2023

Improving Deep Neural Network Training with Knowledge Distillation

Dongdong Wang
University of Central Florida

IMPROVING DEEP NEURAL NETWORK TRAINING BY KNOWLEDGE DISTILLATION

by

DONGDONG WANG
M.S. Duke University, 2017

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2023

Major Professor: Liqiang Wang

ABSTRACT

Knowledge distillation, as a popular compression technique, has been widely used to reduce deep neural network (DNN) size for a variety of applications. However, in recent years, some research had found its potential for improving deep neural network performance. This dissertation focuses on further exploring its power to facilitate accurate and reliable DNN training. First, I explored data-efficient method for blackbox knowledge distillation where the specifics of the DNN for distillation is inaccessible. I integrated active learning and mixup to obtain significant distillation performance gain with limited data. This work reveals the competence of knowledge distillation to facilitate large foundation model application. Next, I extended this work to solve a more challenging practical problem, *i.e.* COVID-19 infection prediction. Due to extremely limited data at the outbreak, it is very difficult to calibrate any existing epidemic model for practical prediction. I applied blackbox knowledge distillation with sequence mixup to distill a comprehensive physics-based simulation system. With the obtained distilled model, epidemic models are better calibrated to fit limited observation data and provide more accurate and reliable projection. This work validates that knowledge distillation can enhance DNN training for complex time series prediction with limited observation data. Next, I applied knowledge distillation to improve DNN reliability which reflects accurate model prediction confidence. Ensemble modeling and data augmentation had been blended to equip distillation process and obtain a reliable DNN. This work justifies that knowledge distillation can equip training for a more reliable DNN. Furthermore, this dissertation extended my knowledge distillation study to semantic segmentation tasks. The study started with investigation of semantic segmentation models, and then, proposed an approach of adaptive convolution to improve the heterogeneity of local convolution fields. The experiments had been carried out across different scales of segmentation benchmarks and justified that this approach outperforms existing state-of-the-art schemes and successfully boosts the performance of various back-

bone models. After this investigation study, semantic segmentation models had been calibrated with ensemble knowledge distillation which had been applied to solve image classification calibration. Stronger augmentation had been incorporated into distillation process. The experiments justify the effectiveness for semantic segmentation calibration.

Dedicate to those who support my life.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Liqiang Wang, for providing significant support without which this research study would not have been possible. He provided a flexible research environment to explore more topics interesting me. This helped to improve my research productivity. I am also very thankful to his valuable suggestions on life and my career plan. Particularly, I am deeply grateful to Dr. Boqing Gong. His enthusiasm for computer vision, suggestions on research topics, and guidance on my research work, inspired me to achieve the success of my Ph.D. training. The skills and work ethic I learnt from him will advance my career. I would also like to show gratitude to my other committee members, including Dr. Chen Chen and Dr. Yanjie Fu. I am very grateful for their support, invaluable advice, and patience with me. Genuine thanks to each of them for the extraordinary amount of time and knowledge they were willing to provide to my dissertation. Getting through my dissertation required more than academic support, and I have many people to thank for their patience with me over the past four years. I cannot begin to express my gratitude and appreciation for their friendship. Yandong Li, Hao Hu, Jie Yao, Yuxiang Yang, Bingbing Rao, Zixia Liu, and Yifan Ding have been unwavering in their personal and professional support during the time I spent at the University of Central Florida. Last but not least, I acknowledge the supports from my family. Without their consistent support, I would never accomplish this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Knowledge distillation is proposed in [6] and its original intention is to solve model compression problems, thus relieving the burden of training large DNNs, like ensemble learning. Hinton et al. reveal that the probability outputs from large networks can be taken as "dark knowledge" to retain its performance on accuracy with light-weight substitute models. Usually, a larger DNN is taken as a teacher model, whilst the smaller one is viewed as a student model. Typical knowledge distillation uses logits, which are before softmax activation, from the teacher model as the source knowledge. The student model utilizes this knowledge to mimic the response of the teacher model, thus retaining its output behavior. Commonly, the mimicking is carried out by minimization logit distribution discrepancy between student and teacher models.

However, traditional knowledge distillation still has some important issues. The first important one is high demand for training data. When it comes to model query for knowledge acquisition, traditional methods usually require a large number of original data to retrieve information from teacher model for accurate distillation, which is infeasible in real applications. To alleviate this data demand, several approaches are proposed, such as few-shot knowledge distillation [7], data free knowledge distillation [8], zero-knowledge distillation [9], and so forth. Nevertheless, these approaches require the gradient information of teacher network, which enables them also intractable in the real world.

Moreover, its potential for model improvement is still under exploration. With respect of training accuracy, lots of research finds out its potential for better generalization. For example, On-the-fly Native Ensemble (ONE) equipped with ensemble distillation outperforms ensemble model [10]. Self-distillation with retraining the model can help increase model accuracy [11, 12]. These works show that when distillation framework is optimized, like moderately increase in model size, it

is possible to render a student network outperform the teacher model in accuracy. This implies knowledge distillation enables model accuracy improvement.

Recently, more research with further exploration also justifies that this approach is effective on model reliability improvement. Although modern DNN achieves striking success in accuracy improvement, its calibration performance is inferior and usually suffers overconfidence issue [13]. This problem affects the application of DNN since the users may obtain misleading inference outputs and take inaccurate decisions. To address this issue, some current research proposes several approaches, such as ensemble training [5], ensemble distribution distillation [14], batch ensemble distillation [15], Dirichlet distribution [14], augmented distillation[15], and so forth.

To promote the application of deep neural networks in more real applications, this dissertation explores knowledge distillation for efficiently training deep neural networks. The dissertation attempts to address multiple pressing issues in or with knowledge distillation:

1. DNNs have limitation in blackbox knowledge distillation, especially when labeled data are very few which causes lower distillation accuracy (cf. Chapter 3).
2. Prediction problems are very difficult when observation data are very limited, which have to be used for time series modeling (cf. Chapter 4).
3. DNNs can exhibit poorer model reliability which yields misleading predictive confidence for inaccurate prediction (cf. Chapter 5).
4. Segmentation models can also confront the reliability challenges and require further calibration improvement (cf. Chapter 6).

This dissertation tackles these challenges from several perspectives. First, given limited labeled data for knowledge distillation, a challenged problem is formulated with three constraints, including data limitation, training efficiency, and distillation effectiveness. Chapter 3 studies how to train a student deep neural network for visual recognition by distilling knowledge from a blackbox

teacher model in a data-efficient manner. Progress on this problem can significantly reduce the dependence on large-scale datasets for learning high-performing visual recognition models. There are two major challenges. One is that the number of queries into the teacher model should be minimized to save computational and/or financial costs. The other is that the number of images used for the knowledge distillation should be small; otherwise, it violates our expectation of reducing the dependence on large-scale datasets. To tackle these challenges, we propose an approach that blends mixup and active learning. The former effectively augments the few unlabeled images by a big pool of synthetic images sampled from the convex hull of the original images, and the latter actively chooses from the pool hard examples for the student neural network and query their labels from the teacher model. We validate our approach with extensive experiments.

Second, this dissertation attempts to extend knowledge distillation to a real-world problem. The problem is formulated based upon limited observation data and constrained computation resource. An accurate and efficient forecasting system is imperative to the prevention of emerging infectious diseases such as COVID-19 in public health. This system requires accurate transient modeling, lower computation cost, and fewer observation data. To tackle these three challenges, Chapter 4 proposes a novel deep learning approach using black-box knowledge distillation for both accurate and efficient transmission dynamics prediction in a practical manner. First, we leverage mixture models to develop an accurate, comprehensive, yet impractical simulation system. Next, we use simulated observation sequences to query the simulation system to retrieve simulated projection sequences as knowledge. Then, with the obtained query data, sequence mixup is proposed to improve query efficiency, increase knowledge diversity, and boost distillation model accuracy. Finally, we train a student deep neural network with the retrieved and mixed observation-projection sequences for practical use. The case study on COVID-19 justifies that our approach accurately projects infections with much lower computation cost when observation data are limited.

Third, in addition to model accuracy, model reliability is further studied and explored for model

training improvement. According to [13], modern deep neural network causes severely overconfident prediction and misleading model users for decision making. An accurate and reliable deep neural network (DNN) is critical. However, due to model uncertainty, it is very challenging to achieve such DNNs. To reduce model uncertainty, Chapter 5 proposes ensemble knowledge distillation, a simple yet effective approach by leveraging accurate point estimation. There are two challenges for accurate point estimation. One is the unbiasness from true parameter and the other is high efficiency with lower estimator variance. To tackle these two challenges, we blend ensemble model and knowledge distillation to improve model point estimation. The former effectively reduces the bias of the estimation in virtue of large sample of model estimators. The latter significantly reduces model estimator variance by distilling the ensemble into a single model. This effective integration successfully provides a more accurate point-estimate single model with lower model uncertainty. We justify our approach with extensive experiments and show its significant improvement in in-distribution model calibration and out-of-distribution data detection. We also reveal the importance of efficient data augmentation in model uncertainty reduction. Last, but not least, we validate our approach with more up-to-date DNNs, like Vision Transformer.

Next, the reliability study is extended to semantic segmentation tasks. Since deep neural networks achieve tremendous success in image classification tasks, more complex computer vision applications incorporate them into problem solving. Semantic segmentation, as an important but challenging computer vision task, employs a variety of modern deep neural networks and show much success in performance improvement. However, like image classification, it can encounter accuracy and reliability problems. To better solve it, this dissertation conducts a study on understanding semantic segmentation through analysis and algorithmic optimization and then, applies image classification calibration techniques to semantic segmentation calibration. Chapter 6 starts with investigation on convolution neural network for semantic segmentation and explores a method to improve convolutional modeling performance on semantic segmentation. Dilated convolution

kernels are constrained by their shared dilation, keeping them from being aware of diverse spatial contents at different locations. We address such limitations by formulating the dilation as trainable weights with respect to individual positions. We propose Adaptive Dilation Convolutional Neural Networks (ADCNN), a light-weighted extension that allows convolutional kernels to adjust their dilation value based on different contents at the pixel level. Unlike previous content-adaptive models, ADCNN dynamically infers pixel-wise dilation via modeling feed-forward inter-patterns, which provides a new perspective for developing adaptive network structures other than sampling kernel spaces. Our evaluation results indicate ADCNNs can be easily integrated into various backbone networks and consistently outperform their regular counterparts on various visual tasks. Then, given the understanding of semantic segmentation modeling with deep neural networks, this dissertation further studies the problem of semantic segmentation calibration. For image classification, lots of existing solutions are proposed to alleviate model miscalibration of confidence. However, to date, confidence calibration research on semantic segmentation is still limited. We provide a study on the calibration of semantic segmentation models and propose a simple yet effective approach, namely ensemble knowledge distillation. Next, we study popular existing calibration methods and compare them with ensemble knowledge distillation on semantic segmentation calibration. We conduct extensive experiments with a variety of benchmarks on in-domain calibration, and show that ensemble knowledge distillation consistently outperforms other methods.

Note that the mathematical notations are consistent in each individual chapter, but the same symbol may refer to different concepts in different chapters.

CHAPTER 2: LITERATURE REVIEW

2.1 Knowledge Distillation and Its Application

Knowledge distillation [6] is widely used to solve DNN compression problem. Since it effectively reduces model size with retaining accurate performance, it makes accurate but complex models feasible for real-world applications. The distillation process is accomplished in an inversion manner. Take the model θ as a teacher model for distillation. To retain the performance of this teacher model θ, we may retrieve its outputs and train another smaller student model ϕ by imitating its outputs. The outputs could be the logits or probabilities from teacher model θ. During the distillation, true observation data can be also incorporated into model training when they are feasible. It can be formulated as the following optimization.

$$\phi = \arg\min_{\phi} \alpha \mathscr{L}_{CE}(P(y|\mathscr{D},\phi),y) + \beta \mathscr{L}_{KD}(P(y|\mathscr{D},\phi),q_\theta). \tag{2.1}$$

θ is model parameter, $\mathscr{L}_{CE}$ is cross-entropy loss, P is model inference probability, y is the true label encoded by one-hot vector, $\mathscr{D}$ denotes input data, ϕ denotes student model parameters, $\mathscr{L}_{KD}$ indicates distillation loss, and q_θ is the soft targets, the probabilities queried from θ. There are two hyperparameters of α and β to balance distillation and cross-entropy training.

In the field of computer vision, this technique has helped solve a variety of complex model size problems. For example, in pose estimation [16, 17, 18], lane detection [19], real-time streaming [20], object detection [21], video representation [22, 23, 24], and so forth. Furthermore, this approach is able to boost the performance of DNN with improvement on efficiency [25] and accuracy [26]. Accordingly, lots of research is conducted to enhance its performance from the per-

spective of training strategy [27, 28], distillation scheme [29, 30], or network properties [31] , etc.

In the field of natural language processing, this technique improves the feasibility of large models like BERT. For example, DistilBERT [32] successfully reduces the size of original BERT model by 40% with maintaining accuracy; TinyBERT [33] leverages knowledge distillation to design a framework for the reduction of transformer-based language model, which leads to the models with lower time and space complexity, thus facilitating its application; relational knowledge distillation [34] further optimizes distillation process and enables more productive student model, which can even outperform teacher model.

