7-2-2021

An Exploration of Controlling the Content Learned by Deep Neural Networks

Liqun Yang
Florida International University, lyang028@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

AN EXPLORATION OF CONTROLLING THE CONTENT LEARNED BY DEEP NEURAL NETWORKS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Liqun Yang

2021

To: Dean John Volakis
College of Engineering and Computing

This dissertation, written by Liqun Yang, and entitled An Exploration of Controlling the Content Learned by Deep Neural Networks, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Deng Pan

Fahad Saeed

Ning Xie

Yuanchang Sun

Wei Zeng, Major Professor

Date of Defense: July 2, 2021

The dissertation of Liqun Yang is approved.

Dean John Volakis
College of Engineering and Computing

Andrés G. Gil
Vice President for Research and Economic Development and
Dean of the University Graduate School

Florida International University, 2021

DEDICATION

To my wife Xinxi Wang.

ACKNOWLEDGMENTS

Thanks, Dr. Wei Zeng, Dr. Bogdan Carbunar, Dr. Yijun Yang, Dr. Jason Liu and everyone who have helped me. I love you!

ABSTRACT OF THE DISSERTATION

AN EXPLORATION OF CONTROLLING THE CONTENT LEARNED BY DEEP NEURAL NETWORKS

by

Liqun Yang

Florida International University, 2021

Miami, Florida

Professor Wei Zeng, Major Professor

With the great success of the Deep Neural Network (DNN), how to get a trustworthy model attracts more and more attention. Generally, people intend to provide the raw data to the DNN directly in training. However, the entire training process is in a black box, in which the knowledge learned by the DNN is out of control. There are many risks inside. The most common one is overfitting. With the deepening of research on neural networks, additional and probably greater risks were discovered recently. The related research shows that unknown clues can hide in the training data because of the randomization of the data and the finite scale of the training data. Some of the clues build meaningless but explicit links between input data the output data called “shortcuts”. The DNN makes the decision based on these “shortcuts”. This phenomenon is also called “network cheating”. The knowledge of such ”shortcuts” learned by DNN ruins all the training and makes the performance of the DNN unreliable. Therefore, we need to control the raw data using in training. Here, we name the explicit raw data as “content” and the implicit logic learned by the DNN as “knowledge” in this dissertation.

By quantifying the information in DNN’s training, we find that the information learned by the network is much less than the information contained in the dataset. It indicates that it is unnecessary to train the neural network with all of the information, which means using partial information for training can also achieve a similar effect of using full information.

In other words, it is possible to control the content fed into the DNN, and this strategy shown in this study can reduce the risks (e.g., overfitting and shortcuts) mentioned above. Moreover, use reconstructed data (with partial information) to train the network can reduce the complexity of the network and accelerate the training. In this dissertation, we provide a pipeline to implement content control in DNN's training. We use a series of experiments to prove its feasibility in two applications. One is human brain anatomy structure analysis, and the other is human pose detection and classification.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

The Deep Neural Network (DNN) is one of the most significant achievements in machine learning. Its impressive performance and broad applicability make it widely used in various fields, such as Object detection [ZSGY19], data classification [LW07], natural language processing [OMK20], etc. The solutions of many hard questions based on DNN have shown their revolutionary effects. Applications based on artificial intelligence can be seen everywhere, which deeply influences the development of our society. As the application of DNN continues to mature, its theory has stagnated for a long time. We cannot give a trustworthy answer about the basis of the DNN to make such an accurate prediction. Like a black cloud under the blue sky, the unexplainable inner weights of DNN cast a veil of mystery on the neural network. Some pessimists even think the success of DNN just a meaningless carnival because it is just a black-box method. The day we open this black box is the day we find how ridiculous it is. Even it is not the comments to DNN from the mainstream, the thinking behind it is still alarming. If we can only input the knowledge to the DNN but cannot get enlightening knowledge from it and improve ourselves, the relationship between human and the machine would inevitably lead to the ending depicted in science horror fiction. In other words, we should concern what is learned by the DNN.

It is not a hypothetical question but the one we have met in practice. Overfitting is one of the most common phenomena in the DNN's training. The accuracy of the model decreases on the test dataset with the training processing. The reason is that the knowledge learned by the DNN from the training data is unrelated to the commonality of the entire data. It will cost extra computing resources and impact the performance of the model profoundly. In addition to the overfitting, Geirhos et al. describe another alarming phenomenon in

[GJM+20]. They find that the DNN can find some hidden clues in training data. These meaningless clues are more explicit and recognizable than the feature which we want the DNN to learn. For example, they provide a toy model to recognize the moon and star patterns in the picture. Artificially, stars were always shown in the top right or bottom left of an image; moons in the top left or bottom right. This pattern is still present in samples from the i.i.d. test set but not in o.o.d. test images, exposing the shortcut. As they expected, the model learned from the training is to use the location to recognize the pattern (star or moon) but not the feature of the pattern itself. These shortcuts will mislead the model and ruin the training. In some aspects, the DNN "cheats" in the test. Moreover, in practice, the data is collected simultaneously. The training data and the test data are just a division in some aspects, which means if the shortcut exists in the training data, the probability of its appearance in the test dataset is pretty high. It means the impact of shortcuts is hard to detect and visualized in most cases.

Therefore, controlling and visualize the content learned by the DNN is an essential and exciting topic. Unlike the artificial feature extraction, controlling the learned content (CLC) does not mean selecting the feature for the DNN artificially. Meanwhile, different from non-content controlling training, it does not provide entire information to the DNN. It is a compromise between these two training strategies. Briefly, it intends to find a mapping between the source and the feature. There are two requirements of this mapping.

1. Focusing on the feature of a certain abstract aspect, such as the shape of boundary, local texture, etc.
2. Being able to identify the source.

And this mapping is only used in training. The model trained by the content controlling strategy should accept the raw data directly, which is the biggest difference with the feature extraction. On the premise of making full use of the ability of DNN to extract

features from input data automatically, it reduces the quantity of input information. It thus reduces the risk of network cheating and overfitting. Moreover, it can accelerate the training and reduce the computing cost. Besides, by analyzing the effect of the feature, we can infer the behavioral logic of neural networks, which is meaningful for the interpretation of DNN.

1.2 Dissertation Organization

The dissertation is structured into four chapters. It is starting from the current chapters that outline the research scope, purpose, and contribution.

Chapter 2 introduces the motivation of this work. Specifically, we summarize the work related to network cheating and overfitting. These researches show the necessity to control the content learned by the DNN. Besides, we also provide an example to show the impact of network cheating in the real world. This case is found in a system designed for worker health monitoring system. This system helps the worker avoid labor injuries by monitoring the worker's pose and sending an alarm when the worker keeps the same pose for a long time. In the research, we found that this system intends to classify the human posture by the feature of clothes, which is not what we want it to learn.

Chapter 3 introduces the information quantification methods which are used to analyze the behavior of the DNN. Specifically, we summarize works related to information quantification. These methods help us to reveal the essence of the effects of DNN. The observing results show that compared with the information provided by the dataset, the information learned by the DNN is much less. There is profound information redundancy in the neural network training process, which implies that there is no necessity to provide the complete information of the dataset to the DNN. In other words, we can simplify or

abstract the data to a fingerprint-like form that contains less information. Still, we can identify the source approximately before sending it to the DNN.

Chapter 4 introduces two of our works about reducing the information quantity in the dataset. Specifically, there are two main kinds of data in computer vision, 3D models and 2D images. For the 3D model, we provide a successful case in human brain MRI analysis. We imply the conformal welding method to simply complex MRI image sequence to a vector. The model based on this feature shows competitive performance in IQ classification. In this case, the conformal welding signature curves play a significant role in reflecting the contour of each region in the human brain. The registration is necessary before the mapping. As a supplement, we provide a novel method to compute the diffeomorphic registration of 3D surfaces with point and curve feature landmarks. For 2D images, we provide a successful case in human pose estimation and detection. In this case, we abstract the images with workers to a series of silhouettes. This abstraction eliminates the DNN's cheating phenomenon mentioned in Chapter 2 without impacting the performance of the DNN.

Chapter 5 concludes this dissertation.

1.3 Contribution

For each chapter, this dissertation has the following novelties.

1. For Chapter 2 on the necessity of the content control, we provide evidence of DNN's cheating in a real system.
2. For the Chapter 3 on the visualize the content learned by the DNN, we provide two novel methods to quantify the task-related information learned by the DNN.

3. For the Chapter 4 on the content control in DNN's training, we provide two successful examples to prove the feasibility of our method for content control. Besides, we provide a novel geometry method that can be used in feature extraction for 3D mesh.

CHAPTER 2

NECESSITY OF CONTENT CONTROLLING IN TRAINING

2.1 Content Guide

In this chapter, we introduce the work about the necessity of content controlling in DNN's training. Without content control, there are two main potential risks in training, overfitting and network cheating.

Overtraining is a concept that we use too many resources to solve a simple machine learning task. When we use too much data or loop too many times in training, the most common result is overfitting. A modeling error in statistics occurs when a function is too closely aligned to a limited set of data points. As a result, the model is useful only to its initial dataset and not to any other data sets. Specifically, for DNN, it performs as the accuracy of the model on the training dataset increases, but the accuracy on the test dataset staying the same or decreasing. One of the widely accepted explanations of overfitting is that the model records or learns information related to a few samples' characteristics but unrelated to the commonalities. Therefore, if we can control the content discovered by the DNN before the overfitting, we can eliminate this risk effectively. When we use the too complex network to solve a simple task, we can observe the double-decent phenomenon. It shows that the error rate of the DNN decreases firstly, then increases, finally drops on the test dataset.

Network cheating is another risk in DNN's training. It usually performs that the DNN finds some "shortcuts" in the dataset. Based on these "shortcuts", the DNN can effectively complete the task. However, these "shortcuts" are always ridiculous and meaningless and not what we want the DNN to learn. The DNN's behavior based on these "shortcuts" is called network cheating. Because of the optimization logic, the impact of these "shortcuts"

is inevitable if they exist in the dataset, which will ruin all the training. We introduce the related work in the first section.

Case of network cheating. As a supplement, we introduce a case we find in the practice of network cheating in the third section. The network cheating is found in a system that is used to helps the worker avoid labor injuries by monitoring the worker's pose and sending an alarm when the worker keeps the same posture for a long time. The core of this system is a DNN which is used to recognize the pose of workers. However, in practice, we found that for the image of workers with special clothes, the accuracy of this DNN is lower than the other samples. We use a series of experiments to prove that it is a case of network cheating. Because the DNN is trained without content controlling, it learned using the cloth feature to classify the human pose. For more details, we introduce it in the third section.

In summary, content controlling is a suitable solution to reduce the risks mentioned above. Meanwhile, it can save computing resources and accelerate the training effectively.

2.2 Introduction to Overtraining

2.2.1 Overfitting

What is overfitting : There is a significant assumption that the input data and the output data are independent, identical distributed ($i.i.d$), which means for any two samples, they are unrelated but obey the same generation rule. If this pre-condition is not satisfied, there is no pattern which the model can learn, and the prediction based on the learned knowledge is impossible. For the success cases of machine learning, this pattern exists but unknown. We have to use the input data to fit, which is the core of machine learning and the reason for two main risks, underfitting and overfitting. Because the capacity of the

DNN is infinite theoretically, underfitting can be eliminated generally. However, reduce the risk of overfitting is one of the most important topics in machine learning.

What causes the overfitting : A widely accepted reason for overfitting is the incompleteness of the dataset. It can be understood from two aspects. First, the training data is incomplete [Die95, Wei94]. They could not reflect the pattern of their common features, and the training always leads the model to an optimal local status but not the optimal global one. Therefore, the model learns some characterized features, which impacts the model's performance on the test dataset. If we also use the test dataset in training, the overfitting could be eliminated, but it is not a reasonable solution. Second, the information provided by the data is incomplete. For example, we use a series of images of apples and pears to teach the model to recognize the "fruit". The model will likely go wrong when the input image is about grapes.

Simple solution of overfitting : Preliminary works show some solutions to overcome overfitting. According to whether it changes the network structure, these solutions can be divided into two parts, hyperparameter-based methods, training strategy-based methods. In the first one, regularization is one of the most popular methods [GJP95]. We can use a regularization term in loss function to balance the generalization and overfitting. This term is usually set as $\lambda W^T W$, where λ is a hyperparameter that controls the impact of this term, and W is the weight. These solutions have a common shortcoming that the hyperparameter cannot be calculated directly. In other words, it is heuristics, empirical and hard to transfer in most cases, which makes the work of AI engineers is much similar to the work of alchemists. In the second one, people intend to use some training strategies to reduce the risk of overfitting. Early stop [MB89] and learning rate decay are two of the most widely accepted ones. By stop the training before the overfitting, people control the learning process effectively. In [Pre98], Perchelt et al. analyze the early stop strategy and

provide an indicator to show the chance of early stop. However, as Perchelt et al. mention, the opportunity to stop training needs to be analyzed base on the model case by case, which means there is no exact mark to inform us to stop training. This conclusion is approved by the following research related to double-decent [NKB+19]. For the learning rate decay, it shows the same idea to control the learning process. Similarly, by reducing the learning rate, people can weaken even stop the network's learning. Although the mainstream open source for DNN, like Tensorflow, Pytorch, etc., widely use it in their functions, the latest work [SKYL17] shows that decaying the learning rate is not a good strategy. Increasing the batch size can achieve the same even better result.

Anyway, **from the solution of overfitting, we can see that no matter the early stop or the learning rate decay, they contain the simple ideas of content control.**

2.2.2 Double-decent

Discover of double-decent : This phenomenon is mentioned in [NKB+19] firstly when Nakkiran et al. analyzing the impact of network scales on the performance. Briefly, they found that the large-scale network is not always performing better than the smaller ones. They found that when they use a network that far exceeds the task requirements, the error rate ($1 - accuracy$) of the network's test dataset (simple task) is changed strangely. Unlike the standard curve of error rate (monotonous decline), the error rate decreases at the beginning, then increases, and finally decreases (decrease twice). One of the most widely accepted explanations is a double-decent curve, a special case of the multi-decent curve, usually related to a classic problem called Column Subset Selection (CSSP) or Row Subset Selection (RSSP).

Introduction to CSSP : CSSP [GE03, BMD09] is a classic problem in machine learning. It can be described as

Definition 2.2.1 *Given an $m \times n$ matrix A, pick a set $S \subseteq \{1, \ldots, n\}$ of k column indices, to minimize*

$$\mathrm{Er}_{\mathbf{A}}(S) := \|\mathbf{A} - \mathbf{P}_S \mathbf{A}\|_F^2$$

where $\|\cdot\|F$ is the Frobenius norm, P_S is the projection onto span $\{a_i : i \in S\}$ and a_i denotes the ith column of A.

Explicitly, suppose matrix A is a series of vector, and each of them corresponds to a pillar. All the pillar prop up a space together. Now we want to select k pillars who can prop up a max space (minimize the space shrinking). The approximation factor $\frac{\mathbb{E}[\mathrm{Er}_{\mathbf{A}}(S)]}{\mathrm{OPT}_k}$ are used to evaluate the result, where OPT_k is defined as

$$\mathrm{OPT}_k := \min_{\mathbf{B}:\mathrm{rank}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_F^2 \leqslant \min_{S:|S|=k} \mathrm{Er}_{\mathbf{A}}(S)$$

With the k increasing, the curve of approximation factor shows multiple peeks, called multiple-decent curve [DKM20]. Another variant of the CSSP emerges in the kernel setting under the name Nystrom method [WS01, DMC05, GM13]. The essence of them is similar. Combined with the double-decent problem, the data feature can be viewed as the "pillar" that prop up the data space. We can view the accuracy as another form of approximation factor. CSSP has reflected the simple thought of content control. Its idea can be expanded to select the most valuable data from the dataset to avoid the impact of the noise and reduce the cost of training. The difference is that in CSSP, the original matrix A is invariant. However, in the content controlling, we intend to reduce the information quantity of the data, which will be introduced in the following chapter.

In summary, from the explanation of the double-decent phenomenon, we can see the importance of content control.

2.3 Introduction to Network Cheating

Networks' cheating : In most cases, the trained network's performance is the only measure in the evaluation of the training process. However, as the author mentioned in [GJM+20], some of the information in the dataset can be the "shortcut" to complete tasks, which leads to the training's failure. The model uses a simple "shortcut" to complete a complex learning task called Clever Hans Effect. This effect was discovered on a performing horse, Hans, in Berlin in the late 19th and early 20th centuries celebrated for demonstrating remarkable intelligence. Hans was said to have been taught to add, subtract, multiply, divide, work with fractions, tell time, keep track of the calendar, differentiate between musical tones, and read, spell, and understand German. However, after a formal investigation in 1907, psychologist Oskar Pfungst demonstrated that the horse was not performing these mental tasks but was watching the reactions of his trainer. It is the same as what the "cheating" DNNs do. Essentially, they do not understand the task anymore, but they can complete the task based on the hints hidden in the data.

In practice, this phenomenon has been observed in the early version of BERT [DCLT18] when it completes the argument reasoning comprehension task. In [NK19], the author shows that the model does not understand the task. It makes the judgment just based on the statistical feature of the dataset. In [GRM+18], the author points out that in some cases, the CNN-based model classifies the image just depending on the texture unexpectedly. In other words, the CNN is misled by the training dataset, it does find the best way to complete the task, but that is not what we want it to learn. In [LBL+18], the author points out that the unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the datasets. In the examples above, the network makes the correct judgment based on a hidden trick but not the logic we want it to learn. Even though we can visualize some of the knowledge learned by networks as

[ZCS+18, ZWW+19, ZLM+20, ZWC+20] show, we still cannot impact their learning. It is one of the motivations of this research.

Therefore, to eliminate the "shortcut" hidden in the dataset, **control the information provided by the training data is the best option.**

2.3.1 Case: Network cheating in human pose detection

Because the learning process is out of control, the network's cheating (Clever Hans Effect) is inevitable. This section demonstrates an example of a network's cheating behavior to show the importance of eliminating the "shortcut" hidden in the dataset.

Introduction of the model : To ensure workers' health and avoid labor injury caused by long working hours, networks for human pose detection are wildly used to monitor worker movements. Based on Mask-RCNN [HGDG17], we design a network to monitor the pose of humans to prevent work-related musculoskeletal disorders. [SRDB01] shows a study of occupational mobility in a cohort of construction workers. It shows that disorders of the back and spine are one of the major causes of early retirement. Therefore, this model is designed to recognize three primary poses related to workers' back and crest on the construction site, standing, bending, and squatting (see [Y+]). Because there are only three kinds of the pose, to simplify the computing, we do not use the network to extract human skeleton information but identify the region with a human directly, as Fig. 2.1 shows. VGG Image Annotator (VIA) [DZ19] is used to mark the label.

Clues of the network's cheating : The model is trained by the real data (see Fig. 4.18(a)), and its mAP (IoU = 50%) can get to 89% on the similar data (test dataset of workers). However, when we use another test case of workers, the mAP decreases to 71% rapidly, caused by the accuracy decrease in classification but not the detection. This phenomenon

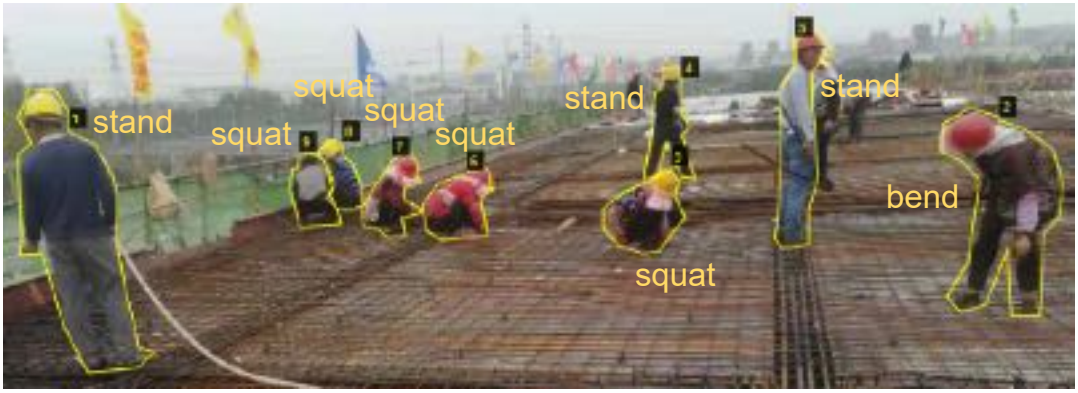


Figure 2.1: Labels of the dataset. The region of human are marked in the image ,and we provide corresponding label for every region based on the human pose.

(a) Quality supervisor (b) Concrete pouring worker

Figure 2.2: Two typical works on the construction site, quality supervisor and concrete pouring worker.

catches our attention. After analyzing the dataset, we find some clues about the network's cheating. On the construction site, workers with different types of work are usually with different kinds of cloth. In Fig. 2.2, we present two typical works on the construction site, quality supervisor and concrete pouring worker. The former wear usual clothes for the clothing, and the latter is required to wear the special vest for their safety. Meanwhile, the former usually stands and plays the role of a supervisor in most cases. Moreover, the latter often need to bend or squat because of their work. **There is a "shortcut", classifying the pose of humans by their cloth.**

Figure 2.3: Images of designed “bend” samples.

Figure 2.4: Test cases used to detect network cheating.

To verify our hypothesis, we create a special dataset based on the original dataset. In the training dataset, we replace all the “bend” samples with the same amount of designed images as Fig. 2.3 shows. Unlike the original images, the worker is always with an orange fluorescent vest and yellow helmet in this group of pictures.

After training, we use it to identify another group of designed images (240 images) as Fig. 2.4 shows. In this group, the worker with the same equipment, but all their pose is “squat”.

As a result, there are 63% samples are classified as “bend”, 30% are detected as “squat”, 3% are classified as “stand”, and 4% are not detected. This result indicates that **to get a trustworthy model based on a neural network, control what is learned by the network is necessary.** However, there is no effective way for us to control the learning process of the network. Controlling the information provided by the dataset is the last way we can choose. In the following part of this dissertation, we demonstrate our work in this aspect.

2.4 Conclusion

In this section, we introduce the necessities of content control and provide an example of network cheating in practice. Over-training risks and network cheating risks are two main reasons for content control. To avoid wasting computing resources and being misled by "shortcuts", we need to control the content learned by the DNN. Before controlling the content, we need to visualize the knowledge and quantify the information learned by the DNN. This part of the work is introduced in the next section.

CHAPTER 3

VISUALIZE THE CONTENT LEARNED BY DNN

3.1 Content Guide

Before control the content learned by DNNs, we need to understand what is learned by them. In this chapter, we introduce the work related to knowledge visualization.

Knowledge Visualization and Quantification. In the first section, we summarize the related work in this field. Based on the measuring object, methods in this field can be divided into three parts, weights-based methods and output-based methods. The former focuses on the weights of DNN and intends to visualize or quantify the knowledge hidden in the weights. The latter focuses on the output of the DNN and wants to visualize or quantify the impact of DNN. The weights-based methods are usually tightly related to the Interpretation & Explanation of AI (XAI). Their research object is tightly associated with the architecture of the network. Compared with the former, the output-based method intends to view the DNN as a black box. Therefore, the former makes more contributions to the XAI, and the latter has broader applicability. We introduce them in the first section.

Case 1: Quantify the Information Learned by the DNN based on Euclidean Distance of the weights In the second section, we provide a novel weights-based method to quantify the information learned by the DNN. Specifically, our work has the following contributions:

1. Provide a metric, the expectation of the Euclidean distance between the initial weights and the output weights, to quantify the information learned by the DNN in training.

2. Provide the corresponding derivation with a reasonable conjecture from information gain in training to the expectation of distance between the weights before and after training.

For the first one, to prove its effectiveness, we test it on seven kinds of models. After fixing the training configuration of the model, we use datasets with different amounts of labels to train them and measure the distance between the weights before and after training repeatedly. The result shows that in the training process with more data, the expected distance is longer than the distance in the training process with fewer data. For the second one, we provide an observation result about visualization of the weights before and after training based on MDS. Then, we make a conjecture and build a geometry model to estimate the probability distribution of the appearance of weights. Finally, we provide a novel QMCM that uses the shortest distance from element to a set to reflect the ratio of the volume between this set and the universe, which builds the connection between the expectation of the distance and information gain.

Case 2: Rethink the Knowledge-distillation In the third section, we provide another method to quantify the information related to the task in the DNN's training. Based on this method, we review the knowledge distillation phenomenon and find that knowledge distillation is not a knowledge transform process but a knowledge weakening process. By weakening the knowledge learned from the dataset, the DNN enhances its generalization capability, which improves its performance on the test dataset effectively. In the end, we use an experiment to verify our discovery. We use a Gaussian noise to replace the output of the teacher network (fake teacher). Compared with the model trained by the real teacher, the model trained by the fake teacher shows almost the same performance on the train and test dataset. It means the output of the teacher network is meaningless. Or strictly, its impact is the same as the noise.

3.2 Introduction of Knowledge Visualization

3.2.1 Introduction

A series of significant improvements for the neural network's training based on the information theory, e.g., cross-entropy loss function [DBKMR05], achieved great success in the past decades. There has been a growing interest in understanding deep neural networks (DNNs) mapping and training using information theory [TZ15, AS18, TMS17]. Meanwhile, more and more works point out that the metrics, the accuracy, and the loss of the DNN, cannot evaluate the model effectively [NK19, GRM$^+$18]. The risk of the DNN's "cheating" [GJM$^+$20] makes people intend to provide other metrics to evaluate and analyze the performance of the DNN and reveal its essence.

There have been many works on analyzing the DNN's training information. The DNN is viewed as a function with parameters (weights) in some of them, like [TPB00, AS18, TZ15]. People intend to analyze its impact on the data to reveal the essence of the training process. For more details, Tishby et al. use the mutual information and the entropy of the DNN's output, called Information Bottleneck Lagrangian, to analyze the behavior of it [TPB00]. Similar method are mentioned in [AS18, TZ15, AFDM16, HMP$^+$16, YWJP20, SZT17]. However, this kind of methods may generate unrepresentative results because of the the variety of input data and statistical error [SBD$^+$19]. The other category views the weights of DNNs as variables and intends to analyze its changing direction to analyze the training process like [FNJN19, WZC$^+$18, Xu18]. Specifically, Fort et al. use the stiffness of the DNN to diagnose the generalization of it, where the stiffness is defined as the expected cosine value, and its sign between two backpropagation vectors [FNJN19]. Similarly, the clever score, which is defined based on the p-norm of the backpropagation results, is used to estimate the robustness of the DNN [WZC$^+$18], and Xu et al. use the Fourier analysis to explain the generalization of DNNs learned by stochastic gradient descent (SGD) [Xu18].

With different analyzing methods, we can evaluate the DNN from different aspects. The quantity of information learned by the neural network is an attractive one. It is meaningful to the interpretability of DNNs. Achille et al. point out that the information learned by the DNN is stored in its weights [APS19]. They define the weights as the Kullback-Leibler (KL) divergence between the weights probability distributions before the training and after training and provide a computable form based on the Fisher Information. That work prompts the research in this field but also has its limitations. The most significant one is that the method is complex and hard to implement. Generally, the weights can identify the DNN's status uniquely in the training process and its updating processes in a high-dimensional Euclidean space. The difference between the weights can be represented as the Euclidean distance between them intuitively. Then, there is a question, can we quantify the information of the network using the Euclidean distance between the weights? Previous works provide related hints [GIP+18, FMLD18]. Especially, Garipov et al. mention an interesting phenomenon in the experiment of [GIP+18]. They find that the Euclidean distance between the weights after the same training steps is similar for the different training processes. Even though they do not provide a related explanation, it is still inspiring to us.

3.2.2 Related works

Information bottleneck : Tishby et al. use the mutual information to analyze the changing of the information quantity in the neural network and put up with the concept called "information bottleneck" (IB) [TPB00]. The information bottleneck theory describes the neural network's behavior and defines the optimal target, preserving the relevant information about another variable (maximize the bottleneck). One step forward, [TZ15], Tishby et al. develop this method and puts it up with a tool, information plain, to vi-

sualize the neural network's behavior. Series of methods related to it are put forward [AFDM16, HMP+16, YWJP20, APS19]. However, statistical errors cannot be eliminated. Compared with the scale of the domain of the possible networks' input, the scale of the test case is too small, which will further magnify statistical errors. Specifically, let $I(X;\tilde{X})$ be the mutual information between the input data X and the compressed representation (like the output of one layer) $\tilde{X}$, defined as:

$$I(X;\tilde{X}) = \sum_{x\in X}\sum_{\dot{x}\in\tilde{X}} p(x,\tilde{x}) \log\left[\frac{p(\tilde{x}\mid x)}{p(\tilde{x})}\right]. \tag{3.1}$$

Based on Eq. (3.1), we need to know the joint distribution $P(X,\tilde{X})$ and the distribution $P(\tilde{X})$, which can only be estimated in the experiment. Theoretically, $\tilde{X}$ is in a continuous space for the neural network. Discretization is necessary to count the probability distributions mentioned above. Different discretization functions will impact the result of observation significantly. Moreover, $\tilde{X}$ is tightly related to the layer's activation function based on the definition, and the activation function's feature will impact the observed result and mislead us. Shwartz-Ziv et al. explain one behavior of the network [SZT17]. The experiment result indicates that the network's goal is to optimize the IB trade-off between compression and prediction, successively, for each layer. However, the related conclusion is challenged by other researchers. Saxe et al. prove that these two phases are just a special case caused by the non-linear activation function [SBD+19]. In a word, based on analyzing the relationship between input and output, the results will be impacted by the experiment's bias. Whereas directly analyzing the weights of the neural network can avoid the errors mentioned above.

