusing on multi-modal studies on point cloud learning. Section 5.3 presents our proposed PCExpert, outlining the key components and algorithms. In Section 5.4, we illustrate the experimental setup and present the evaluation results, as well as ablation studies, and discussions on our findings. Finally, Section 5.5 concludes the chapter, summarizing the key findings and discussing future directions.

5.2 Related Works

5.2.1 Contrastive Learning for Point-image Modality

A major subset of point cloud SSRL methodologies are based on contrastive learning principles. The primary objective of these approaches is to maximize the agreement between different views of the same 3D object while simultaneously minimizing the agreement between unrelated ones. An effective ingredient to this learning paradigm is harnessing the image modality to provide complementary information for point cloud understanding [169], [197]–[199], [201]–[203]. For instance, Jing *et al.* [197] introduced the Center loss, aimed at aligning features across multiple modalities. Some studies [204], [205] leveraged a pre-existing embedding distribution with a pre-trained image model, to guide point cloud feature distillation.

A notable contribution by Afham *et al.* [169] was the proposition of intra- and cross-modal contrastive loss, which enhances point-image alignment and point instance discrimination, simultaneously. As distinct from the instance-level contrastive, Li *et al.* [199] proposed a patch-level contrastive approach for better spatial comprehension, using the Hungarian Algorithm. Moreover, some research works [198], [201] advocate pixel/point-level contrastive learning to facilitate local feature correspondence. Zhou *et al.* [202] proposed multi-scale contrastive objectives between multi-modality objects, enabling local-to-global feature alignment.

Predominantly relying on feature alignment, these methodologies naturally adopt separate feature extractors for each modality. In contrast, our approach employs an image model for the encoding of both modalities. Additionally, we introduce a unique task that leverages "regression loss" for transformation parameter estimation. To the best of our knowledge, we are the first to apply this objective to point-image contrastive learning.

5.2.2 Knowledge Transfer with Pre-trained Image Models

Instead of exploiting feature-level guidance for knowledge transfer, an alternate strand of research seeks to directly conduct point cloud understanding with pre-trained *image* models.

For instance, Xu *et al.* [206] proposed an "inflating" method to convert 2D convolutional networks, pre-trained on image datasets, to 3D convolutional networks, thus catering to point cloud/voxel processing. Studies in [207], [208] conducted point cloud analysis, by transforming point data into images that are recognizable by pre-trained image models. To realize the transformation, techniques including geometry-preserving projection with geometry-aware coloring [208], and multi-view projection [207] are applied, respectively. Rong *et al.* [209] utilized a pre-trained image semantic segmentation model to process images rendered from point clouds in 3D scene segmentation tasks. Moreover, Dong *et al.* [162] leveraged a pre-trained image model as a cross-modal teacher during point cloud masked modelling.

In this study, based on our perspective that point clouds are "specialized images", we directly feed tokenized point data into a pre-trained image model, and introduce PCExpert for point-specific knowledge acquisition.

5.3 The Proposed Method

As depicted in Figure 5.2, the architecture of the PCExpert module, combined with the pre-trained Vision Transformer (ViT), serves as a foundation for processing multi-modal inputs, *i.e.*, point and image data. Before being fed into the transformer blocks, point and image data are initially embedded in a D-dimensional space as sequential input representations. Within each transformer block, the point and image input representations are first processed by the Multi-head Self-Attention (MSA) module of the original ViT. The representations are then subjected to parallel projection paths in separate feed forward networks (FFNs), according to their modality. During point cloud SSRL, the parameters in ViT are kept frozen, while only the parameters of PCExpert and the projection heads are optimized, via three objectives: cross-modal contrastive, intra-modal contrastive and transformation parameter (*i.e.*, the rotation angle) regression. This architecture and training strategy focus on point-specific representation learning with extensive image-to-point knowledge transfer, without affecting the original ViT performance on image-related tasks.

In the following sections, we provide a concrete explanation of the construction of input representations (Section 5.3.1), the architecture of PCExpert (Section 5.3.2) and the pre-training process of point cloud SSRL (Section 5.3.3).

5.3.1 Input Representations

In this section, we explain the process to generate sequences of point and image representations as the input of Transformer blocks. In point cloud SSRL, a training batch comprises N triplets, *i.e.*, $\{\mathbf{X}_i^{\mathcal{P}}, \mathbf{X'}_i^{\mathcal{P}}, \mathbf{X}_i^{\mathcal{I}}\}_{i=1}^N$, where the superscripts $\mathcal{P}$ and $\mathcal{I}$ denote the point and image modalities[1], respectively. To generate data for the intra-modal contrastive objective (see Section 5.3.3), each point data $\mathbf{X}^{\mathcal{P}}$ is applied with a random transformation, resulting in $\mathbf{X'}^{\mathcal{P}}$. Then, the point and image data are embedded in a D-dimensional space, as the input for the Transformer blocks, which we describe below.

Point Input Representations

A point cloud $\mathbf{X}^{\mathcal{P}} \in \mathbb{R}^{M\times 3}$ (or $\mathbf{X'}^{\mathcal{P}}$) consists of M points defined by coordinates in an (x, y, z) Cartesian space. Following the previous study in [192], we sample $N^{\mathcal{P}}$ centroids using farthest point sampling (FPS). To each of these centroids, we assign k neighbouring points by conducting a k-nearest neighbour (kNN) search. Thereby, we obtain $N^{\mathcal{P}}$ local geometric groups $\{G_i\}_{i=1}^{N^{\mathcal{P}}}$, where each group G_i consists of a centroid $\mathbf{x}_{i,0}^{\mathcal{P}}$, and its k neighboring points $\{\mathbf{x}_{i,j}^{\mathcal{P}}\}_{j=1}^k$, *i.e.*, $G_i = \{\mathbf{x}_{i,j}^{\mathcal{P}}\}_{j=0}^k$.

The patch embeddings $\{\mathbf{Z}_i^{\mathcal{P}}\}_{i=1}^{N^{\mathcal{P}}}$ for $\{G_i\}_{i=1}^{N^{\mathcal{P}}}$ are extracted with a two-layer PointNet++ [210], where $\mathbf{Z}_i^{\mathcal{P}} \in \mathbb{R}^D$ and D is the embedding size. Concretely, for $j = 1, ..., N^{\mathcal{P}}$,

$$\tilde{\mathbf{Z}}_i^{\mathcal{P}} = \max_{\mathbf{x}_{i,j}^{\mathcal{P}} \in G_i} [\, f_1(\mathbf{x}_{i,j}^{\mathcal{P}}\, ; \, \mathbf{x}_{i,j}^{\mathcal{P}} - \mathbf{x}_{i,0}^{\mathcal{P}})\,] \tag{5.1a}$$

$$\mathbf{Z}_i^{\mathcal{P}} = \max_{\mathbf{x}_{i,j}^{\mathcal{P}} \in G_i} [\, f_2(\mathbf{x}_{i,j}^{\mathcal{P}}\, ; \, \tilde{\mathbf{Z}}_i^{\mathcal{P}})\,], \tag{5.1b}$$

where f_1 and f_2 are Multi-layer Perceptrons (MLPs). A learnable class embedding $\mathbf{Z}_{\texttt{CLS}}^{\mathcal{P}} \in \mathbb{R}^D$ is prepended to the sequence of the patch embeddings.

To obtain the point input representations $\mathbf{H}_0^{\mathcal{P}} \in \mathbb{R}^{(N^{\mathcal{P}}+1)\times D}$, we sum the sequence of patch embeddings with point position embeddings $\mathbf{Z}_{pos}^{\mathcal{P}} \in \mathbb{R}^{(N^{\mathcal{P}}+1)\times D}$ and a point

[1] This notation is maintained consistently throughout this chapter to signify the two modalities.

type embedding $\mathbf{Z}^{\mathcal{P}}_{type} \in \mathbb{R}^{D}$:

$$\mathbf{H}^{\mathcal{P}}_{0} = [\mathbf{Z}^{\mathcal{P}}_{\texttt{CLS}},\, \mathbf{Z}^{\mathcal{P}}_{1},\, ...,\, \mathbf{Z}^{\mathcal{P}}_{N^{\mathcal{P}}}] + \mathbf{Z}^{\mathcal{P}}_{pos} + \mathbf{Z}^{\mathcal{P}}_{type} \tag{5.2}$$

Position embeddings $\mathbf{Z}^{\mathcal{P}}_{pos}$ are derived by applying a non-linear MLP on centroid points $\{\mathbf{x}^{\mathcal{P}}_{i,0}\}^{N^{\mathcal{P}}}_{i=1}$. In the case of $\mathbf{Z}^{\mathcal{P}}_{\texttt{CLS}}$, a virtual centroid with coordinates set at $(0, 0, 0)$ is used to generate the positional embedding.

Image Input Representations

We follow the studies in [171], [196] and split the image data $\mathbf{X}^{\mathcal{I}} \in \mathbb{R}^{H\times W\times C}$ into $N^{\mathcal{I}}$ patches $\{\mathbf{x}^{\mathcal{I}}_{i}\}^{N^{\mathcal{I}}}_{i=1}$, where $N^{\mathcal{I}} = HW/P^2$, $\mathbf{x}^{\mathcal{I}}_{i} \in \mathbb{R}^{P^2\times C}$, C is the number of channels, and (H, W) and (P, P) are the resolutions of the image and patches, respectively. The sequence of image patch embeddings $\{\mathbf{Z}^{\mathcal{I}}_{i}\}^{N^{\mathcal{I}}}_{i=1}$ are linearly projected from these patches: $\mathbf{Z}^{\mathcal{I}}_{i} = \mathbf{V}\mathbf{x}^{\mathcal{I}}_{i}$ with $\mathbf{V} \in \mathbb{R}^{(P^2\times C)\times D}$.

