Disentangled Representations beyond Vectors for Multimedia Content

Ting-Yao Hu

CMU-LTI-22-007

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Alexander G. Hauptmann, Chair
Alan W Black
Kris Kitani
Yu Tsao (Academia Sinica, Taiwan)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies

Copyright © 2022 Ting-Yao Hu

Keywords: Distribution/Subspace representation, Hidden p-norm regression, Mutual information minimization, Image set retrieval, Pose guided image generation, Speech synthesis

Dedicated to my family
for their wholehearted love and support all along the journey.

Abstract

In recent years, a tremendous amount of multimedia data is being generated and published on a variety of platforms such as Instagram, Podcast, Clubhouse, and YouTube. This phenomenon inspires the research works of large-scale multimedia analysis, including the foundation of analysis methodology, and some specific downstream applications (e.g. recognition, retrieval, and information extraction). Particularly, representation learning of multimedia is one of the most crucial research directions. A good feature representation for a multimedia data instance provides interpretability and generality, improving the performance and efficiency of downstream tasks.

It is challenging to obtain a good representation of multimedia content due to its richness and noisiness. For instance, in the task of speech processing, human speech utterances contain linguistic information, and other factors such as speaker identity, speaking style and background noise. In this case, we need a type of representation that captures the information from all these factors, and recovers the useful factors for downstream applications. Most of the mainstream techniques exploit a feature vector to represent each instance in a training dataset, and optimize the feature extractor by conducting a pretraining task. However, vector based representation is not enough to preserve the richness and handle of the noisiness of multimedia data. Also, common pretraining procedures, such as the ImageNet classification task in computer vision research area, only focus on a single type of discriminative information, which might be insufficient for certain applications. Thus, in this thesis, I explore two research directions addressing these issues.

In the first part of this thesis, I develop two new types of representation: a probability distribution and a linear subspace, for multimedia content. Compared with vector based representation, both of them are capable of dealing with the richness and noisiness of multimedia. To leverage the two types of representation in downstream tasks, it is essential to design particular algorithms and training strategies. In this part of thesis, I introduce methods incorporating distribution and subspace representations with deep neural network architectures, which can be optimized in an end-to-end manner. The experiment results on downstream tasks show that two proposed representations yield better performance comparing to mainstream vector based methods.

In the second part of this thesis, I investigate style and content disentanglement techniques, which explicitly preserve different factors within multimedia content during the representation learning process. The disentangled representation provides better interpretability, and enables the manipulation of hidden factors in data synthesis scenarios. Based on this motivation, I propose two methods to effectively separate the hidden factors in multimedia data. The first method models the relation between style and content as a simple matrix operation in hidden feature space. The second method minimizes the mutual information between two hidden factors by

formulating an adversarial training criterion. The advantages of the two proposed methods are evaluated in qualitative and quantitative experiments of data synthesis/generation tasks. Besides, I further demonstrate the applicability of style and content disentanglement techniques by constructing a pretraining framework with generative models. Specifically, the synthetic data produced by the generative models can support the supervised training process of downstream tasks, such as speech recognition and person re-id. Also, the disentangled generative process extends the idea of data augmentation from the raw data space to an interpretable representation space, allowing us to incorporate more prior knowledge in downstream tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent years, a tremendous amount of multimedia data is being generated everyday. These data are stored and published on a variety of plate-forms such as Instagram, Podcast, Clubhouse, and YouTube. This phenomenon motivates the investigation of large-scale multimedia analysis. Research topics about the foundation of analysis methodology, and some specific downstream applications (e.g. recognition, retrieval, and information extraction) attract the attention of researchers nowadays. Particularly, representation learning of multimedia is one of the most crucial research directions, which receive much research attention in recent years. For example, in the context of large scale multimedia retrieval, previous works [16, 152, 161] rely on hash code as representation for each data instance. The purpose of using hash code is to enhance the efficiency of retrieval. In general, a good feature representation for a multimedia data instance provides interpretability and generality, improving the performance and efficiency of downstream tasks.

It is challenging to obtain a suitable representation for multimedia content due to its richness and noisiness. For instance, in speech processing, human speech utterances contain linguistic information, and other factors such as speaker identity, speaking style and background noise. On the other hand, in the view of an image of a video, one can observe not only objects of interest but also some noisy background. In these cases, we need a type of representation that captures the information from all these hidden factors, and recovers the useful factors during downstream applications. In the area of representation learning, most of the main stream techniques exploit a feature vector to represent each instance in a training dataset, and optimize the feature extractor by conducting a pretraining task. However, vector based representation is not enough to preserve the richness and handle of noisiness of multimedia data. Figure 1.1 illustrates the limitation of vector based representation. Also, common pretraining procedures, such as the ImageNet classification task, only focus on single type of discriminative information, which might be insufficient to certain applications. Thus, in this thesis, we explore two research directions addressing these issues.

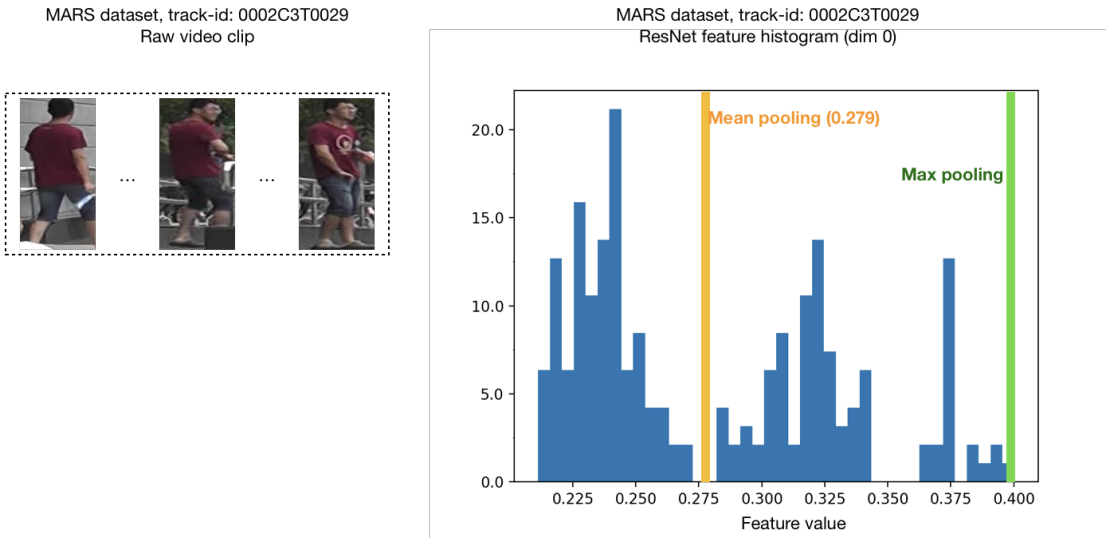


Figure 1.1: The limitation of vector based representation for multimedia content. In a image set retrieval task (e.g. multi-shot person re-id), we observe that the local representation in hidden space has multi-mode property. However, mainstream approaches aggregate local hidden representation by pooling strategies (e.g. mean/max pooling), which fail to preserve the multi-mode property.

1.2 Structure of this thesis

In the first part of this thesis, we explore novel types of representation beyond vectors for multimedia content. Specifically, we propose to use a probability distribution (chapter 2) [55, 57] or a linear subspace (chapter 3) [59] to represent a data instance in a multimedia dataset. Along with two novel types of representation, we extend the concept of metric learning from vector space to distribution and subspace, respectively. To leverage the two types of representation in downstream tasks, it is essential to design particular algorithms and training strategies. In this part of thesis, we introduce methods incorporating distribution and subspace representations with deep neural network (DNN) architectures, which can be optimized in an end-to-end manner. For distribution representation, we propose a Statistical Distance Metric Learning (SDML) framework, which integrates the density estimation and the calculation of statistical distance between two distributions to a DNN architecture. Similarly, for subspace representation, we establish a subspace representation learning (SRL) framework, which extracts a subspace in local CNN feature space for each data instance, and supports meta learning via optimization on Stiefel manifold. Compared to deep metric learning with vector based representation, both proposed methods circumvent the need of feature aggregation step and successfully deal with the richness and noisiness of multimedia. Also, the metric learning techniques with distribution and subspace don't introduce additional trainable parameters over that with feature vector, but do increase the computation requirement. The experiment results on image set retrieval and few-shot image classification tasks illustrate that two proposed representations yield better performance comparing to mainstream vector based methods.

The second part of this thesis aims at the extraction of multiple hidden factors in multimedia

content. A well-known approach to this goal is to discover a unsupervised disentangled representation for these hidden factors [19, 21, 49, 67]. These approaches optimize their representation encoders while enforcing the discovered hidden factors to be uncorrelated to one another. However, a previous work [89] also argues that it is extremely difficult to achieve the disentanglement without proper inductive bias or supervision. In this part of the thesis, we approach this final goal via the investigation of style and content disentanglement. This technique explicitly defines the "content", and preserves style and content hidden factors within multimedia data during the representation learning process. The disentangled representation provides better interpretability, and enables the manipulation of hidden factors in data synthesis scenario. Based on this motivation, we propose two methods to effectively separate the hidden factors in multimedia data. The first method (chapter 4) [56] models the interaction between style and content as a simple matrix operation in hidden feature space. Based on this idea, the decomposition of style and content can be interpreted as solving a regression problem in the hidden space. This method is effective and flexible, applicable to multiple training/testing scenarios. The second method (chapter 5) [58] explicitly minimizes the mutual information (MI) between two hidden factors. By using a neural estimator of mutual information [14], the MI minimization can be formulated as an adversarial training criterion. The advantages of two proposed methods are evaluated in qualitative and quantitative experiments of two data synthesis/generation tasks: pose guided person image generation and speech synthesis. In addition, we demonstrate the applicability of our style and content decomposition methods by introducing a pretraining framework with generative models (chapter 6). Utilizing the methods developed in chapter 4 and 5, we control the hidden factors of multimedia data, and customize the data generation process. By doing so, one can use synthetic datasets to train recognizers for different types of downstream multimedia applications. Particularly, if we want to recognize some attributes (style factors) in multimedia which we don't have any annotation, the our framework manipulates the other attributes (content factors) in the data generation step, and creates a proper synthetic dataset for supervised training techniques. In chapter 6, we consider speech recognition and low-resource person re-identification as examples of downstream tasks to illustrate this capability. Furthermore, the pre-training framework also extends the idea of data augmentation from raw data spaces to interpretable representation spaces. It helps to incorporate more prior knowledge in the training process, enhancing the robustness of the trained recognizers.

The list of publications

1.3 Summary of Contribution

The main contribution of this thesis are three folds:

1. We propose two novel types of representation for multimedia content: distribution and subspace. They result in the extension of distance metric learning concept. Thus, one can adapt the techniques developed for vector-based distance metric learning to distribution and subspace. Both of them yield superior performance than vector based representation. The related papers of this part are
   (a) [55] Ting-Yao Hu and Alexander G. Hauptmann, "Multi-shot person re-identification through set distance with visual distributional representation," ICMR 2019

(b) [57] Ting-Yao Hu and Alexander G. Hauptmann, "Statistical distance metric learning for image set retrieval," ICASSP 2021

(c) [59] Ting-Yao Hu, Zhi-Qi Cheng and Alexander G. Hauptmann "Subspace representation learning for few-shot image classification," arXiv preprint arXiv:2105.00379, 2021

2. We investigate two novel style and content disentanglement methods for multimedia data, approaching the analysis of multiple hidden factors. These two methods are evaluated on two data synthesis/generation tasks, controllable text-to-speech (TTS) and keypoint guided image generation. The results demonstrate the ability of hidden factor control in the generation process. The corresponding publications of this part are:

   (a) [58] Ting-Yao Hu, Ashish Shrivastava, Oncel Tuzel, and Chandra Dhir, "Unsupervised style and content separation by minimizing mutual information for speech synthesis," ICASSP 2020.

   (b) [56] Ting-Yao Hu and Alexander G. Hauptmann, "Pose guided person image generation with hidden p-norm regression," ICIP 2021

3. We explore the applicability of disentangled representation by introducing a pre-training framework with generative models. By manipulating the hidden factors of multimedia during the generative process, one can synthesize customized data for model training. This pre-training framework also empowers the data augmentation in interpretable representation spaces, incorporating additional prior knowledge to improve the downstream tasks.

Chapter 2

Representation beyond Vector: Distribution

2.1 Introduction

It is crucial to pick a good content representation for multimedia analysis. Most recent works choose to represent a data instance as a fixed size feature vector. This design choice is intuitive and advantageous in certain aspects. For example, in the applications such as recognition and retrieval, one needs to measure the similarity between two instances. It can be done by calculating a distance metric in vector space (e.g. Euclidean distance). Also, with the recent development of deep metric learning (DML) techniques [48, 50, 54, 98, 103, 134, 166], the extraction of the feature vector is implemented as a DNN architecture, which enhances the modeling capability of vector based representation. However, the extraction of vector representation includes a step to aggregate all the local information by pooling mechanisms, which might deteriorate the capability of handling the richness and noisiness of multimedia content.

In this chapter, we utilize the image set retrieval task as a concrete example. Aiming to search the identity captured by a set of images, image set retrieval has been a crucial problem in recent years, especially due to the rapid development of camera surveillance systems, and the availability of multiple images of the same identity. There exist many practical applications for image set retrieval such as video face recognition [113, 164], multi-shot person re-identification [52, 174] and gait recognition [18, 160]. Comparing to a single image, a set of images usually contains richer discriminative information and provides better performance. However, it is also more challenging to analyze an image set and measure similarity between two sets because of the large intra-set diversity and noisiness.

Many recent approaches for image set retrieval benefit from DML, which learns a proper metric space to capture data similarity. To deal with the varied number of images within a set, these approaches usually extract image-level features to form a feature set, and aggregate them as a fixed size vector using pooling strategies, such as mean/max pooling and attention mechanism. However, using DML with feature aggregation has several drawbacks. First, a fixed size vector is not able to properly summarize all the information observed from an image set. For example, feature aggregation may wash out the multi-mode property (multiple peaks in the probabilistic

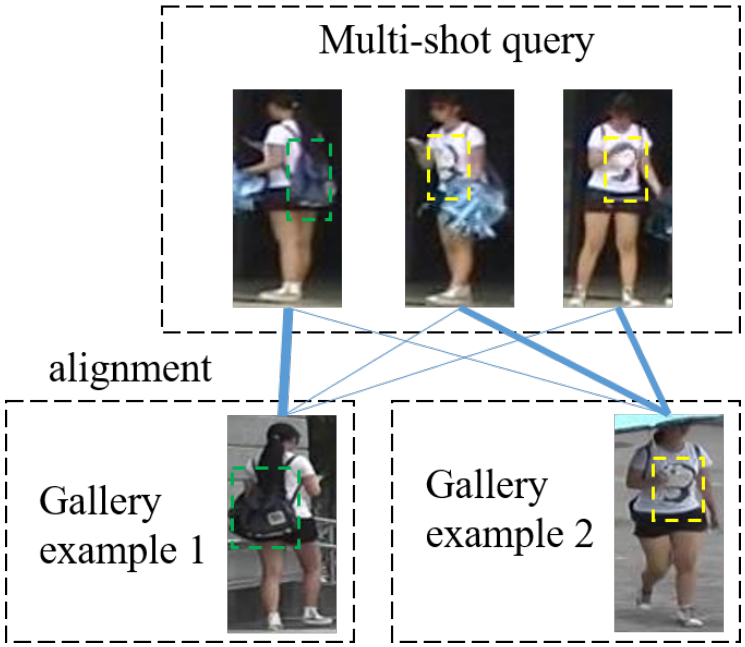


Figure 2.1: A good representation for a image set should preserve the matching evidences located in different elements of the set, so that they can be utilized for retrieval task.

density). Second, matching pieces of evidence captured by single or few images in the whole set (as shown in Fig. 2.1) may be eliminated by pooling methods.

In this chapter, we propose to use a probability distribution to represent an instance in a multimedia dataset. Specifically, we extract a set of features from the spatial and temporal local regions of multimedia content, such as patches of an image and frames of a video clip. Then, these feature vectors are treated as empirical samples drawn from an underlying distribution in hidden space, which characterizes the original multimedia data instance. Similar to DML, the modeling capacity of distribution representation can also be enhanced by leveraging the deep learning architectures in the feature extraction step. To do so, we introduce a statistical distance metric learning (SDML) framework. The SDML framework exploits a statistical distance between two distributions to measure the similarity between two data instances, and optimizes the backbone deep network for feature extraction in an end-to-end manner. Among all types of statistical distance, we select Wasserstein distance (WD) and Jeffrey's distance (JD) considering the efficiency of their empirical estimation. Compared with DML with vector based representation, SDML can properly describe the diversity and uncertainty of an image or a video clip. Also, all the techniques developed for DML in vector space can possibly be applied to SDML in probability measure space. In this chapter, we propose two types of loss functions for SDML, which are analogous to the triplet loss [151] and center loss [159] in DML.

The proposed SDML framework is designed to model diversity and uncertainty within an instance of multimedia datasets. It can be leveraged to model the multimedia data with a varied amount of information, such as speech utterances and videos with different duration and images of different sizes. In this chapter, we evaluate the SDML framework on two image set retrieval tasks: multi-shot person re-id and gait recognition. The experiment results show that SDML

outperforms conventional DML with feature aggregation, and also receives competitive/superior performance comparing to the previous state-of-the-art methods on the aforementioned tasks. From a qualitative investigation, we also observe that SDML method captures alignment of matching evidence, and the multi-mode property of raw data.

In summary, the main contributions in this chapter are three-fold: (1) We propose a distribution based representation for multimedia, and extend the concept of deep metric learning (DML) from vector space to probability measure space by proposing SDML. (2) To apply SDML framework to image set retrieval, we exploit empirical WD and JD to measure the similarity between two image sets, and propose a new statistical-centroid loss to enhance the training process. (3) Experiment results on two image set retrieval tasks, multi-shot person re-id and gait recognition, show that the proposed SDML outperforms DML and provides competitive/superior performance comparing to state-of-the-art approaches.

2.2 Related Work

Deep metric learning (DML) methods have been successfully applied to many areas ranging from image retrieval [98, 103], face recognition [54], person re-id [166], and speaker recognition [78, 153]. The goal of this technique is to learn a mapping function transforming the raw data into a feature embedding space. The distance function in this space should preserves the semantic similarity between raw data samples. Several loss functions have been proposed to measure the quality of the feature embedding space. Prominent examples include triplet loss [151] and center loss [159]. Triplet loss forces the distance between the positive pair to be smaller than the distance between the negative pair with a margin. On the other hand, center loss minimizes the distance between samples and their corresponding centers to reduce the intra-class variance. In comparison, our proposed statistical-centroid loss learns an empirical distribution as the template for each class, which is used to obtain the intra-class variance among image sets.

In order to calculate the distance between two image sets, most of these works aggregate the features of all the images within a set using pooling functions, and form a fixed size feature vector. Only a few works tried to avoid feature aggregation. In [90] and [61], authors modeled a image set as a manifold, or a subspace of the embedding feature space, and compare image sets by using manifold-manifold distance. Specifically, Lu et al. [90] computed the average distance from each element to the nearest neighbors of the other set, and Huang et al. [61] chose the approximation of Grassmannian geodesic distance, which is the Frobenius norm of the difference between two covariance matrices.

In this work, we choose Wasserstein distance and Jeffrey's divergence as the statistical distance, the core component of SDML. Wasserstein distance (WD) has been successfully applied to many different areas, such as generative adversarial network (GAN) training [8], multi-label classification [34], representation learning [105], dictionary learning [119], and domain adaptation [127]. Jeffrey's divergence (JD), also known as symmetric KL divergence, has mostly been applied to signal processing tasks [93, 94].

The empirical estimation of WD is usually formulated as a linear programming task. Thus, the training of SDML with WD can be treated as a bi-level optimization problem, which consists of an outer problem and inner problem The former needs to be solved subject to the optimality

of the latter. It has been utilized in many tasks of computer vision or machine learning, such as hyper-parameter optimization [33], multi-task learning [31], image segmentation [112], and video classification [29]. In order to conduct end-to-end training, most of the methods require the inner problem to be twice differentiable. Several recent works [39, 102] incorporate a differentiable update rule of non-smooth inner problems to solve the whole problem with gradient based method.

2.3 Statistical Distance Metric Learning

In this section, we introduce the proposed Statistical Distance Metric Learning (SDML). Specifically, we will start from overview of SDML in the context of image set retrieval tasks, then discuss the three key components: (1) the empirical estimation of statistical distance (2) training loss functions (3) bi-level optimization (for SDML with Wasserstein distance).

2.3.1 Overview of SDML

Given a dataset $D = \{(X_i, y_i)\}$, where $X_i = \{x_i^k | k = 1, 2..., n_i\}$ is a set of images, and y_i is the corresponding label of X_i, the proposed method aims to learn a embedding function F to extract the image-level embedding feature $z_i^k = F(x_i^k)$. Using this function F, we are able to transform the original image set to a embedding feature set $Z_i = \{z_i^k | k = 1, 2..., n_i\}$, where n_i is the size of this image/feature set. Following the spirit of deep metric learning, these embedding feature sets should preserve the label information within the distance measure. That is, the embedding feature sets of the same class should be pulled together, and the sets from different classes should be separated from each other. In order to achieve this goal, we treat a set of embedding features as empirical measure of a probability distribution, and estimate the statistical distance between two distributions, which is exploited to describe the set-to-set similarity.

Comparing to conventional DML methods for image set, representing an image set as a distribution preserves the diversity and uncertainty. For instance, if the underlying distribution of an image set is multi-modal, SDML framework can easily capture this property. Fig. 2.2 illustrates the overall architecture of SDML framework in the context of image set retrieval.

2.3.2 Empirical Estimation of Statistical Distance

The proposed SDML framework requires a statistical distance to measure the similarity between two distributions, given their empirical sample sets, Z_i and Z_j, in embedding feature space. While any types of statistical distance can be integrated with SDML, we explore Wasserstein distance and Jeffrey's divergence in this chapter, because of the efficiency of their empirical estimators.