Information in the weights : Achille et al. use the Fisher information to quantify the information stored in convolutional neural networks (CNN) by analyzing the weights before and after perturbation [APS19]. They mention that the weights w, if they add a

perturbation $w' \leftarrow w + \delta w$, it does not impact the loss ($L_{\mathcal{D}}(p_{w'}) \approx L_{\mathcal{D}}(p_w)$), they can make a decision that such weights contain "no information". Conversely, suppose slight changes were to yield a large increase in loss. In that case, one could say that such weights are very "informative" and store them with high precision. However, the freedom of perturbation makes the result of this method unstable. Therefore, they introduce the other two parameters, an arbitrary choice of code "pre-distribution" P, chosen before seeing the dataset, and a measure of coding length, represented by a "post-distribution" Q, chosen after training, to limit them. Their derivation is based on the assumption $P(w) \sim N(0, \lambda^2 I)$ and $Q(w \mid \mathcal{D}) \sim N(w^*, \Sigma)$, where w^* is any local minimizer of the cross-entropy loss obtained with any optimization algorithm. However, they do not provide enough evidence to support this assumption. Even so, their work is very inspiring to us. They use the Fisher information to quantify the information contained by the weights w about the training process $\mathcal{D}$. However, updating the weights of a network usually processes in a high-dimensional Euclidean space. The changing of $\mathrm{KL}(Q\|P)$ is not easy to obtain without a series of approximations and assumptions. Therefore, we intend to prove if the Euclidean distance can be used as the metric directly for information quantification for the BP-based training.

DNN training and Euclidean distance between weights : Flennerhag et al. provide a method to minimizes the expected length of this path from initialization to final weights [FMLD18]. They analyze the weights changing with the geometry view. Their method implies that the Euclidean distance between transfer origin and the target is positively correlated to the quantity of information in the transfer. However, they do not provide exact proof. Garipov et al. find that the optima of the complex loss functions are connected by simple curves over which training and test accuracy are nearly constant [GIP$^+$18]. Based on this geometric insight, they provide an ensembling method entitled Fast Geometric

Ensembling (FGE) to train high-performing ensembles in a shorter time. In the experiment, they find that for the different training processes, the Euclidean distance between the weights in different training stages is similar. It implies that some unknown stable factors determine this distance. However, they do not provide a reasonable explanation for this interesting phenomenon.

3.3 Case 1: Quantify the Information by the Euclidean Distance

3.3.1 Introduction

In the following parts of this section, we introduce our method of quantifying the information learned by the DNN in training:

1. We show and explain how we view the weights of the DNN, which is the fundamental of our derivation.
2. We show an observed result of the weights visualization, based on which we make a significant conjecture.
3. We derivative the representation of information gain above and reveal the link between the Euclidean distance of the weights and the information gain of the training.

3.3.2 Quantify the Information about a Variable

As Shannon mentioned [Sha48], information can be thought of as the resolution of uncertainty. Generally, we can use $H(X)$ to represent the chaos of the variable X, $H(X) = \mathrm{E}[I(X)]$. An increase in information or energy will lead to a decrease in system

entropy in the view of physics. Oppositely, if the entropy decrease, usually called information gain, is quantified, we can quantify the quantity of information accumulated by the system in this process:

$$IG(X, Y) = H(X) - H(X|Y),$$

where Y is the unknown factor that impacts the chaos of the variable X. If we view the DNN as a system, the information gain from it in training is equal to the status entropy reduction. Specifically, if we use W to represent the status of the DNN and $\mathcal{D}$ as the training process, the information gain can be written as

$$IG(W, \mathcal{D}) = H(W) - H(W|\mathcal{D}). \tag{3.2}$$

The difficulty in calculating Eq. (3.2) is to calculate the entropy of the system.

The Entropy of DNN

About how to quantify the entropy of DNN, we explain it from three aspects.

- The weights identify the status of the system (DNN) uniquely, which means the entropy of the weights is equal to the entropy of the model.
- The weights are unpredictable, which means the appearance of specific weights is a random event with probability in training.
- The domain of the weights is finite, which means the entropy of the weights is measurable.

The weights identify the status of the DNN uniquely : For a specific neural network architecture, the status of one neural network can be identified by its weights uniquely. Defiantly, for the same input, the outputs of two DNNs with the same weights are the same. The entropy of the weights is equal to the entropy of the system's status, and the weights can be viewed as the coordinates of the DNNs status.

The weights are unpredictable : Our viewpoint is the weights of the network are variables in the training process. The appearance of a specific value of the weights is uncertain because of the random factors in the training process, like the randomized initialization, optimizer (e.g., SGD), order of training data, etc. A specific training process is that in which there are no random factors. And it can be viewed as a particular case of the uncertain ones. There is another view to understanding the uncertainty of the weights. Generally, the calculation in the layer (including the activation function) is irreversible, which gives the capability of generalization to the network. Otherwise, we can use the output to recover the input for a network with specific weights, which means the relationship between the input data X and its corresponding compress representation T is bijective. The network degenerates to a codebook of input X. Therefore, for the external observer (only the network's input and output are visible), the network's weights cannot be calculated, which is the same as we cannot make sure the status of the Erwin Schrödinger's Cat (see [MZ97]). The weights' values are hidden in an unknown wave function, like the cat's status is unknown after closing the box. Once we open the black box and observe the weights, the wave function collapses into a constant (see [VN18]), which is the same as taking a sample from the current wave function. In our opinion, superficially, the training changes the weights; essentially, it changes the probability density function (PDF) of the variable W. If the training process is effective, the probabilities of the weights with better performances increase. In this aspect, we can represent Eq. (3.2) as the following form,

$$IG(W, \mathcal{D}) = H_{p_0}(W) - H_p(W), \tag{3.3}$$

p_0 is the PDF of W without effective training (trained by the ideal noise), and p is the PDF after training.

The domain of the weights is finite : Theoretically, the domain of W, denoted as Ω, equals to $\mathbb{R}^n$. However, because of the initialization strategy, the limitation of computer

precision, and the finite training steps, $|\Omega| \ll |\mathbb{R}^n|$, and in most cases, it is finite. For example, for a training process $\mathcal{D}$, suppose $||\delta w||_2 \leq l$, where δw is the result for once backpropagation, and there are k steps in the training, for one initial value w_i, the corresponding space is an n-sphere S_i whose center is w_i and radius is kl. And $\Omega_{\mathcal{D}} = \bigcup S_i$. For a specific training process, $\Omega_{\mathcal{D}}$ is a constant approximately. In this work, we only discuss the case with finite reachable space U, which equals the set of the possible weights after the training process in which the training data is the ideal noise (contains no information). For some special cases, like life-long learning (see [PKP+19]), whose training steps are infinite, the error of the estimation based on our method will be out of control. Therefore, in our work, we just discuss the finite training processes. Intuitively, all the elements in Ω can appear at the end with the same probability without training. Formally, we have

$$H_{p_0}(X) = \log(|supp(p_0)|). \tag{3.4}$$

Now, the question reduces to **how to estimate the entropy of the weights after training**.

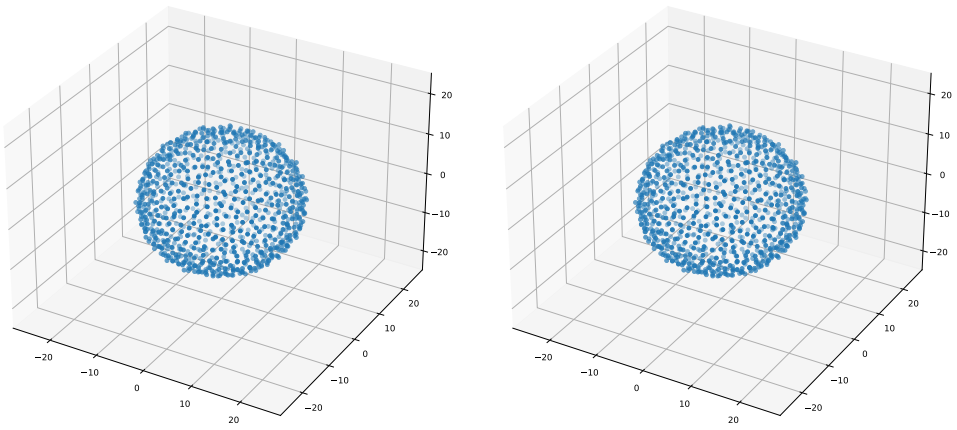
Exploration of the distribution of the weights after training

In this section, we mainly discuss the weights probability distribution after training. Firstly, we discuss the lower bound of $H_p(W)$, if it can be zero or not. Secondly, we use MDS to visualize the distribution of the weights and make a conjecture as Conj. 3.3.1 shows. Finally, based on Conj. 3.3.1, we get Eq. (3.6) and provide corresponding explanation.

Lower bound of the weights entropy after training : Theoretically, the training of the neural network is an optimization problem but the target of the optimization (loss function) is not convex as Fig. 3.2 shows. There is an significant lemma in convex optimization, a high-dimensional convex function can be equivalent to the superposition of countless one-dimensional convex functions. Formally, $f : \mathbb{R}^n \to \mathbb{R}$ is convex iff $g : \mathbb{R} \to \mathbb{R}, g(t) =$

$f(x+tv), \operatorname{dom} g = \{x+tv \in \operatorname{dom} f\}$ is convex for any $x \in \operatorname{dom} f, \quad v \in \mathbb{R}^n$. For the neural network, it is easy to find a point $(x, y) \in \mathbb{R}^n \times \mathbb{R}$ and a vector $\overrightarrow{v}$ s.t. g is not convex in this direction, which means the loss function f is not convex. Therefore, there are many local optimum points in the optimization of neural network. Even the global optimum point (loss equals to zero) is not unique either in common cases as mentioned [SLL$^+$20]. It indicates that the size of support of p, denoted as $supp(p)$, is greater than one, and $H_p(W)$ is greater than zero generally.

The weights distribution visualization : Therefore, we need to analyze the distribution of the weights after training. The MDS is a method used to translate "information about the pairwise 'distances' among a set of n objects or individuals" into a configuration of n points mapped into an abstract Cartesian space (see [Mea92]). The most important feature of the MDS is that it can keep the Euclidean distance between samples after dimensional reduction. The details of the experiment are shown in the experiment section. As Fig. 3.1



(a) (b)

Figure 3.1: The left shows the MDS result of the weights at the beginning. The right shows the MDS result of the weights at the end.

shows, the mapping points are distributed in the space uniformly after training, which

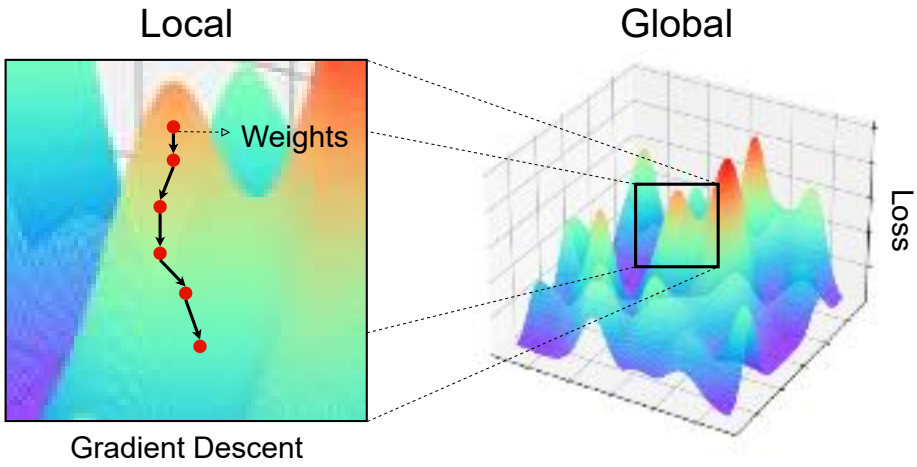


Figure 3.2: The landscape (see [LXT+17]) of a toy model shows that the optimization of neural network based on gradient decent is not a convex problem globally.

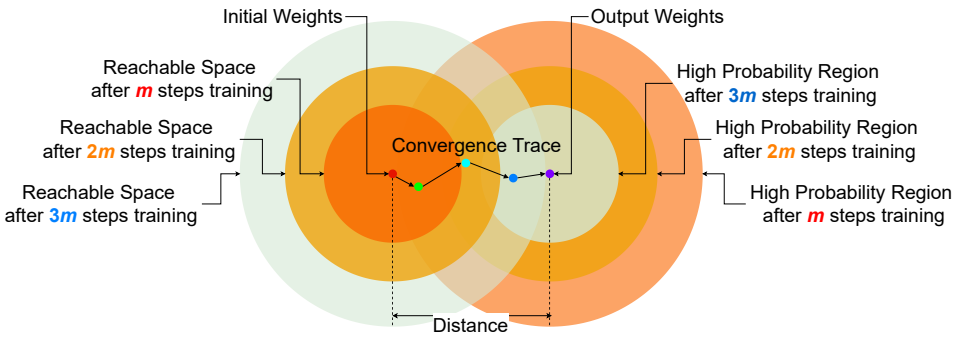


Figure 3.3: The information gain related to the training can be measured by the scale shrinking of the high probability region.

means in the high-dimensional space before mapping, the distribution of the original points are also an approximate uniform distribution. There is no obvious center of the weights. The appearance probability at the end of the training is the same for the weights. Based on the weights distribution visualization results, we make the following conjecture, as Conj. 3.3.1 shows.

Conjecture 3.3.1 *For a randomized training process, all the weights have the same appearance probability if they can appear.*

Inference based on Conj. 3.3.1

Directly, based on Conj. 3.3.1, we have

$$H_p(X) = \log(|supp(p)|). \tag{3.5}$$

Therefore, we have

$$IG = \log(\frac{|supp(p_0)|}{|supp(p)|}). \tag{3.6}$$

We use Fig. 3.3 to explain the conclusion of Eq. (3.6). With the training processed, the reachable space of the weights is extended for a specific initial weight. At the same time, the region of weights with a high appearing probability after training is shrinking. And the information gain related to the training can be measured by the scale shrinking of the high probability region (HPR). The intersection of the reachable space and the corresponding HPR is the region of the next weights that might appear, as the convergence trace shows. The distance between the initial weights and output weights can be viewed as the difference of HPR radius before and after training for output weights. Globally, for a specific training process, the training's impact on the HPR can be represented as Fig. 3.4 shows. Based on Conj. 3.3.1, for weights w, its probability of appearing in HPR can be measured by the volume of the union of a series of n-sphere as Fig. 3.4 shows. We can represent as the probability mass function (pmf) of this distribution

$$P(w \in HPR_{W^*,r}) = \frac{V_{HPR}}{V_U} \tag{3.7}$$

$$\approx \frac{V(\bigcup_{c \in W^*} Sphere_{c,r})}{V(\bigcup_{c \in W^*} Sphere_{c,r+d})}, \tag{3.8}$$

where W^* is the set of the optimized weights, $Sphere_{c,r}$ is the n-sphere whose center is c and radius is r, U is the reachable space.

Because the reachable space is hard to estimate, the exact IG of a training process cannot be calculated exactly. However, for two training process $\mathcal{D}_1$ and $\mathcal{D}_2$, suppose their

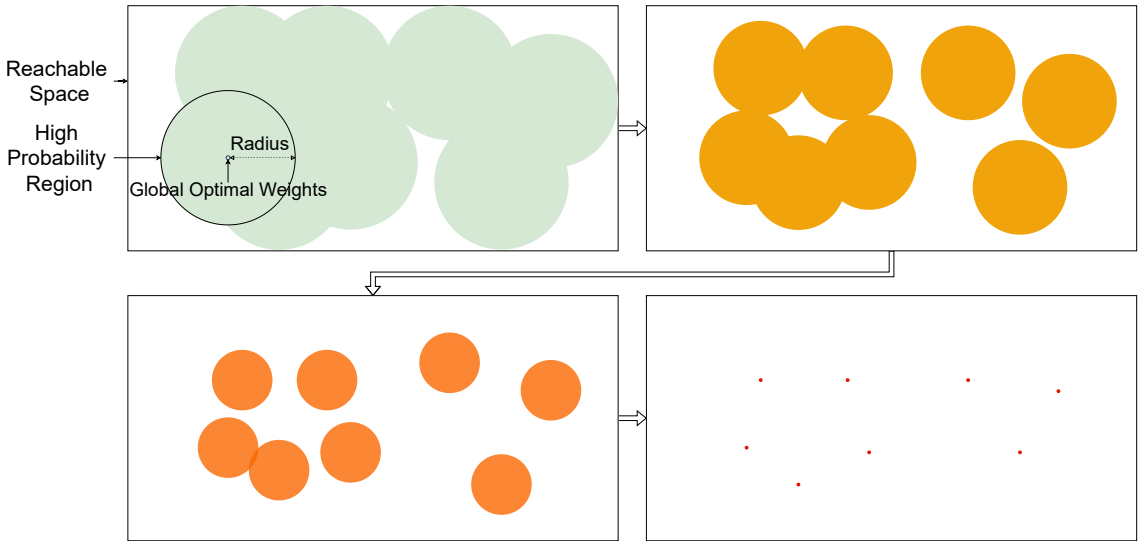


Figure 3.4: From the global view, the high probability region is shrinking as the training continues.

reachable space is the same, we can compare their information gain. Letting $\alpha_{X,X'} := \frac{|X|}{|X'|}$, called shrinking coefficient, $\alpha_{\mathcal{D}} := \alpha_{supp(p_0),supp(p_{\mathcal{D}})}$, we have

$$\alpha_{\mathcal{D}_1} < \alpha_{\mathcal{D}_2} \Leftrightarrow IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}. \tag{3.9}$$

And we have

$$\begin{aligned} \Delta IG_{\mathcal{D}_1,\mathcal{D}_2} &= IG_{\mathcal{D}_1} - IG_{\mathcal{D}_2} \\ &= \log(\frac{\alpha_{\mathcal{D}_1}}{\alpha_{\mathcal{D}_2}}) \\ &= \log(\frac{|supp(p_{\mathcal{D}_2})|}{|supp(p_{\mathcal{D}_1})|}), \end{aligned}$$

which is unrelated to the reachable space. If we can define an origin, we can measure the information gain of one training process. Now, the question is reduced to **how to estimate the shrinking coefficient α for once training.**

3.3.3 Quasi-Monte Carlo Method to Estimate the Set Shrinking Coefficient

This section mainly explains the relationship between the shrinking coefficient and the Euclidean distance of the weights. Statistically, the gradient-based BP optimization intends to find the nearest optimal solution. Based on this characteristic of DNNs' training, we provide a computable QMCM to estimate the shrinking coefficient and reveal the link between the expectation of the Euclidean distance and the shrinking coefficient.

Theoretically, this distribution can be estimated by the Monte Carlo method (see [KBTB14]) (MCM). However, to test if w is in the current HPR, we need to calculate its loss. Considering the scale of space, the cost of multiple randomized sampling is too high. To solve this problem, we provide a QMCM to provide another metric to estimate the shrinking coefficient. Briefly, we use the expectation of the shortest distance from elements to the set as the metric to measure the set scale changing.

Specifically, we define

$$d_{X'}(x) := \inf_{x' \in X'} ||(x, x')||_2^2,$$

as the distance between x and its closest element in X' as Fig. 3.5 shows, and we define $d_{X'}(x) = 0$ if $x \in X'$. Letting $X' = G$ and $x = w(w \in U)$, we notice that

$$P(d_{W^*,w} < r^2) \Leftrightarrow P(w \in HPR_{W^*,r}).$$

Therefore, we provide a new quasi-Monte Carlo method (QMCM) to estimate the scale differences between two sets X and X' when $X' \subset X$. Briefly, QMCM uses the expectation of the element-wised shortest distance to estimate the shrinking rate α^{-1}, and we show the derivation below.

Specifically, the distance from elements in X' to X' is set to 0. Letting $D_{X'}(X)$ denote the sum of shortest distance for $x \in X$, we have

$$D_{X'}(X) := \sum_{x \in X} d_{X'}(x).$$
$$\bar{d}_{X',X} := E_{x \in X}(d_{x'}(x)) = \frac{D_{X'}(X)}{|X|}$$

For specific set X, $|X|$ is a constant. And if $X'' \subset X'$, we have Eq. (3.10). The corresponding proof are shown in the Appendix.

$$\alpha_{X,X''} < \alpha_{X,X'} \Leftrightarrow \bar{d}_{X'',X} < \bar{d}_{X',X}. \tag{3.10}$$

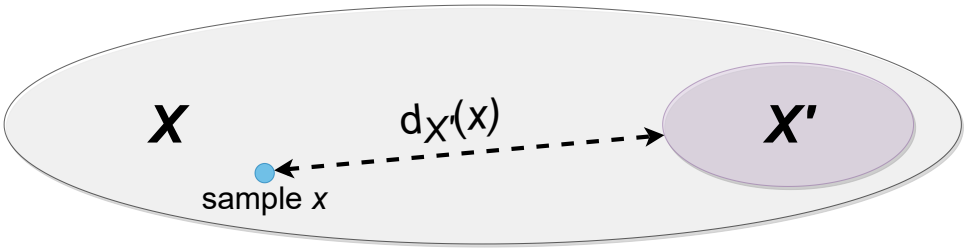


Figure 3.5: $d_{X'}(x)$ is the distance between x and X'

Denoted the support of function $d_{X'}$ as $supp(d)$ and the expectation of $d_{X'}$ on X as $\bar{d}(X)$. Based on the assumption,

$$|supp(d)| = (1 - \alpha^{-1})|X|,$$

we have

$$\bar{d}(supp(d)) = (1 - \alpha^{-1})\bar{d}(X)$$

For a specific X', α^{-1} is a constant. When $\alpha^{-1} \ll 1$, we have

$$\bar{d}(supp(d)) \approx \bar{d}(X).$$

For $\bar{d}(supp(d))$, we can use repeated sampling to estimate it. Denoted the set of sampling as Y, $\hat{d} := \bar{d}(Y)$, if we have two training process $\mathcal{D}_1$ and $\mathcal{D}_2$, we have

$$\hat{d}_{\mathcal{D}_1} < \hat{d}_{\mathcal{D}_2} \Leftrightarrow IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}. \tag{3.11}$$

If $\hat{d}_{\mathcal{D}_1} < \hat{d}_{\mathcal{D}_2}$, we have $IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}$. Oppositely, if $IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}$, we have $\hat{d}_{\mathcal{D}_1} < \hat{d}_{\mathcal{D}_2}$, which can be used to verify the correctness of our method.

Geometrically, $\bar{d}$ is the difference of HPR radius with and without effective training. Basically, we have

$$\alpha_{supp(p_0),supp(p_{\mathcal{D}})} = \frac{V_U}{V_{HPR}}.$$

Suppose the impact of the overlap among these n-spheres can be ignored, we have

$$\alpha \approx \frac{(r+d)^n}{(r)^n}. \tag{3.12}$$

Because $\alpha^{-1} \ll 1$, which means $r \ll d$. we have

$$\alpha \approx (\frac{d}{r})^n. \tag{3.13}$$

Therefore, we have

$$IG \approx n\log(d) - n\log(r). \tag{3.14}$$

Because $r \ll d$, therefore, the impact of the second part of Eq. (3.14) can be ignored. The information gain of a training process is positively correlated to the dimension of weights and the logarithm of the distance between the weights before and after training. It fits our basic understanding of this variable intuitively.

3.3.4 Experiments

The Weights distribution visualization

To visualize the distribution of the weights before and after training, we use the same training configuration to repeatedly train a specific network and collect the input weights

(randomized) and output weights. Then, we use MDS to visualizing the similarity of individual weights. The classic MDS algorithm (see [Wic03]) is shown in Alg. 1.

Algorithm 1 Multidimensional Scaling

1: Set up the squared proximity matrix $D^{(2)} = \left[d_{ij}^2\right]$, where d_{ij} is the Euclidean distance between i^{th} and j^{th} elements.
2: Apply double centering (see [Mar96]):
$B = -\frac{1}{2}JD^{(2)}J$ using the centering matrix $J = I - \frac{1}{n}11'$, where nn is the number of objects, 1 being an $N1$ column vector of all ones.
3: Determine the m largest eigenvalues $\lambda_1, \lambda_2, ..., \lambda_m$ and corresponding eigenvectors $e_1, e_2, ..., e_m$ of B(where m is the number of dimensions desired for the output).
4: Now, $X = E_m\Lambda_m^{1/2}$, where E_m is the matrix of m eigenvectors and Λ_m is the diagonal matrix of m eigenvalues of B.

Specifically, in this experiment, we use a TensorFlow CNN Demo (see [Ten20a]) to identify the images in CIFAR-10 (see [KH$^+$09b]). We fix all the super parameters in the training process and train the network from different scratches repeatedly. The experiment process is shown in Exp. 1 and the training configuration is shown in Tab. 3.1.

Algorithm 1 Input and output weights distribution by randomized training from scratches

1: Fix the training configuration, including all super parameters.
2: Initialize 1000 networks randomly and save their initial value of weights.
3: Train the networks to fine-tuned and save their weights at the end of the training.
4: Use the MDS algorithm to visualize the input and output weights.

As Fig. 3.1(a) and Fig 3.1(b) show, all the points are distributed on the spherical surface uniformly. It means there is no point with special meaning and indicates their source vectors are also distributed uniformly in the corresponding high-dimensional space.

As the reference, we add a constraint that limits the initial weights into a small range $\{w_0', w_0"\}$. Then train the network repeatedly. The experiment process is shown in Exp. 2. We use the MDS algorithm to reduce the dimension of the initial weights and the output weights together. The output is shown in Fig. 3.6. It shows that the output weights' mapping points (shown as the blue points) are distributed near their initial weights' mapping points

(shown as the red points), which means the weights have a higher appearance probability if its mapping point is in the region with more points. We can infer that if the mapping points distribute uniformly in a region, their source's appearance probability is similar. Based on the experiments mentioned above, we have Conj. 3.3.1.

Algorithm 2 Input and output Weights distribution by randomized training from 2 specific scratches

1: Fix the training configuration, including all super parameters.
2: Initialize 200 networks. Half of them use w_0' as the initial weights, and the others use the w_0''.
3: Train the networks to fine-tuned and save their weights at the end of the training.
4: Use the MDS algorithm 4 to visualize the input and output weights.

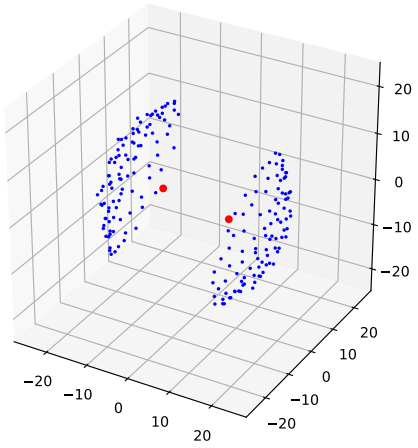


Figure 3.6: The initial weights are mapped to these two red points, and the output weights are mapped to these blue points.

Experiments of Information Quantification

As mentioned in Eq. (3.11), we have $I_{\mathcal{D}_1} < I_{\mathcal{D}_2} \Rightarrow \hat{d}_{\mathcal{D}_1} < \hat{d}_{\mathcal{D}_2}$, which can be used to verify the correctness of our method. We can construct two training processes $\mathcal{D}_1$ and $\mathcal{D}_2$ such that $IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}$. Then calculate $\hat{d}_{\mathcal{D}_1}$ and $\hat{d}_{\mathcal{D}_2}$ respectively.

Model		TF CNN	TF MNIST	GoogleNet	ResNet	VGG	AlexNet	Yolo v3
Dataset		CIFAR-10	MNIST	TF Flower	TF Flower	TF Flower	TF Flower	Pascal
Super params	LR	0.01	0.01	0.01	0.01d	0.01d	0.01d	0.01d
	Epoch	10	10	20	40	20	20	20
	Batch Size	128	128	32	32	32	32	64
	Opt	Stochastic Gradient Descent						
Pre-process	Norm	Yes						
	Mean Sub	No		Yes	Yes	Yes	Yes	No
	Rescale	Yes		Yes	Yes	Yes	Yes	Yes
	ST	Yes		No	No	No	No	Yes

Table 3.1: Training configuration of experiments.

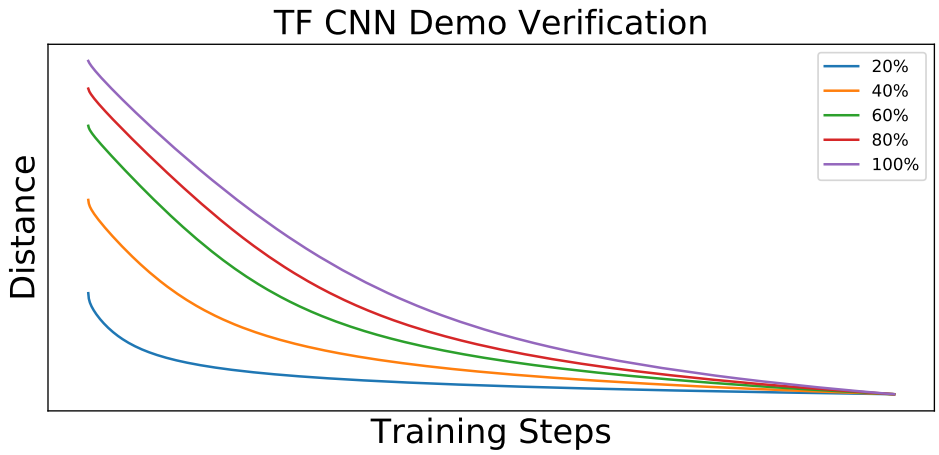
We need to construct these two training processes $\mathcal{D}_1$ and $\mathcal{D}_2$. If we control all the other random factors and make $\mathcal{D}_2$ contain more kinds of samples than $\mathcal{D}_1$, we have $IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}$. Specifically, we use the same models to verify our method (see Tab. 3.1).