Similar to the point input representations, the image input representations $\mathbf{H}^{\mathcal{I}}_{0} \in \mathbb{R}^{(N^{\mathcal{I}}+1)\times D}$ are calculated by summing the image patch embeddings (prepended by the class embedding $\mathbf{Z}^{\mathcal{I}}_{\texttt{CLS}}$) with image position embeddings $\mathbf{Z}^{\mathcal{I}}_{pos} \in \mathbb{R}^{(N^{\mathcal{I}}+1)\times D}$ and an image type embedding $\mathbf{Z}^{\mathcal{I}}_{type} \in \mathbb{R}^{D}$:

$$\mathbf{H}^{\mathcal{I}}_{0} = [\mathbf{Z}^{\mathcal{I}}_{\texttt{CLS}},\, \mathbf{Z}^{\mathcal{I}}_{1},\, ...,\, \mathbf{Z}^{\mathcal{I}}_{N^{\mathcal{I}}}] + \mathbf{Z}^{\mathcal{I}}_{pos} + \mathbf{Z}^{\mathcal{I}}_{type} \tag{5.3}$$

5.3.2 PCExpert

Inspired by previous works [196], [211], we propose PCExpert (Point Cloud Expert), a specialized network for enhancing point cloud understanding based on image knowledge. We employ a pre-trained ViT to encode both point and image data. Different from the standard ViT, our architecture incorporates separate feed forward networks (FFNs), each dedicated to a specific modality (denoted by $\text{FFN}^{\mathcal{P}}$ and $\text{FFN}^{\mathcal{I}}$). Concretely, if we denote by $\mathbf{H}^{\mathcal{P}}_{l-1}$ and $\mathbf{H}^{\mathcal{I}}_{l-1}$ the point and image input representations for the l-th transformer block, then the output representations for point cloud and image can be computed respectively as:

$$\tilde{\mathbf{H}}^{\mathcal{P}}_{l} = \text{MSA}\,(\,\text{LN}\,(\,\mathbf{H}^{\mathcal{P}}_{l-1}\,)) + \mathbf{H}^{\mathcal{P}}_{l-1} \tag{5.4a}$$

$$\mathbf{H}^{\mathcal{P}}_{l} = \text{FFN}^{\mathcal{P}}\,(\,\text{LN}^{\mathcal{P}}\,(\,\tilde{\mathbf{H}}^{\mathcal{P}}_{l}\,)) + \tilde{\mathbf{H}}^{\mathcal{P}}_{l} \tag{5.4b}$$

and

$$\tilde{\mathbf{H}}_l^{\mathcal{I}} = \mathrm{MSA}\,(\,\mathrm{LN}\,(\,\mathbf{H}_{l-1}^{\mathcal{I}}\,)) + \mathbf{H}_{l-1}^{\mathcal{I}} \tag{5.5a}$$

$$\mathbf{H}_l^{\mathcal{I}} = \mathrm{FFN}^{\mathcal{I}}\,(\,\mathrm{LN}^{\mathcal{I}}\,(\,\tilde{\mathbf{H}}_l^{\mathcal{I}}\,)) + \tilde{\mathbf{H}}_l^{\mathcal{I}} \tag{5.5b}$$

where LN denotes the layer normalisation operation. The mutual Multi-head Self-Attention (MSA) module facilitates image knowledge sharing with the point modality, while the separate FFNs ensure that the unique features of each modality are effectively captured and integrated into the overall representations.

5.3.3 Training

PCExpert is primarily trained with a point-image contrastive objective, to exploit the guidance offered by the image modality. Furthermore, we follow the method in [169] and implement an intra-modal contrastive learning for the purpose of enhancing invariance learning of point semantics. Drawing inspiration from [200], we integrate transformation parameter estimation as an additional pretext task. This enables the model to capture the causal knowledge embodied in the representations and to mitigate the influence of confounding factors of variation, thus refining the quality of the learned representations.

As for the optimization process during training, only the parameters of PCExpert are updated via back-propagation, while the original parameters of ViT are frozen. This training strategy ensures the focus of optimization specific to point clouds, and does not compromise the model performance on images. This also significantly reduces computation and storage requirements, as only a small fraction ($\approx 6.6\%$) of parameters are updated.

Cross-modal Contrastive Learning

Given a batch comprising N point-image pairs $\{\mathbf{X}_i^{\mathcal{P}}\}_{i=1}^N$ and $\{\mathbf{X}_i^{\mathcal{I}}\}_{i=1}^N$, the purpose of point-image contrastive learning is to discern the corresponding (positive) pairs from a pool of N^2 potential pairs. The output [CLS] tokens of the final (L-th) transformer block $\{\mathbf{H}_{\texttt{CLS},L,i}^{\mathcal{P}}\}_{i=1}^N$ and $\{\mathbf{H}_{\texttt{CLS},L,i}^{\mathcal{I}}\}_{i=1}^N$ are used as the global representations of the point and image data, respectively.

Subsequently, these [CLS] tokens are mapped to an invariant space via two projection heads $f_{\mathcal{P}}$ and $f_{\mathcal{I}}$, *i.e.*,

$$\mathbf{h}_i^{\mathcal{P}} = f_{\mathcal{P}}(\mathbf{H}^{\mathcal{P}}_{\texttt{CLS},L,i}) \tag{5.6}$$

$$\mathbf{h}_i^{\mathcal{I}} = f_{\mathcal{I}}(\mathbf{H}^{\mathcal{I}}_{\texttt{CLS},L,i}) \tag{5.7}$$

The distances between the (normalized) output embeddings $\mathbf{h}_i^{\mathcal{P}}$ and $\mathbf{h}_i^{\mathcal{I}}$ in this space are calculated using cosine similarity. Thereby, the loss function for the positive point-image pair is defined as:

$$\mathcal{L}_{cm,i}^{\mathcal{P}2\mathcal{I}} = -\log \frac{\exp{(\mathbf{h}_i^{\mathcal{P}\top} \mathbf{h}_i^{\mathcal{I}}/\tau)}}{\sum_{j=1}^{N} \exp{(\mathbf{h}_i^{\mathcal{P}\top} \mathbf{h}_j^{\mathcal{I}}/\tau)}}, \tag{5.8}$$

where τ stands for the temperature co-efficient, and the superscript $\mathcal{P}2\mathcal{I}$ signifies that, optimizing this loss facilitates the alignment of the i-th point with the corresponding image among N images. Similarly, if we denote by $\mathcal{I}2\mathcal{P}$ the reciprocal task to align an image with its corresponding point cloud, the cross-modal contrastive loss $\mathcal{L}_{cm}$ is expressed as:

$$\mathcal{L}_{cm} = \frac{1}{2N} \sum_{i=1}^{N} (\, \mathcal{L}_{cm,i}^{\mathcal{P}2\mathcal{I}} + \mathcal{L}_{cm,i}^{\mathcal{I}2\mathcal{P}} \,) \tag{5.9}$$

Intra-modal Contrastive Learning

In addition to aligning the features between point and image modality, we conduct contrast within the point cloud modality.

Given a batch of point cloud data $\{\mathbf{X}_i^{\mathcal{P}}\}_{i=1}^{N}$, we apply transformation T on each sample to get $\{\mathbf{X'}_i^{\mathcal{P}}\}_{i=1}^{N}$. A positive pair is defined as the original sample and its transformed version. Similar to the point-image contrastive loss, the intra-modal contrastive loss $\mathcal{L}_{im}$ can be expressed as:

$$\mathcal{L}_{im} = \frac{1}{2N} \sum_{i=1}^{N} (\, \mathcal{L}_{im,i}^{\mathcal{P}2\mathcal{P}'} + \mathcal{L}_{im,i}^{\mathcal{P}'2\mathcal{P}} \,) \tag{5.10}$$

where the superscripts $\mathcal{P}2\mathcal{P}'$ and $\mathcal{P}'2\mathcal{P}$ signify the original-to-transformed and transformed-to-original sample pair matching, respectively, and $\mathcal{L}_{im,i}^{\mathcal{P}2\mathcal{P}'}$ can be computed as:

$$\mathcal{L}_{im,i}^{\mathcal{P}2\mathcal{P}'} = -\log \frac{\exp{(\mathbf{h}_i^{\mathcal{P}\top} \mathbf{h'}_i^{\mathcal{P}}/\tau)}}{\sum_{j=1}^{N} \exp{(\mathbf{h}_i^{\mathcal{P}\top} \mathbf{h'}_j^{\mathcal{P}}/\tau)}} \tag{5.11}$$

Transformation Parameter Estimation

Furthermore, we use the transformation T in intra-modal contrastive learning as an additional supervisory signal to guide point understanding. The objective of the transformation parameter estimation task is to perform numerical regression on the transformation T's value.

Specifically, we apply y-axis rotation as the transformation in this study, thus making the rotation angle as the ground truth. In order to circumvent numerical cycles caused by periodic symmetry in rotation, we quantize the rotation angles into d categories, and project each category into a $\mathbb{R}^d$ space as a one-hot vector $\{y_i\}_{i=1}^N, y_i \in \mathbb{R}^d$.

To calculate the regression loss, we first calculate the difference between $\mathbf{h}_i^{\mathcal{P}}$ and $\mathbf{h'}_i^{\mathcal{P}}$, and linearly project the resultant difference vector into the same $\mathbb{R}^d$ space using f_T, *i.e.*, $\hat{y}_i = f_T(\mathbf{h}_i^{\mathcal{P}} - \mathbf{h'}_i^{\mathcal{P}})$. Thus, the regression loss $\mathcal{L}_{reg}$ can be represented as:

$$\mathcal{L}_{reg} = \frac{1}{N}\sum_{i=1}^{N}(1 - {y_i}^{\mathsf{T}}\hat{y}_i). \tag{5.12}$$

Finally, the overall objective of the point cloud SSRL is to optimize PCExpert with the combination of the above three losses:

$$\mathcal{L} = \mathcal{L}_{cm} + \mathcal{L}_{im} + \mathcal{L}_{reg}. \tag{5.13}$$

5.4 Experiments

The pre-training of our PCExpert is conducted on the ShapeNet [65] dataset, using the methodology described in Section 5.3. Comprehensive details regarding this pre-training setup are described in Section 5.4.1. We subsequently evaluate the pre-trained model across a variety of 3D point cloud classification benchmarks. Prior to this evaluation, the model is fine-tuned on each downstream task. The model's performance on these downstream tasks is reported in Section 5.4.2. In Section 5.4.3, we engage in a series of ablation studies exploring various aspects of PCExpert.