Wasserstein distance (WD) is the first selected type of statistical distance. The empirical estimation process of WD starts from a kernel density estimation (KDE) with Dirac delta. Specifically, given Z_i and Z_j, the two underline distributions can be represented by: $\nu_{Z_i} = \frac{1}{n_i}\sum_{k=1}^{n_i}\delta(z_i^k)$ and $\nu_{Z_j} = \frac{1}{n_j}\sum_{l=1}^{n_j}\delta(z_j^k)$, the average of n_i and n_j Dirac delta with mass at

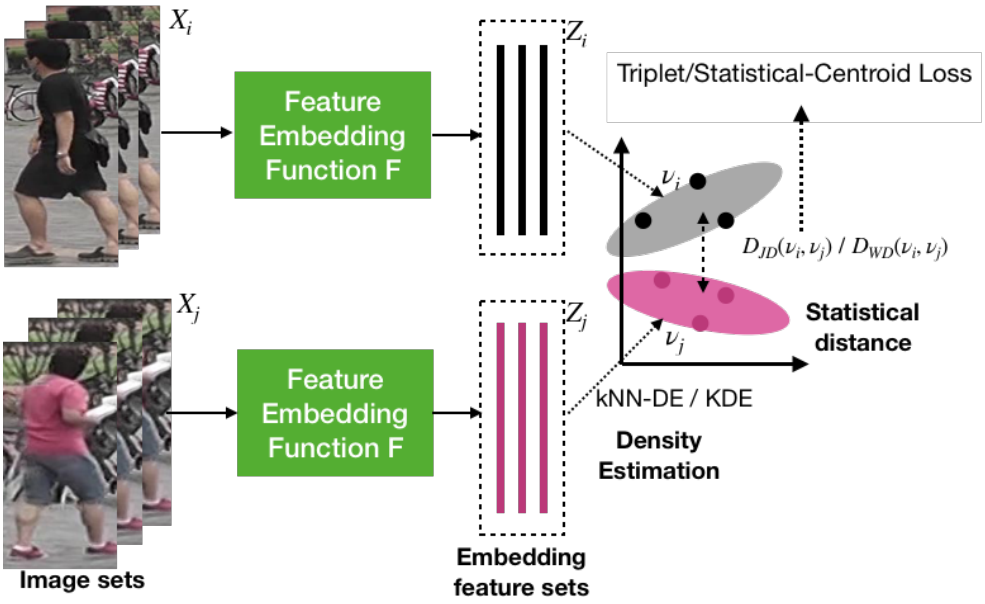


Figure 2.2: The overall architecture of SDML in the context of a image set retrieval task (multi-shot person re-id).

positions z_i^k and z_j^l. Then, the WD between ν_{Z_i} and ν_{Z_j} is expressed as:

$$D_{WD}(Z_i, Z_j) = \min_{P \in \Pi(n_i, n_j)} \langle P, M \rangle_F \tag{2.1}$$

where M is a $n_i \times n_j$ distance matrix and M_{kl} stores the Euclidean distance between z_i^k and z_j^l. $\Pi(n_i, n_j) = \{P \in \mathbb{R}^{n_x \times n_y} | P\mathbf{1} = \frac{\mathbf{1}}{n_j}, P^T\mathbf{1} = \frac{\mathbf{1}}{n_i}\}$ is the feasible set of joint distributions, which are also encoded as $n_i \times n_j$ matrices. Please note that eq. 2.1 is a linear programming problem. Thus, when we apply WD to SDML framework, the overall training process for feature embedding function becomes a bi-level optimization task. The detail of the bi-level training procedure will be elaborated in the Sec. 2.3.4

The second choice of statistical distance is Jeffrey's Divergence (JD), the symmetric version of KL divergence. Given a set of embedding features, $Z_i = \{z_i^k | k = 1, 2..., n_i\}$, one can estimate the underlying density function ν_i using the nearest-neighbor based method developed in [108]:

$$\nu_i(z) = \frac{\Gamma(d/2+1)}{n_i \pi^{d/2} r(z, Z_i)^d} \tag{2.2}$$

where $\pi^{d/2}/\Gamma(d/2+1)$ is the volume of a d-dimensional unit-ball, $r(z, Z_i)$ is the Euclidean distance between z and its nearest neighbor in Z_i. If $z \in Z_i$, the candidate set of nearest neighbor would become $Z_i \setminus z$. Based on Eq. (2.2), the empirical JD between Z_i and Z_j can be expressed as:

$$\begin{aligned} D_{JD}(Z_i, Z_j) &= \hat{KL}(\nu_i || \nu_j) + \hat{KL}(\nu_j || \nu_i) \\ &= \frac{d}{n_i} \sum_{k=1}^{n_i} \log \frac{r(z_i^k, Z_j)}{r(z_i^k, Z_i)} + \frac{d}{n_j} \sum_{l=1}^{n_j} \log \frac{r(z_j^l, Z_i)}{r(z_j^l, Z_j)} + C \end{aligned} \tag{2.3}$$

where constant C is independent to Z_i and Z_j. The estimator described in Eq. (2.3) requires pair-wise Euclidean distance computation, which can be parallelized easily.

2.3.3 Loss Functions

Similar to DML, the design of training loss function is a crucial step for SDML framework. In this section, we introduce two types of loss functions: triplet loss and statistical-centroid loss. The weighted combination of the two loss functions serves as the final training objective for all the experiments in this chapter.

The first loss function is directly adapted from the popular triplet loss [151]. In the context of SDML, we assume $\Gamma = \{(a, p, n)\}$ being the set of triplets, and Z_a, Z_p, Z_n being the embedding feature sets of anchor, positive, and negative samples, respectively. Anchor and positive samples belong to the same class, while the negative sample is from a different one. Triplet loss for SDML can be written as following:

$$\mathcal{L}_{Triplet} = \frac{1}{|\Gamma|} \sum_{(a,p,n)\in\Gamma} max(0, D(Z_a, Z_p) - D(Z_a, Z_n) + \Delta) \tag{2.4}$$

where D is statistical distance (D_{WD} or D_{JD}). Minimizing this loss function aims at forcing the distance between the positive pair (Z_a, Z_p) to be smaller than negative pair (Z_a, Z_n) with a margin Δ.

The second loss function is the proposed statistical centroid (SC) loss. It is designed based on center loss [159] in DML methods, which can be represented as:

$$\mathcal{L}_{Center} = \frac{1}{N} \sum_{i} ||z_i - c_{y_i}|| \tag{2.5}$$

where c_k is the center vector of the $k-th$ class, and N is the size of the whole dataset. Comparing to the triplet loss, center loss explicitly constraints the intra-class variation to increase the discriminative power of embedding feature.

In context of SDML, each image set is represented as a empirical distribution. Hence, instead of a center vector, we use the statistical centroid (barycenter) of a set of distributions [2, 10, 100] to describe the template of each class. The statistical centroid of the $c-th$ class, μ_c, is the minimizer of the following objective function:

$$f_{SC}(\mu_c) = \frac{1}{N_c} \sum_{i|y_i=c} D(Z_i, \mu_c) \tag{2.6}$$

where N_c is the number of image sets belonging to the $c-th$ class. Combining the idea of center loss, and statistical centroid of distributions, we propose SC loss:

$$\mathcal{L}_{SC}(Z, \mu) = \frac{1}{N} \sum_{i} D(Z_i, \mu_{y_i}) = \sum_{c} \frac{N_c}{N} f_{SC}(\mu_c) \tag{2.7}$$

which is equal to the weighted summation of the SC objective functions of each class. Similar to center loss, SC loss maintains a template μ_c for each class. In this work, we assume that μ_c is also an empirical distribution with a fixed number of supports.

Combining Eq. (2.4) and (2.7), we formulate the training objective as:

$$\min_{F,\mu} \mathcal{L}_{Triplet} + \lambda \mathcal{L}_{SC} \tag{2.8}$$

where λ controls the balance between two loss functions. Then, the parameters of feature embedding function, F, and the set of template distributions, μ, are jointly learned by minimizing this objective in an end-to-end manner.

2.3.4 Bi-level Optimization

Since the estimation of Wasserstein distance (WD) is a optimization problem itself, the training process of SDML framework with WD is treated as a bi-level optimization problem:

$$\min_{F,\mu} \mathcal{L}_{Triplet} + \lambda \mathcal{L}_{SC} \tag{2.9}$$

$$s.t. \quad D_{WD}(Z_i, Z_j) = \min_{P \in \Pi(n_i, n_j)} \big\langle P, M \big\rangle_F \tag{2.10}$$

Inspired by previous works [32, 39, 102], we solve the bilevel problem (eq. (2.9)) by exploiting the automatic differentiation of fix point iteration update, which serves as the solver of the inner problem (eq. (2.10)). In this work, we choose a recent proposed differentiable update rule, Inexact Proximal point method for exact Optimal Transport (IPOT) [162] for WD calculation.

IPOT algorithm finds the best transportation plan P in empirical WD estimation (eq. (2.10)) by exploiting generalized proximal point method. The process of IPOT can be summarized as the following iterative update rule:

$$P^{(t+1)} = \arg\min_{P} \big\langle P, M \big\rangle_F + \beta^{(t)} D_h(P, P^{(t)}) \tag{2.11}$$

where $D_h(P, P^{(t)})$ is the Bregman divergence based on KL divergence. One step of this update rule also forms an optimization problem, which can be solved by executing Sinkhorn iteration [132]. Practically, IPOT algorithm does not solve eq. (2.11) exactly in each update, but conducts a single Sinkhorn iteration. After executing the update rule (eq. (2.11)) L times, we receive a sequence of transportation plan matrices $P^1, P^2, ..., P^L$, and the approximated WD can be obtained by:

$$\hat{D}_{WD}(Z_i, Z_j) = \big\langle P^{(L+1)}, M \big\rangle_F \tag{2.12}$$

The overall process of IPOT is shown in Algorithm 1. $\mathbf{1_n}$ represents an all-one vector with dimension n. Please note that this algorithm and the original IPOT differ in some aspects. First, Algorithm 1 assumes that every elements in Z_1 and Z_2 are uniformly weighted, while the original IPOT considers different weights for them. Second, Algorithm 1 conducts L times of updates while original IPOT keeps doing until $P^{(l)}$ converges. Because of the computation constraint, we set $L = 10$ in our implementation.

In the previous works considering auto-differentiation, IPOT was not a typical choice. Instead, they made use of another type of Wasserstein distance approximation with entropy regularization, which is known as Sinkhorn distance. The reasons why we select IPOT in this work

Algorithm 1 IPOT Algorithm [162] for eq. (2.10)

Input: Feature sets Z_1 and Z_2, number of iteration L, β
Output: Estimated Wasserstein distance $\hat{D}_{WD}(Z_1, Z_2)$

initialize $M \in \mathbb{R}^{n_1 \times n_2}, M_{i,j} = ||z_1^{(i)} - z_2^{(j)}||$
$b \leftarrow \frac{1}{n_2}\mathbf{1_{n_2}}, G_{i,j} \leftarrow e^{\frac{-M_{i,j}}{\beta}}, P^{(1)} \leftarrow \mathbf{1_{n_1}}\mathbf{1_{n_2}}^T$
for $l = 1, 2, .., L$ **do**
    $Q \leftarrow G \odot P^{(l)}$
    $a \leftarrow \frac{1}{n_1 Qb}, b \leftarrow \frac{1}{n_2 Q^T a}$
    $P^{(l+1)} \leftarrow diag(a) Q\, diag(b)$
end for
$\hat{W}(Z_1, Z_2) = \left\langle P^{(L+1)}, M \right\rangle_F$

are: (1) it converges to the exact WD. (2) it is less sensitive to the weight of proximal regularizer, β [162].

Algorithm 2 describes the overall training scheme for SDML with WD. This algorithm updates F and μ_k alternatively by gradient-based method with mini-batch. The estimation of WD is accomplished by algorithm 1, so that the gradient of $\mathcal{L}$ w.r.t F and μ_k can be easily calculated. Please note that the learning rates for ϵ_F and ϵ_{μ_k} are different, and usually ϵ_{μ_k} is much larger. More details will be elaborated in the experiment section.

Algorithm 2 Bilevel Training for SDML with WD

Input: image sets and labels $\{X, y\}$, learning rates $\epsilon_F, \epsilon_{\mu_k}$
Output: feature embedding function F

Initialization F, μ_k
while F, μ not converged **do**
    Sample a mini-batch of $(X_i, y_i), i = 1, 2..., B$.
    $Z_i = F(X_i), i = 1, 2, ..., B$
    for i=1,2,...,B **do**
        Obtain $D_{WD}(Z_i, \mu_{y_i})$ by Algorithm 1
        for j=1,2,...,B **do**
            Obtain $D_{WD}(Z_i, Z_j)$ by Algorithm 1
        end for
    end for
    Calculate $\mathcal{L}_{Triplet}$ and $\mathcal{L}_{SC}$ by eq. (2.4), (2.6) and (2.7)
    $\mathcal{L} = \mathcal{L}_{Triplet} + \lambda \mathcal{L}_{SC}$
    Obtain $\nabla_F \mathcal{L}, \nabla_\mu \mathcal{L}$ through auto-differentiation
    $F \leftarrow F + \epsilon_F \nabla_F \mathcal{L}, \mu \leftarrow \mu + \epsilon_{\mu_k} \nabla_\mu \mathcal{L}$
end while

Figure 2.3: Examples of three image set retrieval tasks: gait recognition (top), multi-shot person re-identification (middle), and video face verification (bottom).

2.4 Experiment

The proposed SDML framework is evaluated on two image set retrieval tasks: multi-shot person re-identification and gait recognition. In Figure 2.3, one can compare the appearance characteristics between these two tasks.

2.4.1 Multi-shot Person Re-identification

For multi-shot re-identification, we use two large scale datasets: MARS [174] and LPW [135]. The MARS dataset contains 1,261 identities and 20,715 tracklets captured by 6 cameras in different views. The bounding boxes of tracklets are detected and tracked by DPM detector and GMMCP tracker, respectively. 3,248 distractor tracklets appeare due to false detection or tracking. LPW dataset contains 2,731 identities and 7,694 tracklets collected from 11 cameras installed in 3 different scenes. The bounding boxes and tracklets are generated automatically, and cleaned up manually. We follow the evaluation protocol of the original works [135, 174], reporting Cumulated Matching Characteristics (CMC) on both datasets, and mean average precision (mAP) on MARS.

Following the suggestion of many previous works [35, 52, 83, 87], we adopt ResNet-50 pretrained on ImageNet as feature embedding function. The spatial average pooling of its last convolutional layers is extracted, and form a 2048 dimensional feature vector for each image. In the training phase, the number of tracklets in each minibatch is 32, and 8 images are randomly selected for each tracklet. The margin of Wasserstein triplet loss function is set to 0.3. The parameters in feature embedding function F and the supports of statistical centroid μ_k are optimized using Adaptive Moment Estimation (ADAM) algorithm [70]. The learning rates of F and μ_k start at 0.0001 and 0.01, respectively, and decrease by a factor 0.1 every 100 epochs, until the model finishes training at 400 epochs. For SC loss, the control parameter λ is set to 0.0001 and the number of supports is 3 for both datasets. Most of the previous works notice that cross-entropy (CE) loss works very well together with triplet loss. Hence, we also incorporate the CE loss by adding a distance classifier [20] branch in training phase. During the testing phase, all

the images of a tracklet are used to extract the embedding feature set.

Table 2.1 shows the comparison among recent state-of-the-art approaches, the baseline implementation, and SDML with WD and JD. One can see that our baseline system (Triplet + Center + CE) is already competitive. SDML frameworks with WD and JD both improve the performance upon the baseline. SDML with JD receives the best score in mAP, and the second best scores in top-k accuracy. A very recent proposed approach, NVAN [87], achieves the best performance in top-1 accuracy. While their model utilizes non-local blocks to learn the spatial/temporal relationship within a image set, it may require more data to support training.

In Table 2.2, we report the performance of all methods on LPW dataset. Only a few results were published on this dataset. In general, the performance reported on LPW are lower than those on MARS. This set of experiment results indicates that the proposed SDML outperforms other previous methods by a large margin, and SDML with JD achieves the best performance.

Methods	Top-1	Top-5	Top-20	mAP
MARS [174] (2016)	68.3	82.6	89.4	49.3
SeeForest [177] (2017)	70.6	90.0	97.6	50.7
TriNet [47] (2017)	71.8	86.6	93.1	56.5
RQEN [135] (2018)	73.7	84.9	91.6	51.7
DRA [83] (2018)	82.3	-	-	65.8
OSM+CAA [154] (2019)	84.7	94.1	97.0	72.4
STA [35] (2019)	86.3	95.7	97.1	80.8
VRSTC [52] (2019)	88.5	**96.5**	-	82.3
GLTR [80] (2019)	87.0	95.8	**98.2**	78.5
NVAN [87] (2019)	**90.0**	-	-	82.8
Triplet+Center+CE	87.7	96.0	97.9	81.7
SDML (WD)	88.8	96.3	98.1	83.1
SDML (JD)	89.3	96.4	98.1	**83.9**

Table 2.1: Comparison with state-of-the-art methods on MARS dataset.

2.4.2 Gait Recognition

For gait recognition, we conduct experiments on CASIA-B dataset [168]. It consists of 124 subjects with 11 different viewing angles. In each view angle, one subject was captured by 10

Methods	Top-1	Top-5	Top-20
RQEN [135] (2018)	57.1	66.7	91.5
OSM+CAA [154] (2019)	71.7	89.8	96.6
Triplet + Center+ CE	74.6	92.5	97.1
SDML (WD)	78.8	93.4	**97.8**
SDML (JD)	**79.6**	**93.9**	**97.8**

Table 2.2: Comparison with state-of-the-art methods on LPW dataset.

Methods	NM	BG	CL
CNN-LB [160] (2017)	-	72.4	54.0
GaitSet [18] (2019)	95.0	87.2	70.4
Triplet + Center	94.8	86.8	69.8
SDML (WD)	**95.1**	89.0	**74.5**
SDML (JD)	**95.1**	**89.2**	**74.6**

Table 2.3: Comparison with state-of-the-art methods on CASIA-B dataset. All the numbers are rank-1 accuracy

sequences of silhouette images under three walking conditions: six sequences are in "normal" (NM) state, two sequences in "wearing coat" (CL) state, and two sequences in "carrying bag" (BG) state. The dataset is collected indoors and each sequence lasts about 5 seconds. We emulate previous works, using the first 74 subjects for training and the rest 50 subjects for testing. During testing phase, the gallery set is composed of four NM sequences for each identity, while the remaining six sequences (two for each condition) are in the probe set.

We follow the design from Chao et al. [18] to construct the feature embedding function F, which consists of 6 layers of CNN and a Horizontal pyramid pooling (HPP) layer (without the Multiple Global Pooling (MGP) strategy the authors proposed). In the training phase, F and μ are optimized using ADAM algorithm [70] with learning rates 0.0001 and 0.01, respectively. For SC loss, the control parameter λ is set to 0.0005 and the number of support is 3.

Table 2.3 summarizes the performances of our approach and previous state-of-the-arts on CASIA-B dataset. The probe set is divided into three subsets according to the walking conditions, and the results on these probe subsets are reported separately. While calculating the performance of a given query, the gallery sequences with the same camera angles are not considered. Observing this set of results, we first notice that GaitSet [18], the baseline and SDML all achieve high score (around 95%) in top-1 accuracy for NM probe subset. For BG and CL probe subsets, SDML surpass GaitSet by 2% and 4%, respectively. SDML frameworks with WD and JD perform similarly in this task.

2.4.3 Visualization

To understand hidden semantic behind SDML training, we visualize (1) the supports of empirical distributions forming statistical centroids, and (2) the matching pattern between two sets discovered by WD estimation.

For the visualization of statistical centroids, we conduct SDML training with JD on MARS, LPW and CASIA-B datasets. The number of supports is set to be 2 for each identity. Once the training process is finished, the nearest neighbor images of the learned supports are selected from all the image sets of a identity. Several examples of statsitcal centroid are illustrated in Figure 2.4. The images located in the same rectangular are the nearest neighbor images of the same SC supports in the embedding feature space. One can see that the centroids of gait recognition data (CASIA-B) seem to capture the pose of the person. One of the examples from MARS (MARS, pid:0279) focuses on the human pose as well. The other example from MARS (MARS, pid:0107)

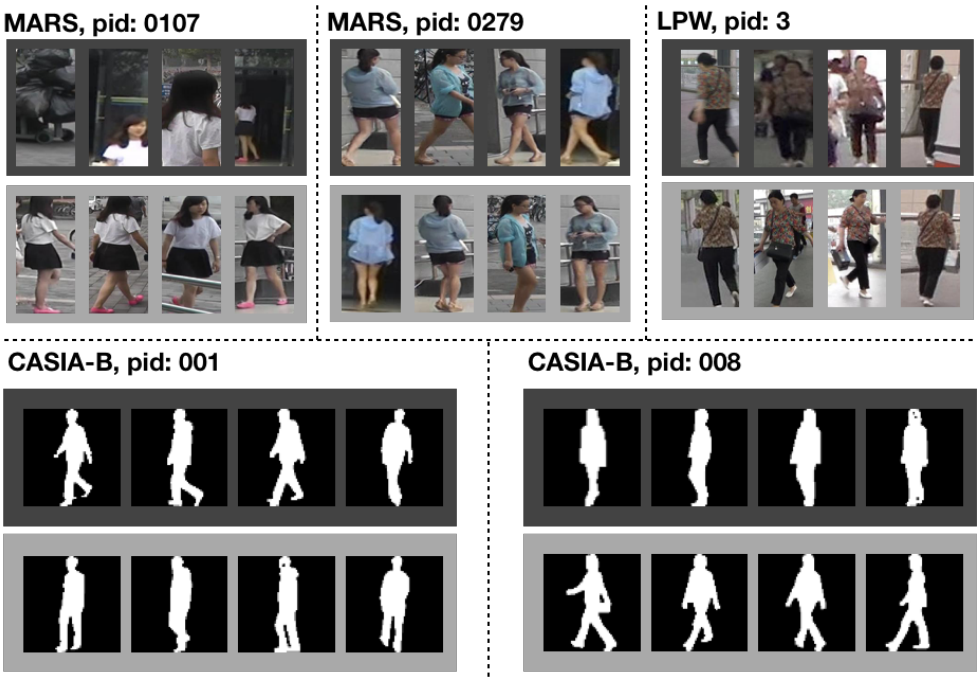


Figure 2.4: Visualization of statistical centroids. The images located in the same rectangular area are the nearest neighbor images of the same SC supports in embedding feature space.

pulls noisy images closer to one barycenter. The centroids from LPW example seem to capture the illumination and resolution, but not as obvious as other datasets.

For the matching pattern from WD, we show a example of temporal alignment between person tracklets in Fig. 2.5. Three person tracklets are picked from the MARS datasets. Among these three tracklets, two of them belong to the same identity, and the rest one is different. We use color bounding boxes to mark the image pairs with highest joint probability, which is automatically assigned by the optimal P in eq. 2.1. From Fig. 2.5, one can see that the proposed method attends to different images of a tracklet while comparing to different candidates. Please note that the images without color bounding boxes are manually selected to indicate the appearance of the whole tracklet.

2.5 Conclusion

In this work, we propose *Statistical Distance Metric Learning* (SDML) framework for image set retrieval task. This framework represents an image set as empirical distribution in embedding feature space, and measures the statistical distance to describe the dissimilarity between two sets. By doing so, SDML can effectively capture the diversity and uncertainty within an image set.Experiment results on two image set retrieval tasks show that SDML outperforms conventional DML methods, and reach competitive/superior performance comparing to previous state-of-the-art approaches.

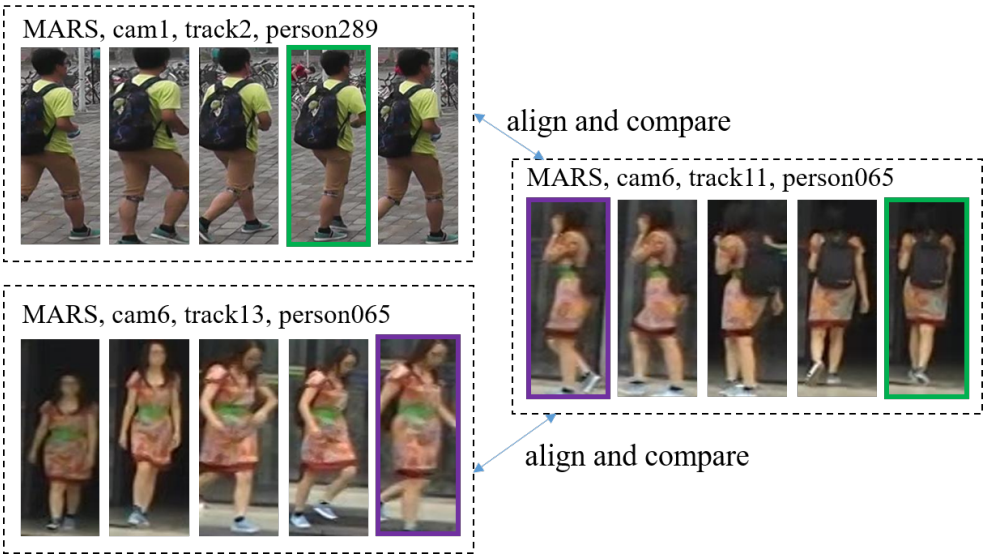


Figure 2.5: Visualization of the alignment automatically discovered by SDML with WD. Image pairs assigned with highest joint probability are marked by bounding boxes in the same color.