Algorithm 3 Training with different amount of labels

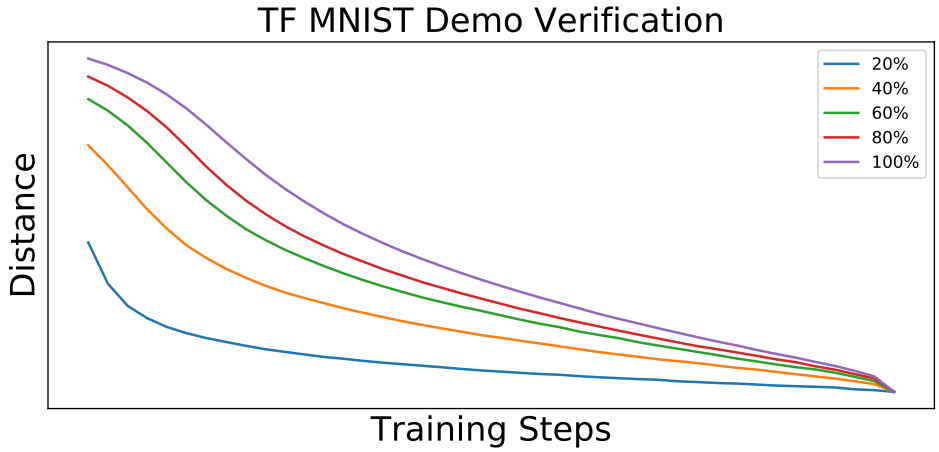
1: Fix the training configuration, including all super parameters.
2: Initialize 5 networks $\{m_i\}(i \in [0, 4])$.
3: Use data with $(20 \times i)\%$ labels to train the network separately and record the weights changing.
4: Calculate $d(w, w')$ of each weight, where w is the current weights, w' is the output weights of the same training process.

Wang et al. prove that few samples can also be used to guide the network to complete a complex task [WYKN20]. Because of that, to ensure $IG_{\mathcal{D}_1} < IG_{\mathcal{D}_2}$, we control the numbers of labels. There are $[20\%, 40\%, 60\%, 80\%, 100\%]$ kinds of samples used in the training. To eliminate the impact of the weights update numbers, we train the same steps in all the model training processes. Finally, we calculate the $\hat{d_{\mathcal{D}_i}}$ for each training process (see Exp. 3).

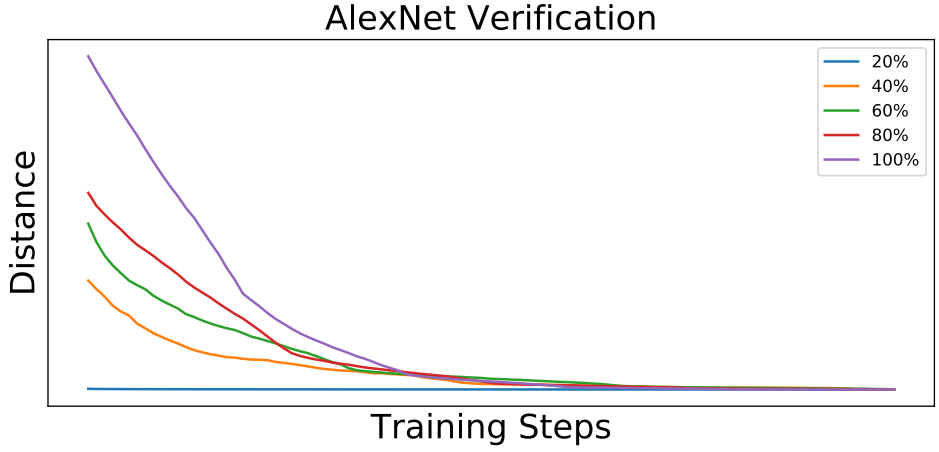
As Fig. 3.7, 3.8,3.9 shows, in all experiments, as the IG increases, $\hat{d_{t_i}}$ increases.



(a) TF CNN Demo

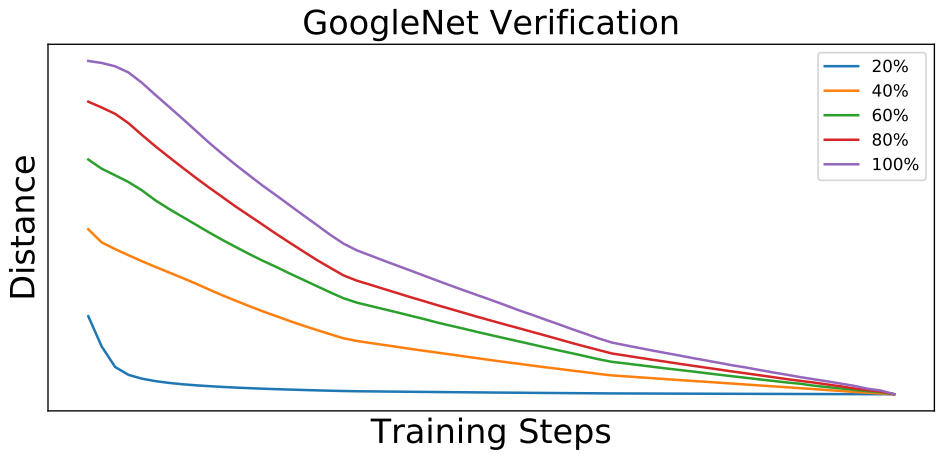


(b) TF MNIST Demo

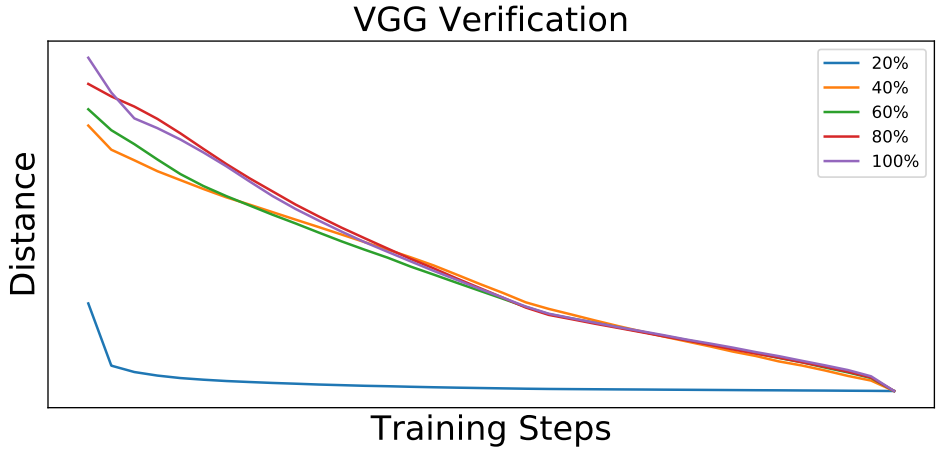


(c) AlexNet

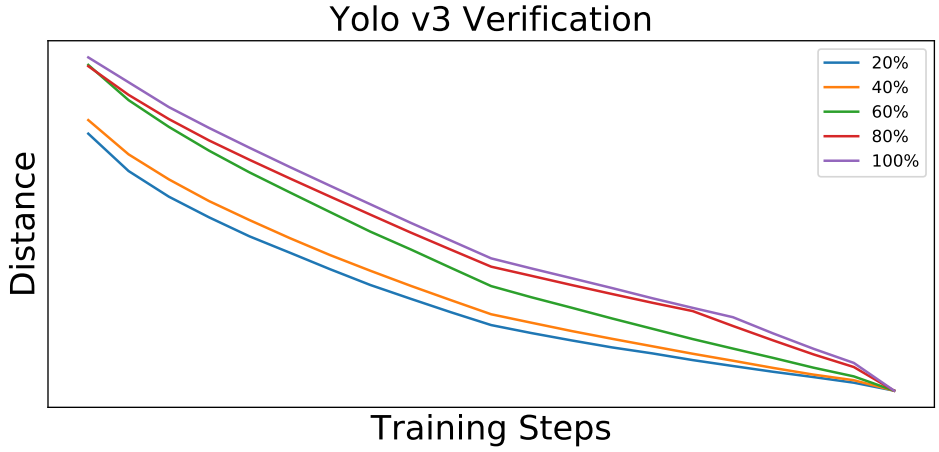
Figure 3.7: The Exp. 3 results of simple networks (AlexNet, TF MNIST Demo, TF CNN Demo).



(a) GoogleNet



(b) VGG



(c) Yolo

Figure 3.8: The Exp. 3 results of complex networks (VGG, GoogleNet. Yolo).

Errors in the estimation

The method in this work is an estimation of the information quantity learned by the neural network in training. As an estimation, the error is inevitable. There are two main approximations in our method.

- Using the output weights in the same training process to approximate the closest weights in HPR.
- Assuming that for a successful training process, $\alpha^{-1} \ll 1$

For the first one, we provide a corresponding method to reduce the error. For the second ones, we use experiments to prove that our approximation is reasonable.

Shortest distance estimation : For the network's training process, we have its initial weights and the output weights. The condition to implement QMCM to estimate the information gain is that the output weight is the closest one of the initial weight in $supp(p)$, which means the output weight is the nearest optimal solution based on the research. This conclusion has been shown in many works [Hay10, A$^+$18, HZXH19]. However, some strategies implementation, such as SGD and others with similar methods ideas [DHS11, KB14], make the answer not always stable. Therefore, we use experiments to verify it in our verification.

We select seven network models from simple to complex to verify this, TensorFlow MNIST classification Demo (see [Ten20b]), TensorFlow CNN Demo (see [Ten20a]), GoogleNet (see [SLJ$^+$15]), AlexNet (see [KSH17]), ResNet (see[HZRS16]), VGG (see [SZ14]) and Yolo v3 (see[RF18]). To ensure that these networks are used in scenarios that adapt to them, we select four datasets with different input scale and complexity, MNIST (see [LBBH98]), CIFAR-10 (see [KH$^+$09b]), TensorFlow Flowers (see [Tea19]) and Pascal VOC (see [EVGW$^+$]).

Network	Mean	STD	$c_v(\%)$
TF CNN Demo	4.026	0.053	1.316
TF MNIST Demo	1.358	0.031	2.282
GoogleNet	28.084	0.482	1.718
ResNet	8785.243	1172.836	13.350
VGG	0.556	0.012	2.158
AlexNet	4.849	0.589	12.163
Yolo v3	26.863	6.286	23.400

Table 3.2: The statistic result (mean value, standard deviation and coefficient of variation) about the distance between initial and end states of neural networks.

We train each kind of model from scratch to fine-tune it with the same configuration 1000 times repeatedly. The basic information of the training is shown in Tab. 3.1, where d after learning rate means the learning rate decay is used in training, ST means the standardization is used in training. And then, we calculate the distance of arbitrary pairs of initial and end states. The result shows that all the output weight is the closest one of the initial weights in $supp(p)$.

Moreover, we calculate the mean, standard deviation, and coefficient of variation (c_v) value of the distance (see Tab. 3.2). It shows that although the difference in the mean value among models is big, the coefficient of variation is still low, reflecting the stability of this estimation.

Shrinking rate exploration : The estimation of the shrinking rate can also cause the error theoretically. In our common sense, the global optimal point and its neighbors have a higher probability than the local ones if all of them could appear in the final output. However, based on the theory mentioned by [APS19], if the global optimal point is the one with maximum information, the impact of perturbation to it is also the maximum. It indicates that in the Euclidean space, the neighbors' performances of the global optimization are not as well as expected. In other words, the radius of the neighbor with good performance is far shorter than our intuition. We use a fine-tuned toy model's weight w to prove this.

Specifically, we add an disturb g to the w, and then, test the performance of w' ($w' := w + g$). Theoretically, w' is a sample point in the high dimensional sphere (*n*-sphere) whose center is w and radius is R_n. In [Ste09], Stewart et al. prove that for a *n*-sphere,

$$V_n = \frac{2\pi R_n^2}{n} V_{n-2},$$

where V_n is the volume, R_n is the radium, n is the dimensions. And we have

$$\lim_{n\to\infty} \frac{V_n}{R_n} = 0,$$

which means the part of the *n*-sphere close to the surface occupies most of the volume. Therefore, the sampling in the n-sphere is approximated to the sampling on its surface.

Without loss of generality, in the experiment, g follows *i.i.d.* Gaussian distribution,

$$g \sim \mathcal{N}\left(0, \Sigma = \mathrm{diag}\left(\sigma_1^2, \ldots, \sigma_n^2\right)\right),$$

If

$$\sigma_0 = \sigma_1 = \ldots = \sigma_n = \sigma,$$

we have

$$||g||_2^2 \sim \Gamma(0.5n, 2\sigma^2),$$

whose expectation is

$$E[||g||_2^2] = n\sigma^2,$$

variance is

$$Var[||g||_2^2] = 2n\sigma^4,$$

and coefficient of variation is

$$c_v[||g||_2^2] = \sqrt{\frac{2}{n}} \tag{3.15}$$

Let $\sigma_i \in [0, 0.2]$, $R \in [0, 0.2\sqrt{n}]$ with c_v less than $\sqrt{\frac{2}{n}}$. In the experiment, for each σ, we generate 100 w' and calculate the average of their accuracy and loss function.

As Fig. 3.10 shows, the average accuracy drops rapidly with the increase of σ, and the average loss increases at the same time. Specifically, when $\sigma > 0.025$, the model's performance is impacted; When $\sigma > 0.10$, the model's performance is not different from the initial weights, which means the weight is completely overwhelmed by noise. It means the volume of HPR is far less than the volume of reachable space, and we have $\alpha^{-1} \ll 1$.

Proof of the relationship between α and $\bar{d}$: The main difference between our QMCM and the traditional MCM is that we use the shortest distance d from one element to a set to observe the ratio changing between the set and the universe (α). Therefore, we need to prove that the expectation of d is a monotonically increasing function of α. For this proposition, if we can prove that for the minimum increase of α, $\bar{d}$ will increase at the same time, and any increment can be expressed by the minimum increment with the same form, we can prove it. **Proof.** Letting X'' be the union of set X' and $\{x''\}$ ($X'' := X' \cup \{x''\}$), we have

$$\alpha_{X,X''} < \alpha_{X,X'},$$

and can ensure that,

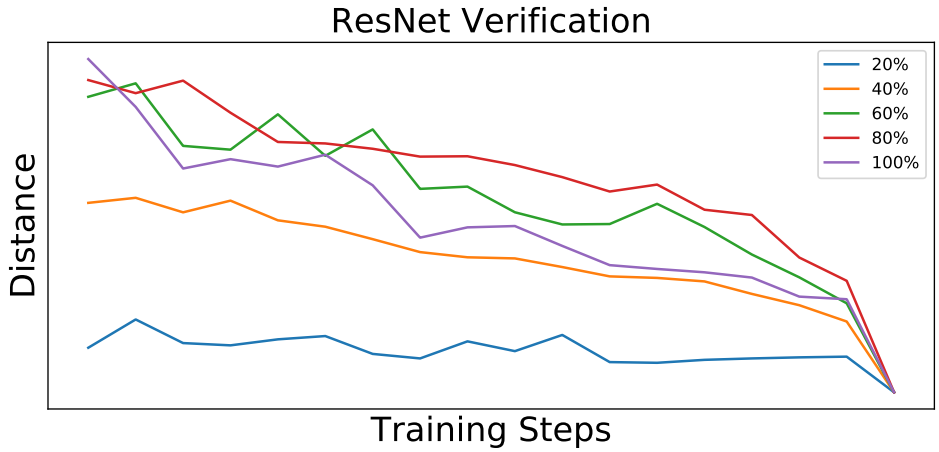
$$\nexists(X''' \subset X) \wedge (X' \subset X'''), s.t.$$
$$\alpha_{X,X'} - \alpha_{X,X'''} < \alpha_{X,X'} - \alpha_{X,X''}.$$

$d_{X'}(x'')$ is always positive, we have

$$D_{X'}(X) - d_{X'}(x'') < D_{X'}(X),$$

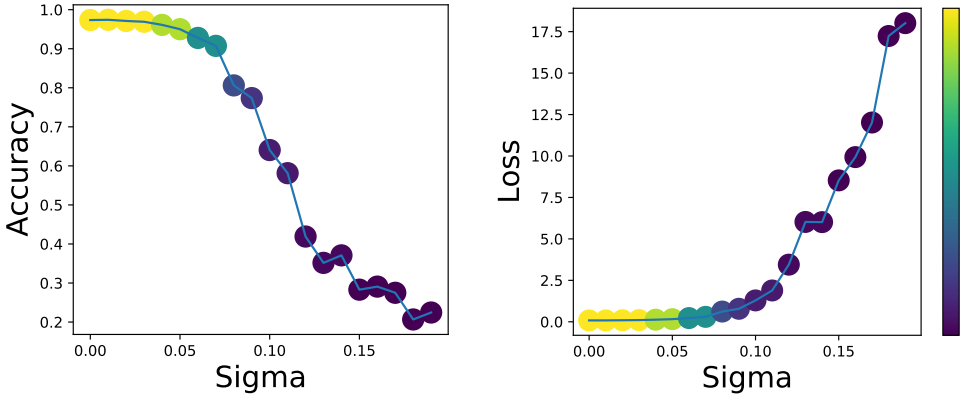
which equals to

$$D_{X'}(X - \{x''\}) < D_{X'}(X).$$



(a) ResNet

Figure 3.9: The Exp. 3 results of ResNet.



(a) Accuracy (b) Loss

Figure 3.10: The left one shows the changing of the model's accuracy with the σ, and the right one shows the changing of the model's loss with the σ. The color of the point shows the training steps of the model who performs similarly.

Adding a new element x'' to X' will update the distance from some elements in X to X', noted as $\sum \Delta d$, and we have

$$\sum \Delta d := \sum_{x \in X} (d_{X'}(x) - d_{X''}(x))$$
$$D_{X''}(X) = D_{X'}(X - \{x''\}) - \sum \Delta d.$$

Based on the definition, we have

$$\sum \Delta d \geq 0.$$

Therefore, we have

$$D_{X'}(X - \{x''\}) - \sum \Delta d \leq D_{X'}(X - \{x''\}),$$

and we have

$$D_{X''}(X) \leq D_{X'}(X - \{x''\}),$$

which leads to

$$D_{X''}(X) < D_{X'}(X).$$

And we have

$$\frac{D_{X''}(X)}{|X|} < \frac{D_{X'}(X)}{|X|},$$

$$\bar{d}_{X'',X} < \bar{d}_{X',X}.$$

And we have

$$\alpha_{X,X''} < \alpha_{X,X'} \Rightarrow \bar{d}_{X'',X} < \bar{d}_{X',X}.$$

The proof of other side is similar, and we have

$$\alpha_{X,X''} < \alpha_{X,X'} \Leftrightarrow \bar{d}_{X'',X} < \bar{d}_{X',X} \square$$

3.3.5 Discussion

Questions about the connection between information quantity and accuracy : The quantity of information is tightly related to the performance of the network. To guarantee the relationship between them is positive, the precondition is that the information learned by the network is useful. However, in practice, this precondition is not always satisfied. The best counterexample is the overfitting phenomenon, which shows that the network performs well on the training dataset but performs badly on the test dataset. The network learns extra knowledge from the training dataset, not the commonality between the training dataset and the test dataset. It indicates that not all the information learned by the network is useful and helpful. Therefore, in some special cases, high information quantity does not mean high accuracy or low loss.

The Question about cross-modal verification : We use four datasets from simple to complex (MNIST, CIFAR-10, TensorFlow Flowers, Pascal VOC) in the experiment and use seven matching models to receive the information from these four (TF MNIST Demo, TF CNN Demo, AlexNet, VGG, ResNet, GoogleNet, Yolo). Intuitively, we believe that complex networks are more potent in learning, which means learning more in training. People may ask why the experiment results do not verify this guess. Above all, our method is derived with the precondition, which is the DNN's structure is fixed. For different kinds of DNNs, the comparing of the results of our approach is meaningless.

Summary : In this work, we prove that the expectation of the Euclidean distance between the weights before and after training reflects the network's information gain in the training process. Based on the observed result, we make a conjecture that the probability of weights distributes uniformly on its support. It is easy to get the support's volume shrinking is related to the information gain tightly. Finally, we provide a QMCM to estimate this

volume shrinking and derive a measurable form of Shannon Information [Sha48] for the training process, which will prompt the research about the interpretation of the DNN.

3.4 Case 2: Rethink the Knowledge Distillation

3.4.1 Introduction

As a successful training strategy, the knowledge distillation [HVD15] (see Fig. 3.11) is proven to be effective in many works (see [RBK$^+$14, FLT$^+$18, YJBK17, CRCZ20]). Generally, it describes the process which transfers knowledge from one deep neural network (DNN), namely the teacher network, to another DNN, namely the student network (see [GYMT20, WY21]). As an effective method, knowledge distillation can improve the performance of the student network. Based on the changing of the teacher network, there are two kinds of knowledge distillation, online methods such as [ZWB$^+$20, GWW$^+$20b, CMW$^+$20] and offline methods such as [HVD15, RBK$^+$14]. The former changes the teacher network while training the student, and the latter does not. Based on the number of teachers, there are two main kinds of knowledge distillation, single teacher such as [HVD15, RBK$^+$14, WZLT18] and multiple teachers such as [YXXT17, PK19, YSP$^+$20, WC20]. From the teacher's knowledge form, there are two main kinds of methods, learning from *logits* such as [HVD15, CCCY18, ZZY19] and learning from intermediate layers' feature such as [RBK$^+$14, ZK16, JPW$^+$19]. The former uses the softened labels and regularization as the knowledge from the teacher. The knowledge origin of the latter has a more flexible form which can be one of the layer's outputs such as [RBK$^+$14, AHD$^+$19, XRLG20, WFL$^+$] and also can be pre-processed feature such as [ALZ$^+$20, CMZ$^+$20]. Based on the structure of the teacher, there are three main kinds of methods, learning from the teacher, learning from the peer, or self-learning. The first one intends to use a more

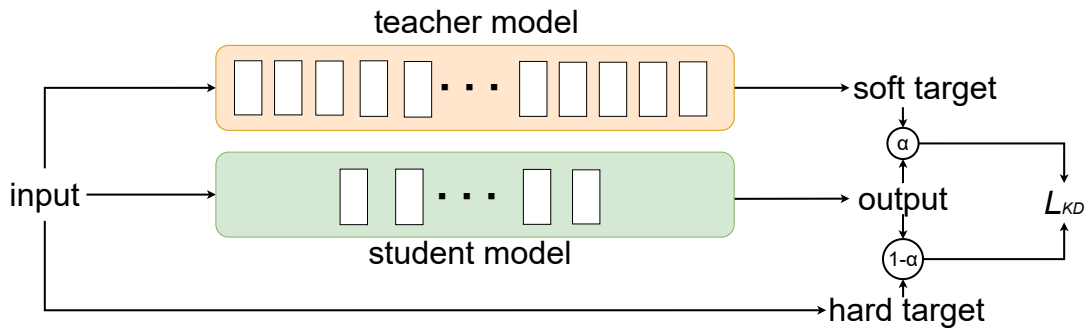


Figure 3.11: The knowledge distillation instruction.

complex network as the teacher, and the second intends to use a network with a structure similar to the student (see [ZXHL18, CMW+20]). The third intends to use the structure same as the student, which is usually related to some kinds of online methods such as [YTL+19, ZSG+19, YXSY19].

Generally, based on the theory mentioned in [HVD15], knowledge distillation is a process in which the teacher network transfers its knowledge to the student network. The teacher network has a higher capability to process the information, which allows it provides purified (compressed) knowledge to the student. Therefore, the teacher needs to be bigger than the student theoretically. However, this conclusion is challenged by Furlane et al. in [FLT+18]. In that work, they provide a model called Born-Again Network (BAN). They use a fine-tuned network with the same architecture as the student to be the teacher. This knowledge distillation process successes and the student's performance is better than the teacher at the end, which means the teacher's capability needs not to be bigger than the student's. One step further, in [YTL+19], Yuan et al. prove that the teacher needs not to be fine-tuned in the knowledge distillation. Now that the teacher network needs not to be powerful and fine-tuned, what can the student network learn from it? In other words, what is the essence of knowledge distillation?

In [CRCZ20], Cheng et al. use the concepts visualizing method to analyze the effect of the knowledge distillation. They find that the knowledge distillation process has three main influences on the student network, including letting the student learn more visual concepts, learning visual concepts simultaneously, and making the optimization direction stable. In [LPBSV15], Lopez et al. interpret knowledge distillation as a form of learning with privileged information. In [FLT$^+$18], the knowledge distillation is explained as some of the knowledge transfer processes.

Since the booming of knowledge distillation, most research had focused on distilling large networks into shallower, wider ones, who could ignore the importance of networks' depth. In [RBK$^+$14], Romero et al. provide a training strategy. In this method, the teacher can use *hint*, the output of the intermediate layer, to guide a student network during the training process. It makes full use of intermediate information on the teacher [RBK$^+$14].

$$\mathcal{L}_{HT}\left(\mathbf{W}_{\text{Guided}}, \mathbf{W}_{\mathbf{r}}\right) = \frac{1}{2}\left\|u_h\left(\mathbf{x}; \mathbf{W}_{\text{Hint}}\right) - r\left(v_g\left(\mathbf{x}; \mathbf{W}_{\text{Guided}}\right); \mathbf{W}_{\mathbf{r}}\right)\right\|^2 \tag{3.16}$$

where u_h and v_g are the subnet of teacher/student up to their respective hint/guided layers, $\mathbf{W}_{\text{Hint}}$ and $\mathbf{W}_{\text{Guided}}$ are its weight parameters, r is the regressor function on top of the guided layer with parameters W_r. After guided intermediate layer of student model, teacher network continue to distill student by Eq. (3.17),

$$\mathcal{L}_{KD} = (1-\alpha)\mathcal{H}\left(\mathbf{y}_{\text{true}}, P_{\text{St}}\right) + \alpha\mathcal{H}\left(P_{\text{Te}}^{\tau}, P_{\text{St}}^{\tau}\right) \tag{3.17}$$

where y_{ture} means ground-truth label, α is a balance weight, $\mathcal{H}$ refers to cross-entropy and $P_{\text{St}}^{\tau}/P_{\text{Te}}^{\tau}$ is the soft-target from student/teacher model based on the distillation temperature τ shown as Eq. (3.18).

$$P_{\text{Te}}^{\tau} = \text{softmax}\left(\frac{\mathbf{a}_T}{\tau}\right), \quad P_{\text{St}}^{\tau} = \text{softmax}\left(\frac{\mathbf{a}_S}{\tau}\right) \tag{3.18}$$

where $\mathbf{a}_T$ and $\mathbf{a}_S$ is the vector of teacher/student pre-softmax activations. It uses the weights of the intermediate layer as the hint to guide the students learning in the first stage. However, Romero et al. do not provide enough evidence to explain the essence of this loss

function as Eq. (3.16) shows. Additionally, because of the freedom of this loss function, the output lakes of robustness, and we can not use the first stage distillation to complete the training directly. It implies that compared with the first stage, the second stage is the key to knowledge distillation.

In [YTL+19], Yuan et al. proved that a bad-trained teacher can still improve the student. Based on these experimental results, Yuan et al. propose a Teacher-Free Knowledge Distillation (Tf-KD) framework. Specifically, Tf-KD can be broken down into two aspects. One is called self-training, denote as Tf-KD$_{self}$, and the other is Label Smoothing Regulation (LSR), denote as Tf-KD$_{reg}$. Different from "self-training" in [ZSG+19], in which knowledge from the "future" of the network is distilled into its "past," Tf-KD$_{self}$ training network by using knowledge from "another self." In short, it means training a network in a normal way and then use this pre-trained network as the teacher to train itself by using Eq. (3.17). In addition, Yuan et al. proved that knowledge distillation is a special kind of label smoothing regularization. Specifically, let $p(k)$ is the logit of model *S* and $q(k)$ is the ground-truth label, the the smoothed label distribution $q'(k)$ formulated as

$$q'(k) = (1-\alpha)q(k) + \alpha u(k) \tag{3.19}$$

where $u(k) = 1/K$ is a uniform distribution.

Thus, the loss function of label smoothing can be written as

$$\mathcal{L}_{LS} = (1-\alpha)\mathcal{H}(q,p) + \alpha D_{KL}(u,p) \tag{3.20}$$

where D_{KL} is the Kullback-Leibler divergence (KL divergence). On the other hand, Eq. (3.17) can be written as:

$$L_{KD} = (1-\alpha)\mathcal{H}\left(\mathbf{y}_{\text{true}}, P_{\text{St}}\right) + \alpha(D_{KL}\left(P^{\tau}_{\text{Te}}, P^{\tau}_{\text{St}}\right) + \mathcal{H}(P^{\tau}_{\text{Te}})) \tag{3.21}$$

where $\mathcal{H}(P^{\tau}_{\text{Te}})$ is a constant for a fixed teacher model, so the final result is:

$$L_{KD} = (1-\alpha)\mathcal{H}\left(\mathbf{y}_{\text{true}}, P_{\text{St}}\right) + \alpha D_{KL}\left(P^{\tau}_{\text{Te}}, P^{\tau}_{\text{St}}\right) \tag{3.22}$$

Combining Eq. 3.22 and Eq. 3.20, it is easy to get a conclusion that KD is a special case of LSR. It implies that there might be no knowledge transfer in the knowledge distillation. In our work, by measuring the task-related information learned by the model, our observed results prove this conclusion from other aspects.