5.4.1 Pre-training Setup

Dataset

Following previous studies [162], [169], we utilize ShapeNet, a dataset encompassing over $50,000$ CAD models across 55 categories, as the pre-training dataset for PC-Expert. We employ $40,523$ instances across 13 categories in ShapeNet to generate point and image data triplets. For point cloud data, we follow [212] and sample $2,048$ points for each instance, and group them into 160 local patches with group size of 32, whose centroids are sampled with FPS. As described in Section 5.3.3, the intra-modal contrastive loss $\mathcal{L}_{im}$ and the regression loss $\mathcal{L}_{reg}$ are calculated based on the original point data and its transformed version. To obtain the transformed point, we rotate the original point cloud about the y-axis by a predetermined degree between $[0°, 360°]$ according to the corresponding image in the triplet (described below).

For image data, we use two types of rendered images. The first type of images, derived from study [213], are rendered from the CAD meshes with 36 random views for each mesh.

The second type of images are rendered directly from point cloud data in real-time, using the Pytorch3D [214] library, with random rotation angles around y-axis (the yaw angle) and linear grey scale along z-axis. The rotation angles are then recorded and utilized for the corresponding rotation of point cloud samples. In the rasterization pipeline, the radius and point-accumulation for each pixel are set to 0.03 units and 8 points, respectively. The `FoVPerspectiveCameras` is used to produce images with more realistic perspective and depth cues.

The dimensions of image data are set to $224 \times 224 \times 3$, with the patch size of 16×16 and no augmentation applied. The images are then normalized using the standard ImageNet means and standard deviations. Examples of the two types of rendered images are illustrated in Figure 5.3.

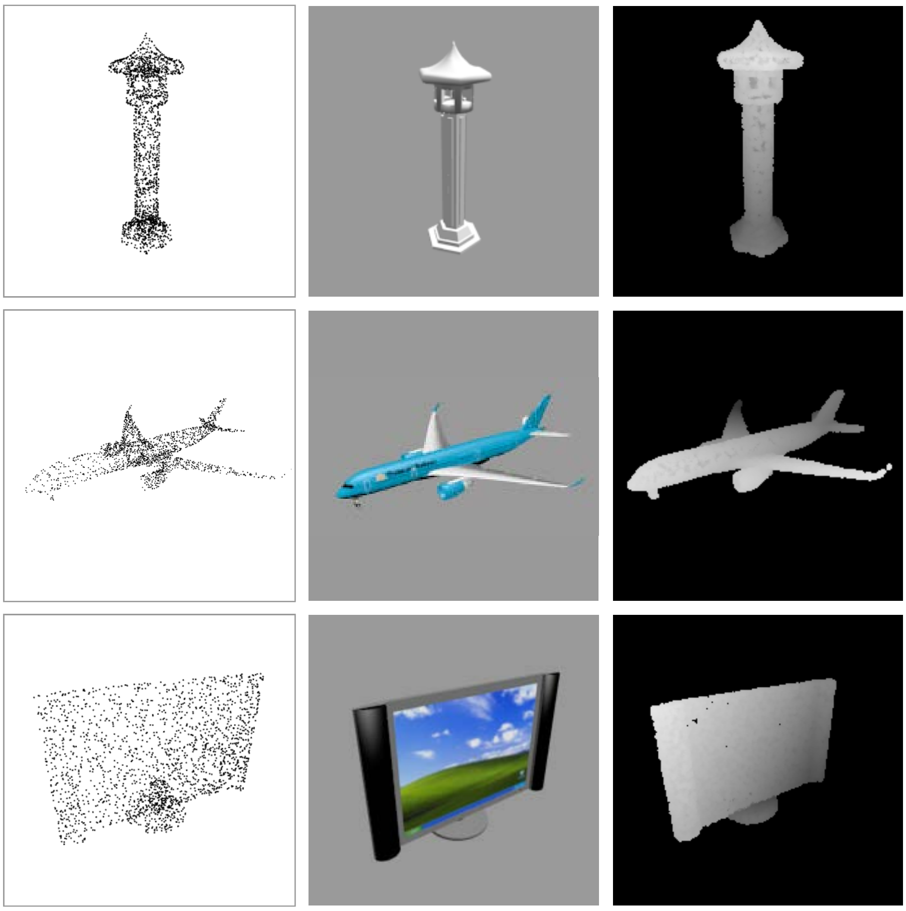


Figure 5.3: Training samples used in point-image contrastive learning. **Left**: Point cloud samples. **Middle**: Images rendered from 3D CAD meshes. **Right**: Images rendered directly from the original point clouds, with the shape and details well preserved.

Model

We adopt the image tower of the CLIP model [161] as the base ViT for point and image data encoding, which consists of 12-layer Transformer blocks, with 768 hidden size and 12 attention heads. The PCExpert module is applied on each Transformer block with a projection dimension of 192.

The model is pre-trained for 300 epochs, with a batch size of 1024. AdamW [215] optimizer is employed with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The learning rate is initialized to $1e-3$ for the model which uses the mesh-rendered images, and $5e-4$ for the model using the point-rendered images, with both weight decays set to 0.01. The training incorporates a linear warmup over the first 10 epochs, followed by a cosine decay.

5.4.2 Evaluation

Fine-tuning

In this study, we fine-tune the pre-trained model on widely used 3D point cloud classification datasets, including ScanObjectNN [174] and ModelNet40 [173], before evaluating its performance on them. ScanObjectNN, a challenging point cloud dataset, comprises $2,880$ objects spanning 15 categories, all generated via scanning real indoor objects. In line with common practices, three variants of this datasets are used in this study, *i.e.*, 1) OBJ_ONLY: the vanilla dataset including only segmented objects; 2) OBJ_BG: a noisier variant including objects with their background elements; and 3) PB_T50_RS: the most challenging perturbed variant, where each instance is extracted from a bounding box that is randomly shifted up to 50% of its original size from the ground-truth, in addition to random rotation and scaling. ModelNet40 is a synthetic dataset, produced by sampling from 3D CAD models, featuring $12,331$ objects across 40 categories. Consistent with established practice, we sample $1,024$ points from each instance for fine-tuning and test, and the results are denoted by ModelNet40$-$1k. Only random rotation is performed as data augmentation for the training set, following the methodology described by Dong *et al.* [162].

During the fine-tuning stage, the `CLS` tokens from the final output representations are used as the global representations of the samples. Classification heads are employed to project the representations into the target class space. We follow standard research protocols [162] to conduct fine-tuning:

- `FULL`: All parameters of PCExpert and the classification head (a three-layer non-linear MLP) are updated in fine-tuning, while parameters of ViT are kept frozen.
- `LINEAR`: Only parameters of the classification head (a single-layer MLP) are updated during fine-tuning.
- `MLP3`: Only parameters of the classification head (a three-layer non-linear MLP) are updated during fine-tuning. The classification head is the same as that in protocol `FULL`).

During the evaluation stage, no voting techniques are used. All classification results are reported in terms of overall accuracy (OA) unless stated otherwise.

3D Object Classification

The classification results of ScanObjectNN and ModelNet40 are presented in Table 5.1 and Table 5.2, respectively.

Firstly, it can be observed that our PCExpert outperforms the existing state-of-the-art (SOTA) SSRL methods across all benchmarks in `LINEAR` and `MLP3` protocols, especially for the challenging real-world dataset ScanObjectNN, where it achieves the highest accuracy improvement of +4.8% in `LINEAR`. Because the majority of model parameters are not updated during `LINEAR` fine-tuning, the performance on this benchmark heavily relies on the model's generalizability and understanding of the underlying point cloud semantics. This attests to PCExpert's exceptional representation capabilities. With a much smaller model size, PCExpert can still achieve performance comparable to other models under benchmarks using the `FULL` protocol.

Secondly, in comparison to the studies based on point-image contrastive learning (*e.g.*, CrossPoint [169] and MVR [216]), PCExpert significantly outperforms the existing methods, with average improvements of +4.9% and +8.3% under the `FULL` and `LINEAR` protocols, respectively.

Thirdly, despite having minimal inductive bias towards 3D understanding (through its patch embedding module), PCExpert still outperforms models with a strong emphasis on this specific inductive bias, such as Point-M2AE [212], across the majority of benchmarks.

Furthermore, it is found that PCExpert demonstrates better performance improvement on the real-world dataset ScanObjectNN compared to the synthetic ModelNet40. We postulate this superiority is likely a result of the effective knowledge transfer from CLIP, leveraging the vast quantity of real-world image-based training data.

Table 5.1: Classification results on ScanObjectNN. SO-BG, SO-OBJ, and SO-PB: the OBJ_BG, the OBJ_ONLY, and the PB_T50_RS variants of the ScanObjectNN dataset, respectively. *: Results based on Support Vector Machines (SVMs). `CL`: Methods that are based on contrastive learning are marked with √. The overall accuracy (%) is reported.

Method	`CL`	#Params (M)	SO-BG	SO-OBJ	SO-PB
Supervised Learning Only					
PointNet [217]		3.5	73.3	79.2	68.0
PointNet++ [210]		1.5	82.3	84.3	77.9
DGCNN [170]		1.8	82.8	86.2	78.1
PointCNN [218]		0.6	86.1	85.5	78.5
GBNet [219]		8.8	-	-	80.5
PointMLP [220]		12.6	-	-	85.4±0.3
PointNeXt [221]		1.4	-	-	87.7±0.4
with Self-supervised Representation Learning (`FULL`)					
MVR [216]	√	1.8	84.5±0.6	84.3±0.6	-
CrossNet [203]	√	1.8	-	-	-
Point-LGMask [222]	√	-	89.8	89.3	85.3
Transformer [223]		22.1	83.04	84.06	79.11
OcCo [191]		22.1	84.85	85.54	78.79
Point-BERT [192]		22.1	87.43	88.12	83.07
Point-MAE [193]		22.1	90.02	88.29	85.18
Point-M2AE [212]		15.3	91.22	88.81	86.43
ACT [162]		22.1	92.48±0.59	91.57±0.37	87.88±0.36
PCExpert (Ours)	√	**6.1**	**92.66±0.36**	91.39±0.17	87.10±0.20
Improvement			(↑ 0.18)		

(Table continues on next page.)

(Table continues.)

