Controlling Latent Representation for Image Generation and Few-shot Classification

March 2022

Paulino Cristovao

Controlling Latent Representation for Image Generation and Few-shot Classification

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2022

Paulino Cristovao

Abstract

The performance of machine learning models depends on the quality of data representation. It is stated that good representation is the one that makes subsequent tasks easier. Representation learning defines the problem of learning good representation that is aligned with the target downstream task. Existing approaches have demonstrated an excellent ability to produce good representations for the downstream task. However, such models offer limited to no control over the latent representations. Variational Autoencoders and Generative Adversarial Network have revealed the potential to relieve this problem. However, explicit control of the latent representation is still an unsolved problem in computer vision. The latent representation learning of such learning approaches might not have the desired properties for the learning problem. Thus, a fundamental question in representation learning is how to control them for machine learning models to be a more effective problem solver. The work presented in this thesis studies the challenge of controlling latent representation space to improve the downstream task such as image generation and classification tasks.

Furthermore, the representation learning employed in this work is classified into Generative and Discriminative Models. In the case of generative learning, we propose an objective function that constrains the latent representation. The primary objective is to encourage the latent representation to have the desired property for image in-between generations. While, in the case of discriminative models, we address the challenge of generalization on unseen classes in few-shot learning. Thus, we propose an objective function that models the latent representation. The proposed objective function forces the model to learn features that well describe the images. Furthermore, we introduce good layer selection for the same task of classification. We demonstrate that transferring features from a good layer improves generalization in few-shot learning.

Controlling representation learning enables specific control of the output. In particular, we propose three different use-cases where controlling representation might improve the downstream task. First, we present a use case showing how controlling the latent representation space can generate the desired structure of images in-between generations. Then, second, we reveal how classification accuracy in a Few Shot learning model is improved when applying an objective function that models the latent representation. Finally, we introduce a new design for transferring good representations from a task-agnostic layer for the weight imprinting models. The proposed methods will help address the critical limitation of models based on representation learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning models (ML) from multilayer perceptrons (MLPs) to Deep Neural Networks (DNNs) have shown incredible accomplishment in many fields. From Computer Vision (CV), Audio, and Speech to Natural Language Processing (NLP). These models take raw input data such as image, text, audio, then pass into a series of transformation ”latent representation learning” or simply ”representation learning” and then output a probability according to a task being addressed. Such tasks can be generation or classification.

The central problem in ML models is to turn raw data into a relevant representation that makes the subsequent tasks easy. The performance of ML models primarily relies on the quality of representation learning. The representation learning carries meaningful information of the initial data, which is highly dependent on downstream tasks [3]. However, these representations are often unstructured and lack interpretation. Usually, the representation learning space does not exhibit the desired property without regularization. For that reason, for machine learning models to be more powerful, much of the time is spent designing data preprocessing pipelines and data augmentations that result in a good representation of the data.

More recently, deep learning architectures (DL) such as AlexNet [4], ResNet [5], VGG [6] and Inception [7] have been used to learn powerful semantic representation from complex data, i.e, to extract useful representation for the learning problem. Despite their success in CV, representation learning is handled differently. For example, representation learning in supervised learning models (SL) is a side effect of the objective function that the model is trained to minimize. While in self-supervised learning (SSL), the representation is learned through an implicit loss function using a pre-defined teaching signal or pseudo label. In unsupervised learning models, representation learning is learned through implicit objectives such as clustering [8].

Although these learning models produce a good representation, they suffer from the following limitations: 1) They face a trade-off between preserving as much information about the raw input data and attaining excellent properties for downstream tasks, and 2) They offer limited to no control over the latent representation. The Variational Autoencoders (VAE) and Generative Adversarial Networks (GANs) frameworks have unveiled the potential to alleviate these limitations. However, explicit control of the latent representation is still an unsolved problem in computer vision. Furthermore, the representation learning of these models might not have the desired property for the downstream learning problem. Despite these challenges, one approach that is gaining popularity to address the same subject is disentanglement. Bengio et al. [3], and Higgins et al. [9] state that good representation is the one that disentangles the underlying factors of variation. If this is justified, any lack of latent representation uncontrollability should present some learning barrier. Controlling representation learning allows machine learning models to present the desired property, which will enable us to address the learning problem.

This thesis presents use-cases demonstrating the benefits of controlling representation learning. In the first use case, we generate images in-between using representation learning based on VAE. We designed an objective function that enforces the latent representation space to generate a desired property for the task of in-between. We demonstrate that our proposed model outperforms both pixel-based and conventional VAEs models. In the second use case, we attempt to improve the generalization for novel data in the Few-Shot learning problem. The remarkable success of deep learning models is mainly due to many annotated data. However, acquiring labeled data is very expensive and time-consuming.

The performance of FSL models heavily depends on the quality of the representations learned during the pre-training phase. Nevertheless, it is unknown what characteristics are required for generalization. Drawing inspiration from information theory, we propose an objective function that models the latent representation. Furthermore, we show that cross-entropy alone is not enough for generalization. Finally, in the third use case, we show that for FSL based imprinted-weight models, good layer selection improves classification accuracy. We provide a simple design choice that yields substantial improvements over the baseline model. Our empirical results show that introducing nonlinear projection heads in-between representation and classifier improves classification accuracy. Furthermore, we demonstrate that imprinting from the task-specific layer does not provide good generalization for novel classes. Instead, we propose imprinting from the task-agnostic representation.

1.1 General Objective

This thesis presents evidence that controlling representation learning can be determinant for improving the downstream task. First, we attempt to build a model that allows having a controlled output. Then, we present two techniques that focus on representation learning that attempt to address learning from limited data. The performance of our proposed methodology and related research questions are assessed employing several quantitative and quality experiments involving distinct tasks.

1.1.1 Specific Objectives

The specific objectives are described as follows:

1. The objective is to generate images in-between using latent representation learning. We introduce an objective function that constrains the latent space representation. The proposed objective function encourages the latent space to have a structure that is the desire for our downstream task. Furthermore, our objective function flats the latent manifold so that image in-between in latent space and image space is possible.

2. The objective is to improve generalization on novel tasks where a single or a few samples are available. Unfortunately, the representations learned with the cross-entropy loss function do not generalize well on unknown data. One reason is that it does not model the latent representation. Therefore, we borrow from information theory an objective function that models the latent variables.

3. The second objective is the same as the previous one. However, the approach is different. The goal is to improve generalization by promoting a good layer selection. For transfer learning, particularly domain adaptation, selecting good representations is essential for generalization on novel tasks.

1.2 Motivation

Our interest in representation learning models primarily comes from the increasing interest in using them for creative applications of machine learning models. In particular, this work is motivated by the desire to control representation learning to have the desired structure aligned with our downstream task. Innovative and creative research on representation learning has been demonstrated, especially in computer vision, music, speech, and text generation.

For example, performing arithmetic operations on the latent variables, then decoding the datapoint allow us to generate novel and creative output. Representation learning is powerful since it encodes the representations that best describe the data in low dimensions. For example, training a model in one dataset can transfer the knowledge to other target domains. Moreover, the representations learned may generalize across different learning problems.

Although deep learning models have achieved remarkable performance on various tasks, still it is seen as a black box. Understanding and interpreting latent representation is one step toward general machine learning. Studying representation learning will open diverse creative ideas for addressing many learning problems. Below, We present more examples of creative work on deep learning models based on latent representation.

1. **Image Generation.** Manipulating representation learning allows us to synthesize novel data with attractive properties. For example, [9, 10, 11], generate novel examples of faces which are not present in the training distribution. The synthesized example shows mixed features, such as gender, age, race transformation. This model finds application in scenarios with limited availability of data of certain races, gender, or category.

2. **Image Style Transfer.** [12] introduced a style transfer model which aims at rendering the semantic content of an image in a different style. This algorithm allows synthesizing of new images, such as a simple image of a cat with the style of the Mona Lisa painting. Recently, much creative work has been published [13, 14, 15].

3. **Deep Dream.** Provides visualization of different patterns learned by the neural network. It enhances the patterns of the images, allowing to generate artistic images.

4. **Sketch-drawing.** This work [16] presents a creative idea, where the authors use latent representation to construct stroke-based drawings of objects. Another example is transforming a photo image into a sketch-drawing [17, 18].

5. **Anime Generation.** These models [19, 20, 21] explore the latent space to generate anime images. In the anime industry, every single frame needs to be drawn. Using deep learning to generate the frame alleviates this need. For instance, we can interpolate two nearby frames to generate the in-between frame.

6. **Video Generation.** Deep Fake [22] has recently gained attraction. It is an algorithm that creates images and videos that humans can barely distinguish are authentic. The target video contains the effects of the source video. For instance, we can generate a video of a celebrity speaking in another language or acting in a different movie.

7. **Text Generation.** Latent representation is also used in sequential data to generate novel data, mainly text, and audio. For example [23, 24, 25] showed that manipulating the latent representation produce diverse and well-formed sentences. The model is capable of generating sentences based on style and topic.

8. **Music Style Transfer.** In deep generative models, we can generate novel data and change the property of the initial data. For example, [26] showed that by manipulating the latent space of the music data, we could change the pitch, dynamics, and instruments of music. The authors give an example of style transfer from classical music to jazz.

1.3 Thesis Organization

This thesis is divided into four main chapters:

1. Theoretical Background - This chapter provides comprehensive information on the background work on representation learning.

2. Controlling Representation Learning - We propose a method to control representation learning for addressing the downstream task.

3. Regularizing Representation Learning for Image In-between Generation - We show a use case of controlling the representation to have a controlled output.

4. Variational Information Bottleneck for Few-shot Learning Problem & Good layer Selection for FSL based imprinted models - We demonstrate that improving the quality of representation learning is associated with a good generalization of the FSL classification task. We also introduce a new method to transfer knowledge in imprinted-weight models.

Chapter 2

Background

Before discussing data augmentation techniques, it is crucial to frame the context of the problem and consider what makes image classification and generation such a difficult task in the first place. For instance, In discriminative learning, the model has to overcome issues related to different viewpoints, lighting conditions, overlapping objects, occlusion, background, scale, color, and more. On the other hand, the task of data transformation is to expand the initial data to learn to be invariant to such modifications.

There has been growing interest in representation learning. In this chapter, we assess how current works use representation learning to improve the performance of the downstream task. Several strategies have been explicitly used in machine learning to improve latent representation. The general objective is to minimize the training error and possibly improve the downstream task performance. We categorize existing work on how representation learning is improved. We summarize the three categories in the following sections as shown in Figure2.1.

We first survey data (transformation), which is the first step machine learning practitioners engineer the data. The objective is to introduce different variants of the data. Distinct data transformation (variants) allows the model to expand its hypothesis space. The model is trained to be invariant to various viewpoints. In particular, we review the automatically learned features using data augmentation, data hallucination, auxiliary data, and neural style transfer. Next, we review how representation learning helps improve the downstream task such as classification and generation tasks. First, in discriminative models, we explore how different objective function helps to improve the latent representation across various tasks, including supervised learning and self-supervised learning.

The representation learning in standard machine learning models is seen as a side effect of the classifier. The representation learning in self-supervised models attempts to avoid the burden of human annotation. The representation is improved by inducing a model to be invariant to a predefined task or trained in the contrastive loss. Finally, in the generative model, we analyze how using the prior knowledge, and conditional label information improves the latent representation. Finally, we evaluate some regularization techniques that have been shown to perform well. We study how the size of the network help to get better representation. In particular, we focus on how Convolutional Networks allow feature selection and knowledge transfer across different tasks or domains.

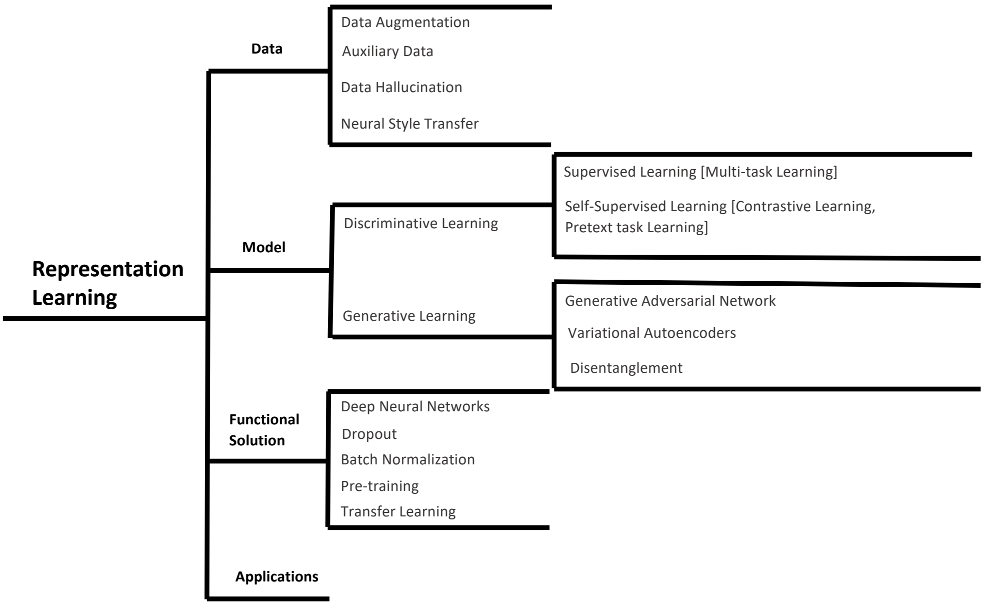


Figure 2.1: Representation Learning methods based on the focus of each method

2.1 Data

Data pre-processing is a set of methods that transform the data to make it easy for the machine learning model to interpret and extract relevant information or features for the learning problem. The transformations can be performed by domain expertise, which infers which attributes or features are essential for the learning problem. The brute-force method is also used; in this case, the model is trained and hoping to select the suitable representation.

One of the challenges in machine learning is to train models to be invariant to data transformation. One way to encourage invariance of a model is to set transformation, which expands the original training samples. There are two ways to expand the size of the dataset, namely data augmentation and data hallucination. The first extends the data size by applying some techniques such as rotation, cropping, grayscale to RGB, etc. The former makes use of the available data to generate similar data. The generated images are the hallucination of the initial data in different scenarios.

Data transformation can improve representation by expanding the image space of the dataset. The assumption is that ample image space allows the extraction of more discriminative features. Representation learning is defined as a set of methods that allow machine learning models to be fed existing datasets and automatically discover the valuable representation for the learning problem. Having more discriminative factors will allow more effective learning of relevant features. We review in detail the two mentioned approaches to improve the representation using data transformation.

2.1.1 Data Augmentation

Machine Learning models have performed well on many tasks. However, It is generally accepted that best-performing models heavily rely on a large amount of data. For instance, the well-known and widely used image dataset among the machine learning community ”ImageNet” has millions of annotated data.

One reason for the need for extensive data is to avoid overfitting. Overfitting is when the model performs well at the training phase. However, the model performs poorly on test time. Regrettably, many fields have no access to big data. For instance, the data is available in small quantities in the medical area. Also, the data might be imbalanced in other areas, meaning that we have large labeled data for one class and few samples for others. Data Augmentation (DA) refers to a set of techniques that enhance the size and quality of the data aiming to get better representation learning. A comprehensive survey of DA for the image is presented in the work of Shorten et al. [27].

The earlier examples illustrating the effectiveness of DA in deep learning come from simple image transformation such as geometric and color. For instance, image flipping vertically and horizontal and RGB color to grayscale images conversion. These transformations allow the models to encode relevant and invariances present in the data. If the model can encode and learn invariant transformations, it is said that the model is robust. Robustness is essential for machine learning models to generalize across various domains and tasks. For example, one application of the powerful model is in autonomous cars, where the vehicle has to classify and predict objects in different conditions.

Other forms of data augmentation include cropping, translation, mixing images, random erasing, and noise injection. Recently, several augmentation techniques have been proposed. Among them, we will review how augmentation works. Zhang et al [28]. introduced mixup model , the authors suggest a mixup training. The network model is trained with two mixed training samples and their corresponding labels. The Mixup model addresses the issue of memorization in the neural network. Memorization is when the model is overconfident about the relationship between the label information and the input features. DeVries et al. [29] introduced the Cutout technique, which is another regularization that randomly masks out a squared region of the input image at the training phase. This technique can be seen as injecting noise into the raw input image. The model is encouraged to utilize better the full context of the image, rather than capturing specific features of the raw input image.

Yun et al. proposed the Cutmix [1] technique. The technique is conceptually similar to cutout; however, Cutmix suggests masking a region of the image and filling it with another image from a different class. The authors claim that cutout or injecting noise in the image removes some information about the input data that can improve representations. The Cutmix augments the input pair of images, and the label information is mixed. Conceptually, it shares the similarity with the Mixup method by interpolating the two samples and the label. Seo et al. suggested the Selfmix [30], instead of mixing two masks from different images (classes), the authors propose to mask from the same image. Conceptually, the objective is similar to the Cutout and Cutmix, and the model is trained not to learn the specific structure from the input data but to learn the whole structure of the data.

These data augmentation techniques have improved the representation of deep neural net-

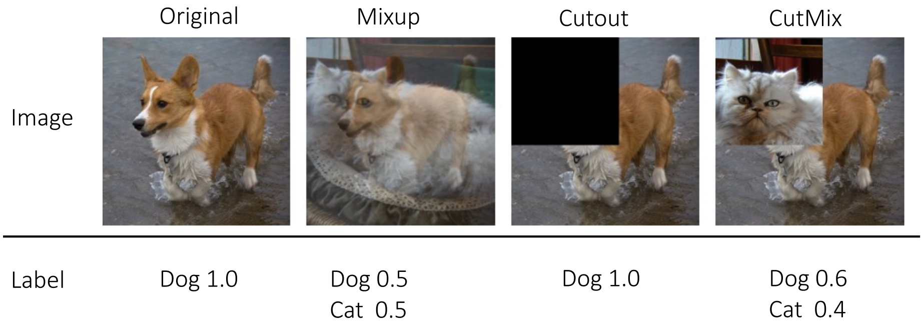


Figure 2.2: Data augmentation techniques. Image taken from [1].

works and led to better generalization. The general idea is to avoid the model being overconfident about the specific structure of the data. In Figure 2.2, we show the visual transformation of the image and how the label is defined. We might want to add different features or structures to the training data in some scenarios. However, the above technique might not be possible.

2.1.2 Auxiliary Data

The advantages of Auxiliary Data can be described in two ways.

1. It helps expand the initial dataset's size by using auxiliary data related to the learning problem. The model is trained with the extra data that do not belong to the learning problem but shares standard features. For instance, we can train the model to classify digits. We use the digits dataset and alphabet data as auxiliary data at the training phase. Another example is illustrated in Figure 2.3, the model is trained to classify lion vs. tiger. At training time, we feed images of lions and tigers (training data) and images of other animals such as leopards, horses, cats, and more (auxiliary data).

Using additional but related data allows for incorporating novel features and structures not present in the initial data. This characteristic helps to address the issue of memorization and the lack of generability. The choice of the auxiliary dataset has to be carefully chosen. It is essential to find the data that share similarities with your target data domain. The example used above is ideal for showing that the target domain and auxiliary data share commonalities. We assume that the underlying structure of the alphabet and digits is similar. Incorporating novel structures or features is relevant, in particular in domain adaptation.

2. The auxiliary dataset is used to find a good initialization for the target downstream network. For instance, we use unlabeled auxiliary data to pre-train a model contrastively and then transfer the learned representations. The auxiliary data does not necessarily have to be related to the downstream task. For example, Wang et al. [31] used auxiliary data for improving the representation for the object tracking problem in a video. The authors pre-trained a stacked Autoencoder using a contrastive objective function to learn generic image features.

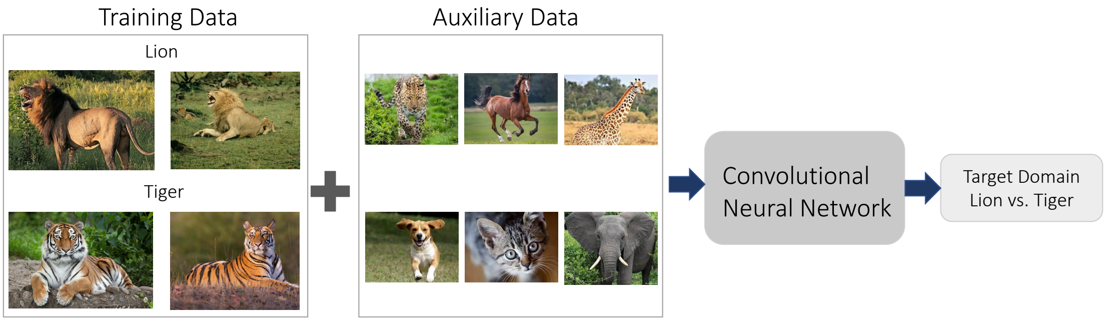


Figure 2.3: Auxiliary Data

Why Auxiliary Data are helpful?

Finding auxiliary data is based on the assumption that auxiliary data should belong to the environment of the related target problem in some way and should be helpful to learn features that might be discriminative or help understand the structure of the data. Auxiliary data approach conceptually can be related to Auxiliary task discussed in Figure 2.2.1. However, the notion of similar or related auxiliary data is not known. For instance, Caruana et al. [32] define two tasks to be similar if they use the same feature to make a decision. While Baxter et al. explain theoretically that related learning tasks share a common inductive bias [33].

Despite these, we assume that using auxiliary data will expand the image space. Furthermore, the model will capture variations from the auxiliary data that might be useful for the learning problem. It is not always possible to find auxiliary data that share a similar underlying representation. Therefore, an alternative approach is to hallucinate novel data from the initial data.

2.1.3 Data Hallucination

Humans can learn from a single example and connect it to novel samples related to the learned category. Perhaps humans can also quickly imagine or hallucinate novel objects from different views. The human capability to recognize novel objects is said to benefit from prior knowledge. By contrast, current machine learning models are data-hungry. As a result, the performance accuracy largely depends on the size of the data. One way to expand the scope of the data is to incorporate this ability to hallucinate data. Data Hallucination is an intuitive alternative to extend the data set and enhance the diversity of the data without recollecting new examples. The hallucinated data has to belong to the same initial data distribution. For instance, if we have a training set that is composed of birds in the trees. Then, we can hallucinate the birds flying or in another environment that is possible to find the birds.

Few approaches to generate hallucinated or imaginary novel data are based on the Generative models. For instance, Dixit et al. [34] proposed to augment the training data by using the feature descriptors (pose and depth) as a guide to synthesizing novel features. This method allows the model to learn how the images are transformed when the pose changes. Wang et al. [35] hallucinate novel examples by training a GAN framework in meta-learning settings. The generator takes input noise and sample train data and generates augmented examples. Then the classifier is trained on

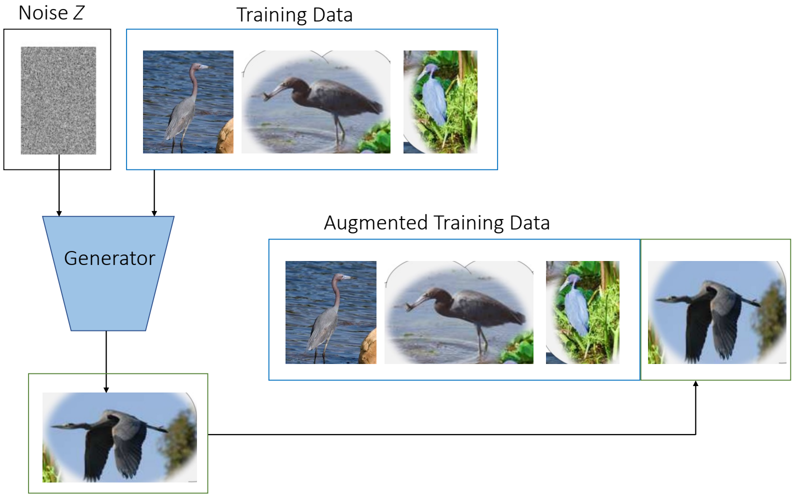


Figure 2.4: Data Hallucination: Given an image, novel concept can be visualized.

both the initial training set and hallucinated sets. The primary intention is to use hallucinated data as additional training examples to build better classifiers. Hallucinated-based approaches attempt to augment the data without any external side information [36]. The aim is to incorporate diversity into the training set. For example, the initial data and hallucination might share some modes of variation. Borrowing from [35], Figure 2.4 shows the conceptual idea of hallucination (imaginary data).

Data hallucination in image space is compelling. However, it assumes that intra-class variances can generalize well to novel classes; besides, it does not model the latent distribution [37]. One way to overcome this limitation is to hallucinate the feature space. Hariharan and Girshick [38] proposed to sample two representations z_1 and z_2 belonging to the same class representing a smooth transformation. Then, given a new class latent representation, a variance from z_1 to z_2 is applied.

2.1.4 Neural Style Transfer

Neural Style Transfer is one example of data hallucination. It is somewhat similar to color space transformation. However, the image generated by Neural Style Transfer encodes different textures and artistic styles. The Neural Style Transfer can apply an artistic style to the image without changing the high-level semantic content. The main objective is to jointly minimize the distance between the content and a style representation of the two images using a neural network [39]. One of the earlier works on neural style transfer was proposed by Gatys et al. [40], further enhanced by Johnson et al. [2], by replacing the per-pixel loss, which does not capture the perceptual differences between output and ground-truth images with a perceptual loss function.

For the purpose of Data Augmentation, earlier, we defined data augmentation as adding diversity and variation into the initial training set. Thus, Neural Style Transfer adds a substantial variation to the data. Neural Style is analogous to color space; besides, it extends lighting variations and enables the encoding of different textures and artistic style [27]. Then, the network model is optimized to be invariant to such variations in the data.

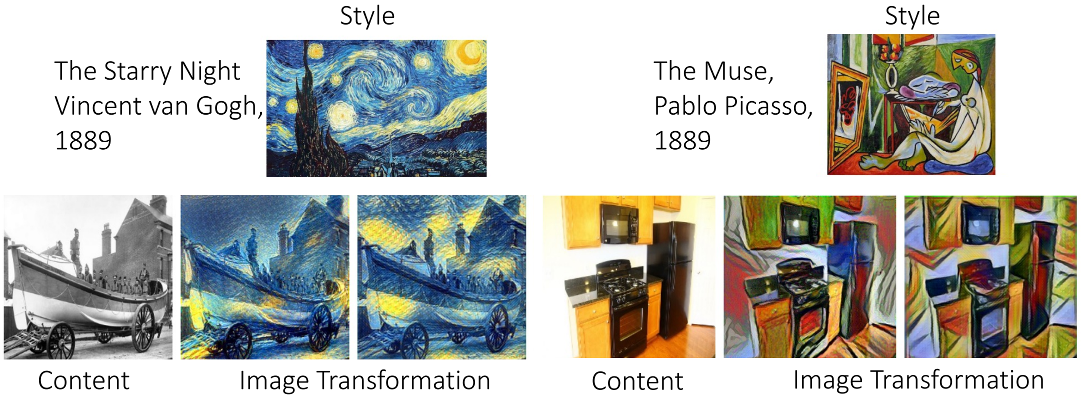


Figure 2.5: Style and content reconstructions in Neural Style Transfer [2]

An excellent example of the real-world application of Neural Style Transfer is in a self-driving car. Image scenes can be transferred from one domain to another, for instance, night-to-day and vice-versa, winter-to-summer, or rainy-to-sunny scale. Let us assume that we aim to train a classifier to discriminate between wolf and husky. Our dataset is composed of wolves with a snow background. The model trained on such a dataset will associate the snow to the wolf category. This behavior is not desired for generic classification. If we test on the image of a husky with a snow background, the model can classify it as a wolf. Using Neural Style Transfer as a data augmentation technique can be ideal to avoid such association. Zheng et al. [39] proposed to use eight different style images as a palette to augment the original training set.

These data transformation techniques aim to create models that are invariant to the particular transformation of the inputs. Modern approaches to computer vision exploit these transformations to improve representations. These methods are helpful, particularly in scenarios where the data is limited.

2.1.5 Handcrafted feature vs learned feature

For decades, machine learning models required careful engineered and domain expertise to select good features from input data. Good feature selection based on keypoint localization or keypoint descriptor is referred to as handcraft feature. A standard approach used for feature selection is Bag of Visual Words model (BoVW) [41, 42, 43]. In contrast to conventional handcrafted features [44, 45], learned features refers to those that automatically merge from complex model such as neural networks. The key aspect of learned features from large-scale data [46] is that humans do not design these features and they can be learned and transferred to a variety of downstream tasks [47, 48, 6].

Although, automatically learned features show good performance in downstream tasks. Some researchers have proposed combining both methods for generalization capability. For instance, Georgescu et al. [49] proposed to combine automatic features learned by convolutional neural networks and handcrafted features computed using a bag-of-visual-words model. The representations from the neural network and bag-of-visual-words are concatenated. The model is trained to solve a facial recognition problem. The authors show that facial features are better separated using the combination of both approaches.

There is a long debate on which one is better. However, we believe that the use of one depends on the downstream task. Learning features using neural networks have proved to achieve good performance in complex tasks; however, it requires a large amount of data, which might be hard to collect in some scenarios. On the other hand, handcrafted feature models are fast, playing an essential role in the medical field [50, 51, 52, 53]. In many pattern recognition applications, it is known that predictions should be constant or invariant under one or more transformations of the input variables. The before-mentioned transformations create significant changes in the raw data, expressed in terms of the intensities at each pixel, yet should give rise to the same output from the classification or prediction [54].

2.2 Models

2.2.1 Discriminative Models

Based on the training label, we review the visual feature learning methods that attempt to improve representation learning in this section.

Supervised Learning Models

The most common form of machine learning is supervised learning (SL). In SL, the instances or training examples X are given with known labels y (corresponding ground truth label). The main goal in SL is to learn a target function that can predict the classes. The objective function is obvious; we want to minimize the number of misclassifications on the training data as expressed in Equation 2.1:

$$Loss(\theta) = \min \sum_i L(f_\theta(X_i), Y_i) \tag{2.1}$$

Figure 2.6 shows a supervised learning pipeline. The first step is collecting the dataset and apply the data pre-processing and define the training and validation set. The second phase is algorithm selection (model). The choice of the algorithm is a critical step. Once the model is defined, the last phase is selecting the classifier, and a linear classifier is often used. The classifier maps from unlabelled instances to classes [55]. The accuracy is based on the percentage of correct classification divided by the total number of predictions. Pos-training if the error rate is poor, several factors can be examined, such as the size of the dataset. Possibly a large set is required for generalization, learning rate, and model capacity.

Representation Learning in Supervised learning. Unlike SL model, where the classifier is optimized to minimize the number of misclassification in the training set. As explained by Bengio [3], the objective of representation learning is far removed from the ultimate aim of classification or prediction. Good representation has to be general and robust. In SL, the representations are often treated as side effects of the classification task, rather than explicitly attempted [56]. Scrutinizing the role of intermediate representations and extending the depth is an interesting open question.

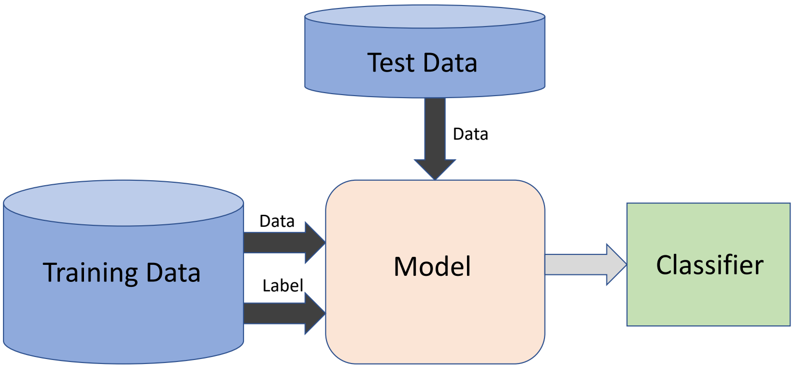


Figure 2.6: Supervised Learning Model

A critical perspective on representation learning in SL is based on ”Depth” and ”Abstraction”. Depth is the premise that deep network architectures allow feature re-use as discussed in section 2.3.1. Feature re-use enables different parts of the network (representations) to be shared in high-level layers. Abstraction (as discussed in section 2.3.1) has the assumption that general concepts are invariant to local changes such as data transformation. More abstract representations also mean that the model captures a larger data manifold of the observations. This property allows the model to improve its classification or prediction capability.

However, ”Depth” and ”Abstraction” will not be enough with the cross-entropy loss. Therefore, the loss function does not model the representation. Instead, it attempts to obtain relevant features of the label Y in the latent representation Z. However, it does not care how the encoder $f(X)$ encodes the features. It is known that the encoder might choose to encode features that are not relevant for the downstream task and generalization. To get better representation learning, we train models in a self-supervised setting. For instance, we can train using contrastive and pretext task techniques.

Issues of Representation Learning in Supervised learning Models. The main issue in SL is dealing with the dataset. In order to obtain better representation at training time, a considerably annotated dataset is required. However, labeling the data is time-consuming and expensive. Moreover, we lack enough data for some specific domains, such as a medical dataset. Also, we require domain expertise for labeling the data. In some scenarios, the experts might not reach the same conclusion about labeling an image or object. Besides, to achieve better representation, we also require deeper models with large numbers of layers. These models are hard to train and need a careful tuning of the parameters to converge and take a considerable amount of time, i.e., the training time might take days.

Another burden of SL is that the objective function, often a cross-entropy loss does not explicitly model the representations. The loss attempts to minimize the error between the feature extractor activations and the true label. As a result, the encoder has the freedom to choose what features to encode, some features might not contribute to the downstream task and consequently for generalizing well. One strategy used to avoid these burdens is to train the model in self-supervised learning. The model is trained with large unlabeled data and few supervised data. Other techniques include Contrastive learning and the Pretext tasks approaches. However, these approaches rely on the type of augmentation used in contrastive learning and the pre-defined task. An explicit objective function that models the representation for leveraging the learned representation is still required.

Self-Supervised Models

The performance of supervised models somewhat depends on the model capacity and the size of the training set. For ML models to perform well on various learning problems, large-scale annotated data is required for training the neural networks. Different architectures have been developed [5, 6] to increase the model capacity. Besides, more labeled data has been collected [57]. However, collecting and labeling large-scale datasets is time-consuming and often expensive. For instance, one of the most used datasets in vision problem ImageNet contains nearly 1.3 million labeled images with 1000 classes. Each image was handly labeled by human workers.

To relieve these burdens and to remove humans from the loop, a self-supervised method is gaining considerable attention of researchers [58, 59, 60, 61, 62, 63, 64]. Much of the attention comes from the fact that SSL methods learn features without using manually labeled data, i.e., the labels are generated automatically.

Self-supervised learning (SSL) is a form of supervised learning that aims to uncover the underlying structure of the data without explicit labels. Compared to SL methods, SSL does not require human annotation. A standard way to learn visual features from the vast unlabeled data is to use a pseudo label (P_i) that is defined based on the attribute present in the images. For instance, the pseudo label can be a pretext task such as predicting the rotation of the image [65], or solving the zigsaw puzzle [64]. While the network is trained to solve this task, the features are learned through this process [59]. The objective function is defined as in Equation 2.2:

$$Loss(\theta) = \min \sum_i L(f_\theta(X_i), P_i) \tag{2.2}$$

Conceptually, the SSL objective function is similar to SL, and the objective is to minimize the error between the actual output X_i and the pseudo label P_i. After finishing an SSL training, the feature representations learned through the process are often transferred to a target downstream task, usually with a smaller network and fewer annotated data.

Representation Learning in Self-supervised Models.

The objective function of SSL models does not explicitly model the representation learning. However, this approach is widely used to improve representation learning due to the free labels present in the data. For instance, the pretext task approach improves the representation learning by training a network to be invariant to attributes such as geometric transformation [65], jigsaw puzzle (identifying correct position of the scrambles patches in image [66], object segmentation [67], image inpainting [68] and colorization transformation [69, 70].