Chapter 3

Representation beyond Vector: Subspace

3.1 Introduction

In this chapter, we propose a subspace representation learning (SRL), another framework going beyond vector based representation, for multimedia content. Given a data instance such as an image or a video, the SRL framework represents it as a subspace extracted from its spatial/temporal local feature set. Then, the similarity between two instances is measured by a subspace-to-subspace distance. While many strategies [179] have been proposed to compute the distance between two subspaces, we select the weighted subspace distance (WSD) [79], which considers the importance of each dimension and reflects the original distribution of local features. Similar to SDML, the SRL framework supports efficient end-to-end training by customized training algorithms and a loss function related to distance based classifier [20].

The proposed SRL framework is mainly evaluated by the performance of a few-shot image classification task [30, 77, 84, 114, 133, 138, 149, 165, 170], which aims at obtaining a reliable prediction model that can easily be generalized to unseen concepts. The development of few-shot image classification is inspired by the limitation of Deep neural networks (DNN). Although it has enabled huge advances in many computer vision tasks, such as image classification [122] and object detection [116], the astounding success of DNN is conditioned on the availability of large scale datasets with thorough manual annotation, which is usually too expensive for real-world applications. In contrast, the human visual system is capable of learning a new visual concept with only a few annotated examples. This phenomenon motivates the recent research efforts for few-shot learning.

One of the mainstream approaches to few-shot image classification is based on metric learning [133, 138, 149, 165]. This type of method focuses on learning a good metric function from known concepts with sufficient labeled data, and transferring the learned metric to unseen concepts. Specifically, they exploit a Convolutional Neural Network (CNN) to extract a feature vector for each image, and measure the similarity between two images in hidden feature space based on distance functions, such as Euclidean and cosine distance. While receiving state-of-the-art performance, metric learning based methods are still not able to handle some unseen visual concepts with large intra-class variation and cluttered background [170]. One major reason is that an image-level feature vector ignores the spatial structure and diversity of an image. In the

context of few-shot image classification, this problem becomes more severe since there is not enough supervision signal to guide the network focusing on the correct local regions.

To resolve this issue in few-shot image classification, we apply the proposed SRL framework, representing an image as a subspace extracted from its local CNN feature set. When multiple example images are available for a novel concept, SRL can also be incorporated with meta learning strategies in testing phase. Specifically, we propose two types of template subspace to aggregate the information of multiple meta training instances. The first one is to obtain a class-specific subspace prototype by calculating the average of subspaces, while the second one is to learn a set of task-specific discriminative subspaces. The acquisition of these two types of template subspace is formulated as an optimization problem in a Stiefel manifold.

To evaluate the proposed SRL framework qualitatively and quantitatively, we conduct experiments on three popular benchmarks for few shot image-classification: MiniImageNet, TieredImageNet and Caltech-UCSD Birds-200-2011 (CUB). Experimental results show that SRL achieves competitive/superior performance compared to state-of-the-art few-shot learning approaches. In summary, the main contributions are three fold:

- proposing the idea of subspace representation, another type going beyond vector based representation.
- applying SRL to few-shot classification task and compare two types of template subspace to aggregate K-shot information.
- achieving state-of-the-art performance on three public benchmarks.

SDML and SRL both extend the concept of deep metric learning (DML) from vector space to another domain (sets of distribution and subspace), and captures the local structure of multimedia content. In this chapter, we also analyze the pros and cons of them by conducting qualitative and quantitative studies.

3.2 Related Work

Previous approaches to few-shot image classification can be roughly divided into three categories: (1) Distance metric based methods [133, 138, 149] construct a proper hidden feature space, whose distance metric is used to determine the image-class or image-image similarity. The distance metric can be a non-parametric function [133, 149] or a parametric network module [138]. (2) Optimization based methods [30, 64, 99] aim at learning a good initialization for the model so that it can be quickly fine-tuned to a target task with limited amount of data. (3) Hallucination based methods [44, 69, 86, 155] solve data scarcity by generating more training samples. The generation process is done in either hidden feature space or raw image space. While achieving promising performance, these methods adopt a global feature vector to represent an image.

As an extension of distance metric based methods, some recent works [84, 170] rely on a local feature set representation to preserve the spatial structure of an image, and define their own metric to measure the similarity between two feature sets. Li et al. [84] calculates cosine distance between local feature pairs, and aggregates them by a k-NN classifier. Zhang et al. [170] adopt

earth mover's distance (EMD) to discover an optimal matching between local feature sets. In comparison, our method extracts a subspace representation for each local CNN feature set, and computes the weighted subspace distance between two sets.

The concept of subspace learning has been utilized to solve few-shot image classification [130, 167]. However, these works consider the subspace in a global, image-level feature space. Also, they make use of a subspace projection operation to compute subspace-point distance and measure class-to-image similarity. In comparison, our method represents an image as a subspace in local feature space, and calculates subspace-subspace distance.

3.3 Subspace Representation Learning

3.3.1 Problem formulation for few-shot image classification

The goal of few-shot image classification task is to build a prediction model that can be quickly adapted to unseen classes with limited amount of annotated examples. Most previous works validate their approaches to this task using N-way K-shot classification as testing scenario. Specifically, we are given a support set $S = \{\{(x_{i,j}, y_i)\}_{j=1}^{K}\}_{i=1}^{N}$ and query set $Q = \{(x_q, y_q)\}_{q=1}^{N_q}, y_q \in \{y_i\}_{i=1}^{N}$, where (x, y) is the pair of raw image and class label. The prediction model is trained/adapted based on S, and evaluated on the classification results of Q.

In this chapter, we propose subspace representation learning (SRL) framework to tackle this task. The overall architecture is illustrated in Fig. 3.1. Concretely, the proposed method represents an image as a subspace, which is estimated from the reconstruction of local CNN features of this image. The dis-similarity between two images can be determined by a weighted subspace distance (WSD) between two subspaces. In the rest of this section, we will first introduce the concept of subspace representation. Then, we will elaborate the WSD adopted in SRL framework, and the end-to-end training process in the context of few-shot image classification. Finally, we will describe two types of template subspace, which can effectively summarize information about a specific class from K-shot examples.

3.3.2 Subspace Representation

The proposed method exploits a subspace to represent each training/testing image. Given an image x, we extract the hidden feature map ($h \times w \times d$ tensor) using a backbone CNN with parameter Φ, and collect the d-dimensional local feature vectors at all the spatial locations to form a matrix $H \in \mathbb{R}^{d \times (h \cdot w)}$. Then, the proposed method finds the best-fit s-dimensional subspace $U \in \mathbb{R}^{d \times s}$, which minimizes the reconstruction error with respect to H:

$$\begin{aligned} &\min_{U} ||H - UU^T H||_F \\ &s.t.\ U^T U = I \end{aligned} \tag{3.1}$$

where $|| \cdot ||_F$ is the Forbeneous norm of a matrix. This optimization problem can be solved by singular value decomposition (SVD) of H, and the optimal U is obtained by the top-s right-singular vectors with the largest singular values.

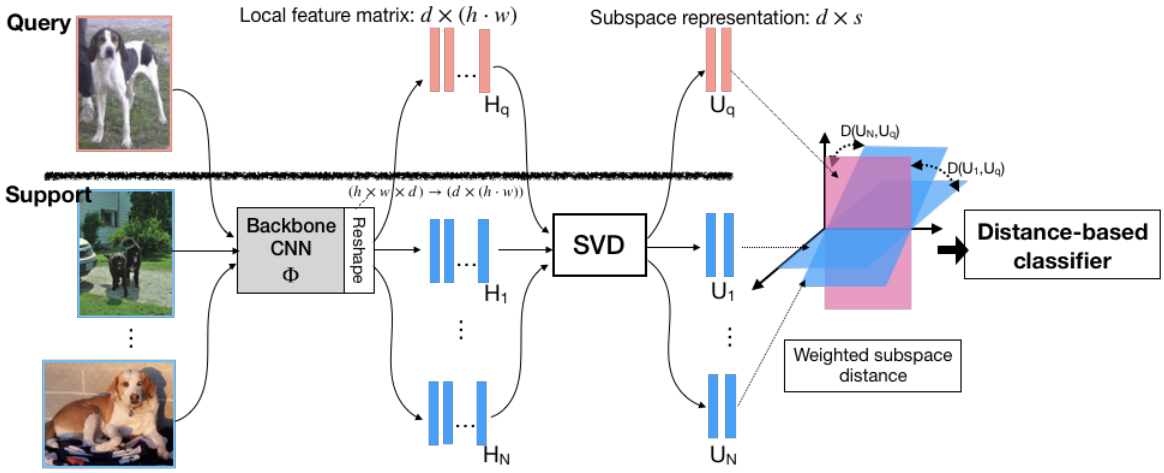


Figure 3.1: The overall architecture of proposed subspace representation learning (SRL) framework. The backbone CNN extracts the local feature map ($h \times w \times d$) from each query and support image. After reshaping the feature map into a matrix $H \in \mathbb{R}^{d \times (h \cdot w)}$, whose columns are the CNN features at every spatial location, we extract the subspace representation by conducting SVD. The similarity between two images is determined by a weighted subspace distance (WSD). The end-to-end training is guided by the loss function of a distance based classifier.

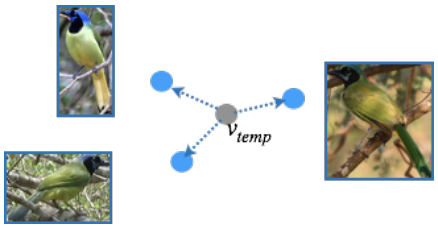
Similar to other works [84, 170] leveraging local CNN features, the SRL framework is able to capture the spatial structure of an image. However, there are two additional reasons why we choose to construct a subspace representation. First, a subspace encourages the preservation of diversity because of the orthonormal constraint in eq. (3.1). Second, using a subspace results in a compact representation for image, since we can set a small s without sacrificing the performance. More analysis of these two properties will be elaborated in the experiment section.

3.3.3 Weighted Subspace Distance

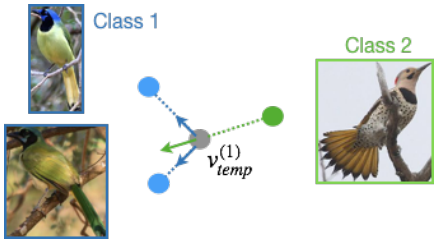
To conduct metric learning in the space of subspace representation, one needs to calculate subspace-to-subspace distance. While several types of distance have been proposed, the SRL framework utilizes the weighted subspace distance (WSD) introduced in [79]: Given two subspaces U_1 and U_2 representing two images x_1 and x_2, WSD is expressed as:

$$D(U_1, U_2) = \sqrt{1 - \sum_{i=1}^{s} \sum_{j=1}^{s} \sqrt{\lambda'_{1,i} \lambda'_{2,j}} (u_{1,i}^T u_{2,j})^2} \tag{3.2}$$
$$\lambda'_{1,i} = \frac{\lambda_{1,i}}{\sum_{l=1}^{s} \lambda_{1,l}}, \ \lambda'_{2,j} = \frac{\lambda_{2,j}}{\sum_{l=1}^{s} \lambda_{2,l}}$$

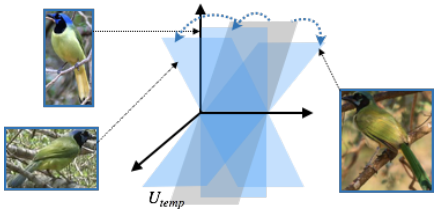
where $u_{1,i(2,j)}$ and $\lambda_{1,i(2,j)}$ are the $i(j)$-th column of $U_{1(2)}$, and the corresponding singular value obtained from SVD operation, respectively. Comparing to other types of subspace distance, WSD considers the relative importance of each basis component in a subspace, so it can better capture the distribution of original local feature set.



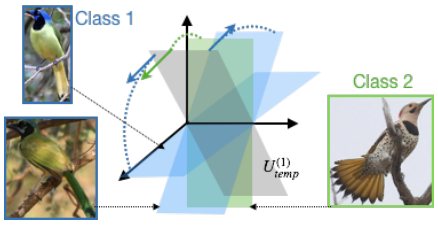
(a) Prototypical network [133]



(b) Distance-based classifier. [20]



(c) Prototypical subspace (PS). (eq. (3.4))



(d) Discriminative subspace (DS). (eq. (3.5))

Figure 3.2: Comparison among different strategies for template vector/subspace (v_{temp}/U_{temp}) extraction from K-shot information. (a) A prototypical network takes mean vector of K-shot vector representations. (b) A distance based classifier obtains the class template vector whose Euclidean/Cosine distances to K-shot examples minimize a cross-entropy loss. (c) A prototypical subspace (PS) is the average subspace of K-shot subspaces. (d) A discriminative subspace (DS) optimizes a distance based classifier with WSD.

3.3.4 End-to-End Training

The training process of the SRL framework follows the episodic learning mechanism [149], which mimics the situation of a testing phase. In each training iteration, we sample a pair of support and query sets (S, Q) from training dataset. Then, we extract the subspace representation for every image in S and Q, and plug the WSD (eq. (3.2)) between the two subspaces into a distance based classifier [20]. Thus, the training objective of SRL can be formulated as:

$$\mathcal{L}_{e2e}(\Phi) = \sum_{q} log \left(\frac{exp(-D(U_{y_q}, U_q))}{\sum_{i=1}^{N} exp(-D(U_{y_i}, U_q))} \right) \tag{3.3}$$

where U_q is the subspace representation of query. U_{y_i} is the subspace representation of the supported image of class y_i if $K = 1$. In the case of $K > 1$, U_{y_i} stands for the template subspace that summarizes the K-shot information from class y_i. The estimation of template subspace will be discussed in Sec. 3.3.5. By minimizing eq. (3.3), the parameter set Φ of backbone CNN can be learned in an end-to-end manner.

3.3.5 Template Subspace for K-shot Learning

To cope with the K-shot learning scenario, the SRL framework computes a template subspace U_{temp} to aggregate the information from K subspaces for each class. In this work, we propose

two types of template subspace to represent a class: A prototypical subspace and a discriminative subspace.

The prototypical subspace (PS) is the "average" of all K subspaces, following the spirit of ProtoNet [133] in vector space. Specifically, given K subspaces $U_1, U_2, ..., U_K$ representing K images of the same class, the prototypical subspace is obtained by the minimizing the summation of the distances between U_{temp} and U_i:

$$\mathcal{L}_{ps}(U_{temp}) = \sum_{j=1}^{K} D(U_{temp}, U_j) \tag{3.4}$$

On the other hand, the discriminative subspace (DS) can be calculated by training a distance-based classifier with respect to a support set S of a N-way, K-shot classification task. Given the set of NK subspaces, $\{\{U_{i,j}\}_{j=1}^{K}\}_{i=1}^{N}$, extracted from S, the set of N template subspaces $\boldsymbol{U_{temp}} = \{U_{temp}^{(i)}\}_{i=1}^{N}$ for N classes would be the minimizer of the following loss function:

$$\mathcal{L}_{ds}(\boldsymbol{U_{temp}}) = \sum_{i=1}^{N}\sum_{j=1}^{K} log\left(\frac{exp(-D(U_{i,j}, U_{temp}^{(i)}))}{\sum_{l=1}^{N} exp(-D(U_{i,j}, U_{temp}^{(l)}))}\right) \tag{3.5}$$

Please note that $\{U_{temp}^{(i)}\}_{i=1}^{N}$ are task-specific, since they are derived jointly from a N-way, K-shot task. In contrast, PS is class-specific because the optimal U_{temp} in eq. (3.4) is independent to other $(N-1)K$ images in S. Fig. 3.2 compares PS and DS, along with their correspondence in vector space.

While minimizing $\mathcal{L}_{ps}$ and $\mathcal{L}_{ds}$, U_{temp} is subject to the orthonormal constraint ($U^T U = I$), which prevents a closed-form solution of both problems. To solve this type of optimization problem with a SGD-like algorithm, we exploit the Cayley transform [101], projecting the gradient to the tangent space of a Stiefel manifold. We describe the update rule of estimating PS as an example. Let $Z = \partial\mathcal{L}_{ps}/\partial U_t$, where U_t is the current estimated U_{temp} in eq. (3.4). The calculation of the next U_{temp} estimation U_{t+1} can be expressed as:

$$\begin{aligned} W &= \hat{W} - \hat{W}^T,\ \hat{W} = ZU_t^T - \frac{1}{2}U_t(U_t^T Z U_t^T) \\ U_{t+1} &= (I - \frac{\alpha}{2}W)^{-1}(I + \frac{\alpha}{2}W)U_t \end{aligned} \tag{3.6}$$

where α is a hyper-parameter analogous to the learning rate in SGD-like algorithms.

3.4 Experiment

3.4.1 Implementation

For a fair comparison, we select the commonly used ResNet-12 as the backbone network of our SRL framework. The softmax layer and the spatial average pooling are removed from the backbone. All the images are resized to 84×84 pixels, and become $5 \times 5 \times 640$ tensors after analyzed by the backbone network. To optimize the parameters of this backbone ResNet-12, we

	MiniImageNet		TieredImageNet	
Methods	5-way, 1-shot	5-way, 5-shot	5-way, 1-shot	5-way, 5-shot
Baseline++ [20]	$53.97 \pm 0.79\%$	$75.90 \pm 0.61\%$	$61.49 \pm 0.51\%$	$82.37 \pm 0.67\%$
ProtoNet* [133]	$63.56 \pm 0.34\%$	$81.08 \pm 0.18\%$	$69.53 \pm 0.36\%$	$84.02 \pm 0.23\%$
MetaOpt-SVM [77]	$62.64 \pm 0.82\%$	$78.63 \pm 0.46\%$	$65.99 \pm 0.72\%$	$81.56 \pm 0.53\%$
MatchNet [149]	$63.08 \pm 0.80\%$	$75.99 \pm 0.60\%$	$68.50 \pm 0.92\%$	$80.60 \pm 0.71\%$
DSN-MR [130]	$64.60 \pm 0.62\%$	$79.51 \pm 0.50\%$	$67.39 \pm 0.82\%$	$82.85 \pm 0.56\%$
FEAT [165]	$66.78 \pm 0.20\%$	$82.05 \pm 0.15\%$	$70.80 \pm 0.23\%$	$84.79 \pm 0.16\%$
Seq-distill [142]	$64.80 \pm 0.60\%$	$82.14 \pm 0.43\%$	$71.52 \pm 0.69\%$	$86.03 \pm 0.49\%$
DN4*† [84]	$63.72 \pm 0.32\%$	$81.54 \pm 0.20\%$	$70.23 \pm 0.33\%$	$84.01 \pm 0.24\%$
DeepEMD† [170]	$65.91 \pm 0.82\%$	$82.43 \pm 0.56\%$	$71.16 \pm 0.87\%$	$86.03 \pm 0.58\%$
SRL, DS (ours)†	$\mathbf{67.00 \pm 0.27\%}$	$\mathbf{82.68 \pm 0.18\%}$	$\mathbf{71.88 \pm 0.32\%}$	$\mathbf{86.24 \pm 0.22\%}$

Table 3.1: Results on MiniImageNet and TieredImageNet. All the methods use ResNet-12 as the backbone network. For SRL, we set subspace basis size, $s = 5$. *: Our re-implementation. †: Using local CNN feature.

conduct a two-step training process. In the first step, we pre-train the parameters by minimizing a cross entropy loss function of a standard classification task using training classes. In the second step, we perform the episodic training mechanism described in Sec. 3.3.4. We adopt SGD optimizer for 10k iterations with an initial learning rate 0.002, which decreases by a factor 0.1 for every 2k iteration. No data augmentation methods are applied during the episodic training.

The PS and DS are initialized by the subspaces extracted from the union of K local CNN feature sets. Then, we update these template subspaces using SGD with Cayley transform for 50 iterations. The learning rate α for PS and DS are 0.1 and 0.01, respectively.

In our experiments, we follow the standard 5-way, 1-shot and 5-shot classification protocols, and sample 5,000 tasks with 15 query images for each target class. The category of each query image is predicted independently (inductive scenario). We report the average accuracy with the 95% confidence interval of all the sampled tasks.

3.4.2 Dataset

To evaluate the efficacy of SRL, we conduct experiments on three benchmark datasets: MiniImageNet [114], TieredImageNet [115] and Caltech-UCSD Birds-200-2011 [150].

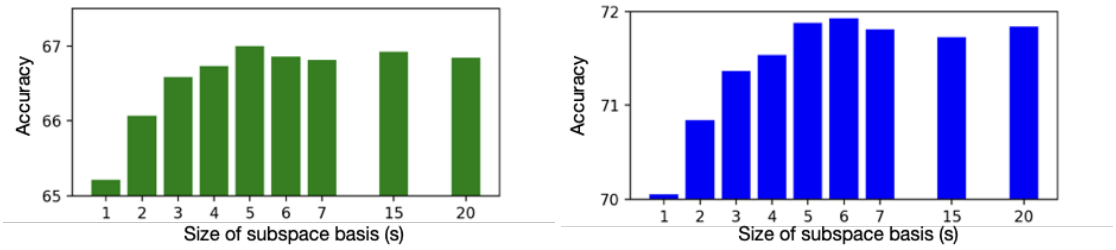
MiniImageNet is a subset of ImageNet [122]. It consists of 100 classes of images, and 600 images per class. The 100 classes are divided into 64, 16, 20 for training, validation and testing sets, respectively.

TieredImageNet contains 608 classes from 34 super-class of ImageNet, and 779,165 images in total. The set of 608 classes is divided into subsets with 351, 97, and 160 classes for model training, validation, and testing, respectively, according to their super-class. This arrangement increases the domain gap between training and evaluation phase.

Caltech-UCSD Birds-200-2011 (CUB) was designed for fine grained image recognition. It contains 200 classes of bird images, 11,788 images in total. Following the setup in previous

Methods	5-way, 1-shot	5-way, 5-shot
ProtoNet*	72.45±0.34%	85.94±0.23%
MatchNet	71.87±0.85%	85.08±0.57%
Baseline++	69.55±0.89%	85.17±0.50%
DN4*†	72.30±0.32%	85.23±0.23%
DeepEMD†	**75.65±0.83%**	88.69±0.50%
SRL, DS (ours)†	75.32±0.27%	**88.81±0.21%**

Table 3.2: Results on CUB. All the methods use ResNet-12 as the backbone network. For SRL, we set subspace basis size, $s = 5$. *: My re-implementation. †: Using local CNN feature.



(a) Results on 5-way, 1-shot task of MiniImageNet

(b) Results on 5-way, 1-shot task of TieredImageNet

Figure 3.3: Sensitivity analysis with respect to the size of subspace basis. The results show that the accuracy saturates around $s = 6$.

works [165, 170], we split the 200 classes into 100, 50, 50, classes for model training, validation and testing, respectively. Comparing to other two aforementioned datasets, CUB is challenging because of the subtle difference among bird types.