Method	CL	#Params (M)	SO-BG	SO-OBJ	SO-PB
with Self-supervised Representation Learning (`LINEAR`)					
CrossPoint* [169]	√	1.8	81.7	-	-
CrossNet* [203]	√	1.8	83.9	-	-
Point-MAE [193]		22.1	82.58±0.58	83.52±0.41	73.08±0.30
ACT [162]		22.1	85.20±0.83	85.84±0.15	76.31±0.26
PCExpert (Ours)	√	**6.1**	**90.02±0.34**	**89.56±0.20**	**79.42±0.10**
Improvement			(↑ 4.82)	(↑ 3.72)	(↑ 3.11)
with Self-supervised Representation Learning (`MLP3`)					
Point-MAE [193]		22.1	84.29±0.55	85.24±0.67	77.34±0.12
ACT [162]		22.1	87.14±0.22	88.90±0.40	81.52±0.19
PCExpert (Ours)	√	**6.1**	**89.96±0.43**	**89.76±0.42**	**82.57±0.62**
Improvement			(↑ 2.82)	(↑ 0.86)	(↑ 1.05)

Few-shot Point Cloud Classification

The results of few-shot 3D object classification experiments are summarized in Table 5.3.

Several key findings are as follows: First, our PCExpert consistently outperforms the existing methods across all experiments. Specifically, significant performance gains of +4% to +8% are noted under `LINEAR`, and +4% to +10% improvement compared with point-image contrastive methods (*e.g.*, CrossPoint [169]).

Secondly, it can be observed that our PCExpert's performance under the `LINEAR` protocol closely approximates that under the `FULL` protocol. This observation implies that the model is already robustly generalizable after pre-training, and requires only linear projections for effective application. This suggests a decreasing need for extensive fine-tuning of the whole model parameters on specific tasks, a process which typically demands significant time and resources. Interestingly, we observe that as the number of training samples decreases, our model's superiority over the existing SOTA becomes more apparent. For instance, under `LINEAR`, the performance improvement in 10-shot settings is consistently higher than that in 20-shot

Table 5.2: Classification results on ModelNet40−1k. "1k" signifies that 1, 024 points are sampled from each sample during the training and test stages. *: Results based on Support Vector Machines (SVMs). `CL`: Methods that are based on contrastive learning are marked with √. The overall accuracy (%) is reported.

Method	CL	#Params (M)	*Supervised Learning Only*
PointNet [217]		3.5	89.2
PointNet++ [210]		1.5	90.7
DGCNN [170]		1.8	92.9
PointCNN [218]		0.6	92.2
GBNet [219]		8.8	93.8
PointMLP [220]		12.6	94.1
PointNeXt [221]		1.4	93.2

Method	CL	#Params (M)	*Self-supervised Representation Learning*		
			`FULL`	`LINEAR`	`MLP3`
MVR [216]	√	1.8	93.2±0.1	-	-
CrossPoint* [169]	√	1.8	-	91.2	-
CrossNet [203]	√	1.8	93.4	91.5	-
Transformer [223]		22.1	91.4	-	-
OcCo [191]		22.1	92.1	-	-
Point-BERT [192]		22.1	93.2	-	-
Point-MAE [193]		22.1	93.8	91.22±0.26	92.33±0.09
Point-M2AE [212]		15.3	94.0	-	-
ACT [162]		22.1	93.7	91.36±0.17	92.69±0.18
PCExpert (Ours)	√	**6.1**	92.7	**92.22±0.11**	**92.73±0.12**
Improvement				(↑ 0.72)	(↑ 0.04)

settings. This phenomenon indicates that PCExpert can extract and use meaningful features more effectively, while remaining robust to overfitting. This further substantiates the representation capability and generalizability of PCExpert.

These results collectively indicate that our method provides a robust and effective approach for point cloud classification, showing notable improvements over existing

Table 5.3: Few-shot classification results on ModelNet40. The overall accuracy (%) is reported.

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
with Self-supervised Representation Learning (`FULL`)				
CrossPoint [169]	92.5 ± 3.0	94.9 ± 2.1	83.6 ± 5.3	87.9 ± 4.2
Point-LGMask [222]	97.4 ± 2.0	98.1 ± 1.4	92.6 ± 4.3	95.1 ± 3.4
Transformer [223]	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
OcCo [191]	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
Point-BERT [192]	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
Point-MAE [193]	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
Point-M2AE [212]	96.8 ± 1.8	98.3 ± 1.4	92.3 ± 4.5	95.0 ± 3.0
ACT [162]	96.8 ± 2.3	98.0 ± 1.4	93.3 ± 4.0	95.6 ± 2.8
PCExpert (Ours)	**98.0 ± 1.8**	**98.8 ± 0.9**	**93.8 ± 4.4**	**96.2 ± 3.0**
Improvement	(↑ 0.6)	(↑ 0.7)	(↑ 0.5)	(↑ 0.6)
with Self-supervised Representation Learning (`LINEAR`)				
Point-MAE [193]	91.1 ± 5.6	91.7 ± 4.0	83.5 ± 6.1	89.7 ± 4.1
ACT [162]	91.8 ± 4.7	93.1 ± 4.2	84.5 ± 6.4	90.7 ± 4.3
PCExpert (Ours)	**97.2 ± 1.9**	**97.7 ± 1.4**	**92.9 ± 4.2**	**94.8 ± 3.4**
Improvement	(↑ 5.4)	(↑ 4.6)	(↑ 8.4)	(↑ 4.1)
with Self-supervised Representation Learning (`MLP3`)				
Point-MAE [193]	95.0 ± 2.8	96.7 ± 2.4	90.6 ± 4.7	93.8 ± 5.0
ACT [162]	95.9 ± 2.2	97.7 ± 1.8	92.4 ± 5.0	94.7 ± 3.9
PCExpert (Ours)	**97.0 ± 2.6**	**98.5 ± 1.0**	**92.8 ± 3.8**	**95.5 ± 2.9**
Improvement	(↑ 1.1)	(↑ 0.8)	(↑ 0.4)	(↑ 0.8)

techniques, even with reduced trainable parameters. The benefit of incorporating an image-text pre-trained model, specifically CLIP, has been particularly demonstrated in the tasks of the `LINEAR` fine-tuning and few-shot classification. To further explain this point, it is necessary to revisit the findings presented in previous chapters. The experimental results in Section 3.4.2 and Section 4.3.2 show the effectiveness of generalizable knowledge learning and the regression loss in self-supervised learning paradigms. For image-text contrastive learning, its objective can be viewed as to predict the probability of consistency between the content in an image (result of transformation) and the textual description (transformation parameters). Therefore, image-text contrastive learning can be considered as a variant of generalizable knowledge learning (see Section 4.4.3 and Figure 4.3). This novel perspective suggests that enormous textual training data can significantly enhance generalizability of knowledge about 2D images acquired with CLIP. This generalizability is demonstrated most effectively in tasks regarding 3D understanding using the model trained mainly on 2D images, especially in scenarios of linear fine-tuning and few-shot learning.

5.4.3 Ablation Studies

Feed Forward Networks (FFNs)

The PCExpert architecture employs an additional FFN within each transformer block. To ascertain the contribution of these FFNs, an ablation study is performed. This involves the removal of all additional FFNs from the architecture, making the remaining ViT tower from CLIP the sole encoder for data across both image and point cloud domains.

The experimental results presented in Table 5.4 reveal that the architecture without FFNs struggles to generate effective representations for point clouds, causing reduced accuracy score under the `LINEAR` fine-tuning protocol. The results highlight the apparent discrepancy between the domains of image and point cloud. While both image and point cloud can be employed to represent similar objects, there exist distinct conceptual dimensions in the representation that are not mutually shared between these two modalities. Therefore, it is necessary to integrate additional parameters to encode modality-specific differences, and thus bridge the substantial gap between modalities. This investigation highlights the critical role of FFNs in enhancing the model's performance, and thus validating their incorporation in the PCExpert architecture.

Table 5.4: Ablation study on the effectiveness of incorporating FFNs. SO-BG: the OBJ_BG split of the ScanObjectNN dataset. MN-1k: the 1k-sampling setting of the ModelNet40 dataset. The overall accuracy (%) under the `LINEAR` protocol is reported.

Method	SO-BG	MN-1k
w/o FFNs	76.59	81.30
w/ FFNs (PCExpert)	90.02	92.22

Parameter Sharing

Based on our novel idea of reinterpreting point clouds as specialized images, we propose that extensive parameter sharing with image encoders can be beneficial for point cloud understanding. To validate this, we compare the performance between the proposed PCExpert and separate point encoders (*e.g.*, Transformer and DGCNN [170]), as shown in Table 5.5. The comparisons are based on the same pre-training dataset and objectives, with the only difference being whether ViT participates in point data encoding. For the separated encoders, the losses are calculated based on the point output representations from the Transformer or DGCNN, and the image output representations from ViT, where ViT does not access or process point data.

It can be observed from the results that PCExpert exhibits better performance in both evaluations, despite the inductive bias of the separate encoder (*e.g.*, DGCNN [170]). This outcome strongly suggests the crucial contribution of parameter sharing in image knowledge transfer, which thereby enhances the model's representation generalizability. Notably, given that Transformer architectures possess advantages in: 1) end-to-end learning on data from heterogeneous modality, and 2) the scalability with increased computational resources, our approach in this study presents a promising direction for future multi-modal studies and applications on point clouds.

Point Cloud-Rendered Images

Table 5.6 shows the performance of PCExpert pre-trained using point cloud-rendered images (dubbed "PCExpert-P").

Table 5.5: Ablation on network architecture. SO-BG: the OBJ_BG split of the ScanObjectNN dataset. MN-1k: the 1k-sampling setting of the ModelNet40 dataset. The overall accuracy (%) under the `LINEAR` protocol is reported.

Method	Parameter Sharing	SO-BG	MN-1k
Transformer	×	86.19±0.26	89.42±0.12
DGCNN	×	88.73±0.43	89.44±0.26
PCExpert	√	90.02±0.34	92.22±0.11

Our experimental results reveal that PCExpert, pre-trained using mesh-rendered data (dubbed "PCExpert-M"), consistently exhibits superior performance in the majority of benchmarks, particularly in the few-shot experiments and those under the `LINEAR` protocol. This suggests that the exploitation of mesh-rendered images, which more closely resembles real-world photos, is more beneficial for robust representation capabilities. Nonetheless, it is important to highlight that PCExpert-P, in some experiments under the `FULL` protocol, demonstrates superior performance to PCExpert-M. This can be attributed to PCExpert-P's pre-training that is solely based on the single modality of point clouds, which makes it easier for the model to optimize in tasks related to point clouds.