2.2 Data Efficiency

When it comes to model query for knowledge acquisition, traditional methods usually require a large number of original data to retrieve knowledge information for accurate distillation, which is infeasible in real applications. Several approaches are proposed to solve this problem. For example, few-shot knowledge distillation is proposed to retain teacher model performance with pseudo samplers which are generated in adversarial manner [7]. Another approach called data free knowledge distillation leverages extra activation records from teacher model to reconstruct original datasets, thus recovering teacher model [8]. Recently, a zero-knowledge distillation method is developed by synthesizing data with gradient information of teacher network [9]. Nevertheless, these approaches require the gradient information of teacher network, which enables them intractable in the real world. To tackle this challenge, we formulated a blackbox optimization process for knowledge distillation and solved it a data-efficient manner in [35]. We propose Active Mixup [35], which blends active learning and mixup to efficiently invert teacher model without specifics like gradient information.

2.3 Model Accuracy

Recently, more research found that knowledge distillation can help model generalization due to its optimization on soft targets [36], which can be taken as more general case of label smoothing [37]. Therefore, model accuracy improvement from knowledge distillation draw more attention and more techniques are integrated or proposed to boost its performance. For example, data augmentation is very effective, especially in image classification task, since it can help model capture multiple views on data [36].

Among multiple approaches, mixup is a simple yet effective approach to augment training data and improve model performance [1]. This method is proposed to improve the generalization of DNN by enhancing coverage of data distribution, especially when training data are limited. The main idea is to incorporate convex combination into data synthesis, which involves mixing features and mixing labels. It has been widely used to address both computer vision and natural language processing problems, like Between-Class learning in speech recognition [38] and image classification[39], AutoAugment with learning strategy augmentation for classification [40], and wordMixup or senMixup with embedding mixup for sentence classification [41]. More studies explore its potential for data-efficient learning, such as Active Mixup [35] and ranking distillation in [42].

Another interesting accuracy boosting method is self-distillation. Self-distillation is the knowledge distillation where the labeled data and architectures of teacher and student are identical. This scheme can effectively boost model generalization [11, 12] in virtue of its amplifying model regularization [43]. Some regularization techniques can help model generalization to unseen data, such as weight decay [13] and label smoothing [44]. Since knowledge distillation can be taken as an learned label smoothing regularization[37], we can use self-distillation to regularize ensemble member for better generalization.

2.4 Model Reliability

In addition to model accuracy, model reliability is also critical to real-world application [13]. Although DNNs achieve tremendous success in inference accuracy, its reliability is weak, which usually refers to overconfidence in predicted results [13]. This causes potential risk because erroneous prediction may yield high inference probability and severely mislead decision making, which may be fatal to real-world applications like autonomous driving. To address this issue, more research is conducted to explore how to better calibrate DNN.

Among a range of proposed approaches, deep ensemble is found to be simple but effective[5] because it is closer to Bayesian inference process which is justified to be more accurate to capture posterior distribution pattern[45]. Upon this observation, several approaches are developed. For example, [46] proposes a prior network to more accurately describe prior distribution of the training data. Upon this work, [14] proposes ensemble distribution distillation with help of Dirichlet distribution to achieve better calibrated inference outputs. However, it requires lots of ensemble networks to sample model output distribution, which renders training a prior network computationally expensive and impractical for large models. Moreover, Dirichlet distribution is an approximate solution to ensemble model outputs, which is not always guaranteed. This leads to the gap between ensemble model and distilled distribution network. [15] proposes batch ensemble distillation, which takes advantage of batch ensemble training with more diverse data by input perturbation to fill the gap between ensemble model and distilled network, but the ensemble model as an upper bound still limits model uncertainty performance.

CHAPTER 3: ACTIVE MIXUP FOR DATA-EFFICIENT KNOWLEDGE DISTILLATION FROM A BLACKBOX MODEL

3.1 Problem Introduction

Data curation is one of the most important steps for learning high-performing visual recognition models. However, it is often tedious and sometimes daunting to collect large-scale relevant data that have sufficient coverage of the inference-time scenarios. Additionally, labeling the collected data is time-consuming and costly.

Given a new task, how can we learn a high-quality machine learning model in a more data-efficient manner? We believe the answer varies depending on specific application scenarios. In this paper, we focus on the case that there exists a *blackbox* teacher model whose capability covers our task of interest. Indeed, there are many high-performing generic visual recognition models available as Web-based APIs, in our smart devices, or even as an obsolete model built by ourselves some while ago. The challenge is, however, we often have limited knowledge about their specifics, e.g., not knowing the exact network architecture or weights. Moreover, it could be computationally and/or financially expensive to query the models and read out their outputs for a large-scale dataset.

To this end, we study how to distill a *blackbox* teacher model for visual recognition into a student neural network in a data-efficient manner. Our objective is three-fold. First of all, we would like

This chapter contains previously published materials from"Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model", by Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1498-1507. 2020.

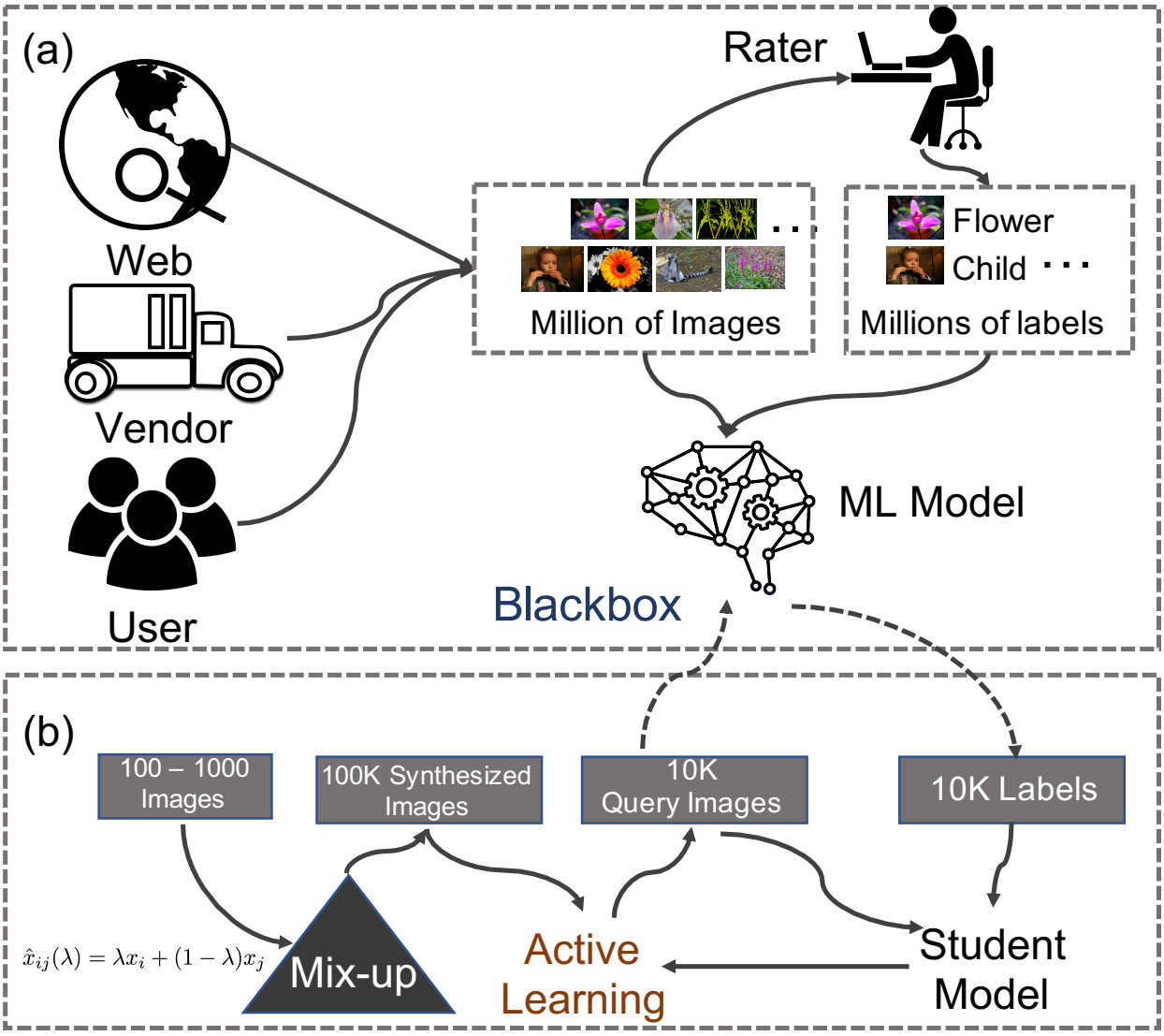


Figure 3.1: Data-efficient blackbox knowledge distillation. Given a blackbox teacher model and a small set of unlabeled images, we propose to employ mixup [1] and active learning [2] to train a high-performing student neural network in a data-efficient manner (b) so that we do not need to re-do the heavy and expensive data curation used to train the teacher model (a).

the distilled student network to perform well as the teacher model as possible at the inference time. Besides, we try to minimize the number of queries to the blackbox teacher model to save costs. Finally, we also shall use as a small number of examples as possible to save data collection efforts. It is hard to collect abundant data for rare classes or privacy-critical applications.

We propose to blend active learning [47, 2] and image mixup [1] to tackle the data-efficient knowledge distillation from a blackbox teacher model. The main idea is to synthesize a big pool of images from the few training examples by mixup and then use active learning to select from the pool the most helpful subset to query the teacher model. After reading out the teacher model's outputs, we simply treat them as the "groundtruth labels" of the query images and train the student neural network with them.

Image mixup [1, 48, 49] was originally proposed for data augmentation to improve the generalization performance of a neural recognition network. It synthesizes a virtual image by a convex combination of two training images. While the resultant image may become cluttered and semantically meaningless, it resides near the manifold of the natural images — unlike white-noise images. Given 1000 images, we can construct $O(10^5)$ pairs, each of which can further generate tens to thousands of virtual images depending on the coefficients in the convex combination. We conjecture that the big pool of mixup images provides good coverage of the manifold of natural images. Hence, we expect that a student network that imitates the blackbox teacher on the mixup images can give rise to similar predictions over the test images as the teacher model does.

Instead of querying the blackbox teacher model by all the mixup images, we resort to active learning to improve the querying efficiency. We first acquire the labels of the small number of original images from the blackbox teacher model and use them for training the student network. We then apply the *student* network to all the mixup images to identify the subset with which the current student network is the most uncertain. Notably, if two mixup images are synthesized from the same pair of original images, we keep only the one with higher uncertainty. We query labels for this subset, merge it into the previously labeled data, and then re-train the student network. We iterate this procedure of subset selection, querying the blackbox teacher model, and training the student neural network multiple times until reaching a stopping criterion.

To the best of our knowledge, we are the first to distill knowledge from a blackbox teacher model while underscoring the need for data-efficiency and query-efficiency. We empirically validate our approach by contrasting it to both vanilla and few/zero-shot knowledge distillation methods. Experiments show that, despite the blackbox teacher in our work, our approach performs on par or better than the competing methods that learn from whitebox teachers.

Note that the mixup images are often semantically meaningless, making them almost impossible for human raters to label. However, the blackbox teacher model returns predictions for them regardless, and the student network still gains from such fake image-label pairs. In this sense, we say that the blackbox teacher model is more productive than human raters in teaching the student network.

3.2 Background

Knowledge Distillation. Knowledge distillation is proposed in [6] to solve model compression problems, thus relieving the burden of ensemble learning. This work suggests that class probabilities, as “dark knowledge", are very useful to retain the performance of original network, and thus, light-weight substitute model could be trained to distill this knowledge. This approach is very useful and has been justified to solve a variety of complex application problems, such as pose estimation [16, 17, 18], lane detection [19], real-time streaming [20], object detection [21], video representation [22, 23, 24], and so forth. Furthermore, this approach is able to boost the performance of deep neural network with improvement on efficiency [25] and accuracy [26]. Accordingly, lots of research is conducted to enhance its performance from the perspective of training strategy [27, 28], distillation scheme [29, 30], or network properties [31] , etc.

However, there is an important issue. Traditional knowledge distillation requires lots of original

training data which are very difficult to be obtained. To alleviate this data demand, few-shot knowledge distillation is proposed to retain teacher model performance with pseudo samplers which are generated in adversarial manner [7]. Another approach called data free knowledge distillation leverages extra activation records from teacher model to reconstruct original datasets, thus recovering teacher model [8]. Recently, a zero-knowledge distillation method is developed by synthesizing data with gradient information of teacher network [9]. Nevertheless, these approaches require the gradient information of teacher network, which enables them intractable in the real world.

Blackbox Optimization. Blackbox optimization is developed based on zero knowledge in the gradient information of queried models and widely used to solve practical problems. Recently, this work is widely used in deep learning, especially model attack. A rich line of blackbox attacking approaches [50, 51, 52, 53, 54] are explored by accessing the input-output pairs of classifiers, most of which are focusing on attacks resulting from accessing the data. [55] instead investigates that the adversaries are capable of recovering sensitive data by model inversion. However, there is no work for blackbox knowledge distillation.