3.4.3 Analysis for SRL Design

To better understand the property of SRL and validate our design choices, we conduct two quantitative studies: (1) sensitivity analysis for basis size of subspace representation (2) validating the choice of WSD.

In the first study, we adjust subspace basis size s, which is also the number of columns of U in eq. (3.1), and report the performance of 5-way, 1-shot classification task in MiniImageNet and TieredImageNet. From the results shown in Fig 3.3, we find that the performance gets better when s becomes larger, but saturates around $s = 6$. This observation is expected since the later included subspace basis components are with smaller singular values, and less important according to the definition of WSD. The results also suggest that subspace representation with basis size $s = 6$ is enough to preserve essential information for few-shot classification task.

In the second study, we compare the adopted WSD with another commonly used subspace distance measure, projection F-norm [28, 130]:

$$D_p(U_1, U_2) = ||U_1U_1^T - U_2U_2^T||_F^2 = 2s - 2||U_1^TU_2||_F^2 \tag{3.7}$$

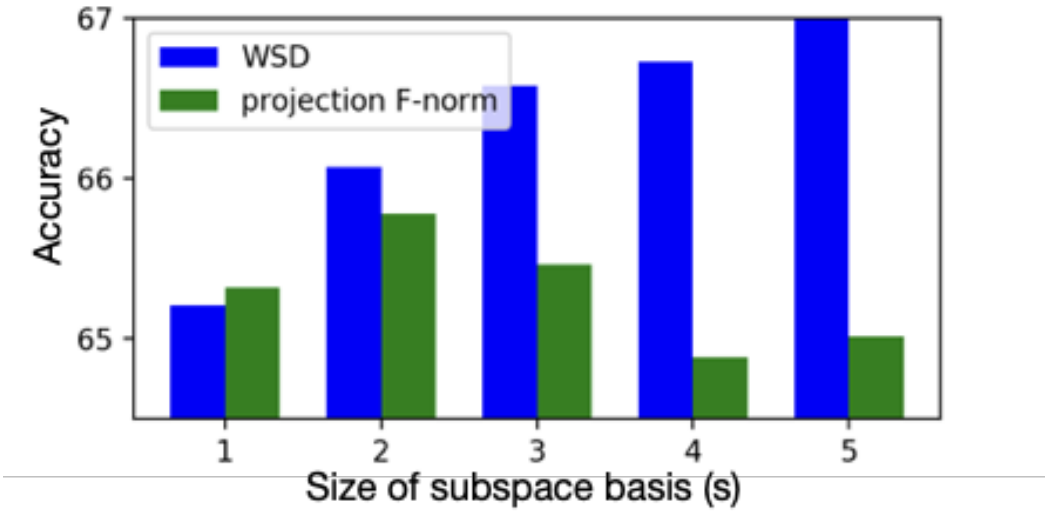


Figure 3.4: Comparison between SRL implementations with two types of subspace distance: WSD (eq. (3.2)) and projection F-norm (eq. (3.7))

	MiniImageNet	TieredImageNet
SRL, PS	$82.14 \pm 0.18\%$	$85.86 \pm 0.23\%$
SRL, DS	$\mathbf{82.68 \pm 0.18\%}$	$\mathbf{86.24 \pm 0.22\%}$
Baseline (Union)	$80.93 \pm 0.20\%$	$83.98 \pm 0.23\%$
Baseline (NN)	$80.56 \pm 0.23\%$	$83.11 \pm 0.26\%$

Table 3.3: Comparison among K-shot aggregation methods on 5-way, 5-shot task of MiniImageNet and TieredImageNet.

Specifically, we replace WSD with $D_p(U_1, U_2)$ in SRL, and follow the same training procedure to optimize the backbone CNN. The performance of SRL with these two subspace distance functions are compared on the 5-way, 1-shot task of MiniImageNet. From the results illustrated in Fig. 3.4, one can observe that the performance of SRL with projection F-norm (eq. (3.7)) drops while s increasing, and performs worse than SRL with WSD in general. The potential reason is that WSD re-weights the importance of each basis component, reflecting the distribution of local CNN features.

3.4.4 Analysis for Template Subspace

In this experiment, we compare two types of template subspace, PS and DS, along with two other naive methods, baseline (union) and baseline (NN), in K-shot learning scenario. Baseline (union) extracts the subspace from the union of all the local CNN features from K-shot images. It is also adopted as the initialization step of PS and DS. Baseline (NN) computes the subspace-subspace distance from query image to the nearest neighbor support image in terms of WSD. All the methods are evaluated on the 5-way, 5-shot classification task of MiniImageNet and

TieredImageNet.

From the results illustrated in Table 3.3, one can see that both PS and DS can outperform two naive baselines, and DS receives the best performance. A possible explanation is that DS is optimized based on task-specific information, while PS is only conditioned on the intra-class information. Thus, when two confusing unseen concepts appear in the same sampled task, DS has a better chance to distinguish them.

We also conduct a follow-up study trying to combine PS and DS. To do so, we derive the template subspace U_{temp} by optimizing the weighted summation of eq. (3.4) and eq. (3.5):

$$U_{temp} = \arg\min_{U} L_{disc}(U) + \alpha L_{proto}(U) \tag{3.8}$$

where α is the hyper-parameter controlling the balance between PS and DS objectives. However, the result shows that the best accuracy of 5-way, 5-shot task on MiniImageNet only increases 0.1% from SRL with DS. Therefore, while compared with previous state-of-the-art, we choose DS as the template subspace for class-specific representation.

3.4.5 Comparison with State-of-the-art

We compare the performance of our SRL framework with two approaches using local feature set, DeepEMD [170] and DN4 [84], as well as other previous state-of-the-art methods. The experiment results are summarized in Table 3.1 and 3.2. From this set of results, we have the following observations. First, ProtoNet from our implementation receives competitive performance on three datasets, serving as a strong baseline. Second, for 1-shot, 5-way task, the proposed SRL performs the best on MiniImageNet and TieredImageNet, and is only slightly worse than DeepEMD on CUB dataset. Third, SRL outperforms all the other methods on the 5-way, 5-shot task of all three datasets.

3.4.6 Visualization

To understand the underline information captured by subspace representation, we visualize the subspace basis components extracted from images of MiniImageNet and CUB datasets. Specifically, we calculate the cosine similarity between each basis component and the local CNN feature of each 5×5 regions. Fig. 3.5 shows the visualization results of the first two basis components. The brightness of each spatial region is proportional to the cosine similarity between the components and the local CNN feature. From the results, we can see that the first component contains shared information among all the features, while the second focuses on some specific local regions. These local regions reflect the characteristics of the corresponding class, which would be useful for classification task. However, observing the third and fourth example images from MiniImageNet, we find that their second basis components are highly correlated to the background regions It indicates that the proposed SRL sometimes suffers from some dataset bias, and fails to represent the object.

3.5 Conclusion

In this chapter, we propose a subspace representation learning (SRL) framework. It represents an image as a subspace in local CNN feature space, and compares two images by calculating a weighted subspace distance (WSD). It successfully extends the concept of deep metric learning (DML) from vector to subspace, and serves as a general tool for modeling the local structure of an image. It also can be easily applied to end-to-end learning network architectures. We investigate the applicability of SRL on few-shot image classification task. To leverage the situation when K-shot information is available, we propose two types of class template representation for SRL: a prototypical subspace (PS) and a discriminative subspace (DS). The estimation of PS and DS can be formulated as an optimization problem in a Stiefel manifold. The experiment results on three public benchmark datasets show that our SRL framework achieves state-of-the-art performance.

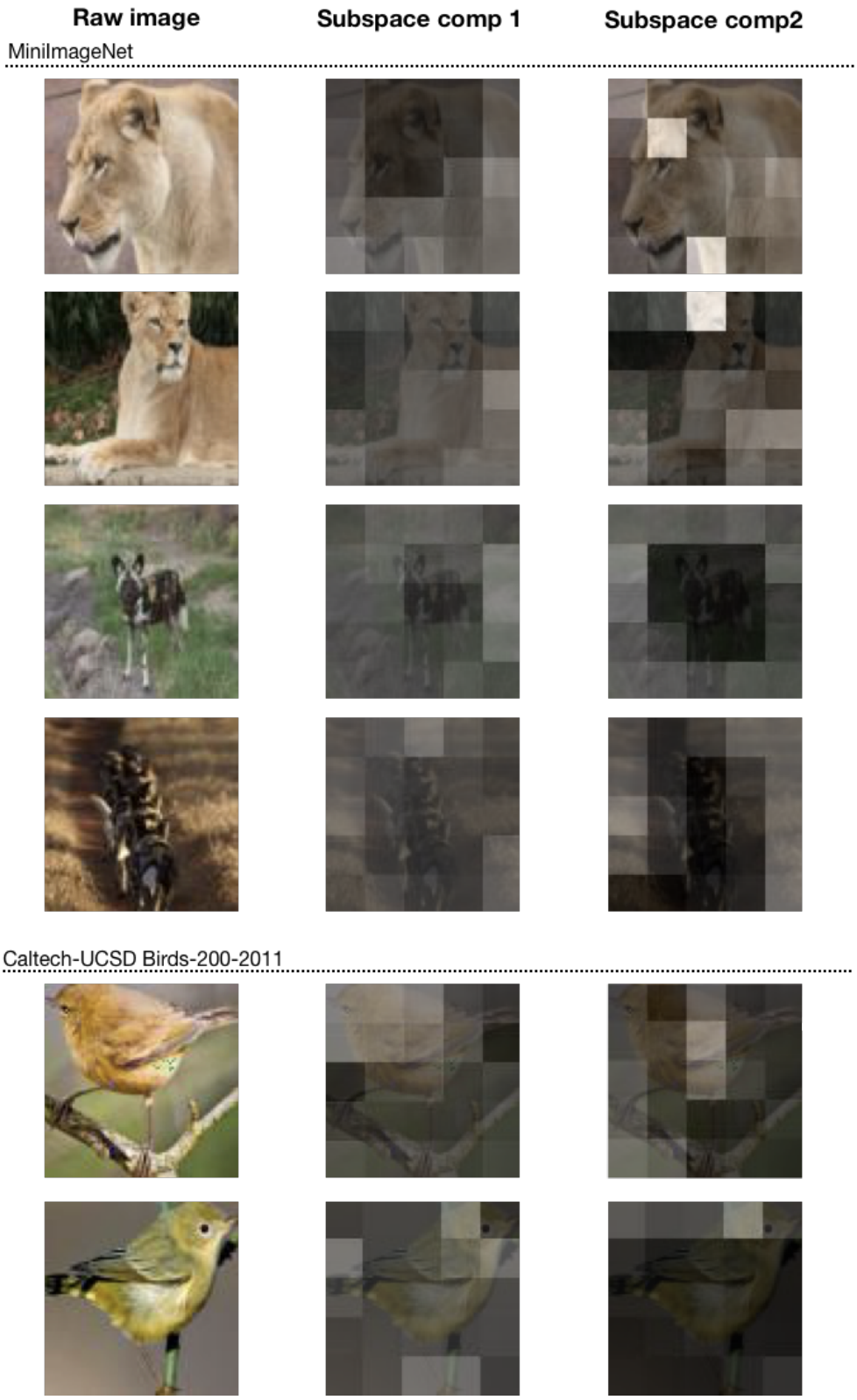


Figure 3.5: Visualization of subspace representation. Raw images are from MiniImageNet and CUB dataset. Brighter regions indicate higher cosine similarity between subspace basis component and the local CNN feature.

Chapter 4

Style and Content Disentanglement: p-Norm Regression in Hidden Space

4.1 Introduction

The purpose of style and content disentanglement is to interpret and manipulate the hidden factors in multimedia content. A variety of applications, such as artistic style transfer and image generation with attributes, have been explored in the previous works. In this chapter, we mainly focus on the investigation of keypoint guided image generation task, which considers the appearance deformation from a source image to target image according to the keypoint pose guidance. Although Generative Adversarial Network (GAN) [40, 97] allows computers to generate photo-realistic images, it is still difficult to capture the deformation from source pose to target pose. This task has attracted the attention of researchers because it provides benefits to multiple applications, such as video synthesis [88] and data augmentation for person re-identification [111].

Several methods have been proposed to resolve the pose guided person image generation task [91, 118, 129, 141, 178]. One type of approaches [141, 178] utilizes attention mechanisms to model the pose-appearance relation. Another type of works [118, 129] relies on the deformation of hidden appearance feature map according to affine transformation. While receiving promising performance, these methods usually lack flexibility comparing to real-world settings. Specifically, they are developed based on the following two assumptions: (1) the availability of identity information of person images, and (2) the generation process is always conditioned on a single source image. However, the first assumption is invalid if the human annotation resource is constrained, while the second becomes a limitation if one can collect multiple images of the same person during the inference phase. Although methods for unsupervised training [136] and multi-shot generation [76] have been investigated to resolve these two issues, respectively, they are still designed for a specific training/testing scenario.

In this chapter, the goal is to develop a simple yet effective, flexible approach that is suitable for different situations: with/without identity information in training, single/multi-shot information in inference. Hence, we propose a p-Norm regression (pNR) module, which models the relation among input appearance and pose feature matrices H, P, and a pose invariant feature set F for each identity as a simple matrix operation in hidden space: $H \approx PF$. Based on this design,

pNR module estimates F by solving a regression problem, and uses the optimal F and the target pose feature P_t to reconstruct the target appearance feature matrix H_t, which becomes the input of an image generator. Then, comparing the generated image with ground truth target image, one can train the appearance/pose feature extractors, and the image generator in an end-to-end manner.

The proposed pNR module is also applicable to unsupervised training and multi-shot inference. In unsupervised training scenario, we use the pNR module to reconstruct the input appearance H from partial observation of H, following the spirit of denoising auto-encoder [148]. In multi-shot generation, we exploit multi-shot information to estimate pose-invariant feature matrix F by constructing a larger regression problem. These two strategies make our overall framework more flexible. Also, the proposed pNR module can be integrated with any pose/appearance feature extractor based on CNN, and contains no additional trainable parameters.

The main contributions in this chapter are two fold: (1) proposing p-Norm regression (pNR) module, which estimates pose-invariant feature and predicts the target appearance feature by solving a regression problem in hidden space. (2) demonstrating the applicability and efficacy of pNR module for pose guided person image generation task in supervised, unsupervised and multi-shot scenarios.

We propose to validate that the proposed pNR module applies to different keypoint guided generation tasks. This method will also be integrated with the work about learning from synthetic data.

4.2 Related Work

Pose guided person image generation has been extensively studied recently [118, 129, 141, 178]. A major stream of works utilizes attention mechanism to model the interaction between appearance and pose information. Zhu et. al. [178] used a sequence of attention modules to conduct pose transfer progressively. Tan et. al. [141] further improved the model capacity by introducing bidirectional appearance-pose attention. Another type of works [118, 129] relies on deformation of hidden appearance feature map according to an affine transformation, which can be non-parametric [129] or parametric [118]. Compared to previous works, the proposed approach focuses on estimating a pose-invariant feature matrix, which can be easily adopted to a wide range of scenarios.

The proposed pNR module is inspired by the techniques of differentiable optimization as layers [1, 4, 11]. These methods solve a optimization problem in the hidden space of a deep network architecture, and use the optimal solution as the input of the later part of forward process. The differentiability of the solution enables the end-to-end training of the whole network. The pNR module is a special case of these general approaches, but the solver and back-propagation of pNR module requires specific development.

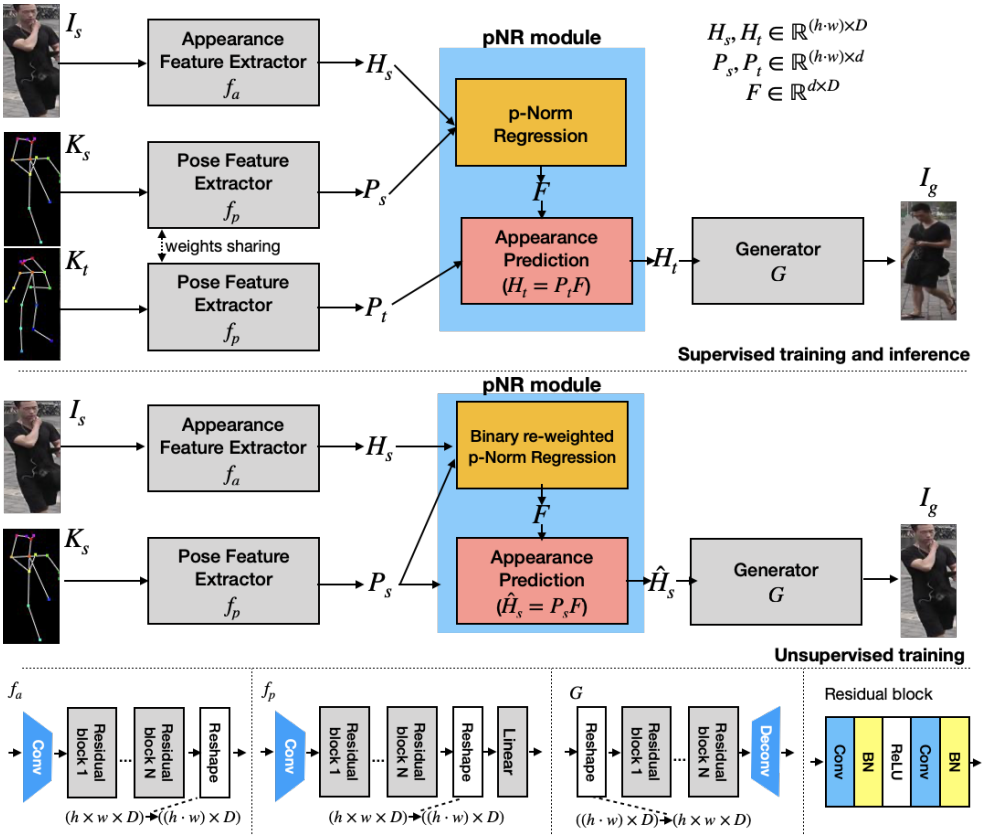


Figure 4.1: Overall architecture of our approach to pose guided person image generation. Our pNR module estimates a pose-invariant feature F in hidden space, and exploits it to predict target appearance.

4.3 Hidden p-Norm Regression

The main purpose of the proposed method is to disentangle the pose information from a raw image by analyzing the given pairs of a person image and a key-point based representation of human pose, (I, K). Following previous works, K is a keypoint heatmap of the 18 joints extracted from I using Human Pose Estimator (HPE) [17]. Specifically, here we focus on the pose guided image generation task, which aims at producing an image I_g given a 3-tuple of source image, source pose, and target pose (I_s, K_s, K_t). The generated I_g should reflect the pose described by K_t, and preserve the identity information of I_s, simultaneously. The availability of the identity information is not always the same. For example, when the human annotation resource is constrained, one may have no or little identity information in training phase. Also, one may want to collect more information for a particular identity during inference phase, so the generation process would be conditioned on multiple pairs of (I_s, K_s). Hence, the ultimate goal is a effective and flexible approach, that is suitable for supervised/unsupervised training, and single/multi shot generation. To achieve this goal, we propose a novel p-norm regression (pNR) module, along with a end-to-end learning framework.

4.3.1 Overall system architecture

The system consists of the proposed pNR module, and three other components: pose and appearance feature extractors f_a, f_p, and an image generator G. In the typical single-shot inference scenario, we are given a 3-tuple (I_s, K_s, K_t). The system first extracts the appearance feature matrix $H_s \in \mathbb{R}^{(h \cdot w) \times D}$ and pose feature matrices $P_s, P_t \in \mathbb{R}^{(h \cdot w) \times d}$ using appearance and pose feature extractors: $H_s = f_a(I_s), P_s = f_p(K_s), P_t = f_p(K_t)$. Each row of $H(P)$ encodes the appearance(pose) characteristics at one of the $h \cdot w$ local regions in the raw image. Then, the proposed pNR module estimates a pose-invariant feature $F \in \mathbb{R}^{d \times D}$ from (H_s, P_s), and produces the target appearance feature matrix H_t. Finally, the generator G takes H_t as input and produces the output image I_g. In unsupervised scenario, identity information is unknown, so we can only exploit pairs of (I_s, K_s) for training. Thus, we solve a binary re-weighted version of p-norm regression, and leverage the reconstruction of I_s as the supervision signal. The overall architectures for all situations are illustrated in Fig. 4.1.

In the remaining of this section, we introduce the details of the proposed pNR module, and the strategies for unsupervised training and multi-shot generation. Then, we elaborate the loss function for training.

4.3.2 p-Norm regression (pNR) module

Given appearance and pose feature matrices H, P in hidden space, we assume that there exists a pose-invariant feature F, and three of them follows the simple relation: $H \approx PF$. The motivation behind this assumption is that the appearance features at every spatial location share some common characteristics, which can be expressed by a set of d feature vectors ($d \ll h \cdot w$). Hence, each appearance feature (row of H) would be a linear combination of rows of F, and the combination weights are encoded in P, Based on this motivation, the proposed p-norm regression (pNR) module contains two steps: estimation of F and prediction of H_t.

In the first step, pNR module estimates F from source appearance and pose information, H_s, P_s. Specifically, the optimal F is the solution of the following p-norm regression problem:

$$F = \arg\min_{F'} ||H_s - P_s F'||_p \tag{4.1}$$

In this work, we investigate two cases: $p = 1$ and $p = 2$. If $p = 2$, eq. (4.1) would be a least square error (LSE) minimization problem, and F can be calculated in closed form:

$$F = (P_s^T P_s)^{-1} P_s^T H_s \tag{4.2}$$

If $p = 1$, eq. (4.1) would become a least absolute deviation (LAD) problem, which is more robust to outliers. However, F has no analytic solution. In this case, we adopt iterative re-weighted least square (IRLS) algorithm [124]. Let F_t be the current estimation of F, the update rule for F_{t+1} in the next iteration can be expressed as:

$$F_{t+1}^{(i)} = (P_s^T W_t^{(i)} P_s)^{-1} P_s^T W_t^{(i)} H_s^{(i)} \tag{4.3}$$

where $F_{t+1}^{(i)}$ and $H_s^{(i)}$ are the i-th column of F_{t+1} and i-th column of H_s, respectively, $i = 1, ..., D$. $W_t^{(i)}$ is a $(h \cdot w) \times (h \cdot w)$ diagonal matrix, whose diagonal elements are from the $(h \cdot w)$-dimensional vector, $1/|H_s^{(i)} - P_s F_t^{(i)}|$. In my implementation, we use the solution of LSE as initial F_0 and execute this update rule for a fix number of iteration. The result of the last iteration would be assigned to F. Given F, the second step of pNR module predicts target appearance based on the same assumption, $H_t = P_t F$.

Since pNR module is treated as an intermediate layer of the whole network architecture, we need to differentiate through it in order to train the parameters in f_a, f_p and G with SGD-like algorithm. In the case of LSE ($p = 2$), the derivative of F in eq. (4.2) can be obtained easily. However, when $p = 1$, the calculation of F is an iterative process, so the precise derivative is difficult to compute. Considering eq. (4.3) in the last iteration of IRLS update, we calculate $\partial F^{(i)}/\partial H_s^{(i)}, \partial F^{(i)}/\partial P_s$ only, and ignore the derivative with respective to the recursive term, $\partial F^{(i)}/\partial W_t^{(i)}$, during the backward propagation. Although this calculation is an approximation, it still aims to preserve the robustness of LAD to outliers, and receives good empirical performance.