Moreover, while being translated into a suitable form for image encoders, the point cloud-rendered images have preserved crucial semantic characteristics of the original point clouds, as shown in Figure 5.3. As a result, the performance gap between the two models is marginal, and PCExpert-P also surpasses existing SOTA in many benchmarks. Given that utilizing point cloud-rendered images can significantly reduce dataset creation costs and difficulties, the minor performance deficiencies can be compensated by leveraging larger quantities of data in the absence of mesh-rendered images.

Pre-training Objectives

We conduct an ablation study to assess the significance of different pre-training objectives. The results are summarized in Table 5.7.

Our analysis reveals that the inclusion of the regression loss $\mathcal{L}_{reg}$ yields better performance in both scenarios of $\mathcal{L}_{cm}+\mathcal{L}_{reg}$ and $\mathcal{L}_{cm}+\mathcal{L}_{im}+\mathcal{L}_{reg}$. These results provide evidence that the parameter estimation task contributes to enhancing the model's representation capability.

However, we have observed a notable decrease in performance when incorporating $\mathcal{L}_{im}$ with $\mathcal{L}_{cm}$, a finding that stands in contrast to that reported in study [169]. The discrepancy can likely be attributed to loss balancing issues within our architecture, as the intra-modal contrastive task might be more challenging for the ViT model employed in our study, compared to the 3D-specific encoder (*i.e.*, DGCNN [170]) used in study [169]. This difficulty may cause the model to focus excessively on minimizing $\mathcal{L}_{im}$, thereby neglecting $\mathcal{L}_{cm}$ and compromising the generalizability of the overall representation.

Interestingly, when $\mathcal{L}_{reg}$ is introduced into the mix, it appears to interactively reduce the difficulty of optimizing for $\mathcal{L}_{im}$. As shown in Figure 5.4, when optimizing for $\mathcal{L}_{reg}$, there is a concurrent reduction in $\mathcal{L}_{im}$ (the left plot in Figure 5.4), even though the latter loss is intentionally excluded from the gradient calculation. However, the reverse relationship is not true (the right plot in Figure 5.4). This discovery suggests that the characteristics of point cloud learned through $\mathcal{L}_{reg}$ contribute to the objective of $\mathcal{L}_{im}$, establishing a beneficial synergy between the two losses. This interplay results in less optimization difficulty and, consequently, optimal model performance.

5.5 Conclusion

In conclusion, this chapter proposed PCExpert, a novel architecture for point cloud self-supervised representation learning. By employing extensive parameter sharing with a pre-trained ViT and the parameter estimation task with the "regression loss", PCExpert has demonstrated remarkable performance across multiple benchmarks, especially in experiments under `LINEAR` protocol and in few-shot scenarios. PCExpert's performance serves as a solid validation for our proposition of reconsidering point clouds as images. This standpoint is also reflected in our novel approach of generating contrastive images directly from point cloud rendering, which opens up new possibilities for augmenting point cloud datasets for contrastive learning. By

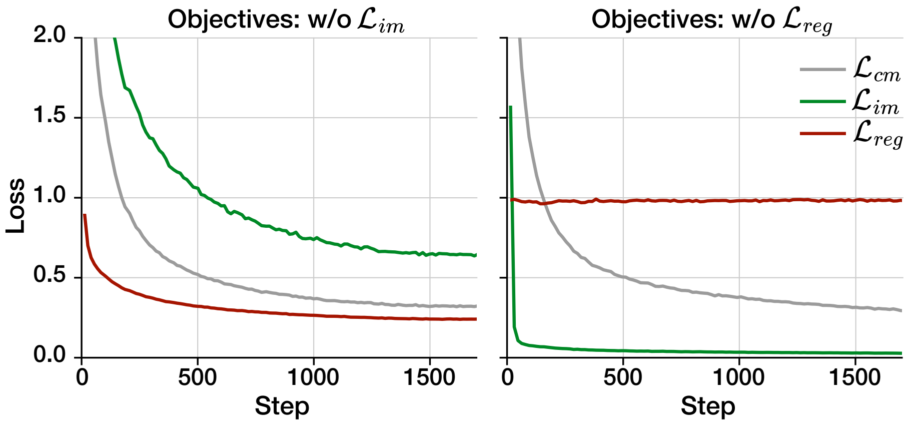


Figure 5.4: **left**: Gradient calculation is based on $\mathcal{L}_{cm}$ and $\mathcal{L}_{reg}$, excluding $\mathcal{L}_{im}$. Optimizing for $\mathcal{L}_{reg}$ (the red curve) concurrently results in a reduction of $\mathcal{L}_{im}$ (green). **right**: Gradient calculation is based on $\mathcal{L}_{cm}$ and $\mathcal{L}_{im}$, excluding $\mathcal{L}_{reg}$. Optimizing for $\mathcal{L}_{im}$ has no effect on $\mathcal{L}_{reg}$.

these means, this study strengthens our understanding on the exploitation of generalizable knowledge, and indicates a promising direction for future studies on transfer learning based on text-image models, through parameter sharing and the scalability of Transformers.

Table 5.6: Comparison of PCExpert performance pre-trained with mesh rendered (PCExpert-M) and point cloud rendered images (PCExpert-P). SO-BG, SO-OBJ, and SO-PB: the OBJ_BG, the OBJ_ONLY, and the PB_T50_RS variants of the ScanObjectNN dataset, respectively. MN-1k: the 1k-sampling setting of the ModelNet40 dataset. MN-*i*w*j*s: the *i*-way *j*-shot few-shot setting of the ModelNet40 dataset. The overall accuracy (%) is reported.

Protocol	Benchmark	PCExpert-M	PCExpert-P
`FULL`	SO-BG	91.91	**92.66**
	SO-OBJ	91.22	**91.39**
	SO-PB	87.09	**87.10**
	MN-1k	92.50	**92.67**
	MN-5w10s	**98.0±1.8**	96.5 ± 2.7
	MN-5w20s	**98.8±0.9**	98.0 ± 1.5
	MN-10w10s	**93.8±4.4**	93.2 ± 4.6
	MN-10w20s	**96.2±3.0**	95.6 ± 3.2
`LINEAR`	SO-BG	**90.02**	88.30
	SO-OBJ	**89.56**	87.09
	SO-PB	**79.42**	77.03
	MN-1k	**92.22**	91.33
	MN-5w10s	**97.2±1.9**	97.0 ± 2.8
	MN-5w20s	**97.7±1.4**	97.0 ± 2.2
	MN-10w10s	**92.9±4.2**	90.8 ± 5.2
	MN-10w20s	**94.8±3.4**	93.1 ± 4.3
`MLP3`	SO-BG	89.96	**90.53**
	SO-OBJ	89.76	**89.85**
	SO-PB	**82.57**	81.96
	MN-1k	**92.73**	92.34
	MN-5w10s	**97.0±2.6**	96.9 ± 2.5
	MN-5w20s	**98.5±1.0**	97.8 ± 1.5
	MN-10w10s	**92.8±3.8**	91.2 ± 4.8
	MN-10w20s	**95.5±2.9**	94.2 ± 4.1

Table 5.7: Ablation on pre-training objectives. Overall accuracy (%) on the ScanObjectNN OBJ_BG (SO-BG) benchmark under the `LINEAR` protocol are reported.

$\mathcal{L}_{cm}$ (cross-modal)	$\mathcal{L}_{im}$ (intra-modal)	$\mathcal{L}_{reg}$ (regression)	SO-BG
√	×	×	84.71±0.18
√	√	×	82.96±0.12
√	×	√	85.48±0.17
√	√	√	90.02±0.34

Chapter 6

Generalization and Beyond

In this concluding chapter, we comprehensively revisit the primary motivations, methodologies and outcomes of the research. This chapter aims to summarize our key findings, and evaluate the contributions of our research to the field (Section 6.1). Furthermore, we will discuss on the research gaps that have emerged, particularly concerning the generalizability of current generative models, which will lead to potential future research directions (Section 6.2).

6.1 Summary of Outcomes

In this research, we address a crucial challenge in deep learning: out-of-distribution generalization. In contrast to machines, humans demonstrate remarkable ability in acquiring and utilizing generalizable knowledge. Therefore, the underlying motivation throughout our study is to understand the mechanisms and principles under the acquisition and exploitation of such generalizable knowledge.

A characteristic of human knowledge is its systematicity, which enables its independent reuse regardless of other factors of variation. Inspired and motivated by this, our research proposes that only by conditioning on images before and after the transformation caused by a target mechanism, can we learn disentangled knowledge about this mechanism, based on the causal theory. This proposal has been validated in experiments on 2D transformation learning, as detailed in Chapter 3. Additionally, Chapter 4 expands upon this finding to a broader scale, by experimenting on different datasets, different mechanisms and different deep learning architectures. The results also validate our hypothesis consistently. Building on this proposal, we introduce "InterpretNet", a novel architecture that emulates the human hypothesis-verification process in perception, by explicitly exploiting the acquired knowledge. InterpretNet exhibits some interesting properties, especially its ability to classify hand-written digits, even if it was trained only on black-and-white noise images. However, recognizing its inefficiency in computation due to its image-level analysis, our exploration pivots towards the embedding-level knowledge acquisition and exploitation. Chapter 4 proposes implicit knowledge exploitation via innovative integration of regression loss with contrastive learning, and further utilization of pre-trained image-text model for transfer learning. Our experiments suggest that both methodologies effectively learn disentangled knowledge and thus enhances the ability to better discriminate samples with potential influence by the target mechanism, as demonstrated by the model's remarkable representational capabilities. Finally, we devise PCExpert based on these principles in Chapter 5, and apply this architecture in real-world challenging tasks.

One of the primary contributions of this research lies in its application of causal theory on explicit disentanglement of a specific target concept or mechanism, presented as the form of generalizable knowledge. This methodology facilitates the application of regression loss in general self-supervised representation learning, and provides solid explanation to its effect on model performance. Another important contribution of our research is the introduction of regression loss. While previous studies [134], [179] have investigated transformation estimation as pretext tasks for self-supervised learning, we illustrate further that image-text contrastive learning can be considered as a variant of regression loss, based on the similarity of model structures and underlying principles (in Chapter 4). With this novel perspective, image-text contrastive learning can be aligned with generalizable knowledge learning on the basis of causal theory. The alignment lays a foundational framework for future studies in language-based multi-modal contrastive learning. Last but not least, the introduction of the PCExpert architecture in Chapter 5, which sets a new benchmark in state-of-the-art performance, serves as a compelling validation of our theoretical framework.