For the effect of the knowledge distillation, in [CRCZ20], Cheng et al. use visual concepts extracted from intermediate layers of a DNN to explain knowledge distillation. It proved experiments to show that DNN learned more visual concepts from knowledge distillation than learning from raw data based on the method mentioned in [GWZ+19, MZZZ19]. The most important hint in that dissertation is knowledge distillation ensures that DNN learns several visual concepts simultaneously with stabler optimization directions. Based on the information bottleneck theory [TPB00, SZT17], the training of the network has two phrases, empirical error minimization and information compression. In the first stage, the individual characteristics of data may impact the training profoundly. The convergence direction is unstable in most cases. The knowledge distillation can improve the training stability, which implies that the feature of commonality is enhanced. It helps us to explain why knowledge distillation can improve the performance of the network.

The gap between teacher and student is one of the most significant factors in the quality of KD. In [MFLG19], Mirzadeh et al. use 4, 6, 8, and 10 layers CNN as the teacher to train a 2-layer CNN to demonstrate that the student achievement does not increase with the complexity of the teacher, but rather like a quadratic curve, which goes up and then goes down. To remedy the problem brought by the gap, Mirzadeh et al. propose a method named Teacher Assistant Knowledge Distillation (TSKD), which means introducing an assistant model with medium complexity between the teacher and the student. The knowledge can distill from the teacher to the assistant and then from the assistant to the student. The problem is solved by breaking the whole into parts. In [GWW20a], Gao et al. use another method to solve this problem. Different from [MFLG19], two distillations are independent

of each other, the assistant A in this dissertation learn the residual error between the teacher T and the student S at the same time, so the A of RKD (Residual error based Knowledge Distillation) can get information from both of the T and S. In our dissertation, for describing the results of our experiment and comparing task-related information in students with different teachers better, the architecture of the teacher models and the student model we used is similar with [MFLG19].

Contribution : Unlike the previous work, we use the Shannon Information to analyze the changing of entropy of the DNN's prediction results before and after the knowledge distillation and make the following contributions.

1. Provide a metric, called task-related information (TI), to quantify the task-related information learned by the network in the training process.
2. Reveal that the knowledge distillation is not a knowledge transfer process but a knowledge reduction process.

3.4.2 Quantify the Information Related Task

Based on the information theory [Sha48, Qui86], the information can be viewed as the reduction of the uncertainty, we have

$$IG_t = H_0 - H, \tag{3.23}$$

IG_t represents the information gain of training related to the task t, H_0 is the task-related entropy of the DNN before training, and H is the entropy after training. For the event of prediction, it can be defined by a tuple $(x, y, \hat{y})$, where x is the input data, y is the ground truth, and $\hat{y}$ is the prediction result. Therefore, the entropy H can be defined as

$$H = -\sum_{x \in X} \sum_{y \in Y} \sum_{\hat{y} \in \hat{Y}} p(x, y, \hat{y}) \log p(x, y, \hat{y}), \tag{3.24}$$

where X is the set of all possible input, Y is the set of ground truth, $\hat{Y}$ is the set of prediction. Generally, all the samples in X has the same appearance probability, and we have

$$H = -\frac{1}{|X|}\sum_{x\in X}\sum_{y\in Y}\sum_{\hat{y}\in\hat{Y}} p(y,\hat{y}|x)\log p(y,\hat{y}|x) - \log|X|, \tag{3.25}$$

where $|X|$ is the scale of the set X. Generally, the ground truth y is an exact label for every input x. Therefore, for an input data x_i whose label is y_i, we have

$$p(y,\hat{y}|x=x_i) = \begin{cases} p(\hat{y}|x=x_i) & y=y_i \\ 0 & y\neq y_i. \end{cases} \tag{3.26}$$

Therefore, we have

$$H = -\frac{1}{|X|}\sum_{x\in X}\sum_{\hat{y}\in\hat{Y}} p(\hat{y}|x)\log p(\hat{y}|x) - \log(|X|). \tag{3.27}$$

For a fixed task, $|X|$ is a constant, and we have

$$IG_t = -\frac{1}{|X|}\sum_{x\in X}\sum_{\hat{y}\in\hat{Y}} p_0(\hat{y}|x)\log p_0(\hat{y}|x) + \frac{1}{|X|}\sum_{x\in X}\sum_{\hat{y}\in\hat{Y}} p(\hat{y}|x)\log p(\hat{y}|x) \tag{3.28}$$

$$= \mathbf{E}_X[H_0(\hat{y}|x)] - \mathbf{E}_X[H(\hat{y}|x)], \tag{3.29}$$

where p_0 is the probability of the model making correct prediction before training and p is the probability after training, $H_0(x)$ is the prediction entropy of input x before training t and $H(x)$ is the prediction entropy after training. For more details of implementation about computing of $p(\hat{y}|x)$, we use softmax to map the output of the model to the prediction probability distribution $p(\hat{y}|x)$ and calculate H. Different from the accuracy, H is the entropy of the prediction, which is unrelated to the ground truth y. It represents the chaos of the prediction and reflects the confidence of the DNN about this prediction. More knowledge means more confidence and also means lower prediction chaos. Therefore, IG_t reflects the knowledge incremental related to the task in training theoretically.

	4-layers	6-layers	8-layers	10-layers
Accuracy	0.745	0.787	0.808	0.843
Loss	0.808	0.676	1.947	1.287

Table 3.3: Teachers' performance in the experiment.

In this section, we introduce experiments to evaluate task-related Information (TI) in different conditions. Based on [MFLG19], we design a group of experiments in which the student network is 2-layer CNN, and teacher networks are 4-layer, 6-layer, 8-layer, 10-layer CNN. Besides, we add two teachers in our experiments, one is Label Smoothing Regulation (LSR) (see [YTL$^+$19]), and the other is Fake Noise Teacher (FNT). FNT is a group of Gaussian noise. We control the mean and variance of noises as the output of the teacher. These noise vectors are used to replace the output of the teacher to distill 2-layer CNN.

Dataset : CIFAR-10 [KH09a] is an image dataset with color close to real objects. Specifically, CIFAR-10 consists of 60000 32×32 images (including 50000 training images and 10000 test images) in 10 classes, with 6000 per class. We use CIFAR-10 to evaluate the performance of knowledge distillation include accuracy, loss, and TI.

Models : The student model in our experiments is fixed as 2-layer CNN, and teacher models are 4-layer, 6-layer, 8-layer, 10-layer CNN, LSR, and FNT, respectively. We train each of these models from scratch 100 times repeatedly and then calculate the mean of their performance. The loss function of KD in our dissertation referred to Eq. (3.22) (see [YTL$^+$19]). Like the loss function in KD, the LSR loss function is referred to Eq. (3.20) as mentioned in [YTL$^+$19]. We set the KD temperature $\tau = 20$, and balance weight $\alpha = 0.9$ in Eq. (3.22). The batch size in experiments is 128. The number of epochs is 60, and the learning rate is 0.001 in each training sequence. The filter size of all the convolution layers is 3×3, and the max-pooling layers hold a stride of 2 and kernel size of 2.

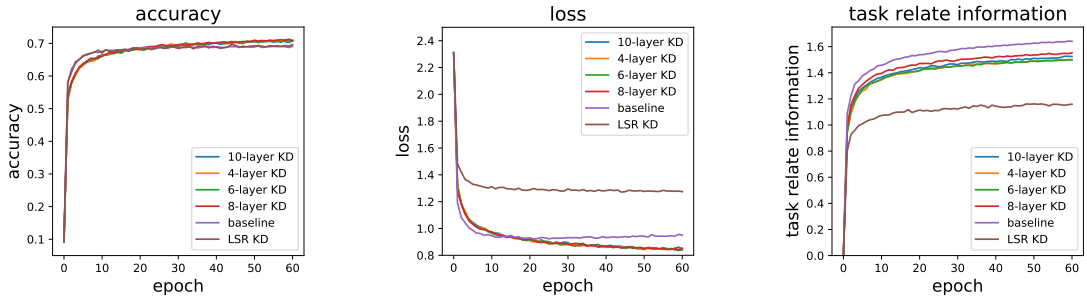


Figure 3.12: The performance of models in the experiment. The left one shows the accuracy of models in experiment; the middle one shows the loss of models; the right one shows the task-related information of the model.

Contrary to our perception, the knowledge distillation does not provide extra information and enhances TI accumulation (see the right figure in Fig. 3.12). Oppositely, the knowledge distillation process weakens the accumulation of the TI. In this aspect, the result fits the conclusion in [YTL+19] that knowledge distillation is a special case of LSR. However, this result contradicts our subjective cognition. Why weakening the learning of the DNN can improve its performance? To explain it, we analyze the loss and the accuracy of all models. As Fig. 3.12 shows, for the baseline, the loss of the model on the test dataset increases continuously after getting to the lowest point (see the baseline curve), which is an obvious trend of overfitting. It indicates that most of the knowledge learned by the network is unrelated to the commonality between the training dataset and the test dataset in this part of the learning process. Although the knowledge distillation weakens the knowledge accumulation, it enhances the mainstream knowledge related to the commonality between the training dataset and the test dataset relatively. Therefore, in the later stage of training, models trained by knowledge distillation avoid the overfitting successfully(see Fig. 3.12).

One step further, we use a Gaussian noise ($\mathcal{N}(\mu, \sigma)$) to replace the teachers (6-layers CNN) output in the knowledge distillation process, where μ and σ are dynamic based on

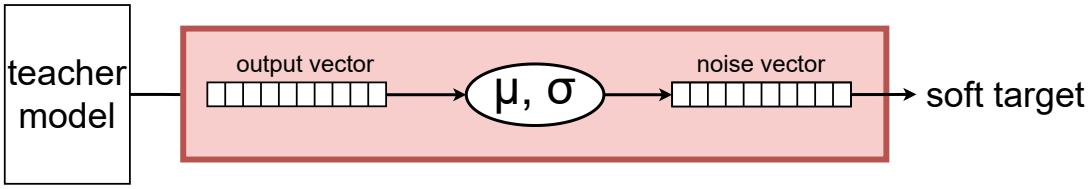


Figure 3.13: Using the noise to replace the normal teacher's output as the soft target in knowledge distillation.

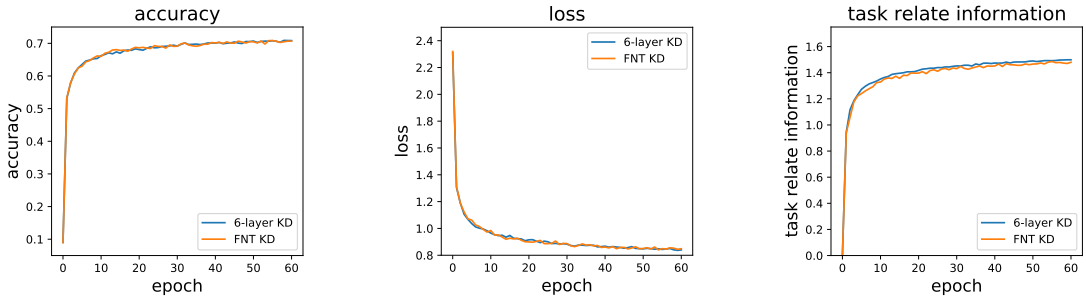


Figure 3.14: The comparing of the performance of models trained by the fake teacher (Gaussian noise) and trained by the normal teacher (6-layer CNN).

the corresponding teacher's output as Fig. 3.13 shows. As Fig. 3.14 shows, the performance of the model trained by the fake teacher is almost the same as the model's performance trained by the normal teacher. This result verifies our conclusion that **the knowledge distillation process is not a knowledge transfer process but a knowledge weakening process**.

3.4.3 Discussion

In [CRCZ20], Cheng et al. point out that the knowledge distillation has three main effects, including letting the student learn more visual concepts, letting the student learn visual concepts simultaneously, and making the optimization direction stable. Their conclusion is not in conflict with our result. As we mentioned before, though knowledge distillation

weakens the accumulation of knowledge, this effect is the same for all kinds of knowledge, whether related to the task or not. Therefore, it enhances the mainstream knowledge, reflecting the commonality of the data, relatively and reduces the impact of the datas' characteristics. In this aspect, our result provides another explanation to support their conclusion that knowledge distillation can make the optimization direction stable. What needs to emphasis is that we do not doubt the effect of knowledge distillation. We reveal the essence of this phenomenon from our viewpoint.

Summary : In this dissertation, we use a novel metric, task-related information (TI), to measure the information learned by the DNN related to the task. We analyze the knowledge distillation process based on this method and find that it is not a knowledge transfer process but a knowledge reduction process. The knowledge distillation weakens the knowledge accumulation in the training process, enhancing the mainstream knowledge. To further verify our conclusion, we use noise to replace the teacher's output in the knowledge distillation. The result shows that the effects of these noises are almost the same as a normal teacher's outputs, which meets our expectations.

3.5 Conclusion

This section summarizes related works and introduces two of our works about information quantification and knowledge visualization of DNN. The first reveals the relationship between the information quantity and the distance of the DNN's weights before and after training, providing an effective tool to evaluate the training effectiveness. The second shows a discovery when using the task-related information quantification method to analyze the knowledge distillation phenomenon. We find that knowledge distillation is not a knowledge transfer process but a knowledge weakening process. Based on these methods, we can

take a closer look at the neural network and visualize the content learned by the DNN effectively, which lays the foundation for us to control the content learned by the DNN. This part of the works is introduced in the next section.

CHAPTER 4

CONTENT CONTROL IN DNN'S TRAINING

4.1 Content Guide

In this chapter, we will introduce our work in content controlling of DNNs. Our research is mainly based on recent achievements in computer vision. Based on the input data type, computer vision can be divided into two parts, 2D image-based tasks, and 3D meshes-based tasks. For each kind of task, we provide one successful work respectively in the following section. **These successful attempts show that the content controlling is meaningful and avoids the risk of DNN's cheating effectively.**

This chapter is structured into four main sections. It is starting from the current section that outlines the research and introduces the basic idea of content control.

In the second section, for the 3D model analysis, we introduce the work in human brain structure analysis. The preliminary research shows that the regional distribution of the brain, including the location, the contour, the volume, etc., is highly related to the function of the human brain. It implies that human intelligence is measurable without a subjective exam, which provides another view to analyze brain mechanisms. For a complex human brain mesh, it is hard to learn directly. Moreover, serious overfitting happens in training. Therefore, content controlling is concerned. We divide the source data from the following aspects, region area, region volume, and boundary skeleton without the brain matter's attributes. From the experiment results, the boundary skeleton has the best performance. Specifically, we use the conformal welding method to extract and represent the boundary skeleton. In this process, we need to register each region of the brain. As a supplement, we provide a novel method to register 3D Surfaces with point and curve landmarks. Strictly, this work also belongs to the traditional feature extraction. Before sending the data to the DNN, it needs pre-processing to extract the feature. The main novelty is shown in

the advantage of geometry tools that can accurately and effectively segment the contour feature. However, an interesting discovery in the experiment inspired us. We found that compared with the single feature of boundary skeletons, the more features we added, the worse the model shows, which inspires us that the information provided to the DNN is not the more, the better.

In the third section, for the 2D image analysis, we introduce the work in human pose estimation and detection. As mentioned in Chapter 2, we find the network cheating phenomenon in a system for labor injury protection. Theoretically, by monitoring the poses of workers and reminding workers to take breaks, this system can effectively protect the work from cumulative labor injury. The core of this system is a DNN to detect and recognize the workers' poses. However, because of the dataset's bias, this DNN learned to use the color of cloth (and helmet) to classify the human pose. Therefore, we simplify the data and remove the information related to the color of workers in the images. Without impacting the DNN's accuracy, this method eliminates the cheating phenomenon and accelerates the training simultaneously. **It is the first successful case of content controlling because the model can accept the images in the real world without pre-processing.**

4.2 Definition of Content Controlling

content controlling is the next stage of feature extraction based on DNN. In the traditional feature extraction, we set the target feature firstly, like the texture, boundary, etc., and then try to extract them. For example, as Fig. 4.1 shows, we can use a convolution kernel to extract the boundary of the image. Then based on the boundary's pattern, we can use the program to recognize the content in the image. Here, the feature is set artificially, which means we don't know the exact effectiveness of these features.

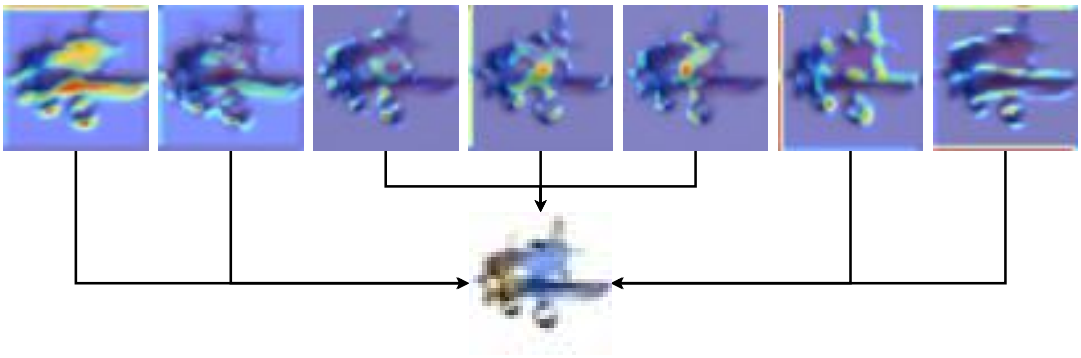


Figure 4.1: The input images can be viewed as superposition of various features.

Unlike the traditional feature extraction, for content controlling, we intend to visualize the feature extracted by DNN and reconstruct them to an understandable form as Fig. 4.2 shows In other words, we use a fine-tuned DNN to help us to find the feature from the data. Then we use the reconstructed data to feedback the network and repeat this loop until the DNN is fully trained. What needs to emphasis is that the main goal of content controlling is not improving the performance but improving the robustness of the DNN. For the implementation, we introduce it formally in the following part of this section.

Here, we view each sample of input data, denoted as S, as superposition of various features, denoted as s ($S = \sum s$), as Fig. 4.2 shows. Because of the arbitrariness of feature extraction, it is hard to divide S into a series of s. Unlike the feature extraction, in content controlling, s's definition is based on the DNN's extraction, which means we have had a fine-tuned DNN ($M = B, C$) who can deconstruct S into a graph of s and make predictions. Here, B is the backbone which can be viewed as a set of feature extractor $\{b_i\}$ and C is the determiner or classifier which makes the final prediction based on the feature graph. Formally, s is defined as Eq. (4.1).

$$s_i = b_i(S) \tag{4.1}$$

Based on the effectiveness of these features in DNN's classification, we can select the most

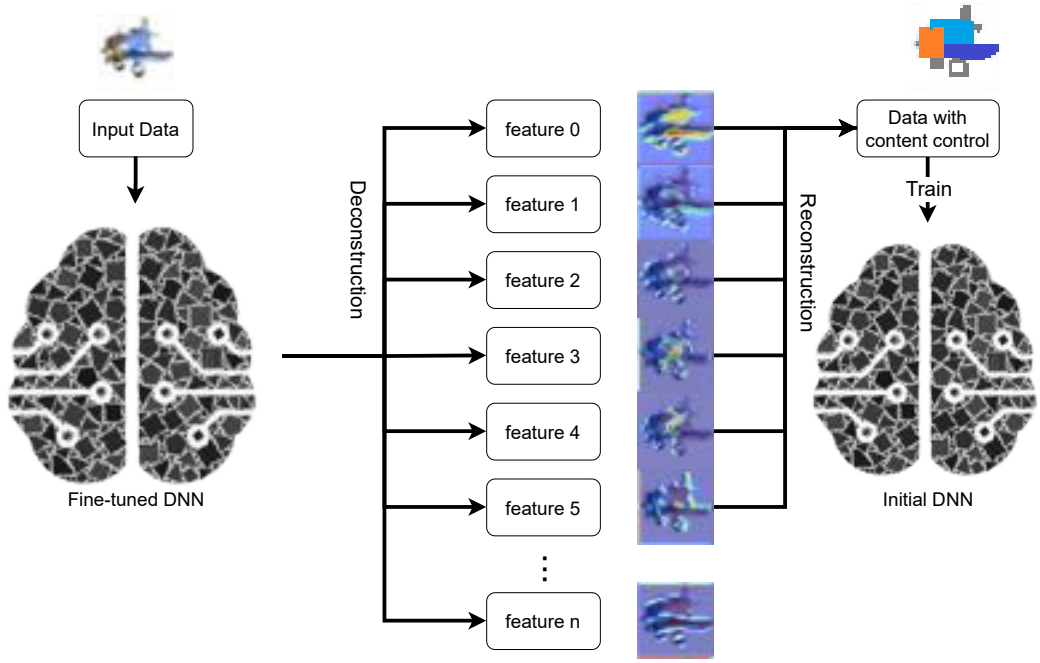


Figure 4.2: The Pipeline of content control.

significant ones and reconstruct them into a new feature S'. Formally, we have

$$S' = \sum_{i=0}^{n} s_i. \tag{4.2}$$

The learning process of the DNN is a process of weights updating basically. The weights updating based on the original data S can be represented as

$$\Delta W = \eta \frac{\partial L(S)}{\partial W} \tag{4.3}$$

$$= \eta \frac{\partial L(\sum s_i)}{\partial W}. \tag{4.4}$$

Inside of each layer, the relationship between the input and output is linear. Therefore, we have

$$\Delta W = \eta \frac{\sum \partial L(s_i)}{\partial W}. \tag{4.5}$$

If we rank the items based on the contribution to the task and keep the first n items in Eq. (4.5), we can imply the content control. The most significant difference between the content controlling and the traditional training is that the reconstruction process of S' is

under the premise of ensuring that the features are understandable and meaningful, which will be shown in the following sections.

4.3 Preliminary Work: Diffeomorphic Registration of 3D Surfaces

4.3.1 Introduction

Surface registration plays a fundamental role in computer graphics and engineering fields, which is widely used in tracking, classification, recognition, and so on [HKDH04, CF01, WVGKP99]. Given the 3D source and target surfaces with point and curve feature landmarks (S_k, p_k, ℓ_k), $k = 1, 2$, the registration problem is to find a mapping $f : (S_1, p_1, \ell_1) \to (S_2, p_2, \ell_2)$ such that the point and curve feature landmarks are aligned correspondingly. The desirable mapping should be a smooth diffeomorphism (bijective). Also it needs to satisfy the accuracy and speed requirements. In reality, most objects have their natural landmarks, including feature points and landmark curves. Consistent curve feature landmarks exist extensively and are often utilized to guide the 3D surface registration and analysis in computer-aided medical diagnosis and tumor or abnormality detection. For example, sulci and gyri landmark curves are used in brain registration for morphometry analysis; facial symmetry curves are tested in adolescent idiopathic scoliosis and autism diagnosis; taenie coli muscle lines and haustral folds are applied for colon wall registration in virtual colonoscopy. The consistent feature curve landmarks can be manually labeled or automatically extracted by auxiliary detection methods. One of the most challenging problems in 3D surface registration is to generate a bijective mapping that matches the corresponding feature landmarks consistently, especially when a large number of landmark constraints are enforced. For 3D surface registration, we prefer to align the curve landmarks rather than try to align the numerous points on the corresponding

curves for the following two reasons. First, the registration method using curve landmarks has more freedom since the interior points of the source curve can slide on the target curve. Second, the correspondence computation is much simpler for the methods based on curve landmarks than on point landmarks. This work motivates us to search for an intrinsic mapping for 3D surfaces with both point and curve landmarks, which is diffeomorphic and minimizes the local geometric distortions.

According to surface uniformization theorem [Far04], any 3D surfaces can be conformally mapped to 2D canonical domains. The mapping between two 3D surfaces can be converted to the mapping between their 2D conformal domains. This framework has been extended to deal with surface registration with point feature landmarks [LGYL13, ZG11, ZMLG14]. To deal with curve feature landmarks, quasiconformal optimization methods [SZS$^+$13, ZY14, ZMLG14] have been presented to register the 3D surfaces with curve feature landmarks. However, the curve landmarks are mapped to horizontal/vertical line segments in the canonical domain, whose styles are determined by some heuristic methods. As a result, both the registration freedom and its accuracy are limited by this forced restriction. Furthermore, the quasiconformal optimization methods can not guarantee the bijectivity of the surface registration theoretically. It always stops at a local optimum and hardly get a global solution.

This section proposes a novel method to register the 3D surfaces with point and curve feature landmarks, which extends the conformal mapping-basedmapping-based framework to intrinsically deal with both point and curve landmarks. First, the 3D surfaces with point and curve feature landmarks are mapped to the canonical domain by a curve constrained harmonic map, which globally minimizes the harmonic energy and maps the curve landmarks to canonical shapes (straight line segment), whose positions as well as inclining angles are determined intrinsically by the surface geometry and its feature landmarks. The resultant mapping is unique and diffeomorphic. The 2D domains

are then registered by the dynamic quasiconformal method to align the corresponding points and straight-line segments. The endpoints of each source line segment are mapped to the corresponding endpoints of the target segment while the target positions of the interior points are computed automatically by the dynamic quasiconformal method, which introduces the diagonal edge switches to the quasiconformal energy optimization such that the resultant map is bijective. The combinatorial switching is required to generate the final diffeomorphic quasiconformal maps; otherwise,, one might get fold singularities due to the corresponding computation of the curves much simpler than that of the point landmarks. Compared with the registration methods with only point landmarks [LGYL13, ZG11, ZMLG14], our proposed intrinsic registration method has more flexibility to deal with the curve landmarks as the interior points of the source curves can slide along the corresponding target curves, which will lead to more accurate and intrinsic registration results. Furthermore, our framework can deal with mixed feature landmarks with both points and curve landmarks, which is intrinsic, general, and more useful than the previous methods only dealing with point landmarks.

4.3.2 Related Works

In the past decade, 3D surface registration method have been intensively explored [FMK+03, HKDH04, LCS16, SH07, SA01, WVGKP99], which has a broad range of applications including shape matching and recognition, shape modeling, morphological study and animations. Most existing methods directly deal with non-rigid deformations but always stop at a local optimum and hardly get a global solution.

Recently, a lot of research focuses on surface conformal and quasiconformal mapping-based methods [LGYL13, ZG11, ZMLG14, SWGL15, ZMLG14]. According to surface uniformization theorem [Far04], any arbitrary surface can be conformally mapped to one

of three canonical domains, the unit sphere, the Euclidean plane, or the hyperbolic disk. By mapping surfaces to 2D canonical domains, the problems of 3D surface registration are reduced to a 2D image registration problem. As a general mapping, quasiconformal mapping has been arousing more attention recently [LGYL13, ZMLG14, SZS+13]. The auxiliary metric [SZS+13], and holomorphic Beltrami flow [LWZ+12] were presented to compute the quasiconformal surface maps from the given Beltrami coefficient. For concave boundaries, the extremal quasiconformal maps with a unique extremal Beltrami coefficient are computed in [WMZ12] for surface parameterization.

In real applications, landmark constraints are usually prescribed to guide the surface registration, which may fold singularities in the resultant mapping. Among the various feature landmarks (usually points or curves), the curve feature landmark plays an important role in the constrained surface registration. Although there are many methods for surface registration, few of them satisfy both the diffeomorphism and curve constraints. To avoid singularities folding, many methods utilize highly nonlinear functions [CLR04, FLG15, HG00, SSGH01] to ensure bijectivity for surfaces with feature landmarks (usually point landmarks), where the function grows fast to infinity as the mapping develops folds. As a result, the functionals are highly nonlinear, require intensive computing, and also stops at a local minimum. Furthermore, they require a valid starting point for further optimization. The constrained parameterization method in [Lév01] optimizes Dirichlet energy and imposes point position and gradient constraints with penalty terms without attempting to guarantee bijectivity. The quasiconformal method computes the approximations of the extremal quasiconformal methods for surface registration with the point feature landmarks, where the distribution of the Beltrami coefficient is optimized. For the surfaces with curve feature landmarks, traditional methods first sample the landmark curves and align the feature points. As a result, the curve landmarks cannot be strictly aligned along with the curve landmarks. The accuracy of the point-constrained methods can be improved by dense

sampling, which will increase the time cost and affect their efficiency. The hyperbolic metric [SZS$^+$13] is presented to handle landmark curve constraints by slicing the surface open along with them and map them to boundaries of canonical domains, which changes the surface topology, is highly nonlinear and restricted to deal with disjoint curves. The methods in the map the curve landmarks on the surface to horizontal/vertical line segments in the canonical domain. The style of the curve landmark is computed separately, which will introduce more distortion to the resultant map. For a 3D surface with consistent graph constraints, an intrinsic method is presented in [YRZ18] to generate the bijective registration. However, the method cannot deal with the disjoint curve landmarks. In this dissertation, the curve feature landmarks are intrinsically mapped to canonical shapes in the canonical domain, which is then registered by the dynamic quasiconformal map method.

contribution : The major contribution of the current work is to present a diffeomorphic registration method for surfaces with point and curve feature landmarks, where the point and curve landmarks of the source and target surfaces are aligned intrinsically. The endpoints of the source and target landmarks are matched, and the interior points of the source landmarks can slide on the corresponding target curves, and their positions are determined by the surface geometry and its feature landmarks. By utilizing the freedom of the combinatorial structures (diagonal edge swaps), our registration method can guarantee to generate a diffeomorphic mapping for surfaces with point and curve feature landmarks. To our best knowledge, this is the first work that 1) satisfies both point and curve feature constraints, 2) guarantees the mapping diffeomorphism theoretically and 3) has small stretch distortion.