Active Learning. Active learning is a learning process by interaction between oracle and learner agents. This strategy is widely used to solve learning problems which exhibit costly data labelling since it could exploit existing data information to efficiently improve obtained model, thus reducing the number of queries. Lots of effective approaches are proposed to optimize this process, such as uncertainty-based [2, 56, 57] and margin-based methods [58, 59]. Form the review by [60], uncertainty-based methods, despite simple, are able to obtain good performance.

Mixup. Zhang *et al.* first proposed mixup to improve the generalization of deep neural network [1]. Between-Class learning [38] (BC learning) was proposed for deep sound recognition, and then, they extended this approach to image classification [61]. Following them, Pairing Samples [62] was proposed as a data augmentation approach by taking an average of two images for each pixel.

More recently, an approach called AutoAugment [40], explores improving data augmentation policies by automatically searching.

3.3 Approach

We present our approach to the data-efficient knowledge distillation from a blackbox teacher model in detail in this section. Given a blackbox teacher model and a small number of unlabeled images, the approach iterates over the following three steps: 1) constructing a big candidate pool of synthesized images from the small number of unlabeled images, 2) actively choosing a subset from the pool with which the current student network is the most uncertain, 3) querying the blackbox teacher model to acquire labels for this subset and to re-train the student network.

3.3.1 Constructing a Candidate Pool

In real-world applications, data collection could consume a huge amount of time due to various reasons, such as privacy concerns, rare classes, data quality, etc. Instead of relying on a big dataset of real images, we begin with a small number of unlabeled images and use the recently proposed mixup [1] to augment this initial image pool.

Given two natural images x_i and x_j, mixup generates multiple synthetic images by a convex combination of the two with different coefficients,

$$\hat{x}_{ij}(\lambda) = \lambda x_i + (1-\lambda)x_j, \tag{3.1}$$

where the coefficient $\lambda \in [0,1]$. Note that this notation also includes the original unlabeled data x_i and x_j when $\lambda = 1$ and $\lambda = 0$, respectively.

This technique comes handy and effective for our work. It can exponentially expand the size of the initial image pool. Suppose we have collected 1000 natural images, and we generate 10 mixup images for each image pair by varying the coefficient λ. We then arrive at a pool of about 10^6 images in total. Besides, this pool of synthetic images also provides good coverage of the manifold of natural images. Indeed, this pool can be viewed as a dense sampling of the convex hull of the natural images we have collected. The test images likely fall into or close to this convex hull if the collected images are diverse and representative. Hence, we expect the student neural network to generalize well to the inference-time data by enforcing it to imitate the blackbox teacher model on the mixup images.

3.3.2 Actively Choosing a Subset to Query the Teacher Model

Let $\{\hat{x}_{ij}(\lambda), \lambda \in [0,1], i \neq j\}$ denote the augmented pool of images. It is straightforward to query the teacher model to obtain the (soft) labels for these synthetic images and then train the student network with them. However, this brute-force strategy incurs high computational and financial costs. Instead, we employ active learning to reduce the cost.

We define the student neural network's confidence over an input x as

$$C_1(x) := \max_y P_S(y|x), \tag{3.2}$$

where $P_S(y|x)$ is the probability of the input image x belonging to the class y predicted by the current student network. Intuitively, the less confidence the student network has over the input x, the more the student network can gain from the teacher model's label for the input.

Therefore, we could rank all the synthetic images in the candidate pool according to the student network's confidences on them, and then choose the top ones as the query subset. However, this

simple strategy results in near-duplicated images, for example $\hat{x}_{ij}(\lambda = 0.5)$ and $\hat{x}_{ij}(\lambda = 0.55)$. We avoid this situation by choosing at most one image from any pair of images.

In particular, instead of ranking the synthetic images, we rank image pairs in the candidate pool. We define the confidence of the student network over an image pair x_i and x_j as the following,

$$C_2(x_i, x_j) := \min_{\lambda} C_1(\hat{x}_{ij}(\lambda)), \quad \lambda \in [0, 1], \tag{3.3}$$

which depends on a coefficient λ^* for the image pair. Hence, we obtain a confidence score and its corresponding coefficient for any pair of the original images. The synthetic image $\hat{x}_{ij}(\lambda^*)$ is selected into the query set if the confidence score $C_2(x_i, x_j)$ is among the lowest k ones. We study the size of the query set in the experiments.

3.3.3 Training the Student Network

With the actively selected query set of images, we query the blackbox teacher model and read out its soft predictions as the labels for the images. We then merge them with the previous training set, if there is, to train the student network using a cross-entropy loss. The soft probabilistic labels returned by the teacher model give rise to slightly better results than the hard labels, so we shall use the soft labels in the experiments below.

3.3.4 Overall Algorithm

Algorithm 1 presents the overall procedure of our approach to the data-efficient blackbox knowledge distillation. Beginning with a teacher model $\mathscr{M}^T$ and a few unlabeled images $X = \{x_1, x_2, ..., x_n\}$, we firstly train an initial student network $\mathscr{M}_0^S$ with (X, Y_0), where Y_0 contains the labels for the im-

ages in X and is obtained by querying the teacher model. We then construct a big pool of synthetic images $\mathscr{P}$ with mixup [1] (eq. (3.1)) to facilitate the active learning stage. We iterate the following steps until the accuracy of the student network converges. 1) Actively select a subset $\Delta\mathscr{P}_t^s$ of the synthetic images $\mathscr{P}$ with the lowest confidence scores, $C_2(x_i,x_j)$, as predicted by the current student network so that the resulting subset $\Delta\mathscr{P}_t^s$ contains hard samples for the current student network $\mathscr{M}_{t-1}^S$. 2) Acquire labels $\Delta\mathscr{Y}_t$ of the selected subset of synthetic images $\Delta\mathscr{P}_t^s$ by querying the teacher model. 3) Train a new student network $\mathscr{M}_t^S$ with all the labeled images thus far, $(\mathscr{P}_t^s, \mathscr{Y}_t)$. Note that, in Line 6 of Algorithm 1, we only keep one synthetic image for any pair (x_i,x_j) of the original images to reduce redundancy.

Algorithm 1 Data-efficient blackbox knowledge distillation

INPUT: Pre-trained teacher model $\mathscr{M}^T$
INPUT: A small set of unlabeled images $X = \{x_i\}_{i=1}^n$
INPUT: Hyper-parameters (learning rate, subset size, etc.)
OUTPUT: Student network $\mathscr{M}^S$

1: Query $\mathscr{M}^T$ and acquire labels Y_0 for all images in X
2: Train an initial student network $\mathscr{M}_0^S$ with (X,Y_0)
3: Construct a synthetic image pool $\mathscr{P} = \{\hat{x}_{ij}(\lambda)\}$ by using the unlabeled images X with eq. (3.1)
4: Initialize $\mathscr{P}_1^s = X, \mathscr{Y}_1 = \mathscr{Y}_0$.
5: **for** $t = 1,2...,T$ **do**
6: Select a subset $\Delta\mathscr{P}_t^s$ from $\mathscr{P}$ with lowest confidence scores $\{C_2(x_i,x_j)\}$ returned by student $\mathscr{M}_{t-1}^S$
7: Query $\mathscr{M}^T$, acquire labels $\Delta\mathscr{Y}_t$ for all images $\Delta\mathscr{P}_t^s$
8: $\mathscr{P}_t^s \leftarrow \mathscr{P}_t^s \cup \Delta\mathscr{P}_t^s$, $\mathscr{Y}_t \leftarrow \mathscr{Y}_t \cup \Delta\mathscr{Y}_t$
9: Train a new student network $\mathscr{M}_t^S$ with $(\mathscr{P}_t^s, \mathscr{Y}_t)$
10: Update $\mathscr{P} \leftarrow \mathscr{P}$ - $\Delta\mathscr{P}_t^s$
11: **end for**

3.4 Experiments

We design various experiments to test our approach, including both comparison experiments with state-of-the-art knowledge distillation methods and ablation studies. Additionally, we also chal-

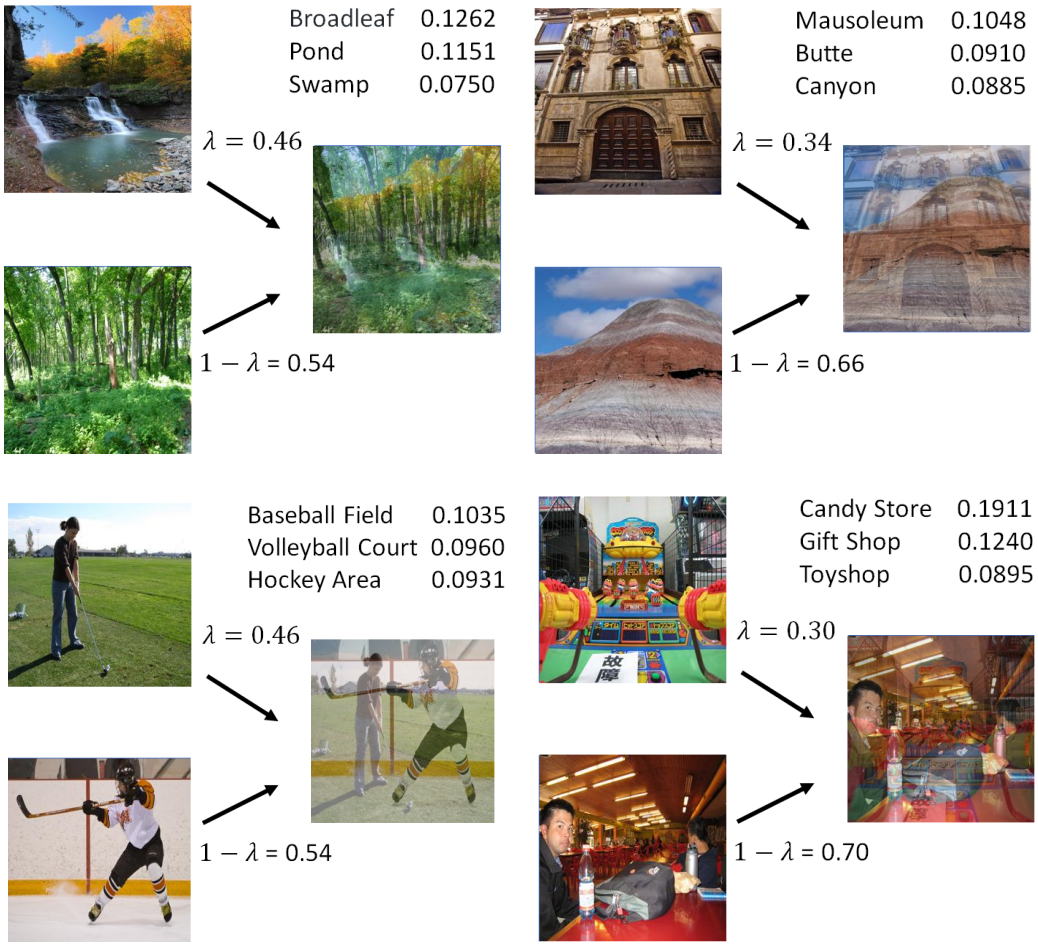


Figure 3.2: Mixup images whose confidence scores (cf. eq. (3.3)) are the lowest among all candidates in the third iteration. For each mixup image, we show the top three labels and probabilities returned by the blackbox teacher model.

lenge our approach when the available data is out of the distribution of the main task of interest. In practice, across all experiments, we select $\lambda \in \{0.3, 0.7\}$ (with an interval of 0.04) to generate synthetic images to produce more diverse mixup images.

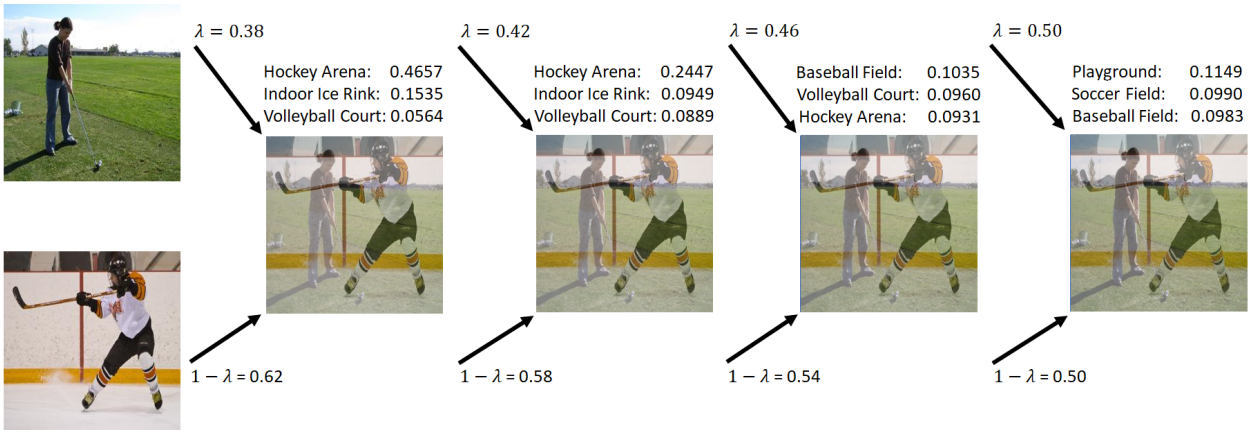


Figure 3.3: Different mixup images from the same pair of the original images by varying the mixup coefficient λ. We show the top three labels and probabilities predicted by the teacher model for each of them. It is interesting to see how the top-1 label changes from Hockey Arena, to Baseball Field, and to Golf Course.