Despite the current limitation of applicability in real-world tasks for the hypothesis-verification process in the architecture of InterpretNet, we have identified some potential research directions, as discussed in the subsequent section, in which the design of this approach might have implications.

6.2 Looking Forward

Learning of Metrical Variables

In this research, the acquisition of generalizable knowledge about concepts with continuum values (or *metrical* variables, such as rotation, *etc.*) is conducted by conditioning on pairs of images. On the other hand, image-text models pre-trained with contrastive learning can learn generalizable knowledge about concepts with discrete values (or *categorical* variables, such as an elephant, *etc.*). If learning of metrical variables can also benefit from large-scale, text-based contrastive learning, models could achieve more descriptive representations of the physical world, leading to advancements in tasks that reply on deep understanding of spatial and temporal dynamics. Recent studies in video-text contrastive learning [224], [225] provide foundational framework in this direction. However, a closer examination of the pre-training datasets indicates that the textual descriptions are still categorical, such as "Move the red box next to the yellow ball." Such descriptions only provide qualitative signals, and also fail to establish grounding in the temporal dimension. We argue that for video-text contrastive learning to effectively acquire knowledge of metrical variables, the textual descriptions needs to be quantitatively and temporally detailed. This requirement potentially increases the cost of dataset production, and even make the annotation of real-world data infeasible in some scenarios. We propose using game engines, such as Unity [67] and Unreal Engine [68], for the creation of synthetic datasets in virtual environments, as a plausible solution to this challenge.

Top-Down Signalling

At the time of writing, we are witnessing the remarkable capabilities emerging from large language models, notably GPT-4 [226]. In this context, we briefly discuss the gap between these models and human cognitive capabilities, and try to scratch the surface in exploring key elements essential to bridge the gap. Current research [7], [155], [227], [228] shows that modern deep learning models, GPT included, are still limited in their generalizability beyond the training distribution. The discussion on experimental findings in Section 3.4.2 indicates that while the models demonstrate generalizability beyond the training semantic space, it is still essentially constrained in the bounds of in-distribution learning. Current deep learning models still predominantly rely on recognition of patterns from past data, which differs fundamentally from humans, whose interpretation of things is grounded in our understanding of *concepts*, as indicated by the demonstration of Figure 4.8 in Section 4.4.1. Contemporary models of cortical activity suggest the presence of top-down predictive processes, alongside bottom-up pattern recognition [229]–[231]. Based on the theories on grid cells [232], [233], we hypothesize that effective top-down predictions may require the involvement of algebraic symbol manipulation. Inspired by the hypothesis-verification process of InterpretNet, we argue that a key missing element in current large language models is the capability for targeted guidance and discrete expectation (instead of probabilistic estimation) through top-down signalling. The ability of these models to apply knowledge in a top-down, algebraic manner could be a critical indicator of their evolution towards artificial general intelligence.

Bibliography

[1] Z. Shen, J. Liu, Y. He, *et al.*, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[3] A. Barbu, D. Mayo, J. Alverio, *et al.*, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

[5] Y. Zhao, Y. Wu, C. Chen, and A. Lim, "On isometry robustness of deep 3d point cloud models under adversarial attacks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1201–1210, 2020.

[6] B. M. Lake, "Compositional generalization through meta sequence-to-sequence learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] D. Hupkes, V. Dankers, M. Mul, and E. Bruni, "Compositionality decomposed: How do neural networks generalise?" *Journal of Artificial Intelligence Research*, vol. 67, pp. 757–795, 2020.

[8] G. Bao, N. Zhuang, L. Tong, *et al.*, "Two-level domain adaptation neural network for eeg-based emotion recognition," *Frontiers in Human Neuroscience*, vol. 14, p. 605 246, 2021.

[9] Z. Wan, R. Yang, M. Huang, N. Zeng, and X. Liu, "A review on transfer learning in eeg signal analysis," *Neurocomputing*, vol. 421, pp. 1–14, 2021.

[10] L. Wang, P. Luc, Y. Wu, *et al.*, "Towards learning universal audio representations," *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4593–4597, 2022.

[11] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[12] M. Cordts, M. Omran, S. Ramos, *et al.*, "The cityscapes dataset for semantic urban scene understanding," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[13] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3d to 2d label transfer," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3688–3697, 2016.

[14] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 767–783, 2018.

[15] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8198–8207, 2019.

[16] M. A. Alcorn, Q. Li, Z. Gong, *et al.*, "Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4845–4854, 2019.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[18] E. Téglás, E. Vul, V. Girotto, M. Gonzalez, J. B. Tenenbaum, and L. L. Bonatti, "Pure reasoning in 12-month-old infants as probabilistic inference," *science*, vol. 332, no. 6033, pp. 1054–1059, 2011.

[19] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, pp. 18 327–18 332, 2013.

[20] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.

[21] B. Schölkopf, F. Locatello, S. Bauer, *et al.*, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[22] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[23] G. F. Marcus, *The algebraic mind: Integrating connectionism and cognitive science*. MIT press, 2003.

[24] Weisstein, Eric W, *Symmetry*, `https://mathworld.wolfram.com/Symmetry.html`.

[25] A. E. Stahl and L. Feigenson, "Observing the unexpected enhances infants' learning and exploration," *Science*, vol. 348, no. 6230, pp. 91–94, 2015.

[26] L. E. Schulz, A. Gopnik, and C. Glymour, "Preschool children learn about causal structure from conditional interventions," *Developmental science*, vol. 10, no. 3, pp. 322–332, 2007.

[27] C. Cook, N. D. Goodman, and L. E. Schulz, "Where science starts: Spontaneous experiments in preschoolers' exploratory play," *Cognition*, vol. 120, no. 3, pp. 341–349, 2011.

[28] H. Schmidt and E. Spelke, "The development of gestalt perception in infancy," *Infant Behavior and Development*, vol. 9, p. 329, 1986.

[29] E. S. Spelke, "Principles of object perception," *Cognitive science*, vol. 14, no. 1, pp. 29–56, 1990.

[30] Y. Bengio, T. Deleu, N. Rahaman, *et al.*, "A meta-transfer objective for learning to disentangle causal mechanisms," *arXiv preprint arXiv:1901.10912*, 2019.

[31] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" *International Conference on Machine Learning*, pp. 5389–5400, 2019.

[32] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382, 2021.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[35] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning," *arXiv preprint arXiv:1412.6614*, 2014.

[36] D. Arpit, S. Jastrzebski, N. Ballas, *et al.*, "A closer look at memorization in deep networks," *International Conference on Machine Learning*, pp. 233–242, 2017.

[37] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[38] R. Atienza, "Improving model generalization by agreement of learned representations from data augmentation," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 372–381, 2022.

[39] Y. Miao and W. Luo, "Improve generalization ability of cnn by data augmentation and se block in landmark classification," *2022 14th International Conference on Computer Research and Development (ICCRD)*, pp. 250–255, 2022.

[40] Z. He, L. Xie, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Data augmentation revisited: Rethinking the distribution gap between clean and augmented data," *arXiv preprint arXiv:1909.09148*, 2019.

[41] E. Battenberg, S. Mariooryad, D. Stanton, *et al.*, "Effective use of variational embedding capacity in expressive end-to-end speech synthesis," *arXiv preprint arXiv:1906.03402*, 2019.

[42] L. Qiu, P. Shaw, P. Pasupat, *et al.*, "Improving compositional generalization with latent structure and data augmentation," *arXiv preprint arXiv:2112.07610*, 2021.

[43] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation

to the real world," *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, 2017.

[44] J. Tremblay, A. Prakash, D. Acuna, *et al.*, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pp. 969–977, 2018.

[45] R. Khirodkar, D. Yoo, and K. Kitani, "Domain randomization for scene-specific car detection and pose estimation," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1932–1940, 2019.

[46] A. Prakash, S. Boochoon, M. Brophy, *et al.*, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255, 2019.

[47] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," *arXiv preprint arXiv:1804.10745*, 2018.

[48] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *arXiv preprint arXiv:1805.12018*, 2018.

[49] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Multi-component image translation for deep domain generalization," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 579–588, 2019.

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

[51] N. Somavarapu, C.-Y. Ma, and Z. Kira, "Frustratingly simple domain generalization via image stylization," *arXiv preprint arXiv:2006.11207*, 2020.

[52] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint arXiv:2104.02008*, 2021.

[53] S. Hinterstoisser, V. Lepetit, S. Ilic, *et al.*, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," *Asian conference on computer vision*, pp. 548–562, 2012.

[54] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4340–4349, 2016.

[55] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[56] *Rockstar games*, https://www.rockstargames.com/.

[57] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" *arXiv preprint arXiv:1610.01983*, 2016.

[58] Y. Zhang, P. David, H. Foroosh, and B. Gong, "A curriculum domain adaptation approach to the semantic segmentation of urban scenes," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[59] Q. Lian, F. Lv, L. Duan, and B. Gong, "Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6758–6767, 2019.

[60] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 102–118, 2016.

[61] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *arXiv preprint arXiv:1710.06924*, 2017.

[62] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

[63] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object

detection in video," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5296–5305, 2017.

[64] X. Peng, B. Usman, K. Saito, N. Kaushik, J. Hoffman, and K. Saenko, "Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation," *arXiv preprint arXiv:1806.09755*, 2018.

[65] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[66] Y. Yao, L. Zheng, X. Yang, M. Naphade, and T. Gedeon, "Simulating content consistent vehicle datasets with attribute descent," *arXiv preprint arXiv:1912.08855*, 2019.

[67] *Unity 3d*, https://unity3d.com/.

[68] *Unreal engine*, https://www.unrealengine.com/.

[69] R. Huang, H. Sun, J. Liu, *et al.*, "Feature variance regularization: A simple way to improve the generalizability of neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 4190–4197, 2020.

[70] F. Bordes, R. Balestriero, Q. Garrido, A. Bardes, and P. Vincent, "Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning," *Transactions on Machine Learning Research*, 2023.