4.3.3 Unsupervised training and multi-shot generation

The proposed pNR module can be easily adapted to unsupervised training and multi-shot generation scenarios.

In unsupervised training, pNR module estimates F by solving a binary re-weighted p-norm regression:

$$F = \arg\min_{F'} \sum_{j=1}^{h \cdot w} v_i ||H_s^j - P_s^j F'||_p \tag{4.4}$$

where $v_j \in \{0, 1\}$ is a binary random variable, which blocks out the information from the j-th row in H_s (H_s^j) when $v_j = 0$. In our case, we set $p(v_i = 0) = p(v_i = 1) = 0.5$. Then, we predict the source appearance feature $\hat{H}_s = P_s F$ in the second step, and use $\hat{H}_s$ to reconstruct the input source image.

In multi-shot generation, we aggregate the information from M pairs of source image and pose map (I_s, K_s). Specifically, we concatenate all the appearance and pose feature matrices, and construct a larger p-norm regression problem. It can still be expressed by eq. (4.1), but $H_s \in \mathbb{R}^{(M \cdot h \cdot w) \times D}$, $P_s \in \mathbb{R}^{(M \cdot h \cdot w) \times d}$. Please note that M can be different in training and inference phase, and can also be a varied number.

4.3.4 Loss functions

Following the design of previous works [141, 178], the training objective of our system framework contains four components: L1 loss, Perceptual loss, GAN_I loss, GAN_K loss.

L1 loss computes the pixel-wise L1 distance between generated image and target image: $\mathcal{L}_{L1} = ||I_t - I_g||_1$. Perceptual loss compares two images in the space of pretrained features: $\mathcal{L}_{per} = ||\Phi_\rho(I_t) - \Phi_\rho(I_g)||_1$, where Φ is a VGG19 [131] network pretrained on ImageNet [122], and ρ is the index of hidden layers ($\rho = Conv1\_2$ in our case). On the other hand, the purpose of GAN_I and GAN_K loss is to align the output of generator to two probability distributions: $p(I_t|I_s)$ and $p(I_t|K_t)$, respectively. To do so, we measure the two types of distribution discrepancy by two discriminators, D_I and D_K, respectively. The former distinguishes generated images from target images conditioned on source image I_s, while the later does so conditioned on the pose map K_t. Thus, two loss functions are formulated as:

$$\begin{aligned} \mathcal{L}_{GAN_I} =& E[\log(D_I(I_t, I_s))] + E[\log(1 - D_I(I_g, I_s))] \\ \mathcal{L}_{GAN_K} =& E[\log(D_K(I_t, K_t))] + E[\log(1 - D_K(I_g, K_t))] \end{aligned} \tag{4.5}$$

where the expectation is computed over the distribution of I_s, I_t pairs. The overall training objective is the weighted combination of the four components, and the training process can be expressed as:

$$\min_{f_a, f_p, G} \max_{D_I, D_K} \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{per} + \lambda_3 \mathcal{L}_{GAN_I} + \lambda_4 \mathcal{L}_{GAN_K} \tag{4.6}$$

In unsupervised training scenario, source and target images are the same, so we replace (I_t, K_t) to (I_s, K_s) in all the loss functions, and disable $\mathcal{L}_{GAN_I}$ by setting $\lambda_3 = 0$.

4.4 Experiment

4.4.1 Implementation

Pose and appearance feature extractors, f_p and f_a both consist of two downsampling CNN layers, a sequence of N residual blocks proposed in [66] and a reshape operator, while f_p has another linear layer to reduce the dimension to d. Image generator G contains N residual blocks followed by two unsampling deconv layers. We set $N = 4$ and $N = 2$ in supervised and unsupervised training, respectively. For the discriminators D_I and D_K, we exploit the same implementation as in [178]. For pNR module, we set $d = 20, D = 256$.

In the training phase, Adam [71] optimizer with learning rate of 0.002, $\beta_1 = 0.5$, $\beta_2 = 0.999$ is adopted. For the hyper-parameters of loss function, we set $\lambda_1 = \lambda_2 = 5, \lambda_3 = \lambda_4 = 10$ for supervised training, and change λ_3 to 0 for unsupervised training.

Figure 4.2: Qualitative comparison between pNR module and other methods. *: Unsupervised training.

	IS	SSIM	mask-IS	mask-SSIM
PG^2 [91]	3.460	0.253	3.435	0.792
Def-GAN [129]	3.185	0.290	3.502	0.805
PATN [178]	3.323	0.311	3.773	0.811
XingGAN [141]	3.506	**0.313**	**3.872**	**0.816**
pNR (LSE)	3.435	0.298	3.741	0.802
pNR (LAD)	**3.631**	0.305	3.796	0.807
SPT* [136]	3.449	0.203	**3.680**	0.758
pNR* (LSE)	**3.688**	0.241	3.501	0.783
pNR* (LAD)	3.681	**0.248**	3.610	**0.789**

Table 4.1: Quantitative results on Market-1501. All the metrics are the higher the better. *: Unsupervised training.

4.4.2 Dataset and evaluation protocols

We evaluate the proposed pNR module on two tasks: pose guided person image generation, and landmark guided facial expression generation.

The experiments of pose guided person image generation are conducted on the challenging Market-1501 [173] dataset, which was designed for person re-identification. Performing pose guided image generation on this dataset is challenging because of its low resolution (128×64 in pixel), and high diversity in pose, background and illumination. Following previous works, we detect the keypoint-based pose representation by HPE, and remove images in which no human body can be detected. Consequently, the training and single-shot testing sets consists of 263,632 and 12,000 pairs of images with the same identity. Sets of identities for training and testing are mutually exclusive. We also use Market-1501 to perform generation with multi-shot source images. For this purpose, we keep those identities in single-shot testing set with 6 or more images, and sample 12,000 tuples with 5 source images and 1 target image for testing. In this set of the experiments, we adopt Structure Similarity (SSIM) [157] and Inception Score (IS) [123] as the evaluation metrics. SSIM measures correctness of pose transfer by comparing generated and ground truth images, while IS uses a pretrained image classifier to assess the image quality. The masked version of both metrics are also utilized to reduce the distraction from irrelevant background regions.

The experiments of landmark guided facial expression generation are conducted on Radboud Faces dataset [75]. It contains about 8000 images from 5 camera views and 67 subjects. They were asked to perform facial expressions according to 8 different emotion: anger, fear, disgust, sadness, happiness, surprise, neutral and contempt. Following previous work [139], we re-scale all the images to $256 \times 256 \times 3$, and remove those images in which the human face can't be detected by OpenFace [5]. It results in a subset with 7035 images in total. Then, this subset is further divided into a training (5628 images) and testing set (1407 images). In this set of experiments, we focus the unsupervised training scenario, which assumes that no identity annotation is available. Also, we adopt PSNR and SSIM to evaluate the generation results quantitatively.

	IS	SSIM	mask-IS	mask-SSIM
[76], M=1	3.251	0.270	3.614	0.771
[76], M=3	3.442	0.291	3.739	0.783
[76], M=5	3.444	0.306	**3.814**	0.788
pNR, M=1	3.631	0.305	3.796	0.807
pNR, M=3	3.642	0.311	3.804	0.812
pNR, M=5	3.629	**0.313**	3.804	**0.818**
pNR*, M=1	**3.684**	0.248	3.610	0.789
pNR*, M=3	3.640	0.254	3.616	0.801
pNR*, M=5	3.662	0.259	3.614	0.805

Table 4.2: Results of multi-shot generation. All the metrics are the higher the better. *: Unsupervised training

4.4.3 Experiment Results – Pose Guided Person Image Generation

We compare the proposed method with some previous state-of-the-art, including PG^2 [91], Def-GAN [129], PATN [178], XingGAN [141], and SPT [136] (unsupervised). From the results shown in Table 4.1, we make the following observations. First, pNR with LAD ($p = 1$) performs better than that with LSE ($p = 2$) in both supervised and unsupervised training scenarios. Second, in supervised training, pNR yields competitive performance compared with most recent state-of-the-art, PATN and XingGAN. Third, in unsupervised training, pNR* outperforms the previous work, SPT, by a large margin on every metrics except mask-IS.

We present a qualitative study for both supervised and unsupervised training scenarios in Fig. 4.2. Compared with other previous works, the proposed method is more capable of capturing pose and appearance information. We also list the results from pNR* (unsupervised training), which are still in good quality but contain more artifacts than supervised methods.

In Fig. 4.4, we illustrate more results of a qualitative study for unsupervised training scenario, which is better aligned with the assumption of style and content disentanglement. From the results of this study, we can see that the model successfully disentangles some lower level characteristics, such as the color/type of cloths, position of arms and legs, etc. However, some higher level semantic concepts are not preserved by our model. For example, the generated images in the first and third rows don't contain the bags carried by the target persons. In the generated image of the fifth row, one can also see that the upper body and lower body are moving toward different directions. These examples show that the proposed method should be further improved for real world applications.

We also demonstrate that the proposed method can effectively integrate information from multi-shot source images. For model training, we apply LAD ($p = 1$) in pNR module, and exploit single-shot ($M = 1$) training dataset. From the quantitative comparison in Table 4.2, one can see that the pNR module outperforms previous work [76], and yields better SSIM and mask-SSIM with larger number of source images in both supervised and unsupervised training scenarios. Also, the qualitative results in Fig. 4.3 show that pNR module reconstructs more details in generated images when more source images are available.

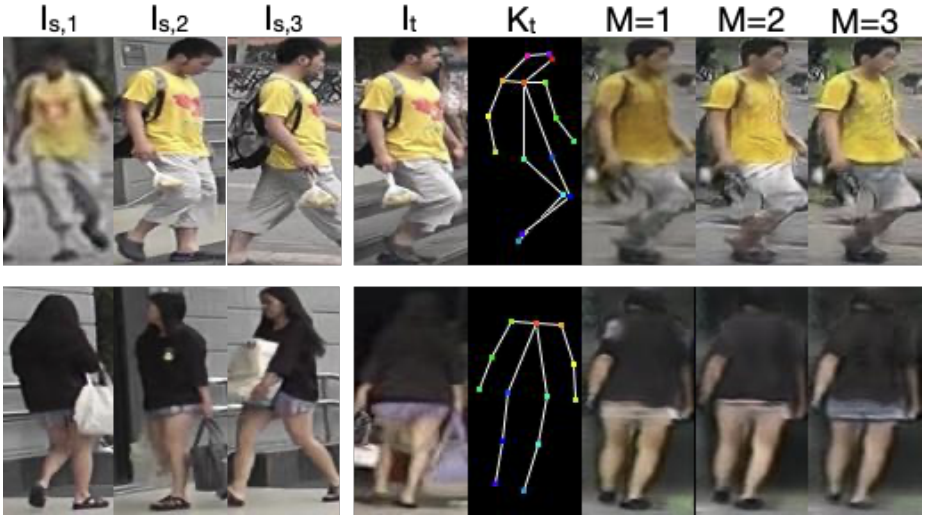


Figure 4.3: Qualitative results of multi-shot generation using pNR module (supervised training, LAD).

4.4.4 Experiment Results: Facial Expression Generation

We also verify the effectiveness of pNR module on another key-point guided image generation task. Specifically, we focus on the face expression generation task with Radboud Faces dataset [75]. It contains about 8000 images from 5 camera views and 67 subjects. They were asked to perform facial expressions according to 8 different emotion: anger, fear, disgust, sadness, happiness, surprise, neutral and contempt, and to show three different gaze directions. Following the previous work [140], we randomly select 66% of images as training data, use the rest for evaluation.

For the facial expression generation task, we train the generative model with pNR module in the unsupervised manner (without identity information). The qualitative results are illustrated in Fig. 4.5 and 4.6. From Fig. 4.5, we observe that the proposed pNR module can disentangled keypoint and identity information, and generate unseen facial expression given the clue from keypoint locations. However, from Fig. 4.6, we also notice that the proposed pNR module is sensitive to the correctness of keypoint locations. If the resource of keypoint locations (e.g. a detector) is not accurate, the proposed method fails to recover the correct appearance information.

4.5 Conclusion

In this section, we aim at the disentanglement of hidden factors in multimedia content. In the context of pose/keypoint guided image generation, we propose a novel pNR module. It estimates a pose-invariant feature matrix for each identity and predicts the target appearance feature by solving a p-norm regression problem in hidden space. Integrated with CNN-based pose/appearance feature extractors, pNR module serves as a layer of the whole network architecture and supports end-to-end training. The experiment results demonstrate the efficacy of the pNR module in a supervised and unsupervised training scenario, as well as generating images from multi-shot

Figure 4.4: More qualitative results of pNR module with unsupervised training.

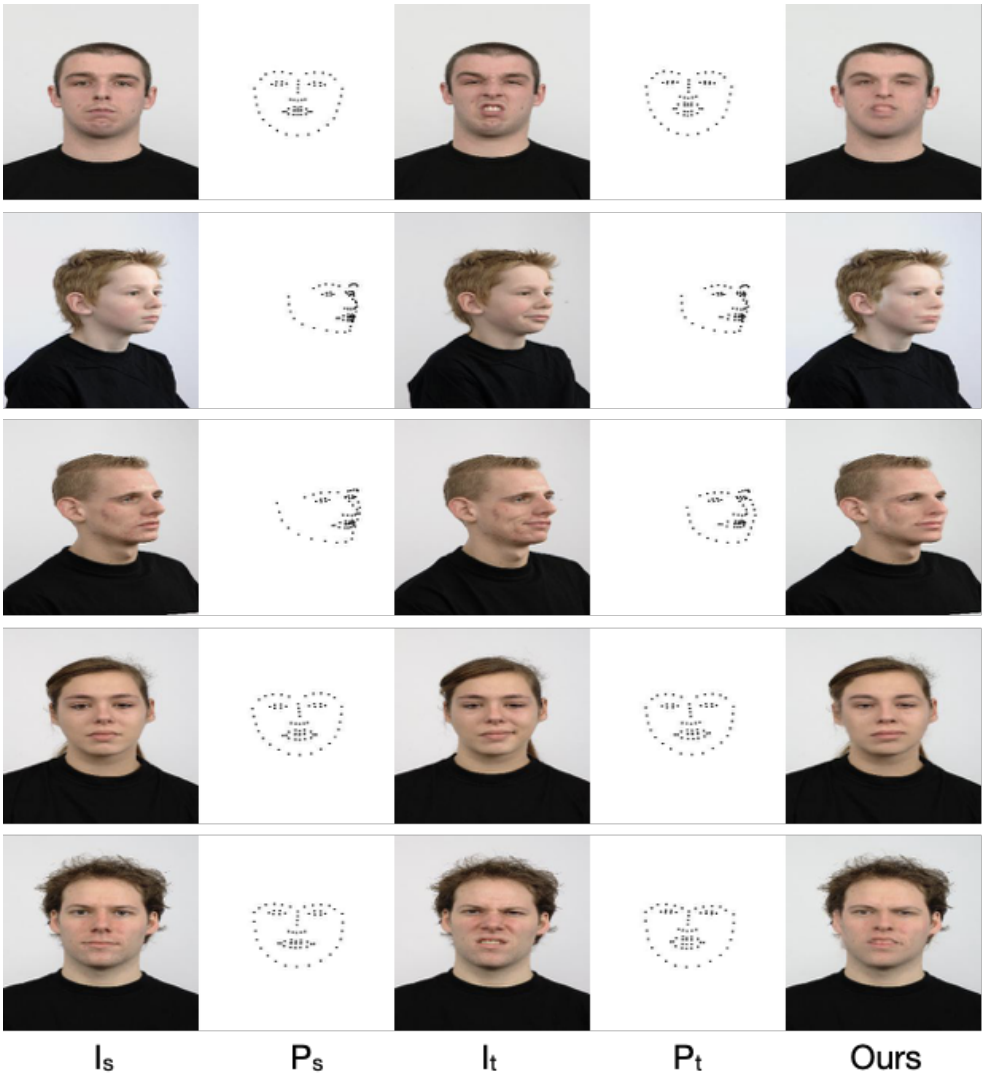


Figure 4.5: Qualitative results of facial expression generation.

person image data.

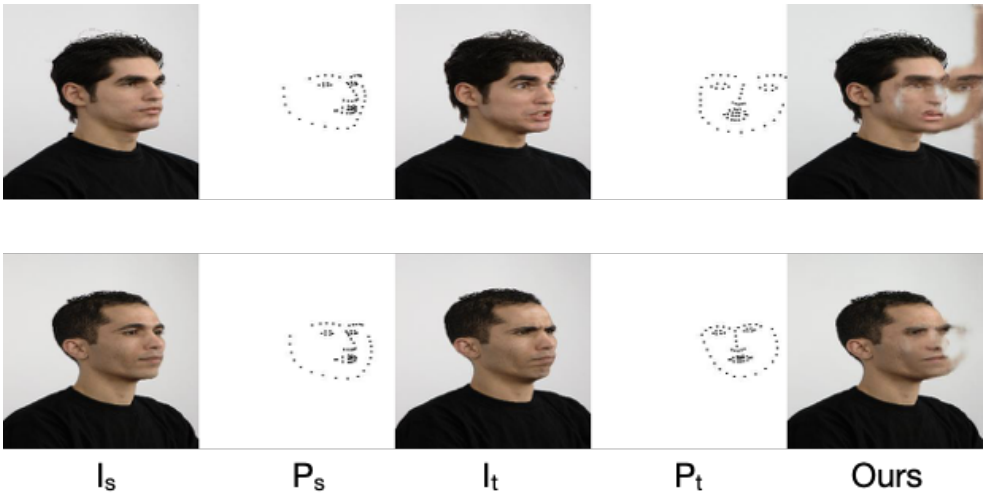


Figure 4.6: Qualitative results of facial expression generation (failed examples)

Chapter 5

Style and Content Disentanglement: Mutual Information Minimization

5.1 Introduction

In this chapter, I approach to style and content disentanglement by explicitly measuring the degree of disentanglement. Concretely speaking, the goal is to obtain a numerical value to judge if style and content factors are separated. This concept has been explored in the works of unsupervised disentangled representation learning [19, 21, 49, 67], which utilize total correlation to evaluate the uncorrelated hidden factors discovered by their algorithms. In some previous works of style and content disentanglement [74], they considered the predictability, the content classification performance given style representations. However, it is not applicable when the content information is not categorical. Thus, I propose to measure the mutual information between style and content during the representation learning process, and demonstrate that it is suitable for applications with non-categorical content information such as speech synthesis.

Mainstream neural network based text-to-speech (TTS) methods [7, 68, 82, 96, 117, 126] are able to produce high quality speech. However, they ignore the other hidden factors within

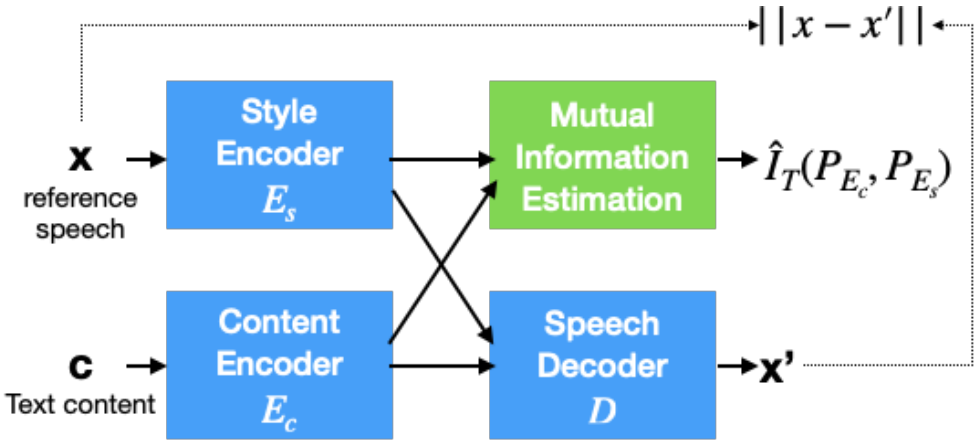


Figure 5.1: The overall architecture of our method for TTS stylization.

speech utterances such as speaker identity, the speaking style, prosody, or the environmental factors. Considering that the conversion from text to speech is actually a one-to-many mapping, some prior works [53, 65, 92, 156] enhanced these neural TTS models by providing an additional reference speech signal to control the style, so that the generated speech has the same style as the reference. In these controllable TTS methods, the reference speech is encoded into an embedding (called a style vector) that is input with the content features to a speech decoder. These controllable TTS methods encode the reference speech into a style vector, which is fed into a speech decoder along with content representation. Most recent works in this direction generate the style vector with the guidance of speaker identity annotation [53, 65, 92, 145], which may be hard to extend to the cases where the speaker information is not available (e.g. a new language, a new environment, or due to privacy reasons). Furthermore, the user information requires additional annotations to use the unlabeled audio data.

To overcome this limitation, Global Style Token (GST) method [156] learns speaker embeddings in an unsupervised manner by jointly training the style encoder network as well as the encoder-decoder part of the TTS model, while minimizing the reconstruction loss. Specifically, this method computes style vectors using a set of trainable vectors called style tokens, which are linearly combined using style coefficients generated from the input reference speech. Style tokens are trainable parameters that are optimized together with the TTS network parameters. To compute the style coefficients, it uses an additional style encoder that is trained jointly with the TTS model. The style coefficients are passed through a Softmax layer (so that they sum to 1) before computing the style vector with them.

Furthermore, for computational efficiency, we use Transformer TTS [82, 146] for the content encoder and decoder. This model uses self-attention [146] and does not have any recurrent connections, which is significantly faster to train compared to LSTM-based models such as Tacotron 2 [126]. During training, text is given as the content input and the corresponding mel-spectrogram is used as reference speech for encoding the style.

However, in the unsupervised training scenario, the target output speech is the same as the reference input speech for style encoding, which causes some of the content information to leak into the style vector. This leaked content can be used by the decoder to reconstruct the speech while ignoring the actual content input. At inference time, when the reference speech has different content from the input text, the decoder expects the content from the style vector and ignores some part of the content text. We refer to this problem as "content-leakage" which results from having the same style input as the desired output during training. Ideally, the style vector should not be able to reconstruct the content vector, i.e., there should be no information about the content in the style vector. To this end, we bring in the idea of measuring the mutual information between the underline distributions of style and content representations. While receiving the empirical samples of both style and content from hidden distributions, we estimate their mutual information by using Mutual Information Neural Estimation (MINE) in [14]. The MINE algorithm computes a lower bound of the mutual information using a neural network, which is optimized to maximize this lower bound. We alternate between maximizing the lower bound (i.e., estimating the mutual information) and minimizing the estimated mutual information and the reconstruction loss. The maximization problem is solved w.r.t. the MINE network, while the minimization problem is solved w.r.t. the style encoder, the content encoder, and the decoder.

To summarize the contributions in this chapter, we propose to disentangle the style and con-

tent information by minimizing the mutual information (MI) between them. The estimation and minimization steps of MI are jointly formulated as a adversarial training task. In the context of text-to-speech application, the proposed method helps prevent content leakage issue. The qualitative and quantitative evaluation results show that the proposed method outperform state-of-the-art unsupervised controllable TTS methods.

5.2 Related Work

Recent neural TTS methods, such as Tacotron 2 [126], MelNet [145], Deep Voice 3 [109], and TransformerTTS [82], map input text to speech features (e.g. mel-spectrogram) using a content encoder and a speech decoder. To recover the original time domain speech signal from the speech features, one can rely on a conventional vocoder such as Griffin Lim algorithm [42], or a neural network based vocoder, such as WaveNet [104] and WaveGlow [110]. We choose TransformerTTS as our neural TTS backbone because of the substantially reduced training time, and WaveNet [104] as our vocoder.