Another method to improve representations is to train the model using a contrastive learning approach. The contrastive objective function forces the model to group representation from similar classes and pushes away representation from different categories. The SSL approach explicitly defines desirable invariances through pretext tasks and contrastive learning approaches. The core intuition is that the model will learn meaningful features representation implicitly that can be useful for the downstream learning problem. We review contrastive learning and pretext task approach in the next section.

Contrastive Learning

SSL approach is gaining attraction because it avoids the cost of vast annotated data and alleviates the burden of human annotation. It aims to improve the latent representation by training a neural network model with a pseudo label and then uses the learned representation for the target learning problem. One foremost SSL approach in computer vision and natural language processing is contrastive learning. The general objective of contrastive learning is to learn feature representations.

1.With Label Information. The objective is to learn to group representations from the same class (x, x^+) and push away representations from different categories (x, x^-). To achieve this objective a similarity metric is used to measure how close two representations are [71]. The contrastive objective function favors a small distance for samples from the same class and large distances for pairs of different classes.

For example, a standard objective function used in n computer vision based contrastive learning methods is a triplet loss. The semantic feature representation is learned using an anchor (original image) x, a positive image (often augmented original image) x^+, and a negative image (image from another class) x^-. The representation is obtained using an encoder network. The network architecture has three networks that encode the images (anchor, positive and negative). The weight is shared among the three networks to reduce the number of parameters. Figure 2.7 shows an example of contrastive learning using label information to improve representations.

First, we have an original image with label two, and then we augment it (rotated slightly) to form a positive pair. The negative pair includes an original image and another image from a distinct class, label three. The network is trained to maximize the similarity between representations from the same category and minimize the representations from other classes as demonstrated in Equation 2.3.

$$TripletNet(x, x^-, x^+) = \begin{bmatrix} \|Net(x) - Net(x^-)\|^2 \\ \|Net(x) - Net(x^+)\|^2 \end{bmatrix} \tag{2.3}$$

The objective function is expressed as in Equation 2.4, and the details can be find in [56].

$$Loss(d_+, d_-) = \quad \|d_+, d_- - 1\|_2^2 = const.d_+^2 \tag{2.4}$$

where

$$d_+ = \quad \frac{e^{\|Net(x)-Net(x^+)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}} \tag{2.5}$$

and

$$d_- = \quad \frac{e^{\|Net(x)-Net(x^-)\|_2}}{e^{\|Net(x)-Net(x^+)\|_2} + e^{\|Net(x)-Net(x^-)\|_2}} \tag{2.6}$$

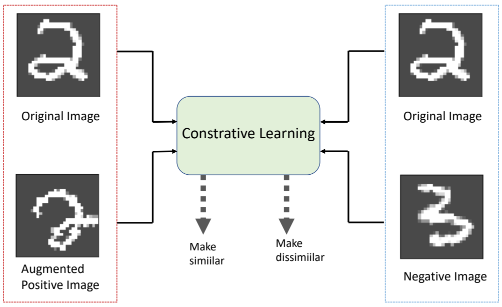


Figure 2.7: The basic intuition behind contrastive learning approach: Group representations from the same class closer and push away representations from dissimilar classes [71].

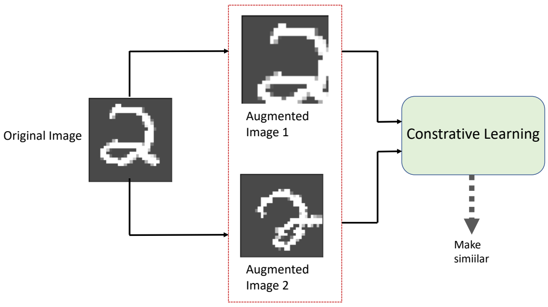


Figure 2.8: The original image is augmented to form two views. The model is optimized to make the representations agree using contrastive learning.

2.Without Label Information. Making representations agree using contrastive learning. Contrastive learning provides a way to uncover good feature representations without explicit labels. The idea is to learn an encoder that produces similar output vectors when given two different augmentations of the same image or views of the same object [72]. Figure 2.8 demonstrates the basic intuition of constrastive learning when the the label information is discarded. We obtain the two ciews of the original image and the encoder is optimized to approximate the representations of the two views.

Recently, new approaches conceptually similar to triplet loss have been introduced. Chen et al. [70] introduce $SimCLR$ a contrastive framework for contrastive learning of visual representations. $SimCLR$ architecture has two networks with shared parameters. Each network takes the patch (cropped image) of the same image with different augmentation (x_i, x_j). A network with a projection head is trained to maximize the agreement using contrastive loss. The $SimCLR$ performs extensively on the supervised task due to nonlinear transformation between the representations and the contrastive loss.

Using a similar approach Grill, et al. [73] proposed $BYOL$, a self-supervised model that aims to learn powerful representations using two networks. One of the networks is called an online network and seeks to predict the other network's parameter (target network). Each network takes different views (augmentation) of the same image. Therefore, $BYOL$ is optimized to minimize the similarity loss between the representations of both networks. Also, other works that follow this approach include [74, 75, 76, 77, 66].

Different from the above methods that augment the image to create different views of the same image, Oord et al. [78] proposed to combine a powerful autoregressive model with contrastive loss. The model learns to extract useful representations from high-dimensional time series data by predicting the future in latent space.

Models based on contrastive learning have closed the gap between SL and unsupervised models. However, there are a few challenges to contrastive approaches. One of the significant challenges is choosing what type of augmentation to apply since the contrastive models are sensitive to the choice of augmentation. Another critical issue is determining the number of negative samples in a batch. Another approach that relies less on data augmentation for the same learning problem is training a model with a pre-defined task.

Pretext

Learning high-level representation from annotated data is still the main reason for the success of deep learning models. Manually specifying the label has contributed to advancement on several real-world applications such as image classification, objection detection, image segmentation, and natural language processing tasks such as language translation and text classification. However, the supervised approach for learning features from labeled data has almost reached its saturation due to large annotated samples [71]. That's why self-supervised methods such as contrastive learning play a vital role in deep learning models since they do not require annotated data to learn good feature representation.

Recently, a pretext task training technique has been proposed to learn useful feature representation from the data without human annotation. The pretext task training solves pre-defined tasks in which the pseudo labels are automatically generated based on image attributes such tasks include image rotation [65], jigsaw puzzle [66], image inpainting [68], and colorization transformation [69, 70]. The labels (pseudo label) for the pretext task are automatically created based on properties of images [59]. The objective of pretext task is not to maximize the performance on the pretext task, but rather to learn rich and transferable features for downstream tasks [79].

Another objective of pretext tasks is to learn image representations that are semantically meaningful and invariant to such pre-defined pretext tasks. Figure 2.9 shows a general pretext task pipeline training. The first phase is the pre-training; the network is trained on a large unlabeled dataset. In the second phase, we use the learned knowledge (representation) for the downstream tasks. Lastly, the performance on the downstream task is used to assess the quality of learned representation in the first stage (pretext training).

The pretext task model have shown to learn good representation learning for distinct downstream task such as self-supervision task (we discuss in section 2.2.1 and Few Shot task (we discuss in section2.4.1).

Gidaris et al. [65] trained a ConvNet to recognize a 2D image using a geometric transformation rotation (image rotation with 0, 90, 180, and 270 degrees) as a pretext. For the model to recognize the applied transformation (orientation), it must understand the concept of objects such as location in the image, their type, and pose. Another pretext task model was introduced by Mehdi and Paolo [64], the authors trained a network to solve jigsaw puzzles for solving the problem of image representation without human annotation. The network takes image tiles as input and explicitly predicts the index of each tile. The goal is to learn to identify each tile as an object part and how

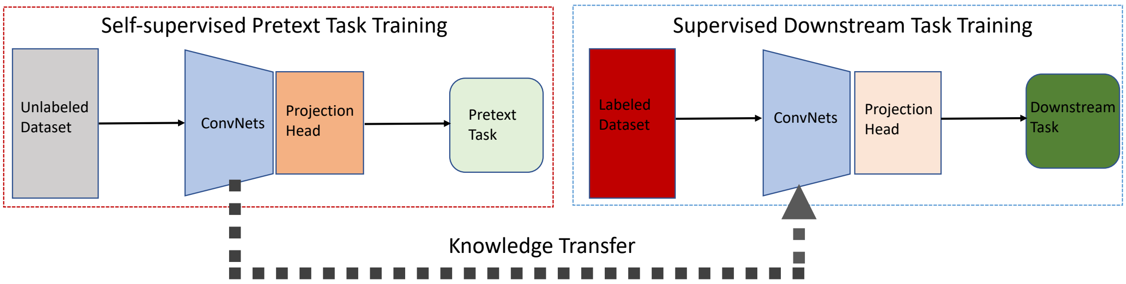


Figure 2.9: Pretext task pipeline

they are assembled.

Using a similar approach, Misra and Lauren [66] trained the model to solve jigsaw puzzles for learning pretext task invariant representation. The input image is transformed into four patches (images) which are then randomly passed to two networks. The model objective function forces the patches to be similar to the transformed version of the exact image representation and different from other images. The task of predicting the transformation provides a powerful signal for the further supervision learning task. Furthermore, it enables the model to learn general-purpose feature representations that are useful for different visual learning problems. The features learned with these approaches are then transferred to object recognition and object detection to downstream tasks. Their performance has been demonstrated to be competitive with supervised learning models.

The self-supervised methods (Contrastive and Pretext task) make use of the supervised objective function. However, we don't care about the final performance in the pre-defined task. Instead, we are interested in training the model to capture an excellent semantic structure that can generalize well for the various learning problems.

Representation Learning with Multi-Tasks

Humans can learn multi-tasks and use the knowledge learned from one task to improve the performance of another one. Multi-task learning (MTL) [80] is a paradigm in machine learning that strives to learn from multiple problems. The primary objective is to use relevant information (shared information) from different tasks to improve the generalization. The standard Machine Learning approach is optimized to reduce the error rate on a single problem. Therefore, the information from other related tasks that might be helpful to improve the generalization is ignored.

Standard MTL are generally categorized into main architectures:hard [81] and soft [82] parameter sharing [83].

1. Hard parameter sharing - The hidden layers of the ConvNet are shared among all tasks, and on top, there is a specific fully-connected layer and classifier for each learning problem, as shown in Figure 2.10. The feature extractor (hidden layers) learns task-agnostic features of the data. At the same time, the task-specific fully-connected layer learns task-specific. Hard parameter sharing performs well if the source domains and target tasks are related [83]. The advantage of this approach is that the more tasks we are training, the network captures representation that can generalize well across different learning problems. For instance, Long and Wang [84] introduce

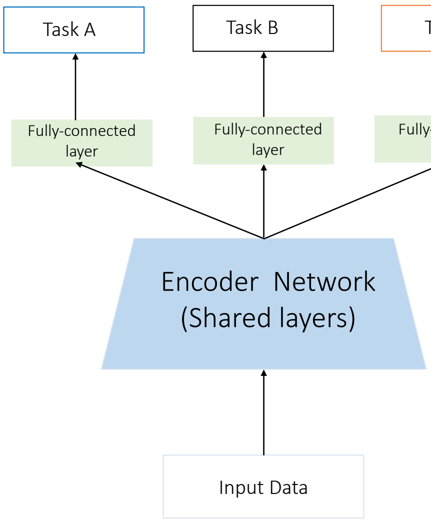


Figure 2.10: Hard parameter: MTL of three related tasks with single-source domain

a deep relation network that learns a shared representation on lower layers and higher layers on top that are n-task-specifics layers. Using a similar network concept, Kendall et al. [85] proposed weighting the loss between tasks. The authors propose multiple loss functions to learn various problems simultaneously. Their model learns to balance the weights between learning tasks. Li et al. [86] suggested a random loss weight method to normalize the weights after every iteration.

2. Soft parameter sharing - The hidden layers and parameters are task-specific. The weights are shared among the networks, as shown in Figure 2.11. The network is encouraged to output similar embeddings. For instance, in order to encourage similar representations, Duong et al. [82] employs Euclidean distance for regularization. While [87] introduce a cross-stitch network. The model has two networks, and the representation is shared by learning a linear combination of input activation maps.

Recently, there has been a growing interest in MTL [80, 88] since we can explore the shared representations from different tasks. A reason to learn common feature representation from multiple tasks is to explore the expressiveness of various tasks. Besides, single-tasks might not generalize well due to the data constraint. MTL alleviates the problem of scarce labeled data by learning from different sources. Suppose we have a shortage of annotated data in our target learning problem. Then, with the help of MTL, we train the model with other source domain data simultaneously. As a result, MTL can achieve better performance compared with single-task learning. In addition, data from different domains can help to learn more robust and universal representations.

However, the MTL method is conceptually similar to transfer learning, with significant differences. In transfer learning, the objective is to improve the target task using representations from the source domain. In contrast, in MTL, there is no distinction between all tasks, and the objective is to improve the performance of all tasks [89]. The representation learning in MTL is improved based on the following mechanism listed by [33]:

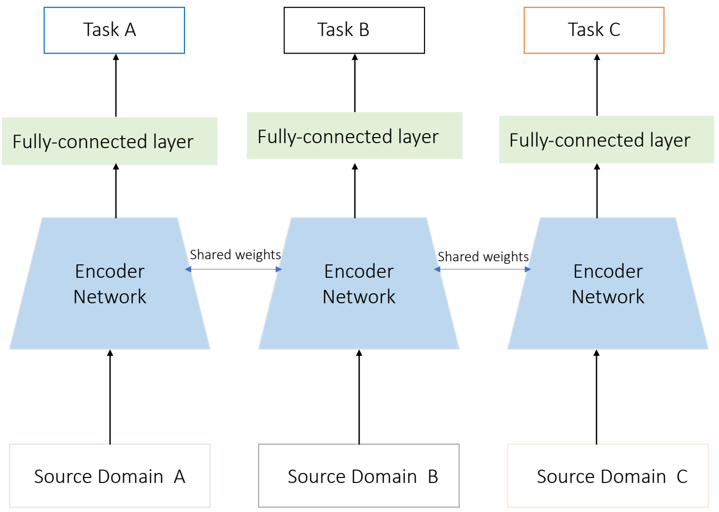


Figure 2.11: Soft parameter: MTL of three tasks with three source domains

Implicit data augmentation. Primarily, by using data from different but related problems, we increase the sample size at the training phase. As a result, the model learns a good representation for the primary task that ideally ignores the data-dependent noise and generalizes well. The model learns the data invariance from the different learning problems. This feature enables the model to obtain a better representation.

Attention focusing. Due to noisy and insufficient data, it can be challenging for the model to discriminate salient and irrelevant features for the learning problem. MTL helps the model to focus its attention on relevant features, as features from other tasks might provide additional evidence for the contribution or not of those features.

Representation bias. MTL training might bias the model to favor representations that are shared among tasks. This property is helpful for generalization on novel, but related categories since the hypothesis space are sufficiently large.

Regularization. If the features from multiple tasks are complimentary, MTL acts as a regularizer. It minimizes the risk of overfitting by introducing inductive bias.

MTL has been used successfully across various applications, from computer vision [90] ,natural language processing [91] and speech recognition [92]. Although the excellent performance in some learning problems, classic approaches in MTL suffer from degradation due to competition between tasks, learning from multi-tasks might penalize the primary task [93]. Learning with auxiliary tasks addresses this shortcoming.

Representation Learning with Auxiliary tasks

Auxiliary tasks enable the model to learn shared representation between learning tasks in order to improve the main task. In some scenarios, we are only interested in the performance of a single task. Therefore, we use the auxiliary task to learn a representation that might be useful for our target task. However, selecting the relevant auxiliary task is challenging. Below are the commonly used methods for selecting the task.

Related Task The example of using a related task is mentioned in 2.1.2. Using auxiliary data at the training phase that is related to our target domain helps to improve representation. Another approach is to work with single data with many attributes. For instance, Caruana et al. [32] predict different characteristics of the road as auxiliary tasks for predicting the steering direction in autonomous driving cars.

Adversarial The dataset for the target domain might be insufficient or unavailable. However, large annotated data from opposite learning problems is available. We train the model using the adversarial loss aiming to maximize the training error. The objective is to encourage the model to learn representations that cannot distinguish between domains [33].

The multi-task learning and auxiliary task approaches allow the model to learn a good representation shared between tasks or hidden layers for the main learning problem. However, the representation is learned implicitly for most approaches, considering that the loss function employed does not explicitly model the latent representation. Therefore, a more explicit objective function that models the latent representation is needed. For instance, a model that learns the discriminative features and the data structure.

2.2.2 Generative Models

Among others, machine Learning is divided into a generative and discriminative model. First, the discriminative model learns the classifier given the observations. Then, it takes input pixels and maps them to the labels. In contrast, the generative model tries to solve the more general problem of learning a joint probability $p(X, Y)$, or just $p(X)$ if there is no label information.

A generative model learns the data distribution. It has shown an enormous ability to generate novel data with highly realistic content such as images, texts, and audio. The generative model describes how the data is generated in terms of a probabilistic model. For example, Figure 2.12, illustrate a pipeline of a generative model. For instance, if we have data that includes birds, the generative model might capture correlations among classes or features close to birds, such as trees and nets. These features (classes) will be nearby in the representation space.

In contrast, the discriminative model does not care about the correlations between features or classes. Instead, it attempts to learn the difference between classes and draw the boundary in the data space. Applications of the generative model include creating fake portraits of celebrity images, generating artistic style images, music generation such as the composition of novel songs, and recently researchers have attempted to finish the tenth symphony of Beethoven.

Training the generative models is said to be needed. Since we can make the generative models can understand the world represented in the data [94]. This section introduces the generative model

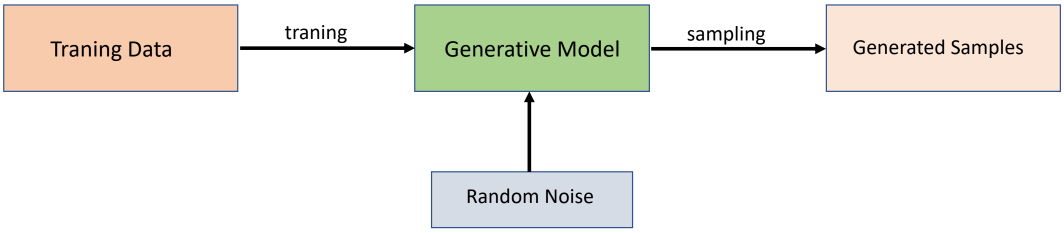


Figure 2.12: Generative Model Framework

frameworks and core mathematical formulation that will enable us to structure the base model this thesis leans on. Among the Generative models, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) deserve special attention.

GANs and VAE are two distinct approaches for learning latent space representation, with each owning its characteristics. VAEs are great for learning well-structured latent spaces, where specific directions encode a meaningful axis of variation in the data. In contrast, GANs generate images that can probably be highly realistic, but the latent space they come from may not have as much structure, and continuity [95].

Generative Adversarial Networks (GAN)

The Generative Adversarial Network was first introduced in 2014 by Ian Goodfellow (GAN) et al. [94]. It is an algorithmic that employs two neural networks: generative (G) and discriminative model (D). Fugure 2.13 shows the GANs framework. The generator network learns the data distribution defined by a prior on input noise variable $p(z)$ which is then mapped to data space $G(z; \theta_g)$. While the discriminator network $G(x; \theta_d)$ estimates the probability that a sample came from the training set rather than from the generator network G. Where (θ_g, θ_d) are the parameters of the networks.

The generator G is simultaneously trained to minimize $\log(1 - D(G(z)))$. The two networks D and G play the adversarial minimax game with value function $V(G, D)$ [94]. More formally, the following expression gives the minimax game.

$$\min_G \max_D V(D, G) = \mathbb{E}_{z \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.7}$$

GANs latent space enables the generation of reasonably realistic artificial images by forcing the generated images to be statistically almost indistinguishable from real ones. However, GANs latent space has less explicit guarantees of meaningful structure, and it is not continuous. Furthermore, GANs are notably difficult to train because we do not seek to minimize a function but rather find the equilibrium between the generator and discriminator.

There are many works involving improving the latent space representations of GANs. For disentangled representation learning, Chen et al. [96] proposed infoGAN, which maximizes the mutual information between a small subset of latent codes z and the observation x. Unlike GANs, InfoGAN does not use a single unstructured noise vector; it decomposes the noise vector into z and c. Where z is the source of incompressible noise, and c is the latent code that targets semantic features of the data distribution [96]. The authors proposed the following information-regularized

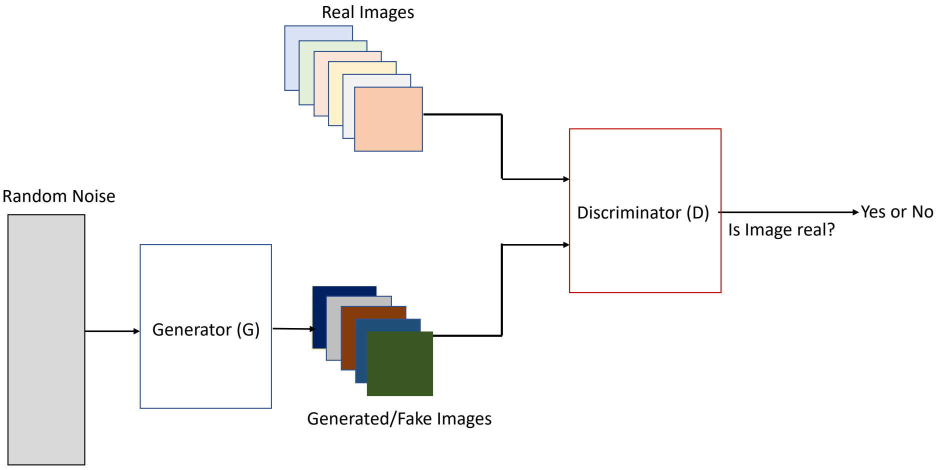


Figure 2.13: Generative Adversarial Network

minimax game:

$$\min_{G}\max_{D} V_I(D,G) = V(D,G) - \lambda I(c; G(z,c)) \tag{2.8}$$

While Karras T. et al. [10] control the image generation by re-designing the generator. The generator adjusts the style of the image at each convolution layer based on the latent code. The authors claim that the embedding produces intermediate latent space free from limitation and allows disentanglement.

Variational Autoencoders (VAEs)

The Autoencoders models (AE) consists of two neural networks, the encoder E and a decoder D network, and is trained to learn the best encoding-decoding scheme of the input data. Despite good performance on the data dimensionality reduction task, it comes with two limitations. First, once the model (AE) has been trained, there is no way to produce novel content. Second, the latent space is not regular enough (the decoder acts as Generative Adversarial Network). In order words, the objective function of AE attempts to minimize the L_2 loss as shown in Equation 2.9. However, it does not enforce a prior distribution on the latent space. Therefore AE does not allow a generative model.

$$L_{AE}(\theta,\phi) = \frac{1}{2}\mathbb{E}_{p(x)}[||x - D_\theta(E_\phi(x))||_2^2] \tag{2.9}$$

For the AE decoder to be used as a new content generator, there is a need to introduce an explicit regularization during the training process. Hence, VAEs can be seen as latent space regularization to ensure an appropriate structure (properties) that enable novel content generation. VAEs belong to the family of AE. Therefore, it can be viewed as a modern take on AE. It was first proposed simultaneously by two teams, Kingma et al. [97], and Rezende et al. [98].

Figure shows the VAEs framework Figure 2.14 The model consists of two coupled but independently parameterized networks as in AE. First, the encoder maps input data space into latent space. Then, reversely, the decoder network decodes the latent space back to its initial state. The in-

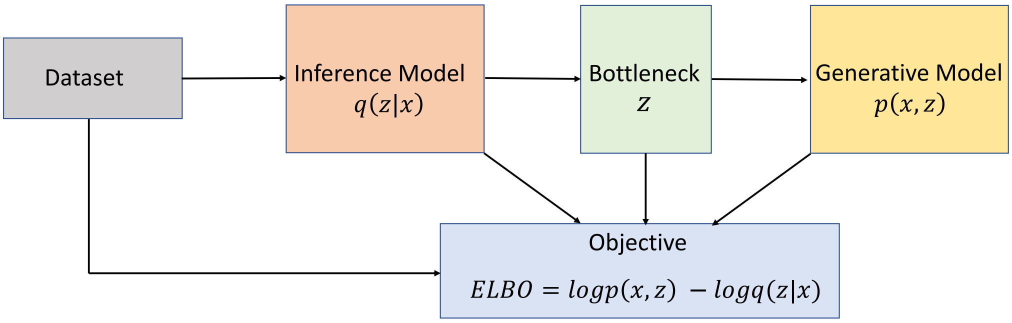


Figure 2.14: Variational Autoencoders Framework

put data is often compressed (encoded) into the lower dimension space before the decoder network increases. Thus, the phase between encoder and decoder is called a bottleneck.

The model is trained to reconstruct the initial input training set; for that reason, it learns to preserve all relevant information that describes the raw input data. Furthermore, the model's output can be controlled if an appropriate constraint (penalty) is imposed in the latent space. The training is done by optimizing the tractable evidence lower bound (ELBO) rather than directly performing maximum likelihood estimation on the intractable marginal log-likelihood as shown in Equation 2.7.

$$ELBO(\theta, \phi) = \mathbb{E}_{z \sim q_\theta(z|x)}[\log p_\phi(x|z)] - KL(q_\theta(z|x)||p(z)) \tag{2.10}$$

where: The first term measures the reconstruction error $logp_\phi(x|z)$, and the second term quantifies how well the approximate posterior $q_\theta(z|x)$ matches the prior is the prior distribution, often a unit Gaussian $p(z)$. The structure of latent space heavily depends on the prior [99]. The reconstruction quality and generation diversity is affected by the KL divergence. If too high, the model is encouraged to focus on diversity, resulting in poor reconstruction quality. Reversely, if too low, the variety is reduced since the model focus on reconstructing high-quality output. One way to balance the trade-off between reconstruction quality and diversity generation is by introducing a parameter $\beta(0 \leq \beta \leq 1)$ at the objective function.

Using similar methodology Higgins et al.[9] developed β-VAE ($\beta = 4$). The parameter β works as a form of regularizer that acts on the approximate posterior $q_\phi(x|z)$. The objective function of β-VAE imposes a stricter information bottleneck, forcing the latent codes to be more independent. Such penalization has been shown to lead to a higher degree of disentanglement. However, it is not explicit why $\beta > 1$ leads to learning latent variables that exhibit disentangled transformations for all data samples [100]. The β-VAE model allows users to exercise explicit control over KL-divergence in the objective function by controlling the value of KL-divergence [101].

Disentangle latent representation

The latent variables of some architecture have been shown to disentangle important factors of variation in the training examples, which makes such models worthwhile for representation learning [96, 9, 102]. While there is no explicit definition of disentanglement, we borrow one from Bengio et al. [3] which states that a disentangled representation should separate distinct informative factors of variations in the data. A change in a single latent factor of variation z should lead to a change in a single factor in the learned representation $r(x)$ [103].

The β-VAE [9] is a variant of the Variational Autoencoder that attempts to learn disentangled representation by imposing a more decisive penalty on the objective function. Kim and Mnih [104], Chen et al. [100], Zhao et al. [105] propose regularizing the latent space by maximizing the mutual information between latent representation z and observation x. In particular, all suggested to decompose the second term of Equation 2.10 into Equation 2.11.

$$D_{KL}(q_{\theta}(z|x)||p(z)) = I_{q_{\phi}}(x; z) - D_{KL}(q_{\theta}(z|x)||p(z)) \tag{2.11}$$

This objective function discourage entanglement of latent representations. While the InfoGAN [96] a variant of generative adversarial networks attempts to incorporate structure into representation by maximizing the mutual information between latent variables and the observation. Though not as relevant to our work, there is also recent work on discovering latent structure in an unsupervised fashion [103]. Disentangled representations have been claimed that it is an important step towards a better representation learning [3, 99]. However, a good disentangled representation depends on the end task.

2.3 Functional Solutions

Advancements in deep learning have resulted in models that require a large amount of training data and powerful hardware. As a result training these models can be a complex task. Many strategies used in machine learning are explicitly designed to minimize the test error. These strategies are known as regularization [94]. In this section, we describe some functional solutions for enhancing the training and test error. These techniques aim to improve the generalization performance of deep models.

2.3.1 Deep Neural Network Models

Data augmentation and data hallucination alone are not enough to build a good representation that can improve the generalization across the various task. Although they create more diverse data representation, it is crucial to have a network architecture to learn such complex data variation. We review the role of a deep neural network for improving representation. Another approach to enhance representation learning is to build deep neural networks architecture.

Since their introduction, Convolutional Networks (ConvNets)[106] have shown excellent outcomes at complex tasks such as classification and prediction. The most outstanding performance of ConvNets was initially proposed by Krizhevsky et al. [4] which achieved an error rate of 16.4% compared to 26.1% of the 2^{nd} place model in the ImageNet classification challenge. Since then, new alternative architectures for ConvNets have been proposed.

Recent deep network models have been shown to perform well due to various factors, such as; (1) massive data availability, (2) powerful hardware such as GPU, (3) better regularization routines such as Dropout [107, 108], and Batch Normalization [109]. The availability of large annotated data is crucial to building models that are invariant to data transformation and can capture various and significant discriminative features of the data space. Having extensive data aligned with deep networks help to improve the representations, which may lead to better generalization. Another factor that helps deep networks to perform well is the availability of powerful hardware. For example, GPU graphic cards allow training deep learning models with large batches. Training longer has been shown to improve the performance of the models. These two factors, massive data and powerful hardware aligned with better regularization techniques, have led deep learning models to address successfully various tasks.

For instance, models such as [110, 111, 112, 113, 114, 115], have been successfully applied in various tasks including image classification, image prediction, speech and audio, and natural language processing. Bengio et al. [3] state that two factors that contribute to the excellent performance of deep network models are (1) deep network models support re-use of the features, and (2) deep network models can create more abstract features at higher layers of representations.

Feature Selection

Good representation is the one that makes the subsequent task easier. In a hierarchical deep learning network, different layers of the network typically learn distinct feature representations. The representation captured by each layer shows the hierarchical nature of the features in the network. The representation of the first layer typically learns task-agnostic features such as texture, traces, colors, corners, and edges. The representation of the second layer detects particular arrangements of the edges. In contrast, subsequent (high-level) layers learn task-specific features related to the downstream task. Extended work by Zeiler et al. [116] unveils the input stimulus that excites unique feature maps of any layer in the model.

The notion of feature selection in this context means that we can extract features from different network layers. Two approaches that benefit from feature selection are transfer learning [117], which we will discuss in this section, and residual training [5, 114]. We transfer features from one layer to another, as demonstrated by He et al. [5].

This approach addresses the vanish gradient problem and answers the question, ”Does stacking more layers improve learning?”. Vanish gradient problem refers to the phenomena where the gradients become smaller during back-propagation [118]. Smaller updates during training mean more extended training and less learning since the network will stop learning at a certain point. Stacking multiple layers exposes depth as an essential dimension in neural networks. Another application is to transfer features from one model (domain) to another. In scenarios where the data is scarce, we fine-tune the model.

Fine-tuning is when we unfreeze some of the layers of the pre-trained model and jointly retrain the newly added part model (new classifier) with new target domain data available in small sizes. It slightly modifies the more abstract representations of the model being reused to make them more relevant for the learning problem [119]. Fine-tune is an effective method to avoid overfitting if the data is limited.

Abstract Features at Higher Layers

The depth of representations is of fundamental importance for many downstream tasks. The “levels” of features can be enhanced by the number of stacked nonlinear layers (depth) that transform the representation at one level into a representation at a higher level. The composition of such transformation offers the ability to learn complex functions [72]. He et al. [5] state that depth can lead to more abstract features at higher layers of representation. While Bengio et al. [3] state that abstract concepts are generally invariant to local changes of the input, also, the author mentioned that more abstract representations detect categories that cover large data manifold.

Many works have benefit from higher level of abstraction [120, 121, 5, 109, 114, 122]. Among the field which benefit from abstract concept are FSL [123, 124], in particular the domain adaptation problem [125, 126]. In these areas, the models are trained to capture the abstract structure of the training examples. The objective is to exploit the shared representations between different learning tasks and transfer relevant knowledge across other targets.

In FSL, the training and target distribution remain the same, while the target distribution is distinct in the domain adaptation setup. Representation learning is essential due to shared variation in the latent representation across tasks. Many works have shown the benefits of robust networks for image recognition and language.

2.3.2 Dropout

Dropout was first proposed by Hinton et al. [108]. The dropout regularization method zeroes out the activation values of randomly chosen neurons during training, i.e., it randomly drops or ignores neurons during the training phase. At each forward or backward pass, individual neurons are dropped out of the network with probability $1-p$. Dropout is an important regularization technique. It avoids co-dependence amongst each node during training which helps to reduce interdependence among the neurons. Besides, it forces the model to learn robust features that are useful for subsequent tasks.

One advantage of dropout is that it is computationally cheap and can be combined with other forms of regularization techniques to yield a further improvement [94]. Another advantage of dropout is that it is not limited to the type of model. It can be used in probabilistic models and discriminative models. Dropout can also be seen as adding noise to the network hidden units. The idea of adding noise to the hidden units has been used to prevent overfitting.

After dropout was introduced, many variants have followed. Wan et al. [127] proposed DropConnect that randomly drops the weights rather than the activations. Ba and Frey [128] proposed Standout which the difference is that the probability p of dropping out a unit is not constant on each hidden layer. Instead, it is computed based on the value of the weights. This approach favors small weights, i.e., if the weight is higher, the probability that the unit will be drop is significant.

Thompson et al.[129] proposed dropout values across the entire feature map (spatial dropout). Spatial dropout addresses the problem of standard dropout in which, due to pooling operation, the spatial information is lost. Spatial dropout can be seen as dropping the entire feature. This forces the model to generalize based on some other features.

2.3.3 Batch Normalization

Batch Normalization [109] is another technique that makes the training of deep network models faster and stable. It normalizes the activations by calculating each input channel's mean, zero, and standard deviation across the batch. This technique guarantees that any optimization appropriately handles the normalization. Batch normalization is often set after dense or convolutional layers to normalize the output from those layers. Batch normalization aims to improve optimization. However, sometimes the use of batch normalization makes dropout unnecessary [94].

2.3.4 Transfer Learning

We review the pre-trained model technique before we review the transfer learning approach.

Pre-training - Deep learning models require extensive data to work well on the downstream task. However, in small data settings, the models overfit, and the generalization is poor due to many parameters. Recently different techniques have been developed to train deep learning models for wide applications such as computer vision and natural language processing. The primary purpose of these techniques is their ability to alleviate the feature engineering problem [130].

One way to alleviate the need for extensive data is to use a pre-trained model. A pre-trained model refers to deep network models trained on a large-scale dataset. We transfer or fine-tune the representations to address further downstream tasks. The predominant pre-trained models such as ResNet [5], VGG [6] and [7] are trained on ImageNet dataset. ImageNet dataset is a prominent visual dataset with more than 14 million images with annotated labels. In addition, it contains more than 20000 categories. Many works have shown the effectiveness of pre-trained models due to their capability to learn abstract representations from the data.

Furthermore, the learned representations turn out to be useful not only for downstream tasks of related categories present at the source domain but even for categories outside the source domain since the learned representations are task-agnostic. Fine-tuning the model has been shown to work better than a random initialization because the model learns general image features [130, 94]. For instance, Erhan et al. [131] demonstrate that Pre-training works as some form of regularization. It minimizes variance and introduces a bias towards configurations of the parameter space that are helpful for learning. In computer vision, the pre-training tasks are not so crucial as in NLP; the encoder trained on the ImageNet dataset has been shown to encode a general-purpose representation that is useful for AI-tasks as stated by Bengio et al. [3].

Transfer features - Machine Learning models work well under the assumption that the training and testing sets belong to the same distribution. This statement is true for many applications. However, when the distribution changes, most models perform poorly. Therefore, it is necessary to collect extensive novel data and then re-train the model from scratch. This task can be expensive, especially gathering labeled data that might not be available in large amounts in every domain. Transfer learning (TF) is a method used to avoid collecting a large amount of data.