3.4.1 Comparison Experiments

Since our main objective is to explore how to train a high-performing student neural network from a blackbox teacher model in a data-efficient manner, it is worth comparing our approach with existing knowledge distillation methods although they were developed for other setups. The comparison can help review how data-efficient our approach is given the blackbox teacher model.

3.4.1.1 Experiment Setting

Datasets. We run experiments on MNIST [63], Fashion-MNIST [64], CIFAR-10 [65], and Places365-Standard [66], which are popular benchmark datasets for image classification. The MNIST dataset contains 60K training images and 10K testing images about ten handwritten digits. The image resolution is 28×28. Fashion-MNIST is composed of 60K training and 10K testing fashion product images of the size 28×28. CIFAR-10 consists of 60K (50K training images and 10K test images) 32×32 RGB images in 10 classes, with 6K images per class. In addition to evaluating

the proposed approach on the above described low-resolution images, we also test our approach on Places365-Standard, which is a challenging dataset for natural scene recognition. It has 1.8M training images and 18,250 validation images in 365 classes. We use the resolution of 256×256 for Places365-Standard in the following experiments.

Evaluation Metric. We mainly use the classification accuracy as the evaluation metric. Additionally, we also propose a straightforward metric to measure how much "knowledge" the student network distills from the teacher model. This metric is computed as the ratio between the student network's classification accuracy and the teacher's accuracy, and we call it the distillation *success rate*.

Table 3.1: Comparison results on Places365-Standard, CIFAR-10, MNIST, and Fashion-MNIST. The "Teacher" column reports the teacher model's accuracy on the test sets, "KD Accuracy" is the student network's test accuracy, "Success" stands for the distillation success rates, "Black/White" indicates whether or not the teacher model is blackbox, "Queries" lists the numbers of queries into the teacher models, and "Unlabeled Data" shows the numbers of original training images used in the experiments. (* results reported in the original paper)

Task (Model)	Teacher	KD Accuracy	Success	Black/White	Queries	Unlabeled Data
Places365-Standard (ZSKD) [9]	–	–	–	–	–	0
Places365-Standard (FSKD [7])	53.69	38.18	71.11	White	480K	80K
Places365-Standard (KD)	53.69	49.01	90.35	Black	1,800K	1,800K
Places365-Standard (Ours)	53.69	**45.71**	**85.14**	Black	480K	80K
CIFAR-10 (ZSKD) [9]	83.03*	69.56*	83.78	White	>2,000K	0
CIFAR-10 (FSKD [7])	83.07	40.58	48.85	White	40K	2K
CIFAR-10 (KD)	83.07	80.01	96.31	Black	50K	50K
CIFAR-10 (Ours)	83.07	**74.60**	**89.87**	Black	40K	2K
MNIST (ZSKD) [9]	99.34*	**98.77***	99.42	White	>1,200K	0
MNIST (FSKD [7])	99.29	80.43	81.01	White	24K	2K
MNIST (KD)	99.29	99.05	99.76	Black	60K	60K
MNIST (Ours)	99.29	98.74	**99.45**	Black	24K	2K
Fashion-MNIST(ZSKD) [9]	90.84*	79.62*	87.65	White	>2,400K	0
Fashion-MNIST (FSKD [7])	90.80	68.64	75.60	White	48K	2K
Fashion-MNIST (KD)	90.80	87.79	96.69	Black	60K	60K
Fashion-MNIST(Ours)	90.80	**80.90**	**89.10**	Black	48K	2K

Blackbox Teacher Models. For each task except Places365-Standard, we prepare a teacher model by following the training setting provided in [9]. For Places365-Standard, there is no training setting reference for the knowledge distillation research yet, so we use a pre-trained model from the dataset repository [66] as our teacher model. On MNIST and Fashion-MNIST, we use the LeNet-5 architecture [67] as the teacher model and optimize it to achieve 99.29% and 90.80% top-1 accuracies, respectively. On CIFAR-10, we have an AlexNet [68] as the teacher model and train it to obtain 83.07% top-1 accuracy. As shown in Table 3.1, the above teacher models are comparable to the teacher models in [9]: 83.03% *vs*. 83.07% on CIFAR-10, 99.34% *vs*. 99.29% on MNIST, and 90.84% *vs*. 90.87% on Fashion-MNIST. For Places365-Standard, the teacher model is a ResNet-18 [69] and yields 53.68% top-1 accuracy.

Competing Methods. We identify three existing relevant methods for comparison.

- One is zero-shot knowledge distillation (ZSKD) [9], which distills a student neural network with zero training example from a *whitebox* teacher model. It synthesizes data by backpropagating gradients to the input through the whitebox teacher network.
- The second method is few-shot knowledge distillation (FSKD) [7], which augments the training images by generating adversarial examples. It is the most relevant work to ours, but it depends on the computationally expensive adversarial examples [70] and has no active learning scheme to reduce the query cost at all. The original work assumes a *whitebox* teacher neural network so that it is straightforward to produce the adversarial examples, whereas there exist blackbox attack methods [54, 50].
- The third is the vanilla knowledge distillation [6], which accesses the whole training set of the teacher model and is somehow an upper bound of our method.

3.4.1.2 Quantitative Results

Table 3.1 shows the comparison results. For simplicity, we run the active learning stage for only one step (i.e., $T = 1$ in Algorithm 1). Section 3 presents the results of running it for multiple steps.

Accuracy. Our approach significantly outperforms FSKD over all the datasets. On CIFAR-10, MNIST, and Fashion-MNIST, ours yields 41%, 18%, and 14% success rate improvements over FSKD, respectively. On Places365-Standard, whose images are high-resolution about natural scenes, we also outperform FSKD by 14% success rate. Compared to ZSKD, which relies on a whitebox teacher network, our approach also shows higher accuracies and success rates except on MNIST. We were not able to reproduce ZSKD on Places365-Standard because its images are all high-resolution, making it computationally infeasible to generate a large number of gradient-based inputs. Similarly, the advantage of ours over ZSKD is larger on CIFAR-10 than other MNIST or Fashion-MNIST, probably because the CIFAR-10 images have a higher resolution. In contrast, the computation cost of our active mixup approach does not depend on the input resolution. Overall, the results indicate that active mixup has a higher potential to solve the larger-scale knowledge distillation in a data-efficient manner.

Queries. Our approach saves orders of queries into the teacher model compared to ZSKD. For example, we only query the blackbox teacher model up to 40K times for CIFAR-10. In contrast, ZSKD requires more than 2M queries and yet yields lower accuracy than ours. The big difference is not surprising because the gradient-based inputs in ZSKD are less natural than or representative of the test images than our mixup images. Besides, ZSKD incurs additional queries into the whitebox teacher model every time it produces an input.

3.4.1.3 Qualitative Intermediate Results

We show some mixup images in Figures 3.2 and 3.3. These images are selected from the candidate pool constructed using the natural images in the Places365-Standard training set. Figure 3.2 shows some mixup images with low confidence scores. They can potentially benefit the student network more than the other candidate images if we use them to query the teacher model. Figure 3.3 demonstrates some mixup images synthesized from the same pair of natural images by varying the mixup coefficient λ. It is interesting to see that the mix of "Hockey Arena" and "Golf Course" leads to a "Baseball Field" at $\lambda = 0.46$ predicted by the blackbox teacher model. This indicates that our active mixup approach can effectively augment the originally small training set by not only bringing in new synthetic images but also comprehensive coverage of classes.

3.4.2 Ablation Study

We select CIFAR-10 and Places365-Standard to study our approach in detail since they represent the small-scale and large-scale settings, respectively. For CIFAR-10, we switch to VGG-16 [71] as the blackbox teacher model, which gives rise to 93.31% top-1 accuracy.

3.4.2.1 Data-Efficiency and Query-Efficiency

We investigate how the results of our active mixup approach change as we vary the total number of unlabeled real images (data-efficiency) and the number of synthetic images selected by the active learning scheme (query-efficiency). Here we run only one step of the active learning stage ($T = 1$ in Algorithm 1) to save computation cost. Tables 3.2 and 3.3 show the results on CIFAR-10 and Places365-Standard, respectively. Each entry in the tables is a classification accuracy on the test set, and it is obtained by a student network which we distill by using the corresponding number of

unlabeled real images (Real images) and the number of selected synthetic images (Selected Syn.).

Table 3.2: Classification accuracy on CIFAR-10 with different numbers of real images and selected synthetic images.

Selected Syn. \ Real images	0.5K	1K	2K	4K	8K	16K
0	44.72	56.87	68.09	76.59	83.61	86.89
5K	66.97	71.67	77.76	81.76	85.76	87.05
10K	73.60	77.27	81.27	83.27	86.56	88.79
20K	77.44	81.18	84.19	86.29	88.07	89.01
40K	82.28	84.25	86.06	87.71	89.00	90.49
80K	85.18	86.53	87.89	88.71	89.61	90.96
160K	86.56	88.94	89.42	90.26	90.87	91.51

Table 3.3: Classification accuracy on Places365-Standard with different numbers of real images and selected synthetic images.

Selected Syn. \ Real images	20K	40K	80K
100K	40.72	41.95	43.52
200K	41.15	42.86	44.77
400K	41.94	43.42	45.71

We can see that the more synthetic images we select by their confidence scores (cf. eq. (3.3)), the higher-quality the distilled student network is. It indicates that the mixup images can effectively boost the performance of our method. Meanwhile, the higher the number of unlabeled real images we have, the higher the distillation success rate we can achieve. What's more interesting is that, when the number of synthetic images is high (e.g., 160K), the gain is diminishing as we increase the number of real images. Hence, depending on the application scenarios, we have the flexibility to trade-off the real images and synthetic images for achieving a certain distillation success rate.

We can take a closer look at Tables 3.2 and 3.3 to obtain an understanding about the "market values" of the selected synthetic images. In Table 3.2, 10K selected synthetic images and 8K unlabeled real images yield 86.56% accuracy; 20K synthetic images and 4K real images lead to 86.29% accuracy; and 40K synthetic images with 2K real examples give rise to 86.06% accuracy. The accuracies are about the same. There is a similar trend along the off-diagonal entries in Table 3.3, implying that if we reduce the number of real images by half, we can complement it by doubling the size of synthetic images to maintain about the same distillation success rate.

3.4.2.2 Active Mixup vs. Random Search

We design another experiment to compare active mixup with the random search to understand the effectiveness of our active learning scheme. We keep 500 real images for CIFAR-10 and 20K for Places365-Standard. We then use them to construct 100K and 300K synthetic images, respectively. For a fair comparison, we let random search and active mixup share the same sets of natural images. Since our active learning scheme avoids selecting redundant images by using the improved confidence score in eq. (3.3), we also equip the random search such capability by using a single mixup coefficient of $\lambda = 0.5$ to construct the synthetic images. This guarantees that, like our approach, no two synthetic images selected by the random search are from the same pair of real images.

Figure 3.4 shows the comparison results of our active mixup and the random search. On CIFAR-10, we select 10K synthetic images every time and run the active learning stage for 10 steps ($T = 10$ in Algorithm 1). On Places365-Standard, we run it for six steps and choose 50K synthetic images per step. We can see that active mixup significantly outperforms random search over the whole course of knowledge distillation, verifying its effectiveness on improving the query-efficiency. More concretely, 80K actively selected synthetic images yield 86.76% accuracy, which is about

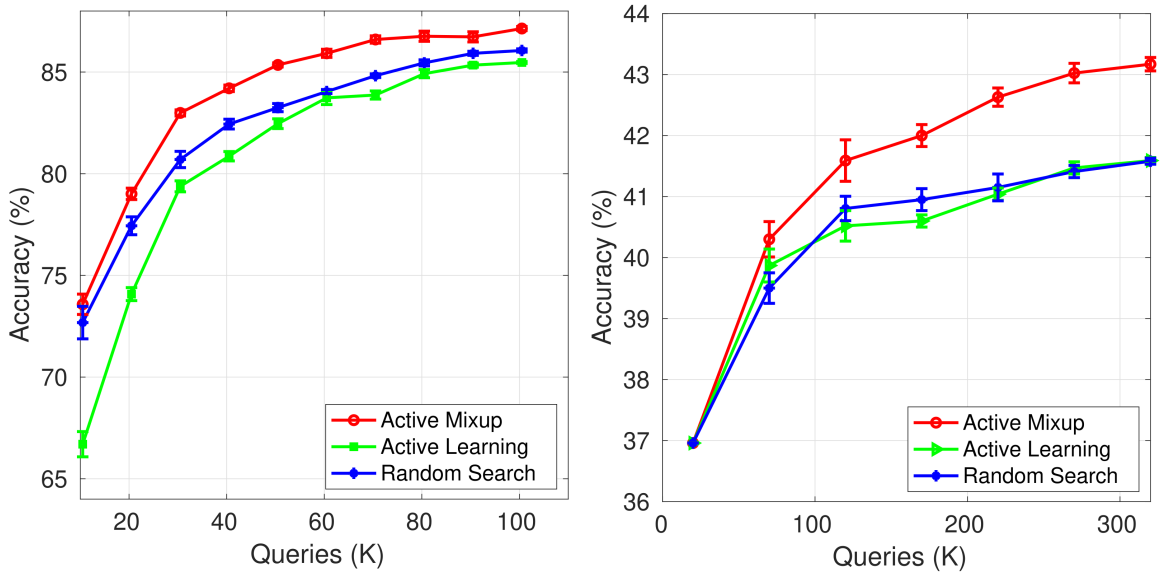


Figure 3.4: Test accuracy of student networks vs. number of queries into the blackbox teacher model on CIFAR-10 (left) and Places365-Standard (right). We use 500 and 20K natural images for the two datasets, respectively. The plot for CIFAR-10 starts from first active learning stage ($t = 1$ in Algorithm 1) and the one for Places365 starts from the initial student network training by natural images. The initial student network for CIFAR-10 trained by using natural images only yields 43.67% accuracy.

the same as what 160K randomly selected synthetic images can achieve on CIFAR-10. Similarly, 40K synthetic images by active mixup lead to 84.2% accuracy, on par with the 85.18% accuracy by 80K randomly chosen synthetic images.