[71] Y. Gao, Y. Qu, C. Cui, and D. Han, "Improving the generalization via coupled tensor norm regularization," *arXiv preprint arXiv:2302.11780*, 2023.

[72] N. Bacanin, M. Zivkovic, F. Al-Turjman, *et al.*, "Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application," *Scientific Reports*, vol. 12, no. 1, p. 6302, 2022.

[73] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," *International Conference on Machine Learning*, pp. 1282–1289, 2019.

[74] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[75] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International Conference on Machine Learning*, pp. 448–456, 2015.

[76] R. Cakaj, J. Mehnert, and B. Yang, "Spectral batch normalization: Normalization in the frequency domain," *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2023.

[77] A. S. Morcos, D. G. Barrett, N. C. Rabinowitz, and M. Botvinick, "On the importance of single directions for generalization," *arXiv preprint arXiv:1803.06959*, 2018.

[78] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," *arXiv preprint arXiv:1710.05468*, vol. 1, no. 8, 2017.

[79] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Knowledge and Information Systems*, vol. 63, pp. 2585–2619, 2021.

[80] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[81] J. C. Ye and J. C. Ye, "Generalization capability of deep learning," *Geometry of Deep Learning: A Signal Processing Perspective*, pp. 243–266, 2022.

[82] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[83] D. A. Sousa, *How the brain learns.* Corwin Press, 2016.

[84] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2107–2116, 2017.

[85] E. Wood, T. Baltrušaitis, L.-P. Morency, P. Robinson, and A. Bulling, "Learning an appearance-based gaze estimator from one million synthesised images," *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 131–138, 2016.

[86] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, “Appearance-based gaze estimation in the wild,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511–4520, 2015.

[87] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3722–3731, 2017.

[88] G. J. Stein and N. Roy, “Genesis-rt: Generating synthetic images for training secondary real-world tasks,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7151–7158, 2018.

[89] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[90] S. Bak, P. Carr, and J.-F. Lalonde, “Domain adaptation through synthesis for unsupervised person re-identification,” *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 189–205, 2018.

[91] A. Rozantsev, M. Salzmann, and P. Fua, “Beyond sharing weights for deep domain adaptation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 801–814, 2018.

[92] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, “A kernel method for the two-sample-problem,” *Advances in Neural Information Processing Systems*, pp. 513–520, 2007.

[93] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[94] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” *International Conference on Machine Learning*, pp. 1180–1189, 2015.

[95] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[96] S. Lee, E. Park, H. Yi, and S. Hun Lee, "Strdan: Synthetic-to-real domain adaptation network for vehicle re-identification," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 608–609, 2020.

[97] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018.

[98] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10 285–10 295, 2019.

[99] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in Neural Information Processing Systems*, pp. 1640–1650, 2018.

[100] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6778–6787, 2019.

[101] Y. Li, X. Tian, M. Gong, *et al.*, "Deep domain generalization via conditional invariant adversarial networks," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018.

[102] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. C. Kot, "Domain generalization for medical imaging classification with linear-dependency regularization," *arXiv preprint arXiv:2009.12829*, 2020.

[103] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[104] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[105] S. Hu, K. Zhang, Z. Chen, and L. Chan, "Domain generalization via multidomain discriminant analysis," *Uncertainty in Artificial Intelligence*, pp. 292–302, 2020.

[106] X. Jin, C. Lan, W. Zeng, and Z. Chen, “Style normalization and restitution for domaingeneralization and adaptation,” *arXiv preprint arXiv:2101.00588*, 2021.

[107] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, “Adversarial invariant feature learning with accuracy constraint for domain generalization,” *arXiv preprint arXiv:1904.12543*, 2019.

[108] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, “Learning to generate novel domains for domain generalization,” *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 561–578, 2020.

[109] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?” *Advances in Neural Information Processing Systems*, vol. 33, pp. 512–523, 2020.

[110] L. Wan, R. Liu, L. Sun, H. Nie, and X. Wang, “Uav swarm based radar signal sorting via multi-source data fusion: A deep transfer learning framework,” *Information Fusion*, vol. 78, pp. 90–101, 2022.

[111] D. Joshi, V. Mishra, H. Srivastav, and D. Goel, “Progressive transfer learning approach for identifying the leaf type by optimizing network parameters,” *Neural Processing Letters*, vol. 53, no. 5, pp. 3653–3676, 2021.

[112] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[113] B. Ay, B. Tasar, Z. Utlu, K. Ay, and G. Aydin, “Deep transfer learning-based visual classification of pressure injuries stages,” *Neural Computing and Applications*, vol. 34, no. 18, pp. 16 157–16 168, 2022.

[114] L. Alzubaidi, M. Al-Amidie, A. Al-Asadi, *et al.*, “Novel transfer learning approach for medical imaging with limited labeled data,” *Cancers*, vol. 13, no. 7, p. 1590, 2021.

[115] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, “Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping,” *arXiv preprint arXiv:2002.06305*, 2020.

[116] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[117] E. Ben-David, N. Oved, and R. Reichart, “Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 414–433, 2022.

[118] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., pp. 3045–3059, 2021.

[119] E. J. Hu, yelong shen, P. Wallis, *et al.*, “LoRA: Low-rank adaptation of large language models,” *International Conference on Learning Representations*, 2022.

[120] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[121] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.,” *Journal of machine learning research*, vol. 11, no. 12, 2010.

[122] S. Chen, E. Dobriban, and J. H. Lee, “Invariance reduces variance: Understanding data augmentation in deep learning and beyond,” *arXiv preprint arXiv:1907.10905*, 2019.

[123] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

[124] A. Laishram and K. Thongam, “Automatic classification of oral pathologies using orthopantomogram radiography images based on convolutional neural network,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. Regular Issue, pp. 69–77, 2022.

[125] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir, “3d common corruptions and data augmentation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18 963–18 974, 2022.

[126] X. Hao, Y. Zhu, S. Appalaraju, *et al.*, “Mixgen: A new multi-modal data augmentation,” *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 379–389, 2023.

[127] F. Muratore, T. Gruner, F. Wiese, B. Belousov, M. Gienger, and J. Peters, “Neural posterior domain randomization,” *Conference on Robot Learning*, pp. 1532–1542, 2022.

[128] T. Dai, K. Arulkumaran, T. Gerbert, S. Tukra, F. Behbahani, and A. A. Bharath, “Analysing deep reinforcement learning agents trained with domain randomisation,” *Neurocomputing*, vol. 493, pp. 143–165, 2022.

[129] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, “Spatial transformer networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 2017–2025, 2015.

[130] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, “3d object detection with pointformer,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7463–7472, 2021.

[131] M. Adimoolam, S. Mohan, G. Srivastava, *et al.*, “A novel technique to detect and track multiple objects in dynamic video surveillance systems,” *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. Regular Issue, pp. 112–120, 2022.

[132] Y. Zhang, C. Wang, X. Wang, W. Liu, and W. Zeng, “Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2613–2626, 2022.

[133] V. N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, and V. Lepetit, “Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6771–6780, 2022.

[134] L. Zhang, G.-J. Qi, L. Wang, and J. Luo, “Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data,” *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2555, 2019.

[135] X. Wang, D. Kihara, J. Luo, and G.-J. Qi, "Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning," *arXiv preprint arXiv:1911.09265*, vol. 2, 2019.

[136] F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen, "Weakly-supervised disentanglement without compromises," *International Conference on Machine Learning*, pp. 6348–6359, 2020.

[137] J. A. Weyn, D. R. Durran, and R. Caruana, "Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 9, e2020MS002109, 2020.

[138] K. K. Verma and B. M. Singh, "Deep multi-model fusion for human activity recognition using evolutionary algorithms," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 2, pp. 44–58, 2021.

[139] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input cnn-gru based human activity recognition using wearable sensors," *Computing*, vol. 103, no. 7, pp. 1461–1478, 2021.

[140] E. Q. Wu, P. Xiong, Z.-R. Tang, G.-J. Li, A. Song, and L.-M. Zhu, "Detecting dynamic behavior of brain fatigue through 3-d-cnn-lstm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 90–100, 2021.

[141] M. Sameer and B. Gupta, "Cnn based framework for detection of epileptic seizures," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17 057–17 070, 2022.

[142] S. T. Aung, M. Hassan, M. Brady, *et al.*, "Entropy-based emotion recognition from multichannel eeg signals using artificial neural network," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[143] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 4027–4035, 2021.

[144] M. N. Dar, M. U. Akram, R. Yuvaraj, S. G. Khawaja, and M. Murugappan, "Eeg-based emotion charting for parkinson's disease patients using convolu-

tional recurrent neural networks and cross dataset learning," *Computers in Biology and Medicine*, vol. 144, p. 105 327, 2022.

[145] X. Li, W. Zheng, Y. Zong, H. Chang, and C. Lu, "Attention-based spatio-temporal graphic lstm for eeg emotion recognition," *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2021.

[146] G. Altan, A. Yayık, and Y. Kutlu, "Deep learning with convnet predicts imagery tasks through eeg," *Neural Processing Letters*, vol. 53, no. 4, pp. 2917–2932, 2021.

[147] K. Ellis, C. Wong, M. Nye, *et al.*, "Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning," *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pp. 835–850, 2021.

[148] W. Lee and H. Cho, "Inductive synthesis of structurally recursive functional programs from non-recursive expressions," *Proceedings of the ACM on Programming Languages*, vol. 7, no. POPL, pp. 2048–2078, 2023.

[149] X. Duan, X. Wang, Z. Zhang, and W. Zhu, "Parametric visual program induction with function modularization," *International Conference on Machine Learning*, pp. 5643–5658, 2022.

[150] S. Kumar, C. G. Correa, I. Dasgupta, *et al.*, "Using natural language and program abstractions to instill human inductive biases in machines," *Advances in Neural Information Processing Systems*, vol. 35, pp. 167–180, 2022.

[151] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 539–546, 2005.

[152] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017.

[153] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[154] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[155] S. Madan, T. Henry, J. Dozier, *et al.*, "On the capability of neural networks to generalize to unseen category-pose combinations," Center for Brains, Minds and Machines (CBMM), Tech. Rep., 2020.

[156] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: `10.1007/s11263-015-0816-y`.