TF is another exceptional technique for enhancing generalization because it allows the reuse of the pre-trained model on a new problem. Figure 2.15 shows the TF framework. In transfer learning, we have two stages, the training or pre-trained model and the second stage is to transfer the knowledge from the pre-training to the target learning problem. Knowledge transfer has proved effective, particularly in scenarios where the training and the downstream task share statistical strength. For example, the knowledge learned at one task stage might capture the underlying factors that might help describe the data for other learning problems. We often train a network on big data such ImageNet [6], and use the learned representations (weights) as the initialization for a new classifier. First, it was intended to prevent overfitting, then to improve the performance of domain adaptation.

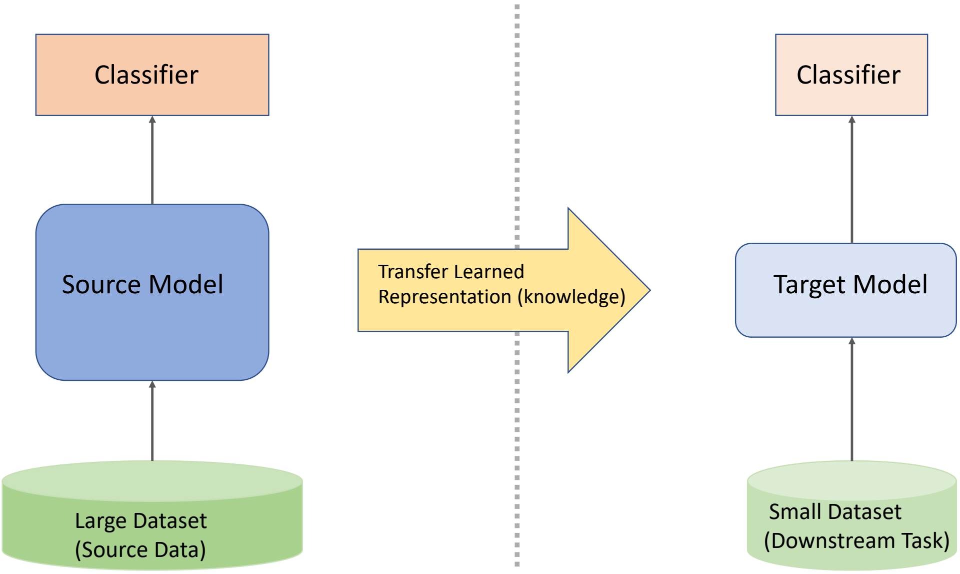


Figure 2.15: Transfer Learning Overview

In Computer vision and NLP, transfer learning is used to learn good feature representation from one domain (task) to another target, and this has been demonstrated [132, 133]. For example, if we train a model to classify cats and dogs, we can use the knowledge (representations) learned to recognize other species. In transfer learning, it is essential to know what to transfer and when to transfer. Different layers have different representations, one with more task-specific and others task agnostic. What to transfer is related to the downstream tasks. Nevertheless, some shared features might be helpful to improve the representation of the learning problem.

When to transfer, ask whether the source and target domain have shared representations. Although, for the most natural dataset, we have the assumption that the underlying representations are similar. It has been shown that even the source domain and target domain in image space do not share similar features. Nevertheless, the representation learned from different problems still helps to improve the downstream representation task.

Recently, transfer learning approaches have been applied in many real-world applications. For example, Zoph et al. [134] used transfer learning for improving a neural translation model. Likewise, Wang et al. [135] successfully applied to speech recognition, document classification, and sentiment analysis. Raghu et al. [136] and Huynh et al. [137] also applied transfer learning for medical images. There has been a comprehensive survey on transfer learning for deep learning in machine learning literature in which show the benefits of it [117, 138, 139].

2.4 Techniques that use learned representation

The above sections explain different models and techniques that attempt to improve the latent representation for the downstream task. There are classes of models in which make use of these representations in a different context. Such models include those which attempt to improve generalization in low data regimes. In this section, we discuss the relevant works related to this thesis.

2.4.1 Few Shot Learning

Learning quickly is a hallmark of human intelligence, whether it involves identifying objects from a few examples or quickly learning novel skills after just a few experiences [140]. Machine learning models have been successful in the high-data regime but often perform poorly in the low-data regime. In some scenarios, it is not always possible to gather a large volume of data due to privacy, safety, ethics; in the worst-case scenario, the data is available in little quantities such as medical and drug discovery data. Therefore, learning effectively from a small training sample is vital. Few-Shot Learning (FSL) techniques is proposed to address this problem. FSL help to relieve the burden of collecting detailed labeled data by using prior knowledge (representations) to rapidly generalize novel tasks with only a few samples with supervised information. In addition, it is impossible to add new classes in SL learning models once the training is done. Therefore, it is required to re-train the model or finetune.

FSL methods allow the addition of novel classes with no re-train. When one novel class is presented, this is referred to as one-shot learning, and when few unknown instances are presented, we refer as Few Shot learning. The training strategy often contains n classes and k labeled samples, divided into support and query sets. The core idea in the FSL paradigm is to use the prior representations as a baseline. This prior knowledge carries the assumption that different classes might share modes of variation of the data. In other words, the FSL approach for classification learns to compare new examples in a learned metric space using distance metrics such as triplet loss, cosine similarity, nearest neighbor, etc.

Most contemporary work on FSL take one of the categories: 1) Meta-learning-based approaches [140] aim to train a model on a class of tasks, such that it can solve novel tasks using a small number of instances given good initial conditions (learning to learn). 2) Embeddings-based approaches [141, 142, 143] learn a metric space in which classification can be performed by computing distances to prototype representation of each class [144] and 3) Metric learning based approaches [144, 141, 142, 143] learn a set of projection functions (embedding that transform data) such that when represented in this embedding space, images are easy to recognize using linear classifier or nearest neighbor [144].

Chelsea et al. [140], introduce a MAML framework, a class of meta-learning methods that are trained on the episode to adapt to novel tasks quickly. The meta-learning is decomposed into two stages: 1)The meta-learning phase, where the model learns a set of parameters gradually across various tasks, and 2)Adaptation, where the model quickly adapts learned knowledge to learn task-specific parameters. The objective in MAML is to learn good initial parameters in the meta-training phase, maximizing its performance on the novel tasks. Sung et al. [144], introduced Relation Network (RN), an end-to-end model which once is trained, can recognize novel images by computing relation scores between query images and the few examples of each novel class without updating the network. Snell et al. [142], introduced a prototypical network (ProtoNets) for the problem of FSL. The model learns a metric space in which classification is performed by measuring the distances to prototype representations of each class. By finding the nearest neighbor between each class prototype and novel class, the data is categorized.

Despite their success, most FSL methods do not improve the latent representation. The representation is biased to the classification loss rather than explicitly attempting to improve it. For instance, a novel sample can be classified quickly through a simple distance metric within the learned embedding space. Therefore, the performance relies heavily on the trained embedding space. A lack of explicit objective function on the embedding space affects the generalization performance of the model.

Few Shot Learning with Contrastive Loss for Improving the representations

Recent works introduce a contrastive loss to the objective function for improving latent representation to alleviate the pitfall of lack of explicit objective function in FSL. The objective is to incorporate a structured model in which instances from the same classes (positive) are grouped while different samples (negative) are far apart. The contrastive learning approach trains the embedding network to extract relevant features with good generalize capability using unlabeled data. The encoder (feature extractor) is trained to map representations of similar images to nearby points while maps representations of different images far apart points. Often after the embedding network a linear layer is used for classification.

Li and Liu [145] proposed a similar concept; however, they train a classifier using a graph aggregation network to obtain more discriminative features. Gao et al.[146] address the FSL problem using contrastive learning with augmented embeddings. The augmented embedding consists of concatenated representations of the original and augmented image. Chen and Zhang [147] proposed a multi-level contrastive learning model for addressing the FSL problem. Their model uses a multi-level contrastive loss at the different layers of the projection head to learn multiple representations from the encoder. Chen et al. trained a model using contrastive loss with a pretext task to maximize the mutual information between two augmented views of the same image. Contrastive learning tries to learn powerful representations from the data with pseudo labels. The result representation is intended to be more discriminative and generalizable across other tasks.

Imprinted-weight Models

Among FLS based approaches, the one which is closer to one of our works is Imprinted-weights [141]. Therefore, we adopt this model for its easy interpretation, and imprinting is a straightforward method. Imprinted-weight models fall into the category of FSL techniques which aim to adapt the model such that they perform well on novel instances which are not present in training, given just one or few samples for each class. The classification is performed directly by comparing the distance of the novel data with other class prototype representation.

2.4.2 Domain Adaptation

Machine Learning models excel at learning from large labeled data, but the model generalization performance can be degraded when the novel data. Unfortunately, collecting massive labeled and curating datasets is a costly and time-consuming process. Another shortcoming of Machine Learning models is that they work exceptionally because the training and testing sets belong to the same distribution. However, when the data distribution domain changes, the model performs poorly. Domain adaptation (DA) is a subfield of transfer learning (discussed in 2.3.4) that arises when the training distribution is different from the testing distribution. Domain adaptation techniques seek to minimize the effects of domain shift [148]. The domain adaptation can be categorized into three classes depending upon the available target domain dataset.

Unsupervised Domain Adaptation

Unsupervised domain adaptation makes use of the available labeled source data. The source data may belong to different distributions from unlabeled examples in the target domain [149]. In other words, the premise is that the training domain has large labeled data (source domain), and the class information is not available in the testing domain. The model is trained on multiple sources domains and tries to generalize well on different but related target domains. The main challenge is to minimize the domain shift (distribution shift) between the source and correspondent target domain. Following [150] three types of methods are used:

Adversarial based The model is trained in an adversarial manner to minimize an approximate domain discrepancy distance between the source and target domain [151, 152, 153]. For instance, [148] proposed to learn a discriminative representation using labels in the training domain and then separate the embeddings that map the test domain to the same space using asymmetric mapping learned through a domain-adversarial objective function.

Reconstruction based The model is trained with an additional task to reconstruct the representations from the source and target domain data. The focus is on creating representations that are shared by the training and testing domain. For example, [154] proposed to learn domain-invariant representations by using autoencoders and explicit loss functions, which captures shared representations by the domains.

Semi-Supervised Domain Adaptation

Conceptually similar to UDA, semi-supervised learning (SSL) studies a different way to learn the target domain with little manual effort [149]. Instead, SSL learns the target model from a few annotated target data and an extensive amount of unlabeled target data. For instance, a standard experiment trains a deep learning network with source data such as Street View House Number (SVHN) dataset. SVHN is a real-world image dataset of house numbers. The testing is performed with a different target domain, such as the MNIST dataset. To adapt one domain to another, the model learns to extract relevant features from the source, which might generalize well in the target dataset. The objective function is designed to maximize similar characteristics, i.e., learn shared features between the two domains.

Following [126], DA is essentially categorized into classes: instance-based DA and feature-based AD. The instance-based DA class reduces the discrepancy by reweighting the source examples and trains on the weighted source data. The feature-based AD learns a common shared space in which the two domain matches. In general, DA aims to explore shared knowledge from the labeled source domain to the unlabeled target domain by searching domain-invariant features that link different domains [155].

Layer Selection in Representation Learning

Layer selection in representations has arisen as a new learning technique for various applications from a deep dream to transferable knowledge (transfer learning). Compared to shallow methods, the teaching is done in one forward pass. Moreover, the layer selection allows representations to be reused in the same network (residual networks) or posterior network training (fine-tune).

In deep network architectures, the representations of the last layers tend to be task-specific. For settings where the testing dataset is limited (FSL problem), relying on deeper layers might not be efficient. Therefore, there is a need for selecting features that might generalize well. The objective is to make the best use of representations, and task-agnostic features are often optimal compared to task-specific ones. Although it has received little attention in the FSL problem, this technique has become well-known in self-supervised learning. In chapter 3, we show that optimal feature selection improves the generalization capability, in particular for imprinted-weighted models.

A similar concept called feature importance ranking (FIR) [156] is used in explainable or interpretable AI. It aims to measure the contribution of individual input features to the model's performance. The explainable or interpretable AI is out of the scope of this thesis. Instead, we aim to clarify the difference between optimal feature selection and feature importance in explainable AI. The good feature selection employed in this thesis addresses the problem of transferring features from a layer that does not generalize well, for instance, transferring features from a task-specific layer. Good layer selection of the representation learning is a straightforward way to improve the generalization of the imprinted-weighted models.

In the next chapter, we introduce our proposed approaches to improving the quality of representation so that the latent representation space is aligned with our target downstream tasks, such as generation and classification.

Chapter 3

Controlling Representation Learning for Image Generation and Few-shot Learning

3.1 Overview

The ultimate objective of representation learning is different from the classic Machine Learning methods, such as the classification and generative approaches. It has been stated that setting a clear and direct training objective function for representation learning is challenging. In the case of the generative model, the objective is to recover a minimal set of latent variables that explain the data distribution. In the case of representation learning, it has been claimed that the aim is to disentangle the underlying explanatory factors hidden in the data with minimal information. However, there is a trade-off between preserving relevant information that explains the data and attaining excellent properties such as interpretability and controllability. Besides, the latent variables are found to allow less or no control of the output unless the label information is provided at the training. We propose an objective function that controls the latent representation to resolve these issues. We show the application of this objective function in the task of image in-between. Our objective function forces the latent representation to have a structure that is aligned to our downstream task.

In the case of discriminative learning, the objective is explicit. The aim is to minimize the classification error between the label information and the input data. In the case of representation learning, we aim to learn a classifier and learn good representation. We address the problem of generalization on novel or unseen classes. We propose that a good representation is the one that maximizes the mutual information between the label information y and the latent code z while keeping minimal information about the input data X. Also, in the case of transfer learning, we propose that transferring representations (knowledge) from task-agnostic layers generalize well on novel classes. The second and third part of this work proposes a variational information bottleneck loss and a good layer selection for few-shot learning applications.

3.2 Introduction

A machine learning model transforms input data into a meaningful output. Therefore, the fundamental problem in machine learning is to turn data into a significant representation, or in other words, to learn good representations that describe the raw input data. Furthermore, it has been claimed that extracting good representation leads to the expected output. However, it has been stated that the visual feature representation contains unnecessary information to describe the generative data process. Besides, there is a trade-off between preserving as much information about the input as possible and attaining excellent properties [94]. While current models have shown an incredible ability to produce good representations for the learning problem, one of the main criticism of such models is that they offer limited to no control over the latent variables. Besides, the models lack interpretability and controllability.

Variational Autoencoders and Generative Adversarial Network have unveiled the potential to alleviate these shortcomings. However, explicit control of the latent codes is still an unsolved problem in computer vision. Besides, the properties of the learned representation might not necessarily be aligned to the target task. For instance, different approaches produce distinct representations. A classic generative model learns the representation by maximizing the information between the input space and latent variables. In contrast, discriminative models using deep hierarchical networks learn the latent representation by passing the raw input data into a series of transformations. As a result, discrete layers learn different representations. In addition, the representation is biased to the label information.

Other approaches, such as self-supervised models, learn the representation by training the network using contrastive loss. The contrastive objective function groups similar input representation while placing different input far away. This method is beneficial for extracting common representation from samples. However, they are biased to pretext tasks. The representations learned by these models are arbitrarily dependent on the downstream task. Therefore, they do not generalize and offer a way to manipulate the representation to produce a controlled output.

Despite these challenges, one model that has shown promising results in addressing these shortcomings is the disentanglement model. Bengio et al. [3], and Higgins et al. [9] state that good representation is the one that disentangles the underlying factors of variation. If this is justified, any lack of latent representation uncontrollability should present some learning barrier. Consequently, it will limit human creativity. Controlling representation enables Machine Learning models to present the desired property, which allows us to address the learning problem. This chapter aims to clarify and give a reasonable explanation for our research question.

We first list some applications of representation learning, and then, next, we contextualize the domains of representation learning that this work lies on. Next, we introduce the core problem in representation learning and describe our proposed methods to address the limitations. Finally, we discuss the importance of our approaches and related works.

3.3 Representation Learning in Discriminative Model vs. Generative Model

Machine learning models can be classified into discriminative and generative modeling. The discriminative model aims to learn a classifier from conditional probability $P(Y|X = x)$ of the target Y given observations. In contrast, the generative model seeks to solve a more general problem learning via a statistical model of joint probability distribution $P(X, Y)$ over all variables X and a target variable Y.

In discriminative learning, representation learning is a side effect of the objective function. Good representation learning is obtained from large datasets. Having a large dataset enables the model to capture a large number of possible input configurations. In addition, good representation is said to be the one in which the intra-class variation is small, and the inter-class variation is significant. For example, the latent variables from the same class are grouped while from different categories are far away.

Another way to achieve such qualities is to train the model using contrastive methods. A contrastive method measures the distance similarity and dissimilarity of features between images. Although contrastive learning methods perform well, one might want to generalize for classes not seen at the training phase. The contrastive method used a cross-entropy loss function to discriminate the classes. However, it does not model the latent space itself. The encoder might choose to encode irrelevant discriminative features for the learning problem.

Another technique used to improve representation learning in discriminative models is transfer knowledge (layer selection). For example, the intermediate layers of a pre-trained model have been shown to have good intermediate representation such as shared statistical features, which can be relevant to a variety of learning problems, such as classification [157].

There is no explicit objective function modeling the latent variables in the discriminative model. In contrast, in the generative model, in VAE, the objective function attempts to model the latent variables. The objective function of VAE enforces a global structure in the latent space through a prior distribution. Sampling from structured latent space allows generating new and realistic samples. This latent space embedding of data points makes VAEs an effective tool for understanding variations in raw data [158].

However, several aspects of the baseline VAE framework prevent it from performing well. First, there is a shortage of theoretical explanations on why constraining the representation leads to the desired outcome. For example, Higgins et al. [9] demonstrated that imposing a decisive penalty on the latent space z forces the model to learn salient features of the data that are relevant for generating novel data with attractive properties such as disentanglement. Second, in VAE, the generated images tend to be blurry. The sample quality can be improved with complex latent structures such as hyperspheres [159].

3.4 Problem Definition

Recently representation learning has become a subfield of Machine Learning [3]. However, before we uncover the limitations on representation learning, we recall its definition and goal. Representation learning is a set of techniques that aims to extract relevant features that make subsequent learning tasks easier, such as classification and generation. The quality of the representation is evaluated on the downstream learning problem. Existing problems on representation learning can be categorized into generative and discriminative methods.

3.4.1 Generative models

The core issue in representation learning is the lack of interpretability and allows less or no output control. In this work, we address the core issue of controllability in representation learning. Although extensive research on representation learning has been carried out on controlling the representation, few works attempt to control the representations to control the output. Existing works have not focused on controlling representation learning to have a desired structured required for the downstream task, such as controlling the output generation. Instead, previous works use label information to have a controlled output or employ a loss function that disentangles the underlying factors of variations in the data.

The first approach requires labeling information that is often hard to obtain or requires human annotation. Besides, the output is controlled solely with label information present in the training data. Examples of this method include the conditional generative adversarial network and variational autoencoders. The second method employs a decisive penalty on the representation and aims to disentangle factors of variation present in the data.

The state-of-the-artwork VAE may allow controlling the latent representation learning. However, the representations might not be aligned to the downstream task. Besides, VAE does always give better representation. Therefore, we aim to improve the representations of VAE for the downstream task. This work shows that controlling the representation learning allows controlling the output of generative models. Furthermore, we demonstrate the application of our approach to the problem of image in-between.

3.4.2 Discriminative models

The core issue in the discriminative model is the lack of generalization on unseen classes. Several studies have attempted to improve the classification accuracy in supervised models. However, few works do not focus on modeling latent representation learning. Instead, existing works rely solely on cross-entropy loss function to pre-train the model, i.e., the objective of the cross-entropy loss function is straightforward. It attempts to minimize the number of misclassification in the training sample. The representation learning is a side effect of the loss.

Existing works that attempt to improve the representations rely on self-supervised methods such as contrastive and pretext tasks. Although the results have shown to achieve good performance, almost similar to supervised models, the objective function of these methods does not directly model the latent representation. For example, the encoder is free to encode features that might not be relevant for generalization on novel classes. In the case of representation learning, our objective is not to train a classifier; instead, we aim to have a good latent representation that can generalize well on unseen classes. This work argues that the cross-entropy loss alone is insufficient in the Few-shot learning paradigm. Instead, we introduce the variational information bottleneck objective to the classification task. This work introduces the variational information bottleneck loss function to the FSL classification task.

Another core issue we address in this paper is related to the task of layer selection (transfer features) for FSL models. Despite many works on transfer learning (transfer knowledge), there has been little attention on imprinted weight approaches. The imprinted weight model assumes that the representations from the feature extractor generalize well on novel classes. However, it is has been studied that the quality of learned representation from encoder affects the generalization on unknown classes. Besides, it is not known what features are relevant for abstraction. In addition, the features which best describe the base classes (training) might not be the best for unseen classes. This paper highlights the importance of layer selection in imprinted models to improve generalization.

3.5 Proposed Method in Generative Model

We explore how representation can be useful in generative models. The representation in the generative model is learned by modeling the data distribution $p(x)$. For example, In VAEs, the objective function minimizes the data likelihood. Each input feature is encoded into a latent distribution, from which the variables are sampled to reconstruct the initial examples, and In GANs, the representation is learned through a minimax adversarial game. The discriminative network can work as a good representation when the model is trained.

The representation learning in generative models have shown attractive property such as semantic interpolation and attribute arithmetic. By sampling the latent space, we can create novel artistic artworks, i.e., one might create unique songs, videos, or creative images. For example, Gatys et al. [40] proposed a neural style transfer that combines content from one image with style from other. Google's deep dream model generates images with features from different neural network layers. Recently, we have deep fake videos created by imposing face images of a target person onto a video of the source person to create a video of the target person acting or speaking as the source person. The generated music, images, and videos exemplify what deep learning models can do if the representation learning is manipulated. However, although they have such characteristics, controlling representation learning to have a controlled output is not straightforward. Therefore, we assume that controlling representation will generate a controlled output.

3.5.1 Learning Representation Learning

The premise behind representation learning is to model the high-dimensional representation space X into low-dimensional latent space z using observation in the training dataset. Then learn a mapping function that can take a point in the underlying space and map it to a point in the original domain. In other words, each point in the latent space represents some high-dimensional input data. The latent space of representation learning enables us to perform operations that affect high-level properties of the input data.

We borrowed the example used by David Foster [160] to explain at a higher level what representation is trying to accomplish. For instance, suppose we have training data consisting of grayscale images of tins, as shown in Figure 3.1. Two features may represent each of these tins: the tin's height and width. If we are given the task to draw a corresponding container, which does not belong to the training set, we could do it. However, the same task is not so straightforward for machines. First, it has to learn that height and width are the two latent features that better describe

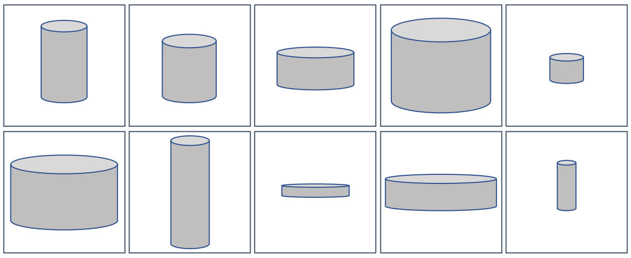


Figure 3.1: Training image of tins

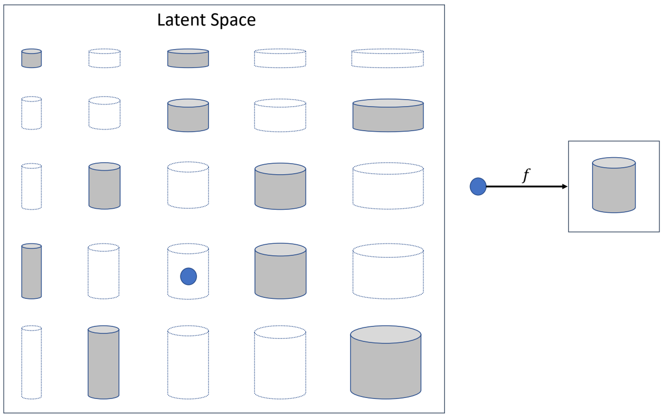


Figure 3.2: The blue point in the latent space is mapped to image space

the dataset, then discover a mapping function that takes a point in this latent space and maps it into the tin image. Figure 3.2 shows the resulting latent space of tins and the generation process. The blue point represents the sampling operation. The decoder then generates the data. This operation is not evident in the image space (pixel space). The representation learning is powerful because it learns relevant features that are more likely to describe the given observation and the generation process. In other words, it finds a highly nonlinear manifold on which data lies and then establish the dimensions required to represent this space [160].

3.5.2 Proposed Objective Function to Control Representation Learning

Learning to control the data representations is fundamental for developing models that require controlled output. In deep learning learning to control the latent representation is an active area of research [161, 9, 162]. Unfortunately, state-of-the-art Variational Auto-Encoders (VAEs) [98, 97] for learning representation learning might not give the desired representation for the downstream task. For example, in the image in-between downstream task, in chapter 4, we show that vanilla VAE fails to have a proper structure that allows interpolation in image space. Furthermore, we also demonstrate that the latent variables of vanilla VAE shows to have a degree of freedom. Both these shortcomings lead to poor control of the output. To address these limitations, this work proposes an objective function that attempts to improve the representation learning in vanilla VAE. Our proposed loss function imposes a strong constraint in the latent representation space by limiting the degree of freedom of the latent variables. Following is the idea behind the degree of freedom. The latent representation learning of vanilla VAE might have several structures or forms. If we do not place a penalty, the latent structure might not align with the desired downstream tasks.

Inspired by these limitations of VAE, we modify the standard objective function 3.1 by explicitly designing a loss function that allows controlling the latent variables' degree of freedom. But, conversely, this objective function penalizes the degree of freedom of the latent variables. For instance, chapter 4 shows the importance of controlling the representation to have a controlled output using an example of in-between image generation. However, our proposed loss function is not restricted to in-between image generation, and we apply it to linear interpolation. Below we express our proposed objective function 3.2.

$$L(\theta, \phi) = \mathbb{E}_{z \sim q_\theta(z|x)}[\log p_\phi(x|z)] - KL(q_\theta(z|x)||p(z)) \tag{3.1}$$

$$L(X_0, X_1, X_2) = L_{VAE}(X_0) + L_{VAE}(X_1) + L_{VAE}(X_2) + \alpha(D(q_{(X_1)}||\frac{q_{(X_0)} + q_{(X_2)}}{2})) \tag{3.2}$$

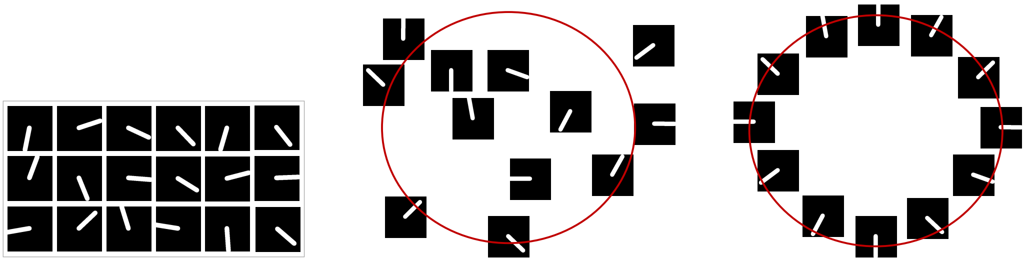
Where the three losses $L_{VAE}(X_0)$, $L_{VAE}(X_1)$ and $L_{VAE}(X_2)$ are vanilla VAE loss function as shown in Equation 3.1, and the last term is distance between the average latent representation and the teaching signal. α is the adjustable hyperparameter and controls the importance of the difference between average and teaching latent signal representation. This objective function encourages the model to learn the data representation structure aligned with our downstream task.

In chapter 4, we give a detailed explanation of this objective function, and we applied it to a real-world application such as for the in-between image generation. Our proposed loss function is straightforward, simple, and outperforms the vanilla VAE. Furthermore, we experimentally show that our proposed objective function obtains accurate results for the downstream task in diverse datasets.

3.5.3 Justification for the proposed objective function

In machine learning, we assume that the data distribution in the latent is smooth. The probability of feature should be similar in two neighboring points. Representation learning aims to discover the data structure aligned to the downstream task, i.e., ensure that the latent space has good properties required for the learning problem. We show two examples that illustrate the application of the proposed objective function.

Example 1. Suppose that we have a synthetic dataset containing similar images with geometric rotation such as angles $(0^o - 360^o)$ as shown on the left side of Figure 3.3. The bar is fixed in the center of the image, and we randomly generate images of the bar with random angles (rotations). The objective is to train a generative model to discover the hidden structure of the data and ensure that the latent space has good properties for the downstream task. For example, the latent representation might assume many representations; however, we assume that the structure is likely to be a circle in image space. The Vanilla VAE model can reconstruct the image; however, it fails to preserve or discover the hidden structure of the data aligned to the learning problem. Our proposed loss function reconstructs the initial data and discovers the data's latent structure.



(a) Image data

(b) what can happen without controlling representation

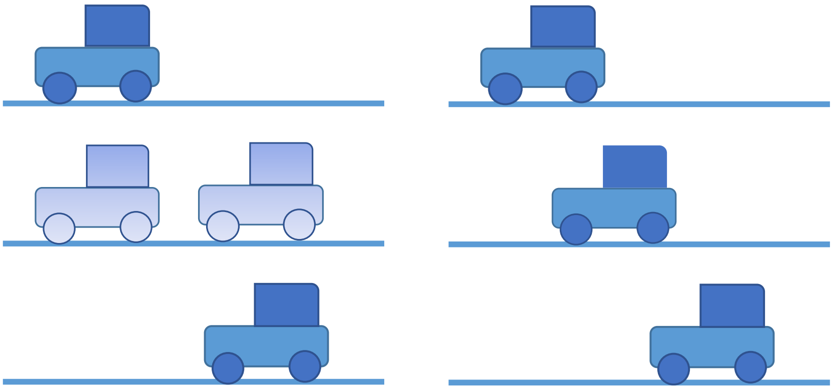
(c) what we want to obtain with controlling representation

Figure 3.3: Predicting the structure of the data. The red circle shows the actual structure of the data. The representation learning of vanilla VAE that is not controlled learns to reconstruct the data. However, it fails to preserve the structure of the data. In contrast, we propose an objective function that learns to reconstruct the data and discovers the data structure aligned to our downstream task.

Example 2. In Figure 3.4, we demonstrate an example of controlling representation to have a controlled output. Suppose we have the task of generating the image in-between. Our dataset contains a synthetic image of cars. First, we feed the raw input data (first and third car) to the generative network model. The input images do not need to be consecutive. The objective is to have a latent space such that interpolation in latent space is possible, i.e., with a property that allows the generation of in-between in the image space which desired characteristic. To have such behavior, we claim that the latent space should be constrained, i.e., control its degree of freedom. The vanilla VAE is likely to generate the inbetween as shown in the left-side of Figure 3.4. The inbetween image might not be aligned with the true inbetween. The representation from VAE does not preserve the actual structure of the in-between. The generated image might have a random position. The objective function proposed in this work, encourages the latent space to generate a realistic inbetween image as shown in right-side in Figure 3.4.

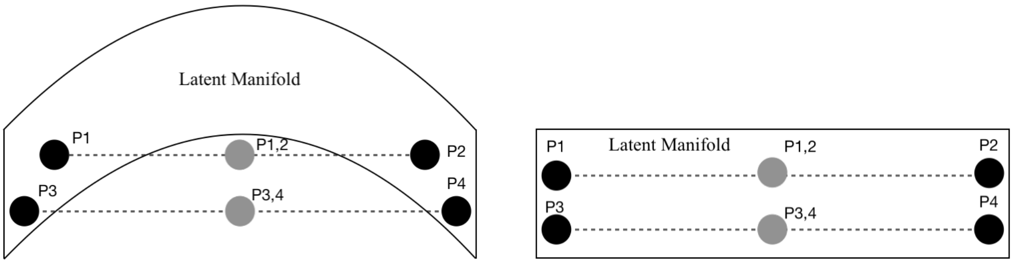
3.5.4 Latent Manifold Flattening for Data Interpolation

We intend to explain the importance of a flat manifold for images in-between generations. There are many possible paths between two data points in the latent manifold. Our objective function restricts the degree of freedom of the latent variable and forces the manifold to be locally flat. Looking at the Equation 2.10, we have the adjustable hyperparameter; in standard VAE, such parameter does not exist, which is ideal because it is one less parameter that the model needs to train. However, models such as β-VAE have shown that introducing an adjustable hyperparameter to balance the KL divergence and the reconstruction loss forces the model to disentangle the features in the data. Our hyperparameter α forces the latent manifold to be locally flat. To know if the manifold is flat, we take two points at the latent space and then average. If the point in-between looks like actual data, then we assume that data is realistic, and if this is true for every linear combination, we hypothesize that the manifold is flat.



(a) Uncontrolled representations (b) Controlled representation

Figure 3.4: Schematic representation of the overall objective of controlling latent representation. a) Vanilla VAE does not preserve the property of the actual in-between image. Our objective function aims to control the latent representation space to generate the image with desired properties.



(a) Vanilla VAE (b) ours

Figure 3.5: Vanilla VAE is likely to interpolate in a curved latent manifold. Our model forces the manifold to be locally flat, resulting in smooth interpolation.

3.6 Proposed Method in Discriminative Model

In a discriminative model, representation learning is seen as a side effect of the loss function. Good representation learning is achieved from large datasets. Having a large dataset allows the learned representation to capture a large number of possible input configurations. Existing methods rely on a two-stage optimization pipeline, where the network is pre-trained on a large source dataset then fine-tuned on a smaller target dataset. However, this method assumes that both source and target domain share similar features. This work follows the same training pipeline; however, we assume we have a single image per novel or unseen class.

Another way of acquiring rich feature representation from a large unlabeled dataset is through unsupervised visual representation. The critical ideas are, first, to train models with contrastive loss. This method aims to group representations from the same class while pushing away representations from different categories. The second method typically creates the pretext tasks with free supervision. This method includes predicting the rotation and the frames' sequence in video. However, these methods rely on cross-entropy loss function, and they do not explicitly model the latent representation. To address these methods' shortcomings and the generalization problem in transfer learning, we propose two solutions.

1- This work introduces an objective function that models the latent representations. We claim that cross-entropy loss function alone is not sufficient for generalization.

2- We propose a good layer selection for the weight imprinting models.

3.7 Advantages of representation learning in discriminative models

Representation learning in discriminative learning aims to learn rich features from the data that can be useful for downstream tasks such as classification or prediction. There are two significant advantages of representation learning in deep neural models. For instance, it allows the transfer of knowledge from one task to another. This property has motivated researchers to pay much interest in learning representation in discriminative learning. By transferring the knowledge from a model trained on a large dataset, we can train a smaller network with a smaller dataset without overfitting. This extraordinary property of knowledge transfer is motivated by the assumption that the underlying data representation across domains is similar.

Methods that benefit primarily from such observation are called transfer learning, particularly domain adaptation. A common assumption on discriminative models is that the training and test data follow the same distribution. When the distribution changes, most statistical models must be reconstructed from newly collected data [163]. While this might be feasible in some domains, in other applications can be costly or impossible to recollect the data. Therefore, it is fundamental to develop approaches that the need and effort to collect novel annotated data by exploiting the representation that can be useful across various tasks. Humans can acquire novel knowledge and transfer it across domains without much effort.