In this section, the 3D constrained surfaces are first mapped to the 2D canonical domain, where the curve landmarks are mapped to canonical shapes (straight line segments).

The positions and inclining angles are determined by the surface geometry and its feature landmarks. Then the two canonical domains are registered by the diffeomorphic dynamic quasiconformal maps, which align the corresponding point and curve landmarks. In summary, our method is 1)general, it can deal with both point and curve landmarks; 2) intrinsic, the point and curve landmark alignment is computed automatically; 3) diffeomorphic, the resultant registration mapping minimizes the harmonic energy, which is bijective and determined by the surface geometry and the feature landmarks; 4) rigorous, the method has solid theoretical foundations.

4.3.3 Diffeomorphic Surface Registration

Suppose two 3D surfaces $S_k = (V_k, E_k, F_k)\,, k = 1, 2$ with point $p_i, i = 1, 2$ landmarks and curve landmarks $\ell_1 = \{l_i\}\,, \ell_2 = \{r_i\}\,, i = 1 \cdots m$ ri. Each curve landmark is represented as a set of continuous connected edges, $l_i\,(r_i) = [e_0, \cdots, e_j, \cdots, e_{n_i}]$, $e_j \in 1(E2)$, where ni is the number of consecutive edges in the curve landmark. To differentiate with the pure surfaces without constraints, we denote the constrained surfaces with point and curve feature landmarks as (S_k, p_k, ℓ_k), where p_k and ℓ_k are its associated point and curve landmarks, respectively. By the Curve Constrained Harmonic Map (CCHM) presented in [YRZ18], the 3D surfaces S_k with point landmarks pk and curve landmarks ℓ_k are mapped to 2D canonical domains D_k with canonical landmarks q_k (points) and L_k (straight line segments), which is proved to be unique and bijective. Here each curve landmark on the 3D surface is mapped to a straight line segment in the 2D canonical domain, whose position and inclining angle are determined by the surface geometry and the curve landmarks, and computed automatically by the CCHM map. Suppose D_1 and D_2 are the canonical mapping domains of S_1 and S_2, respectively. After mapping the 3D curve constrained surfaces to the canonical domains by CCHM, the mapping between two constrained surfaces f :

$(S_1, p_1, \ell_1) \to (S_2, p_2, \ell_2)$ is converted to the mapping between their constrained domains $\widetilde{f} : (D_1, q_1, L_1) \to (D_2, q_2, L_2)$, which is computed by the Dynamic Quasiconformal Map (DQCM) method. Let $\phi_k : S_k \to D_k$ denote the corresponding CCHM map between 3D surfaces S_k and their corresponding constrained domains D_k. The mapping between parameter domains $\widetilde{f} : D_1 \to D_2$ induces the registration $f : S_1 \to S_2$ between two 3D surfaces, which can be expressed by a combination of the above mappings $\phi_2^{-1} \circ \widetilde{f} \circ \phi_1$. f is a diffeomorphism if and only if $\widetilde{f}$ is a diffeomorphism. By introducing the combinatorial freedom (diagonal edge swaps) to the quasiconformal optimization, the DQCM map generates a diffeomorphic mapping $\widetilde{f}$.

Harmonic Map

Harmonicity is closely related to conformality. The discrete harmonic map can be computed by the convex combination map with the Dirichlet condition, where each interior vertex v_i can be expressed as a linear combination of its neighboring vertices v_{ij} as follows.

$$\begin{cases} v_i = \sum_{j=1}^{k} \lambda_{ij} v_{ij} \quad j = 1, \cdots, k \\ \sum_{j=1}^{k} \lambda_{ij} = 1 \\ \lambda_{ij} > 0 \end{cases} \tag{4.6}$$

, where λ_{ij} are the harmonic weights. The mean value coordinate [Flo03] is defined as follows:

$$\lambda_{ij} = \frac{\tan\left(\alpha_{ij-1}/2\right) + \tan\left(\alpha_{ij}/2\right)}{|v_{ij} - v_i|} \tag{4.7}$$

, where the α_{ij-1} and α_{ij} are the adjacent angles in triangles $[v_{ij-1}, v_i, v_j]$ and $[v_{ij}, v_i, v_{ij+1}]$, respectively. For the convex combination map, we have the following lemma.

Lemma 4.3.1 *(Convex Combination map [Tut60]) Given a simply connected triangular mesh M and a convex domain Ω, if the map : $\phi : M \to \Omega$ is a convex combination map,*

i.e. for every interior vertex, it satisfy the conditions in Equation (4.6), and ϕ maps ∂M to ∂Q homeomorphically, then ϕ is one-to-one.

For the cases $k = 2, 3$, the weights λ_{ij} can be determined automatically by Equation (4.6); they are the barycentric coordinates. For the general cases $k > 3$, the mean value coordinate in Equation (4.7) can be obtained by approximating the harmonic energy using the Circumferential Mean Value Theorem at each interior vertex [Flo03]. A conformal mapping can be obtained by solving the convex combination map in Equation (4.6).

To compute the harmonic map of the source domain with the prescribed Beltrami coefficient μ, we need to modify the mean value weight accordingly. Suppose the Beltrami coefficient is defined on the vertices of the triangle $[v_i, v_j, v_k]$, which can be isometrically mapped to the planar domain by a linear mapping

$$\tau(z) = z + \frac{1}{3}\left(\mu\left(v_i\right) + \mu\left(v_j\right) + \mu\left(v_k\right)\right)\bar{z} \tag{4.8}$$

, where z denotes the local coordinates of the triangle. Then we measure the angles of the distorted triangle in the planar domain and compute the mean value weight using the distorted angles.

Curve Constrained Harmonic Map (CCHM)

A straightforward method to compute the harmonic map with curve straightening constraints is first to determine the inclining angle of the target line segment by some heuristic methods and introduce the formulated linear constraints to the harmonic minimization problem. However, the resultant map may introduce fold singularities with these straightening constraints. By applying the Circumferential Mean Value Theorem of harmonic functions adaptively according to the feature landmarks, Yang [YRZ18] derives the intrinsic harmonic map of the curve constrained surfaces. The mean value coordinate is

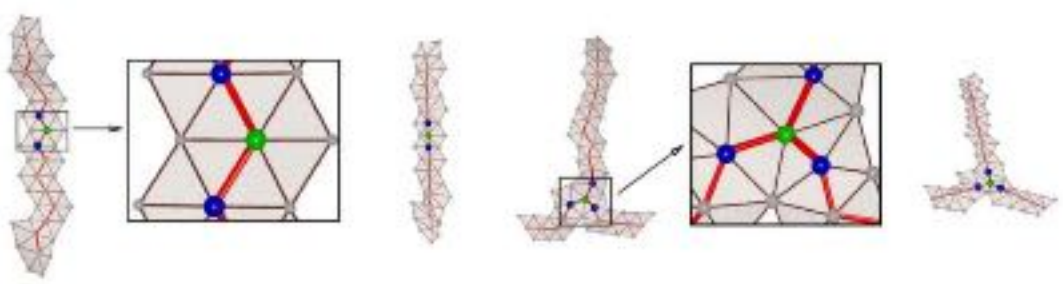


Figure 4.3: CCHM map. The left column and right column show the two cases of the vertex lying inside the interior of the curve landmark and the vertex connecting multiple curves, respectively. For the original mesh with curve landmarks (left), each row shows the zoomed view (middle) and the CCHM map (right). The blue points are the one ring curve neighborhood of the green ones. [YRZ18]

modified adaptively according to the landmark curves such that the convex combination map defined in Equation (4.6) satisfies the Circumferential Mean Value Theorem at every interior vertex, and it straightens the landmark curves to line segments in the canonical domain. For a vertex lying inside the landmark curve, its *one-ring curve neighborhood* is defined as its adjacent vertices lying on the curves while the *one-ring neighborhood* includes all adjacent vertices. For the vertices lying inside the landmark curves, their one-ring curve neighborhood is utilized during the computation of the adaptive mean value coordinate, and the interior points of the landmark curves will move to the linear interpolation of their two adjacent curve neighbors on the feature landmarks, which will result in a planar straight line segment in the canonical domain (see Fig. 4.3). In detail, to compute the intrinsic harmonic map of curve constrained surfaces, the harmonic weights are computed as follows. If the vertex v_i is

1. not on the curve, the mean value coordinate defined in Equation (4.7) is utilized.
2. lying inside the landmark curves, the barycentric coordinate is applied to its one-ring curve neighborhood instead. Let v_{i1} and v_{i2} denote its two adjacent neighboring vertices on the landmark curves. The adaptive harmonic weight is defined as $\omega_{i1} = \frac{|v_{i2}-v_i|}{|v_{i2}-v_i|+|v_{i1}-v_i|}$ and $\omega_{i2} = \frac{|v_{i1}-v_i|}{|v_{i2}-v_i|+|v_{i1}-v_i|}$

3. connecting multiple curves, the Circumferential Mean Value Theorem is applied to its one-ring curve neighborhood to compute the adaptive harmonic weight.

For the curve constrained harmonic map (CCHM), we have the following lemma.

Lemma 4.3.2 *The curve constrained harmonic map (CCHM), which maps the curve landmarks to canonical shapes (straight line segments) in the canonical domain, is unique, globally optimal and diffeomorphic when the target domain is convex. [YRZ18]*

For the vertices lying outside the feature curves, according to the adaptive scheme, it is the same as the mean value coordinate defined in [Flo03]. Using the one-ring curve neighborhood during the weight computation for the vertices lying on the feature curves and removing the pulling to other directions, the feature curves will be straightened to canonical shapes (straight line segments) in the intrinsic harmonic map. At the same time, the formulated harmonic energy remains convex, and each vertex can be expressed as a convex combination of its one-ring neighborhood. The intrinsic harmonic map is a diffeomorphism when the boundary is a convex polygon. Furthermore, it is unique and a global minimum of the Dirichlet energy under the curve straightening constraints.

Dynamic Quasiconformal Map (DQCM)

This section describes the computational algorithm of the dynamic quasiconformal map, which deforms the 2D domain with point and curve landmarks while preserving shapes as much as possible. Delaunay triangulation has been applied in geometry processing to guarantee the diffeomorphism and convergence of conformal mappings [BS07]. This dissertation introduces constrained Delaunay triangulation to quasiconformal maps to deal with the point and curve landmarks.Thiss is the first work to present dynamic quasiconformal map to the author's best knowledges, which utilizes the freedom of combinatorial structure to generate bijective quasiconformal mappings for surfaces with point and curve

constraints. The computational method is based on the quasiconformal optimization and the constrained Delaunay switches. The strategy is first to obtain the Beltrami coefficients using the CCHM map with point and curve endpoint constraints by matching the point landmarks and endpoints of the curve landmarks. By fixing the positions of the points (both the point landmarks and endpoints of the curve landmarks) in the target domain, the initial registration map may introduce fold singularities. To generate a bijective map, the source domain is optimized by moving a small step using a scaled Beltrami coefficient induced by the CCHM map with point constrains, which is interleaved with the constrained Delaunay diagonal switches such that each unconstrained edge is constrained Delaunay, that is the sum of the opposite angles of each unconstrained edge is less than π. Starting from the new intermediate domain with CDT, we perform the above optimization procedures repeatedly until the current CCHM map is bijective and all unconstrained edges are constrained Delaunay. Even for large point and curve deformations, our algorithm will converge steadily. Our resultant solution is diffeomorphic. The computational pipeline is as follows.

1. Compute the CCHM map with point constraints. For the 2D registration between (D_1, q_1, L_1) and (D_2, q_2, L_2), where q_1 and q_2 are the 2D point landmarks, L_1 and L_2 are 2D straight line segments to be aligned, we first compute its CCHM map with point constraints, which maps the two end points of each landmark $l_i \in L_1$ to the end points of the corresponding landmark $r_i \in L_2$in the target domain. With additional point constraints involved, the CCHM map ϕ may introduce fold singularities near the landmarks.

2. Quasiconformal Optimization. Let μ denote the Beltrami coefficient of the map ϕ. Let $t \in (0, 1]$ denote the step length and compute the quasiconformal map ϕ_t by the harmonic method with the auxiliary metric induced by the Beltrami coefficient t_μ.

3. Combinatorial Improvement. The triangles of the intermediate domain $\phi_t(D_1)$ are improved by diagonal switches ϕ to satisfy the constrained Delaunay property. Let $D_1 = \phi(\phi_t(D_1))$ and perform Step 1-3 repeatedly, until ϕ is a bijective map.

The 2D desired mapping ef is a combination of diffeomorphic maps $\varphi_n \circ \phi_{t_n} \cdots \varphi_i \circ \phi_{t_i} \cdots \varphi_0 \circ \phi_{t_0}$. After the 2D mapping $\widetilde{f}$ of the domains is obtained, the corresponding 3D surface registration can be obtained by a combination of the 2D dynamic quasiconformal mapping and the CCHM mappings $f = \phi_2^{-1} \circ \widetilde{f} \circ \phi_1$. The resultant surface registration f is bijective and intrinsic to the surface and its feature landmarks.

Theorem 4.3.3 *The dynamic quasiconformal map is diffeomorphic.*

The previous optimization-based methods suffer from the highly nonlinear formulated energy, always get stuck in local minima, and can not guarantee bijectivity of the final registration theoretically. By introducing the combinatorial freedom to the quasiconformal optimization, we can always obtain a bijective registration for 3D surfaces with point and curve landmarks (CDT always exists for given point and curve landmarks), and the solution has small distortions.

4.3.4 Experiments

The primary goal of this work is to solve the problem of surface registration with point and curve landmark constraints, which are required to be diffeomorphic. The main strategy is to present the canonical parameterization of curve constrained surface and then introduce the dynamic quasiconformal map (DQCM) to perform registration over the canonical 2D domains, where curve landmarks are straightened and curve constraints become linear ones. After defining the connection patterns, the curve landmarks can be derived from the point landmarks by connecting each pair of point landmarks in the connection patterns by the shortest path method. Many parameterization-based methods are presented to generate

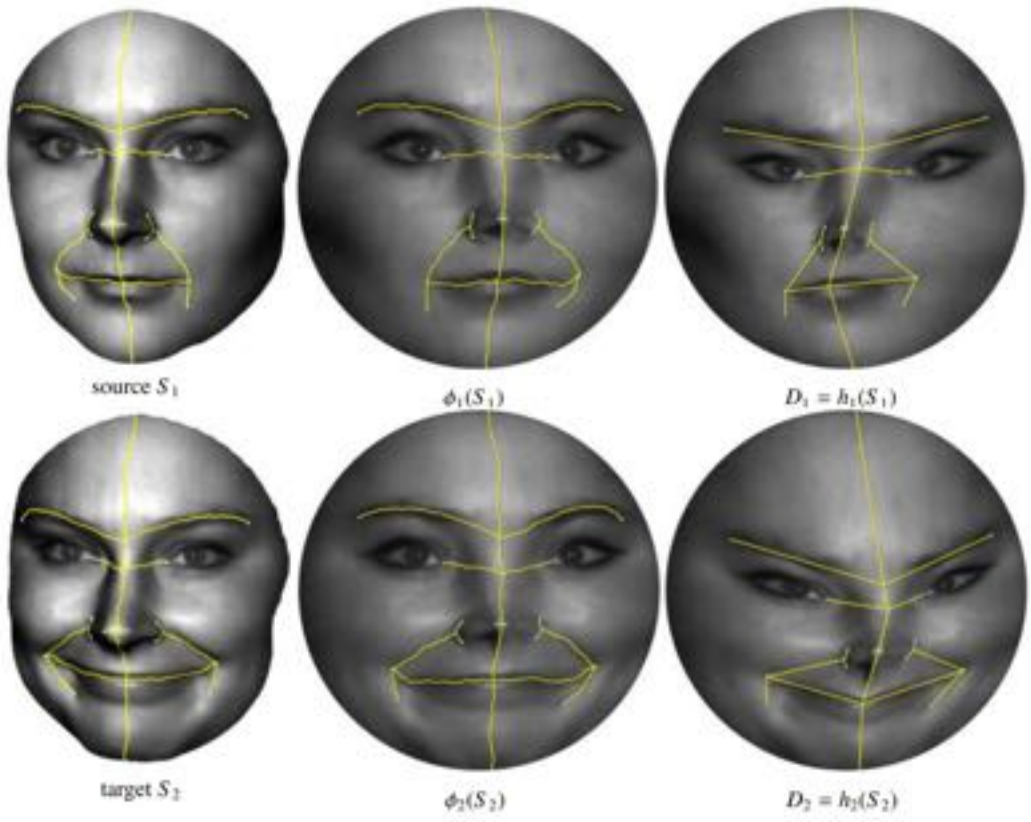
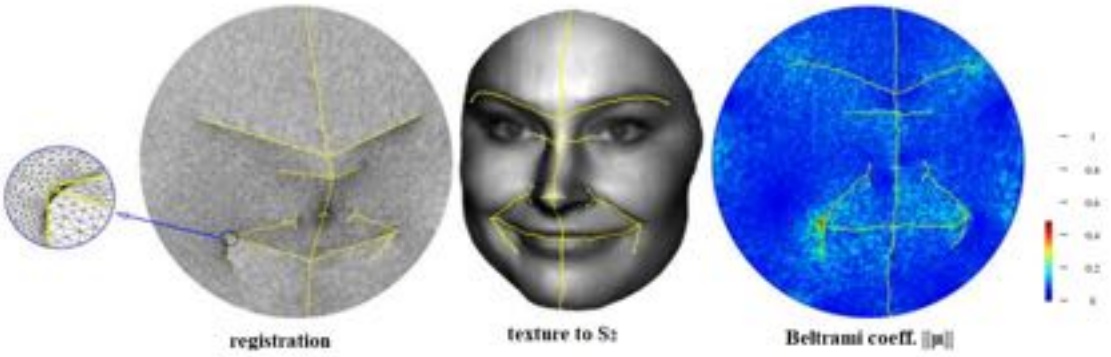
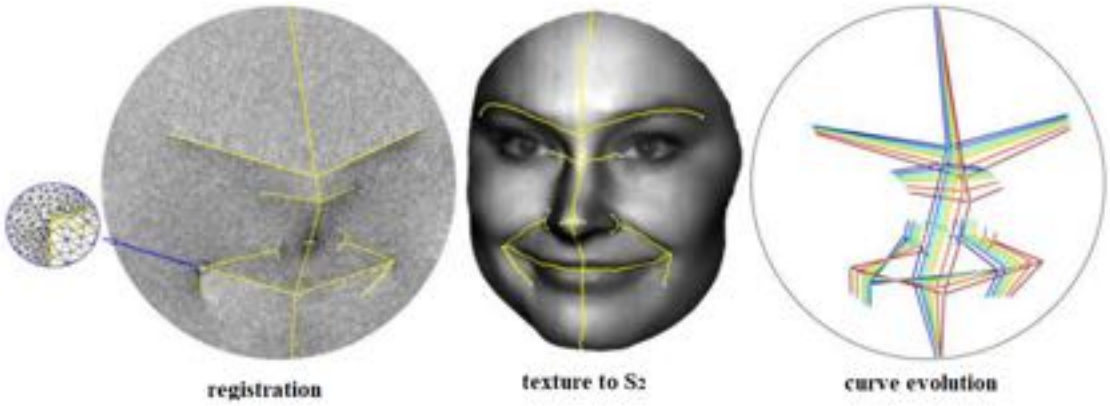


Figure 4.4: Conformal map and CCHM of neutral and smile expressions of the same subject

the registration of 3D surfaces with landmarks. Several important ones are summarized in Table 1. We focus on several important properties, whether methods in this class can deal with point feature landmarks (PFL), curve feature landmarks (CFL), bijective (BIJ) theoretically. Our method is the first one to guarantee to generate a bijective registration theoretically for 3D surfaces with point and curve landmarks due to the introduction of the combinatorial freedom to the quasiconformal optimization. In the following, we test our algorithms on two categories of curve constraints: 1) intersecting curves and 2) isolated curves, and then give the performance discussion of our method. The experimental results demonstrate the efficiency and flexibility of the proposed method for point and curve constrained surface registration.



(a) registration using one-step constrained harmonic map



(b) registration using one-step constrained harmonic map

Figure 4.5: Facial surface registration of neutral and smile expressions of the same subject

Surface Registration with Curve Landmarks

Figure 4.4 shows the registration between the neutral and smile expressions of the same subject, where the curves are intersecting and manually labeled. First, the source and target 3D surfaces with 21 consistent curve landmarks are mapped to the unit disk domain using CCHM. As shown in Fig. 4.4, CCHM maps curve landmarks to straight-line segments on the planar domain, while conformal maps can not. Then, with the CCHM domains, the one-step constrained harmonic map with the straight line constraints over the planar domains cannot guarantee the diffeomorphism, which uses the endpoints of the curve landmarks as constraints and computes the interior points of the curves sliding on the corresponding curves. As shown in Fig. 4.5(a), the one-step registration generates fold singularities near the left mouth corner (see close-up mesh view), which is visualized by the self-flipping triangles and the color encoded Beltrami coefficients μ, where $||\mu|| > 1$. Thus, the DQCM is applied to generate registration to remove the self-flips by minimizing the constrained harmonic energy iteratively. Both the CCHM and DQCM can guarantee the diffeomorphism of the resulted maps. Figure 3b shows the smooth evolution of landmark curves during the optimization, demonstrating both the energy and geometry very smoothly during the whole process. Fig. 4.6 illustrates the bijectivity between source and target surfaces by consistent circle-packing texture mappings and demonstrates the registration quality of our method. Fig. 4.7 gives another example to register two facial surfaces from BU-3DFE database [YWS$^+$06] with neutral expression from different subjects, where the 11 curve landmarks are disjoint and are computed automatically by connecting the given landmark points using shortest paths. The consistent texture mapping results also visualize the registration. Also, we test our registration method on the human brain surface by using two disjoint sulci as the landmark curves (see Fig. 4.9), which has big geometry variance, and the deformation is quasiconformal. The color transfer result can visually check the registration effects.

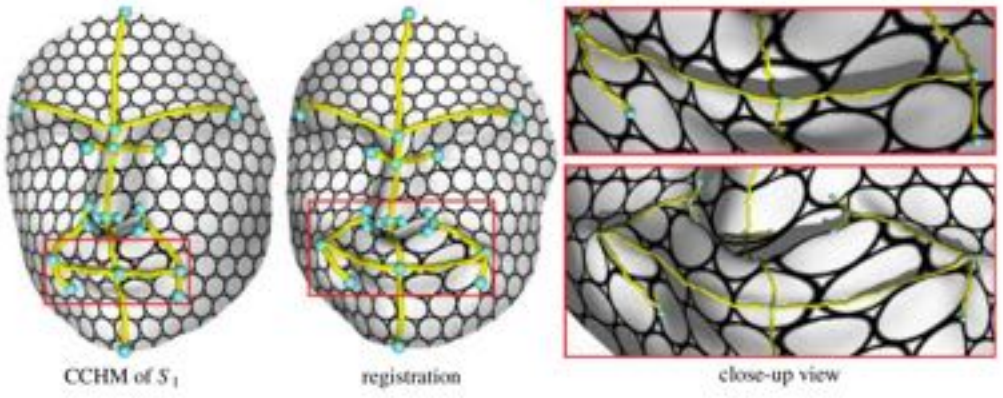


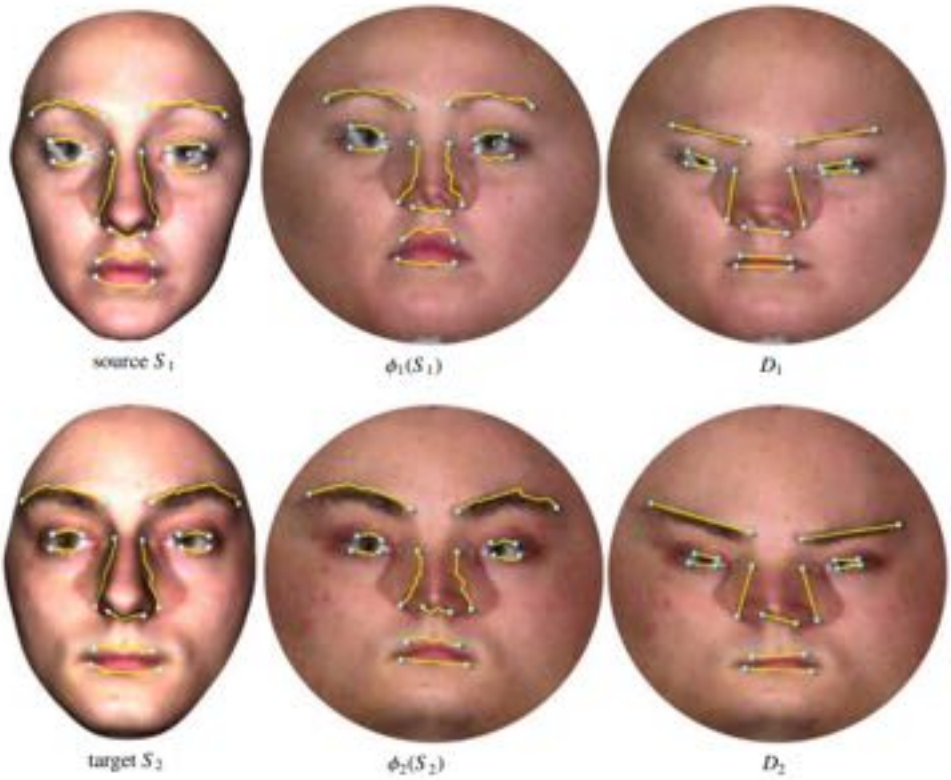
Figure 4.6: Registration results visualized by consistent texture mappings for surfaces.

Surface Matching

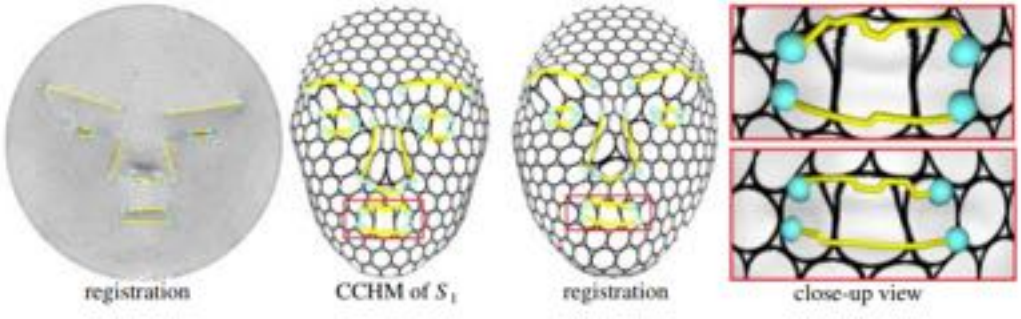
Our method offers an intrinsic canonical shape representation for surfaces with landmark curves. We employ the positions of endpoints of the landmark curves as the shape signature. The signature is aware of feature details and has a powerful discriminative ability for surfaces decorated with curve landmarks. Our experiment employed the signature for shape matching on the human facial surfaces from different subjects (BU-3DFE database, 100 subjects with neutral and other six expression types, each type contain four intensities). For faces, we use the unit disk as the parameter domain and normalize it by rotating the disk such that the lines between the two eye corners are horizontal (see Fig. 4.8). The signature vector computed above is used as input parameters to the SVM method to classify the face expression. It achieves a recognition accuracy of 90.41%, which has a large potential to be further promoted by using more related landmark curves and CNN methods.

Performance Discussion

Diffeomorphism Guarantee. The resultant registration map f is guaranteed to be diffeomorphic. First, the curve constrained harmonic maps $\phi_i, i = 1, 2$ is proved to be bijective. Second, the registration map between the 2D domains D_1 and D_2 successfully introduces



(a) conformal map and CCHM



(b) registration using dynamic quasiconformal map

Figure 4.7: Facial surface registration of neutral expression from different subjects in BU-3DFE database

Figure 4.8: Expression set from the same subject in the BU-3DFE database.

Figure 4.9: Brain registration.

	FFSS	FFDS	BRAIN
point based [LWZ+12]	0.00619	0.02146	0.01026
curve based [ZMLG14]	0.00389	0.01367	0.00901
our method	0.00323	0.01153	0.00822

Table 4.1: Accuracy Comparison

the combinatorial freedom to avoid the fold singularities. The triangles, which are the potential to generate flips, are removed by the diagonal switches. As a result, the combination of maps $\phi_2^{-1} \circ \widetilde{f} \circ \phi_1$ is a diffeomorphic map.

Registration Accuracy. Our method generates surface registrations that strictly align the point and curve landmarks, while the point-based methods in [LGYL13, ZG11, ZMLG14] can only enforce the alignment on the point set. However, the intervals between point samples on the curves can not be guaranteed to be exactly aligned. The traditional strategy to handle curve constraints is to convert curves to points, while dense sampling will introduce more correspondence computation, increase the time cost and lower the registration accuracy. In addition, Our method can exactly align the curves, which have been demonstrated in Fig. 4.5 (FFSS), Fig. 4.7(b) (FFDS) and Fig. 4.9 (BRAIN) by consistent texture mappings and color transfer result. Numerically, we compute the registration accuracy metric as $d(M_1, M_2) = \frac{1}{n}\sum_i^n \|\mathbf{g}(v_i) - \mathbf{g}(f(v_i))\|^2$, where $f(v_i)$ is the corresponding vertex in the deformed source mesh and g denotes the Gauss curvature. Table 4.1 shows the comparison between our method and point-based method (point sampling on the land-

mark curves) and curve-based method with rectilinear constraints, which demonstrates the superiority of our registration method. Compared with the previous methods, our method generates more accurate registration results for the 3D surfaces in Fig. 4.5, 4.7 and 4.9.