3.4.2.3 Active Mixup vs. Vanilla Active Learning

Our active learning scheme (eq. (3.3)) improves upon the vanilla score-based active learning (eq. (3.2)) by selecting only one synthetic image at most from any pair of real images. This change is necessary because two nearly duplicated synthetic images could both have very low scores according to eq. (3.2).

To quantitatively compare the two active learning methods, we run another experiment by replacing

our active learning scheme with the vanilla version. The candidate pool is the same as ours, i.e., mixup images generated by varying $\lambda \in \{0.3, 0.7\}$ with an interval of 0.04. Figure 3.4 shows the results on both CIFAR-10 and Place365-Standard.

Generally, the vanilla active learning yields lower accuracy than our active mixup and the random search. This shows that the vanilla score-based active learning even fails to improve upon random search because it selects nearly duplicated synthetic images to query the teacher model. In contrast, our active mixup consistently performs the better than the vanilla active learning and random search. The prominent gap justifies that the constraint by C_2 in eq. (3.3) is crucial in our approach.

3.4.3 Active Mixup with Out-of-Domain Data for Blackbox Knowledge Distillation

In real-world applications, it may be hard to collect real training images for some tasks, e.g., due to privacy concerns. Under such scenarios, we have to use out-of-domain data to distill the student neural network. Hence, we further challenge our approach by revealing some images that are out of the domain of the training images of the blackbox teacher model.

We conduct this experiment on CIFAR-10 by providing our approach some training images in CIFAR-100 [72]. To reduce information leak, we exclude the images that belong to the CIFAR-10 classes and keep 2K images to construct the candidate pool. Equipped with these synthetic images, we run active mixup to distill student neural networks from a blackbox teacher model for CIFAR-10. The teacher model is VGG-16, which yields 93.31% accuracy on the CIFAR-10 test set.

Table 3.4 shows the results of different numbers of selected synthetic images. We still run only one iteration of the active learning to save computation costs. The best distillation performance is 83% top-1 accuracy and success rate is 88.9%. Comparing the result to Table 3.2, especially the

Table 3.4: CIFAR-10 classification accuracy by the student neural networks which are distilled by using out-of-domain data.

Selected Syn.	10K	20K	40K	80K
Accuracy (%)	64.10	71.39	77.89	83.03

entry (87.89%) of 80K selected synthetic images and 2K real images, we can see that our approach leads to about the same performance by using the out-of-domain data as the in-domain data.

Table 3.5: CIFAR-10 classification accuracy by the student neural networks which are distilled by using out-of-domain data. We set the number of selected synthetic images to 40K and vary the numbers of real images.

Real images	500	1000	1500	2000
Accuracy (%)	70.21	74.60	75.54	77.89

To better understand how different factors influence the distillation performance, we also decouple the number of available real images from the number of selected synthetic images in Table 3.5. We fix the number of selected synthetic images to 40K and vary the numbers of real images. Not surprisingly, the more real images there are, the higher distillation accuracy the active mixup achieves. Furthermore, the number of synthetic images still plays a prominent role in distillation accuracy, according to Table 3.4. Without the original training data, mixup augmentation is probably more critical to enhancing the distillation performance than otherwise.

3.5 Summary

In this paper, we formalize a novel problem, knowledge distillation from a blackbox teacher model in a data-efficient manner, which we think is more realistic than previous knowledge distillation

setups. There are two key challenges to this problem. One is that the available examples are insufficient to represent the vast variation in the original training set of the teacher model. The other is that the blackbox teacher model often implies that it is financially and computationally expensive to query.

To deal with the two challenges, we propose an approach combining mixup and active learning. Although neither of them is new by itself, combining them is probably the most organic solution to our problem setup for the following reasons. First of all, we would like to augment the few available examples. Unlike conventional data augmentations (e.g., cropping, adding noise), which only probe the regions around the available examples, mixup provides a continuous interpolation between any pairwise examples. As a result, mixup allows the student model to probe diverse regions of the input space. We then employ active learning to reduce the query transactions to the teacher model. Extensive experiments verify the effectiveness of our approach to the data-efficient blackbox knowledge distillation.

CHAPTER 4: DEEP EPIDEMIOLOGICAL MODELING BY BLACK-BOX KNOWLEDGE DISTILLATION

4.1 Problem Introduction

The spread of infectious diseases is a serious threat to public health and may cause million deaths every year. To effectively battle against infectious diseases, accurate modeling on their transmission patterns is critical. This issue becomes more pressing when the infectious disease, like COVID-19, is unprecedented, transmission dynamics is complex, and observation data are limited. Due to data limitation, we need to solve this problem with the help of conventional physics-based epidemiological models. However, it is still difficult to accurately describe complex dynamics with a single model.

Mixture models are widely used to accurately solve complex transient modeling problems. They can refine temporal scale into several states with different onsets, model these states separately, and then mix modeling results to represent complex dynamics. Although this refinement on temporal scale more accurately depicts the variation in a physical system, the difficulty of calibrating a mixture model and computational complexity can exponentially increase since it can result in very large parameter space, *i.e.*, curse of dimensionality. When prior knowledge about an infectious disease, such as COVID-19, is limited, exhaustive search in such large space is inevitable for accurate model calibration, which can easily render a mixture model impractical. In reality, some modelers

This chapter contains previously published materials from "Deep epidemiological modeling by black-box knowledge distillation: An accurate deep learning model for covid-19.", by Dongdong Wang, Shunpu Zhang, and Liqiang Wang. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 17, pp. 15424-15430. 2021.

propose some assumptions to truncate search space with coarse grid and trade for efficiency and feasibility, but it can cause large uncertainty and model degradation.

To address this problem, we formulate a new approach with black-box knowledge distillation. This approach is developed based on three-fold objectives, including higher prediction accuracy, lower modeling cost, and higher data efficiency. To achieve higher prediction accuracy, we first leverage mixture models to create a comprehensive, accurate, but probably impractical epidemic simulation system. This system is viewed as a black-box teacher model which contains sophisticated modeling knowledge. To reduce modeling cost and make this system feasible, we employ knowledge distillation to transfer the accurate modeling knowledge from this impractical black-box teacher model to a deep neural network for practical use. To realize this knowledge transfer, we collect a set of simulated observation sequences to query the teacher model and acquire their corresponding simulated projection sequences as knowledge. Particularly, for improvement in model performance with limited data, we propose sequence mixup to augment data pool, thus reducing model queries, increasing sequence diversity, and boosting modeling accuracy. With all retrieved and mixed observation-projection sequence pairs, we train a student deep neural network for infection prediction. This student network can perform prediction as accurately as teacher model, but save lots of computation cost, and require fewer observation data.

To the best of our knowledge, we are the first to propose a black-box knowledge distillation based framework to solve epidemiological modeling by leveraging mixture models. Besides this novelty, our work also includes the following contributions: (1) the distilled student deep neural network enables accurate model calibration and projection automatically. (2) Sequence mixup is proposed to reduce teacher model queries for higher efficiency, improve the coverage of obtained data for better accuracy, and further enhance knowledge transfer with fewer observation data. (3) We justify our approach by solving COVID-19 infection projection and it performs on par or even better than some state-of-the-art methods, like CDC Ensemble, with adequate accuracy over the evaluation

period. (4) Our approach provides a general solution to render impractical physics-based models feasible.

4.2 Background

4.2.1 Epidemiological Modeling

Epidemiological modeling has been extensively studied for decades. It is focused on how to accurately quantify infectious disease transmission dynamics. The proposed methods can be classified into two main categories, classical physics-based modeling and data-driven approach. For physics-based modeling, compartmental modeling, like SEIR [73], is well justified for practical projection. Different from physics-based modeling, thanks to the improvement on data collection, data-driven approaches have been developed based upon statistical modeling on real observation data and widely used for transmission dynamics projection, such as ARIMA[74] and ARGO[75, 76]. With rapid advances in artificial intelligence, deep learning based modeling as an alternative is proposed to solve infection projection, especially for emergency pandemic like COVID-19 [77, 78, 79, 80]. However, these data-driven approaches can suffer from observation data limitation. Recently, a hybrid approach named DEFSI [81] adopts compartmental modeling to alleviate data limitation problem in deep neural network training.

4.2.2 Knowledge Distillation

Knowledge distillation [6] is widely used to solve deep neural network compression problem. Conventional distillation process is carried out by training a smaller neural network called student model with class probability, which is referred to as “dark knowledge", to retain the performance of original cumbersome ensemble of models called teacher model. This approach can effectively

reduce model size, which makes complex models feasible for real-world applications. Many complex applications in computer vision or natural language processing have justified its merits for model size reduction. For example, DistilBERT [32] successfully reduces the size of original BERT model by 40% with maintaining accuracy; TinyBERT [33] leverages knowledge distillation to design a framework for the reduction of transformer-based language model, which leads to the models with lower time and space complexity, thus facilitating its application; relational knowledge distillation [34] further optimizes distillation process and enables more productive student model, which can even outperform teacher model. However, this effective approach has not been applied to solve complex epidemiological modeling, especially the infeasibility of mixture epidemiological models.

4.2.3 Mixup

Mixup is a simple yet effective approach to augment training data and improve model performance [1]. This method is proposed to improve the generalization of deep neural network by enhancing coverage of data distribution, especially when training data are limited. The main idea is to incorporate convex combination into data synthesis, which involves mixing features and mixing labels. It has been widely used to address computer vision and natural language processing problems, like Between-Class learning in speech recognition [38] and image classification[39], AutoAugment with learning strategy augmentation for classification [40], and wordMixup or senMixup with embedding mixup for sentence classification [41]. More studies explore its potential for data-efficient learning, such as active mixup [35] and ranking distillation in [42]. However, there is no work using mixup to enhance epidemiological modeling efficacy and efficiency.

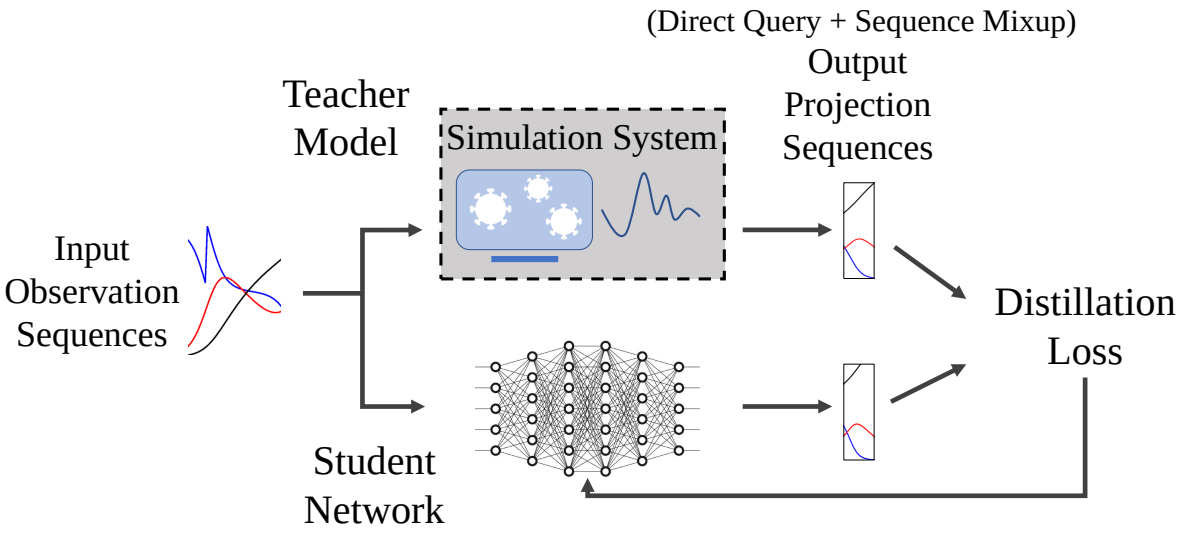


Figure 4.1: Modeling with black-box knowledge distillation. Teacher model is an accurate but significantly complex comprehensive simulation system. Both observation and projection sequences are simulated results. Model query is optimized by sequence mixup.

4.3 Methodology

Figure 4.1 shows an overview of our approach on epidemiological modeling by black-box knowledge distillation. We leverage mixture models to build a comprehensive simulation system with accurate modeling knowledge yet significantly high complexity. Then, we use simulated observation sequences to query this system to retrieve simulated projection sequences as knowledge. To improve query efficiency and enhance knowledge transfer, sequence mixup is designed to further efficiently augment data pool. With retrieved and mixed observation-projection sequence pairs, a deep neural network is trained to retain the modeling accuracy of the original impractical simulation system and prepared for practical use.