For instance, we can teach an infant to recognize a tiger with few or one sample. The infant can generalize to other species. Another illustrative example is that, for example, animal species

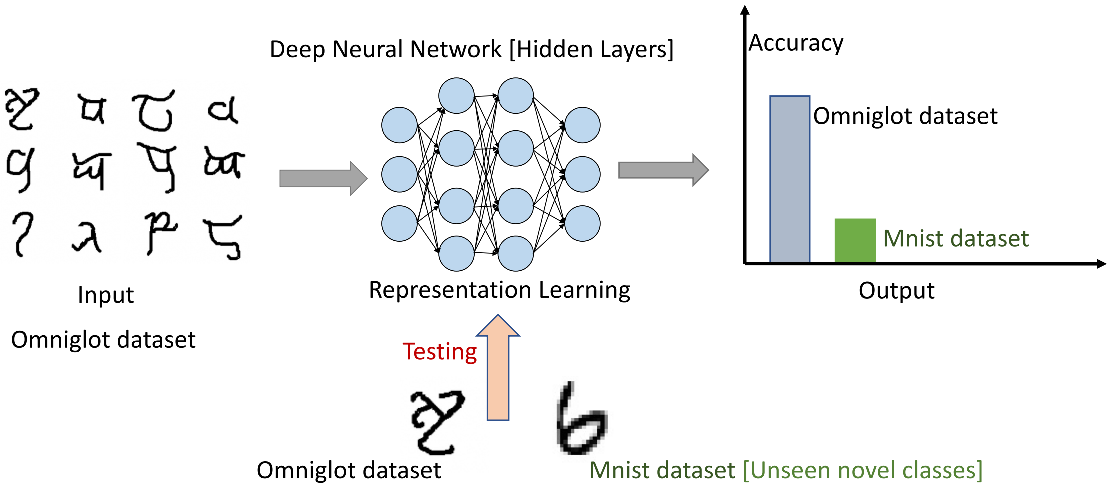


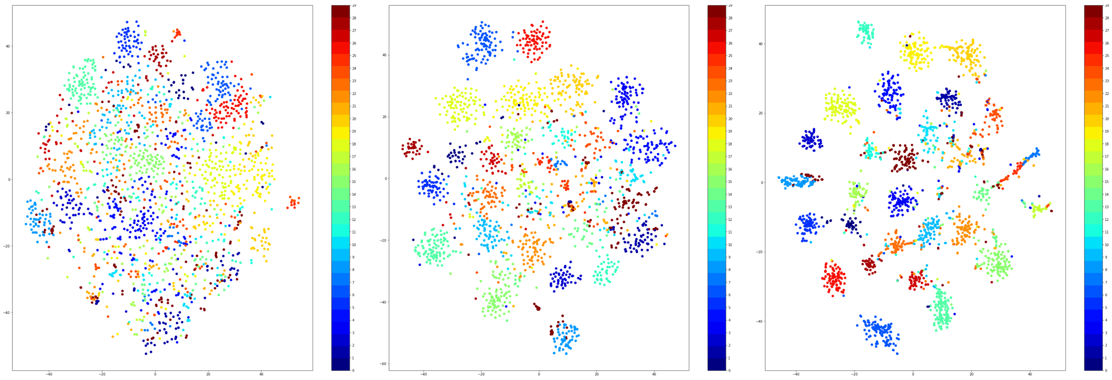
Figure 3.6: Transfer representation: Training on Omniglot and testing on Mnist dataset

that might look different and can be found in separate regions due to evolution background and climate. However, they might belong to the same family. It is desirable to develop a robust model that considers changes in the environment and adapts quickly to new problems. In response to this problem, we look into representation learning, i.e., what knowledge should be transferred across the domain to perform well on test samples from a different but related domain.

For example, in Figure 3.6, we illustrate an example of knowledge transfer between two different but related domains. The source domain is the Omniglot dataset, where we assume that there are enough datasets. The target domain is the MNIST dataset, where we have a single example per class. The objective is to train the network on the source domain (Omniglot) and use the learned representation to generalize on a novel target domain (MNIST). The performance on novel data (MNIST) is often poor, and the learned representation does not generalize well. One reason is that the latent variables from where the knowledge is transferred are task-specific. Therefore, the model does not learn the structure of the data. However, it knows specific discriminative features of the class.

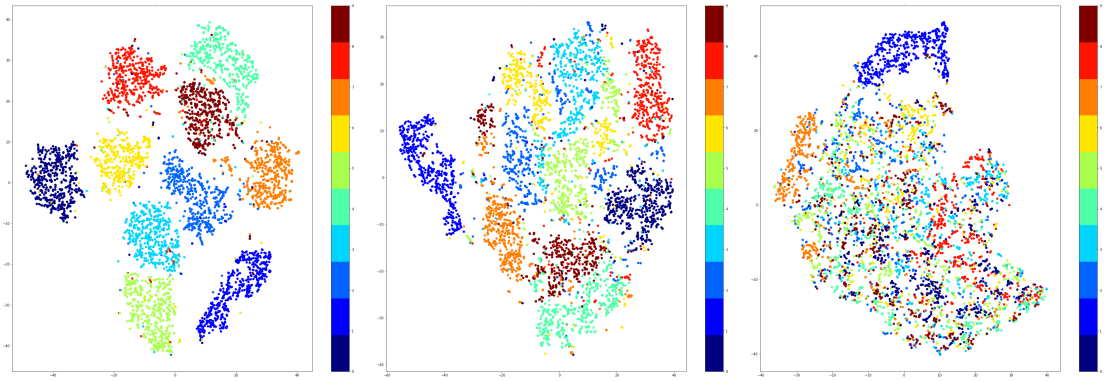
3.8 Controlling representations in discriminative learning

Controlling the representation in discriminative models is related to the question of what features to transfer. For example, the higher-level layers tend to produce task-specific latent variables. In contrast, intermediate layers tend to be more task-agnostic. For example, in Figure 3.7, we show an example of controlling the representation in a discriminative model. We trained a deep network model with a feature extractor $f(x)$ and three fully-connected layers. In Figure 3.7, we project the embeddings from the three fully-connected layers. The embeddings from the third layer exhibit a clear boundary between classes. We hypothesize that the model has learned specific discriminative features of each category. Therefore, transferring knowledge from the third layer might not generalize well.



(a) First Full-connected layer (b) Second Full-connected layer (c) Third Full-connected layer

Figure 3.7: Latent representation in different layers at the training phase



(a) First Full-connected layer (b) Second Full-connected layer (c) Third Full-connected layer

Figure 3.8: Projection of the latent representation at test phase on novel classes

We propose selecting representation for the task of domain adaptation from a good layer. By good layer, we mean from task-agnostic layers. We illustrate in Figure 3.7 the performance at test time. For this example, we tested the model using the MNIST dataset. The latent projection shows that transferring the representation from the first fully-connected and second yields good performance on novel data. We also propose that one way to learn good representation is by employing an objective function that explicitly models the latent variables. Finally, we demonstrate that cross-entropy alone is not enough for learning good latent representation that can generalize well across various domains. In this thesis, we demonstrate that controlling the latent representation (manipulating and layer selection) for some tasks is helpful and necessary for the learning problem.

3.9 Proposed objective function- Variational Information Bottleneck

This work introduces variational information bottleneck (VIB) for the classification task in few-shot learning. Existing works have obtained outstanding results when fine-tuned on a variety of tasks.

However, they overfit when trained on single or few examples per class. Many of these features might not be relevant for the downstream task when transferring features from pre-trained models. Inspired by the effectiveness of the information principle [164, 165, 166], we propose Variational Information Bottleneck loss function. This objective function aims at learning relevant features that might generalize to unseen classes and ignore irrelevant features for the classification. Below we express the Variational Information Bottleneck loss function.

$$L_{VIB}(\theta) = I(Z, Y; \theta) - \beta I(Z, X, \theta) \tag{3.3}$$

Where the first term $I(Z, Y; \theta)$ is the cross-entropy loss function, the second term $I(Z, X, \theta)$ is the mutual information between the encoding and the original dataset. The first term encourages the representations to predict Y, and the second term encourages latent representation Z to forget X. The objective is to maximize the first term to predict Y, while Z keeps minimal information about the input data X. The β hyperparameters balance the trade-off between the cross-entropy loss and the mutual information between the input data X and latent representation Z. β=0 is equivalent to a standard classification model. Variational Information Bottleneck loss is similar to VAE loss function with one main difference, in VAE, the first term is the reconstruction loss, and in Variational Information Bottleneck is a classification loss. Variational Information Bottleneck can be seen as a regularization to discard irrelevant features, making it easier for the downstream tasks classifier. The efficient variational estimate of Equation 3.3 and Equation 3.4 have been shown in [166], also used in [167] for the task of transfer learning in natural language processing (NLP).

$$L(\theta, \phi) = \mathbb{E}_{z\sim q_\theta(y|z)}[\log p_\phi(y|z)] - \beta KL(q_\theta(z|x)||p(z)) \tag{3.4}$$

This work proposes to use Variational Information Bottleneck for improving generalization in few-shot learning models. We show that the cross-entropy alone is not enough for good generalization on unseen classes.

3.10 Why Does Representation Learning Matter?

Many of the best-performing models suffer from poor data efficiency. Besides, their performance often lacks the equivalent level of robustness and generalisability that is characteristic of biological intelligence [168]. Representation learning is exciting because one can think of a way to find more straightforward representations of data for the learning problem. Besides, representation learning explicitly expresses many common priors about the world around us. It is not restricted to task-specific, i.e., it is to be helpful for task-agnostic hopefully [3].

For instance, finding a good feature representation may help us build abstract representations for multiple downstream tasks. In this context, good feature representation refers to the disentanglement of semantically meaningful, statistically independent, and causal factors of variation in data. Alternatively, by imposing a constraint in the latent representation for data generation, one may view it as an implicit form of regularization [169].

Another advantage of representation learning is that we can sample points in the latent space and manipulate them to generate novel and attractive samples with different properties. Besides, latent space is not limited to simple manipulation. For example, we can transfer the learned representations from one source domain to other downstream tasks since they encode shared features.

The objective of representation learning is to extract relevant representations that enable models to perform well on machine learning downstream tasks. Good representation is the one that produces higher performance on the downstream task. The representation learning employed in this thesis is classified into Discriminative and Generative models.

3.11 How do we evaluate the quality of a representation?

There is no straightforward approach for evaluating the quality of learned latent representation. Machine learning methods such as generative and discriminative models rely on various proxies such as linear separability and mutual information between representations and class labels. A popular method in the case of classification problems is to transfer the learned representation (transfer learning) to a smaller neural network and train on different data domains. This setting is well-known in self-supervised learning. Next, we fine-tune the encoder $f(x)$ by replacing the pre-trained classifier with additional fully-connected layers and a new supervised classifier.

Then, the performance of the initially learned representation is evaluated on the target downstream task. If the model performs well on the target task, we assume that the representation learning is good or generalizes well. Another proxy used to evaluate the quality of learned representation is based on measuring the mutual information between the information encoded in the representation and the class labels [170, 74].

3.12 Assumptions of Latent Space

As state by Bengio et al. [3], ”One reason explicitly dealing with representations is interesting is that they can be convenient to express many general priors that are not task-specific but would likely be useful for other learning problems.”

The characteristics that the a good latent space is expected to have may be explicit in generative models, such continuity and completeness. However, in discriminative models, we expect that a good latent representation space may be able to separate (form a different cluster) a novel data point. For example, suppose we have a classification task. Our training sets contain images of cats and dogs. When the model is optimized with cross-entropy. The latent space can form a clear cluster of the mebedding of cats and dogs. If we test, we unseen class such as bird, the latent space has to create a new cluster a bit further than the embeddings of cats of dogs. We aim that a good latent space has to have such property.

3.12.1 Continuity

Two neighboring points (representations) in the latent space should produce similar content once decoded. We use the above example of cats and dogs. Give two images of cat and dog. The latent variable of these images should be closer. And if we interpolate the two latent codes, we should see a smooth transition from one class to another.

3.12.2 Completeness

A random point sampled from latent space distribution should generate meaningful content, i.e., not far removed from the initial data distribution. For example, if we randomly sample a point that is in-between the embedding of cat and dogs. This point should be meangful, that is, it should contain features that describe one of the classes or both.

3.12.3 Transferable/shareable across tasks

The latent representation should be transferrable across domains. For example, a model trained on a large labeled source domain should encode properties that are task-agnostic. The trained model does not have to learn useful features related only to the task-specific problem. In some scenarios, we are not aware of the target tasks. Therefore, learning to encode features that might generalize well is said to be helpful for future and unknown tasks.

3.13 Latent Space Manipulation

Bengio et al. [3] have explained that, unlike a classifier where the learning is explicit (minimize the number of misclassification during training phase), representation learning purpose is far removed from the ultimate objective of classification or prediction task. Therefore, it is challenging to set specific goals or targets to evaluate the quality of latent space. One way to assess the quality of latent space is through latent space manipulation and regularization.

Regularization of Latent Space Regularization aims to improve the latent representation (learning useful representations) by penalizing the degree of freedom of the latent variables. The regularization has two objectives. First, samples from the same class should be closer together in representation space. Second, the structure of the latent space should be aligned with the desired downstream task, i.e., the quality of the representation cannot be disassociated with the learning problem. For example, we have confirmed empirically that our proposed regularization generates latent codes that address our learning problems well. This result may suggest a possible link between interpolation abilities and learning meaningful representations.

An excellent example of the potential of manipulating the representations is shown in Neural Style Transfer, where the core idea is to control images created in a deep network. Neural style transfer works by controlling the sequential representations across a CNN such that the style of one image can be transferred to another while maintaining its original content [27].

3.14 Latent Space Interpolation

The latent space can be sampled to generate seemingly ”unique” data. A typical application of this property is a continuous interpolation. Interpolation is used to traverse two known points in latent space; by mixing the variables in the latent space and decoding the result. As a result, the model can generate a semantically meaningful combination of the corresponding data points [102]. Linear latent interpolation is used to demonstrate that the quality of representations also that the model has not merely memorized the training data but has learned the latent structure of the training set [171, 157, 172].

It is argued that interpolation might be helpful for creative application, where a smooth transition between two decoded images is desired. Another concept used to assess the disentanglement is linear separability. However, linear separability is possible only if the latent space is sufficiently disentangled, i.e., the axis-vector corresponds to a single factor of variation.

3.15 Problems with Latent Space Representation

There are two core dilemmas in representation learning.

1- Latent space structure The latent space may take many formats. Without regularization, the structure may not be desired for the downstream task. Designing an objective function that forces the latent space to have the desired structure aligned with the downstream task is vital.

2- Perserving Information There is a trade-off between preserving information about the raw input data and attaining suitable property for classification or generation. A constant criticism of neural networks has been that they are black-box methods [157]. There is no understanding and control on what features to encode. Moreover, the codes are not always aligned with human knowledge. The encoder may choose to encode features that are not relevant to the learning problem and discard or ignore features that describe the data well.

3.16 Related Work

3.16.1 Unconditional Generative Model

A good representation captures the posterior distribution of the underlying explanatory factors of the data [3]. The unconditional generative model has been successful among various ways to learn representations from complex data distributions, mainly due to VAE and GAN frameworks.

Conditional generative allows interactive control, but creating new controls requires expensive training [173]. Often the unconditional generative model is used to generate outputs with desired attributes. We condition the latent space with an objective function that enforces such properties in the output. The conditional latent space in this context is different from the conditional generative model, which we will discuss next (3.16.2).

In the unconditional generative model, we constraint the representations to reduce their degree of freedom. This constraint allows us to generate novel samples with modified attributes while preserving its core identity. This feature is a crucial objective of a generative model. However, achieving such an objective is not straightforward. With current approaches, we require enough labeled data with various properties. Although, even with enough data, the model is less prone to learn the commonalities between tasks.

Generative Adversarial Networks [94] and Variational Autoencoders [97, 98], learn to unconditional generation of realistic and diverse sampling from a semantically structured latent space [173]. Chapter 4 hopes to leverage such structured latent space by conditioning it to create a controlled output. We show that constraining the latent space enforces the model to generate the in-between image. This approach removes the need for extra information (label) and does not rely on pixel (optical flow) to generate the in-between.

3.16.2 Conditional Generative Model

Incorporating structure into the representation can be done using label information directly at the latent variables. For example, this is the case of conditional generative models. In an unconditioned generative model, there is no direct control over the data being generated. However, it is possible to direct the data generation process by conditioning the model on additional information. For example, such conditioning could be based on class labels.

Mehdi Mirza proposed Conditional Generative Adversarial Nets (cGANs) [174], a model which is an extension of GANs. The generator G and discriminator, D, are conditioned on additional input information y representing the labels. The label y is passed into both G and D. The modified loss function of the adversarial minimax game is expressed in Equation 3.5.

$$min_G max_D V(D,G) = \mathbb{E}_{z \sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{3.5}$$

In VAE, the objective function does not force the generation of controlled output. The encoder network models the latent variable z directly based on X, without caring about the label y. The decoder model X directly from latent variable z. Similarly, with cGANs, conditional VAE (cVAE) [175, 176] condition the encoder and decoder with label information c. Finally, the objective function is expressed in Equation 3.6.

$$ELBO(\theta, \phi) = \mathbb{E}_{z \sim q_\theta(z|x)}[\log p_\phi(x|z,c)] - D_{KL}(q_\theta(z|x,c)||p(z|c)) \tag{3.6}$$

Conditional generative models control the output generation. It generates data with specific attribute knowledge information; however, the output is limited to the knowledge of the attribute labels presented in the training set. There is no straightforward interpretation of the representations' structure, such as discovering features correlated to the labels, i.e., it is not clear how to incorporate new attributes without retraining from scratch. The representation or generalization is often improved by providing a larger quantity of representative data [3]. Another way is said to disentangle the underlying factors of variation of the data.

3.16.3 Disentanglement

Representation learning is the core of machine learning models. Unfortunately, despite good performance, current models lack generality and robustness due to their implicit representation learning. In particular, it has been argued that disentangled representations might help overcome this problem. Bengio [3] defines disentanglement as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors. Similar definition adopted by [168, 177, 9].

For instance, a model trained on the Celebrity dataset might learn independent latent units sensitive to single independent data generative factors, such as object identity, position, scale, lighting, or color [177]. Thus, a disentangled representation is interpretable. Learning disentangled representations is helpful for relevant unknown tasks as one can independently control salient attributes of the data. Chen et al. stated that disentangle representation can be helpful for tasks that include face recognition and object recognition. In particular, disentangled representation might allocate a separate set of dimensions for distinct attributes.

Another reason is that suppose we have a machine-learning algorithm that learns the factors that explain the statistical variations in the data and how they interact to generate the data we perceive. Then, we would assume that the model understands those aspects of the world covered by these factors of variation. But, unfortunately, in general, and for most characteristics of variation underlying natural images, we do not have a precise understanding of these factors of variation [133]. Disentangled representation is one way to evaluate the quality of representation learning. The prominent framework for disentanglement is GANs and VAEs.

Disentanglement based on GANs models

Karras T. et al. [10], make use of GANs to control the image generation. They re-design the generator to adjust the image style at each convolution layer using the latent code. Furthermore, the authors claim that the embedding produced by intermediate latent space is free from restrictions, allowing disentanglement. Chen et al. [96], introduced infoGAN, a model that explores the mutual information between latent variables and observation data. The model modifies the GANs by encouraging interpretable latent representation through its objective function, maximizing the mutual information between observations and a small subset of latent codes. Furthermore, the proposed approach decomposes the input noise vector into two parts: the source of incompressible noise z and latent code c.

Disentanglement based on VAEs models

Higgins et al. [9], using VAEs model, proposed a loss function that puts extra constraints on the implicit capacity of the latent representation z. The result is a more disentangled latent representation.

3.16.4 Discriminative Learning

In a discriminative model, the encoder network can encode less discriminative features in the representation space. However, when a new classifier is learned, the classifier will not have enough data to discriminative features, particularly in low-data regimes with very few samples available[38]. Controlling representation limits the degree of freedom on what features the learner chooses to encode. The objective of controlling representations can be achieved through an explicit or implicit objective function using label or pretext task information. There is no explicit objective function to model the representation.

The objective function forces the encoder to encode important features, which will help the model to generalize on the downstream tasks. The previous statements are examples of a standard classifier. Nevertheless, stating Bengio et al. [3] good representation is the one that makes the subsequent task easier, i.e., representation learning aims to map representations to other representations. A good example is the feed-forward deep network model and Transfer learning approaches. The objective of controlling representation is entangled with the downstream task.

We argue that a good selection of the feature representation is beneficial for downstream tasks. Chapter 5 shows that selecting task-agnostic features from CNN leads to better generalization accuracy. In many cases, the representation selection (fine-tuning) is made from a layer before the classifier. We argue that the features from the last layer are task-specific and thus will not help generalize to novel tasks.

3.17 Applications of representation learning

There are several reasons why representation learning is attractive. This section describes the real-world applications that we use controlled representation to improve the downstream tasks. First, we describe the general applications in different fields such as computer vision, natural language process, and audio and speech signals. Finally, we show real-world applications of representation learning that we address in this work.

3.17.1 General Applications

Computer Vision

Computer vision models have advanced in recent years due to novel learning approaches such as generative adversarial networks and variational autoencoders. As a result, we are now able to improve the quality of latent representations, which allow us to generate photorealistic images [178, 179]. In addition, the representation learning of generative models, has been used in wide range of applications, including image editing [180], style transfer [10], image segmentation [181], artistic images [182].

For example, Zhu et al. [183] introduced the CycleGAN model, which can convert a given image into a painting in the style of a specific artist. We can transform a simple photo into a well-known Monet or Van Gogh style. This process transfers the latent representation containing the style information from the original to the content base image. This technique is also known as neural style transfer and has been commercially used in some apps to output users' photos with a given painting style.

More recently, the representation learning of VAE models has shown that if manipulated, we can disentangle the underlying features of the data. For example, Higgins et al. [184] using the celebrity image dataset demonstrated that we can control the features generated by simply doing latent arithmetic. First, the authors show the image transformation from female to male. To do this, we find the vector in the latent space that points in the male direction. Then, we add this vector to the encoding of the original female image in the latent space. This operation will give us a novel point that, once decoded, has new features.

Natural Language Process

The representation learning of generative models is explored on text data. NLP aims to develop linguistic algorithms for machines to understand languages. Thus, representation learning may help to represent the semantics of the languages in unified semantic space and build complex semantic relations among different languages [185]. The primary application of representation learning for NLP is word embedding. Encoding the semantic relation between words can serve as a good representation for spam filtering, text classification, and information retrieval applications.

Examples of good representation learning in NLP include the word2vec [186], the ELMo [187] and the BERT model [188]. They use large corpora to compute word dynamic representations based on their context. Then, the learned semantic knowledge is transferred to related downstream tasks such as summarization, machine translation, text classification [189], relation extractor [190] and dialogue system. Lately, the representation of learning of VAE has been used to control sentence generation. For example, [23], [24] introduce a model for generating text with holistic properties such as style and topic. Niu et al. [191] control the level of formality in text generation while Sennrich et al. [192] control the level of politeness. These models find application in text and content generation.

Audio and Speech Signals

Representation learning has been applied in audio and speech generation for generating broad topics, including controlling the style, emotion, timbre, and content transfer. For example, we can control or adjust the speaker accent, speaking rate, timbre [193] and prosody [194]. It is critical to disentangle factors of variation present in the data to control and transfer such properties. Other speech control applications involve noisy suppression for clean speech, stress, and rhythm of speech.

Another application is cross-speaker transfer [195], which addresses the lack of data of one speaker. The data from other speakers is used to improve the synthesis quality of this speaker [196]. We can manipulate the latent representation for the task of cross-lingual transfer [197]. In music generation, [198], we can control latent representation to generate pieces of music with different attributes, such as mixing two different melodies or transforming from jazz to acoustic genre.

3.17.2 Applications addressed in this work

Generative models

The core issue in generative models is to generate a latent space with properties desired for the downstream task. The generative model's representation learning goal is different from classification learning. In generative models, the objective is to improve the representation and learn the latent structure of the data useful for the downstream task. Chapter 4 demonstrates the application of controlling the representation learning to have a controlled output. For example, our downstream task is to generate the image in-between with a given random sequence of image data. To have an accurate image in-between, we assume that the model has to learn the data structure.

The vanilla variational autoencoder model learns powerful latent representations of the data. However, the latent representation might take several structures that might not necessarily be aligned with our downstream task. For example, we demonstrate that the structure generated by vanilla VAE allows interpolation. However, the generated in-between point might not be realistic, i.e., we do not know what features this point might hold. For example, the point might not belong to the latent manifold. We hypothesis that this behavior is due to the degree of freedom of the latent variables. There is no direct control on the representations to generate the desired structure, which allows the interpolation in the latent and image spaces.

Therefore, we propose to control representation learning so that the generated latent space is structured and has desired structure that allows accurate interpolation. We propose an objective function that constrains the representation. We show in chapter 4 that penalizing the degree of freedom of vanilla VAE allows generating an in-between image with good properties. We also demonstrate that our objective function forces the latent manifold to be locally flat. Finally, we illustrate an example of linear interpolation to validate this claim.

Furthermore, we demonstrate the results on several synthetic datasets. We compare our proposed approach with vanilla VAE and other works, including FlowNet 2.0 and SloMo. For qualitative evaluation, we show that our proposed model outperforms the vanilla VAE. We illustrate the in-between generation, and we find that our model generates the accurate in-between image for all the datasets. While the vanilla VAE fails to preserve the structure of the in-between image. We quantitatively evaluated the in-between image generation between our proposed model and other works. As a result, our model achieves better performance on all metrics used, such as MSE, PSRN, and SSIM. Chapter 4 demonstrates the network structure, the dataset, and the results in detail.

Discriminative models

The core issue in discriminative models is generalization across various learning problems. Unlike representation learning in the generative model, representation learning is the side effect of the explicit objective function. The objective function minimizes the misclassification error between the expected and actual output. Chapter 5 demonstrates two ways to control representations to improve generalization on novel classes. First, we propose an objective function that models the latent variables. Traditional approaches on the few-shot learning classification task do not model the representations. Instead, existing works rely on data transformation, self-supervised methods such as contrastive and pretext task learning. The performance of these models depends upon the quality of the representations created by the encoder.

Furthermore, since the encoder is only optimized with cross-entropy loss, the encoder might choose to encode features that might not be useful for generalization across different downstream tasks. We introduce a variational information bottleneck loss function. Our loss function aims to suppress or ignore latent features that are irrelevant for generalization. We claim that the cross-entropy loss function alone is not enough for generalization. We evaluate our model on the MNIST dataset. We find that the representation learned to generalize better on novel classes with our proposed objective function. We use a single image per class to extract the representations (class prototype). We also show the importance of the number of latent dimensions for generalization.

The second part of this experiment addresses the same problem of generalization on novel classes. Again, we propose good layer selection. Standard models often transfer representations from the feature extractor or encoder. One work that achieves good performance is weight imprinting. We adopted this model. We study the importance of including projections head on top of the encoder. We claim that the representations generated by the encoder are task-specific. We propose transferring feature representations from a good layer, i.e., from a task-agnostic layer. We show that the fully-connected layer at the projection head has a better representation than the feature extractor. We visualize the projection at each layer at projection layer.

Furthermore, we claim that directly imprinting from feature extract, the model loss a lot of information that might be useful for generalization. We evaluate our model trained on one dataset (Omniglot) and test on the different datasets (MNIST). The results show that our approach improves generalization on novel classes. Chapter 5 shows the network structure, the dataset, and the experiments' details.

Chapter 4

Regularizing Representation Learning for image in-between generation

4.1 Overview

The core issues of representation learning in the generative models are that the latent representation allows little to no control. Besides, the latent representation learning may not be aligned with our downstream task. Latent representations of Vanilla VAE may help minimize this issue since it may learn to encode the complex structure of data. However, the latent representation space may not have the desired property to solve the learning problem. This work addresses the problem of controlling the representations by introducing an objective function that allows having a controlled output. We demonstrate an application of this model on images in-between generations.

Image in-between is often implemented using one of two methods: optical flow or convolutional neural networks. However, these methods are typically pixel-based; they do not work well on objects between images far apart. Furthermore, because they either rely on a simple frame average or pixel motion, they do not have the required knowledge of the semantic structure of the data. This work proposes a method for image interpolation based on latent representations. In particular, we present an objective function that controls representation learning. Furthermore, our objective function encourages the latent space to generate appropriate structure to make interpolation in image space possible. We use a simple network structure based on a variational autoencoder. To visualize the role of the proposed approach, we evaluate synthetic datasets. We demonstrate that our proposed method outperforms both pixel-based and a vanilla variational autoencoder.

4.2 Contributions

- We show that we generate a controlled output from the unconditional model.
- Although our task is not reconstruction, we show no tradeoff between reconstruction quality and sample quality. Enforcing such a penalty on the latent space does not sacrifice diversity.
- The generation preserves the structure and properties of the distribution data. In addition, it adds an attribute that is the in-between (spatial information).

4.3 Introduction

The process of generating in-between images from a sequence of images is known as image interpolation. Image interpolation reveals the dynamics of objects in a scene by relating spatial features (i.e., distinct viewpoints) to temporal changes (i.e., different timestamps) [199]. Image interpolation methods are used in various computer vision applications, including the movie and animation industry. It aims to enhance the quality of images displayed in different scenarios. Original videos often have a high frame rate in the digital and movie industry. However, because of the limitations on network bandwidth, the rate has to decrease before transmission. This reduction is often made by skipping some frames[200]. Here, image interpolation can help restore clarity to the image. Some of the challenges in image interpolation occur when the variations in pixel values are significant, i.e., objects in the images vary considerably, overlapping objects, occlusions, missing objects, and noise.

Optical flow [201, 202] and convolutional neural networks (CNNs) [203, 204, 205, 206] are two common approaches for image interpolation based on pixel motion. The former considers the pixel motion of the objects and performs a simple pixel average. The latter learn optical flow feature representations by convolving input images with spatially adaptive kernels that account for pixel motion [207]. Finally, a pixel-based method algorithm generates image interpolation of arbitrary sequences in both approaches. However, when objects in input images are far apart, it may cause problems because the temporal dependence between objects may be lost. The resulting images may appear with holes, overlapping objects, and ghost artifacts.

This work proposes a novel method for the problem of image interpolation based on latent variables. Our model learns to encode the spatial and temporal structure of the image based on latent representations (inherent action) and not image context (pixel motion). The model then generates the in-between image based on the learned representations. Because the model relies on stochastic latent representations of the data, it is not straightforward to assess whether the generated structure is accurate. Therefore, we introduce a loss function that constrains the latent space information capacity to address this limitation.

We use the Variational Autoencoders (VAE) framework to address the challenge of image in-between generations. VAE offers stability during training and the ability to provide meaningful representations. Besides, the latent space allows semantic operations with vector space arithmetic [171]. Furthermore, by introducing the trade-off (hyperparameter) alpha (α) between the average and the teaching signal, the model generates good semantic properties of the in-between image.

In summary, we make the following contributions. First, we design a simple model that relies on latent variables for image interpolation of nonconsecutive images. The model generalizes well to unseen objects (i.e., objects with occlusion or overlap). We demonstrate that constraining latent representations can lead to interpretable data representation.

4.3.1 Problem definition

This work addresses the challenge of controlling representation learning for image in-between generations task. Existing models such as the vanilla VAE may give a structure not aligned with the downstream task. As a result, the generated in-between images may not be realistic or have artifacts. Besides, the generated image may not preserve the structure of the actual in-between.

4.3.2 Variational Autoencoders (VAE)

VAE [97, 98] has shown promising results in various tasks, including image classification [208], image segmentation [175], text generation [23], and artistic applications [209]. The model is composed of the encoder network and decoder network. The role of the encoder network is to map the input data into a latent space distribution, whereas the decoder network maps the latent space representation back to the input.

The VAE framework modifies the deterministic Autoencoder network with a probabilistic one. The latent variable z is sampled from the mean μ and standard deviation σ from a continuous latent space to make VAEs more useful for generative modeling. The μ vector controls where the encoding of the input should be centered, while σ controls the area, i.e., how much the encoding can vary. The decoder learns the data distribution rather than a single point, exposing a wide range of encoding for the same input during training. In addition, VAE models enable random sampling and arithmetic operations on its latent space. Following the general formulation introduced in [97, 98], the VAE loss function (4.1) minimizes the lower bound on the marginal loglikelihood.

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

The first term is the reconstruction error, which measures how well the latent variable describes the image. A pixel-wise quadratic error is often chosen between the actual and reconstructed images. The second term represents Kullback–Leibler's divergence (D_{KL}) between the prior $p(z)$ and the approximate posterior distribution $q_\theta(z|x)$; it assesses the regularization of the latent space, and (θ, ϕ) parameterizes the distributions of the encoder and decoder. VAE aims to generate new samples not present in the training set.

4.3.3 Image Interpolation based on Variational Autoencoders

To connect our work with existing approaches for learning latent representations, we provide practical analysis of vanilla VAE [97, 98] and β-VAE [9]. We attempted to generate image interpolation based on latent representations. We found that the results were not very encouraging, and it did not perform well. The generated image did not resemble the structure of the in-between image.

We empirically assume that the latent space does not have any constraint under its learning representations, and the generated latent variables have certain degrees of freedom. Another possible explanation is that the latent space does not have the necessary structure that enables interpolation. We then designed a network structure to enforce the latent space to have the appropriate structure. Later in this section, we compare our model with these baseline models.

4.3.4 Latent Representations

The data often in high-dimensional space can be represented in a lower dimension, often referred to as latent representations. These latent representations hold relevant information of the initial data, which are highly dependent on downstream tasks [3]. However, these representations are often unstructured and hard to control or interpret [202]; without the pressure to regularize the latent space, they do not exhibit the desired structure [162]. To address this limitation, Higgins *et al.* β-VAE [9] proposed to constrain the latent space capacity, forcing the model to learn salient features of the data, which results in a more interpretable representation of the data. In this work, we demonstrate the benefits of using learned latent representations for image interpolation.

4.4 Proposed Model

4.4.1 Method Overview

The success of image interpolation is restricted to pixel-based approaches. The pixel-based approach works well on consecutive homogeneous images. Because these images are highly similar, they often do not require good knowledge of the semantic structure of the objects. However, when the motion is complex, such as the case of large displacements between objects, pixel-based approaches do not perform well; to restore the in-between image, semantic information is necessary [210]. Based on this insight, we propose a novel approach based on latent variables to the objects' problems in images far apart from each other. The proposed model benefits from the ability to constrain the freedom of the latent representation.

In this section, we begin the discussion by explaining and describing the motivation of our proposed network structure. Then, to improve the performance of the proposed model, we introduce an additional loss function that restricts the latent space to probable structures. We also provide detail of the related hyperparameter.

4.4.2 Proposed Network Structure

Details of the network structure

Our network structure Figure 4.1 follows the vanilla VAE structure [97, 98]. The key components are the Z', which averages the latent space of input images (first image and second image). The α component balances the importance given to the average inputs and actual in-between latent representation. The α term penalizes the network if the generated image has deviated from the actual in-between. Suppose Z' is ignored $\alpha = 0$ (which corresponds to vanilla VAE). In that case, the model is not strongly penalized if the generated in-between does not reflect the actual in-between—giving the model the freedom to sample any possible latent point. This scenario is not ideal; we have to control the latent space if we aim to learn an interpretable representation of the data manifold for image in-between. The effects of α are further explained in this work.

Network implementation

The network structure is based on three variational autoencoders, as illustrated in Figure 4.1. The network receives a pair of images (X_0, X_2) and actual in-between (X_1). Each network has an encoder X, and decoder (X') network, and (z) corresponds to the latent space. To generate the in-between image, we average the latent representations of the adjacent networks(z_0, z_2) and the actual in-between (z_1). To reduce the model complexity, all the networks share the same weights. The weight-sharing technique is a method for building translation-invariant networks [54] and also used for multi-modal knowledge transfer [211, 212].

The encoders have six hierarchical layers, consisting of five convolutional layers and a fully connected layer. In addition, there is a pooling layer with stride two and 4x4 kernels at each hi-

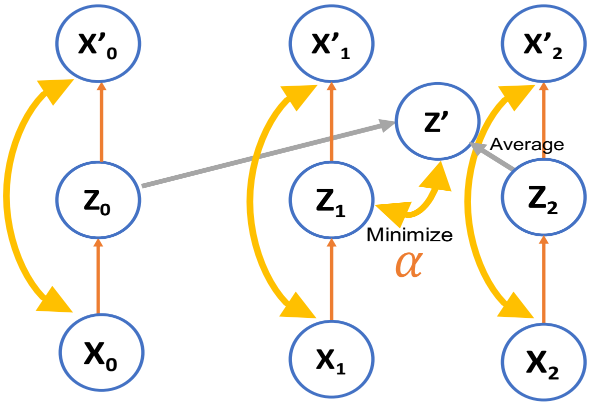


Figure 4.1: Network structure of our approach.