Efficiency. Our method is efficient, which includes two parts: (1) the curve constrained harmonic map, which is solved using a sparse linear system and has linear complexity; and (2) the dynamic quasiconformal map, which usually converges in several iterations and each iteration has the worst time complexity of O(nlgn) (vertex number n). In practice, the diagonal switches usually occur locally near the landmarks, and the time cost is far less than the worst case. For the above examples with around 10k vertices, our algorithm generates surface registration within one minute on a PC with Intel Duo CPU 3.06 GHz, 2G Memory.

Robustness. In all our examples, our method takes less than ten times to converge to the global minima. At the same time, our method is rigorous and has a solid theory guarantee to converge. To demonstrate the practicality of our method, we have tested our method on 2500 individual surfaces with curve feature landmarks (four intensity of six different expressions plus the neutral expression from 100 subjects), which always generate satisfying registration results. Fig. 4.8 illustrates seven expressions from one subject in the BU-3DFE database, where the 3D surfaces with 11 disjoint landmarks as well as their straightening parameterization are given.

4.3.5 Discussion

We present a novel surface registration method for point and curve constrained surfaces based on the dynamic quasiconformal method, straightening the curves to line segment in the canonical domain. The positions and the inclining angles of the line segments are determined intrinsically by the surface geometry and the associated feature landmarks. To

overcome the shortcomings of the traditional harmonic method (may have fold singularities under feature constraints), the combinatorial freedom is introduced to the quasiconformal optimization to generate a 2D bijective map between the source and target domains consistent curve feature landmarks. The resultant registration map is easy to compute and guaranteed to be diffeomorphic. Experiments on the face and brain surfaces demonstrate the efficiency and efficacy of our method to deal with sixteen practical surfaces with point and curve feature landmarks.

4.4 Case 1: Content Controlling for MRI Image Analysis

All the content in this section has been published in [YRH+19].

4.4.1 Introduction

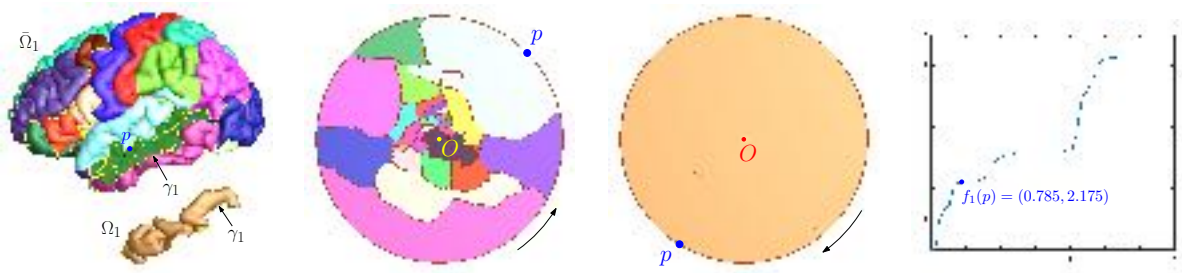
One of the large scientific challenges identified for the 21st-century concerns how the brain, body, and mind interact to produce thought, feeling, and behavior. How differences in brain structure and function contribute to differences in social and cognitive behavior are central to this endeavor. Methods with theoretical rigor, numerical accuracy, and processing efficiency to translate pictorial descriptions of cortical surfaces into quantitative mathematical descriptions are urgently needed in human brain mapping research. This work aims to explore this direction through the innovations in computational conformal geometry that quantify the relationship between shape of the brain and its functionality, here with the focus of *human intelligence*.

Previous methods : A wide number of studies have been done to correlate human intelligence with brain structure in recent decades [Hai09, YYY+13]. Most commonly used properties to represent brain structure include *density*, *area*, *thickness*, and *curvature*,

computed based on voxel-based morphometry (VBM) analysis methods (see [AF00] for a review).

In detail, Jung et al. [JH07] linked differences in grey matter density to human intelligence quotient (IQ). Despite the important advances made possible by VBM analysis, much information is lost in these analyses, and the steps necessary for the analysis can lead to artifacts. Haier et al. [HST+92] showed an inverse correlation between IQ based on Raven's advanced progressive matrices (RAPM) and glucose metabolism in areas around the cortex. Shaw et al. [SGL+06] showed that IQ is most closely related to the trajectory of change in the cerebral cortex thickness. Narr et al. [NWT+06] suggested that variation in the thickness of prefrontal and temporal association cortices is specifically relevant to IQ based on the Wechsler adult intelligence scale (WAIS). Luders et al. [LNB+07b] found a significant positive correlation between intelligence and the corpus callosum thickness, corresponding to posterior body, isthmus, and anterior sections of the splenium. Im et al. [ILY+06] found that IQ (WAIS) was correlated with fractal dimension positively, which represents the cortical complexity, and that the correlation is significantly positive in the right hemisphere. Karama et al. [KADH+09] reported the relationship between the cortical thickness of multimodal association areas and cognitive ability factor (an estimate of general intelligence, derived from adjusted WAIS).

Luders et al. [LNB+07a] found the correlation of IQ (WAIS) with cortical convolution based on mean curvature of lateral and medial surfaces of each cortical model. They found IQ positively associated with the degree of folding in the temporo-occipital lobe, particularly in the outermost section of the posterior cingulate gyrus. In addition, Yang et al. [YYY+13] proposed the combination of different morphometric properties of a complex cortical surface with cortical thickness, surface area, sulcal depth, and absolute mean curvature using partial least squares regression can be used to predict 30% of IQ (WAIS). Shape-based morphometry analysis is another category where, instead of curvatures, shape



(a) partition Ω_1, $\bar{\Omega}_1$ (b) disk $\bar{\mathbb{D}}_1$ (c) disk $\mathbb{D}_1$ (d) signature f_1

Figure 4.10: Conformal welding signature for a cortical region.

descriptors of surface geometry can be applied. Su et al. [SZW+15] used the Wasserstein distance based on the optimal mass transport theory to classify IQ (RAPM).

Our method : With the advancement of computational conformal geometry, we can explicitly compute the shape of the brain by rigorous and effective methods to quantitatively measure the similarities and classify and differentiate the complicated cortical shapes.

In this work, we propose a geometric representation, *Conformal Welding Signature*, to describe the global structure of the 3D cortical surface and characterize shape differences between 3D cortical surfaces, and then based on this, we explore how the brain shape differences contribute to the differences of intelligence. Discrete conformal welding theory was initially proposed in [SM04] for 2D simply-connected domain for shape analysis in computer vision, and later was generalized to multiply-connected domains [LZYG14, ZSW+13] for shape classification. The corresponding conformal welding signatures were computed for the non-intersecting regions of interest (ROIs) on a domain. We extract a signature for every cortical region in our method and combine all-region signatures as the signature for the whole cortical surface.

As shown in Fig. 4.10, one cortical surface can be partitioned into different anatomical regions. The boundary of a cortical region is a closed-loop, which separates the cortical surface into two connected components: the region and its complement. The region can be mapped onto the planar unit disk by a Riemann mapping, so is its complement. The

boundary loop of the region is mapped to the circular boundary of both disks. This work induces a mapping between the two circles, which is the conformal welding signature. The conformal welding signature for a region is determined by the region's geometry, the geometry of the complement on the cortical surface, and the relative geometric relationship between the region and its complement; and its conformal welding signature can recover vice versa, the contour of the region. We compute all-region signatures and combine them. By comparing the conformal welding signatures, we can determine the shape distance of regions or whole surfaces across different brains and further use that to discover its relationship to human intelligence.

Contribution : This work presents a theoretically sound geometric approach to discovering brain structure-function relationships. The *novelty* of this work is to compute the conformal welding signature for a closed genus zero surface with an atlas graph by combining all atlas region signatures and apply the conformal welding signature to analyze how brain structure correlates to human intelligence. We found that the conformal welding signature can classify human intelligence (IQ) through experiments on real data set, with a more competitive classification accuracy rate than traditional features.

4.4.2 Our Conformal Welding Signature

Here we describe the major concepts and theorem for the proposed method. Readers may refer to [GL99] for more details. We have given a genus zero closed surface S with an atlas graph. Suppose the graph regions have the boundaries, $\Gamma = \{\gamma_0, \gamma_1, \cdots, \gamma_n\}$, which is a set of simple closed curves on S. γ_i segments the surface to two connected components, Ω_k (the foreground domain, contoured by γ_k) and $\bar{\Omega}_k$ (the complement background domain), $0 \leq k \leq n$. Construct the uniformization mapping $\varphi_k : \Omega_k \to \mathbb{D}_k$ to map the foreground segment Ω_k to a circle domain $\mathbb{D}_k$, and similarly, $\bar{\varphi}_k : \bar{\Omega}_k \to \bar{\mathbb{D}}_k$ map the background

segment $\bar{\Omega}_k$ to a disk domain $\bar{\mathbb{D}}_k$. Let $f_k|_{\mathbb{S}^1} := \varphi_k \circ \bar{\varphi}_k^{-1}|_{\mathbb{S}^1} : \mathbb{S}^1 \to \mathbb{S}^1$ be the diffeomorphism from the circle to itself, and $f_k : [0, 2\pi] \to [0, 2\pi]$. We call the diffeomorphism f_k the *signature of* γ_k. The *conformal welding signature* of the family of non-intersecting closed curves Γ on a genus zero closed surface can be defined as:

$$\mathcal{W}(\Gamma) := \{f_0, \cdots, f_k\}.$$

The conformal welding theory [GL99] guarantees that the signature is determined by a family of curves unique up to a Möbius transformation, and inversely that the curves can be uniquely recovered by the signature unique up to a conformal transformation.

As shown in Fig. 4.10, the contour γ_1 in (a) is mapped to the circles of $\bar{\mathbb{D}}_1, \mathbb{D}_1$ in (b-c). The diffeomorphism $f_1 : \partial\bar{\mathbb{D}}_1 \to \partial\mathbb{D}_1$ induces the signature for the region, plotted as a monotonically increasing curve in (d). For example, given a point p on the boundary and compute the angles from that. Note that because two domains are obtained by slicing the common boundary open on the original surface, the boundaries of the two disk domains have opposite orientations, as shown in (b-c).

Concept novelty : Existing conformal welding signatures [SM04, LZYG14, ZSW$^+$13] considers a closed genus zero surface with a family of non-intersecting closed curves $\Gamma = \{\gamma_0, \gamma_1, \cdots, \gamma_n\}$. They include not only the diffeomorphisms f_k generated by all curves and also the conformal module of the complement background domain $\bar{\mathbb{D}}$, which is a poly-annulus obtained by cutting all non-intersecting regions off the closed surface. The uniformization mapping result of the $\bar{\mathbb{D}}$ is a circle domain, i.e., a disk domain with circular holes. The conformal module of $\bar{\mathbb{D}}$ is defined as the combination of the circle centers and radii of the circle domain. The signature is then defined as $\mathcal{W}(\Gamma) := \{f_0, \cdots, f_k\} \cup \{Mod(\bar{\mathbb{D}})\}$, where Mod denotes the conformal module of $\bar{\mathbb{D}}$. In our case, we propose a novel way to compute the signature for a genus zero closed surface associated with an atlas graph on that. We consider the contours of all atlas regions. Its complement

background domain is a topological disk for each region and mapped to a unit disk, where the conformal module is ignored. Our conformal welding signature describes the shape and the correlation of the whole atlas structure and the surface.

Geometric intuition : The conformal welding signature of a cortical region is intrinsically determined by its geometry, its complement, and its relationship. Intuitively, one can glue two planar unit disks to get a closed surface. The conformal welding signature specifies the gluing pattern along the disk boundaries. If the glued shape can be conformally mapped to the original cortical surface, then the glued circles are mapped to the original region boundary. This shows that the conformal welding signature can determine the position and shape of the loop. To compare the shapes of different regions, or the corresponding regions on different cortical surfaces, or the whole cortical surfaces, one can just compare their signatures, which are in the same space of the diffeomorphisms of the unit circle and much easier to compare and manipulate.

This signature is *global* and captures the difference of *intrinsic* conformal structures. It is *rigorous, unique* and *accurate*, and is *invariant* under Möbius transformation (angle preserving); area distortions in the conformal mappings won't change the signature. Moreover, conformal mappings are robust to geometry noise, so the signature is *stable*.

4.4.3 Computational Pipeline

The pipeline includes the following four steps: (1) reconstruct cortical surfaces from MRI data and generate atlas parcellation; (2) partition anatomical regions one by one from a cortical surface; (3) compute the Riemann mappings for each region (foreground) and the corresponding complement surface (background); and (4) extract the conformal welding signature for each region and combine them to build the final signature. Details are as follows. Figure 4.11 shows the pipeline for computing the signature for one region.

1. Reconstruction and parcellation : First of all, we reconstruct the 3D meshes from the MRI brain scans. This can be done in various ways. Here, we employed FreeSurfer automated pipeline (www.freesurfer.net). It gives the desired cortical surface and the anatomical atlas parcellation encoded by different colors. Each brain hemisphere has 35 anatomical regions, indexed by integer 1, 2, 3, ..., 35. Besides all of these regions, there is a black region with the id 0, which connects left and right hemispheres. Through our experimental analysis, region 4 is totally in the interior and has no exposure on the cortical surface, therefore not considered. So there are a total of 34 regions to be computed for each cortical hemisphere. Note that we only need the cortical surfaces with the atlas under the same atlas protocol and don't need to register them. Other software tools such as BrainSuite can also work for this purpose.

2. Partition : For computing the conformal mappings of the regions, we extract 34 regions from the cortical surface one by one. The cortical surface is a genus zero surface. By slicing the cortical surface open along a region, we obtain the region surface (foreground) and the cortical surface without the region (background). Note that their boundaries are a copy of the contour of the region.

3. Mapping Brains : We run the Reimann mapping on both the foreground and the background surfaces. The computation is based on the discrete holomorphic 1-forms [LZYG14]. With the puncture, each patch is a topological annulus with an exterior boundary γ_0 and an interior boundary γ_1. In practice, the surface is tessellated as a triangle mesh $M = (V, E, F)$, where V, E, F denote the set of vertex, edge, and face, respectively.

We first compute a *harmonic function*, $f : M \to \mathbb{R}$, such that $\Delta f(v_i) = 0, \forall v_i \notin \partial M, f|_{\gamma_0} = 1, f|_{\gamma_1} = 0$. The discrete Laplace-Beltrami operator Δ acts on a function f,

$$\Delta f(v_i) = \sum_{[v_i,v_j]\in M} w_{ij}[f(v_i) - f(v_j)],$$

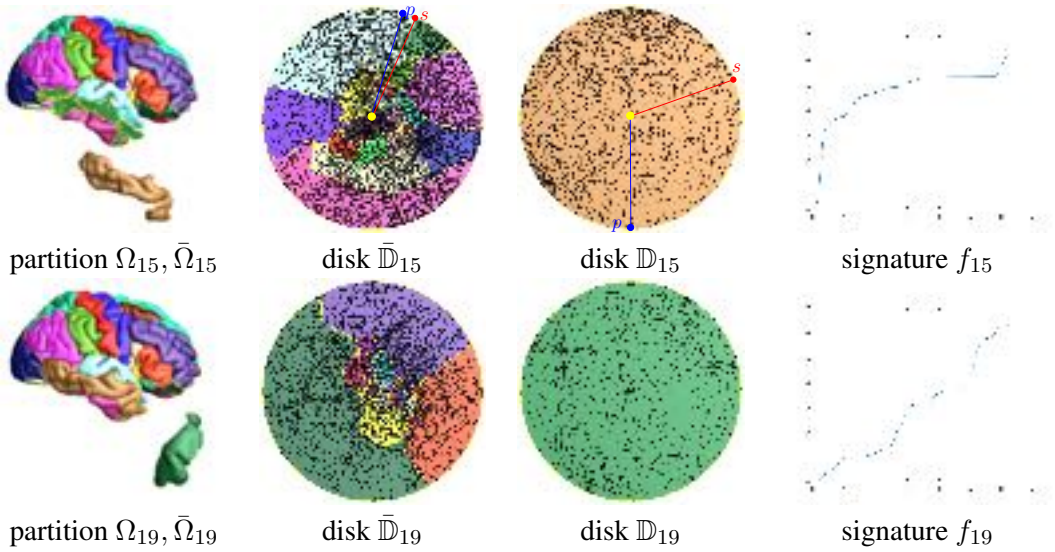
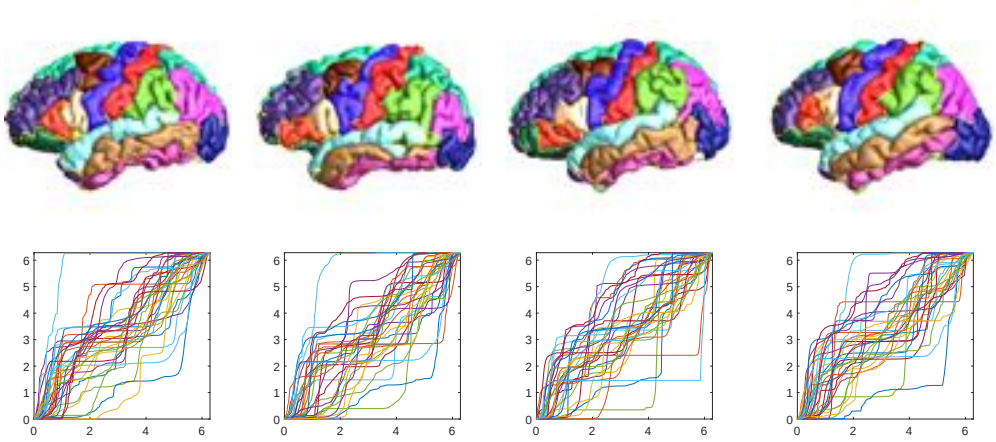


Figure 4.11: Illustration of the signature computation for two regions 15 and 19 with triangular meshes. Red point is the starting point to compute angles.

where the weight w_{ij} is chosen to be the mean value coordinates which guarantees to be positive for any triangulation cases. Then $\omega_1 = df$ is a closed 1-form. We further find the shortest path γ from γ_0 to γ_1. We slice the mesh along γ to get an open mesh $\bar{M}$, γ becomes two boundary segments γ^+ and γ^- on $\bar{M}$. We randomly assign a function $g : \bar{M} \to \mathbb{R}$, such that $g|_{\gamma^+} = 1, g|_{\gamma^-} = 0$. Then $\omega_2 = dg$ is a closed 1-form. We then find another function h, such that $\omega_2 + dh$ is harmonic for all vertices, $\sum_{[v_i,v_j]\in M} w_{ij}[\omega_2([v_i, v_j]) + h(v_i) - h(v_j)] = 0$, and update $\omega_2 = \omega_2 + dh$. Finally, we compute a constant λ, such that $\lambda\omega_2$ is as close to ${}^*\omega_1$ as possible. The discrete holomorphic 1-form $\omega = \omega_1 + i\omega_2$ is obtained.

We then integrate ω over M, $\phi(v_k) = \int_{\gamma(v_0,v_k)\in M} \omega,\ \forall v_k \in V$, where γ_k is an arbitrary path from the base vertex v_0 to the current vertex v_k and $\phi(v_0) = (0, 0)$, and compute the map $v_k \to exp(\frac{1}{T}\int_{\gamma_k} \omega)$, where T is the period $T = \frac{2\pi}{\int_{\gamma_0} \omega_2}$. Thus we obtain the conformal map, which maps the annulus M to a canonical annulus with the unit exterior radius and is independent of the choice of the path γ_k. The computation of harmonic functions is equivalent to solving linear systems and therefore is efficient.



IQ=19.4 (L), age=19 IQ=19.4 (L), age=19 IQ=80.6 (H), age=19 IQ=77.78 (H), age=19

Figure 4.12: The left brain hemispheres and their cofnormal welding signatures with different IQ levels.

4. Extracting Signatures : The contour of each region is a loop embedded in 3D space and is mapped twice to be the unit circles in the two Riemann mappings of both the foreground and background domains. Therefore we can form a diffeomorphism between the two circles since they share the same chain of vertices but in reverse order. We represent the diffeomorphism $f : \theta_0 \to \theta_1$ (θ_i is the radial angle of $\phi(\gamma_i)$) using the pair of radial angles (θ_0, θ_1), which can recover the two circles exactly. In detail, we fix one vertex on the boundary as the starting point, which corresponds to the curve endpoints, $(0, 0)$ and $(2\pi, 2\pi)$ (see Figs. 4.10 and 4.11). For the consistency of the starting point on each contour over various brains, we utilize the branching vertex of common regions along the contour as the starting point. Figure 4.12 gives examples for the brains with various IQ scores. We observe that the signatures are similar for brains with the same IQ level; moreover, there are visible differences between the two groups, showing the promising ability for IQ classification.

4.4.4 Experiments

We performed our analysis on a real data set consisting of 243 subjects, 146 females, and 97 males aged 18 to 30 (mean 20.8). MRI recording was performed using a standard 12-channel head coil on a Siemens 3T Trio MRI system with TIM. The collection of IQ is based on the online questionnaire of Raven's Advanced Progressive Matrices (RAPM) [RRC98] in Qualtrics (www.qualtrics.com). The range of IQ is [0,100], taking the value as the rate of the correct answers multiplied by 100, which is different from the traditional IQ scores using the median score of the norming sample as IQ 100. In this application, due to the size limit of the training data set, it won't be practical to predict the exact value of IQ from a brain. So we take it as a classification problem and then try to label the brains by IQ. Earlier study [Hun10] states that young human IQ has a normal distribution $N(\mu, \sigma^2)$ (μ - mean value, σ - standard deviation) and $(\mu - 0.5\sigma)$ is the segmentation line for low IQ. As shown in Fig. 4.13, our data set admits that and has a corresponding IQ score of 60 as a segmentation line. We used that to group subjects (63 subjects of Low and 180 subjects of High).

Evaluation plan : To evaluate the efficiency and the efficacy of the proposed method, two experiments were performed: (1) feature visualization, to illustrate the distribution of the features of this data set (see Sec. 4.4.4); and (2) classification, to demonstrate the ability of the features to differentiate IQs (see Sec. 4.4.4).

We used FreeSurfer automated pipeline with Desikan Killiany atlas template for parcellation. As we described in Sec. 4.4.3, we computed signatures for 34 regions for each hemisphere, totaling $34\ (regions) \times 2\ (hemispheres) = 68$ signature curves for each brain (see Fig. 4.12). All regions can be handled in parallel, and the running time for a cortical mesh with $270k$ triangles is 30 seconds. Our experiments used the area under the curve (AUC) to build a feature vector to feed the classifier. In detail, the area under

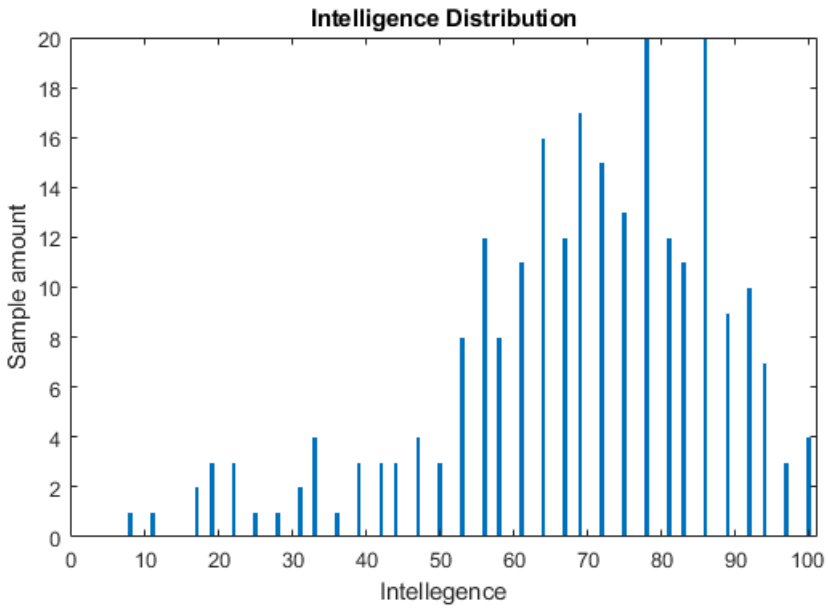
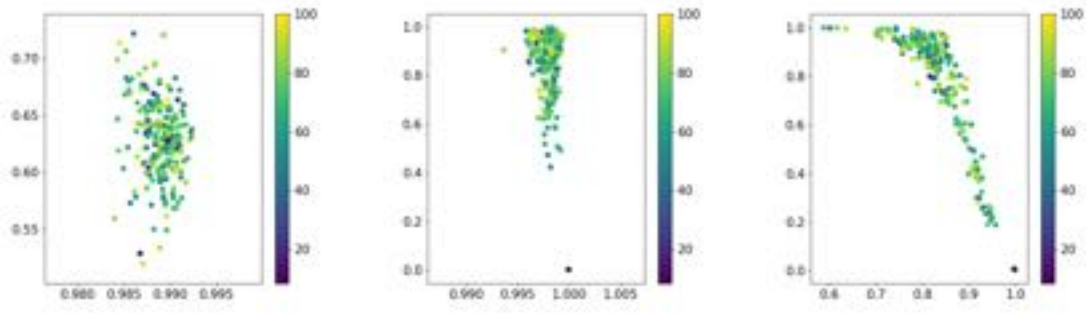


Figure 4.13: The IQ distribution in the data set.

f_i is computed as $AUC(f_i) = \Sigma_0^{2\pi} f_i(x_k)$, where x_k are the uniform samples on x-axis. Therefore, the feature vector for a brain has dimension 68.

At the same time, we compared the performance of our geometric signature with that of the FreeSurfer features for all regions [RRF10, RSRF12]. They include *CurvInd* (Curvature index), *FoldInd* (Folding index), *GausCurv* (Gaussian curvature), *GrayVol* (Volume of gray matter (surface-based)), *MeanCurv* (Mean curvature), *NumVert* (Number of vertices), *SurfArea* (Surface area), *ThickAvg* (Average of thickness), and *ThickStd* (Standard deviation of thickness in ROI). Each feature is a single value, then the feature vector for a brain has dimension 68. If combing all features, then the total feature vector for a brain has dimension 68 $\times$ 9 = 612. All experiments were conducted on a workstation with a 3.7GHz CPU and 16GB RAM. The computation is automatic, stable, and robust without human intervention.

We first visually test the distribution of the signatures and the relation to IQ. We employ the variational autoencoder (VAE) [KW13] method to compress the conformal welding signature and FreeSurfer feature to the same dimension. VAE is a kind of unsupervised



(a) 1 epoch, loss=7010 (b) 1000 epochs, loss=1210 (c) 2000 epochs, loss=264

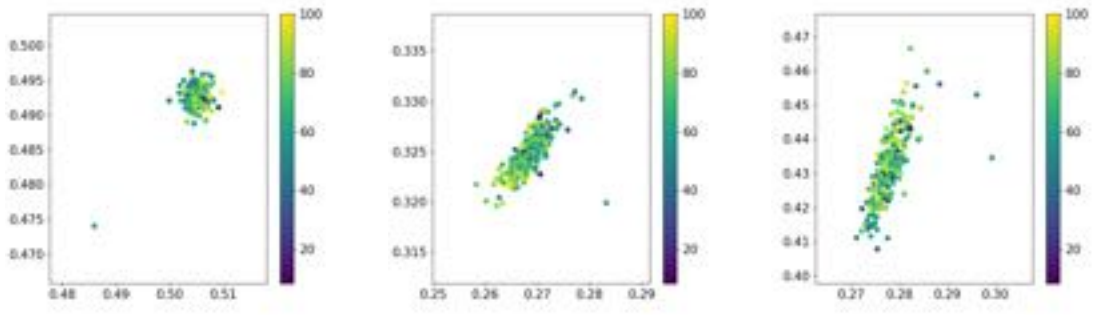
Figure 4.14: The IQ distribution based on encoded conformal welding signature.

neural network, and there is no need to provide a label of IQ. Because the input is composed of continuous values, we define the loss function of VAE as

$$loss = \sum_{i=1}^{n} (x^{(i)} - \hat{x}^{(i)})^2 + \frac{1}{2} \sum_{j=1}^{J} (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2),$$

where $x^{(i)}$ is the i^{th} real data for each batch and the $\hat{x}^{(i)}$ is the i^{th} output result of the decoder for each batch, μ_j is the j^{th} elements in output of z_{mean} layer for each samples and σ_j is the j^{th} elements in output of z_{log} layer for each samples. In practice, we only use the z_{mean} as the code of the original data. For the training process, all samples are trained in one batch, which can avoid the random error in the loss decreasing process of the neural network. The VAE is trained with 100 iterations each epoch, and the learning rate is 0.00001. Figure 4.14 shows the distribution of the conformal welding signatures of the whole brain data set with different epochs. Each point in the plot represents a sample, the position is the encoded feature value, and the color encodes the IQ value of the sample, as shown in the color bar. We can see that as the loss decreases, the encoding result shows a gradually clustered distribution.

comparison with freeSurfer features : We applied the same method to visualize the distribution of the traditional features provided by FreeSurfer. Figure 4.15 demonstrates



(a) 1 epoch, loss=167 (b) 1000 epochs, loss=2.13 (c) 2000 epochs, loss=0.55

Figure 4.15: The IQ distribution based on encoded region features provided by FreeSurfer.

that the combination of the 9 FreeSurfer features could not give obvious clusters compared to the conformal welding feature. These visualization experiments imply that our geometric features are more closely related to the IQ than those traditional features. To analyze the output of VAE numerically, we used IQ as the group mark to calculate the silhouettes scores [Rou87] of VAEs' outputs, which is a useful metric to evaluate the clustering result. The silhouettes score is defined as

where $a(i)$ is the average distance among samples in the i^{th} group, $b(i)$ is the average distance between samples in the i^{th} group and samples out of the i^{th} group. And here, we use IQ as the group mark. The silhouette score has the range $[-1, 1]$. The greater silhouette score denotes the better clustering effect.