4.3.1 Developing a Teacher Model

Many approaches can be used to create mixture models and build a comprehensive simulation system $\mathscr{M}$. To ensure reliability, we select a widely accepted compartmental model of SEIR as the modeling approach. In SEIR, people in the modeled society, aka host society, must be in one of the four health states, *i.e.*, susceptible, exposed, infectious, and recovered. The state transition starts from “susceptible", and then moves to “exposed", then to “infectious", and finally reaches “recovered" state. Thus, the model is constrained with the boundary condition of $N = S + E + I + R$, where S, E, I, and R denote susceptible, exposed, infected, and recovered population, respectively, and N represents the population of the entire host society.

For accurate depiction of transient transmission dynamics, we employ linear mixture model [82] to represent the heterogeneity of host society [83]. The host society N is divided into several component host communities N_i with the linear combination in Equation 4.1, and modeling results from these communities will be mixed to represent the dynamics of entire host society N. The division of host society is based on heuristics, which depends on modeling resolution.

$$N = \sum_{i=0}^{n} N_i = \sum_{i=0}^{n} (S_i + E_i + I_i + R_i) \tag{4.1}$$

Within each community N_i, transmission dynamics can be described by an ordinary differential equation (ODE) system, as shown in Equation 4.2, across all compartments.

$$\begin{aligned} \frac{dS_i}{dt} &= \alpha N_i - \beta S_i^t I_i^t - \mu N_i S_i^t \\ \frac{dE_i}{dt} &= \beta S_i^t I_i^t - (\sigma + \mu) E_i^t \\ \frac{dI_i}{dt} &= \sigma E_i^t - (\gamma + \mu) I_i^t \\ \frac{dR_i}{dt} &= \gamma I_i^t - \mu R_i^t \end{aligned} \tag{4.2}$$

where S_i^t, E_i^t, I_i^t, and R_i^t denote susceptible, exposed, infected, and recovered population, respectively, at time t. β, σ, and γ denote infectious, latent, and recovery rate over the entire incidence, respectively. α and μ are referred to as natural birth and death rates during this period, respectively, which are assumed to be zero in this study.

SEIR modeling is a typical boundary value problem [84], the solution of which relies on boundary condition (BC), initial condition (IC), and ODEs. In this study, for each component host community, constant BC is assigned by the total population N_i due to no vital dynamics, IC is determined by the compartment state information $\{S_i^0, E_i^0, I_i^0, R_i^0\}$ at time step $t = 0$, and ODEs are specified by the dynamics coefficients $\{\beta, \sigma, \gamma\}$. Conventional numerical modeling requires model calibration, , which adjusts parameters to obtain agreement between real observation data and modeled results, using grid search for an optimal combination of BC, IC, and ODEs ($\{$BC, IC, ODEs$\}$) within constraints in search space. If the search space for $\{$BC, IC, ODEs$\}$ is larger and fine-grained, the calibration results are better fit to the real observation data and simulated projected results are more reliable. Therefore, we construct a comprehensive simulation system with an ensemble of simulation scenarios from large and fine search space, which enables accurate model calibration and projection.

However, the complexity of this simulation ensemble system is very time-consuming for grid search due to curse of dimensionality. For example, suppose we have just 2 options for BC, IC, and ODEs (the real problems require much more). For each component host community, there are 8 simulation scenarios. However, if we have 10 component communities, the ensemble for the entire society N will reach 8^{10} simulation scenarios. It is infeasible to find an optimal solution with random grid search. Therefore, we conduct knowledge distillation to distill this ensemble simulation system into a deep neural network for practical use.

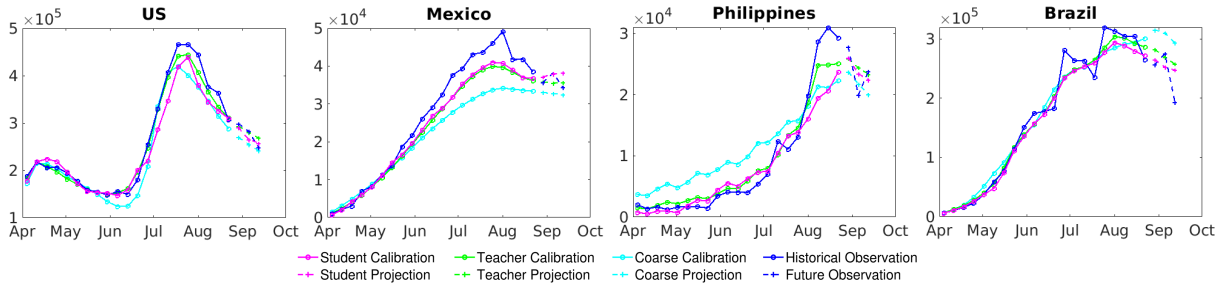


Figure 4.2: Weekly new infection cases over the calibration (04/06-08/23) and projection (08/24-09/13) periods by teacher model, student network, and coarse search.

4.3.2 Querying the Teacher Model

Conventional knowledge distillation is carried out by querying the teacher model to obtain prediction probabilities that are referred to as “knowledge". In our problem, the “knowledge" are simulated projection sequences from the simulation system since they contain the features of modeling process. To facilitate acquiring such kind of modeling “knowledge", we conduct model querying as follows. First, we prepare a simulated observation sequence over the calibration period with a $\{$BC, IC, ODEs$\}$ for each host community. Each $\{$BC, IC, ODEs$\}$ is used as a “key" to query teacher model. Then, the teacher model will use the “key" to return a query answer with a simulated sequence over the calibration and projection period, *i.e.*, a projection sequence. With more queries, more projection sequences are obtained and more accurate modeling knowledge is acquired.

4.3.3 Sequence Mixup

To ensure adequate knowledge, distillation usually requires lots of training data from many model queries. However, too many queries can be time-consuming, and more importantly, the simulated observation sequences are still too limited to acquire diverse knowledge. For improvement in

distillation efficacy and data diversity, we employ sequence mixup to reduce the number of queries and enlarge knowledge coverage.

$$\begin{aligned} \hat{x} &= \omega_1 x_1 + \omega_2 x_2 + ... + \omega_n x_n \\ \hat{y} &= \omega_1 y_1 + \omega_2 y_2 + ... + \omega_n y_n \end{aligned} \tag{4.3}$$

Our sequence mixup is developed with convex combinations of multiple observation sequences x_i and projection sequences y_i with mix rates ω_i, where $\Sigma\omega_i = 1$. Equation 4.3 presents this mixup process which mixes observation sequences x and projection sequences y in the same manner.

$$\begin{aligned} S^{t+1} &= S^t + \frac{dS^t}{dt} = \sum_{i=1}^{n} \omega_i S_i^t + \frac{d\sum_{i=1}^{n} \omega_i S_i^t}{dt} \\ &= \sum_{i=1}^{n} \omega_i S_i^t + \sum_{i=1}^{n} \frac{d\omega_i S_i^t}{dt} = \sum_{i=1}^{n} \omega_i S_i^t + \frac{d\omega_i S_i^t}{dt} \\ &= \sum_{i=1}^{n} \omega_i S_i^{t+1} \end{aligned} \tag{4.4}$$

The mixup projection sequence $\hat{y}$ in Equation 4.3 uses the same coefficients $\omega_1, \omega_2, ..., \omega_n$ as in $\hat{x}$ and it can be briefly proved as follows. Suppose $\hat{x}$ denotes $S^t = \sum_{i=1}^{n} \omega_i S_i^t$ at the current observation time and $\hat{y}$ denotes S^{t+1} at the next projection time. Given the linearity of differentiation, this mixup process $S^{t+1} = \sum_{i=1}^{n} \omega_i S_i^{t+1}$ is justified in Equation 4.4. Similar proof can be completed for E, I, and R.

These mixed sequences as an alternative to query knowledge efficiently augment training data and enhance the knowledge transfer from teacher model. Thus, all retrieved and mixed sequences construct a training set (X, Y).

4.3.4 Training a Student Deep Neural Network

With the acquired observation-projection sequence pairs (X,Y), a deep neural network is trained to distill the modeling knowledge within the comprehensive simulation system. The conventional distillation process is carried out by minimization on the distillation loss function $L_{dis} = D_1(y_n^{true}, S(x_n)) + D_2(T(x_n), S(x_n))$, where $T(x_n)$ is the output of data x_n from teacher model T, $S(x_n)$ is the output of data x_n from student network S, D_1 is the supervised loss for supervised learning with data label y_n^{true}, and D_2 is the imitation loss for model output imitation. In our problem, there is no knowledge about the true label y_n^{true} for x_n, and thus, the distillation loss is modified to the imitation loss only, as shown in Equation 4.5. We select mean squared error loss as distillation loss function.

$$L_{dis} = D_2(T(x_n), S(x_n)) \tag{4.5}$$

The proposed black-box knowledge distillation is a general approach that can be applied to different student networks. In the problem of COVID-19, we use multilayer perceptron (MLP) which is detailed in the case study.

Table 4.1: Error assessment of model calibration (04/06 - 08/23) and projection (08/24 - 09/13).

Metric	Model	Calibration				Projection			
		US	Mexico	Philippines	Brazil	US	Mexico	Philippines	Brazil
MAPE	Teacher	**0.0363**	0.1217	**0.3197**	0.0879	**0.0352**	**0.0369**	0.1030	0.1522
	Student	0.0695	**0.1164**	0.3472	**0.0792**	0.0433	0.0527	**0.0984**	**0.1331**
	Coarse	0.0843	0.2269	1.3159	0.1438	0.0727	0.0910	0.1314	0.2923
RMSE (10^5)	Teacher	**0.669**	0.183	**0.101**	**0.790**	**0.209**	**0.028**	0.048	0.703
	Student	1.321	**0.163**	0.163	0.857	0.218	0.041	**0.041**	**0.593**
	Coarse	1.426	0.333	0.229	0.985	0.399	0.063	0.059	1.215

4.3.5 Overall Algorithm

Algorithm 2 presents the overall procedure of our proposed black-box knowledge distillation based epidemiological modeling. Beginning with a modeling approach, a comprehensive epidemic simulation system is built as a teacher model $\mathscr{M}^T$. We then pick a few simulated observation sequences x to query the teacher model and retrieve their simulated projection sequences y. With obtained sequences (x,y), we construct a large observation-projection pool (X,Y) using sequence mixup. Finally, we train a student deep neural network $\mathscr{M}^S$ with (X,Y).

Algorithm 2 Epidemiological Modeling with Black-box Knowledge Distillation

INPUT: A modeling approach F such as mixture SEIR.
INPUT: A set of observation sequences $X_{obs} = \{x_i\}_{i=1}^n$.
INPUT: Hyper-parameters (mixup rate, learning rate etc.)
OUTPUT: A student deep neural network $\mathscr{M}^S$

1: Develop a comprehensive simulation system $\mathscr{M}^T$ based upon F with a set of conditions {BC, IC, ODEs}s
2: With all observation sequences in X_{obs}, query simulation system $\mathscr{M}^T$, retrieve projection sequences $Y_{query} = \{y_i\}_{i=1}^n$, and form an observation-projection pool (X_{obs}, Y_{query}).
3: Construct a mixed sequence pool $(X_{mix}, Y_{mix}) = \{(\hat{x},\hat{y}) : (\hat{x},\hat{y}) \in (\sum_{i=1}^n \omega_i x_i, \sum_{i=1}^n \omega_i y_i)\}$ with query results (X_{obs}, Y_{query}), where ω is heuristically chosen.
4: Train a student deep neural network $\mathscr{M}^S$ with $(X,Y) = (X_{obs}, Y_{query}) \cup (X_{mix}, Y_{mix})$ to minimize distillation loss L_{dis}.

4.4 COVID-19 Case Study

4.4.1 Experiment Setting

Data. We evaluate our approach on the open COVID-19 datasets provided by Johns Hopkins University [85]. In this dataset, our experiments are focused on daily infection case increase. With these reported data, we derive active infection cases based on 7-day transmission duration [86], as

the data do not explicitly report the number of recovered patients. The observation period starts from 04/06/2020 to 08/23/2020 and the evaluation period is from 08/24/2020 to 09/13/2020.

Black-box Teacher Model. A black-box teacher model is built with aforementioned mixture SEIR. The mixture model consists of 10 compartment host communities. Each compartment host community is simulated with 10 choices for N_i to specify constant BC, 2 choices for $\{S_i^0, E_i^0, I_i^0, R_i^0\}$ to specify IC, and 20 choices for each coefficient in $\{\beta, \sigma, \gamma\}$ to specify ODEs. Such choices of parameters are based on heuristics. Most studies on COVID-19 using SEIR model give a wide range of parameter choices [87]. We refine them to more reliable ranges. With the refined parameter choices, this simulation system contains 160000^{10} scenarios for the entire society N, which is impractical. To facilitate distillation assessment, we conduct random sampling to reduce it to 10^7 scenarios as an approximate version of teacher model to the simulation system for comparative study. The teacher model generates a simulated projection sequence by minimizing the mean squared error between real observation and the simulation over the calibration period, which is similar to exhaustive search.