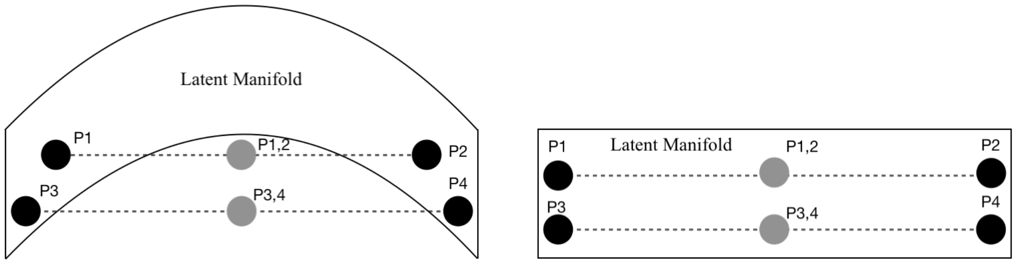
erarchy, except the first layer, which has kernel size 3x3 and strides one. The decoders have six hierarchical layers: five deconvolutional layers and a fully connected layer. Each stride one and kernel size 4x4, except for the last layer, which has kernel size five. We used AdamMax optimization with a learning rate of 0.0001, and the batch size was 100. Later, when we compared our results with FlowNet2.0 and SloMo, we increased the number of layers to 10 since we worked with images of 256x256 instead of 32x32. The learning rate is 0.005, and the batch size of 30. The network was trained to capture salient features from the input data and to minimize the difference between (z') and (z_1).

4.4.3 Proposed Loss Function

We attempted to generate image interpolation based on latent variables of vanilla VAE. However, the results were not expected. The generated in-between image did not resemble the actual in-between image. One reason for that is due to unstructured latent representations. The model is not encouraged to have a structure that reflects the actual in-between image. In addition, the latent representation of vanilla VAE has shown to have limited application in tasks, such as discovering new factors of variation in the data.

This work proposes an objective function (4.2) that is a modification of the vanilla VAE loss function. We demonstrate that this loss forces the latent structure to have the desired property that allows interpolation in the image space. Furthermore, we demonstrate the beneficial effects of introducing the adjustable hyperparameter to improve the downstream task. The idea of introducing the trade-off is not new, Kim and Mnih [104] and Higgins *et al.* [9] have demonstrated the beneficial effects of limiting the capacity of latent representations. This approach forces the model to learn salient features of the data. We limit the information capacity of latent space to generate the actual structure of the in-between image. Finally, we demonstrate that with the proposed loss function, the model generates the actual structure of the in-between image for any given sequence of images.

$$L(X_0, X_1, X_2) = L_{VAE}(X_0) + L_{VAE}(X_1) + L_{VAE}(X_2) + \alpha(D(q_{(X_1)}||\frac{q_{(X_0)} + q_{(X_2)}}{2})) \quad (4.2)$$



(a) Vanilla VAE

(b) ours

Figure 4.2: Vanilla VAE is likely to interpolate in a curved latent manifold. Our model forces the manifold to be locally flat, resulting in smooth interpolation.

A Loss for enforcing flat manifold

Often probabilistic models depend on the way we constraint the learning representations. In Figure 4.2, we show the task of interpolating between two points (P1 and P2; P3 and P4). The vanilla VAE approach often generates a curved manifold, as shown in Figure 4.2 (top). The task becomes complex because the manifold is curved, and the generated point lies off the data manifold (P1,2 and P3,4). Linear interpolation traverses the shortest path regarding the Euclidean distance between the two points. The generated in-between is more likely to be unrealistic. On the other hand, our objective function forces the manifold to be locally flat, as shown in the Figure 4.2 (bottom). Then, interpolation between two points on flat manifold lies on the manifold, and the generated samples from interpolated representation (such as P1,2 and P3,4) will be more plausible. Bengio *et al.* [213], Verma *et al.* [214], have explored the relationship between interpolation and flat data manifold in the context of representation learning.

Adjustable Hyperparameter alpha (α)

The vanilla VAE ($\alpha = 0$) [97] latent information did not learn the structure of the in-between image due to a lack of constraint on the latent information bottleneck. As a result, there was no signal to the model to generate the structure of the in-between image. To learn the latent space representing the structure of the in-between image, we hypothesize that it is relevant to tune ($\alpha > 0$). Alpha balances the relative importance of the difference between ground truth loss and average loss. Alpha ($\alpha > 0$) places a stronger constraint on the latent bottleneck, unlike the vanilla VAE. This ($\alpha > 0$) limits the capacity of latent space z, which, combined with the pressure to maximize the loglikelihood of the training data, and encourages the model to learn the most salient representations of the data [9]. Because the data are generated using some conditional independent ground truth and Kullback–Leibler's divergence term of the loss function, this encourages conditional independence, and higher values of α should promote learning. While tuning α, two factors must be considered: the latent dimension and the complexity of the dataset.

4.5 Experiments

This section presents the datasets and the scenarios tested with individual results and evaluations. We also expose the effects of the hyperparameter and the gains of our proposed method.

4.5.1 Dataset and Degrees of freedom

We relied on a collection of synthetic images for clear visualization of the intended image interpolation result, namely dots, face, teapot, and 2D shapes. These datasets allowed us to create and replicate various possible scenarios. First, training samples were obtained, by randomly sampling 10000 triplets of nonconsecutive images (large displacement between objects in input images), with 10 to 40 degrees from one image to another, and testing random 5000 triplets with 30 to 60 degrees from one image to another. The range prevents the use of consecutive images that are visually very similar. Additionally, we hypothesize that the model does not memorize the training sequence by randomly sampling a triplet. Finally, we do not control the angles between the first and second images. The initial samples consisted of 32x32 image size, except when comparing our approach to Super SloMo and FlowNet2.0. Here, we normalize to 256x256 image resolution. Primarily, we tested ”one degree of freedom” where the object is rotated 360 degrees on the x-axis and then on ”two degrees of freedom” where the object rotates 360 degrees on the x-axis and y-axis.

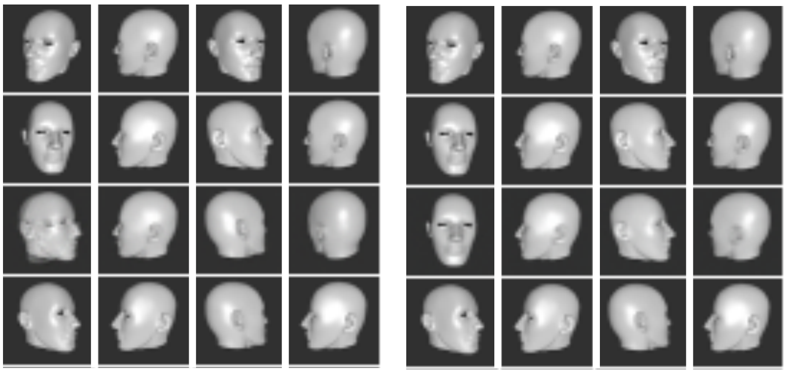
4.5.2 In-between Image Generation

Our model was evaluated far apart images (large displacement between objects in input images). We initially tested image interpolation based on the latent space of vanilla VAE ($\alpha = 0$). There was no constraint applied to the model learning representation. The results show that without limitation ($\alpha = 0$), the generated image interpolation did not preserve an accurate structure of the actual in-between image.

We then applied a constraint to the latent space representation by tuning an adjustable hyperparameter. If tuned ($\alpha > 0$), the model could generate an image that preserves the in-between image's structure. The results demonstrate that our proposed method outperformed vanilla VAE on image interpolation. This is explained by constraining the latent space encourages the model to learn the more salient structure of the in-between image. Next, we show the interpolation results for different scenarios.

One Degree of Freedom

We demonstrated two examples using one degree of freedom. This example represented a simple scenario, with a total of 360 possible angles. The goal was to test the structure of the in-between image (location, angle). As shown on the right side of Figure 4.3 and Figure 4.4, our proposed model preserved the structure of the in-between image in every scene illustrated in the images. This was opposed to vanilla VAE, which failed to preserve the structure of the in-between image.



(a) $\alpha = 0$,epochs=500 (b) $\alpha = 10$,epochs=500

Figure 4.3: Face-testing: 1st row:first frame, 2nd row: ground truth, 3rd row: in-between image, 4th row: second frame. a) The vanilla VAE model failed to preserve the structure of the in-between image. b) Our model generated an accurate structure of the in-between image.

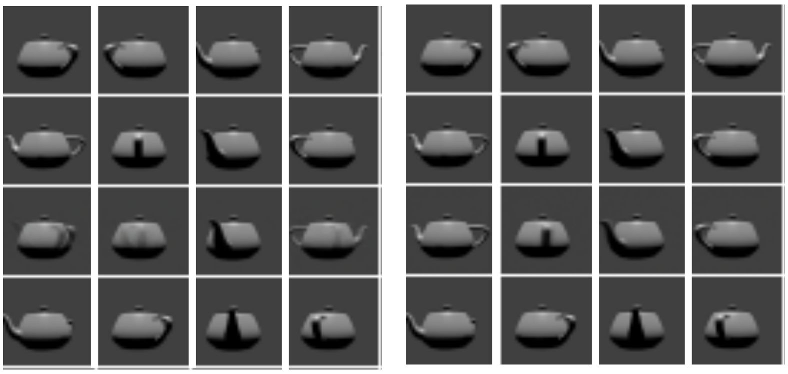
Two Degrees of Freedom

In the next phase of the experiment, we randomly rotated the object under the influence of two variables: ”two degrees of freedom.” In the previous experiment (one degree of freedom), there were only 360 possible scenes, regardless of the number of samples. Working with two degrees squares the number of possible scenes. Then, we randomly sampled the input images to ensure that the model did not see a scene twice. The results highlighted in Figure 4.5 demonstrate that our approach ($\alpha = 10$) preserved the structure of the in-between image, even in a complex scenario.

Moving 2D shapes - Multiple Objects Interpolation

To assess whether our model could generate interpolation in case of multiple objects in the image. We created new training data. Moving 2D shapes is a dataset containing three objects (moving randomly): a white square, a red triangle, and a blue circle. These data are similar to what we can expect in the real world, where different people and objects are moving in random directions. The model must capture the objects’ location, shape, and color. This example represented a more complex scenario since the model has to match similar shapes and colors during the interpolation. One particularity of these data is that small variation (motion) between the objects in the input image cannot be easily noticeable by human eyes. Figure 4.6 shows the results on both vanilla VAE ($\alpha = 0$) and our proposed model ($\alpha = 100$). vanilla VAE failed to generate in-between objects. Additionally, when the objects were displayed, it did not preserve the structure of the in-between image.

Despite the data complexity, our model preserves the accurate structure of the in-between image. Furthermore, the model matches the shape, color, and location even when objects overlap. We highlight the advantages of our model compared to vanilla VAE, as illustrated in Figure 4.7. Restricting the latent space information encourages the model to preserve the semantic structure of the in-between image.



(a) $\alpha = 0$,epochs=500 (b) $\alpha = 10$,epochs=500

Figure 4.4: Teapot-testing: 1^{st} row:first image, 2^{nd} row: ground truth, 3^{rd} row: in-between image, 4^{th} row: second frame. a) The vanilla VAE model failed to preserve the structure of in-between image. b) Our model generated an accurate structure of the in-between image.

4.5.3 Evaluation

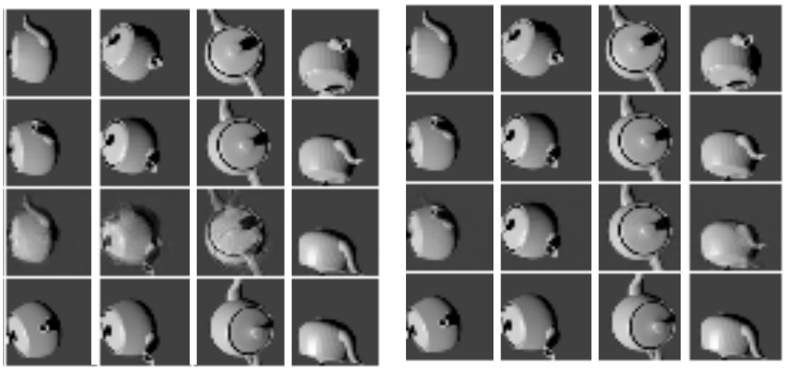
We have so far focused on demonstrating interpolation abilities; in this section, we evaluated our results.

Qualitative Evaluation of Learned Representation

We evaluated the embedded structure of learned representations using two vanilla approaches, principal component analysis (PCA) and T-SNE [215]. PCA is used to reduce the data dimensionality while preserving the variations [216]. T-SNE preserves the metric properties of the original high-dimensional data. In addition, it preserves the information indicating which points neighbor each other [217].

We found that our model effectively showed a consistent loop when projecting the latent representations z learned by the model using TSNE. In contrast, latent representation produced by vanilla VAE preserved the distance in the data but did not preserve the structure of the input images (Figure 4.8). While using PCA, we found that our model preserved the input data structure. Vanilla VAE did not preserve the structure of the input dataset. From its definition, PCA preserves the variation in the data. Two neighboring points in the high dimension should be closer in the low dimension space. Vanilla VAE ignores the variance in the data, while our model keeps the fundamental structure of the input data (Figure 4.9).

β-VAE. We trained β-VAE [9] with different values of β; we found it to have the same behavior as vanilla VAE. β-VAE does not have the necessary structure to generate the latent space that resembles the in-between image. We demonstrated the latent representation in Figure 4.10, and the results on TSNE suggest that vanilla VAE and β-VAE might generate the structure of the in-between image if some form of penalty was imposed in the latent space or input signal is given to the model.



(a) $\alpha = 0$,epochs=500

(b) $\alpha = 10$,epochs=500

Figure 4.5: Teapot-testing: 1^{st} row:first image, 2^{nd} row: ground truth, 3^{rd} row: in-between image, 4^{th} row: second frame. a) vanilla VAE model failed to preserve the structure of the in-between image. b) Our model generated an accurate structure of the in-between image.

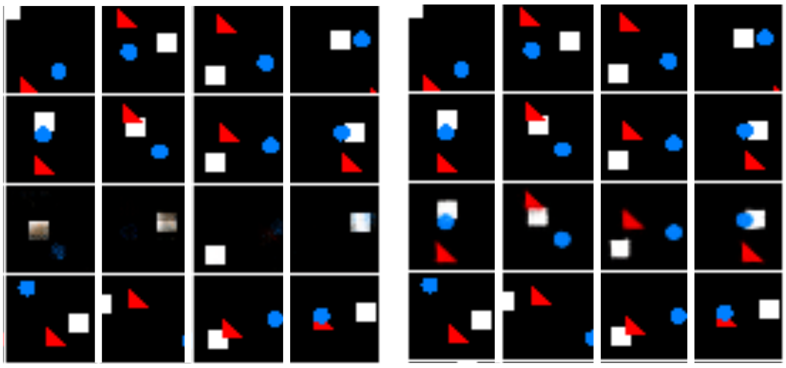
Comparison with state-of-the-art methods - Large Displacement

Quantitative Evaluation. This work lies between image interpolation and latent representations. Since existing works on latent representations focus on disentangled representations, we cannot compare them. The objective of this work is to generate an in-between image based on latent variables. In disentangled representation work, there is an assumption about the number of hidden variables presented in the data, and the data are often arranged to prove this assumption. We did not arrange the training data to disentangle the factors of variations present in the data.

We compared our approach with state-of-the-art approaches on image interpolation based on optical flow and neural networks, including Super SloMo [206], FlowNet [204] and a vanilla VAE. To evaluate the error between the actual in-between and predicted image interpolation, we follow some baseline metrics presented in [218], including the peak signal-to-noise ratio (PSRN), structural similarity index (SSIM), L2, and L1 scores. In Tables 4.1,4.2 and 4.3, we demonstrate the performance of FlowNet and its versions, SloMo, vanilla VAE and our approach. In Table 4.2 and Table 4.3, we used the face and dots datasets respectively. Our model achieves the best performance on all metrics. Despite good accuracy on all metrics, in Table 4.1, for PSRN and L2, our model presents values slightly lower than FlowNet2.0 and FlowNet2S. The performances of our model indicate a plausible generalization capability for distinct datasets.

Visual Evaluation. We compare our approach with two state-of-the-art works on image interpolation based on CNN and optical [206] and latent representation learning [97]. Our model achieved the best performance, particularly where the object is facing and produces fewer artifacts (Figure 4.11). We highlight in a yellow box the errors presented by other models. Optical flow-based methods seem to have more problems with large displacement. It generates the in-between; however, the image resembles one of the input images, not the actual in-between, as illustrated in the figure.

Vanilla VAE does not capture the direction of the object and presents some artifacts in the generated image. One explanation is that learning from a pixel-based approach does not allow



(a) $\alpha = 0$,epochs=2000 (b) $\alpha = 100$,epochs=2000

Figure 4.6: Moving 2D shapes: 1^{st} row:first image, 2^{nd} row: ground truth, 3^{rd} row: in-between image, 4^{th} row: second frame. a) vanilla VAE failed to preserve the spatial location of the objects. b) Our model preserved the structure of the in-between image.

Table 4.1: The results on ”Teapot” dataset

	PSRN	**SSIM**	**L1**	L2
FlowNet2CSS	64.10	0.0047	0.126	0.025
FlowNet2CS	62.37	0.0018	0.166	0.038
FlowNet2SD	54.09	0.0003	0.414	0.254
FlowNet2S	60.01	0.0010	0.180	**0.001**
FlowNet2C	60.39	0.0862	0.113	0.060
FlowNet2.0	**77.13**	0.3873	0.023	**0.001**
Super SloMo	69.97	0.9077	0.020	0.007
VAE	69.15	0.8442	0.023	0.008
Ours	73.19	**0.9078**	**0.015**	0.003

predicting large motion since it does not learn the embedding representations of the data.

Impact of Degrees of Freedom - Additional Evaluation Using MSE. To learn more general data representations, we argue that it is essential to introduce diversity in the training samples. The model is assessed on different degrees of freedom using the mean squared error (MSE). The primary objective is to evaluate the complexity of the datasets, both on the degree of freedom and generalization. The same object is evaluated in two scenarios, one and two degree(s) of freedom: the same epochs, coefficient (α), and latent dimension (z). Figure 4.12 indicates that two degrees of freedom represent a more complex scenario. To generate a plausible in-between image in one degree of freedom, $\alpha = 5$ and $epoch = 1,500$ are required, whereas $\alpha = 100$ and $epoch = 2,000$ are required for generating a suitable in-between image in two degrees of freedom. These results are due to the differences in the number of possible scenarios between one degree (360) and two

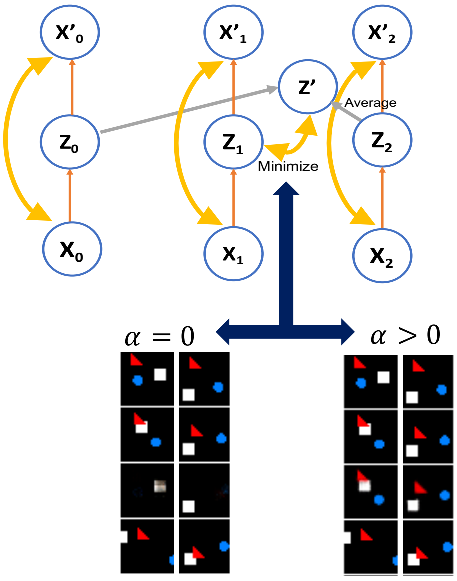


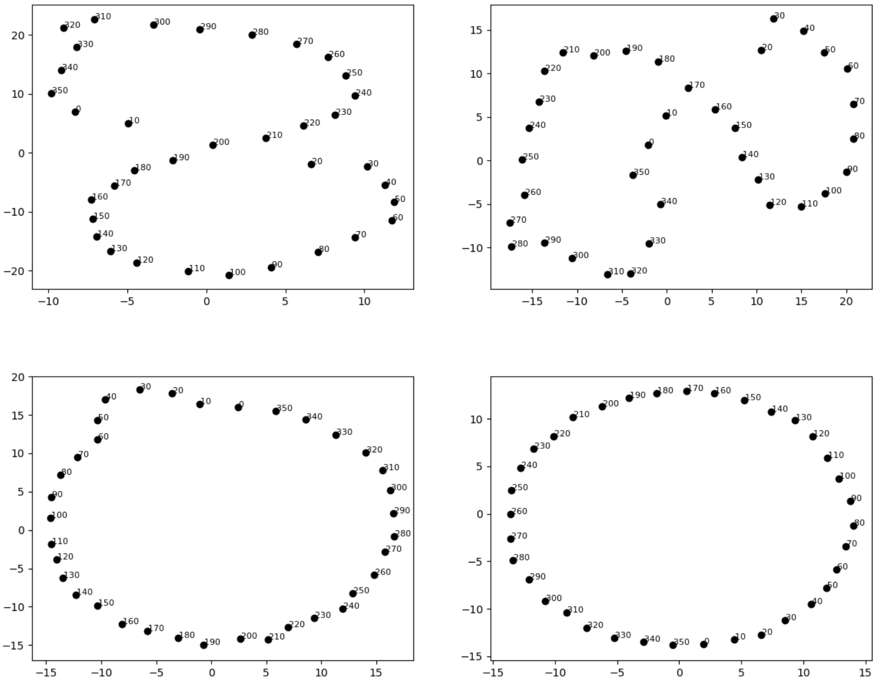
Figure 4.7: The effects α on the in-between image: For $\alpha = 0$: The generated in-between image lacked the structure (location of the moving object) of the in-between image. For $\alpha > 0$: The contextual structure of the in-between image was preserved.

Table 4.2: The results on "Face" dataset

	PSRN	**SSIM**	**L1**	L2
FlowNet2CSS	57.80	0.0015	0.292	0.108
FlowNet2CS	57.80	0.0005	0.374	0.175
FlowNet2SD	54.19	0.0004	0.406	0.248
FlowNet2S	60.44	0.0029	0.198	0.059
FlowNet2C	59.17	0.0005	0.196	0.079
FlowNet2.0	64.00	0.1538	0.068	0.026
Super SloMo	70.91	0.7653	0.032	0.005
VAE	74.04	0.8087	0.018	0.003
Ours	**75.46**	**0.8276**	**0.016**	**0.002**

degrees (360x360).

Impact of Latent Dimension on Different Degrees of Freedom. Latent variables are compressed representations (salient features of the data) of high-dimensional data. In VAE, the latent variables can be found in the bottleneck layer. Depending on the number of variables passed, the output quality might change. To date, the results have been assessed on a single latent dimension $(d_z) = 10$, except for "moving 2D shapes. As shown previously, the decoder can reconstruct the

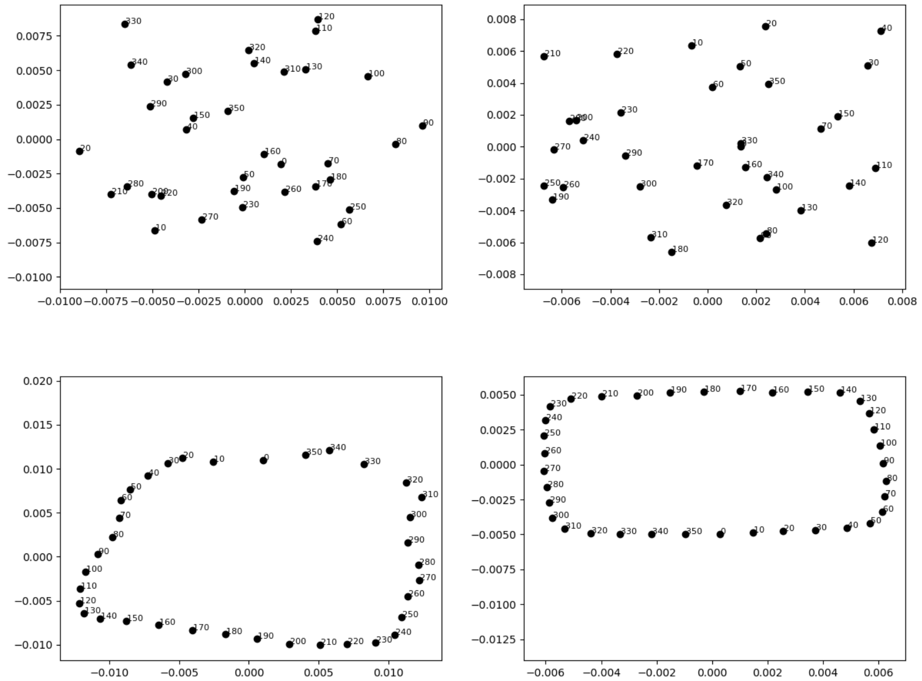


Epochs=50 Epochs=300

Figure 4.8: Projection of the latent representation z after training using T-SNE. Each dot represents an angle with a corresponding value from the face dataset. First row: VAE $(\alpha = 0)$, which shows a loop but does not have a consistent loop structure. Second row: Our model $(\alpha = 10)$ shows a consistent loop since the dataset used for testing has angles ranging from 0 and 360 degrees.

Table 4.3: The results on ”Dots” dataset

	PSRN	**SSIM**	**L1**	L2
FlowNet2CSS	59.87	0.0003	0.219	0.067
FlowNet2CS	57.08	0.0002	0.127	0.127
FlowNet2SD	56.61	0.0007	0.241	0.142
FlowNet2S	60.64	0.0032	0.148	0.056
FlowNet2C	58.87	0.0030	0.224	0.084
FlowNet2.0	70.71	0.3338	0.033	0.006
Super SloMo	50.58	0.9396	0.130	0.570
VAE	64.853	0.863	0.024	0.021
Ours	**72.221**	**0.9437**	**0.010**	**0.004**



Epochs=50

Epochs=300

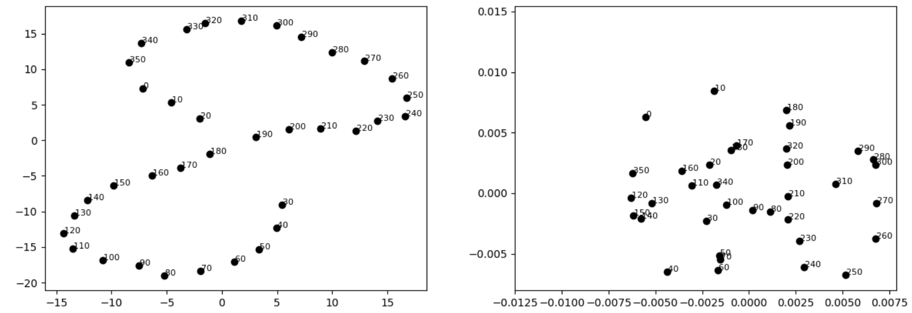
Figure 4.9: Projection of the latent representation z after training using PCA. Each dot represents an angle with a corresponding value from the face dataset. First row: VAE ($\alpha = 0$) ignores the variance in the angle. Second row: Our model ($\alpha = 10$) shows a consistent loop since the dataset used for testing has angles ranging from 0 and 360 degrees.

output, with only 10 variables passed to the bottleneck.

We investigated the impact of the latent dimension on different degrees of freedom using ”moving 2D shapes”. The model was trained for 5,000 epochs with different latent dimensions (1 to 100). The model stabilized on latent dimension z=20, as illustrated in Figure 4.13. For good generalization, passing 20 variables to the bottleneck could be sufficient. The decoder may be able to reconstruct the output.

4.5.4 Linear Latent Space Interpolation

Autoencoders can generate a semantically meaningful combination of features from two distinct data points. David *et al.* [102] have explored autoencoders in the context of regularization to improve linear interpolation. Ideally, latent variables of the data are close to each other but different. This characteristic enables smooth interpolation and stimulates creative design [16, 21]. Sampling latent variables through arithmetic operations can generate diverse outputs. [213] suggests that models that preserve smooth interpolation between points might be relevant for disentangling explanatory factors of variation in data. Another critical application of continuous linear latent interpolation



TSNE

PCA

Figure 4.10: β-VAE shows the same behavior as vanilla VAE. The latent representation lacks the structure to generate the in-between image.

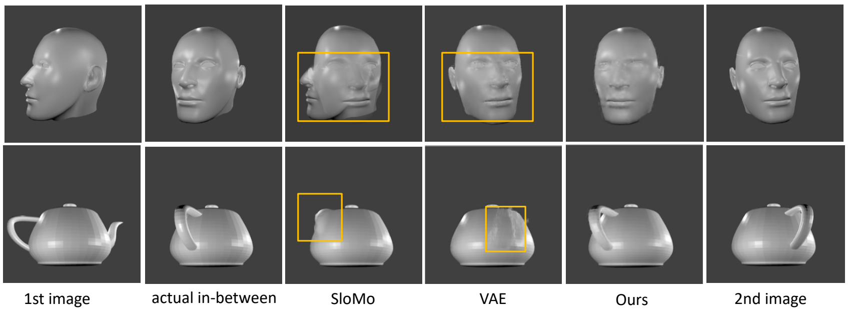
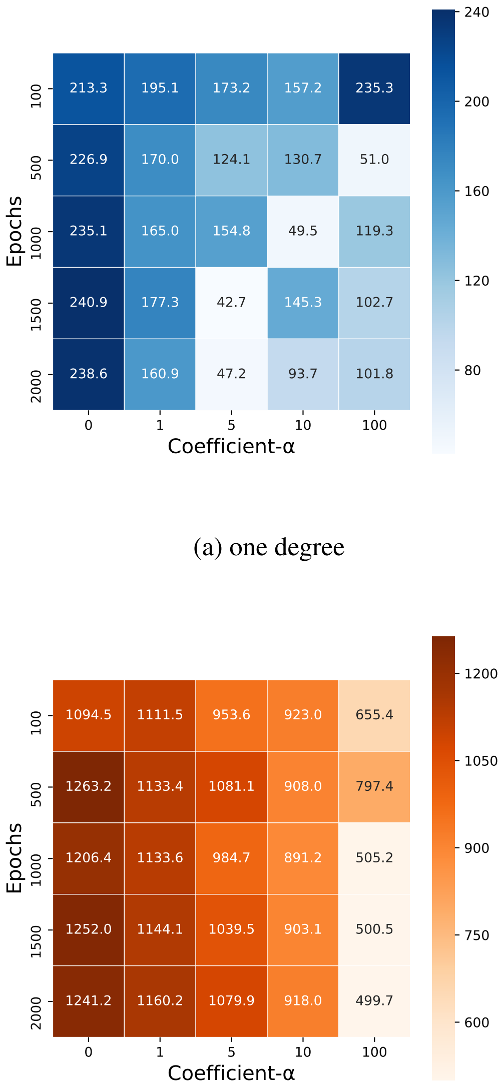


Figure 4.11: Illustration on teapot and face dataset. Our model produces plausible in-between and less artifacts around the object shown in yellow box.

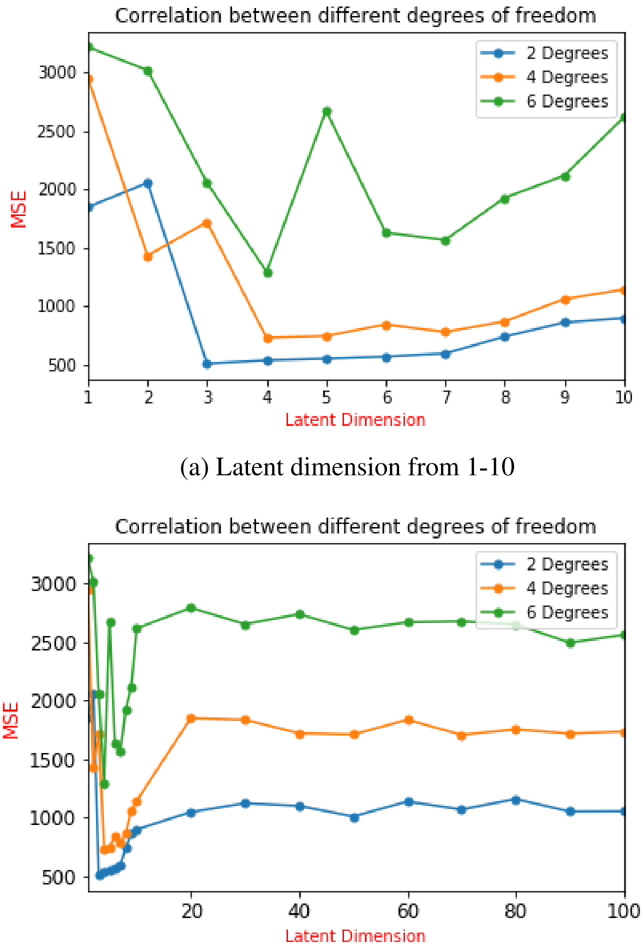
is to test if the model has not merely memorized the training data. By decoding the latent space of two data points, it is possible to visualize a smooth change from one image to the next, as illustrated in Figure 4.14.



(a) one degree

(b) two degrees

Figure 4.12: a) MSE loss for one degree of freedom. b) MSE loss for two degrees of freedom. For one degree, the minimum loss for coefficient $(\alpha) = 5$ and $epochs = 1,500$. For two degrees, the minimum loss for coefficient $(\alpha) = 100$ and $epoch = 2,000$. This indicates a higher complexity of two degrees of freedom.



(a) Latent dimension from 1-10

(b) Latent dimension from 10-100

Figure 4.13: Effects of latent dimensions using MSE. There are two degrees, four degrees, and six degrees. The model begins stabilizing on the latent dimension $z = 20$. For this specific dataset, with 20 dimensions, the decoder could reconstruct the output.

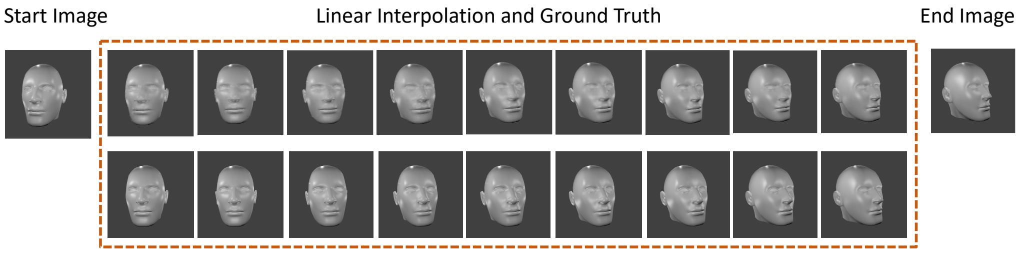


Figure 4.14: Continuous linear latent space interpolation.

4.6 Discussion

There are two main lines of research relevant to our work. The first is similar to [203, 206, 204], and seeks to generate image interpolation based on a pixel-based approach. The second line is similar to [9], which revolves around seeking to learn controllable and interpretable latent representations of data. Of particular relevance to our work are approaches that explore latent space in the context of learning representations. Several works on (unsupervised) learning representations are based on VAE. Prior works [9, 104, 100, 162], enhance the quality of learned representation by modifying the vanilla VAE objective function. These works often considered controlling the level of regularization of the latent space through KL divergence at the cost of reconstruction.

KL divergence allows the model to normalize and smoothly interpolate the latent space [219]. However, if not well-tuned, KL divergence can also induce the network model to a suboptimal [220]. The model does not exploit all the latent variables for generation, the so-called over-pruning/variable collapse discussed in [221]. Placing importance in the KL divergence term leads to a more controllable latent space, which may lead to a better quality of generated samples.

A state-of-the-art study on unsupervised disentanglement representations β-VAE [9], gave relative importance to the KL divergence term by introducing a hyperparameter β to the VAE loss function. The authors argued that this modification encourages the model to learn interpretable representations of the data. In the same line of research, [177] enhanced β-VAE by modifying the training process. The authors claimed that increasing the information capacity of the latent codes during training enables the model to see more factors of variations continuously, thus resulting in better disentanglement. Our objective function is similar to β-VAE, but we do not aim to disentangle factors of variation in the data.