The concept of style and content disentanglement has been explored in many different areas, such as artistic image [37], face attribute manipulation [74], handwriting [21], text generation [60], and neural TTS [92]. The authors in [92] follow the idea of obtaining the style information as the gram matrix of feature maps to capture the style in synthesized speech. Compared to these methods, our approach disentangles the style and the content by explicitly minimizing the mutual information between their latent representations, not the loss of a discriminator.

Neural controllable TTS models [53, 65, 92, 156] generate speech with the input text content, where the style is given by an input reference speech signal that may not have the same content as the input text. These models analyze the reference speech signal and extract style information using an additional style encoder, which is parallel to the content encoder of a neural TTS system. The authors in [65] incorporate external data to train a discriminative speaker encoder, and transfer the learned encoder to build a multi-speaker TTS system. The authors in [53] adopt a variational autoencoder to model both the observed and the latent style attributes. Global style token (GST) method [156] maintains a set of style embedding vectors, and constrain a style embedding of reference speech to be a linear combination of this style embedding set. A recent work [92] enhances this model by latent attribute reconstruction and GAN training [40]. Most of the these works require style annotation, such as speaker identity and emotion, in the training stage. Compared to these methods, our proposed approach is unsupervised, i.e., it does not require style annotations or speaker embeddings. To the best of our knowledge, the only other neural TTS based unsupervised style and content separation method is by [156], but this suffers from content leakage.

5.3 Proposed Method

The proposed method, shown in Figure 5.1, is based on a controllable TTS architecture. We use a backbone TTS model to pre-train the content encoder, E_C To this backbone TTS model, we add a style encoder, E_S, to extract style vector from the reference speech, and the MI estimator

to measure the mutual information between the style and the content vectors

5.3.1 Content Encoder Pre-training

The first step of MIST is content encoder pre-training, which can be simply treated as a neural TTS training process. It is important to use a single-style dataset in the pre-training process because a multi-style dataset usually has same content spoken in different style (e.g. by different speakers). Given a set of speech and content pairs, $\{(\boldsymbol{x}, \boldsymbol{c})\}$, we jointly train the content encoder, E_C, and speech decoder, D, by minimizing the reconstruction loss,

$$min_{E_C,D}||D(E_C(\boldsymbol{c})) - \boldsymbol{x}||_1, \tag{5.1}$$

where $\|.\|_1$ is the ℓ_1 norm. The trained E_C with frozen weights is used in the second stage of our method, while D is re-initialized with random weights.

5.3.2 Style and content disentanglement

In the second step of our method, we train a speech synthesis model that is capable of disentangling the style from the reference speech and generating speech in this style with the content of the input text. During training, the input content is the same as the content of the reference speech. Using only the reconstruction loss to update E_S, E_C, and D, the model suffers from content leakage because the content information in the output can also be extracted from the reference speech. We disentangle the style and content by minimizing the mutual information (MI) between their hidden representations ($E_S(\boldsymbol{x})$ and $E_C(\boldsymbol{c})$), so that the style does not contain information about the content. However, it is not obvious how to compute and minimize the mutual information between two continuous random vectors. First, we briefly describe a recently proposed method to estimate the mutual information, then we present our novel application to minimize it jointly with the reconstruction loss.

Mutual information neural estimation (MINE)[14]: The mutual information, $\mathcal{I}(\boldsymbol{Y}, \boldsymbol{Z})$, of random variables $\boldsymbol{Y}$ and $\boldsymbol{Z}$ is equivalent to the Kullback–Leibler (KL) divergence [73] between their joint distribution, $P_{\boldsymbol{Y},\boldsymbol{Z}}$, and product of marginals, $P_{\boldsymbol{Y}} * P_{\boldsymbol{Z}}$:

$$\mathcal{I}(\boldsymbol{Y}, \boldsymbol{Z}) = D_{KL}(P_{\boldsymbol{Y},\boldsymbol{Z}}||P_{\boldsymbol{Y}} * P_{\boldsymbol{Z}}). \tag{5.2}$$

Using this fact, MINE[14] method constructs a lower bound of mutual information based on Donsker-Varadhan representation of KL divergence [26]:

$$\mathcal{I}(\boldsymbol{Y}, \boldsymbol{Z}) \geq \hat{\mathcal{I}}_T(\boldsymbol{Y}, \boldsymbol{Z}) = \sup_T E_{P_{\boldsymbol{Y},\boldsymbol{Z}}}[T] - \log(E_{P_{\boldsymbol{Y}} * P_{\boldsymbol{Z}}}[e^T]), \tag{5.3}$$

where T can be any function that makes the two expectations in the above equation finite. The authors in [14] propose to use a deep neural network for T, which allows us to estimate the mutual information between $\boldsymbol{Y}$ and $\boldsymbol{Z}$ by maximizing this lower bound with respect to T through gradient descent.

Style and content separation with MI minimization: We minimize the the reconstruction loss along with the estimated mutual information between the style and the content vectors. Since

the MI is always non-negative, we clip the estimated mutual information to zero if it is negative. The clipped value is not only a better estimate of the mutual information than the non-clipped one (because the true MI is always non-negative), it also avoids minimizing a function that is unbounded from below. In one experiment, we found that by clipping the performance of the speech recognition on the generated data can be improved by approximately 30%. Thus, the overall objective function is a min-max problem where we maximize the lower-bound of MI, $\hat{\mathcal{I}}$, w.r.t. T and minimize the MI and the reconstruction loss w.r.t. D and E_S,

$$\min_{E_S,D} \max_{T} \big\{ ||D(E_C(\boldsymbol{c}), E_S(\boldsymbol{x})) - \boldsymbol{x}||_1 + \lambda * \max(0, \hat{\mathcal{I}}_T(E_C(\boldsymbol{c}), E_S(\boldsymbol{x}))) \big\},$$

where λ is a hyper-parameter that balances the two losses. In our experiments, we set $\lambda = 0.1$ and found the algorithm to be insensitive to different values of λ, as shown later in Section 5.4.1. Similar to common GAN training, we update the speech synthesis model (E_S, D) and the MI estimator function, T, alternatively in each step of the training. Since $E_C(\boldsymbol{c}_i)$ is a sequence of vectors of varying length, we randomly sample one of the content vectors to compute the mutual information. By optimizing (5.4), we can jointly ensure the quality of speech feature reconstruction, and make the information extracted from E_C and E_S independent to each other. We summarize the training method in Algorithm 3.

Algorithm 3 Pseudocode for the proposed MIST training

Input: Pairs of speech and text ($\boldsymbol{x}_i$, $\boldsymbol{c}_i$).
Output: E_C, D, E_S.

$E_C, D \leftarrow \arg\min_{E_C,D} \sum_i ||D(E_C(\boldsymbol{c}_i)) - \boldsymbol{x}_i||_1$
$E_S, D, T \leftarrow$ initialization with random weights
while E_S, D, T not converged **do**
  Sample a mini-batch of $(\boldsymbol{x}_i, \boldsymbol{c}_i), i = 1, 2..., b$.
  $\{\boldsymbol{y}_i\} \leftarrow \{E_C(\boldsymbol{c}_i) | i = 1, 2, ..., b\}$
  $\{\hat{\boldsymbol{y}}_i\} =$ random permutation of $\{\boldsymbol{y}_i\}$
  $\{\boldsymbol{z}_i\} \leftarrow \{E_S(\boldsymbol{x}_i) | i = 1, 2, ..., b\}$
  $\mathcal{L}_{MI} = \frac{1}{b} \sum_{i=1}^{b} T(\boldsymbol{y}_i, \boldsymbol{z}_i) - \log(\frac{1}{b} \sum_{i=1}^{b} e^{T(\hat{\boldsymbol{y}}_i, \boldsymbol{z}_i)})$
  $\mathcal{L} = \frac{1}{b} \sum_{i=1}^{b} ||D(\boldsymbol{y}_i, \boldsymbol{z}_i) - \boldsymbol{x}_i||_1 + \lambda * \max(0, \mathcal{L}_{MI})$
  $D = D - \epsilon \nabla_D \mathcal{L}$
  $E_S = E_S - \epsilon \nabla_{E_S} \mathcal{L}$
  $T = T + \epsilon \nabla_T \mathcal{L}_{MI}$
end while

We illustrate the network architecture of T in Figure 5.2. Please note that there are approaches to merge the information from content and style representation. In our preliminary experiments, we found that a bi-linear function works better than other simple strategies, such as vector concatenation and addition. Also, compared with common network architectures of discriminators in GANs, the architecture of T doesn't have to be very deep. Specifically, we apply three fully connected layers after the bi-linear layer.

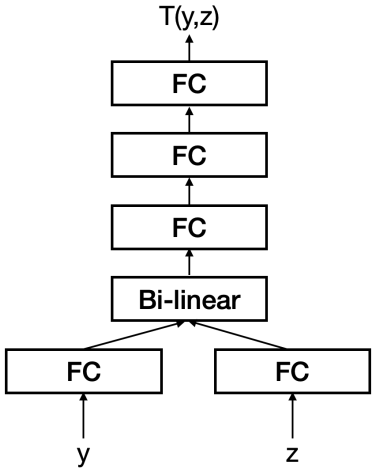


Figure 5.2: The model architecture of the function $T(y, z)$ in MINE.

The pre-training for the content encoder is also a crucial step for style and content disentanglement. If the content encoder is not pre-trained, the model could learn to capture part of the content from style encoder, and still minimize the mutual information between $E_C(\boldsymbol{c})$ and $E_S(\boldsymbol{x})$.

5.4 Experiment

To evaluate the effectiveness of MIST on preventing content leakage and the quality of the generated speech, we conduct qualitative and quantitative studies on the VCTK [163] and the LibriTTS [169] datasets. The VCTK dataset contains 44 hours of clean speech from 109 speakers, and LibriTTS [169] is a large-scale corpus with 585 hours of English speech, which are recorded from 2,456 speakers. For LibriTTS, we use the train-clean-360 set to learn our model. We also use LJSpeech dataset [62], which consists of 13,100 short audio clips from a single speaker, for pre-training the content encoder. For fair comparison, our implementations of the baseline methods also use this pre-trained content encoder.

Baselines: We compare our method with the unsupervised method by [156] that proposed to use global style tokens (GST). The original GST method uses an LSTM based Tacotron 2 [126] as the TTS backbone and an LSTM encoder for computing the style coefficients. For training efficiency and fair comparison, in our implementation of GST, we use Transformer TTS [82] for the content encoder and the decoder, and replace the LSTM with max-pooling for computing the style coefficients. We refer to our implementation of this method as GST*. We also compare our method with a recently proposed supervised controllable speech synthesis method [92]. This method uses speaker identities for optimizing the style vectors. Same as for GST method, our implementation of this method uses Transformer TTS for the TTS backbone. We refer to our implementation of this method as [92]*. All the baseline methods use pre-trained content encoder.

	VCTK	LibriTTS	S / U
[92]*	34.6 ± 0.9	40.0	S
GST* (50 tokens)	50.3 ± 4.2	47.7 ± 1.2	U
GST* (10 tokens)	35.7 ± 0.5	40.3 ± 1.7	U
MIST (50 tokens)	29.3 ± 1.7	44.3 ± 1.7	U
MIST (10 tokens)	$\mathbf{20.3 \pm 1.2}$	$\mathbf{33.3 \pm 1.2}$	U

Table 5.1: Word error rate (WER) on the synthesized speech for the VCTK and the LibriTTS datasets. As shown by the smaller WER, the proposed MIST algorithm preserves the content better than the baselines. The last column shows whether the method is supervised (S) or unsupervised (U).

5.4.1 Quantitative Study

Since the main objective of MIST algorithm is to improve the content quality of the generated speech, we objectively evaluate the performance by measuring the content quality using an ASR (automatic speech recognition) algorithm. Following [92], we adopt WaveNet [104] as the acoustic model in the ASR, and compute word error rate (WER), as a metric for content preservation ability of the model. The Wavenet model is trained on real speech data with Connectionist Temporal Classification (CTC) loss [41] between the predicted and the ground truth characters. For the VCTK dataset, this model achieves a WER of 0.08 on the held-out real data. In the testing phase, we prepare 100 pairs of unmatched text content and reference speech $(\boldsymbol{c}, \boldsymbol{x})$ for both datasets, and report the performance of ASR as WER. A smaller WER indicates less content leakage. We present our results in Table 5.1, where the proposed method improves the WER compared to state-of-the-art methods.

Sensitivity Analysis of the Hyper-parameter λ: To investigate the sensitivity of the hyper-parameter λ, the combination weight between reconstruction loss and MI minimization, we evaluate our model with different values of λ. In this set of experiments, we use 10 tokens in the style encoder, and measure the WER with the VCTK dataset. For a range of λ values, $0.05 - 0.5$, the WER was $0.20 - 0.22$, which shows that MIST is insensitive to exact value of this hyper-parameter.

Analysis of the mutual information loss: After training the speech synthesis model, we expect the mutual information between the style vectors, $(E_S(\boldsymbol{x}))$, and the content vectors, $(E_C(\boldsymbol{c}))$, be small. To verify this hypothesis, we estimate the mutual information between the two random variables (i.e. the style vectors and the content vectors) from our trained model (with frozen weights) using the MINE algorithm, which is shown in Figure 5.3 as function of training epochs. The MINE algorithm optimizes the MINE neural network, T, according to Equation (5.3) and keeps all other parts (D, E_S, E_C) fixed. The MI estimate stays close to 0 for more than 50 epoch with our model, while it increases immediately with the GST* model.

5.4.2 Qualitative Study

To evaluate the quality of the synthesized speech, we conducted a user study with 6 subjects performing a total of 150 tests. Each test consists of a reference speech, a text content (not

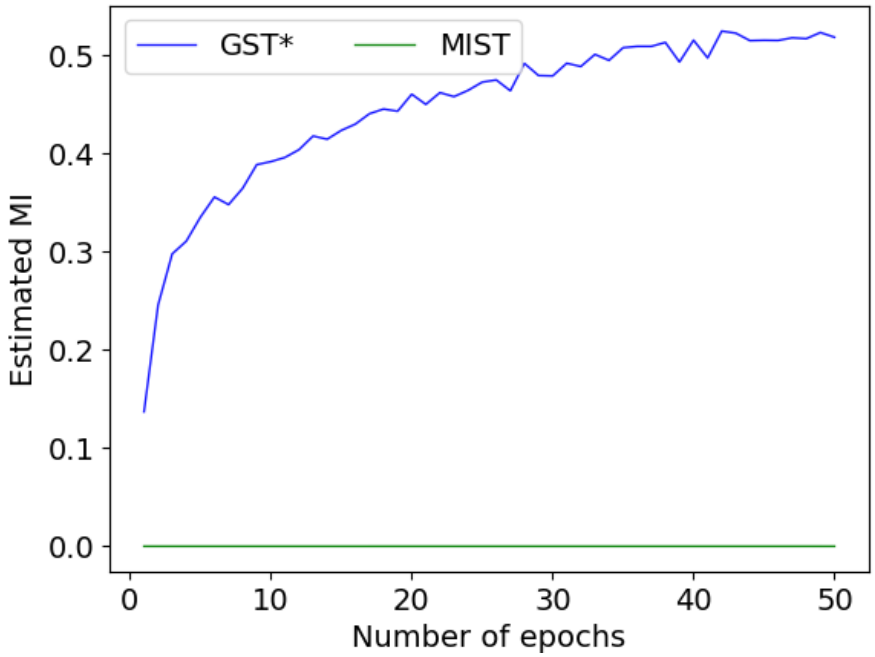


Figure 5.3: The MI estimates, for frozen TTS models, shown as a function of the training epochs of the MINE. Our model has substantially lower mutual information compared to the baseline GST*.

matching with the content of the reference speech), and two synthesized speech samples from GST* and MIST, respectively. The order of both speech samples were randomized for each test. The participants of the study were asked two questions: (1) which synthesized speech preserves content better, and (2) which is more similar to reference speech in terms of style. There were three choices for each question: (1) synthesized speech 1 is better, (2) synthesized speech 2 is better, and (3) both outputs are the same. The results of their ratings are illustrated in Table 5.2. From these results, we can see that MIST preserves content of the input text better, as supported by the better ASR results in Table 5.1, and also preserves the style of the reference speech better, compared to the baseline, GST*, method.

5.5 Conclusion

We proposed an unsupervised mutual information minimization based content and style separation for speech synthesis. In each training step, we estimated the mutual information between the style and the content, and minimized it along with the reconstruction loss. We showed that such training strategy reduces content leakage and results in substantially better WER compared to the baseline approaches.

	Content Preservation	Style Preservation
Both methods are same	29.3	41.3
Baseline (GST*) is better	26.0	17.3
MIST is better	**44.7**	**41.3**

Table 5.2: Qualitative evaluation: The numbers in the first row indicate percentage of time both the methods are rated the same. The second and third row are the percentage of time the method in first column is rated better.

Chapter 6

Style and Content Disentanglement: Pretraining for Downstream Tasks

In this chapter, we show that style/content disentanglement technique preserves useful information in the training dataset, and improves the training process of downstream applications, such as image retrieval and speech recognition. To verify this idea, we propose a pre-training framework with disentangled generative models. This framework aims to prepare customized synthetic datasets for the training of downstream applications. We also demonstrate two benefits provided by the generative model with hidden factor disentanglement in this pre-training framework. First, it can expand the dataset by generating samples of unseen style/content combination. Second, it extends the idea of data augmentation to interpretable hidden spaces. Experiment results in low-resource unsupervised person re-id and speech recognition illustrate the efficacy of these two strategies.

6.1 Introduction

Many applications of multimedia analysis contain a pre-training stage, which aims to learn a general and robust representation from a large-scale dataset. Previous works have shown that the representation obtained from a good pre-trained model can effectively improve the performance of a variety of downstream tasks. A non-exclusive list of such pre-trained models include the image classifier trained on ImageNet [122], the autoregressive language model GPT-3 [15] trained on web-scale text data, and the ViBERT trained on visual-linguistic data [137]. The pre-training process can also be done in an semi/self supervised manners. For example, wav2vec [9, 125] considered the next time step prediction task as the supervision signal, and showed that the pre-trained representation is beneficial to speech recognition.

However, the current mainstream pre-training strategies still have several potential limitations. For example, the objective function for pre-training usually focuses on one single supervised learning task, and the information captured by the pre-trained representation is constrained by the label space of this task, while the downstream applications of interest might require information out of the label space. In this case, we fail to make the best use of the large scale pre-training dataset. Although unsupervised pre-training methods are not dependent on any la-

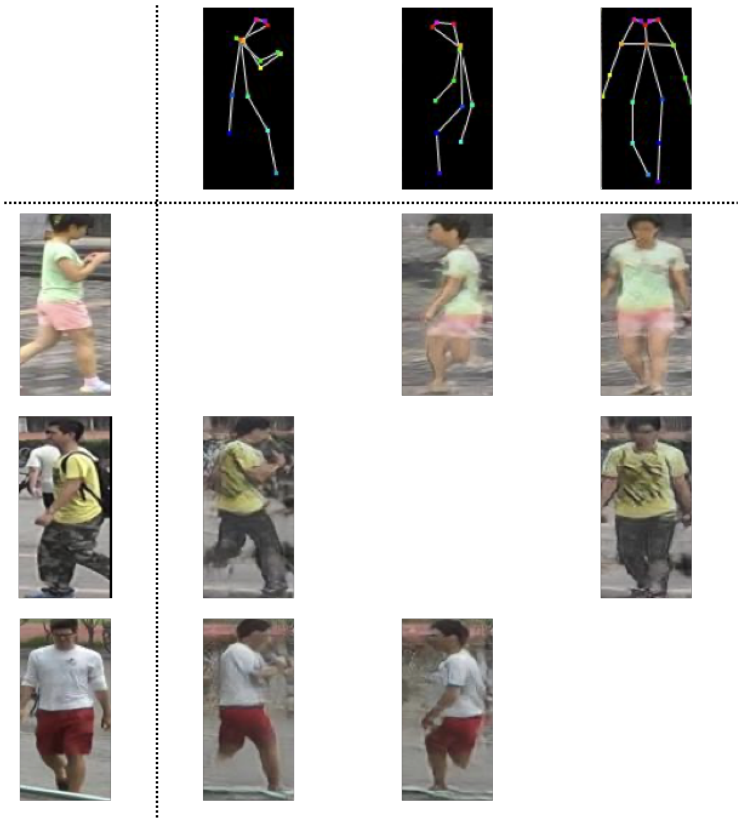
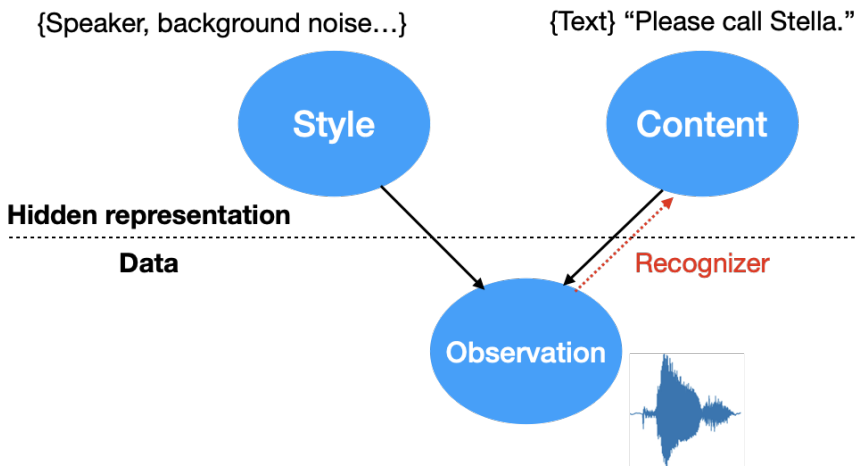


Figure 6.1: Style/content permutation produces unseen training samples and increases the variation of synthetic dataset.

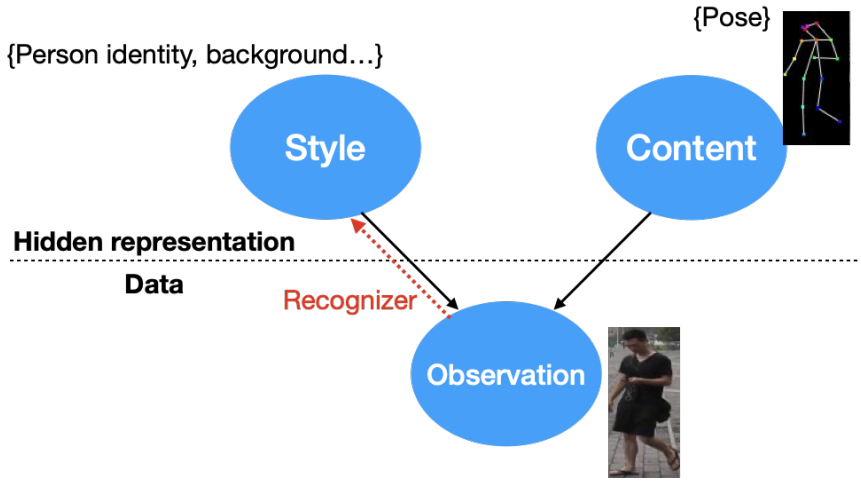
bel information, they usually require a huge amount of data to stabilize the pre-training process. Thus, our goal is to improve the existing pre-training methods by capturing more information hidden in the large scale raw dataset.