Query Sequences and Mixup. We randomly pick 1000 {BC, IC, ODEs}s to prepare simulated observation sequences which are used to query teacher system. Note that, compared to the size of the ensemble, this number is so limited that we acquire little knowledge about simulation system with selected sequences, which still follows black-box teacher model setting. Given 1000 query results, we construct a large pool with 100K sequences by sequence mixup, where ω is set heuristically.

Student Deep Neural Network Training. Our student network architecture is an MLP which has 3 hidden layers with 80 neurons each. The batch size is 128 and learning rate is set to 0.1. Adam optimizer is chosen. Weight decay is specified to 1e-5. The total epoch is set to 300 and learning rate is reduced by 90% after every 100 epochs. We select 1K sequences from the constructed

sample pool as a training set for efficient training.

Studied Cases. We implement our black-box distillation framework to distill comprehensive infection modeling system for US, Mexico, Philippines, and Brazil. The infection patterns of these countries are representative of complex dynamics which involves multiple peaks and complicates model calibration. To achieve an adequate teacher model on each studied country, we heuristically specify the search space boundaries for {BC, IC, ODEs}s with the information of national population, reported positive cases on March 30th (a week before April 6th), and outbreak severity for each country.

Evaluation Metric. We evaluate infection case modeling performance on both accuracy and efficiency. For accuracy, model calibration and projection are assessed. The performance is quantified by mean absolute percentage error, MAPE $= \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i^o - y_i^m}{y_i^o}|$, and root mean square error, RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^o - y_i^m)^2}$, where y^o is the real observation sequence, y^m is the modeled sequence, and n is the total number of sequences. MAPE and RMSE are two widely adopted metrics to evaluate regression models. While lower MAPE suggests that the general trend is better captured, higher error can occur at larger observation data. RMSE is a better indicator for large values since it offers higher penalty for these errors. Therefore, we use both metrics for accuracy evaluation.

As to computation efficiency, we evaluate model complexity with required simulation scenarios and total time cost for each projection query. For student network, the network training cost is included in each query process although network retraining is not always necessary.

Competing Methods.

First, we compare our approach with the approximate teacher model and coarse search to examine accuracy and efficiency. Coarse search is developed upon coarse grid search space for mixture models. We reduce the number of compartment communities to 5, the options for BC to 5, and the

choices for each ODE coefficient to 10, which could be taken as a reduced teacher model, but still with the complexity of 10000^5. Similar to teacher model, for practical performance evaluation, we reduce it to 10^5 scenarios with random sampling, which ensures its similar data complexity to student network. In the following sections, approximate teacher model and coarse grid search are referred to as teacher model and coarse search, respectively. Next, we compare our student network with 7 state-of-the-art forecasting models reported from CDC [88]. These models are developed with machine learning based methods, like UM and UCLA-SuEIR, statistical methods, like DDS, physics-based model, like JHU-IDD and Columbia, and ensemble approaches, like UVA and CDC Ensemble [89].

Table 4.2: MAPE comparison of state-of-the-art models and our method on US weekly infection case increase projection between 08/24 and 09/13. The results of other models are collected from CDC, which are reported by COVID-19 Forecast Hub.

Period	Model							
(from 08/23)	CDC Ensemble	UM	DDS	UVA	UCLA	JHU	Columbia	Ours
1 week ahead	0.0608	0.3866	0.0417	0.0698	0.0367	0.0737	0.0456	**0.0301**
2 week ahead	0.1108	0.0386	**0.0228**	0.0772	0.0889	0.1165	0.0250	0.0623
3 week ahead	0.0581	0.0549	0.0819	0.2724	**0.0077**	0.2572	0.2083	0.0398

Table 4.3: Model complexity measured by the required simulations and the CPU time cost for one projection query.

	Complete Teacher	Approximate Teacher	Student Network	Coarse Search
Simulations	160000^{10}	10^7	10^3	10^5
Time(s)	N/A	$\sim 3\times 10^4$	~ 400	~ 300

4.4.2 Results

Accuracy. Our calibration and projection results are reported with weekly increase cases in Figure 4.2. Student network is comparable to teacher model and significantly outperforms coarse search. These performance differences are quantified with MAPE and RMSE in Table 4.1. It is shown that, compared to the teacher model, student network achieves similarly low or even lower MAPE and RMSE, over the calibration or projection periods. This observation results from the approximation of teacher model and sequence mixup for student network training. Coarse search yields highest errors due to limited search space.

We compare our student network with 7 state-of-the-art models in Table 4.2, which are based on the reported data from CDC [88]. Our model consistently outperforms CDC Ensemble, which incorporates all reported state-of-the-art models, with $30\%-50\%$ MAPE reduction over this period. In particular, our model yields more accurate 1 week ahead prediction and more consistent performance over three weeks compared to other models.

Table 4.4: Calibration and projection errors from student network for US with 100K, 50K, and 25K mixed sequences.

	Metric	100K	50K	25K
Calibration	MAPE	**0.0695**	0.0987	0.1459
	RMSE(10^5)	**1.321**	1.831	2.910
Projection	MAPE	**0.0433**	0.1861	0.2813
	RMSE(10^5)	**0.218**	0.985	1.367

Efficiency. From Table 4.3, student network saves both simulations and time cost by orders of magnitude. Student network and coarse search are on par in total time cost, while the network training takes approximately 300 CPU seconds in our study. This performance gain results from the optimization with sequence mixup and lightweight network design. It justifies that our approach significantly improves modeling efficiency and can facilitate the application of complex

and cumbersome epidemiological models.

Significance of Mixup. Sequence mixup, as an efficient method for data augmentation, is very important to enhance knowledge transfer in our approach. Compared to coarse search and teacher model, our student network can learn more scenarios out of search space due to sequence mixup, and this knowledge can overcome the limit from search space, thus even improving calibration and projection accuracy. To justify its importance, we conduct experiments with 100K, 50K, and 25K mixed sequences from 1000 retrieved observation-projection sequences and evaluate their performance difference in calibration and projection for US. From Table 4.4, the reduction in mixed sequences causes model degradation. The degradation becomes worse in the projection period due to calibration error propagation. Thus, sequence mixup is critical to accurate projection.

4.4.3 Discussion.

First, a comprehensive and accurate modeling system is critical in our framework. When this comprehensive teacher model is more complex and accurate, our student network can yield more accurate results. Next, student network can interpolate information in latent space which can resolve space discretization problem in grid search. The space of grid search is often too sparse to find an optimal solution. Therefore, dense search space is imperative, but its cost will exponentially increase. This can be alleviated by our proposed knowledge distillation. In addition, sequence mixup improves training data coverage and boosts model distillation, which helps student network even outperform teacher model. It implies that our proposed knowledge distillation scheme has potential to improve teacher model. Also, if a well-trained student network is obtained, the model could be reused many times, even when new data are included. In contrast, conventional random grid search, like teacher model or coarse search, has to be reset and query all entries again to retrieve projection solutions. This implies student network can save extra query cost.

4.5 Summary

We propose an innovative accurate modeling approach which leverages mixture models to ensure high accuracy and employs black-box knowledge distillation to reduce complexity and improve accuracy. It consists of teacher model development, model querying, sequence mixup, and student network training. The developed teacher model is a comprehensive simulation system which can accurately model challenged transient dynamics but is impractical. Then, we prepare simulated observation sequences to query this simulation system and retrieve simulated projection sequences as knowledge for distillation. In particular, to save number of queries and enhance knowledge transfer, sequence mixup is designed and effectively augments training data. With retrieved and mixed observation-projection sequences, a student deep neural network is trained as a distilled model for practical use. Our COVID-19 case study on US, Mexico, Philippines, and Brazil justifies that this approach brings in high accuracy but lower complexity. Also, our approach outperforms some state-of-the-art methods, like CDC Ensemble, over the studied period. In future, this work will be extended and applied to more epidemiological studies.

CHAPTER 5: SIMPLE YET EFFECTIVE MODEL UNCERTAINTY REDUCTION

5.1 Problem Introduction

This paper aims at providing a general and effective model uncertainty reduction approach, thus improving in-distribution (InD) model calibration and out-of-distribution (OoD) data detection. Recently, DNN, as a sophisticated modeling approach, has attained tremendous success in many complex practical problems, such as computer vision, natural language processing, and speech recognition. Although this modeling approach exhibits impressive performance on model accuracy, the reliability of DNN is still an issue for real applications. The work in [13] shows that DNN can exhibit over-confidence on the prediction results, which is risky for many safety-critical applications such as autonomous driving. Therefore, model uncertainty reduction is critical and pressing to neural network applications.

Recently, increasing number of research efforts start to explore this problem. [90] formulates and discusses the model uncertainty problem in deep learning and highlights its importance for DNN applications. [46] refines the problem where distributional uncertainty is considered for addressing data distribution shift issue. Given the formulations, there are several methods to reduce model uncertainty. The straightforward method is to find an integral solution, but it is generally intractable[46]. Approximation approaches can be helpful, such as Bayesian Monte Carlo[91] and point estimation, but they still have some limitations. Bayesian learning may help find more accurate solution, but it consumes lots of time and model storage. Ensemble model can be taken as a simplified Bayesian model[45], but it not only consumes lots of time and storage for inference, but also exhibits lower estimator efficiency due to high variance [92]. Conventional single model

training based upon point estimation can save storage and show high estimator efficiency in virtue of low estimator variance, but it can easily cause biased estimator. Accordingly, it is a challenge to seek an unbiased and efficient model estimator, which is essential for model uncertainty reduction.

To tackle this challenge, we propose a simple yet effective model uncertainty reduction approach leveraging point estimation. This approach employs ensemble model to correct the estimator bias first. Then, it integrates knowledge distillation with data augmentation to reduce the estimator variance and improve estimator efficiency. After this effective integration, the obtained model is a bias-reduced and efficient point estimator. This approach is more unbiased and more efficient than deep ensemble such as [5].

The contributions of this work include:

- This paper develops a simple yet effective approach EKD (**E**nsemble **K**nowledge **D**istillation) to reduce model uncertainty. Our method is easily implementable and scalable to any model neural network.
- To the best of our knowledge, we are the first to leverage a bias-reduced and efficient model point estimator to solve model uncertainty problem. Our proposed approach blends ensemble model and knowledge distillation to effectively address model estimator bias and variance simultaneously, thus successfully reducing model uncertainty with point estimation.
- Our study reveals that this simple model uncertainty reduction approach can directly improve InD model calibration and OoD data detection, simultaneously.
- This work studies the effectiveness of different data augmentations on ensemble knowledge distillation and reveals that AutoAugment [40] is an effective approach for model uncertainty reduction by ensemble distillation.

- This work conducts extensive experiments to justify this approach across different scale of datasets and various model architectures such as CNN models and Vision Transformer (ViT) variants.

5.2 Background

5.2.1 Model Uncertainty

Although DNNs achieve wide successes on lots of applications, model reliability is always a concern for its practical use. For example, the study in [93] reveals that DNNs can predict with high confidence for the images unrecognizable to humans. The research in [13] further highlights the overconfidence problem in modern DNNs. This overconfidence issue can cause the high vulnerability to very small perturbation on images and misclassification with high probability prediction [94]. To address these issues on performance generalization, [95] puts forward a baseline approach of maximum softmax probability to distinguish OoD images from misclassified InD images. [96] proposes an approach named ODIN with post-processing on model outputs to enhance the model reliability of OoD image detection. [97] develops a unified framework for OoD detection by improving confidence analysis with energy score.

The work most relevant to our approach is Deep Ensemble (DE) [5]. DE [5] excels in model calibration on InD data due to its approximate form of Bayesian model average [45], but OoD detection can be affected due to the mix of misclassification and OoD detection. According to the formulation in [46], the data uncertainty is further separated into data and distributional uncertainties, which are referred to InD calibration and OoD detection, and a prior network is designed to distinguish InD and OoD data.

5.2.2 Knowledge Distillation

Knowledge distillation [6] is proposed to compress large DNNs to smaller ones. This work reveals that the probability outputs from large networks can be taken as "dark knowledge" to retain its performance on accuracy with light-weight substitute models. The research with further exploration also finds that this approach is effective on model accuracy improvement. For example, On-the-fly Native Ensemble (ONE) equipped with ensemble distillation outperforms ensemble model [10]. Self-distillation with retraining the model can help increase model accuracy [11, 12].

The most relevant methods to our approach are ensemble distribution distillation [14] and batch ensemble distillation [15]. Extending from [46], Dirichlet distribution is employed to describe prior distribution and help distill ensemble networks, which improves model uncertainty distillation [14]. However, it require lots of ensemble networks to sample model output distribution, which causes training a prior network is computationally expensive and impractical for large models. Moreover, Dirichlet distribution is an approximate solution to ensemble model outputs, which is not always guaranteed. This leads to the gap between ensemble model and distilled distribution network. [15] makes use of batch ensemble training with more diverse data by input perturbation to fill the gap between ensemble model and distilled network, but the ensemble model as an upper bound still limits model uncertainty performance.

5.2.3 Data Augmentation

Sufficient training data are critical to DNN training. Lots of approaches are proposed to improve data augmentation effectiveness. AutoAugment [40] optimizes augmentation policies given a range of image processing techniques. Mixup [1] blends different images with linear combination and generates the labels with linear interpolation. Inspired from the striking perforamnce of

pretrained models in natural language processing tasks, contrastive learning based pretraining is carried out for computer vision DNNs. SimClr [98] employs stronger data augmentation to significantly enhance model recognition on image variation in a self-supervised learning manner. MoCo [99] incorporates self-attention mechanism in Transformer network into contrastive learning process, thus improving image recognition.