A different path to learning latent representations was taken by Chen *et al.* [96]. The authors proposed InfoGAN, a model based on a generative adversarial network (GAN). The model encourages disentanglement by penalizing the total correlation[222], i.e., the mutual information between the data and latent representation. Disentangled representation models have been shown to discover factors of variations in the data; the application is still restricted to a synthetic dataset. Locatello *et al.* [223] argued that disentangling a specific factor is nearly impossible without any forms of inductive bias on both the model and the data. Furthermore, the authors were not clear about the relevance of disentanglement for downstream tasks.

4.7 Related Work

4.7.1 Image Interpolation Methods

Research on image interpolation(motion flow, disparity, displacement) has a long history in computer vision. Two directions have been explored: optical flow and, more recently, convolutional neural networks (CNNs).

Optical flow

Initial attempts at image interpolation were based on optical flow methods. Optical flow is used to describe the apparent shifting of pixel values in time-varying images caused by illumination change, camera motion, or noise. Optical flow techniques compute the motion estimation vector for each pixel or group of a pixel in an image, and this involves having an initial image and at least one of its neighbors.

A large part of the work on image interpolation is based on differential algorithms proposed by Lucas and Kanade [224], and Horn and Schunck [225]. These algorithms are based on several assumptions, such as brightness constancy, and temporal consistency [226]. Lucas and Kanade assume that pixels surrounding a pixel being observed behave almost the same as the observed pixel (local variation), while Horn and Schunck consider the global variation in an image. This assumption means that the motion vectors of a pixel depend on the value of its neighbors. Since these algorithms are based on differential methods, temporal smoothing between images is necessary to avoid aliasing caused by the significant differential between pixels.

To overcome the limitations of traditional methods two main directions have been explored, including motion-compensated frame interpolation (MCFI) techniques [227, 201, 228, 229] and phased-based methods [230, 231]. The former estimates the motion based on the previous and current images. It then generates the in-between by averaging the pixels pointed by half of the obtained motion vectors. MCFI is based on assumptions that motion in images is smooth and continuous, which might work well on sequences with relatively small motions [227]; on large displacement, residual information of skipped frames is unavailable, and the generated image might include overlapped objects, holes, and blocking artifacts.

In the second direction, phased-based methods assume that small motion can be encoded in the phase shift on an individual pixel's color. Meyer *et al.* [231] suggested extending flow-based methods to the path-based method; by using a path-based method, the motion accuracy was expanded, improving the range of the motion trajectory. Alternatively, Zhang *et al.* [232] extended the motion range by computing a disparity map, while Elgharib *et al.* [233] proposed combining a phased-based method with optical flow. These methods have primarily improved the performance over differential algorithms but still cannot handle large displacement.

Convolutional Neural Networks

Neural networks have achieved state-of-the-art performance in various applications. Recently, researchers have shown interest in applying CNNs for the task of image interpolation [234, 235, 236, 237, 238, 239]. CNNs are well-known algorithms for extracting semantic knowledge from data. They learn the optical flow feature representations by convolving input images with spatially adaptive kernels that account for pixel motion. Dosovitskiy *et al.* [203] proposed two CNNs (FlowNetS and FlowNetC), which estimated the optical flow based on the U-Net denoising autoencoder [121]. The Dosovitskiy *et al.* model takes an input pair of images and outputs the flow field. The image interpolation results have significant errors in the backgrounds.

Alternatively, Ilg *et al.* [204] suggested combining deep learning with domain knowledge; their model has a small network concentrating on small motion and others on large motion. Jiang *et al.* [206] extended a single image generation to multi-images. Shu *et al.* [240] trained their age progression model with paired images of the same person with different ages. Although the training dataset is similar to our approach, their goal is to train the age progression dictionary, while our interest is to have better latent representation. Interpolation tasks using neural networks have been extended to text[23] and video [239, 206, 207, 241].

Despite the excellent performance, pixel-based methods rely on pixel motion. They are limited to highly similar images. They do not perform well on objects in images that are far apart (large displacement between objects in input images). Because the input images that are far apart may lose temporal dependence between objects, they do not have the required knowledge of the semantic structure of the input images. Thus, the generated in-between image may appear with some errors, such as occlusion, overlapping, and ghost artifacts. CNN models alleviate the problems of pixel-based models to some extent. In this work, we propose a novel method for the problem of image interpolation based on latent variables.

4.8 Summary

This work presented a simple approach to improving image interpolation. Our model produces good performance on all datasets. In addition, the model outperforms some baseline approaches on large displacements between images. The key to the success of this approach is dedicated to latent variables. Learning latent representations of the data and limiting the freedom of latent space has been demonstrated to impact the generated in-between image structure. Previous works are pixel-based except vanilla VAE; however, VAE does not control generated in-between. Therefore, we propose a model that can control the latent space. Future direction includes applying this approach on a smaller dataset and more challenging pedestrian and road traffic datasets.

Chapter 5

Regularizing Representation Learning for Few-Shot Learning Image Classification Using Variational Information Bottleneck and Layer Selection

5.1 Overview

We have so far focused on evaluating representation learning in generative learning models, particularly in the image in-between generations. Next, we assess whether improving the quality of representations is associated with improved performance accuracy on few-shot learning tasks. Specifically, we will evaluate whether enforcing regularization by placing a penalty in the latent space enhances the accuracy of novel data in Few-Shot Learning models and whether selecting representations from a good layer improves generalization. Few-shot learning remains an open issue in computer vision. Among several recently proposed approaches, Weight Imprinting (WI) achieves superior performance on many challenging benchmarks. The imprinted weight models' performance heavily relies on the quality of the representations generated by the encoder. However, it is not known what characteristics are required for weight imprinting. The representations learned optimized for the base classes might not necessarily be ideal for the novel downstream task.

In this work, we introduce two methods. First, we introduce Variational Information Bottleneck (VIB) loss in the few-shot learning with weight imprinting models. The objective of Variational Information Bottleneck is to regularize the latent representation by minimizing the mutual information between input data and representation while keeping the classification accuracy for the pretraining task. We demonstrate that the encoder regularized by Variational Information Bottleneck achieves significantly better performance on few-shot learning tasks with imprinting. Furthermore, we comprehensively investigate the effect of combining Variational Information Bottleneck with other regularization methods, including data augmentation and auxiliary data. We confirmed that we could achieve better accuracy on the downstream task with an auxiliary dataset.

Second, we introduce a good layer selection method. The quality of transferred representations affects the performance accuracy of novel classes. The objective of layer selection is to select a good layer to transfer the information. We propose transferring features from task-agnostic layers. We introduce projections heads in weight imprinting network architecture to achieve this objective. We demonstrate that imprinting from task-agnostic representation generalizes better than task-specific layers.

5.2 Introduction

Few-shot Learning (FSL) is a machine learning paradigm that learns from limited examples. It has achieved state-of-the-art performance on many benchmark tasks such as image classification. One prominent approach to the FSL problem is the weight imprinting (WI) paradigm. A network directly sets the final layer's weights for a few novel classes from the embeddings of the base training data. However, the outstanding performance of WI heavily relies on latent representations. The main challenge is how to seek an appropriate feature representation so that the classifier can classify novel data correctly. One assumption is that a strong regularization is required to have such appropriate representation. Embedding, Meta-Learning, and Hallucination methods attempt to enforce such strong regularization. Embedding approaches provide suitable representations for FSL since this method learns the similarity between representations. Moreover, novel classes can be categorized by performing a distance metric on the representation. Meta-Learning approaches construct a meta learner to improve adaptive capacity. Hallucination methods are a naturally practical and straightforward approach to enforcing diversity in the input space. By increasing the diversity might force the model to generate different representations.

Existing approaches [141, 242, 243, 244, 245] on WI have two significant limitations: objective function and the choice of the encoder. First, the objective function, often cross-entropy, and nearest neighbor do not focus on the latent representations. The model is trained to perform a stochastic optimization step to minimize the objective function between the latent representation and the actual labels. However, there is no constraint on the actual representations. The objective function does not force the encoder to encode relevant features and discard irrelevant input information.

Second, existing models use a deterministic encoder, and a single image is mapped to a single latent representation. There is a risk of sampling from an empty manifold. The classifier might give a lower probability to representations coming from this empty manifold. In addition, since the performance of imprinted models depends on the quality of representation created by the encoder, it is not known what characteristics are required to suit these models. For example, the representations learned might lead to the highest classification accuracy on base classes; however, the classification accuracy might be degraded on the novel classes.

In this work, we introduce two solutions for the same problem. First, we demonstrate the advantage of information bottleneck on FSL for classification problems. We employ a probabilistic encoder rather than a deterministic. We argue that an encoder trained in a probabilistic manner encodes relevant features about the input data. Our objective function constrains the representations, i.e., it works as a latent regularization loss that helps to learn valid representation while avoiding overfitting. We, therefore, propose a previously unexplored information bottleneck in the context of FSL.

Our framework merges the design choice of weight imprinting [141], and variational autoencoders models[97, 166]. We subsume previous work, especially on the evaluation criteria. We further improve classification accuracy by extending the training data by employing auxiliary data similar to the related task. Previous works [38, 35, 246] focus on hallucinate the data. Our approach has three stages: 1)pre-train, 2)weight imprinting, 3) classification. During the first stage, the model is pre-trained on base classes, and the model learns the representations of the base classes. In the second phase, we input the novel classes, and the model outputs the prototype for novel labels. The last phase is the testing; we use the union of unseen base and novel classes. Our approach is simple yet outperforms the baseline weight imprinting model.

Second, we provide an efficient training approach for imprinted weight models. We find that a simple design choice of imprinted weights can yield substantial improvements over the baseline model. Our experiments show that (1) introducing a nonlinear projection heads in-between feature extractor, and classifier substantially improves generalization, (2) imprinting from the last projection head does not provide better generalization for novel classes. Instead, we propose imprinting from a good projection head, and (3) this design choice benefits from a large latent dimension. We validate our findings by achieving 5.6 and 4.1$\%$ improvement on the MNIST dataset trained with the Omniglot dataset.

5.3 Problem Definition

We start by defining our downstream learning problem in the Few-Shot image classification. Then, we intend to train a model on one dataset and test on a distinct one. Let (x, y) denote an image and its true label respectively. The training examples and test examples are $D_s = (x_i, y_i)_{i=1}^{N_s}$ and $D_q = (x_i, y_i)_{i=1}^{N_q}$ respectively, where $y_i \epsilon C$ for some set of labels. The number of ways n correspond to the number of classes C. The number of shot k is defined by the number of new samples. The general objective is to train a network to learn a mapping function $F(x, y, \theta)$ by exploiting the training examples D_s. In this work, we address the following issues.

- How to improve the quality of representation learning.
- How to improve the generalization performance on unseen classes.
- How the design choice of imprinted models can affect the final accuracy.
- Deeper projection head can improve the representations for the downstream task.

To address these shortcoming we propose two solutions: First, we introduce variational information bottleneck objective function, and second we propose transferring knowledge from task-agnostic representations.

5.4 Regularizing Representation Learning for Few-Shot Learning Image Classification Using Variational Information Bottleneck

5.4.1 Overview

Existing FSL tasks rely on cross-entropy loss function and, more recently, on self-supervised approaches such as contrastive learning and pretext tasks. These models have achieved outstanding performance. However, generalization is still an issue. The performance degrades when training on novel examples. One work that aims to learn from a few examples is the imprinted weight models. Although its excellent performance, this model heavily relies on the quality of the representations generated by the encoder. Besides, it is unclear what feature representation is relevant for the downstream task in the imprinting phase. For example, representations created by encoder based on base classes might not necessarily be effective for novel categories. Therefore, we propose a new criterion for constraining the feature representation based on the information theory.

5.4.2 Variational Information Bottleneck

The cross-entropy loss employed in imprinted models does not model the latent representation. The objective is explicit, minimizing the misclassification between the network's output and the actual label information. The latent codes z created by the encoder $f(x)$ are encouraged to be maximally informative about the label information y. At the same time, it is encouraged to forget about the input data. We claim that the objective function should explicitly regularize the latent representation. At the same time, the model is explicitly encouraged to capture and maximize the mutual statistical information between latent representation and actual class labels. It also should keep minimal information of the raw input data in its latent representation. The core idea is that having such information embedding in the loss function forces the model to ignore irrelevant information that does not describe the data and focus on relevant features that can help to generalize on unseen classes.

Inspired by the effectiveness of the information bottleneck principle. We propose employing the Variational Information Bottleneck objective function for the low data regime in the classification problem. The purpose of the loss is to regularize the latent representation learning by suppressing irrelevant features that do not contribute to the generalization of novel categories. The information bottleneck loss is expressed in Equation 5.1.

$$L(\theta, \phi) = \mathbb{E}_{z \sim q_\theta(y|z)}[\log p_\phi(y|z)] - \beta KL(q_\theta(z|x)||p(z)) \tag{5.1}$$

Where the first term $\log p_\phi(y|z)$ is estimate the probability of z belonging to class y. This term can be seen as cross-entropy function. The second term $q_\theta(z|x)||p(z)$ is the Kullback-Leibler divergence, that measures the distance between posterior $q(z|x)$ and the prior $p(z)$ β balances the trade-off between classification and the KL divergence.

Tishby et al.[164, 165] defines information bottleneck as one principle for extracting relevant information in input signal $x \in X$ contained about the target $y \in Y$. The relevant information is expressed as the joint information between input X and targets Y, i.e., $I(X;Y)$. The authors also state that the information bottleneck principle is desirable since it defines the trade-off between a useful representation and a good classifier. We show that the cross-entropy function alone is not enough for ensuring good representations through experimental results. The importance of maximizing the mutual information between embeddings and target labels has been highlighted in [166, 167].

5.4.3 Why VIB objective function

It is not apparent what representations are created by the encoder. Baseline models on FSL do not focus on representations. Therefore, there is no constraint on the latent representation to encode relevant features for the downstream classification task. The loss function is designed to penalize the classifier misclassification. However, the cross-entropy loss function alone cannot ensure good representations for unseen classes. Here, we constraint the encoder's representation by imposing a criterion between embeddings Z and target Y and between the target Y and input data X. The equation 5.1 can be rewritten as shown in 5.2.

$$L_{VIB}(\theta) = I(Z, Y; \theta) - \beta I(Z, X, \theta) \tag{5.2}$$

Where, the first term $I(Z, Y; \theta)$, in information theory, maximizes the mutual information between the embeddings Z and the target label Y. The second term $I(Z, X, \theta)$ maximizes the shared statistical information, i.e., the relevant features of input X concerning target Y. In other words, the first term encourages the latent representation Z to be maximally informative about target Y, while the second term encourages Z to encode minimal information about input information X for predicting Y. Variational Information Bottleneck is similar to Variational Autoencoders [97, 184]; however, it is trained for classification rather than reconstruction. Smaller values of β encourage mutual information between Z and Y. Essentially, two assumptions are derived from the probabilistic approach: First, it is assumed that sampling from a distribution avoids sampling from a manifold with no representation. Second, the model is expected to encode relevant features in input data X, which well describes the target Y.

5.4.4 Weight Imprinting

The baseline weight imprinting model was proposed by Qi et al. [141]. The model aspires to mimic human vision capability by recognizing novel classes from one or a few samples. The model directly sets the classifiers weight from novel training samples during few-shot learning. The authors call this process weight imprinting as it sets weights for novel classes based on embeddings from base classes. For instance, considering a single training example from novel category x_+, the weight imprinting model computes the embeddings $\phi(x_+)$. Then, it uses it to set a new column in the weight matrix of a trained classifier by imprinting additional columns for novel classes [141]. Figure 5.1 illustrates the weight imprinting network pipeline. The network has two phases, the pre-training and imprinting phase.

Training Phase. Given training examples (base classes X_b), the network f is optimized to find the similarity between normalized embeddings $f(\theta)$. The learning is achieved by minimizing the cross-entropy loss between the final embeddings and the target label. The classifier C is composed of weight matrix and feature representation $W_b^T f_\theta(X_b)$ followed by a softmax. The training phase aims to learn a proxy embedding of each class.

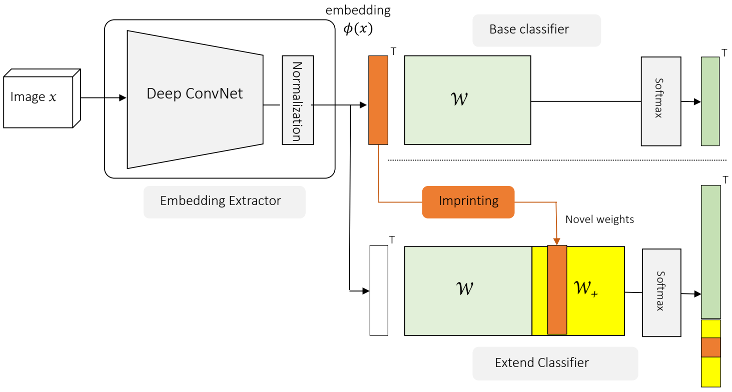


Figure 5.1: Weight imprinting network. The model extends the final layer weight matrix of a trained by imprinting additional columns for novel classes [141].

Imprinting Phase. Given a trained model, the novel class X_n is fed to the imprinting network (fixed network). The size of novel classes is defined as n-way k-shot. Where n-way correspond to the number of classes and k-shot number of novel samples. The output is an embedding vector of each novel class prototype (imprinted weights). At the testing phase, the base weights and novel weights are concatenated and further used for classification. The core assumption of imprinting is that the test samples from new categories are closer to the corresponding training samples [141]. If novel samples $(n > 1)$ are available for novel classes, we average the normalized weights $w_+ = \frac{1}{n}\sum_{i=1}^{n}\phi(x_+^{(i)})$.

5.4.5 Proposed VIB model

Our model architecture differs from the baseline Weight Imprinting . Figure 5.2 illustrates the distinctions between the two. We propose a probabilistic approach using the Variational Information Bottleneck loss function. The loss is similar to the vanilla VAE. However, instead of reconstructing the data, it uses a cross-entropy loss. The input training example is fed to the encoder (feature extractor) $f(\theta)$, where we obtain the embeddings, then we take the mean μ and standard deviation σ from this distribution. Next, we sample the latent representation z that is fed to the classifier. The ultimate goal of Variational Information Bottleneck loss function is to work as an additional regularizer, i.e., to encourage the encoder $f(\theta)$ to ignore irrelevant features to describe the input data.

5.4.6 Experiments - Implementation details and results

The goal of the experiments is to evaluate the effect of variational information bottleneck on weight imprinting models. The implementation details are comparable with the baseline Weight Imprinting method [141]. Although, the encoder (feature extractor) consists of three ConvNets, each followed by batch normalization, ReLU activation function and max-pooling layer, and a classifier. The input images are 28x28, and we applied affine transformation with a degree of 10. The learning rate is 0.001, and we used an adamax optimizer. We tested on different sizes of latent dimensions [16, 32,

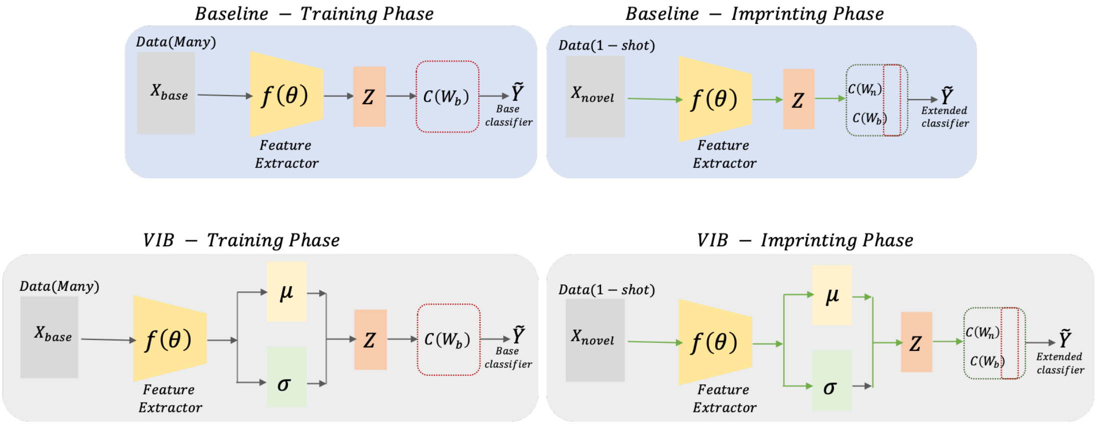


Figure 5.2: An overview of the Baseline and our proposed VIB model. Both models train a feature extractor $f(\theta)$. In the training phase, the base class data is used, while for imprinting where the network is fixed, we use a single image per novel class. VIB differs from the baseline model in its objective function. VIB is trained to constraint the latent representation.

64, 256], and we varied the value of the hyperparameter β from 1e-10 to 1e-1. Furthermore, we apply the mixup, cutout, cutmix data augmentation, which is shown in fig5.6, and auxiliary dataset.

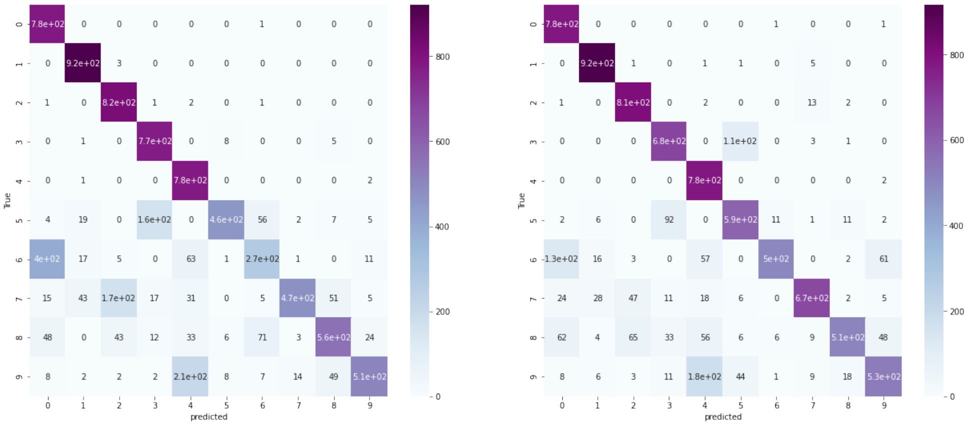
Datasets

We address the FSL problem for the classification task. The main objective is a generalization. We intend to train the model in one source domain and test it on another or unseen categories. We explore the MNIST dataset[247], which consist of 10 classes, relatively balanced. For base classes we use labels $X_b \in \{0, 1, 2, 3, 4\}$, while for novel classes $X_n \in \{5, 6, 7, 8, 9\}$. We randomly sample a single image per label, often defined as 1-shot for imprinting. We evaluate on top-1 classification accuracy using union of base classes and novel class $X_{test} = X_b \cup X_n$ from unseen samples.

Second, we trained on CUB-200-2011 dataset [248] that contains 200 fine-grained classes of birds with 11,788 images. We split the dataset as suggested by [141], 100 classes as base categories and 100 as novel samples, each category contains about 30 images. For imprinting, we used a single image per category for novel examples. We measure the top-1 classification accuracy of the final classifier for all classes.

Results

For this experiment, we use the simple ConvNet with 3 fully connected layers with 32, 64, and 64 hidden units followed by a linear layer and a softmax layer. A single image per label instance is used for imprinting. The test data was not seen during training. Figure 5.3 shows the results of imprinted weight base model ”Baseline” [141] and our approach variational information bottleneck ”VIB”. Our approach improves the baseline model by almost 4%, this accuracy is obtained with 32 latent variables. We also find that using latent variables of 16 with appropriate trade-off outperforms the baseline model. Using the dimension of 64 and 256, the model performs slightly worse



(a) Baseline

(b) VIB

Figure 5.3: Accuracy: The results are for 10 testing sampling. The models are trained with 32 dimensions. a) Baseline: accuracy: 80.01 b)VIB: accuracy:84.40

than baseline. We hypothesize that the VIB model still has some degree of freedom on its latent representation. Next section provide more detailed result of different values of latent variables and the role of the trade-off factor β.

Behavior of the Trade-off β

MNIST-dataset. To better visualize the role of β, we trained the model with different values β from 10^{-1} to 10^{-9} and with latent dimensions of $D \in \{16, 32, 64, 256\}$. Observing the Figure 5.4, we confirm that the performance varies according to β. A smaller value of the trade-off means Z is encouraged to predict Y, while larger values encourage Z to be less predictive about Y. This idea is also expressed in the objective function Eq.5.2 while other works have used this idea for the different downstream tasks. Variational Information Bottleneck [166] proposes to learn an encoding that is maximally informative about the target Y for the classification task, VAE [97] uses for reconstruction while β-VAE [184] proposes for disentangle latent representations. Finding the appropriate trade-off value is relevant for the downstream task. The results show that our model outperforms the baseline model using standard augmentation with appropriate trade-offs and latent dimensions.

CUB-dataset. We perform experiments on CUB-200-2011 dataset. We use 100 categories for training and the remaining is used as novel data. We perform 1-shot classification, i.e., a single image of novel classes is used to obtain the class prototype. Figure 5.5 shows the result for different values of β. Large latent dimensionality contribute for good accuracy. We find that with latent dimension of 1024 and β=10^{-3}, we achieve the best results. The best result is slightly better than our implementation of weight imprinting [141]. The original implementation used the Inception model while we worked with Resnet50. Further, we adapt the pretext task model RotNet [65], that is trained on unsupervised manner. We initial train the RotNet classifier to predict the geometric

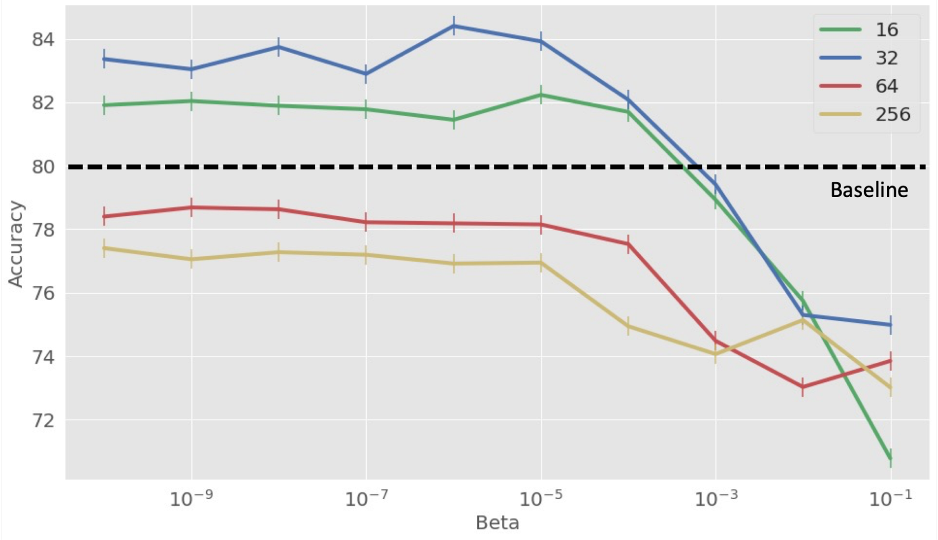


Figure 5.4: Results of VIB on MNIST: Accuracy vs. β. Plotted values are the result of a 10 average test run. The trade-off β clearly affects the behavior of the model. VIB with appropriate hyper-parameters outperforms the baseline model.

Table 5.1: Top-1 accuracy on CUB Dataset for different dimensions of the latent space

Model/Latent Dimension	16	32	64	128	256	512	1024	2048
Weight Imprinting [141]	38.7	45.1	48.4	51.0	51.9	51.5	51.9	**52.8**
RotNet [65]	24.2	34.4	36.4	36.6	42.4	37.4	43.2	45.0
VIB	**39.51**	**45.39**	**50.35**	**51.55**	**52.26**	**52.8**	**53.12**	**52.8**

rotation such as (0, 90, 180, 270) angles. Then, we used the representation for supervised training.

Table 5.1 show the top-1 accuracy of 200-way classification for new samples and all examples in the CUD dataset. Our proposed model performs slightly better than our implementation of the weight imprinting model [141]. The improvement is around 1.22% when the latent space is 1024. However, observing the best results solely, we find 0.2% improvement. The effects of proposed Variational Information Bottleneck significantly improved the downstream task on the previous experiment where we worked we Omniglot and MNIST datasets. However, for CUB-200-2011 dataset, we did not find a similar observation. Therefore, we intend to further research on this matter.

Behavior on Data Augmentation

Data Augmentation has effects on training deep learning models. In particular, it augments the diversity of the training data. When increasing diversity, it is often expected that the model does not see twice the same image. The augmentation prevents the model from overfitting and provides robustness, seeing that the deep model has low bias and high variance. Besides, increasing diversity leads to better generalization. We evaluate our approach and baseline model using recently proposed

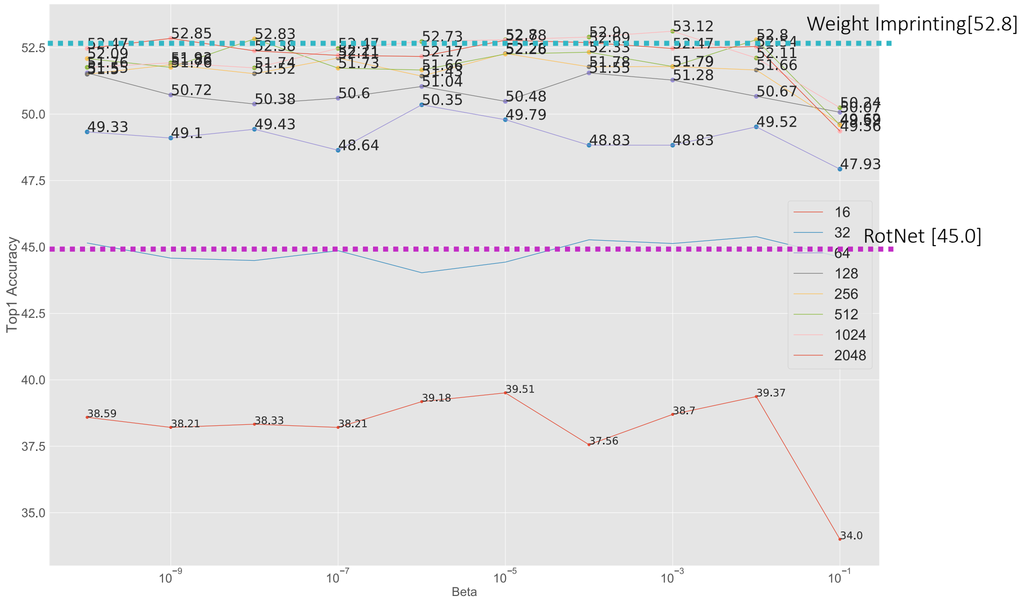


Figure 5.5: Accuracy vs. β. Plotted values are the result of a ten-average test run. The trade-off β affects the behavior of the model. VIB with appropriate hyper-parameters outperforms slightly our implementation of [141] and significantly outperforms [65]

data augmentations which includes mixup [28], cutout [29] and cutmix [1] as shown in Figure 5.6. Table. 5.2 and Table. 5.3 illustrate the results that VIB achieves higher accuracy on Standard and the Cutmix augmentations, while the mixup and cutout show a lower performance. Nevertheless, our model is competitive with the baseline with mixup augmentation using a simple Standard affine transformation with an angle of 20 degrees. This result indicates that these augmentations do not contribute to VIB performance. We argue that this behavior is due to its strong regularization.

Behavior on Auxiliary Dataset

The ability to generalize from limited data is a well-known problem in computer vision. However, it is unclear how to learn rich feature representations. This work proposes an auxiliary dataset similar

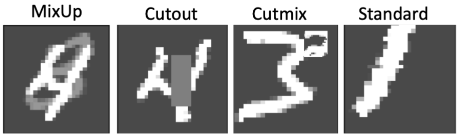


Figure 5.6: Data Augmentation

Table 5.2: Accuracy of Baseline on different augmentation techniques

Epochs	**MixUp**	**Cutout**	**Cutmix**	**Standard**
50	85.53	82.25	41.55	80.01
100	85.96	83.99	42.43	78.31

Table 5.3: Accuracy of VIB on different augmentation techniques

Epochs	**MixUp**	**Cutout**	**Cutmix**	**Standard**
50	79.26	80.17	83.07	84.40
100	79.37	82.12	81.12	81.58

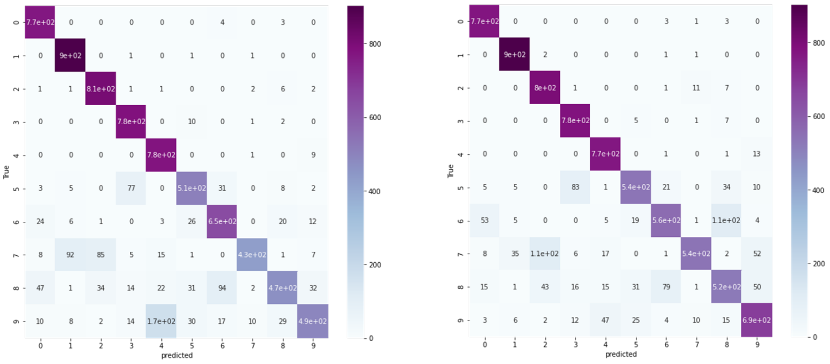
to the task at hand instead of hallucinating additional training samples from the same distribution. We intend to minimize the bias and increase the data variance by using auxiliary data. The model is encouraged to encode different representations. Furthermore, this technique prevents the model from overfitting and increases robustness. Since the downstream task is to classify digits ”MNIST”, we employ the hand-written alphabet dataset ”EMNIST”[249] as additional data. We argue that digits and letters possess common underline representations. The model learns to identify edges and traces that represent the digits and letters. The common property is relevant for improving representations. The result in Figure 5.7 and Table 5.4 show that the VIB model gives a clear improvement on downstream tasks. The classification accuracy is around 4%, and 2% compare to baseline with auxiliary data for 50 and 100 epochs, respectively.

Qualitative Evaluation with t-SNE

We investigate the behavior of VIB and the role of the trade-off factor β on the representations encoded by the encoder. Figure 5.8 shows the latent embeddings for the baseline weight imprinting and VIB. Initially, with a significant value of β, the embeddings for novel classes are densely placed, resulting in overlapping. There is no clear boundary between the categories. This effect might contribute to lower classification at test time since the model can not make sense of distance similarity. Decreasing the value of β, the embeddings become less dense and form a consistent cluster. The similarity between class embeddings is evident. Overall, the results indicate that tuning the trade-off is vital for achieving better representations.