Figure 4.16 shows that as the loss decreases, more unrelated information is involved, and the silhouette score drops gradually; the silhouettes score of conformal welding signature becomes higher than FreeSurfer's, which means that the clustering effect of conformal welding signature is stronger. Therefore, conformal welding signature is more suitable than FreeSurfer's feature to classify IQ.

We numerically tested the ability of our method to classify the IQs into two groups. In our experiments, to balance the data set and increase the sensitivity of a classifier to the minority class, we used the Synthetic Minority Over-Sampling Technique (SMOTE)

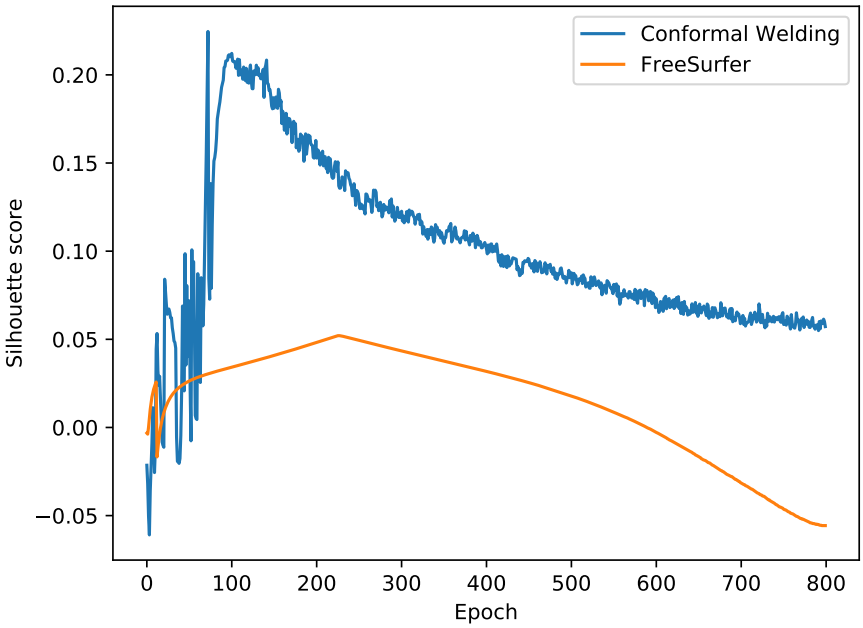


Figure 4.16: Silhouette Coefficient comparison between features based on conformal welding signature curves and features provided by FreeSurfer.

[CBHK02] for data argumentation. We applied the support vector machine (SVM) with linear kernel function (using LIBSVM, www.csie.ntu.edu.tw/ cjlin/libsvm/) as a classifier 5-fold cross-validation. For all tests, 70% of the whole data set is randomly chosen as the training set to prevent bias (the resting as the testing set). We also computed the receiver operating characteristic curve (ROC) to evaluate the classifier. The classification accuracy rate is 81.44% and the AUC of ROC is 0.86926 (see Fig. 4.17). This result demonstrates that our signature is effective for IQ classification.

Biological finding for IQ : We further tested the contribution of each region to the classification result using the infinite latent feature selection method in [RMCV17]. We found that the correlation is significantly positive in the right hemisphere (see Fig. 4.17), which is consistent with the finding in [ILY$^+$06]. The summation of the weights for the left vs. right hemisphere is 0.47 vs. 0.53. The top three significantly positive features are

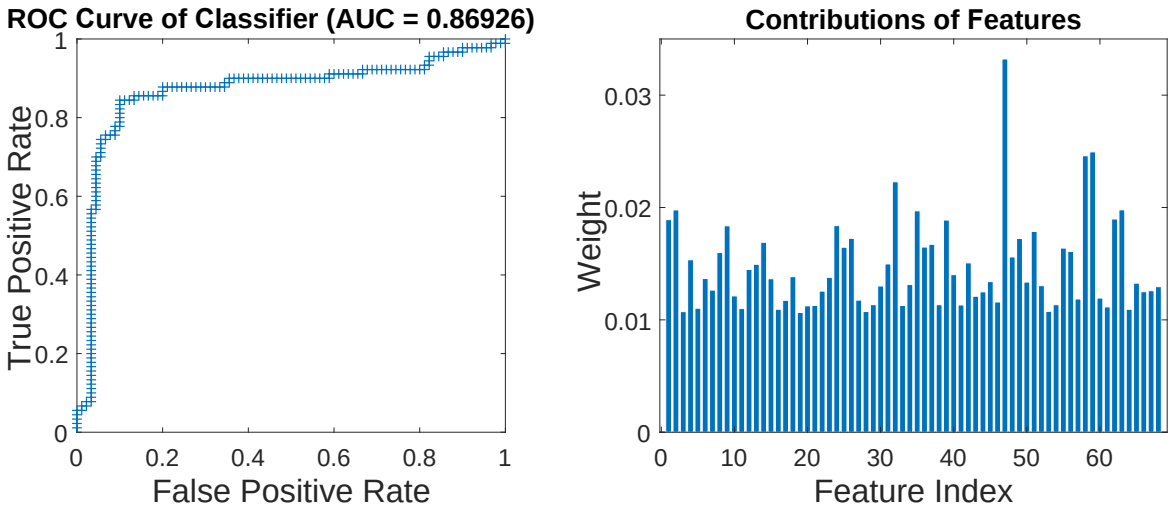


Figure 4.17: Classification results. Left: The receiver operating characteristic curve of our classifier; Right: The contribution of each entry (region) in the feature vector.

Table 4.2: Classification rates of our signature and traditional features.

Method	Rate%
Conformal Welding (ours)	**81.44**
CurvInd	49.31
FoldInd	46.15
GausCurv	48.89
GrayVol	53.11
MeanCurv	60.71
NumVert	48.12
SurfArea	48.55
ThickAvg	73.11
ThickStd	61.20
All 9 above	70.10

entries 47 (the medial orbitofrontal, region 14), 58 (the precuneus, region 25) and 59 (the rostral anterior cingulate, region 26), with weights 0.033, 0.025 and 0.026, respectively.

comparison with literature : In computation, we compared our method with the existing methods for intelligence classification in terms of the accuracy rate given in Table 4.2. With the same experimental configuration and data set, our method performs better than the others. In theory, our method based on conformal welding theory has the advantages of theoretical rigor and computational efficiency. The conformal welding signature is achieved by solving *sparse linear* systems, much more efficient than the nonlinear

Wasserstein distance method based on optimal mass transport theory [SZW$^+$15]. It is invariant to conformal transformations (subsuming rigid motions, scalings, and isometry); the Wasserstein distance is invariant under rigid motions and scalings. The current work computes a novel conformal welding signature, especially for all atlas regions covering the whole surface and for analyzing intelligence. In contrast, the works [ZSW$^+$13, LZYG14] computed non-intersecting regions of interest for medical image analysis and disease diagnosis.

4.4.5 Discussion

Applicability and impact : The conformal welding representation of the brain is fundamental, which can be used to discover the correlations of brain structure with other functionalities, such as well-being, personality, and an autism spectrum disorder. It can be explored on other human organs in medical imaging and cognitive neuroscience, such as human faces and colon walls, with the interest of regions (e.g., abnormality areas).

Limitations : Although conformal welding is a trustworthy tool. The signature-based on it is global (see Fig. 4.12), intrinsic to surface and curve geometry, and invariant to conformal transformations. The computation is efficient through solving sparse linear systems, we cannot use this feature to estimate the IQ of human accurately. There are three main reasons.

1. Limitation of the dataset. Our dataset contains 243 samples that can be used to estimate the relationship between the MRI image features to human IQ.
2. Limitation caused by the label. As we all know, the IQ of a human is not an accurate value. It is just the score of some kinds of test. Even though the test is widely used, we cannot guarantee that the result of one human is static and accurate. The label itself is vague, which means the label corruption is not inevitable.

(a) Level-0 (b) Level-1 (c) Level-2 (d) Level-1f (e) Level-2f

Figure 4.18: Multi-level abstract datasets. These dataset are generated based on the rest of information quantity. Without considering the role of clothes, for the information quantity, Level-0 $>$ Level-1$>$Level-2.

3. Limitation of the network scale. A huge network with high representation capability is necessary to estimate the relationship between human MRI features to their IQ based on the question's complexity.

4.5 Case 2: Train the DNN with Abstract Images

4.5.1 Introduction

This work verifies that the abstract images can also be used in the network's training. Using the data with reasonable abstraction will not affect the performance of the model after training. It can reduce the size of the dataset and avoid models' cheating in the learning process effectively. It is a successful example of content control.

4.5.2 Basis of Abstraction

To eliminate the "shortcut" in the dataset, we try to minimize the quantity of information in the input image based on not affecting model training as much as possible. To generate the abstract data, we need to know what kind of information is useful.

One of the abstract bases is based on the entropy analysis method. In [MZZZ19, GWZ+19], the author proposes a method to quantify the input information that is encoded in a specific intermediate layer of a DNN. The entropy measures how much input information is neglected when the DNN extracted the feature of this layer. We can use the low entropy part to visualize the region with more information. The information discarding is formulated as the conditional entropy $H(X')$ of the input, given the intermediate-layer feature $f^* = f(x)$, as Eq. 4.9 shows,

$$H\left(X'\right) \text{ s.t. } \forall x' \in X', \quad \left\|f\left(x'\right) - f^*\right\|^2 \leq \tau, \tag{4.9}$$

where X' denotes a set of images which correspond to the concept of a specific object instance, τ is a small positive value which represent the tolerance. In [GWZ+19, MZZZ19], x' is assumed following an *i.i.d* Gaussian distribution, $x' \sim \mathcal{N}(x, \Sigma)$, where the Σ is the covariance matrix. To reduce the computing complexity, Σ is simplified as $diag\left(\sigma_1^2, \ldots, \sigma_n^2\right)$ where n is the pixel amount of input image and $\sigma_i^2 = \mathrm{E}[(x_i - \mathrm{E}\left[x_i\right])^2]$. In this way, the assumption of the Gaussian distribution ensures that the entropy $H(X')$ of the entire image can be decomposed into pixel-level entropies $\{H_i\}$ as follows.

$$H\left(X'\right) = \sum_{i=1}^{n} H_i \tag{4.10}$$

where $H_i = \log \sigma_i + \frac{1}{2}\log(2\pi e)$. This entropy is called 1d entropy when H_i is the entropy of one pixel. For a colorful image as the left one shown in Fig.4.19, the entropy pattern is complex, which can provide more features to the network. However, for a binarized image, the pattern of entropy is much simpler than, which can provide fewer features to the network. Therefore, for images with the same content, **the binarized image contains less information than the colorful one** (Cor. 4.5.1) as Fig. 4.19 shows.

Corollary 4.5.1 *For the image with the same content, the binary image contains less information than the colorful one.*

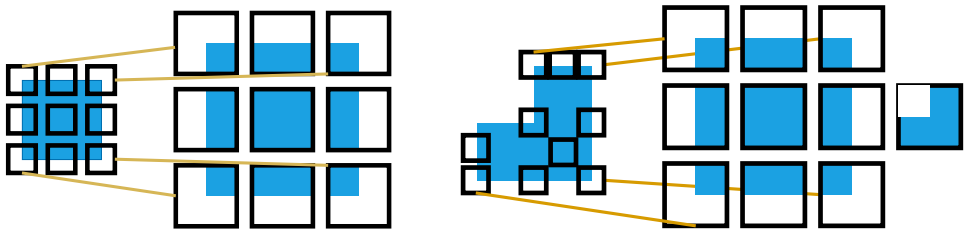
(a) Colorful Image (b) Binary Image

Figure 4.19: Comparing with the colorful image, the the binary images has lower 1d entropy expectation.

One step further, we can expand the definition of Eq. 4.10 by the concept of the "superpixel." Here, the pixel is not a point, but a set contains one pixel and its neighbors, which the radium can define. We can use a tuple (i, j) to briefly describe a superpixel, where i is the pixel value of the center pixel and j is the average value of all its neighbors. For Eq. 4.10, if the object is superpixel, the result is 2d entropy. Based on the definition, we can infer that for two images with the same content, **the simple boundary contains less information than the complex one** (Cor. 4.5.2). For example, in Fig. 4.20, we use the box with four pixels to sample the shape. We can get nine kinds of samples from the left, and for the right with a more complex boundary, we can get ten.

Corollary 4.5.2 *For the image with the same content, the simple boundary contains less information than the complex one.*

Another abstract basis is based on the results of knowledge visualization of CNN. In[ZCS+18], the author provides a method to visualize the pattern learned by the CNN. We use it to analyze our dataset and draw the heat map of inference score (see Fig. 4.21).



(a) Simple Boundary (b) Complex Boundary

Figure 4.20: Comparing with the shape with complex boundary, the shape with simple boundary contains less information.

Comparing with the details of the image, the model seems more sensitive to the boundary of the region, which indicates that **the information related to the boundary of the region plays an important role in the network's training** (Cor. 4.5.3).

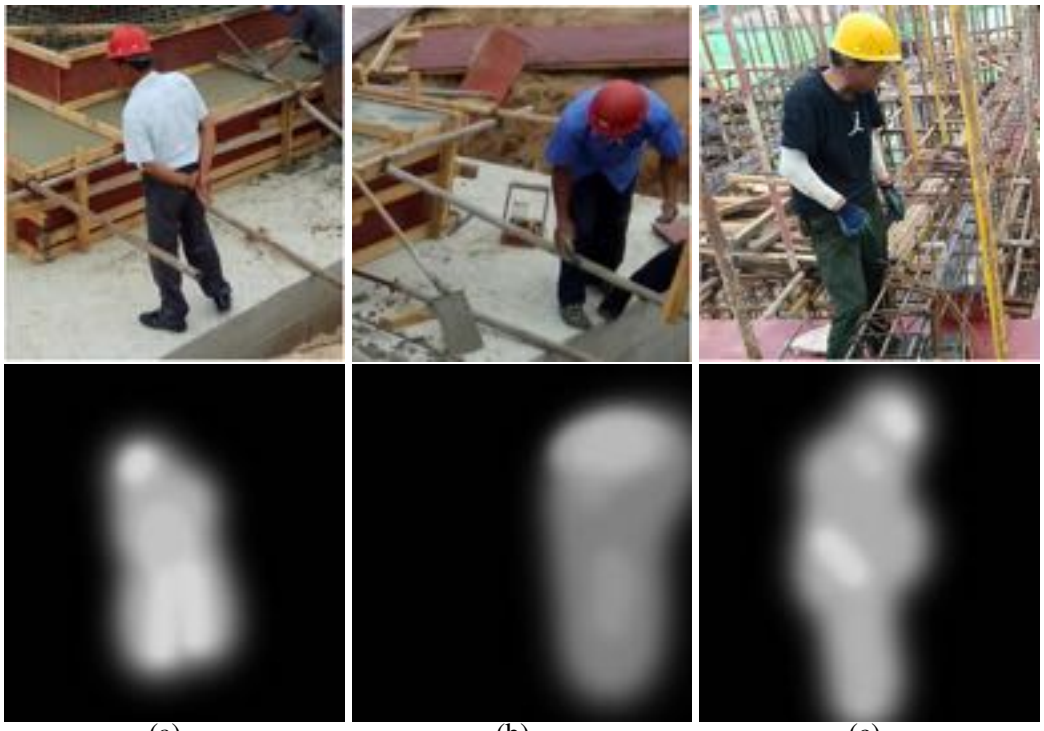
Corollary 4.5.3 *The information related to the boundary is significant in networks' training.*

4.5.3 Multi-level Abstraction

Based on the corollaries mentioned above, we abstract the samples at five levels as Fig. 4.18 shows. To measure the information of images, we calculate 1d entropy and use the pixel and its 8-neighborhoods to calculate 2d entropy based on Eq. 4.10. In our experiments, (H_{1d}, H_{2d}) is used to represent the information contains by the images.

Level-0 : Level-0 data is the images collected from the construction site directly without changing, whose average entropy is $(5.76, 11.85)$ (see Fig. 4.18(a)).

Level-1 : Based on Cor. 4.5.1, removing the information hidden in the colorful pixels can reduce the information provided by the image. And based on Cor. 4.5.3, the boundary



(a) (b) (c)

Figure 4.21: The boundary of the region plays a significant role in the network's training.

of the region needs to be kept. Therefore, we use the silhouette of the original image as the level-1 abstraction whose average entropy is $(0.56, 1.28)$ (see Fig. 4.18(b)).

Level-2 : Based on Cor. 4.5.2, we further simplify the information stored in the boundary. Human pose detection requires a model to represent the human pose, and we want to simplify the boundary of the model as much as possible. There are two kinds of components in the stick-man model, as Fig. 4.22 shows, and its boundary is much simpler than the silhouette (circular border and straight border). Therefore, we use the stick-man to represent the human pose in the images as level-2 abstraction whose average entropy is $(0.48, 0.96)$ (see Fig. 4.18(c)).

Level-1f and level-2f : As Fig. 4.21 shows, the hat and cloth whose color contrasting with the background are obvious in the heat map of inference score, which means they play

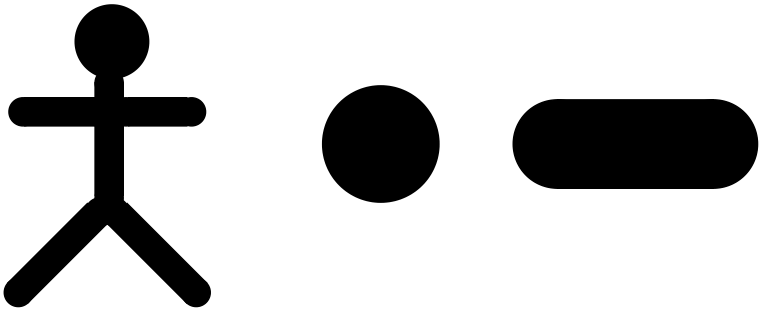


Figure 4.22: Stick-man model and its components. The boundary is simpler than the silhouette without losing semantics.

a more important role in the classification of the human pose. Moreover, [GRM+18] points out that CNN-based models are more interested in the texture. To explore the importance of these features in network training, we add these feature on level-1 and level-2 data to get level-1f and level-2f whose average entropy are $(0.95, 2.12)$ and $(0.76, 1.38)$ (see Fig. 4.18(d) and Fig. 4.18(e)).

4.5.4 Experiments

Network and Dataset

The experiments are based on a personalized Mask-RCNN based on [Abd17]. To make our experiments easily reproducible, we change the default settings of Mask-RCNN provided by [Abd17] as little as possible and use Colab [Bis19] with GPU acceleration as our experiment environment. For the dataset, the details of the distribution are shown in Tab 4.3. Some of the samples of these five datasets are shown in Fig. 4.23, 4.24, 4.25, 4.26 and 4.27.

We select pre-trained ResNet-101 as the backbone. There are two image dataset, ImageNet [RDS+15] and COCO [LMB+14] which are used to pre-train the backbone. The accuracy of the COCO-based model is 89.48%, and the accuracy of the ImageNet-based

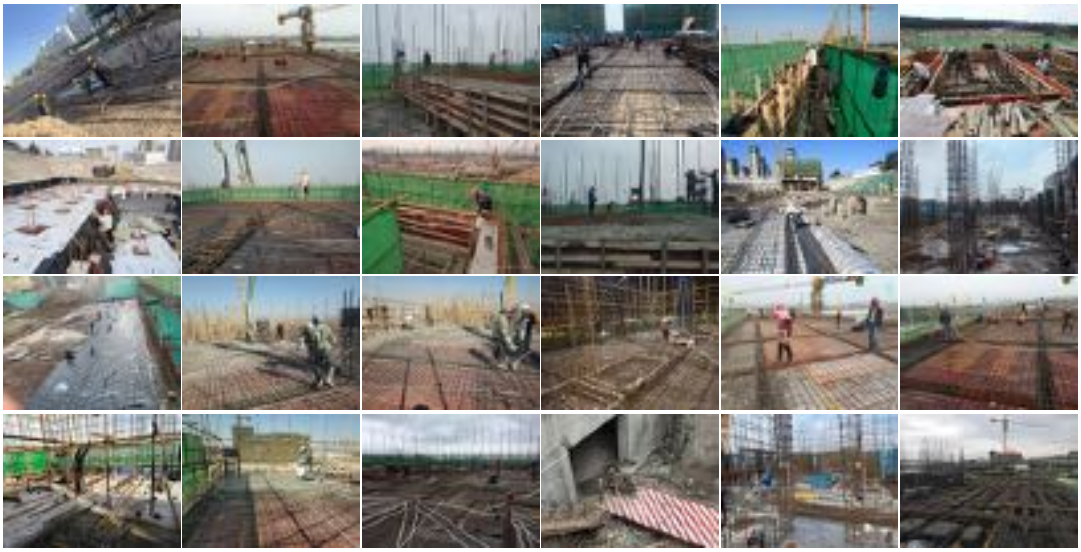


Figure 4.23: Examples of Level-0 data. The samples inside are the original images collected from the construction site.

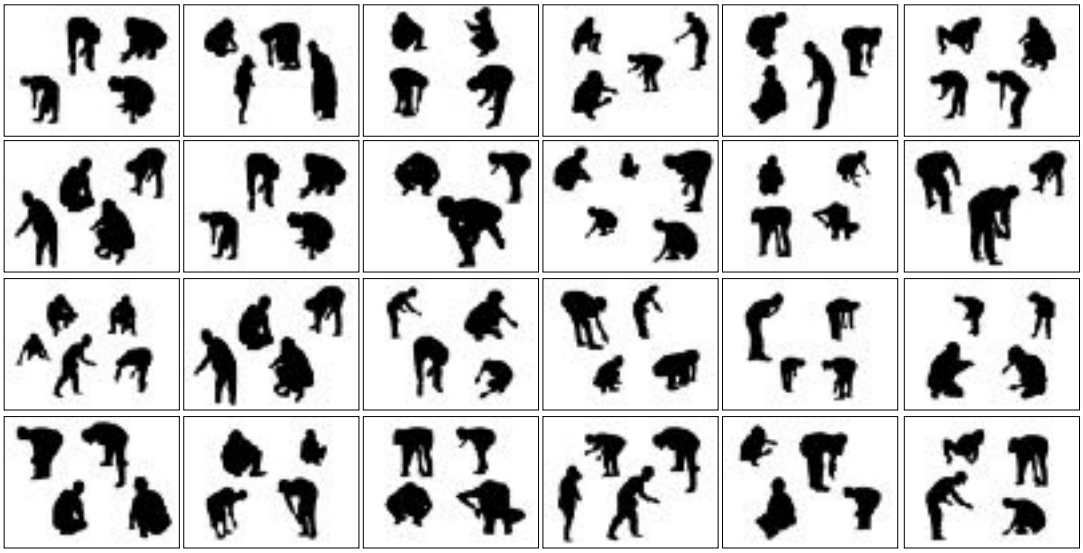


Figure 4.24: Examples of Level-1 data. The samples inside are generated based on images in Level-0 dataset. The human pose is represented by the corresponding silhouette, and extra feature of clothes is discarded.

Figure 4.25: Examples of Level-1f data. The samples inside are generated based on images in Level-0 dataset. The human pose is represented by the corresponding silhouette, and extra feature of clothes is added correspondingly.

Figure 4.26: Examples of Level-2 data.The samples inside are generated based on images in Level-0 dataset. The human pose is represented by the stickman model , and extra feature of clothes is discarded.

	Dataset	Bend	Squat	Stand	Scenes
Train	Level-0	88	209	582	240
	Level-1	269	282	434	240
	Level-2	282	318	274	240
	Level-1f	269	282	434	240
	Level-2f	282	318	274	240
Test	Level-0	26	54	216	80
	Level-1	85	64	191	80
	Level-2	69	97	110	80
	Level-1f	85	64	191	80
	Level-2f	69	97	110	80

Table 4.3: Content of the datasets.

learning rate	Optimizer	epochs	batch size
0.01 (with decay)	SGD [SMDH13]	150	2

Table 4.4: Basic training configuration.

model is 47.8%. In the following part, to control the variable, all the model's backbone is pre-trained by the COCO.

Experiment and analysis

In the experiment, we train five models from scratch to fine-tuned with the same configuration as Tab. 4.4 shows and test their performance on a test dataset of workers. As Fig. 4.28 shows, the models' performances trained by level-1 and level-1f are very close to the model trained by the real data (level-0), which means the level-1 based abstraction is effective. Compared with the level-1 data, the model trained by level-1f is a bit worse, which means that the feature cannot help the network improve its performance in this level of abstraction. On the contrary, compared with level-2, the model trained by level-2f performs better, which means the feature helps the model classify the human pose. We can infer that for a specific kind of feature, its effectiveness could be different in a different situation.

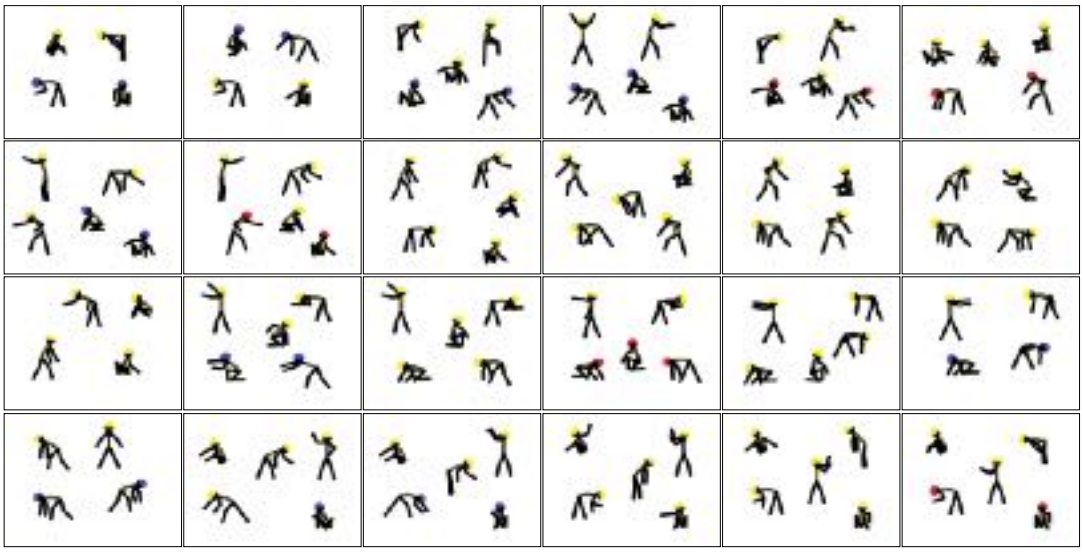


Figure 4.27: Examples of Level-2f data. The samples inside are generated based on images in Level-0 dataset. The human pose is represented by the stickman model , and extra feature of clothes is added correspondingly.

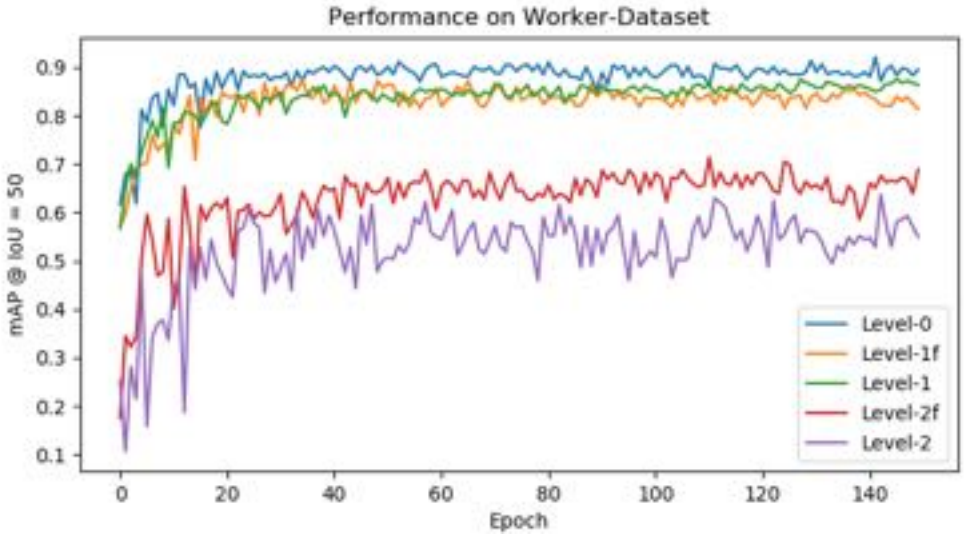


Figure 4.28: Performance on worker dataset. Compared with the level-0 data, the level-1 and level-1f shows competitive effect in training.