[157] A. J. Marcel, "Conscious and unconscious perception: An approach to the relations between phenomenal experience and perceptual processes," *Cognitive psychology*, vol. 15, no. 2, pp. 238–300, 1983.

[158] A. Tscshantz, B. Millidge, A. K. Seth, and C. L. Buckley, "Hybrid predictive coding: Inferring, fast and slow," *PLOS Computational Biology*, vol. 19, no. 8, e1011280, 2023.

[159] M. Draganov, J. Galiano-Landeira, D. Doruk Camsari, J.-E. Ramírez, M. Robles, and L. Chanes, "Noninvasive modulation of predictive coding in humans: Causal evidence for frequency-specific temporal dynamics," *Cerebral Cortex*, bhad127, 2023.

[160] C. Caucheteux, A. Gramfort, and J.-R. King, *Hierarchical organization of language predictions in the brain*, 2023.

[161] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," *International Conference on Machine Learning*, pp. 8748–8763, 2021.

[162] R. Dong, Z. Qi, L. Zhang, *et al.*, "Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning?" *The Eleventh International Conference on Learning Representations*, 2023.

[163] M. Gao, C. Xing, R. Martin-Martin, *et al.*, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1179–1189, 2023.

[164] Z. Qi, R. Dong, G. Fan, *et al.*, "Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining," *International Conference on Machine Learning*, 2023.

[165] J. Platt, “Using analytic qp and sparseness to speed training of support vector machines,” *Advances in Neural Information Processing Systems*, vol. 11, 1998.

[166] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[167] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[168] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[169] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, “Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9902–9912, 2022.

[170] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[171] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[172] J. Kang, W. Jia, X. He, and K. M. Lam, “Point clouds are specialized images: A knowledge transfer approach for 3d understanding,” *arXiv preprint arXiv:2307.15569*, 2023.

[173] Z. Wu, S. Song, A. Khosla, *et al.*, “3d shapenets: A deep representation for volumetric shapes,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

[174] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, “Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data,” *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

[175] K. Hassani and M. Haley, “Unsupervised multi-task feature learning on point clouds,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8160–8171, 2019.

[176] M. Gadelha, R. Wang, and S. Maji, “Multiresolution tree networks for 3d point cloud processing,” *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 103–118, 2018.

[177] J. Sauder and B. Sievers, “Self-supervised deep learning on point clouds by reconstructing space,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[178] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, “Spatio-temporal self-supervised representation learning for 3d point clouds,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6535–6545, 2021.

[179] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, “Self-supervised learning of point clouds via orientation estimation,” *2020 International Conference on 3D Vision (3DV)*, pp. 1018–1028, 2020.

[180] S. M. Lehar, *The world in your head: A gestalt view of the mechanism of conscious experience*. Psychology Press, 2003.

[181] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013.

[182] M. S. Gazzaniga, “The split brain revisited,” *Scientific American*, vol. 279, no. 1, pp. 50–55, 1998.

[183] R. E. Nisbett and T. D. Wilson, “Telling more than we can know: Verbal reports on mental processes.,” *Psychological review*, vol. 84, no. 3, p. 231, 1977.

[184] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.

[185] S. Dehaene, *Consciousness and the brain: Deciphering how the brain codes our thoughts*. Penguin, 2014.

[186] B. Partee *et al.*, “Compositionality,” *Varieties of formal semantics*, vol. 3, pp. 281–311, 1984.

[187] T. Raj, F. Hanim Hashim, A. Baseri Huddin, M. F. Ibrahim, and A. Hussain, “A survey on lidar scanning mechanisms,” *Electronics*, vol. 9, no. 5, p. 741, 2020.

[188] J. Behley, M. Garbade, A. Milioto, *et al.*, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9297–9307, 2019.

[189] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra, "Self-supervised pretraining of 3d features on any point-cloud," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10 252–10 263, 2021.

[190] H. Sun, Y. Wang, X. Cai, X. Bai, and D. Li, "Vipformer: Efficient vision-and-pointcloud transformer for unsupervised pointcloud understanding," *arXiv preprint arXiv:2303.14376*, 2023.

[191] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9782–9792, 2021.

[192] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19 313–19 322, 2022.

[193] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 604–621, 2022.

[194] W. Wang, H. Bao, L. Dong, *et al.*, "Image as a foreign language: Beit pretraining for vision and vision-language tasks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19 175–19 186, 2023.

[195] R. Girdhar, A. El-Nouby, Z. Liu, *et al.*, "Imagebind: One embedding space to bind them all," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15 180–15 190, 2023.

[196] H. Bao, W. Wang, L. Dong, *et al.*, "Vlmo: Unified vision-language pre-training with mixture-of-modality-experts," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 897–32 912, 2022.

[197] L. Jing, E. Vahdani, J. Tan, and Y. Tian, "Cross-modal center loss for 3d cross-modal retrieval," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3142–3151, 2021.

[198] L. Li and M. Heizmann, "A closer look at invariances in self-supervised pre-training for 3d vision," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 656–673, 2022.

[199] Z. Li, Z. Chen, A. Li, *et al.*, "Simipu: Simple 2d image and 3d point cloud unsupervised pre-training for spatial-aware visual representations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 1500–1508, 2022.

[200] J. Kang, W. Jia, and X. He, "Toward extracting and exploiting generalizable knowledge of deep 2d transformations in computer vision," *Neurocomputing*, vol. 562, p. 126 882, 2023.

[201] Y.-C. Liu, Y.-K. Huang, H.-Y. Chiang, *et al.*, "Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining," *arXiv preprint arXiv:2104.04687*, 2021.

[202] H. Zhou, X. Peng, J. Mao, Z. Wu, and M. Zeng, "Pointcmc: Cross-modal multi-scale correspondences learning for point cloud understanding," *arXiv preprint arXiv:2211.12032*, 2022.

[203] Y. Wu, J. Liu, M. Gong, *et al.*, "Self-supervised intra-modal and cross-modal contrastive learning for point cloud understanding," *IEEE Transactions on Multimedia*, 2023.

[204] G. Hess, A. Tonderski, C. Petersson, L. Svensson, and K. Åström, "Lidarclip or: How i learned to talk to point clouds," *arXiv preprint arXiv:2212.06858*, 2022.

[205] J. Zhang, R. Dong, and K. Ma, "Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip," *arXiv preprint arXiv:2303.04748*, 2023.

[206] C. Xu, S. Yang, T. Galanti, *et al.*, "Image2point: 3d point-cloud understanding with 2d image pretrained models," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 638–656, 2022.

[207] R. Zhang, Z. Guo, W. Zhang, *et al.*, "Pointclip: Point cloud understanding by clip," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552–8562, 2022.

[208] Z. Wang, X. Yu, Y. Rao, J. Zhou, and J. Lu, "P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 388–14 402, 2022.

[209] M. Rong, H. Cui, and S. Shen, "Efficient 3d scene semantic segmentation via active learning on rendered 2d images," *IEEE Transactions on Image Processing*, 2023.

[210] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[211] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.

[212] R. Zhang, Z. Guo, P. Gao, *et al.*, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," *Advances in Neural Information Processing Systems*, 2022.

[213] Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann, "Disn: Deep implicit surface network for high-quality single-view 3d reconstruction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[214] N. Ravi, J. Reizenstein, D. Novotny, *et al.*, "Accelerating 3d deep learning with pytorch3d," *arXiv preprint arXiv:2007.08501*, 2020.

[215] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *International Conference on Learning Representations*, 2018.

[216] B. Tran, B.-S. Hua, A. T. Tran, and M. Hoai, "Self-supervised learning with multi-view rendering for 3d point cloud analysis," *Proceedings of the Asian Conference on Computer Vision*, pp. 3086–3103, 2022.

[217] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

[218] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[219] S. Qiu, S. Anwar, and N. Barnes, “Geometric back-projection network for point cloud classification,” *IEEE Transactions on Multimedia*, vol. 24, pp. 1943–1955, 2021.

[220] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, “Rethinking network design and local geometry in point cloud: A simple residual mlp framework,” *International Conference on Learning Representations*, 2021.

[221] G. Qian, Y. Li, H. Peng, *et al.*, “Pointnext: Revisiting pointnet++ with improved training and scaling strategies,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 192–23 204, 2022.

[222] Y. Tang, X. Li, J. Xu, *et al.*, “Point-lgmask: Local and global contexts embedding for point cloud pre-training with multi-ratio masking,” *IEEE Transactions on Multimedia*, 2023.

[223] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[224] H. Xu, G. Ghosh, P.-Y. Huang, *et al.*, “Videoclip: Contrastive pre-training for zero-shot video-text understanding,” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6787–6800, 2021.

[225] Z. Wang, Y. Zhong, Y. Miao, L. Ma, and L. Specia, “Contrastive video-language learning with fine-grained frame sampling,” *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, Eds., pp. 694–705, Nov. 2022.

[226] OpenAI, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.

[227] B. M. Lake and M. Baroni, “Human-like systematic generalization through a meta-learning neural network,” *Nature*, pp. 1–7, 2023.

[228] S. Yadlowsky, L. Doshi, and N. Tripuraneni, “Pretraining data mixtures enable narrow model selection capabilities in transformer models,” *arXiv preprint arXiv:2311.00871*, 2023.

[229] K. Kveraga, A. S. Ghuman, and M. Bar, “Top-down predictions in the cognitive brain,” *Brain and cognition*, vol. 65, no. 2, pp. 145–168, 2007.

[230] R. Weidner, J. Krummenacher, B. Reimann, H. J. Müller, and G. R. Fink, "Sources of top–down control in visual search," *Journal of Cognitive Neuroscience*, vol. 21, no. 11, pp. 2100–2113, 2009.

[231] L. Melloni, S. van Leeuwen, A. Alink, and N. G. Müller, "Interaction between bottom-up saliency and top-down control: How saliency maps are created in the human brain," *Cerebral cortex*, vol. 22, no. 12, pp. 2943–2952, 2012.

[232] M. Lewis, S. Purdy, S. Ahmad, and J. Hawkins, "Locations in the neocortex: A theory of sensorimotor object recognition using cortical grid cells," *Frontiers in neural circuits*, vol. 13, p. 22, 2019.

[233] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, "A framework for intelligence and cortical function based on grid cells in the neocortex," *Frontiers in neural circuits*, vol. 12, p. 121, 2019.