Table 5.4: Accuracy of VIB with auxiliary dataset

Epochs	**Baseline**	**VIB**
50	82.37	86.06
100	82.55	83.81



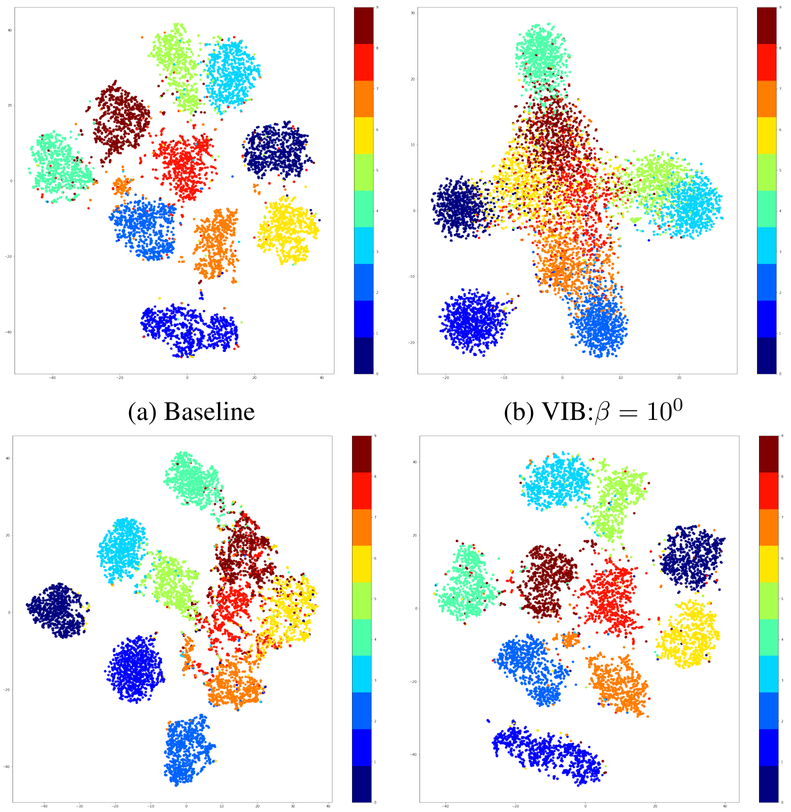
(a) Baseline

True \ predicted	0	1	2	3	4	5	6	7	8	9
0	7.7e+02	0	0	0	0	0	4	0	3	0
1	0	9e+02	0	1	0	1	0	1	0	0
2	1	1	8.1e+02	1	1	0	0	2	6	2
3	0	0	0	7.8e+02	0	10	0	1	2	0
4	0	0	0	0	7.8e+02	0	0	1	0	9
5	3	5	0	77	0	5.1e+02	31	0	8	2
6	24	6	1	0	3	26	6.5e+02	0	20	12
7	8	92	85	5	15	1	0	4.3e+02	1	7
8	47	1	34	14	22	31	94	2	4.7e+02	32
9	10	8	2	14	1.7e+02	30	17	10	29	4.9e+02

(b) VIB

True \ predicted	0	1	2	3	4	5	6	7	8	9
0	7.7e+02	0	0	0	0	0	3	1	3	0
1	0	9e+02	2	0	0	0	1	1	0	0
2	0	0	8e+02	1	0	0	1	11	7	0
3	0	0	0	7.8e+02	0	5	0	1	7	0
4	0	0	0	0	7.7e+02	0	1	0	1	13
5	5	5	0	83	1	5.4e+02	21	0	34	10
6	53	5	0	0	5	19	5.6e+02	1	1.1e+02	4
7	8	35	1.1e+02	6	17	0	1	5.4e+02	2	52
8	15	1	43	16	15	31	79	1	5.2e+02	50
9	3	6	2	12	47	25	4	10	15	6.9e+02

Figure 5.7:]
Accuracy when using auxiliary dataset [EMNIST]: The results are for ten testing sampling. The model is trained with 32 dimensions. a) Baseline: accuracy: 82.37 b)VIB: accuracy: 86.06



(a) Baseline

(b) VIB:$\beta = 10^0$

(c) VIB:$\beta = 10^{-1}$

(d) VIB:$\beta = 10^{-6}$

Figure 5.8: Projections of the embeddings for test set using 32 dimensions. A) The baseline model shows a plausible clustering. b) $\beta = 10^0$, we find that the representations for novel classes overlap each other. c) $\beta = 10^{-1}$, the representations start to show a clear boundary among classes. d) For the MNIST dataset, we find that $\beta = 10^{-6}$ is appropriate. The model creates a consistent cluster. The representations do not overlap. We argue that finding the appropriate β is relevant.

5.5 Good Layer Selection for improving generalization on FSL

5.5.1 Overview

We address the question of finding good representations for generalization on unseen classes on low data regimes. Traditionally, this question has been addressed using transfer learning approaches such as fine-tune and domain adaptation, which transfer the knowledge (representations) from the source domain to the target domain. Transfer learning assumes that the source domain has large annotated data and the target data has enough sample not to overfit the target model. Domain adaptation considers that the source and target data belong to different domains. The source domain has a vast amount of labeled data, and the target has sufficient data to avoid overfitting. The generalization performance of these approaches heavily relies on the quality of the transferred representations.

One prominent method that has achieved good results is weight imprinting. In this method, the model learns the classes' prototype for novel categories from the embeddings. Then, novel image classification is computed by measuring the distance between the image and the class prototypes. Despite its good performance, the Weight Imprinting model also relies on the quality of the representations created by the encoder. It is unknown what features to transfer to generalize well on novel categories. For example, the learned embeddings that lead to higher classification accuracy on base classes might not be ideal for novel labels. As a result, the classification accuracy can degrade on unknown categories.

This paper provides an efficient training approach for imprinted weight models. We find that a simple design choice of imprinted weights can yield substantial improvements over the baseline model. Our experiments show that (1) introducing a nonlinear projection heads in-between feature extractor, and classifier substantially improves generalization, (2) imprinting from the last projection head does not provide better generalization for novel classes. Instead, we propose imprinting from optimal projection head, and (3) this design choice benefits from a large latent dimension. We are also interested in training in one source domain (Omniglot) and test on another target domain (MNIST). The objective is to verify if the underlying representations from one dataset can be enough to classify novel classes that are not present during the training of different data. We validate our findings by achieving 5.6 and 4.1% improvement on the MNIST dataset trained with the Omniglot dataset.

5.5.2 Improving generalization by transferring the features

Traditional deep learning models assume that the training and testing examples have the same feature input and the same data distribution. However, not often is the case. When there is a difference between the training and test distribution, the performance degrades. Furthermore, collecting extensive annotated training data that matches the test distribution can be difficult and time-consuming in specific fields. Therefore, a transfer learning approach was proposed.

Transfer learning is used to improve the performance of target downstream tasks by transferring the representations from the already trained model. The primary need for transfer learning is when there is a lack of training examples for our interested downstream task. For instance, in medicine, drug discovery, where the data is rare or unavailable. With the availability of large datasets, using datasets that conceptually are similar or related to the learning problem makes transfer learning an attractive solution to address the lack of extensive annotated data. However, transferring knowledge from a source to a target problem cannot be done natively. The critical problem in transfer learning methods is a generalization, i.e., how to transfer useful feature representation that the target classifier can correctly identify unseen classes on low data regime.

Existing approaches on transfer learning methods such as fine-tuning adapt to a new learning problem by using representations previously learned by the existing model [95]. For example, by freezing the base feature extractor network and adding on top a new classifier. Another method is to unfreeze some layers in the feature extractor and jointly train these layers with newly added layers. The intuition behind these approaches is that earlier layers encode generic features, while top layers encode specialized features. We apply the same intuition regarding the interpretation of feature representation created at different layers. However, we use Weight Imprinting based model, a transfer learning approach. Unlike previous models, this model does not require re-training.

The baseline Weight Imprinting model has been outstanding in accurately classifying novel data on low data regimes. However, the Weight Imprinting network architecture consists of an encoder (feature extractor) and a linear classifier. As a result, when optimized on base classes, it yields task-specific representations, which do not generalize well on novel categories. It is unclear what features are relevant for the downstream task in Weight Imprinting models. Representations created by encoder based on base classes might not be practical for novel categories. We hypothesize that this is due to the presence of the classifier. The encoder encodes intrinsic representations of each category, i.e., the model learns inherent features that well describe a particular class, unlike supervised methods that are trained to find specific features that well classify the object.

The problem of generalization is more complex. We want to learn features not specific to individual classes but might help generalize unseen examples. This work aims to identify which layers hold the representations that generalize across different downstream tasks. We thus tasked whether a new design choice for training imprinted-weights can lead to better performance.

5.5.3 Proposed Framework - Weight Imprinting with Projection head

The baseline network model proposed by Qi et al. [141] consists of an encoder network (feature extractor) and a linear layer (classifier). At the training phase, the network is optimized to minimize the misclassification error between the base classes X_b (training set) and accurate label information Y_b. The cosine similarity is used to minimize this difference. The objective function encourages the representations created by the encoder to be specific or discriminative about the base categories. There is no signal enforcing the feature to be agnostic. At the imprinting phase, the feature representations transferred from the encoder might not contain abstract information relevant to categorizing novel classes not seen at the training phase. While this method has shown to work well, there is room for improvement. In particular, we assume that transferring task-agnostic representations might be ideal for generalization across different domains.

Therefore, this work proposes a new network architecture. As shown in Figure 5.9, we introduce a learnable nonlinear transformation between the encoder network and the linear layer classifier. We add a projection head consisting of fully-connected layers and a linear rectified unit in-between. The design choice is simple; we intend to encourage the encoder to create task-agnostic features and explore transferring knowledge from different network parts, which is ignored on the baseline model. We demonstrate that imprinting from the task-agnostic layer achieves better results.

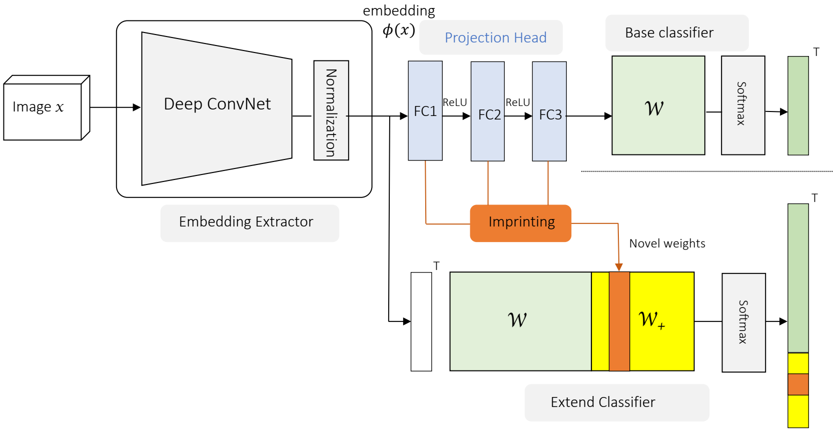


Figure 5.9: Weight Imprinting with Projection head

Furthermore, we evaluate the quality of representations from different network layers. As illustrated in Figure 5.9, this simple framework model has two primary stages.

Training Phase. Given a dataset (base classes X_b), the network $f(\theta)$ is optimized to minimize the cross-entropy loss between the final embeddings and the target label. The classifier C is composed of weight matrix and feature representation $W_b^T f_\theta(X_b)$ followed by a softmax. The model aims to learn the class prototype of base classes.

Imprinting Phase. Given a trained model, the novel class X_n is fed to the imprinting network (fixed network). The output is an embedding vector of each novel class prototype (imprinted weights). In contrast to the baseline model, we imprint from different network parts. Furthermore, to avoid having an equivalent representation between the linear layers (FC1, FC2, FC3), we introduce the non-linear activation function, rectified linear units (ReLU). The activation function is known to help the network learn a complex data pattern. The base weights and novel weights are concatenated at the testing phase and further used for classification. Weight imprinting aims to directly set the final layer weights for novel classes from embeddings [141].

5.5.4 The proposal of Projection Head

It is known that different layers of the network encode different representations [250, 3]. We introduce a small neural network projection head that maps representations to the space where the cross-entropy is applied. We add a multilayer perceptron with three fully-connected layers with non-linearity function ReLU in-between. The purpose is to obtain features that are task-agnostic. We find this hypothesis beneficial to improve generalization on novel classes. Several works have used this approach to improve the downstream task [70, 73]. Thus, we study the importance of projection heads for Weight Imprinting models. First, we trained the network with different numbers of projection heads (1 to 3 fully connected layers with ReLU activation function in-between) as shown in Figure 5.10. Then, we evaluate imprinting from different layers of the projection head.

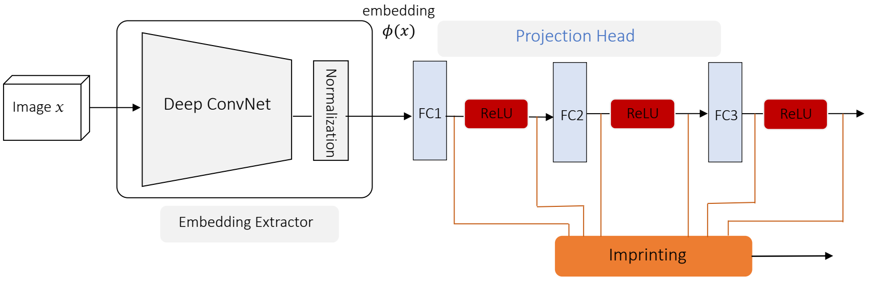


Figure 5.10: Imprinting stages. We consider three distinct phases for imprinting which include before and after ReLU.

We observe that the imprinting model benefits when imprinting from a good layer. The choice of the number of projection heads depends on the downstream task. We find two projection heads for our learning problem to give better accuracy on unseen classes.

5.5.5 Imprinting from different layers

Introducing non-linearity between the representations and classifier improves the quality of embeddings [70]. Since imprinted models rely on representations' quality, it is crucial to introduce a strong regularization. The network has to learn the data's abstract structure (pattern) at the training phase. Then, distinct stages of the network encode different representations. When imprinting from task-agnostic layers such as FC1 and FC2 as shown in Figure 5.10, the performance on a downstream task improves. Also, we find that having more than one projection head results in better performance. Furthermore, we evaluate the effects of the non-linearity activation function. We demonstrate that for our dataset, the performance accuracy is substantially affected.

5.5.6 Experimental Results

Datasets - Omniglot and MNIST

In order to evaluate our proposed method, we first train this model on the Omniglot dataset. A dataset commonly used in deep learning. The Omniglot dataset is designed for developing a more like-human learning algorithm [251]. The dataset consists of 1623 different handwritten characters from 50 different alphabets. The dataset is split into training and testing. The training data contains 30 classes, and the testing includes 20 categories. In addition, each category has 20 images. The images are greyscaled with the size of 28x28. We applied affine transformation with a rotation of 10 degrees for our training. We also use the MNIST dataset. This dataset is also well-known in the machine learning community.

Training - Base Model

Our encoder (feature extractor) is composed of four convolutional networks with [64,128,256,512] hidden units, batch normalization layer[109], ReLU nonlinearity activation function, and a 2x2 max-pooling layer. The latent representation space dimension are [16, 32, 64, 128, 256, 512, 1024]. We pre-train with Adamax optimizer and an initial learning rate of 0.001. We further applied dropout regularization between projection heads. We share the same encoder for pre-train, imprinting, and testing.

Results

We performed experiments on Omniglot dataset [252] and MNIST. The model is trained on the Omniglot dataset, where we used 30 classes containing 20 images per character. Then, we use a single image per class of the MNIST dataset from imprinting. Finally, a single image is used to find the class prototype. This training setup is not the standard in deep learning. However, we are motivated by the assumption that the Omniglot dataset shares or has similar underlying features with the MNIST dataset. Thus, we assume that by training a model on the Omniglot data, the learned representations can also recognize the MNIST digits. In other words, without showing a single number at training time, we want to recognize the digits at test time. We perform 10-ways 1-shot classification for all the experiments, and we average ten runs. The accuracy $(\%)$ measured is the number of correct classifications divided by the total classifications. For better comparison, we plot our results using a bar graph with text information (percentage accuracy) on top of it. The y-axis is the accuracy, and the x-axis is the latent dimension.

The results in Figure 5.11 demonstrate that first, the performance accuracy is better when you have more projection heads. Looking at one projection head result(Figure 5.11(a)), we see that the best accuracy is 43.1%. In contrast, for two projection heads (Figure 5.11(b)), the highest accuracy is 48.7%, and finally, for three projection heads (Figure 5.11(c)), the highest accuracy is 47.2%. These results confirm our initial hypothesis that transferring features from task-agnostic layers leads to better generalization. Second, we find that the model benefits from a large latent space dimension. Increasing the latent space dimension results in better accuracy in all three scenarios.

Figure 5.12 and Figure 5.13 show the top-1 accuracy of 200-way classification for unseen classes and all the samples in the CUB-200-2011 dataset. The results for this particular dataset did not surpass our expectations. For example, observing a) and b) where we imprint from the feature extractor and projection heads. We did not find a significant improvement in the accuracy, and the gain is around 0.2% compared to the baseline [141] Weight Imprinting model. Besides, having more projection heads did not contribute to good accuracy, as demonstrated in Figure 5.13. In the future, we intend to investigate more on this particular issue.

Benefits of Projection Heads

We evaluate the importance of including the projection heads—the results in Figure 5.11 show that having a deeper network improves the downstream task. Network with two and three projection heads present superior top-1 accuracy, 48.7%, and 47.2%, respectively, while one projection head achieves 43.1%. Therefore, it is important to find the optimal number of projection heads. Often the number of projection heads depends on the downstream task. While similar conclusion is reported on [253].

The key idea of including the projection heads is that the model maps representations to other spaces. We claim that these spaces' representations are task-agnostic compared to the baseline architecture that the latent representation is task-specific. These results show that the representations before the last layer before the classifier are better representations for generalization. Another inference is that having a network solely with a feature extractor and a linear classifier forces the model to lose general information about the data and learn features that well describe the classes, and the classifier induces this behavior. The classifier is optimized to be discriminative about the classes and not to understand the data structure. Thus, the layer before the classifier in the feature extractor removes information that may be relevant for generalization. By including the projection heads with non-linear activation function, we allow more transformation to be formed, resulting in less specific features.

Behavior of non-linear activation function

Figure 5.14 shows the performance when imprinting is performed before and after the non-linear activation function ReLU. We find that imprinting before ReLU is more efficient for all network setup, but the gain appears to be more significant for three projection heads. This behavior is expected since introducing more projection heads (fully-connected layers) with a non-linear activation function allows the model to map the representations to different spaces and allows more transformation to be formed at the latent space. For our best result of the network with one projection, we see that imprinting before ReLU achieves 43.1% while imprinting after gives 36.5%. For a network with two projection heads, the best results show 48.7% before ReLU and 42.8% after ReLU. Lastly, in the network with three projection heads, we find 47.2% accuracy before ReLU and 33.0% after ReLU. However, we highlight that this behavior might not generalize across datasets. For example, we might have different accuracy before and after, but the order is not linear.

Training with Large embedding

In order to study the effectiveness of embedding, we train the model by varying the size of the embedding. Figure 5.11 shows the effect of various dimension of latent representation. We find that imprinting benefits from larger dimensionality. A network with a single projection head from 16-dimension to 1024-dimension shows an improvement of 7.6%. For the network with two projections heads, there is an 18% improvement. While for the network model with three projection heads, we see an improvement of 18.9%. In the case of the discriminative models, this result is not a surprise. Since, with few dimensions such as 16-dimensional space, the model is forced to be highly biased, i.e., it loses a lot of information that might be helpful for generalization. Besides, our initial idea is that the model should not be highly discriminative. It should contain general features and not intrinsic features about the classes. Furthermore, we point out that more projection head does not necessarily mean good performance. The optimal number of projection heads depends on the learning problem.

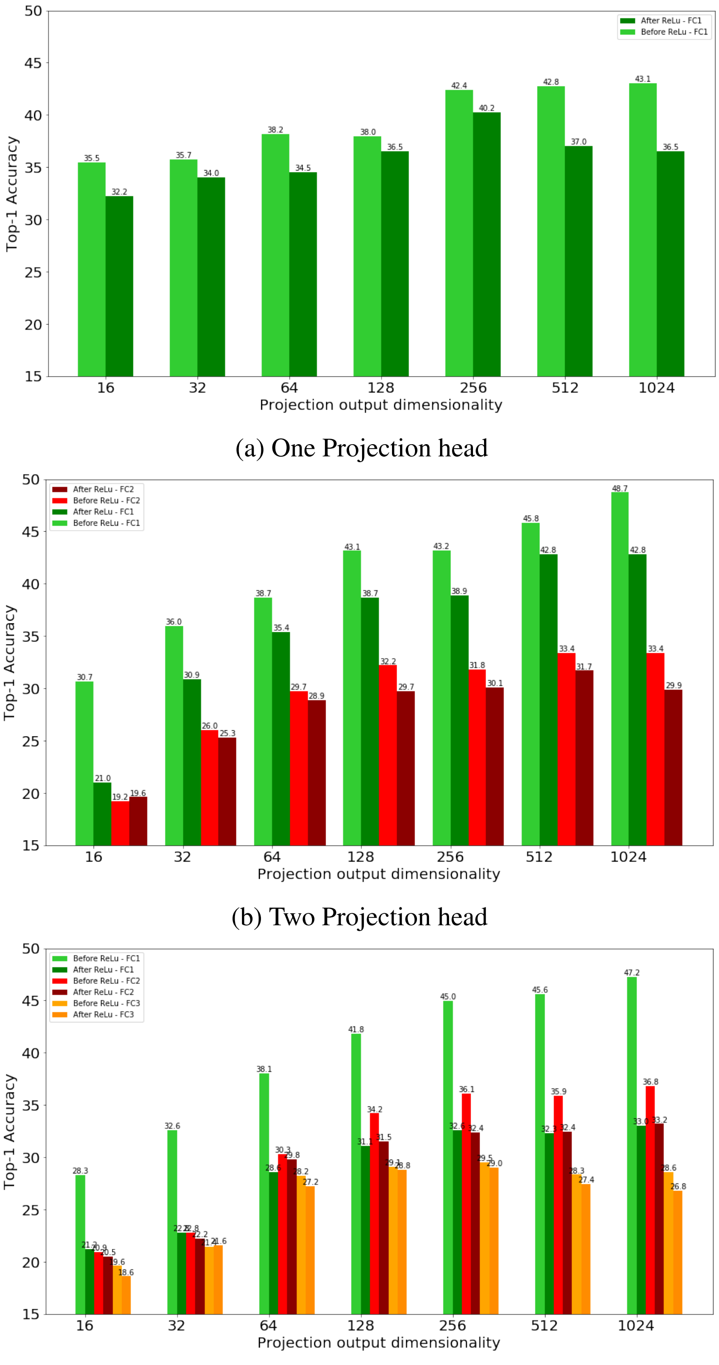
Regularization with Dropout

Dropout is a common way to introduce a strong regularization. The key idea is to randomly drop units (along with their connections) from the network at training time [107]. We examine the effect of randomly decreasing 50% of the neurons at each projection head. Figure 5.15 demonstrates the impact of dropout. The experiment setup is the same, we train on the Omniglot dataset with 30 different categories of the alphabets, and we test on the MNIST dataset. The results show that by introducing dropout, the performance gets better compared to not using dropout. In particular, two and three projection heads benefit from introducing dropout. As a result, we see the results to improve around 2.6% and 1.6% for two and three projection heads, respectively. In contrast, we did not see the same behavior when the projection head is one, the accuracy decreases.

Qualitative Evaluation with t-SNE

Plotting the embeddings is a common approach in machine learning to visualize the behavior of the latent variables learned by the model. Figure 5.16 shows the latent projection at the training phase. Again, we have the representations from fully-connected layers (FC1, FC2, and FC3). Observing the projection of FC3, we find that there is a clear class boundary between the categories. The model has learned intrinsic features that well describe each class. Imprinting or transferring features from the FC3 layer does not generalize well, as shown previously. Observing the projections in the FC2, we see a sign of class boundary. However, it is not evident for all classes. The model has learned specific features of each class and abstract features of the data. Imprinting from this layer gives better performance than the FC3. Finally, observing the projection in the FC1, we visualize that there is no explicit class separation, but some of the latent variables overlap. However, it is possible to see some structure. The representations are task-agnostic, and imprinting from this layer gave better performance accuracy on our downstream task.

We evaluate the actual results of imprinting in the test phase. When imprinting from the FC1, we see an explicit boundary between the classes. Therefore, the classifier performs well, as previously demonstrated. Also, imprinting from the FC2, we see a class separation; however, some representations overlap. Finally, observing the projections in the FC3, we see no evident class separation the representation overlap, and it is shown that in this scenario, the classifier performs poorly. The core idea of qualitative evaluation is to demonstrate that imprinting from task-agnostic representations is likely to generalize well than task-specific layers.

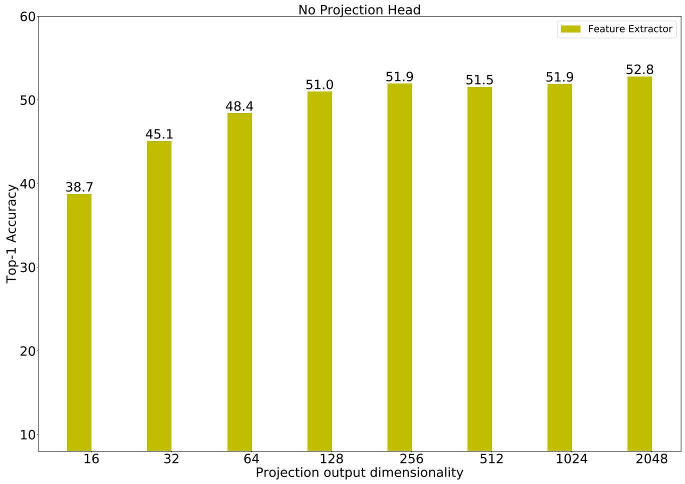


(a) One Projection head

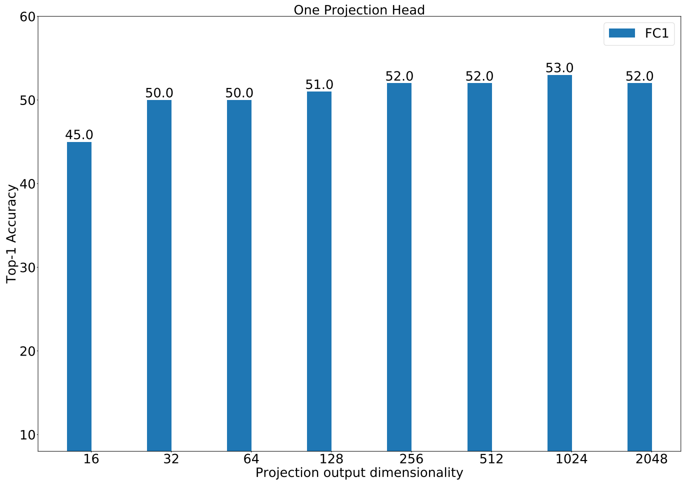
(b) Two Projection head

(c) Three Projection head

Figure 5.11: Effect of Projection head - Top-1 accuracy. Introducing projection heads the performance accuracy Increases. Besides, increasing the size of representations improves downstream task accuracy.

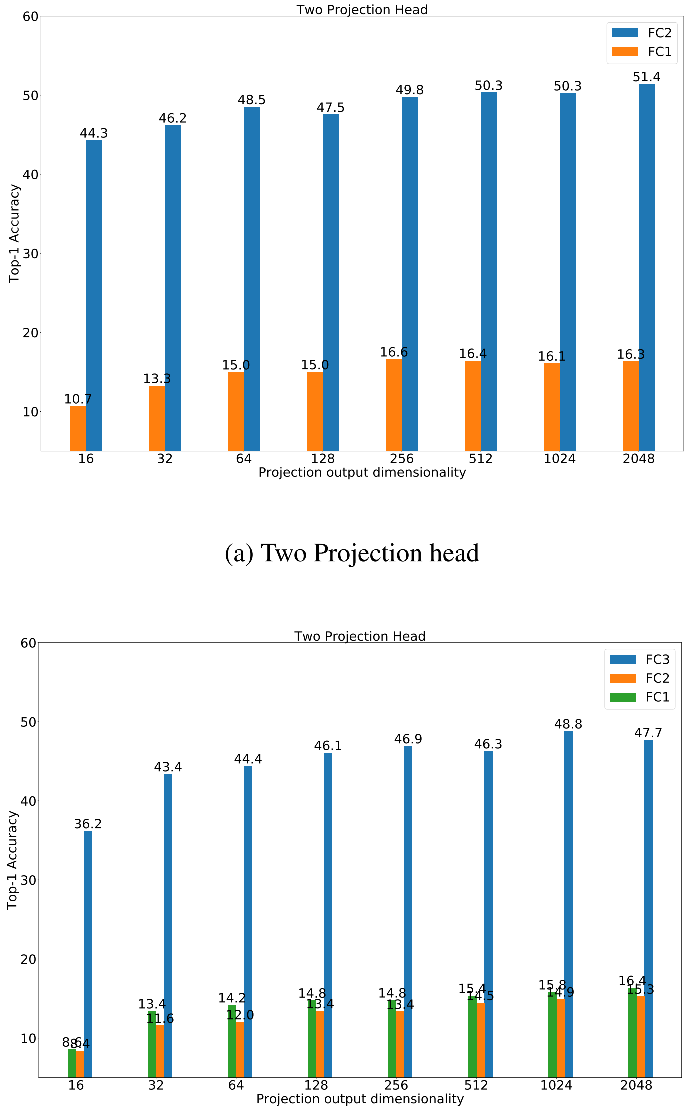


(a) No Projection head



(b) One Projection head

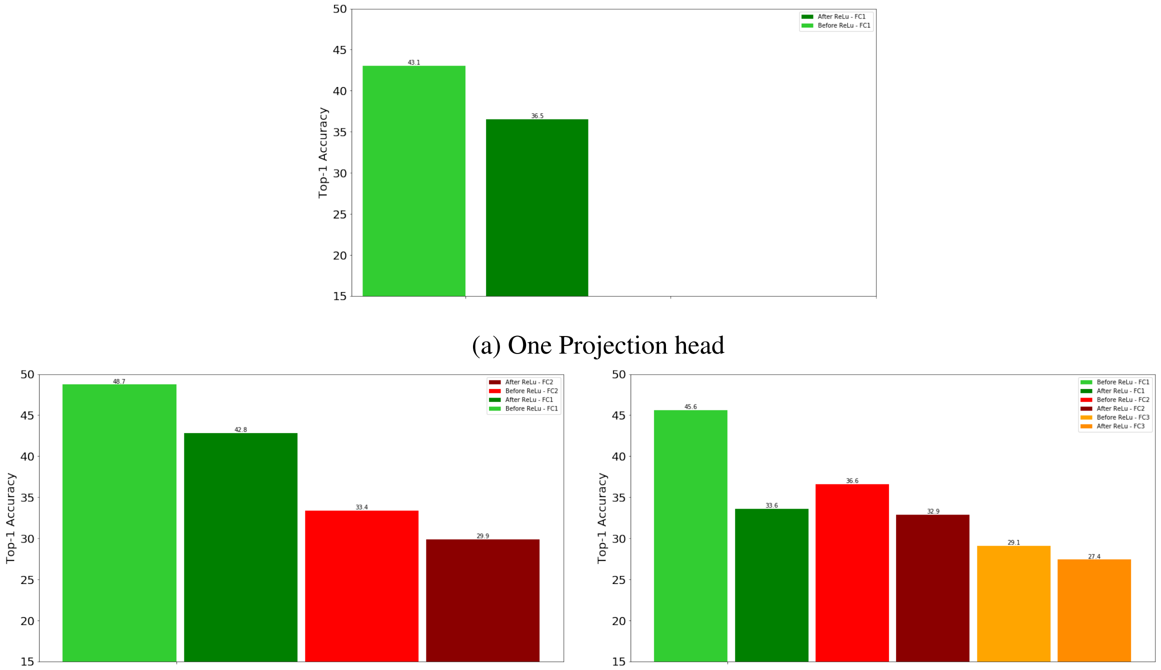
Figure 5.12: Effect of Projection head - Top-1 accuracy for a model trained with different dimensionality on latent representation. a)Imprinting from feature extractor. b)Imprinting from projection head FC1. The performance accuracy did not get significant improvement. However, the progress is around 2% compared to no projection head.