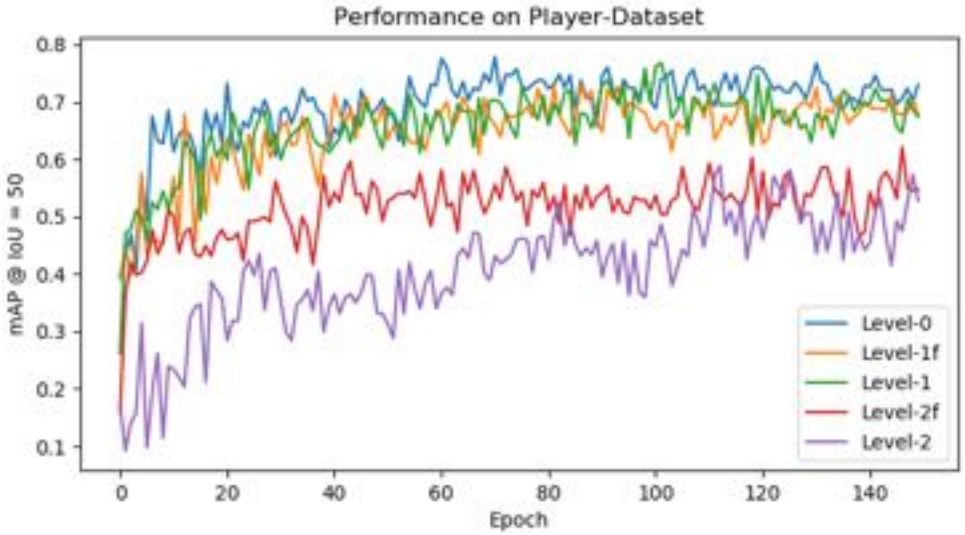


Figure 4.29: Performance on the player dataset. Compared with corresponding performance on the dataset of workers, the accuracy generally drops, which indicates that the "shortcuts" are not eliminated thoroughly.

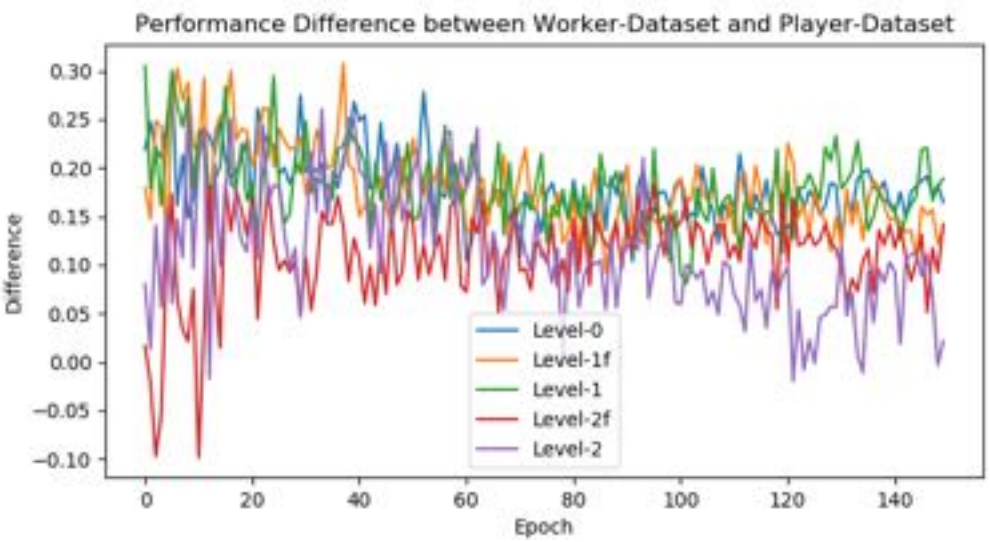


Figure 4.30: The accuracy decrease. Compared with the accuracy decrease of level-0 data, the accuracy decrease of abstract image datasets is generally lower, proving the feasibility of our method.

Dataset	Accuracy Decreasing
Level-0	0.1821
Level-1f	0.1801
Level-1	0.1792
Level-2f	0.1159
Level-2	0.1107

Table 4.5: The accuracy decreasing.

As a reference, we test the models' performances on a test dataset based on a dataset with athletes as Fig. 4.31 shows, and the result is shown in Fig. 4.29. On this dataset, the model's performance trained by level-1 data is almost the same as the model trained by level-0 data, which verifies again for the effectiveness of level-1 abstraction.

We calculate the accuracy difference between the models' performances on the worker dataset and the player dataset. As Tab. 4.5 and Fig. 4.30 shows. the performances of models trained by datasets with features (level-1f and level-2f) decrease more dramatically, which means this part of information hinders the network's recognition of human pose on the player dataset. Moreover, the accuracy decreases are positively correlated to the information quantity of the dataset. The more information the model learns, the more dramatically accuracy decreases. It indicates that there are some kinds of unknown but essential differences between the two datasets (worker dataset and player dataset), which makes not all the knowledge learned from the datasets based on the former can be used in the latter classification. In other aspects, it means the model trained by the abstract data is easier to transfer, which is another advantage of the abstract data.

Finally, we compare the scale and time cost of the two dataset with similar effectiveness, level-1 and level-0 as Tab. 4.6 shows. In these two aspects, the level-1 abstract data has clear advantages.

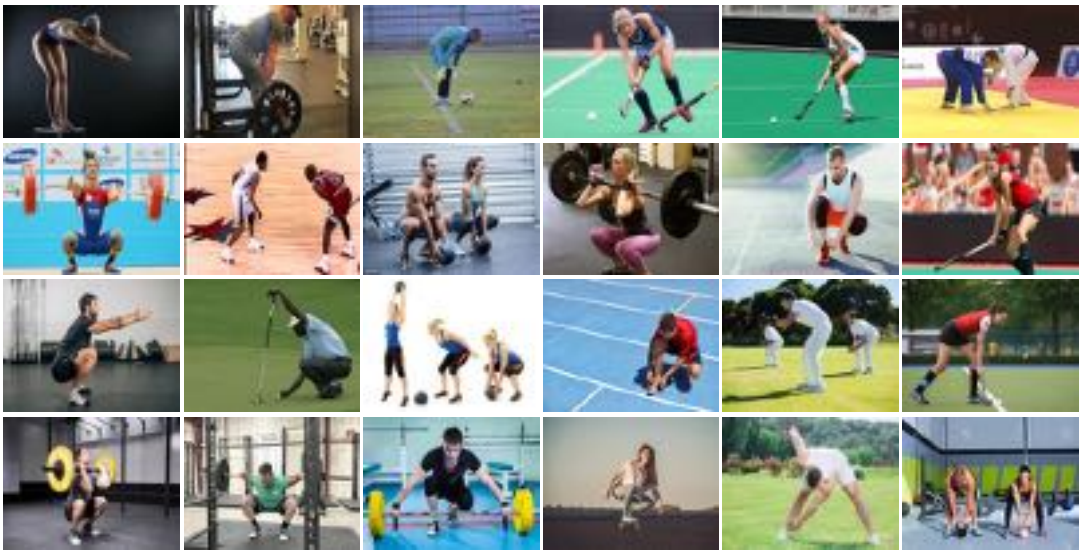


Figure 4.31: Test dataset of athletes. Three kinds of pose are contained inside, standing, bending, and squatting. The main differences of this dataset to the dataset of workers are clothes and environment. These two differences are used to test if the "shortcuts" are eliminated from the DNN or not.

	Level-1	Level-0
Time cost per step (s)	0.772	2.11
Scale (Mb)	10.1	335

Table 4.6: Dataset scale comparing.

4.5.5 Discussion

This dissertation verifies that the data with meaningful abstraction can be used in training. However, there are still some limitations.

Question about the abstract level : In this dissertation, we use five levels of abstraction (level-0, level-1, level-1f, level-2, and level-2f). Someone may question it because there is no mathematical model for the abstraction process. We do not deny that this dissertation's shortcoming, although we list the abstract theoretical basis. However, as a preliminary exploration in this field, our dissertation proves the feasibility of using abstract data to train neural networks, laying the foundation for further exploration in the future.

question about human pose label : In this dissertation, we do not use the traditional human pose representation (skeleton information) but use the region with mark directly. Someone would doubt that the result in this dissertation is not representative of human pose detection and classification. Above all, this strategy satisfies the application's requirement and reduces the computing complexity, which is significant for the potential application platform device. Indeed, this nontraditional strategy might cause some differences between our models' performance and traditional pose detection models. However, this dissertation's main contribution is not in the human pose detection but the verification of the utilization of training based on abstract data. All models are based on the same strategy. Therefore, the conclusion in this dissertation is meaningful and trustworthy.

In conclusion, this dissertation verifies that the data with meaningful abstracts can train the network. It has two main advantages. First, it eliminates the "shortcut" hidden in the dataset, which guarantees the training result is trustworthy. Second, it has a clear advantage on the space occupation comparing with the original data. Moreover, our experiments verify the correctness of the visualization method mentioned in [ZCS+18]. We believe that the well-designed abstract dataset will replace the big data used in neural networks' training in the future.

4.6 Conclusion

In this section, we introduce our works based on the idea of content control. Firstly, We explain the basic framework of content control. Then, we introduce two works based on content control. In the first case, we introduce the work related to the brain MRI analysis based on the conformal welding signature. We have achieved simple recognition of human brain MRI with different IQs based on this feature. In the second case, we use the reconstructed abstract image to train the DNN and eliminate the network cheating

phenomenon mentioned in chapter 2. These cases have strongly proved the feasibility of this idea. However, our research is not perfect. The limitation of this research is introduced in the next chapter.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this dissertation, controlling the content learned by DNN is demonstrated. The corresponding pipeline consists following steps. Firstly, a fine-tuned DNN is used to extract the feature from the raw data. Secondly, knowledge-visualization based feature deconstruction is used to visualize and rank the feature based on the contribution. In this steps, we provide two method to quantify the information learned by the DNN, Euclidean distance-based global information measurement and task-related information measurement. Finally, selected features are used to reconstruct the data which is used to train a initialized DNN. This pipeline is used in two applications and its feasibility has been verified by experiments. This research is meaningful in eliminating the risk of over-training and network cheating. Moreover, with simplified reconstructed data, the training is accelerated, and the computing resource is saved. Compared with the artificial feature extraction, content controlling works based on the DNN. The DNN extracts the feature, and the reconstructed feature is sent back to the DNN to improve it.

5.2 Future Works

Content control is a big topic. Because our understanding of the learning process of neural networks is still limited, most of the work is still in the exploratory stage. Specifically, content controlling demonstrated in this study has at least the following two major limitations.

Limitation with DNN interpretation methods : The first limitation of this research is the inefficiency of the DNN interpretation method. Although we have had much excellent work in XAI, there are still many questions in deep layer semantics understanding. From another aspect, we cannot decide the function of each layer when we design the DNN in most cases. And we have to use a lot of time to test the network structure. Moreover, the feature extracted by the DNN is not "the clearer, the better". The generalization process of DNN is also still unknown, as mentioned in the chapters above. Especially, for some DNNs with unique structures, like ResNet, RNN-based network, etc., the knowledge interpretation method is still blank. Therefore, for now, the content controlling can only be implemented on the convolutional-based neural network. Besides, as mentioned before, for the deep layer of DNN, semantic understanding is still a challenge. For example, for the wings of a plane (as a feature of planes), human beings have a concept about it. However, it is hard for us to describe it formally, and the concept's boundary is ambiguous. Therefore, it is hard to build the map between the concept from our human beings to the concept extracted by the DNN. In most cases, we have to complete this part of the work artificially. However, with the accumulation of the success cases and XAI development, we can reveal the essence of feature-based recognition and classification. The application of content controlling can be expanded in the future.

The Lack of unsupervised feature reconstruction : The second shortcoming is that we currently lack unsupervised algorithms to reconstruct the features. How to rebuild the feature to an understandable form is still a question in this research. In most cases, this part of the work is completed artificially with supervision. In the future, researchers need to pay more attention to this part of the work and provide a formal method without supervision.

Future work : Based on the limitations mentioned above, future work may include:

1. Further improvements in the algorithm of information quantification and knowledge visualization to further deepen our understanding of DNN.
2. Exploration unsupervised understandable feature reconstruction method.

With the development of the research related to content control, we can master the DNN thoroughly.

BIBLIOGRAPHY

[A+18] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10:978–3, 2018.

[Abd17] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[AF00] J. Ashburner and K. J. Friston. Voxel-based morphometry the methods. *NeuroImage*, 11(6):805–821, 2000.

[AFDM16] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[AHD+19] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[ALZ+20] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7350–7357, 2020.

[APS19] Alessandro Achille, Giovanni Paolini, and Stefano Soatto. Where is the information in a deep neural network? *arXiv preprint arXiv:1905.12213*, 2019.

[AS18] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[Bis19] Ekaba Bisong. Google colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 59–64. Springer, 2019.

[BMD09] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. SIAM, 2009.

[BS07] Alexander I Bobenko and Boris A Springborn. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[CCCY18] Wongun Choi, Manmohan Chandraker, Guobin Chen, and Xiang Yu. Learning efficient object detection models with knowledge distillation, September 20 2018. US Patent App. 15/908,870.

[CF01] Richard J Campbell and Patrick J Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[CLR04] Ulrich Clarenz, Nathan Litke, and Martin Rumpf. Axioms and variational problems in surface parameterization. *Computer Aided Geometric Design*, 21(8):727–749, 2004.

[CMW+20] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3430–3437, 2020.

[CMZ+20] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. *arXiv preprint arXiv:2012.03236*, 2020.

[CRCZ20] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12925–12935, 2020.

[DBKMR05] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[Die95] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[DKM20] Michal Derezinski, Rajiv Khanna, and Michael W Mahoney. Improved guarantees and a multiple-descent curve for column subset selection and

the nystrom method. *Advances in Neural Information Processing Systems*, 33, 2020.

[DMC05] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.

[DZ19] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, New York, NY, USA, 2019. ACM.

[EVGW+] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[Far04] Hershel M Farkas. On an arithmetical function. *The Ramanujan Journal*, 8(3):309–315, 2004.

[FLG15] Xiao-Ming Fu, Yang Liu, and Baining Guo. Computing locally injective mappings by advanced mips. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[Flo03] Michael S Floater. Mean value coordinates. *Computer aided geometric design*, 20(1):19–27, 2003.

[FLT+18] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[FMK+03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics (TOG)*, 22(1):83–105, 2003.

[FMLD18] Sebastian Flennerhag, Pablo G Moreno, Neil D Lawrence, and Andreas Damianou. Transferring knowledge across learning processes. *arXiv preprint arXiv:1812.01054*, 2018.

[FNJN19] Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[GIP+18] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *arXiv preprint arXiv:1802.10026*, 2018.

[GJM+20] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

[GJP95] Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269, 1995.

[GL99] F.P. Gardiner and N. Lakic. *Quasiconformal Teichmüler theory*. American Mathematical Society, 1999.

[GM13] Alex Gittens and Michael Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *International Conference on Machine Learning*, pages 567–575. PMLR, 2013.

[GRM+18] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[GWW20a] M. Gao, Y. Wang, and L. Wan. Residual error based knowledge distillation - sciencedirect. *Neurocomputing*, 2020.

[GWW+20b] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

[GWZ+19] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*, pages 2454–2463. PMLR, 2019.

[GYMT20] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*, 2020.

[Hai09] R. J. Haier. Neuro-intelligence, neuro-metrics and the next phase of brain imaging studies. *Intelligence*, 37:121–123, 2009.

[Hay10] Simon Haykin. *Neural networks and learning machines, 3/E*. Pearson Education India, 2010.

[HG00] Kai Hormann and Günther Greiner. Mips: An efficient global parametrization method. Technical report, ERLANGEN-NUERNBERG UNIV (GERMANY) COMPUTER GRAPHICS GROUP, 2000.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[HKDH04] Daniel Huber, Anuj Kapuria, Raghavendra Donamukkala, and Martial Hebert. Parts-based 3d object classification. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[HMP+16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[HST+92] Richard J Haier, Benjamin Siegel, Chuck Tang, Lennart Abel, and Monte S Buchsbaum. Intelligence and changes in regional cerebral glucose metabolic rate following learning. *Intelligence*, 16(3-4):415–426, 1992.

[Hun10] Earl Hunt. *Human Intelligence*. Cambridge University Press, 2010.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZXH19] Wenqing Hu, Zhanxing Zhu, Haoyi Xiong, and Jun Huan. Quasi-potential as an implicit regularizer for the loss function in the stochastic gradient descent. *arXiv preprint arXiv:1901.06054*, 2019.

[ILY+06] Kiho Im, Jong-Min Lee, Uicheul Yoon, Yong-Wook Shin, Soon Beom Hong, In Young Kim, Jun Soo Kwon, and Sun I Kim. Fractal dimension in human cortical surface: multiple regression analysis with cortical thickness, sulcal depth, and folding area. *Human brain mapping*, 27(12):994–1003, 2006.

[JH07] R. E. Jung and R. J. Haier. The parieto-frontal integration theory (P-FIT) of intelligence: Converging neuroimaging evidence. *Behavioral and Brain Sciences*, 30(2):135–154; discussion 154–187, Apr 2007.

[JPW+19] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[KADH+09] S Karama, Y Ad-Dab'bagh, RJ Haier, IJ Deary, OC Lyttelton, C Lepage, AC Evans, Brain Development Cooperative Group, et al. Erratum to "positive association between cognitive ability and cortical thickness in a representative us sample of healthy 6 to 18 year-olds". *Intelligence*, 37(4):432–442, 2009.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBTB14] Dirk P Kroese, Tim Brereton, Thomas Taimre, and Zdravko I Botev. Why the monte carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):386–392, 2014.

[KH09a] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[KH+09b] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBL+18] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

[LCS16] Huu Le, Tat-Jun Chin, and David Suter. Conformal surface alignment with optimal mobius search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2016.

[Lév01] Bruno Lévy. Constrained texture mapping for polygonal meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2001.

[LGYL13] K Lam, Xianfeng Gu, S Yau, and L Lui. Teichmuller mapping (tmap) and its applications to landmark matching registrations. *SIAM Journal on Imaging Sciences*, 2, 2013.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[LNB+07a] Eileen Luders, Katherine L Narr, Robert M Bilder, Philip R Szeszko, Mala N Gurbani, Liberty Hamilton, Arthur W Toga, and Christian Gaser. Mapping the relationship between cortical convolution and intelligence: effects of gender. *Cerebral cortex*, 18(9):2019–2026, 2007.

[LNB+07b] Eileen Luders, Katherine L Narr, Robert M Bilder, Paul M Thompson, Philip R Szeszko, Liberty Hamilton, and Arthur W Toga. Positive correlations between corpus callosum thickness and intelligence. *Neuroimage*, 37(4):1457–1464, 2007.

[LPBSV15] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*, 2015.

[LW07] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[LWZ+12] Lok Ming Lui, Tsz Wai Wong, Wei Zeng, Xianfeng Gu, Paul M Thompson, Tony F Chan, and Shing-Tung Yau. Optimization of surface registrations using beltrami holomorphic flow. *Journal of scientific computing*, 50(3):557–585, 2012.

[LXT+17] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

[LZYG14] L. M. Lui, W. Zeng, S.-T. Yau, and X. Gu. Shape analysis of planar multiply-connected objects using conformal welding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1384–1401, 2014.

[Mar96] John I Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

[MB89] Nelson Morgan and Hervé Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. *Advances in neural information processing systems*, 2:630–637, 1989.

[Mea92] Al Mead. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1):27–39, 1992.

[MFLG19] S. I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. 2019.

[MZ97] Ian Marshall and Danah Zohar. Who's afraid of schrödinger's cat. *London: Bloomsbury*, 1997.

[MZZZ19] Haotian Ma, Yinqing Zhang, Fan Zhou, and Quanshi Zhang. Quantifying layerwise information discarding of neural networks. *arXiv preprint arXiv:1906.04109*, 2019.

[NK19] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

[NKB+19] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

[NWT+06] Katherine L Narr, Roger P Woods, Paul M Thompson, Philip Szeszko, Delbert Robinson, Teodora Dimtcheva, Mala Gurbani, Arthur W Toga, and Robert M Bilder. Relationships between iq and regional cortical gray matter thickness in healthy adults. *Cerebral cortex*, 17(9):2163–2171, 2006.

[OMK20] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[PK19] SeongUk Park and Nojun Kwak. Feed: Feature-level ensemble for knowledge distillation, 2019.

[PKP+19] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[Pre98] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[RBK+14] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[RDS+15] 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*, 115(3):211–252, 2015.

[RF18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[RMCV17] Giorgio Roffo, Simone Melzi, Umberto Castellani, and Alessandro Vinciarelli. Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[Rou87] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[RRC98] J. Raven, J. C. Raven, and J. H. Court. *Raven Manual: Section 4, Advanced Progressive Matrices*. Oxford, UK, Oxford Psychologists Press Ltd., 1998.

[RRF10] Martin Reuter, Herminia Diana Rosas, and Bruce Fischl. Highly accurate inverse consistent registration: A robust approach. *NeuroImage*, 53(4):1181–1196, 2010.

[RSRF12] Martin Reuter, Nicholas J. Schmansky, Herminia Diana Rosas, and Bruce Fischl. Within-subject template estimation for unbiased longitudinal image analysis. *NeuroImage*, 61(4):1402–1418, 2012.

[SA01] Yiyong Sun and Mongi A Abidi. Surface matching by 3d point's fingerprint. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 263–269. IEEE, 2001.

[SBD+19] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

[SGL+06] Philip Shaw, Deanna Greenstein, Jason Lerch, Liv Clasen, Rhoshel Lenroot, NEEA Gogtay, Alan Evans, J Rapoport, and J Giedd. Intellectual ability and cortical development in children and adolescents. *Nature*, 440(7084):676, 2006.

[SH07] Jonathan Starck and Adrian Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[Sha48] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[SKYL17] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew

Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[SLL+20] Ruoyu Sun, Dawei Li, Shiyu Liang, Tian Ding, and Rayadurgam Srikant. The global landscape of neural networks: An overview. *IEEE Signal Processing Magazine*, 37(5):95–108, 2020.

[SM04] E. Sharon and D. Mumford. 2D-shape analysis using conformal mapping. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 350–357, 2004.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[SRDB01] Uwe Siebert, D Rothenbacher, U Daniel, and Hermann Brenner. Demonstration of the healthy worker survivor effect in a cohort of workers in the construction industry. *Occupational and environmental medicine*, 58(12):774–779, 2001.

[SSGH01] Pedro V Sander, John Snyder, Steven J Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416, 2001.

[Ste09] James Stewart. *Calculus: Concepts and contexts*. Cengage Learning, 2009.

[SWGL15] Jian Sun, Tianqi Wu, Xianfeng Gu, and Feng Luo. Discrete conformal deformation: algorithm and experiments. *SIAM Journal on Imaging Sciences*, 8(3):1421–1456, 2015.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZS+13] Rui Shi, Wei Zeng, Zhengyu Su, Hanna Damasio, Zhonglin Lu, Yalin Wang, Shing-Tung Yau, and Xianfeng Gu. Hyperbolic harmonic mapping for constrained brain surface registration. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 2531–2538, 2013.

[SZT17] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[SZW+15] Zhengyu Su, Wei Zeng, Yalin Wang, Zhong-Lin Lu, and Xianfeng Gu. Shape classification using wasserstein distance for brain morphometry analysis. In *International Conference on Information Processing in Medical Imaging*, pages 411–423. Springer, 2015.

[Tea19] The TensorFlow Team. Flowers, jan 2019.

[Ten20a] Tensorflow. Convolutional neural network demo provided by tensorflow. https://www.tensorflow.org/tutorials/images/cnn, 2020.

[Ten20b] TensorFlow. Tensorflow mnist classification classical neural network. https://www.tensorflow.org/quantum/tutorials/mnist, 2020.

[TMS17] Tycho Tax, Pedro AM Mediano, and Murray Shanahan. The partial information decomposition of generative neural network models. *Entropy*, 19(9):474, 2017.

[TPB00] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[Tut60] William Thomas Tutte. Convex representations of graphs. *Proceedings of the London Mathematical Society*, 3(1):304–320, 1960.

[TZ15] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[VN18] John Von Neumann. *Mathematical foundations of quantum mechanics: New edition*. Princeton university press, 2018.

[WC20] Meng-Chieh Wu and Ching-Te Chiu. Multi-teacher knowledge distillation for compressed video action recognition based on deep learning. *Journal of Systems Architecture*, 103:101695, 2020.

[Wei94] Andreas Weigend. On overfitting and the effective number of hidden units. In *Proceedings of the 1993 connectionist models summer school*, volume 1, pages 335–342, 1994.

[WFL$^+$] Xiaobo Wang, Tianyu Fu, Shengcai Liao, Shuo Wang, Zhen Lei, and Tao Mei. Exclusivity-consistency regularized knowledge distillation for face recognition.

[Wic03] Florian Wickelmaier. An introduction to mds. *Sound Quality Research Unit, Aalborg University, Denmark*, 46(5):1–26, 2003.

[WMZ12] Ofir Weber, Ashish Myles, and Denis Zorin. Computing extremal quasiconformal maps. In *Computer Graphics Forum*, volume 31, pages 1679–1689. Wiley Online Library, 2012.

[WS01] Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 14th annual conference*

on neural information processing systems, number CONF, pages 682–688, 2001.

[WVGKP99] Joris Vanden Wyngaerd, Luc Van Gool, R Kock, and Marc Proesmans. Invariant-based registration of surface patches. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 301–306. IEEE, 1999.

[WY21] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[WYKN20] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[WZC+18] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

[WZLT18] Hui Wang, Hanbin Zhao, Xi Li, and Xu Tan. Progressive blockwise knowledge distillation for neural network acceleration. In *IJCAI*, pages 2769–2775, 2018.

[XRLG20] Kunran Xu, Lai Rui, Yishi Li, and Lin Gu. Feature normalized knowledge distillation for image classification. In *The European Conference on Computer Vision (ECCV)*, volume 1, 2020.

[Xu18] Zhiqin John Xu. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018.

[Y+] Liqun Yang et al. mask rcnn data augmentatuion. https://doi.org/10.6084/m9.figshare.11857956.v2.

[YJBK17] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[YRH+19] Liqun Yang, Muhammad Razib, Kenia Chang He, Tianren Yang, Zhong-Lin Lu, Xianfeng Gu, and Wei Zeng. Conformal welding for brain-intelligence analysis. In *International Symposium on Visual Computing*, pages 368–380. Springer, 2019.

[YRZ18] Yi-Jun Yang, Muhammmad Razib, and Wei Zeng. Intrinsic parameterization and registration of graph constrained surfaces. *Graphical Models*, 97:30–39, 2018.

[YSP+20] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. *arXiv preprint arXiv:2012.06048*, 2020.

[YTL+19] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisit knowledge distillation: a teacher-free framework. *arXiv preprint arXiv:1909.11723*, 2019.

[YWJP20] Shujian Yu, Kristoffer Wickstrøm, Robert Jenssen, and Jose C Principe. Understanding convolutional neural networks with information theory: An initial exploration. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[YWS+06] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006.

[YXSY19] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

[YXXT17] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294, 2017.

[YYY+13] J-J Yang, U Yoon, HJ Yun, K Im, YY Choi, KH Lee, H Park, MG Hough, and J-M Lee. Prediction for human intelligence using morphometric characteristics of cortical surface: partial least square analysis. *Neuroscience*, 246:351–361, 2013.

[ZCS+18] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[ZG11] Wei Zeng and Xianfeng David Gu. Registration for 3d surfaces with large deformations using quasi-conformal curvature flow. In *CVPR 2011*, pages 2457–2464. IEEE, 2011.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[ZLM+20] Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. *arXiv preprint arXiv:2009.11729*, 2020.

[ZMLG14] Wei Zeng, Lok Ming Lui, and Xianfeng Gu. Surface registration by optimization in constrained diffeomorphism space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4169–4176, 2014.

[ZSG+19] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.

[ZSGY19] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[ZSW+13] Wei Zeng, Rui Shi, Yalin Wang, Shing-Tung Yau, and Xianfeng Gu. Teichmüller shape descriptor and its application to Alzheimer's disease study. *International Journal of Computer Vision*, 105(2):155–170, 2013.

[ZWB+20] Xinyu Zhang, Xinlong Wang, Jia-Wang Bian, Chunhua Shen, and Mingyu You. Diverse knowledge distillation for end-to-end person search. *arXiv preprint arXiv:2012.11187*, 2020.

[ZWC+20] Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extracting an explanatory graph to interpret a cnn. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[ZWW+19] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song-Chun Zhu. Interpretable cnns for object classification. *arXiv preprint arXiv:1901.02413*, 2019.

[ZXHL18] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[ZY14] Wei Zeng and Yi-Jun Yang. Colon flattening by landmark-driven optimal quasiconformal mapping. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 244–251. Springer, 2014.

[ZZY19] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

VITA

LIQUN YANG

2017-2021 Ph.D. Computer Science
Florida International University, SCIS
Miami, Florida, U.S.

2017-2020 M.S. Computer Science
Florida International University, SCIS
Miami, Florida, U.S.

2013-2017 B.S. Computer Science (Honors Program)
Shandong University, TaiShan College.
Jinan, Shandong, China

PUBLICATION AND PATENT

Wang Y, Yang L, Yang Y, et al. Review the Knowledge Distillation Phenomenonby Quantifying the Task-related Information, 29th International Conference of Case Based Reasoning. (in press)

Zhang H, Yijun Y, Yang L, et al. Diffeomorphic Registration of 3D Surfaces with Point and Curve Landmarks, Computers & Graphics, Elsevier, (in press)

Yang L, Yang Y, et al. The distance between the weights of the neural network is meaningful. arXiv preprint. arXiv:2102.00396

Yang L, Razib M, He K C, et al. Conformal Welding for Brain-Intelligence Analysis, International Symposium on Visual Computing. Springer, Cham, 2019: 368-380.

Yang L, Yang C. A Randomized Large-scale Voronoi Diagram Construction Algorithm Based on Voronoi Area Primitive.2016 17th International Conference on Geometry and Graphics, 57-59.

Patent, Platform of Real-time Photo-realistic Online Rendering, China.

Patent, System of Real-time View-dependent Projection Calibration on Arbitrary Surface, China.