In this chapter, we treat the generative model with disentangled representation as a new approach to pre-training. Specifically, in the pre-training stage, we train a generative model by utilizing all the available data, and apply style and content disentanglement to the generative process. During the training phase of the downstream application, a customized synthetic dataset is produced by the pre-trained generative model, and combined with real data to form a better training dataset. The recognition model trained on the combined dataset should yield better performance compared to the one only trained on real data.

In practice, the synthetic data for downstream task training should be of good quality, and complementary to the information of the real dataset. The methods introduced in chapter 4 and 5 have demonstrated how to achieve style and content disentanglement, which provides the controllability in the generative process. In this chapter, we further utilize the controllability, and enrich the variety of synthetic datasets by two simple yet effective strategies. First, as shown in Figure 6.1, we can conduct style and content permutation, which aims to generate unseen training samples or adjusts the data distribution in terms of controllable hidden factors. Second, we can also perform data augmentation in an interpretable hidden representation space.



(a) Incorporating disentangeld generative model to content (seen attribute) recognition



(b) Incorporating disentangeld generative model to style (unseen attribute) recognition

A pre-trained generative model with disentangled representation can support the training of different downstream tasks. Recall that we defined style and content as known and unknown hidden factors, respectively. Our generative model can be utilized to prepare datasets for both style and content recognition tasks, as shown in Figure 6.2a and 6.2b. The preparation for a content recognition dataset is straightforward, while the dataset generation for style (unknown factor) recognition requires the following two assumptions: 1. the content (known factor) contribute to the main data variation other than the target style factor, and 2. precise content manipulation is achieved. For instance, in the task of unsupervised person re-id, the identity annotation of person images is not available. With the technique introduced in chapter 4, the person identity and pose information can be disentangled in the generative process of person images. It means that one can synthesize images of the same person identity and different poses, which constitute training samples for person re-id.

We conduct experiments to verify the efficacy of our generative model for pre-training, as well as two additional strategies to enhance the synthetic data quality. Specifically, we consider automatic speech recognition (ASR) and low-resource person re-id as the examples of content

and style recognition tasks, respectively. The experiment results show that the synthetic dataset produced by our generative model with disentangled representation is beneficial in both style and content recognition tasks. Also, the two additional strategies, style/content permutation and data augmentation within interpretable hidden space, can further improve the performance.

6.2 Related work

Pre-training techniques have been widely adopted in many applications of multimedia analysis. For example, the models pre-trained on image classification task with ImageNet [122] dataset benefit computer vision tasks such as semantic segmentation [46] and action recognition [144]. Brown et al. pre-trained the large language model GPT-3 [15] using web-scale text data and applied it to multiple NLP tasks. Su et al. [137] conducted pre-training on visual-linguistic data, and used the pre-trained model to derive hidden representation in a visual-text joint hidden space. The pre-training process can also be done in an semi/self supervised manners. For example, wav2vec [9, 125] considered that next time step prediction task as the supervision signal, and showed that the pre-trained representation is beneficial to speech recognition. Compared with previous techniques, our method performs pre-training with a generative model with disentangled representation, and potentially preserve more information of multiple hidden factors.

The concept of using data synthesis for ML model training has been investigated in multiple research areas. For the speech recognition tasks, one line of works [12, 13, 45, 51, 143] adopted TTS and semi/self supervised learning techniques to incorporate unpaired speech and text data in training process. Another line of works [27, 120, 121] considered data synthesis as an augmentation method to expand the training dataset. For image recognition tasks, data synthesis process has been utilized to augment the images of unseen classes [6], or provide a balance dataset for ML fairness [95]. In this chapter, we enhance the data synthesis process by style and content disentanglement, and incorporate the prior knowledge to augment training data in both style and content spaces.

Data augmentation aims to generate more training data and increases the robustness of machine learning models. Many previous works design simple, semantic-preserving transformations to modify the raw data. For example, in speech recognition task, this type of transformations include speed perturbation [72], vocal tract length perturbation [63], and time/frequency masking in log-melspectrogram domain [107]. On the other hand, in computer vision tasks such as classification and segmentation, researchers have investigated geometric transformation (rotation, translation, etc.), random erasing [175], and color space transformation [128]. Although most of these transformations can usually improve the model performance independently, a combination policy needs to be adopted to use multiple transformations simultaneously. To alleviate the sub-optimal manual design and hyper-parameter tuning, policy search algorithms [22, 23, 85, 172] have also been developed to discover the optimal combination of transformations and their corresponding hyper-parameters. All these data augmentation methods consider the modification in raw data space. In comparison, the augmentation strategy proposed in this chapter is capable of perturbing data in raw data, output label, and interpretable hidden spaces.

The idea of data augmentation can be also applied to the hidden feature space of a deep network architecture. DeVries et al. [25] explored the effectiveness of extrapolation between

samples in hidden space. Verma et al. [147] extended the original Mixup [171] method to Manifold Mixup, which utilized hidden feature/label interpolation to increase the training robustness. Compared with these approaches, our method augments the training data in an interpretable hidden space with a generative process. Another previous work [81] in few-shot learning also constructed a generative model to hallucinate the hidden features of unseen classes. However, this approach still lacked of interpretability. In contrast, our generative model with disentangled representation keeps the interpretability, which allows us to encode inductive bias from different domains in the model training stage.

6.3 Training recognizers with a disentangled generative model

In this section, we elaborate the proposed strategy: incorporating a pre-trained generative model with disentangled representation to improve the performance of a downstream task. Generally speaking, it is accomplished by a three-step process: (1) Given a real dataset with content annotation, we first use it to pre-train a generative model that can separate the content information from other hidden factors. (2) Then, this generative model is utilized to produce synthetic samples with unseen combination of style and content information. (3) Finally, we train a recognizer of the downstream task with all the real and synthetic samples. The strategies for generative model pre-training (step (1)) have been described thoroughly in Chapter 4 and 5. The focus of this chapter would be step (2) and (3), the preparation of the customized synthetic dataset and the corresponding training process. Please also note that the implementation for the recognition of content (hidden factors with annotation) and style (hidden factors without annotation) are different, which are discussed in Section 6.3.1 and 6.3.2, respectively.

There are several major reasons why our proposed strategy is favorable. First, the disentangled representation empowers the generative model to produce unseen training samples for a downstream task. In the speech recognition task, a synthetic training sample can be an utterance that a particular person never speakers. In the person re-id task, the person in a synthetic image can be doing a pose from another source image (as shown in Figure 6.1). These unseen but realistic synthetic samples increase the variety of training dataset. Second, the disentangled representation allows us to incorporate the inductive bias of different hidden factors to the downstream task training. For instance, the language prior knowledge is utilized effectively in the training phase of speech recognition if we can transform the text into audio domain. Based on this concept, we can also conduct data augmentation in an interpretable representation space (Section 6.3.3 and 6.3.4).

The most important prerequisites of our proposed strategy are the quality and disentanglement ability of the pre-trained generative model. Specifically, the generative model should be able to manipulate the content information, analyze the style information from a reference real data sample, and combine the style and content in its output. To fulfill these prerequisites, in this chapter, we utilize the generative models introduced in Chapter 4 and 5, which are designed for person image and speech, respectively.

6.3.1 Training a content recognizer

Given a pre-trained generative model g with disentangled representation, we can use it to produce synthetic samples with unseen content and style combination. These synthetic samples enhance the training dataset of a downstream content recognizer. Recall that the generative model g takes a pair of raw data sample x and content y as input, and produces an synthetic sample $\hat{x} = g(x, y)$ that contains the style of x and the content y. With g and a real dataset $\{x_i, y_i\}_{i=1}^N$, the training process of the downstream content recognizer can be formulated as:

$$\Theta^* = \arg\min_{\Theta} \{\sum_i l_c(f_\Theta(x_i), y_i) + \beta l_c(f_\Theta(g(x_{j(\neq i)}, y_i)), y_i)\} \tag{6.1}$$

where the l_c is the per-sample training loss function for the content recognition task, and β is the hyper-parameter describing the relative importance of synthetic data. The first and second terms of Eq. 6.1 are the total loss values of real and synthetic samples, respectively. And the x_j in the second term is sampled from the real dataset.

6.3.2 Training a style recognizer

The generative model g with disentangled representation can also help the training set preparation of a style recognizer. Recall that "style" means a type of hidden attribute whose annotation is no available. Thus, we are not able to train a recognizer in a supervised learning method. For example, we have a person image dataset without identity information, but still want to train a model for person re-id.

To achieve this goal, we use the generative model g to prepare multiple data samples with the same style but different content information. Repeating the same process for a set of style reference samples, we receive a synthetic dataset for supervised learning method. Formally, the training process of a style recognizer with synthetic data is formulated as:

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{N} \sum_{j=1}^{M} l_s(f_\Theta(g(x_i, y_j)), i) \tag{6.2}$$

where l_s is the loss function of the style recognition task, and M is the total number of samples with the same style in the synthetic dataset.

In the context of person re-id, clustering based methods [36, 38] have also been studied to solve this task in an unsupervised manner. However, these methods require some prior knowledge of style information, i.e. total number of identity in the training set. Also, if there are few or no data samples with the same identity in the training set, the clustering would not be applicable. In contrast, our strategy of synthetic data training doesn't have these disadvantages.

6.3.3 Data augmentation in representation space: inductive bias

The generative model with disentangled representation extends the idea of data augmentation from raw data space to interpretable representation space. Conventional data augmentation techniques increase the diversity of training datasets by introducing target-invariant transformations.

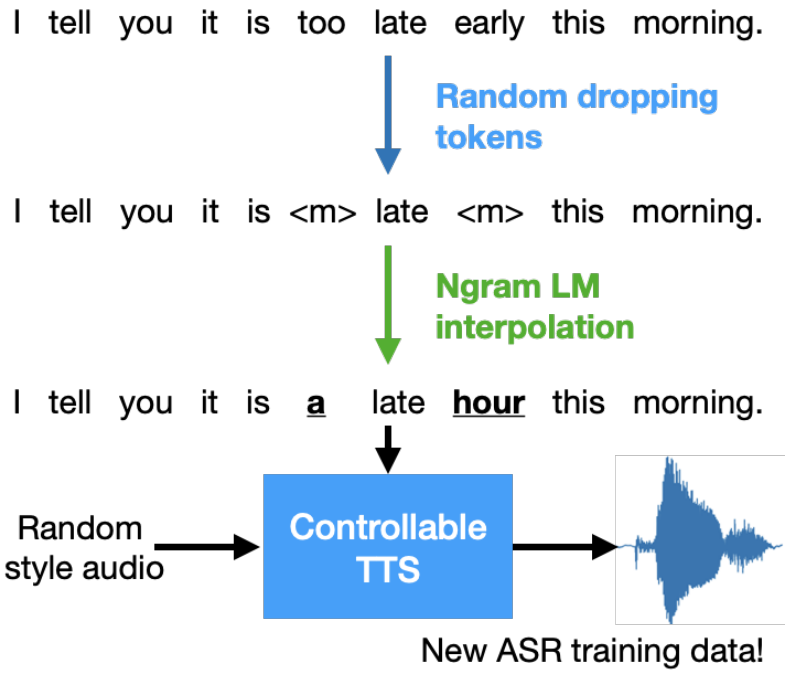


Figure 6.3: ASR data augmentation by n-gram perturbation. Given a sentence in the real training dataset, we first randomly drop a portion of its tokens, and sample the replacement tokens by a n-gram language model. The new sentence with sampled tokens then is taken as the input of our controllable TTS model in order to generate corresponding speech signal.

These transformations are domain specific and designed based on some inductive bias of raw data space. For example, rotation, flipping and color tone are common target-invariant transformations in image recognition tasks.

With our generative model, one can transform the data in a representation space, and this transformation would be reflected in the generated data sample. In the context of speech recognition, we incorporate inductive bias in text domain to the generative process, and produce unseen but reasonable speech data. In this work, we utilize two types language prior as inductive bias for augmented training data generation. The first one is the perturbation by a language model. As shown in Fig. 6.3, given a text sentence, we randomly drop a portion of tokens, and fill these places by tokens generated from a n-gram language model. The second type of inductive bias is paraphrasing. Specifically, we use a pre-trained paraphraser to produce a parallel sentence for the text transcription of each training sample, and synthesize the corresponding speech by our TTS model. Some paraphrasing examples are shown in Tab. 6.1.

6.3.4 Data augmentation in representation space: Rep-Mixup

In this section, we introduce representation mixup (Rep-Mixup), another type of data augmentation method enabled by our pre-trained generative model. Rep-Mixup is inspired by the original Mixup [171], which is an augmentation method operated in the space of image pixel values. Mixup aims to encourage the model to perform linearly in-between two data points. Specifically,

original sentence	The five thirty train has been in and gone half an hour ago.
paraphrase (generated)	The five thirty train left half an hour earlier than expected.
original sentence	he never did any work except to play the pipes.
paraphrase (generated)	he didn't do anything other than playing the pipes.
original sentence	there i stay until all danger is over.
paraphrase (generated)	I stay there until everything is finished.
original sentence	It was dark before he came back to his home and his father was still asleep.
paraphrase (generated)	when he went back to his home it was dark and his father was still asleep.

Table 6.1: Examples of paraphrasing text. The original sentences are from the text corpus of LibriSpeech 100h dataset. The paraphrased sentences are generated from the open source paraphrasing toolkit, Parrot Paraphraser [24].

given two training samples (x_1, y_1) and (x_2, y_2), mixup produces a new training sample (x', y') by the following formulation:

$$\begin{aligned} x' &= (1-\lambda)x_1 + \lambda x_2 \\ y' &= (1-\lambda)y_1 + \lambda y_2 \end{aligned} \tag{6.3}$$

where $\lambda \in [0, 1]$ is drawn from a beta distribution $Beta(\alpha, \alpha)$. Previous works reported that mixup improves the generalization and robustness of the trained model. Inspired by this result, we propose Rep-Mixup, which imposes the same linear relationship in the interpretable representation space, so that the model training process can be more focused on the hidden factors relevant to the downstream task. Specifically, Rep-mixup incorporates augmented training samples produced from the interpolation of target related hidden representation.

In the context of low-resource person re-id, the proposed Rep-mixup first obtains an interpolated hidden representation in the space of person identity information. Then it generates the corresponding person image by the pretrained generative model with disentangeld representation. Considering the generative model introduced in Chapter 4, we can express the person identity representation F of a person image x as following:

$$F = \arg\min_{F'} ||H - PF'||_p \tag{6.4}$$

where $H = f_a(x)$ and $P = f_p(K)$ are the appearance and pose feature matrices, respectively, and K is the keypoint-based pose input. Please check Sec. 4.3 for more details. Based on this generative process and two input person images (x_1, y_1) and (x_2, y_2), Rep-mixup produces the augmented training sample (x', y') by the following formulation:

$$\begin{aligned} y' &= (1-\lambda)y_1 + \lambda y_2 \\ x' &= G(F') \\ F' &= \arg\min_{F} \{(1-\lambda)||H_1 - P_1F|| + \lambda||H_2 - P_2F||\} \end{aligned} \tag{6.5}$$

where G is the mapping function from hidden representation to raw image space. In comparison with the original mixup, the proposed Rep-mixup interpolates two training samples in the representation space instead of raw data space. Hence, Rep-mixup keeps other hidden factors untouched, and generates more realistic images as shown in Figure 6.4.

Figure 6.4: Examples of augmented images for low-resource person re-id. Rep-mixup interpolates image 1 and 2 in the space of person identity representation, in contrast to the pixel level interpolation of Mixup [171]. The pose information of image 1 is adopted in the generation phase.

6.4 Experiment

To evaluate our proposed pre-training framework with generative model and the corresponding data augmentation methods, we choose two types of downstream tasks, automatic speech recognition (ASR) and low-resouce person re-id, which serve as the examples of content and style recognition tasks, respectively. Particularly, the experiment results show that our generative models with disentangled representation provide useful synthetic training data for both types of tasks. And the two data augmentation methods in representation space also further improve the performance.

6.4.1 Experiment setup: ASR

For ASR experiments, we utilize the LibriSpeech dataset [106], which consists of 1,000 hours of speech from public domain audiobooks. Following the standard protocol, we evaluate our method on a 100-hour subset, LibriSpeech-100h, which contains only clean speech with US English accents. To generate synthetic datasets, we adopt the controllable TTS model introduced in Chapter 5. This model is trained with LibriTTS-100h dataset [169] containing 54 hours of speech data derived from LibriSpeech-100h. Please note that the main purpose of this setup is to investigate the benefits of disentangled generative process as a pretraining step. Thus, we explicitly constrain the training set of generative model to be a subset of the dataset for downstream recognizer training.

Baseline ASR model: We use the implementation from ESPNet [158] for an end-to-end ASR model. The ASR model is composed of a conformer-based encoder [43] and a transformer-based decoder [146]. We apply SpecAugment [107] to all speech samples to further enhance the acoustic diversity. The model checkpoint of each epoch is saved, and the final model is produced by averaging the 10 checkpoints with the best validation accuracy. All ASR models are evaluated without a language model.

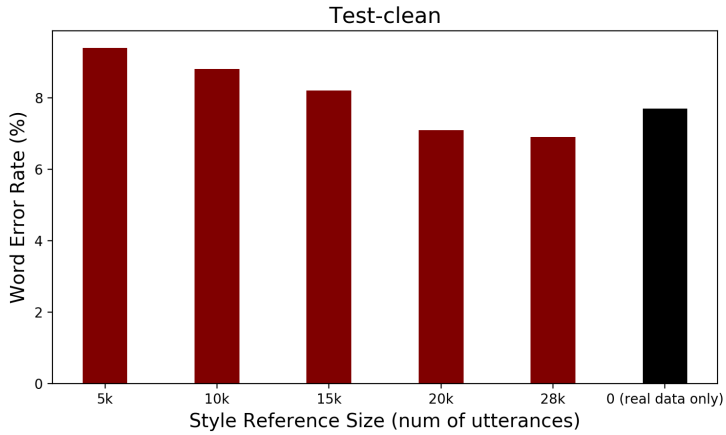
6.4.2 Experiment results: ASR

Pre-training with a generative model and style variation: In the first experiment of ASR, we show the applicability of the proposed pre-training framework with generative model. Specifically, we utilize the controllable TTS model introduced in Chapter 5 and generate a synthetic copy of LibriSpeech-100h. This synthetic speech dataset is combined with the original LibriSpeech-100h to form a new training dataset. Recall that our controllable TTS model requires a reference utterance for the style information. In this experiment, we prepare multiple combined (real+synthetic) datasets with different size of style reference set. With a larger size of style reference set, the synthetic dataset would cover a wider range of speaking styles and background noise conditions. Then, we evaluate the ASR models trained on the combined datasets by reporting the word error rate (WER) on the two standard test sets of LibriSpeech: test-clean and test-other.

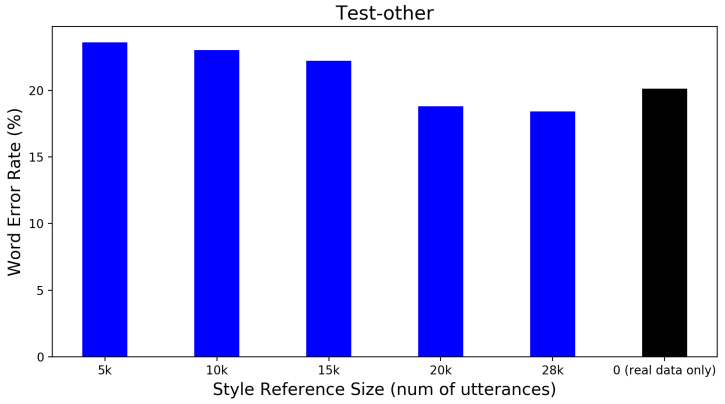
From the results illustrated in Figure 6.5, we observe that adding a synthetic dataset with enough style variation improves the performance of the trained ASR model. In both test-clean and test-other, the ASR models trained on additional synthetic datasets with 20k and 28k (full set of LibriSpeech 100h) style reference utterances outperform the model trained on real data only. This set of results suggest that generating additional data with different style and content combination helps the training process of the downstream task, and our proposed pre-training framework with a generative model enables this procedure.

Data augmentation in the representation space: In the second experiment, we investigate the performance of two data augmentation methods in the representation space. Both of these methods, n-gram replacement and paraphrasing, incorporate the language prior knowledge in the preparation of synthetic dataset. For the n-gram replacement, we set the probability of dropping a text token to 30%, and the language model for replacement is a 4-gram model trained on the text transcription of LibriSpeech-100h. For the paraphrasing, we use a pre-trained provided by the open source toolkit, Parrot Paraphraser [24].

The experiment results on test-clean and test-other are reported in Table 6.2. From the results, one can see that both data augmentation methods provide improvement in WER on both testing sets. It demonstrates the additional benefit of our proposed pre-training framework, which is the usage of inductive bias in the representation space. Please note that the pre-trained paraphrasing model incorporates additional text information, so the results in the last row of Table 6.2 are not a fair comparison to the results of other models.



(a) Results on LibriSpeech Test-clean



(b) Results on LibriSpeech Test-other

Figure 6.5: Experiment on the importance of style variation in the synthetic dataset. The results show that we should use a synthetic speech dataset with enough style variation in order to improve the WER.

6.4.3 Experiment setup: low-resource person reid

For low-resource person re-id task, we utilize Market-1501 dataset [173], which contains 32,668 annotated bounding boxes of 1,501 identities. To demonstrate the benefits of our pretraining framework for unannotated attributes, we extensively constrain the training resource for both generative models and downstream recognizers. Specifically, we take 1 image for each identity in the Market-1501 training set, and the total dataset for both pretraining (generative model) and downstream recognition contains 751 images. Please note that in this low-resource setup, we don't have any pair of images with the same person identities. Thus, the clustering-based methods [36, 38] for unsupervised person re-id are not applicable. To synthesize the person images of the same identity in different poses, we adopt the generative model developed in Chapter 4. In the pretraining phase, we need to apply the unsupervised training technique introduced in Section 4.3.3. For the downstream person re-id, we prepare 10,000 synthetic person images. The

	Test-clean	Test-other
Real	7.7	20.1
Syn	15.6	31.5
Real + Syn	6.9	18.4
Real + Syn, Ngram replacement	6.6	17.2
Real + Syn, Paraphrasing	5.9	17.7

Table 6.2: Evaluation of data augmentation methods in the interpretable hidden space. All the number are word error rate (%)

person identity information of the synthetic dataset is only from the 751 real images in the real training set, while the pose information comes from the detected pose skeleton maps of the full Market-1501 training set. We use the torchreid toolkit [176] to build a standard person re-id model with resnet-50 as its backbone. The model is trained with cross-entropy minimization, and the cosine distance between features of the last fully connected layer is considered as the similarity measure in the inference phase. The results are reported in top-1 accuracy and mean average precision (mAP).

6.4.4 Experiment results: low-resource person reid

From the results illustrated in Table 6.3, we can make several observations. First, the ImageNet pretrained model performs the worst (row 1) since it is not finetuned on the domain specific data. Second, with a very limited amount of real person images and data augmentation in raw space (e.g. mixup), we can still improve from the ImageNet pre-trained model. Third, the synthetic person images provide useful training signal for person re-id model (row 3), and the data augmentation method in the interpretable hidden space (Rep-Mixup, row 5) can further improve the performance.