(a) Two Projection head

(b) Three Projection head

Figure 5.13: Effect of Projection head - Top-1 accuracy for a model trained with different dimensionality on latent representation. Increasing the number of projection heads did not contribute to the performance accuracy

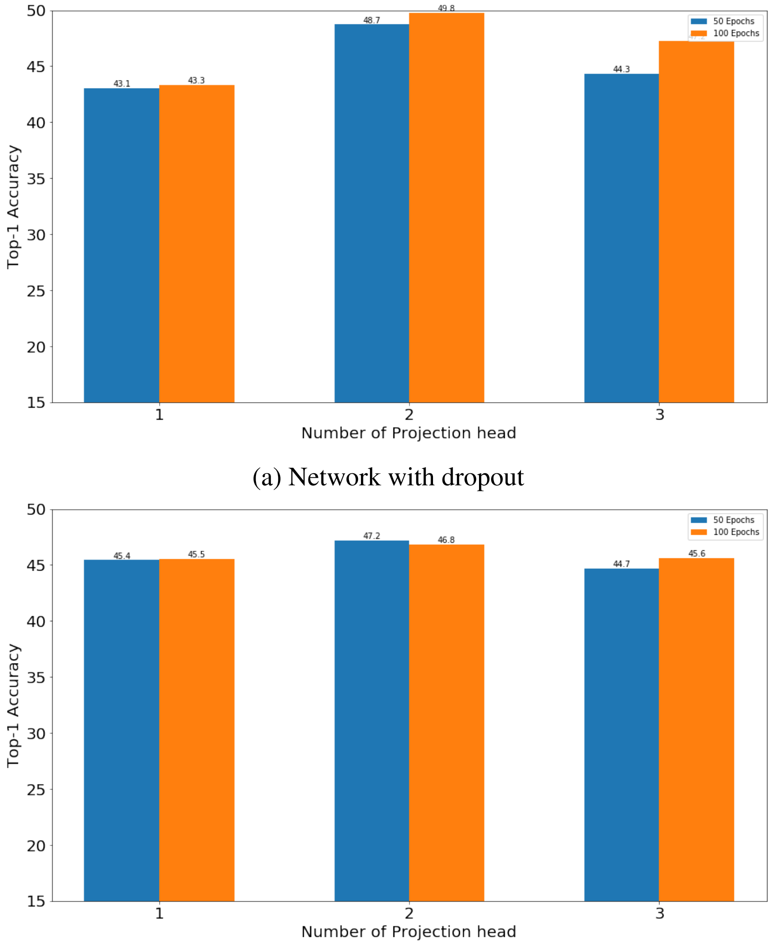


(a) One Projection head

(b) Two Projection head

(c) Two Projection head

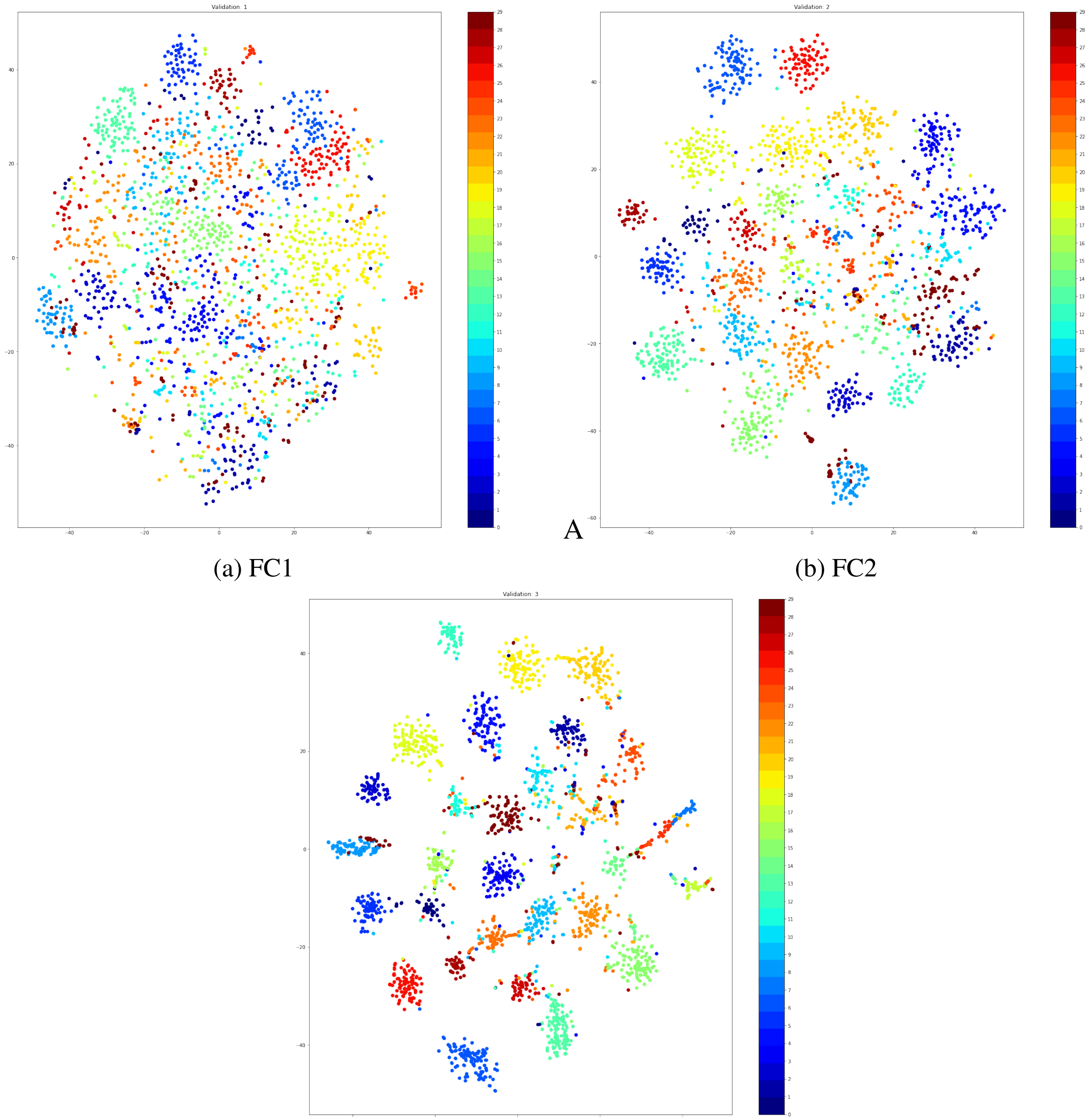
Figure 5.14: Effect of ReLU - Imprinting from before ReLU gives better accuracy rather than after in every projection head



(a) Network with dropout

(b) Network without dropout

Figure 5.15: Effects of dropout - dropout is more effective when there are two projection heads in the network



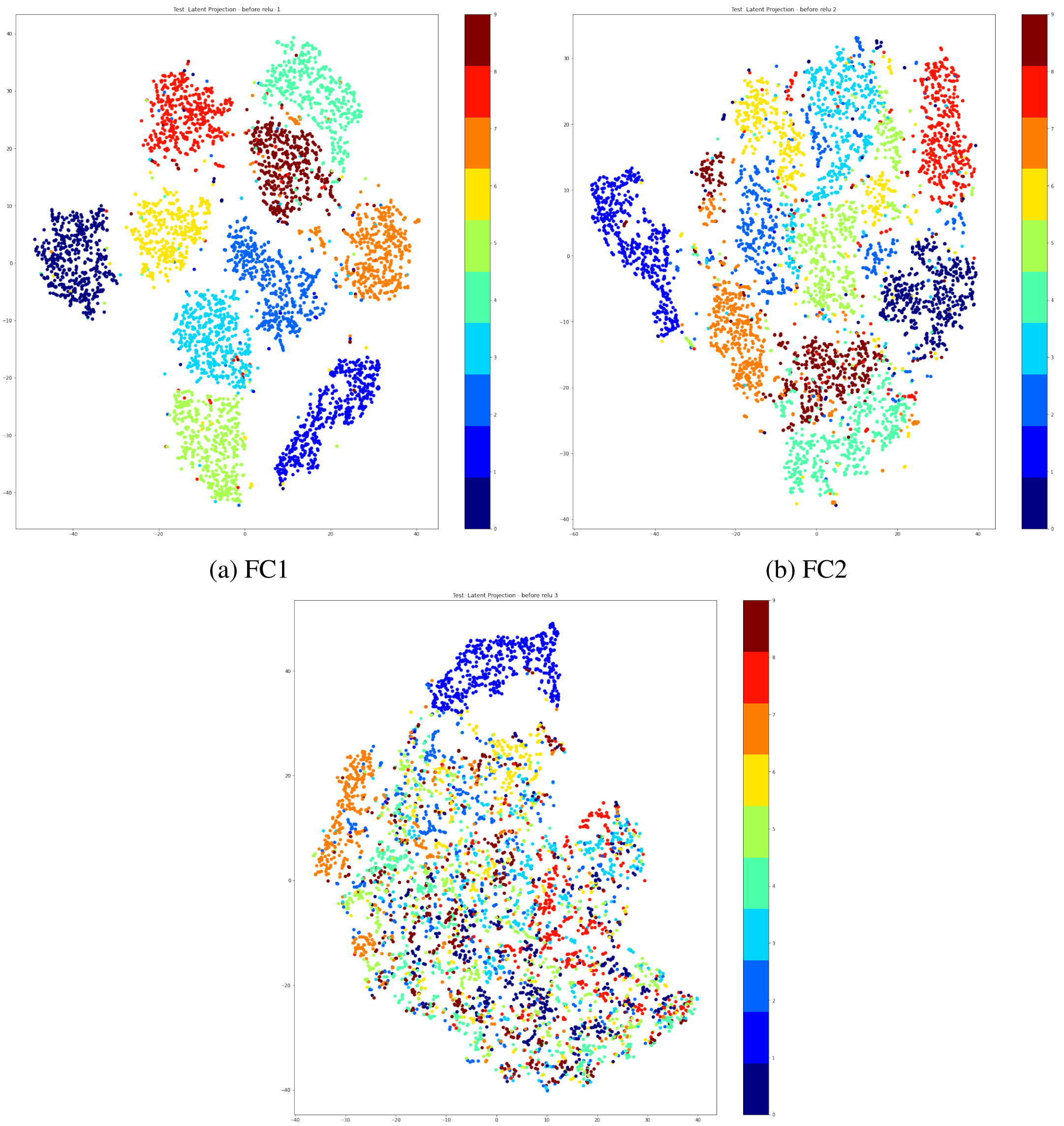
A

(a) FC1

(b) FC2

(c) FC3

Figure 5.16: Training - At the training phase, deeper the network, the representation tends to be more task-specific.



(a) FC1

(b) FC2

(c) FC3

Figure 5.17: Testing - The deeper the network, the more representations overlap, making it challenging to visualize class boundaries.

Table 5.5: 200-way top-1 accuracy on unseen categories of CUB-200-2011.

Models	**n=1**
Weight Imprinting [141]	44.75
Weight Imprinting [our implementation]	52.8
RotNet [65] [our implementation]	45.0
Generator + Classifier [38]	45.42
Matching Networks [143]	41.71
VIB [ours]	**53.12**
Layer Selection [ours]	53.0

5.6 Comparison with other existing models

In Figure 5.5, we compare our two proposed methods Variational Information Bottleneck and good layer selection with existing models using CUB-200-2011 dataset. The results of Generator + Classifier and Matching Networks are taken from [141]. The latent dimensionality is 256 for each method. Our approach improves slightly compared to the baseline weight imprinting (our implementation) and significantly outperforms other models.

5.7 Related Work

The problem of learning to generalize from unseen classes is known as the Few Shot classification. Most existing FSL approaches use different techniques. In the following, we discuss relevant approaches close to our work.

5.7.1 Self-supervised learning

The standard approach in deep learning is to pre-train a supervised model on a large dataset such as ImageNet [57]. Then fine-tune on a small network, where the data is also small. Self-supervised pretraining has emerged as an alternative and practical approach to learning from extensive annotated data. The supervision in self-supervised models comes from the data itself. The pseudo label is automatically generated without human annotation. The self-supervised approach allows models to learn rich features to be transferred to related target tasks of interest. Examples of self-supervised approaches for few-shot learning include contrastive learning [254], that network is optimized to group similar points in latent space and push away representations from different labels, pretext tasks.

5.7.2 Metric based approaches

This class addresses the FSL by learning to compare the distance similarity between representations. In the lower dimension, similar samples are close together, and dissimilar samples are far apart.

Often Cosine Similarity [143, 242] and Euclidean distance is used as a metric[142]. The intuition is that by learning the distance similarity, the classifier can classify a novel class by comparing its prototype to base prototypes. [143] and [142] learn a distance metric using the nearest neighbor classifier between the query set and labeled input, while [144] learns to compare by adding an extra network. These methods use support set at the test time. Not often the support set is available. However, novel ideas have been proposed for learning the similarity between embeddings without requiring a support set. The method proposed in [141] and [243] uses novel features as their weights to classify novel classes, while in a slightly different work [242, 255] learns a weight generator to predict novel classes. In this work, we follow the baseline metric close to our work [141].

5.7.3 Data Augmentation and Hallucination based approaches

A well-known limitation of FSL models is the lack of a large amount of novel data. Data augmentation and data hallucination propose to augment the training data by generating additional training samples using information from the training set. Standard Data Augmentation such as random crop, color jitter, random flip, noisy blur, and random rotation produces a limited alternative for increasing the data distribution. Recent work has proposed a more stable augmentation for diverse scenarios. [28] proposes mixup; the output is a mix of two distinct labels. On the contrary, [29] suggests cutout, the technique throw away a portion of the input image. [1] presents the cutmix, and the authors argue that throwing away information is not beneficial; instead, the authors add a different label in the cropped area of the cutout. Hallucination-based classes learn to generator additional novel data from initial training data. The method proposed by [35] hallucinates new samples by passing an actual sample and noisy image to a generator while [38] hallucinates new samples in the representation space. Since these approaches improve classification for FSL, we include standard data augmentation except random rotation since it has been shown not to work for MNIST data. Also, we include the ablation results for the mixup, cutout, and cutmix in our experiments.

5.8 Summary

Inspired by the effectiveness of the information bottleneck principle, we introduced VIB in to FSL. The VIB objective function strongly regularizes the representation by minimizing the mutual information between input data and representation while keeping the classification accuracy for the pre-training task. The proposed VIB objective function learns to extract useful representation that well describes the data. We claim that cross-entropy loss function alone is not sufficient for generalization. We confirmed that the proposed objective function outperforms models trained with cross-entropy loss, with particular improvement on accuracy by 5% even with a small network. We further improved the classification accuracy by 2% more using an auxiliary dataset. Interestingly, recently proposed augmentations such as the mixup, cutout, and cutmix do not contribute to the proposed model with VIB.

For our second proposed method, we study the importance of an efficient training approach for imprinted weight models. Our experimental results demonstrate that including projection heads is beneficial for Few Shot learning models based on imprinted weight. We also show that imprinting from a good layer improves accuracy. We find 6.5% improvement when imprinting from two projection heads and 3.9% from three projection heads. Interestingly, we find out that imprinting before the activation function ReLU improves the performance of the linear classifier. The strength of this simple design choice suggests that, for imprinted weights models, imprinting from earlier task-agnostic layers improves the generalization task.

We aim to confirm the validity of the proposed methods with more significant and complex datasets such as ImageNet or CUB for future work. We are also interested in incorporating VIB into contrastive learning settings. Future avenues include applying these methods on more challenging datasets and training on contrastive learning loss.

Chapter 6

Conclusion

This thesis addresses the challenge of limiting the existing model based on representation learning to have the desired output. In particular, we proposed methods that 1) allow to control representation learning of generative models, 2) models the latent representation of discriminative model trained in low data regime, and 3) allow transferring good representations in few-shot learning. Compared to existing work proposing to control representation learning, this work presents a significant step towards controlling representation such that we have the desired structure aligned to the downstream task.

Mainly, we identify a few issues related to existing models that prevent them from having the desired output performance, and we present methods to overcome these shortcomings. Our methods introduced in Chapter 3 address these shortcomings in generative and discriminative models. We present a method to control latent representation learning in the generative model. To do so, we proposed an objective function that constrains the latent representation space of the vanilla VAE. The objective function encourages the latent representation space to have the desired structure for the downstream task.

In Chapter 4, we demonstrate the effectiveness of the proposed method on addressing a real-world application. We addressed the challenge of image in-between generation when the sequence of the frames is not continuous; that is, there is a large gap between the frames. Furthermore, we increase the complexity of the interpolation by working with different degrees of freedom of the objects. Our synthetic datasets included more than one object in a single image. Based on the experimental results, we demonstrated that our objective function encourages the latent space to have a structure that allows interpolation in latent manifold and image space, respectively. The proposed method outperforms the vanilla variational autoencoder on images in-between. However, the latent structure of vanilla VAE showed that it is not enough to address the downstream task. Besides, the generated in-between was not realistic. Compared to several state-of-art works on image interpolation. Our proposed model outperforms FlowNet 2.0 and its versions and SloMo on image in-between generations. On the other hand, we demonstrate that FlowNet 2.0 and SloMo generate in-between images with artifacts and lack structure the resembles the actual in-between.

Chapter 5 considers the challenge of generalization of few-short learning in discriminative learning. Despite the recent success of transfer learning models and self-supervised learning, learning from limited data is still an open issue in machine learning. On the other hand, we demonstrate that it is hard to generalize unseen data when the training and test distribution changes. We introduced two approaches to address these challenges. In the first method, we introduce the variational information bottleneck loss function. Then, we showed that introducing an objective function that models the latent representation improves generalization. In particular, our proposed objective function encourages the feature extractor to ignore the features that do not contribute to the classification and focus on features that well describe the data. Moreover, this objective function benefits generalization since it does not rely only on the cross-entropy function to discriminate the data. Finally, our proposed objective function is helpful for inference since the encoder is optimized to learn features that well describe the data rather than encoding irrelevant features. The experimental results shown in Chapter 5 demonstrated that our model outperforms the model-based cross-entropy loss function for the task of generalization in few-shot learning.

In the second method, we consider the same problem of generalization on novel classes in few-shot learning. Furthermore, the existing techniques propose transferring features from the feature extractor or fine-tuning the model. Fine-tuning considers re-training the model and requires enough data to avoid overfitting the model. Instead, we assume having a single image per class in our target domain in our problem setting. The approach proposed in this work aims at a good layer selection for transfer learning. In particular, we assume that transferring features from task-specific layers does not help generalization as done in transfer learning techniques. Also, the model loses a lot of information that may be useful for generalization across domains. Instead, we proposed transferring representations from task-agnostic layers. Using this assumption, we introduced projection heads between the feature extractor and a linear classifier. Introducing the allow more variation to be formed and maintain more abstract information relevant for generalization. Then, we provided experimental results that validated our assumptions. Specifically, we demonstrated that imprinting representations from the task-agnostic projection layer perform better than imprinting from the feature extractor.

6.1 Summary

This thesis proposed methods that allow controlling the latent representation to improve the performance accuracy of the downstream tasks such as generation and classification. Whereas prior approaches often do not focus on the latent space enforcement to align the property with the downstream learning task or naively transfer features that do not generalize across domains. We showed that our methods find applications on a wide range of practical machine learning problems. In addition, the methods presented can easily be implemented and tested on different domains.

6.2 Future Research Avenue

In future work, we are interested in investigating the properties of the latent manifold representation for different downstream tasks such as music, text, and speech generations. For example, we intend to interpolate two genres in music and generate a smooth transition from one genre to another. Similarly, in-text data, we want to generate interpolation of two topics.

A recent avenue of latent models is disentanglement, as stated by Bengio et al. [3] and Higgins et al. [184]. This challenging area disentangles the underlying explanatory factors of variation present in the latent representation space. Disentangling the representation may allow discovering interpretable properties of the data to address critical issues across different domains. For example, in the case of a discriminative model, one might disentangle the hidden factors of the data and use useful ones for the current downstream task and keep others for unknown feature tasks. This idea is motivated by the limitation of current models that throw away a significant amount of information that is considered noisy or does not contribute to the downstream task.

An alternative direction is to learn the manifold structure that allows extracting compacted latent representations to predict future representations that may perform well on various tasks. Furthermore, we intend to investigate the performance capability of the latent space of generative models when the training example is limited. Machine learning models are known to strive on low data regimes. Due to easy visualization and comprehension, we applied our methods to synthetic datasets. Still, we are also interested in using them on complex datasets, such as in non-image datasets.

Acknowledgements

Foremost, I would like to express my sincerest gratitude to my advisors, Professor Hidemoto Nakada, Professor Yusuke Tanimura, and Professor Hideki Asoh, for continuous support on my research, academic life and for sharing their knowledge for me. In particular for their patience and dedication through these years. I have no words to express how much I appreciate and am honored to learn from them and share good conversations.

Besides my professors, I also extend my gratitude to AIST professionals for their support and to Professor Sakurai and Professor Sakuma from the department of Computer Science at the University of Tsukuba for being on this thesis committee. I also express my gratitude to Professor Neil Millar and Anna Bogdanova for their encouragement and support.

I cannot forget my lab mates since I joined the lab Mr. Ming Li, Kaihui Zhang, Siming Zheng, Qing Hong, Duo Zhang, Tianlun Wang, Mu Zhou, Liangyu Liu, and Yingfeng Fu, to them my sincere thank you. I cannot forget my professor of karate and Jiujitsu, Professor Takano, and his family.

Of course, my sincerest appreciation for my mother Cristina Jose and siblings Edgar Cristovao, Luisa Cristovao, Cristina Cristovao, Jessica Cristovao, and Luisa Vincent for their support and help. Thank you for your beautiful advice and conversations with my beloved father, Cristovao Paulino, who is not here to see the end of this journey.

Thank you for the support to my family, particularly my uncle Abreu Muhimua and my nieces and nephews. Thank you for the support to my friends in this incredible city of Tsukuba, Meriem Bejaoui, Aaron Eli, Hamidou Tangara, Harunobu Daikoku, Guli Fernandez, and Ayaka Suzuki my gratitude. Thank you for teaching me the Japanese culture and language to my Japanese colleagues at the Dojo.

Thank you to my friends Edouard and Einat for the wonderful conversation on life. Thank you so much to The Ministry of Education, Culture, Sports, Science, and Technology (MEXT) for giving me this opportunity to live and study in Japan. Lastly, I thank God for this opportunity and protection all this time.

Bibliography

[1] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

[2] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 35, No. 8, pp. 1798–1828, 2013.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, Vol. 25, pp. 1097–1105, 2012.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[8] Qinglin Li, Bin Li, Jonathan M Garibaldi, and Guoping Qiu. On designing good representation learning models. *arXiv preprint arXiv:2107.05948*, 2021.

[9] Irina Higgins, et al. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

[11] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pp. 7354–7363. PMLR, 2019.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[13] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4990–4998, 2017.

[14] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[15] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, Vol. 26, No. 11, pp. 3365–3385, 2019.

[16] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[17] Xiaoou Tang and Xiaogang Wang. Face sketch synthesis and recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 687–694. IEEE, 2003.

[18] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 31, No. 11, pp. 1955–1967, 2008.

[19] Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017.

[20] Devendra Prakash Jaiswal, Srishti Kumar, and Youakim Badr. Towards an artificial intelligence aided design approach: application to anime faces with generative adversarial networks. *Procedia Computer Science*, Vol. 168, pp. 57–64, 2020.

[21] Koichi Hamada, Kentaro Tachibana, Tianqi Li, Hiroto Honda, and Yusuke Uchida. Full-body high-resolution anime generation with progressive structure-conditional generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

[22] Thanh Thi Nguyen, Cuong M Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey. *arXiv preprint arXiv:1909.11573*, 2019.

[23] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[24] Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, Vol. 6, pp. 437–450, 2018.

[25] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. *arXiv preprint arXiv:1907.05789*, 2019.

[26] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018.

[27] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, Vol. 6, No. 1, pp. 1–48, 2019.

[28] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[29] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[30] Jin-Woo Seo, Hong-Gyu Jung, and Seong-Whan Lee. Self-augmentation: Generalizing deep networks to unseen classes for few-shot learning. *Neural Networks*, Vol. 138, pp. 140–149, 2021.

[31] Naiyan Wang and Dit Yan Yeung. Learning a deep compact image representation for visual tracking. *Advances in neural information processing systems*, 2013.

[32] R Caruana. Multitask learning. autonomous agents and multi-agent systems. 1998.

[33] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[34] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7455–7463, 2017.

[35] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7278–7286, 2018.

[36] Jiang Lu, Pinghua Gong, Jieping Ye, and Changshui Zhang. Learning from very few samples: A survey. *arXiv preprint arXiv:2009.02653*, 2020.

[37] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. *arXiv preprint arXiv:1810.11730*, 2018.

[38] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

[39] Xu Zheng, Tejo Chalasani, Koustav Ghosal, Sebastian Lutz, and Aljosa Smolic. Stada: Style transfer as data augmentation. *arXiv preprint arXiv:1909.01056*, 2019.

[40] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[41] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, Vol. 3, pp. 1470–1470. IEEE Computer Society, 2003.

[42] Dawood Al Chanti and Alice Caplier. Improving bag-of-visual-words towards effective facial expressive image classification. *arXiv preprint arXiv:1810.00360*, 2018.

[43] Radu Tudor Ionescu, Marius Popescu, and Cristian Grozea. Local learning to improve bag of visual words model for facial expression recognition. In *Workshop on challenges in representation learning, ICML*. Citeseer, 2013.

[44] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Vol. 60, No. 2, pp. 91–110, 2004.

[45] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 1, pp. 886–893. Ieee, 2005.

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, Vol. 115, No. 3, pp. 211–252, 2015.

[47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, Vol. 28, pp. 91–99, 2015.

[49] Mariana-Iuliana Georgescu, Radu Tudor Ionescu, and Marius Popescu. Local learning with deep and handcrafted features for facial expression recognition. *IEEE Access*, Vol. 7, pp. 64827–64836, 2019.

[50] Wenyi Lin, Kyle Hasenstab, Guilherme Moura Cunha, and Armin Schwartzman. Comparison of handcrafted features and convolutional neural networks for liver mr image adequacy assessment. *Scientific Reports*, Vol. 10, No. 1, pp. 1–11, 2020.

[51] Marleen De Bruijne. Machine learning approaches in medical image analysis: From detection to diagnosis, 2016.

[52] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps*, pp. 323–350, 2018.

[53] Paras Lakhani, Adam B Prater, R Kent Hutson, Kathy P Andriole, Keith J Dreyer, Jose Morey, Luciano M Prevedello, Toshi J Clark, J Raymond Geis, Jason N Itri, et al. Machine learning in radiology: applications beyond image interpretation. *Journal of the American College of Radiology*, Vol. 15, No. 2, pp. 350–359, 2018.

[54] Christopher M Bishop. Pattern recognition. *Machine learning*, Vol. 128, No. 9, 2006.

[55] Sotiris B Kotsiantis, I Zaharakis, P Pintelas, et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, Vol. 160, No. 1, pp. 3–24, 2007.

[56] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pp. 84–92. Springer, 2015.

[57] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[58] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2051–2060, 2017.

[59] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[60] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 762–771, 2018.

[61] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

[62] Nawid Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*, pp. 228–243. Springer, 2018.

[63] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In *Asian Conference on Computer Vision*, pp. 99–116. Springer, 2018.

[64] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

[65] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[66] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

[67] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2701–2710, 2017.

[68] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

[69] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6874–6883, 2017.

[70] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

[71] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, Vol. 9, No. 1, p. 2, 2021.

[72] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

[73] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[74] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019.

[75] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[76] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[77] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

[78] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[79] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020.

[80] Kim-Han Thung and Chong-Yaw Wee. A brief review on multi-task learning. *Multimedia Tools and Applications*, Vol. 77, No. 22, pp. 29705–29725, 2018.

[81] Rich Caruana. Multitask learning. *Machine learning*, Vol. 28, No. 1, pp. 41–75, 1997.

[82] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pp. 845–850, 2015.

[83] Partoo Vafaeikia, Khashayar Namdar, and Farzad Khalvati. A brief review of deep multi-task learning and auxiliary task learning. *arXiv preprint arXiv:2007.01126*, 2020.

[84] Mingsheng Long and Jianmin Wang. Learning multiple tasks with deep relationship networks. *arXiv preprint arXiv:1506.02117*, Vol. 2, No. 1, 2015.

[85] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

[86] Baijiong Lin, Feiyang Ye, and Yu Zhang. A closer look at loss weighting in multi-task learning. *arXiv preprint arXiv:2111.10603*, 2021.

[87] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.

[88] Lecheng Zheng, Yu Cheng, and Jingrui He. Deep multimodality model for multi-task multi-view learning. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 10–18. SIAM, 2019.

[89] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[90] Xiaogang Wang, Cha Zhang, and Zhengyou Zhang. Boosted multi-task learning for face verification with applications to web image and video search. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 142–149. IEEE, 2009.

[91] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

[92] Zhiyuan Tang, Lantian Li, and Dong Wang. Multi-task recurrent model for speech and speaker recognition. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–4. IEEE, 2016.

[93] Dasol Hwang, Jinyoung Park, Sunyoung Kwon, Kyung-Min Kim, Jung-Woo Ha, and Hyunwoo J Kim. Self-supervised auxiliary learning for graph neural networks via meta-learning. *arXiv preprint arXiv:2103.00771*, 2021.

[94] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[95] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2017.

[96] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 2180–2188, 2016.

[97] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[98] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

[99] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[100] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[101] Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. Controlvae: Controllable variational autoencoder. In *International Conference on Machine Learning*, pp. 8655–8664. PMLR, 2020.

[102] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

[103] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

[104] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2649–2658. PMLR, 2018.

[105] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[106] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, Vol. 1, No. 4, pp. 541–551, 1989.

[107] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[108] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[109] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

[110] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[111] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2820–2828, 2019.

[112] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[113] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019.

[114] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[115] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[116] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

[117] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, Vol. 22, No. 10, pp. 1345–1359, 2009.

[118] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pp. 65–93. Elsevier, 1992.

[119] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[121] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

[122] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[123] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, Vol. 53, No. 3, pp. 1–34, 2020.

[124] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. 2019.

[125] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 11, No. 5, pp. 1–46, 2020.

[126] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, Vol. 312, pp. 135–153, 2018.

[127] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pp. 1058–1066. PMLR, 2013.

[128] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. *Advances in neural information processing systems*, Vol. 26, pp. 3084–3092, 2013.

[129] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 648–656, 2015.

[130] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pp. 1–26, 2020.

[131] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 201–208. JMLR Workshop and Conference Proceedings, 2010.

[132] Grégoire Mesnil Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley, Pascal Vincent, et al. Unsupervised and transfer learning challenge: a deep learning approach. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 97–110. JMLR Workshop and Conference Proceedings, 2012.

[133] Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 164–172. JMLR Workshop and Conference Proceedings, 2011.

[134] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*, 2016.

[135] Dong Wang and Thomas Fang Zheng. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1225–1237. IEEE, 2015.

[136] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *arXiv preprint arXiv:1902.07208*, 2019.

[137] Benjamin Q Huynh, Hui Li, and Maryellen L Giger. Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of Medical Imaging*, Vol. 3, No. 3, p. 034501, 2016.

[138] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, Vol. 3, No. 1, pp. 1–40, 2016.

[139] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

[140] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

[141] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5822–5830, 2018.

[142] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[143] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, Vol. 29, pp. 3630–3638, 2016.

[144] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

[145] Jianyi Li and Guizhong Liu. Few-shot image classification via contrastive self-supervised learning. *arXiv preprint arXiv:2008.09942*, 2020.

[146] Yizhao Gao, Nanyi Fei, Guangzhen Liu, Zhiwu Lu, Tao Xiang, and Songfang Huang. Contrastive prototype learning with augmented embeddings for few-shot learning. *arXiv preprint arXiv:2101.09499*, 2021.

[147] Qing Chen and Jian Zhang. Multi-level contrastive learning for few-shot problems. *arXiv preprint arXiv:2107.07608*, 2021.

[148] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

[149] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021.

[150] Yu Zhang, Gongbo Liang, and Nathan Jacobs. Dynamic feature alignment for semi-supervised domain adaptation. *arXiv preprint arXiv:2110.09641*, 2021.

[151] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. PMLR, 2018.

[152] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8050–8058, 2019.

[153] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pp. 934–940, 2020.

[154] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, Vol. 29, pp. 343–351, 2016.

[155] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, Vol. 22, No. 2, pp. 199–210, 2010.

[156] Maksymilian Wojtas and Ke Chen. Feature importance ranking for deep learning. *arXiv preprint arXiv:2010.08973*, 2020.

[157] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[158] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *International Conference on Artificial Intelligence and Statistics*, pp. 2359–2367. PMLR, 2021.

[159] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

[160] David Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[161] Shakti Kumar, Jithin Pradeep, and Hussain Zaidi. Learning robust latent representations for controllable speech synthesis. *arXiv preprint arXiv:2105.04458*, 2021.

[162] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. *arXiv preprint arXiv:1812.02833*, 2018.

[163] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

[164] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[165] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.

[166] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[167] Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. Variational information bottleneck for effective low-resource fine-tuning. *arXiv preprint arXiv:2106.05469*, 2021.

[168] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[169] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

[170] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pp. 531–540. PMLR, 2018.

[171] Tom White. Sampling generative networks. *arXiv preprint arXiv:1609.04468*, 2016.

[172] Paulino Cristovao, Hidemoto Nakada, Yusuke Tanimura, and Hideki Asoh. Generating in-between images through learned latent space representation using variational autoencoders. *IEEE Access*, Vol. 8, pp. 149456–149467, 2020.

[173] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*, 2017.

[174] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[175] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, Vol. 28, pp. 3483–3491, 2015.

[176] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

[177] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[178] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[179] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

[180] Thibaut Issenhuth, Ugo Tanielian, Jérémie Mary, and David Picard. Edibert, a generative model for image editing. *arXiv preprint arXiv:2111.15264*, 2021.

[181] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[182] Yicun Liu, Jimmy Ren, Jianbo Liu, and Xiaohao Chen. Learning selfie-friendly abstraction from artistic style images. In *Asian Conference on Machine Learning*, pp. 113–128. PMLR, 2018.

[183] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[184] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[185] Zhiyuan Liu, Yankai Lin, and Maosong Sun. *Representation learning for natural language processing*. Springer Nature, 2020.

[186] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[187] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[188] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[189] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

[190] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1753–1762, 2015.

[191] Xing Niu, Marianna Martindale, and Marine Carpuat. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2814–2819, 2017.

[192] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 35–40, 2016.

[193] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. *Advances in neural information processing systems*, Vol. 30, , 2017.

[194] Daniel J Hirst. Automatic analysis of prosody for multilingual speech corpora. *Improvements in speech synthesis*, pp. 320–327, 2001.

[195] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Adaspeech: Adaptive text to speech for custom voice. *arXiv preprint arXiv:2103.00993*, 2021.

[196] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

[197] Kurniawati Azizah, Mirna Adriani, and Wisnu Jatmiko. Hierarchical transfer learning for multilingual, multi-speaker, and style transfer dnn-based tts on low-resource languages. *IEEE Access*, Vol. 8, pp. 179798–179812, 2020.

[198] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International conference on machine learning*, pp. 4364–4373. PMLR, 2018.

[199] Hiral Raveshiya and Viral Borisagar. Motion estimation using optical flow concepts. *International Journal of Computer Technology & Applications*, Vol. 3, No. 2, 2012.

[200] Hoda Rezaee Kaviani. *Novel Image Interpolation Schemes with Applications to Frame Rate Conversion and View Synthesis*. PhD thesis, 2018.

[201] Snježana Rimac-Drlje and Denis Vranješ. Fast frame-rate up-conversion method for video enhancement. In *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 1–4. IEEE, 2016.

[202] Jack Klys, Jake Snell, and Richard Zemel. Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 6444–6454, 2018.

[203] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2758–2766, 2015.

[204] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2462–2470, 2017.

[205] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pp. 4473–4481, 2017.

[206] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9000–9008, 2018.

[207] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 261–270, 2017.

[208] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.

[209] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5933–5942, 2019.

[210] Vladislav Samsonov. Deep frame interpolation. *arXiv preprint arXiv:1706.01159*, 2017.

[211] Jinhui Tang, Xiangbo Shu, Zechao Li, Guo-Jun Qi, and Jingdong Wang. Generalized deep transfer networks for knowledge propagation in heterogeneous domains. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 12, No. 4s, pp. 1–22, 2016.

[212] Xiangbo Shu, Guo-Jun Qi, Jinhui Tang, and Jingdong Wang. Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 35–44, 2015.

[213] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *International conference on machine learning*, pp. 552–560, 2013.

[214] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. *arXiv preprint arXiv:1806.05236*, 2018.

[215] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605, 2008.

[216] Ian Jolliffe. *Principal component analysis*. Springer, 2011.

[217] Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.", 2016.

[218] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, Vol. 92, No. 1, pp. 1–31, 2011.

[219] Alexander A Alemi, Ben Poole, Ian Fischer, Joshua V Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

[220] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *arXiv preprint arXiv:2002.07514*, 2020.

[221] Serena Yeung, Anitha Kannan, Yann Dauphin, and Li Fei-Fei. Tackling over-pruning in variational autoencoders. *arXiv preprint arXiv:1706.03643*, 2017.

[222] Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, Vol. 4, No. 1, pp. 66–82, 1960.

[223] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

[224] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[225] BKP Horn and BG Schunck. Determining optical flow artificial intelligence vol. 17, 1981.

[226] Andry Maykol G Pinto, A Moreira, P Costa, and M Correia. Revisiting lucas-kanade and horn-schunck. *J. Comput. Eng. Inf*, Vol. 1, pp. 23–29, 2013.

[227] Jiefu Zhai, Keman Yu, Jiang Li, and Shipeng Li. A low complexity motion compensated frame interpolation method. In *2005 IEEE International Symposium on Circuits and Systems*, pp. 4927–4930. IEEE, 2005.

[228] Jiang Li and Shipeng Li. Low complexity motion compensated frame interpolation method, September 13 2011. US Patent 8,018,998.

[229] Demin Wang, Andre Vincent, Philip Blanchfield, and Robert Klepko. Motion-compensated frame rate up-conversion—part ii: New algorithms for frame interpolation. *IEEE Transactions on Broadcasting*, Vol. 56, No. 2, pp. 142–149, 2010.

[230] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, Vol. 32, No. 4, pp. 1–10, 2013.

[231] Simone Meyer, et al. Phase-based frame interpolation for video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1410–1418, 2015.

[232] Zhoutong Zhang, Yebin Liu, and Qionghai Dai. Light field from micro-baseline image pair. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3800–3809, 2015.

[233] Mohamed Elgharib, Mohamed Hefeeda, Fredo Durand, and William T Freeman. Video magnification in presence of large motions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4119–4127, 2015.

[234] David Gadot and Lior Wolf. Patchbatch: A batch augmented loss for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4236–4245, 2016.

[235] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, Vol. 28, p. 24. ACM, 2009.

[236] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3250–3259, 2017.

[237] James Thewlis, Shuai Zheng, Philip HS Torr, and Andrea Vedaldi. Fully-trainable deep matching. *arXiv preprint arXiv:1609.03532*, 2016.

[238] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pp. 568–576, 2014.

[239] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 498–507, 2018.

[240] Jinhui Tang, Zechao Li, Hanjiang Lai, Liyan Zhang, Shuicheng Yan, et al. Personalized age progression with bi-level aging dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 40, No. 4, pp. 905–917, 2017.

[241] Yu Li, Dominik Roblek, and Marco Tagliasacchi. From here to there: Video inbetweening using direct 3d convolutions. *arXiv preprint arXiv:1905.10240*, 2019.

[242] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

[243] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[244] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9715–9724, 2019.

[245] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[246] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[247] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, Vol. 2, , 2010.

[248] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[249] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

[250] Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, Vol. 11, No. 10, pp. 428–434, 2007.

[251] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. The omniglot challenge: a 3-year progress report. *Current Opinion in Behavioral Sciences*, Vol. 29, pp. 97–104, 2019.

[252] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, Vol. 33, 2011.

[253] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[254] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 539–546. IEEE, 2005.

[255] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21–30, 2019.

List of Publications

Journal Articles

1. Paulino Cristovao, Hidemoto Nakada, Yusuke Tanimura, Hideki Asoh **Generating In-Between Images Through Learned Latent Space Representation Using Variational Autoencoders**. *– IEEE Access, vol 8, pp. 149456-149467 , 2020*

Conference Papers

1. Paulino Cristovao, Hidemoto Nakada, Yusuke Tanimura, Hideki Asoh **Few Shot Model based on Weight Imprinting with Multiple Projection Head**. *– International Conference on Ubiquitous Information Management and Communication - IMCOM 2021*

2. Paulino Cristovao, Hidemoto Nakada, Yusuke Tanimura, Hideki Asoh **Variational Information Bottleneck on Few Shot Model based on Weight Imprinting for Image Classification**. *– 2021 ASIAN CONFERENCE ON INNOVATION IN TECHNOLOGY , 2021*