Training set	Top-1 Acc (%)	mAP (%)
None (ImageNet pretrained)	6.8	2.0
751 real images, Mixup	33.4	15.6
751 real + 10k syn images	42.2	20.4
751 real + 10k syn images, Mixup	45.8	23.0
751 real + 10k syn images, Rep-Mixup	**46.5**	**23.2**

Table 6.3: Experiment results on low-resouce person re-id.

6.5 Conclusion

In this chapter, we propose a pretraining framework for tasks of multimedia analysis. This framework prepares a generative model with disentangled representation, and utilizes it to produce a customized synthetic dataset for the downstream task. Because of the controllability of the generative model, we can generate training samples for the task without any target annotation. This

framework also enables data augmentation methods in an interpretable representation space, and further improves the generalization and robustness of downstream task training. The experiment results on low-resource person reid and ASR show that the generative model successfully preserves information of hidden factors in the pretraining dataset, and provides high quality training samples to improve downstream tasks with/without target annotation.

Chapter 7

Conclusion

In this thesis, we focus on the representation learning of multimedia content. Particularly, we observe the importance of capturing the diversity, uncertainty and multiple hidden attributes in a dataset to achieve good performance of analysis/recognition tasks. Based on the observation, this thesis aims to enhance the capability of multimedia representation by the investigation toward two research directions: representation beyond vectors, and style/content disentanglement in the generative process.

In this chapter, we summarize the main contribution of this thesis in Sec, 7.1, describe the key ideas we learned during the investigation in Sec. 7.2, and discuss some future research directions in Sec. 7.3

7.1 Contributions

The main contribution of this thesis is three-fold:

Representation beyond vectors: In the area of multimedia representation learning, most of the existing works utilize a feature vector to represent one instance in a dataset. However, the capability of vector-based representation is not enough to capture the diversity and uncertainty of multimedia content. For example, a vector fails to preserve the multi-mode property of a video tracklet. Hence, in this thesis, we propose two novel types of representation beyond a feature vector: distribution (Chapter 2) and subspace (Chapter 3). We also develop algorithms which incorporate both types of representation to deep learning architectures, and enable end-to-end training. In general, deep learning models with distribution or subspace representation yield better performance, and keep the same number of trainable parameters compared to models with vectors. This idea has been verified on the retrieval of image sets and video tracklets, as well as the few-shot learning task.

Generative models with style and content disentanglement: The second part of this thesis aims to preserve and manipulate the multiple hidden factors in multimedia content. To achieve this goal, we investigate the idea of style/content disentanglement to capture the unseen (style) and seen (content) factors in generative models of multimedia data. Furthermore, we propose two methods to enhance the quality of disentanglement in generative models. Specifically, both methods enable the unsupervised learning of disentangled style representation. The first method

(Chapter 4) models the relation between style and content as a simple matrix operation in hidden space, while the second method (Chapter 5) explicitly measures and minimizes the mutual information between style and content hidden features. These two methods have been evaluated on keypoint based image generation and controllable text-to-speech, respectively.

Pre-training framework with generative models: Finally, we hypothesize that the preservation of hidden factors is beneficial to the pre-training of large-scale multimedia analysis. To verify this hypothesis, we design a two-step pre-training framework with generative models (Chapter 6). The first step of this framework is the training of generative models with style/content disentanglement, while the second step is the preparation of customized synthetic training datasets for downstream recognition tasks. The disentangled representation also enables style/content permutation and data augmentation in interpretable hidden spaces, which yields additional value of synthetic training datasets (Chapter 6). The efficacy of this pre-training framework has been demonstrated on low-resource unsupervised person re-identification and speech recognition tasks.

7.2 Key Ideas

There are several key ideas we learned from the investigation of multimedia representation. They are summarized in this section in order to provide some suggestions for future researchers working on this direction.

Alleviate the computation requirement of representation beyond vectors by distance function approximation or efficiency/capacity trade-off: In the first part of the thesis, we explore two types of representation beyond vectors: distribution and subspace. Compared with vector-based representation, distribution and subspace representations tend to cost more computation resources while incorporated into a DNN framework. The two main proposed components, estimation of distribution/subspace and distance function calculation, both might introduce additional computation. While the best approach to improve the efficiency should be designed specifically for the representation type and application, we describe two strategies adopted in this thesis. In Chapter 2, we utilize the exact primal form of Wasserstein distance (Eq. 2.1) to measure the dissimilarity between two empirical distributions. By definition, the calculation of primal form Wasserstein distance is a linear programming problem, and difficult to parallelize. To improve this calculation, we incorporate the un-rolled iterative approximation (e.g. IPOT algorithm [162]) to the forward propagation of the DNN framework. Each step of the iterative approximation consists of a set of simple matrix/element-wise operations, which can be easily computed in GPUs. On the other hand, in Chapter 3, the number of basis in a subspace influence the computation of SVD backward propagation and the learning of template subspace (Eq. 3.6). While increasing the number of basis introduces extra computation, we also observe that the performance gain saturates at some point. In few-shot image classification tasks, subspace representation using 6 basis components receives similar accuracy scores as those using more, serving as a reasonable balance between computation and performance. In summary, to improve the learning efficiency of representation beyond vectors, one may first consider two possible directions: (1) approximation of distance functions and (2) the trade-off between efficiency and representation capacity.

Incorporate domain-specific knowledge into the design of the style encoder architecture: Style encoder is the most crucial component in a disentangled generative model. Normally, we would expect that a style encoder should take a reference sample as input, mimic the style of this sample, and get rid of all the information about content. Also, the style encoder should be trained in an unsupervised manner without any annotation. Thus, it is necessary to incorporate some prior knowledge of the style information into the style encoder design. In the context of TTS with style modeling [58, 156], the style encoder maintains a small set of tokens capturing a few major factors in the style domain, and use a pooling layer to make the style representation time-invariant. Also, the performance the trained controllable TTS is sensitive to the number of tokens. We may need to pick a proper number of tokens for the style factor (e.g. emotion, speaker, etc.) we are trying to model. In the context of keypoint based image generation [56], the appearance feature extractor is designed to be a CNN, and focuses on local patterns of an image. Thus, the overall network can learn to reconstruct the style representation by partial, local observation only. If we choose a network architecture (e.g. transformer) that is aware of the global context, preliminary experiment results illustrate that the robustness of unsupervised training would deteriorate significantly. In summary, we believe that the design of style encoder architecture will play an important role while applying style/content disentanglement to a new research problem.

Use real/synthetic cross validation to verify the synthetic dataset quality: In Chapter 6, we utilize customized synthetic dataset to train a recognizer for the downstream application, and an important prerequisite to receive a high quality synthetic dataset. Ideally, our synthetic dataset should preserve the correct content information, and provide a good coverage of style space for the generalization purpose. The quality of a synthetic dataset can be estimated by real/synthetic cross validation. For example, in Chapter 5, we use synthetic speech to form a testing set, and evaluate it by a recognizer (ASR) trained with real data. The results of this evaluation step indicate the content correctness of synthetic data. As another example, in Chapter 6, we train a recognizer with synthetic data only, and use this recognizer to evaluate real validation data. The performance illustrates the domain gap between real and synthetic data in general, but can't distinguish between the content correctness and style coverage. The real/synthetic cross validation is a domain-agnostic strategy, while some domain specific knowledge should also be applicable.

7.3 Future Works

This thesis introduced several new concepts for multimedia representation learning, which enable a wide range of follow-up research directions. Here, I provide a non-exclusive list of examples.

7.3.1 Representation beyond vectors:

Chapter 2 and 3 of this thesis introduce the concept of representation beyond vectors, and demonstrated the advantages by developing the representation learning framework for distribution and subspace. To advance the development of this concept, I elaborate three possible research directions in the following:

(1) Representation with sample-dependent capacity: In a multimedia dataset, the richness, or the total amount of information in a data sample can be varied a lot. For example, the dataset may contain videos with seconds or minutes. To better model this characteristics, one possible way is to design representation with sample-dependent capacity. In distribution and subspace based representation learning frameworks, the capacity is measured by number of supports and basis components, respectively, and it doesn't affect the calculation of distance function. E.g. the distance between empirical distributions with different number of supports is still well-defined. However, the main challenge might be the online, efficient estimation of sample-dependent capacity.

(2) Extension of deep metric learning: Based on the techniques proposed in this thesis, we have extended the concept of deep metric learning (DML) from vector space to distribution and subspace. Thus, any strategies developed in vector based DML could be applicable. For instance, one can try to apply loss functions originally designed for vector based representation. On the other hand, one can also investigate unsupervised learning tasks requiring a metric, such as anomaly detection and distance based clustering.

(3) New types of representation beyond vectors: Besides distribution and subspace, there could be other types of basic unit for multimedia representation. For instance, embedded graph is a potential candidate, since the distance function between two embedded graphs has been explored [3], and vertex level embedding contains enough representation ability. While investigating new types of representation beyond vector, one would need to develop algorithms for end-to-end training, and consider the efficiency of the training process.

7.3.2 Synthetic data for downstream task training:

In Chapter 6, we have shown that synthetic data produced from our generative models can be used for recognition tasks. Also, the models with disentangled representation (Chapter 4 and 5) can further increase the dataset variety, and incorporate domain specific knowledge. However, the potential of synthetic data hasn't been fully investigated yet, and some possible approaches can further provide additional value to synthetic data for downstream task training. For example, the synthetic dataset would be more valuable when the amount of real dataset is limited. Thus, the synthetic data training framework should be able to resolve the ML problems of unseen domains/tasks/demographics. Another possible direction would be to conduct data generation during the downstream task training. In this thesis, synthetic data preparation is always separated from the training of downstream tasks. However, if the synthetic data could be customized for the current checkpoint of recognizers, Hence, the training process might become much more efficient.

Bibliography

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. In *Advances in neural information processing systems*, pages 9562–9574, 2019. 32

[2] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. 10

[3] Hugo A Akitaya, Maike Buchin, Bernhard Kilgus, Stef Sijben, and Carola Wenk. Distance measures for embedded graphs. *Computational Geometry*, 95:101743, 2021. 72

[4] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145, 2017. 32

[5] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, et al. Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2), 2016. 38

[6] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017. 58

[7] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017. 45

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 7

[9] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020. 55, 58

[10] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005. 10

[11] Shane T Barratt and Stephen P Boyd. Least squares auto-tuning. *Engineering Optimization*, pages 1–22, 2020. 32

[12] Murali Karthick Baskar, Shinji Watanabe, Ramon Astudillo, Takaaki Hori, Lukáš Burget, and Jan Černockỳ. Semi-supervised sequence-to-sequence asr using unpaired speech and

text. *arXiv preprint arXiv:1905.01152*, 2019. 58

[13] Murali Karthick Baskar, Lukáš Burget, Shinji Watanabe, Ramon Fernandez Astudillo, et al. Eat: Enhanced asr-tts for self-supervised speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6753–6757. IEEE, 2021. 58

[14] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *ICML*, 2018. 3, 46, 48

[15] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 55, 58

[16] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017. 1

[17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 34

[18] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Regarding gait as a set for cross-view gait recognition. *AAAI*, 2019. 5, 15

[19] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 3, 45

[20] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *ICLR*, 2019. 13, 19, 23, 25

[21] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016. 3, 45, 47

[22] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 58

[23] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 58

[24] Prithiviraj Damodaran. Parrot: Paraphrase generation for nlu., 2021. xiii, 62, 64

[25] Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017. 58

[26] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*. 48

[27] Chenpeng Du and Kai Yu. Speaker augmentation for low resource speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7719–7723. IEEE, 2020. 58

[28] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 26

[29] Basura Fernando and Stephen Gould. Learning end-to-end video classification with rank-pooling. In *International Conference on Machine Learning*, pages 1187–1196. PMLR, 2016. 8

[30] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 19, 20

[31] Rémi Flamary, Alain Rakotomamonjy, and Gilles Gasso. Learning constrained task similarities in graphregularized multi-task learning. *Regularization, Optimization, Kernels, and Support Vector Machines*, page 103, 2014. 8

[32] Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018. 11

[33] L Franceschi, P Frasconi, S Salzo, R Grazzi, and M Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, volume 80, pages 1563–1572, 2018. 8

[34] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015. 7

[35] Yang Fu, Xiaoyang Wang, Yunchao Wei, and Thomas Huang. Sta: Spatial-temporal attention for large-scale video-based person re-identification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8287–8294, 2019. 13, 14

[36] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S Huang. Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6112–6121, 2019. 60, 65

[37] 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*, pages 2414–2423, 2016. 47

[38] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *arXiv preprint arXiv:2006.02713*, 2020. 60, 65

[39] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018. 8, 11

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahra-

mani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 31, 47

[41] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. 51

[42] Daniel W. Griffin, Jae, S. Lim, and Senior Member. Signal estimation from modified short-time fourier transform. *IEEE Trans. Acoustics, Speech and Sig. Proc*, 1984. 47

[43] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020. 64

[44] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proc. CVPR*, pages 3018–3027, 2017. 20

[45] Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda. Back-translation-style data augmentation for end-to-end asr. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 426–433. IEEE, 2018. 58

[46] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 58

[47] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 14

[48] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016. 5

[49] 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. 2017. 3, 45

[50] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015. 5

[51] Takaaki Hori, Ramon Astudillo, Tomoki Hayashi, Yu Zhang, Shinji Watanabe, and Jonathan Le Roux. Cycle-consistency training for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6271–6275. IEEE, 2019. 58

[52] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Vrstc: Occlusion-free video person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192, 2019. 5, 13, 14

[53] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan

Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. Hierarchical generative modeling for controllable speech synthesis. *ICLR*, 2019. 46, 47

[54] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1875–1882, 2014. 5, 7

[55] Ting-Yao Hu and Alexander G Hauptmann. Multi-shot person re-identification through set distance with visual distributional representation. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 262–270, 2019. 2, 3

[56] Ting-Yao Hu and Alexander G Hauptmann. Pose guided person image generation with hidden p-norm regression. *2021 IEEE International Conference on Image Processing (ICIP)*, 2021. 3, 4, 71

[57] Ting-Yao Hu and Alexander G Hauptmann. Statistical distance metric learning for image set retrieval. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021. 2, 4

[58] Ting-Yao Hu, Ashish Shrivastava, Oncel Tuzel, and Chandra Dhir. Unsupervised style and content separation by minimizing mutual information for speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3267–3271. IEEE, 2020. 3, 4, 71

[59] Ting-Yao Hu, Zhi-Qi Cheng, and Alexander G Hauptmann. Subspace representation learning for few-shot image classification. *arXiv preprint arXiv:2105.00379*, 2021. 2, 4

[60] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR, 2017. 47

[61] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 7

[62] Keith Ito. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017. 50

[63] Navdeep Jaitly and Geoffrey E Hinton. Vocal tract length perturbation (vtlp) improves speech recognition. In *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, volume 117, page 21, 2013. 58

[64] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11719–11727, 2019. 20

[65] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31, 2018. 46, 47

[66] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style

transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 36

[67] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 3, 45

[68] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020. 45

[69] Jaekyeom Kim, Hyoungseok Kim, and Gunhee Kim. Model-agnostic boundary-adversarial sampling for test-time generalization in few-shot learning. In *European Conference on Computer Vision*, pages 599–617. Springer, 2020. 20

[70] D Kinga and J Ba Adam. A method for stochastic optimization. In *ICLR*, 2015. 13, 15

[71] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 36

[72] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*, 2015. 58

[73] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 1951. 48

[74] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, et al. Fader networks: Manipulating images by sliding attributes. In *NeurIPS*, 2017. 45, 47

[75] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010. 38, 40

[76] Stéphane Lathuilière, Enver Sangineto, Aliaksandr Siarohin, and Nicu Sebe. Attention-based fusion for multi-source human image generation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 439–448, 2020. 31, 39

[77] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 19, 25

[78] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*, 650, 2017. 7

[79] Fei Li, Qionghai Dai, Wenli Xu, and Guihua Er. Weighted subspace distance and its applications to object recognition and retrieval with image sets. *IEEE Signal Processing Letters*, 16(3):227–230, 2009. 19, 22

[80] Jianing Li, Jingdong Wang, Qi Tian, Wen Gao, and Shiliang Zhang. Global-local temporal representations for video person re-identification. In *ICCV*, pages 3958–3967, 2019. 14

[81] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. Adversarial feature hallucination networks

for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13470–13479, 2020. 59

[82] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Peng Shi. Neural speech synthesis with transformer network. In *AAAI*, 2019. 45, 46, 47, 50

[83] Shuang Li, Slawomir Bak, Peter Carr, and Xiaogang Wang. Diversity regularized spatiotemporal attention for video-based person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 369–378, 2018. 13, 14

[84] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019. 19, 20, 22, 25, 28

[85] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32:6665–6675, 2019. 58

[86] Chia-Ching Lin, Yu-Chiang Frank Wang, Chin-Laung Lei, and Kuan-Ta Chen. Semantics-guided data hallucination for few-shot visual classification. In *Proc. ICIP*, pages 3302–3306. IEEE, 2019. 20

[87] Chih-Ting Liu, Chih-Wei Wu, Yu-Chiang Frank Wang, and Shao-Yi Chien. Spatially and temporally efficient non-local attention network for video-based person re-identification. *BMVC*, 2019. 13, 14

[88] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5904–5913, 2019. 31

[89] 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*, pages 4114–4124. PMLR, 2019. 3

[90] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1137–1145, 2015. 7

[91] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in neural information processing systems*, pages 406–416, 2017. 31, 38, 39

[92] Shuang Ma, Daniel Mcduff, and Yale Song. A generative adversarial network for style modeling in a text-to-speech system. In *ICLR*, 2019. 46, 47, 50, 51

[93] Clement Magnant, Audrey Giremus, and Eric Grivel. Jeffreys divergence between state models: Application to target tracking using multiple models. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5. IEEE, 2013. 7

[94] Clement Magnant, Audrey Giremus, and Eric Grivel. On computing jeffrey's divergence between time-varying autoregressive models. *IEEE Signal Processing Letters*, 22(7):915–919, 2014. 7

[95] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018. 58

[96] Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao. Flow-tts: A non-autoregressive network for text to speech based on flow. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7209–7213. IEEE, 2020. 45

[97] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 31

[98] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017. 5, 7

[99] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *Proc. ICML*, pages 3664–3673. PMLR, 2018. 20

[100] Frank Nielsen and Richard Nock. Sided and symmetrized bregman centroids. *IEEE transactions on Information Theory*, 55(6):2882–2904, 2009. 10

[101] Yasunori Nishimori and Shotaro Akaho. Learning algorithms utilizing quasi-geodesic flows on the stiefel manifold. *Neurocomputing*, 67:106–135, 2005. 24

[102] Peter Ochs, René Ranftl, Thomas Brox, and Thomas Pock. Bilevel optimization with nonsmooth lower level problems. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 654–665. Springer, 2015. 8, 11

[103] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016. 5, 7

[104] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 47, 51

[105] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, pages 15578–15588, 2019. 7

[106] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015. 63

[107] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 58, 64

[108] Fernando Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE international symposium on information theory*, pages 1666–1670. IEEE, 2008. 9

[109] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: 2000-speaker neural text-to-speech. In *ICLR*, 2018. URL `https://openreview.net/forum?id=HJtEm4p6Z`. 47

[110] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. *arXiv preprint arXiv:1811.00002*, 2018. 47

[111] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 650–667, 2018. 31

[112] René Ranftl and Thomas Pock. A deep variational model for image segmentation. In *German Conference on Pattern Recognition*, pages 107–118. Springer, 2014. 8

[113] Yongming Rao, Jiwen Lu, and Jie Zhou. Attention-aware deep reinforcement learning for video face recognition. In *ICCV*, pages 3931–3940, 2017. 5

[114] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proc. ICLR*, 2017. 19, 25

[115] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Proc. ICLR*, 2018. 25

[116] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031. 19

[117] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, 2020. 45

[118] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7690–7699, 2020. 31, 32

[119] Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638, 2016. 7

[120] Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu. Speech recognition with augmented synthesized speech. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 996–1002. IEEE, 2019. 58

[121] Nick Rossenbach, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7069–7073. IEEE, 2020. 58

[122] 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*, 2015. 19, 25, 36, 55, 58

[123] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 38

[124] EJ Schlossmacher. An iterative technique for absolute deviations curve fitting. *Journal of the American Statistical Association*, 1973. 35

[125] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019. 55, 58

[126] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, 2018. 45, 46, 47, 50

[127] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, 2018. 7

[128] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. 58

[129] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. 31, 32, 38, 39

[130] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020. 21, 25, 26

[131] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 36

[132] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. 11

[133] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Proc. NeurIPS*, pages 4077–4087, 2017. 19, 20, 23, 24, 25

[134] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016. 5

[135] Guanglu Song, Biao Leng, Yu Liu, Congrui Hetang, and Shaofan Cai. Region-based quality estimation network for large-scale person re-identification. *AAAI*, 2018. 13, 14

[136] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2357–2366, 2019. 31, 38, 39

[137] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 55, 58

[138] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 19, 20

[139] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2052–2060, 2019. 38

[140] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2052–2060, 2019. 40

[141] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *European Conference on Computer Vision*, pages 717–734. Springer, 2020. 31, 32, 36, 38, 39

[142] 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. 25

[143] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. End-to-end feedback loss in speech chain framework via straight-through estimator. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6281–6285. IEEE, 2019. 58

[144] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 58

[145] Sean Vasquez and Mike Lewis. Melnet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083*, 2019. 46, 47

[146] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 46, 64

[147] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 59

[148] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 32

[149] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wier-

stra. Matching networks for one shot learning. In *Proc. NeurIPS*, pages 3637–3645, 2016. 19, 20, 23, 25

[150] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 25

[151] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014. 6, 7, 10

[152] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014. 1

[153] Jixuan Wang, Kuan-Chieh Wang, Marc T Law, Frank Rudzicz, and Michael Brudno. Centroid-based deep metric learning for speaker recognition. In *ICASSP*, pages 3652–3656. IEEE, 2019. 7

[154] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, and Neil M Robertson. Deep metric learning by online soft mining and class-aware attention. In *AAAI*, volume 33, pages 5361–5368, 2019. 14

[155] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proc. CVPR*, pages 7278–7286, 2018. 20

[156] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *ICML*, 2018. 46, 47, 50, 71

[157] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 38

[158] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018. 64

[159] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016. 6, 7, 10

[160] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE TPAMI*, 39(2):209–226, 2017. 5, 15

[161] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014. 1

[162] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing wasserstein distance. *arXiv preprint arXiv:1802.04307*, 2018. 11, 12, 70

[163] Junichi Yamagishi. English multi-speaker corpus for cstr voice cloning toolkit,. 2012. URL http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html. 50

[164] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4362–4371, 2017. 5

[165] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020. 19, 25, 26

[166] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *ICPR*, pages 34–39. IEEE, 2014. 5, 7

[167] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, pages 7115–7123, 2019. 21

[168] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *ICPR*, volume 4, pages 441–444. IEEE, 2006. 14

[169] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019. 50, 63

[170] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020. 19, 20, 22, 25, 26, 28

[171] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Proc. ICLR*, 2018. xii, 59, 61, 63

[172] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*, 2019. 58

[173] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, 2015. 38, 65

[174] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *ECCV*. Springer, 2016. 5, 13, 14

[175] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 58

[176] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3712, 2019. 66

[177] Zhen Zhou, Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4747–4756, 2017. 14

[178] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019. 31, 32, 36, 38, 39

[179] Guido Zuccon, Leif A Azzopardi, and CJ Van Rijsbergen. Semantic spaces: Measuring the distance between different subspaces. In *International Symposium on Quantum Interaction*, pages 225–236. Springer, 2009. 19