Recently, increasing number of work starts to study the impact of data augmentation on model uncertainty performance. For example, Mixup is explored and found effective on model calibration improvement when models are trained from scratch [100]. However, combining mixup with ensemble model can adversely affect model calibration performance [101].

5.3 Point Estimation

According to [46], the source of DNN uncertainty is composed of data and model uncertainty, which can be formulated as:

$$P(y|x^*, \mathscr{D}) = \int \underbrace{p(y|x^*, \theta)}_{\text{Data}} \underbrace{p(\theta|\mathscr{D})}_{\text{Model}} d\theta, \tag{5.1}$$

where P is the inference probability for test input x^* on label y given the model θ trained with data D. This formulation shows that data uncertainty is confounded with model uncertainty. This confounding harms model prediction confidence, resulting in inferior InD model calibration and weaker OoD data detection [46]. To alleviate this confounding, probability density function (PDF) of $p(\theta|\mathscr{D})$ is required for integration. Unfortunately, it is intractable, so estimation is needed.

Point estimation is a simple but effective approach to approximate model parameters [92]. As shown by [46], it approximates the PDF of $p(\theta|\mathscr{D})$ to a Delta function concentrating on the true

parameter $\hat{\theta}$, as shown in eq.5.2.

$$p(\theta|\mathscr{D}) = \delta(\theta - \hat{\theta}) \tag{5.2}$$

A good point estimator requires both unbiasness and efficiency [92]. An unbiased estimator indicates that the derived θ is sufficiently close to $\hat{\theta}$. An efficient estimator indicates that the variance of θ is small to ensure the concentration on $\hat{\theta}$. To obtain an unbiased estimator, we select ensemble model average as a solution to bias reduction. Suppose that the members in ensemble models are independent and identically distributed (i.i.d.). They are viewed as model estimators. When estimators are consistent, the more number the sampled models include, the more accurate the average estimation is [102]. To derive an efficient estimator, we distill ensemble model with data augmentation, which significantly reduces estimator variance. Particularly, this approach is not tailored for any specific task, such as model calibration improvement or OoD data detection. Accordingly, it can generalize better to any uncertainty improvement task, or be integrated with uncertainty improvement algorithms, like temperature scaling.

5.4 Method

We present our proposed ensemble knowledge distillation (EKD) method in this section. Since a good point estimator reduces model uncertainty, the objective of our approach is narrowed down to achieve a bias-reduced and efficient point estimator. This estimator is a single model θ with parameter probability density function $p(\theta|\mathscr{D}) = \delta(\theta - \hat{\theta})$. Therefore, $p(\theta|\mathscr{D}) = \delta(\theta - \hat{\theta})$ is our objective and the optimization is carried out with $p(\theta|\mathscr{D}) \rightarrow \delta(\theta - \hat{\theta})$.

The approach is specified to two stages including self-distillation ensemble training and ensemble distillation. It consists of three steps including: 1) training a stochastic ensemble teacher to reduce the bias of θ from $\hat{\theta}$; 2) self-distilling each ensemble member for model regularization; 3)

distilling this ensemble to reduce variance. We conduct stronger data augmentation to enhance the distillation.

5.4.1 Self-distillation Ensemble Training

We first train an ensemble to reduce the model estimator bias based upon the assumption of i.i.d. and consistency. The ensemble can be obtained in different manners. According to [5], randomly initialized ensemble members are independently trained with cross-entropy minimization as eq.5.3.

$$\theta = \arg\min_{\theta} \mathscr{L}_{CE}(P(y|\mathscr{D},\theta),y), \tag{5.3}$$

where θ is model parameter, $\mathscr{L}_{CE}$ is cross-entropy loss, P is model inference probability, y is the label encoded by one-hot vector, and $\mathscr{D}$ denotes input data.

Algorithm 3 Simple and Scalable Deep Ensemble (DE) [5]

1: Initialize θ_1, θ_2, . . . , θ_N randomly
2: **for** i = 1 : N **do**
3: Train a model θ_i with D *//Train a teacher ensemble in parallel*
4: **end for**
5: **Return** ensemble teacher $\Theta = \bigcup \theta_i$

Different from DE [5], we propose to use *self-distillation* to derive the ensemble. This extra distillation step can enhance model regularization [43], improve individual model calibration [13], and thus, help seek a better ensemble candidate for distillation. Moreover, self-distillation can be easily parallelized and ensure algorithm efficiency with high scalability.

Self-distillation is the knowledge distillation where the labeled data and architectures of teacher and student are identical. This scheme can effectively boost model generalization [11, 12] in virtue of its amplifying model regularization [43]. Some regularization techniques can help model

calibration, such as weight decay [13] and label smoothing [44]. Since knowledge distillation can be taken as an learned label smoothing regularization[37], we use self-distillation to regularize ensemble member for better calibration distillation.

Take the obtained model θ as a teacher model. Following self-distillation, we distill each teacher model θ to a student model ϕ with identical model architecture, the same labeled data, and the logits queried from teacher model θ. Given eq.5.3, self-distillation can be reformulated as:

$$\phi = \arg\min_{\phi} \alpha \mathscr{L}_{CE}(P(y|\mathscr{D},\phi),y) + \beta \mathscr{L}_{KD}(P(y|\mathscr{D},\phi),q_\theta), \tag{5.4}$$

where ϕ denotes self-distilled model parameters, $\mathscr{L}_{KD}$ indicates distillation loss, and q_θ is the soft targets, *i.e.* the probabilities queried from θ. We do not tune the two hyperparameters of α and β and set them to $\alpha = \beta = 0.5$ for simplification. Note that we select soft targets as query labels, so $\mathscr{L}_{KD}$ is cross-entropy loss between the soft targets and student model probability outputs.

Algorithm 4 Our Self-distillation Scalable Ensemble

1: Initialize $\theta_1, \theta_2, \ldots, \theta_N$ randomly
2: Initialize $\phi_1, \phi_2, \ldots, \phi_N$ randomly
3: **for** i = 1 : N **do**
4: Train a model θ_i with $\mathscr{D}$ *//Train a teacher*
5: Train self-distilled model ϕ_i from θ_i with $\mathscr{D}$ *//ensemble in parallel*
6: **end for**
7: **Return** ensemble teacher $\Phi = \bigcup \phi_i$

We collect N self-distilled student models in parallel and combine them as an ensemble model Φ. For inference, the ensemble prediction is the arithmetic average of the probability outputs across all student models as shown in eq.5.5.

$$P(y|x^*,\Phi) = \frac{1}{N}\sum_{i=1}^{N} P(y|x^*,\phi_i), \tag{5.5}$$

where Φ is the ensemble model, x^* is test data, and ϕ_i is individual self-distilled student model.

Compared to DE[5] in Algorithm 3, the only additional overhead from our Algorithm 4 is self-distillation process, but it could be reduced by early stopping for θ. Also, the overhead cost will be insignificant when ensemble model size increases. More importantly, we find model calibration gain is significant.

5.4.2 Ensemble Distillation

We conduct ensemble distillation to improve estimator efficiency by model estimator variance reduction. Ensemble model can reduce estimator bias, but it increases estimator variance from ensemble process. In particular, a biased estimator with smaller variance may be more useful than an unbiased estimator with large variance [92]. Accordingly, it is imperative to reduce estimator variance in ensemble model. We select knowledge distillation to solve this problem. This distillation can help a single model derive the unbiasness from ensemble model and reduce estimator variance. Therefore, the distilled single model is a good point estimator with better unbiasness and efficiency. The distillation is formulated in eq.5.6.

$$\kappa = \arg\min_{\kappa} \alpha \mathscr{L}_{CE}(P(y|\mathscr{D},\kappa),y) + \beta \mathscr{L}_{KD}(P(y|\mathscr{D},\kappa),q_{\Phi}) \tag{5.6}$$

where κ is the parameter of distilled model, q_{Φ} is the soft target, *i.e.*, the probability output from ensemble model Φ, and $\alpha : \beta = 1 : N$. Here we set the temperature to 1 since we would like the student model to better imitate the posterior probability distribution of ensemble model for better calibration.

We also take efficient distillation into consideration. We separate classification and distillation into two branches to accelerate the convergence of model training. The inference uses the combined

outputs from two branches with combination ratio $\alpha : \beta = 1 : N$.

Data Augmentation is imperative to our EKD. Since the ensemble model is significantly larger than single student network, it is difficult for the student to capture complex feature in the ensemble teacher. Recently, [36] reveals that ensemble model exhibits the recognition on data from multiple views and it is critical to capture these multi-view features over distillation. [36] points out that data augmentation, such as cropping, is a means to enforce networks learn multi-views. Thus, we adopt data augmentation as an effective approach to efficiently distill ensemble model. Considering storage efficiency, we employ online augmentation strategy and select AutoAugment [40] to enlarge training data space.

The effectiveness of data augmentation on boosting knowledge distillation has been empirically justified by lots of work, such as Active Mixup[35], DeiT[4], and ensemble distillation [15]. However, the enlargement of data space can not always boost performance, especially on model uncertainty reduction. Different data augmentation strategies can yield different distillation performance. To the best of our knowledge, the efficiency comparison between different augmentations on model uncertainty reduction has not be studied yet. We will discuss this point in Section 5.

Query Efficiency is critical to ensemble distillation. Conventionally, all ensemble members are queried over each batch for output average. This is time-consuming and limits the practical use. To improve query efficiency, we adopt switched training [103] to reduce query cost and keep distillation accuracy. This scheme of switched training randomly samples one teacher from the ensemble pool for query over each batch. Compared to the average of ensemble member outputs, it significantly reduces the query cost to $\frac{1}{N}$. It may introduce some variance over training, but distillation accuracy is still ensured[103].

With data augmentation and switched training, the optimization formulation for our EKD is finalized with eq.5.7. Moreover, intuitively, this training strategy can moderately augment data labels

through randomly querying ensemble member model over each batch. It can absorb some diversity in ensemble model since it acquires the output from each member instead of the average of the ensemble.

$$\kappa = \arg\min_{\kappa} \alpha \mathscr{L}_{CE}(P(y|\mathscr{D}_{\mathscr{A}}, \kappa), y) + \beta \mathscr{L}_{KD}(P(y|\mathscr{D}_{\mathscr{A}}, \kappa), q_{\phi_i}) \tag{5.7}$$

where $\mathscr{D}_{\mathscr{A}}$ is augmented data based upon $\mathscr{D}$, q_{ϕ_i} is the soft target queried from a randomly selected teacher ϕ_i in ensemble, and the other denotations follow eq.5.6.

5.4.3 Algorithm

We present our algorithm in this section. The distillation consists of two main stages, including ensemble training and ensemble distillation. To fulfill these two stages, there are three steps, including stochastic teacher training, self-distillation, and ensemble teacher distillation. The first step of stochastic teacher training aims at collecting a set of i.i.d. models to get a better unbiased model estimator. The second step of self-distillation intends to regularize each ensemble member, improve model calibration performance, and prepare better ensemble candidate for distillation. The third step of ensemble distillation focuses on deriving the unbiasness from ensemble model and reducing estimator variance for higher estimation efficiency.

We summarize our approach in Algorithm 5. First, we train an initial teacher model θ with labeled data $\mathscr{D}$. Secondly, we obtain an ensemble network Φ, *i.e.*, a set of DNNs ϕ by distilling the teacher model θ separately. Both are carried out in parallel to ensure efficient computation with good scalability, which is the first stage of ensemble training. Thirdly, given the obtained ensemble network Φ, ensemble knowledge distillation is conducted with efficient augmented training. The data augmentation policy Ω is an effective data augmentation, like AutoAugment[40], and model

Algorithm 5 Ensemble Knowledge Distillation (EKD)

INPUT: Training data $\mathscr{D}$, hyper-parameters (learning rate, batch size, etc.)
OUTPUT: Inference model κ
Stage 1: Self-distillation Ensemble Training

// Train a teacher ensemble model Φ *in parallel*
Initialize $\theta_1, \theta_2, \ldots, \theta_N$ randomly
Initialize $\phi_1, \phi_2, \ldots, \phi_N$ randomly
for i = 1 : N **do**
    Train a model θ_i with $\mathscr{D}$ *//Stochastic teacher training.*
    Obtain a self-distilled model ϕ_i from stochastic teacher model θ_i with $\mathscr{D}$
end for
Return ensemble teacher $\Phi = \bigcup \phi_i$

Stage 2: Ensemble Distillation

// Train a student model κ *by distilling ensemble teacher* $\Phi = \bigcup \phi_i$ *with* $\mathscr{D}_{\mathscr{A}}$
Augment data $\mathscr{D}_{\mathscr{A}} = \Omega(\mathscr{D}) + \mathscr{D}$ *//where* Ω *is augmentation policy.*
Initialize κ randomly
for epoch = 1, 2, ... **do**
    for s = 1, 2, ... **do**
        Sample $i \sim \mathscr{U}(1, N)$ *// $\mathscr{U}$ is a uniform distribution.*
        Sample $d^s \sim \mathscr{D}_{\mathscr{A}}$ *// Randomly sample a batch of data.*
        $\kappa = \kappa + \nabla\kappa(\alpha\mathscr{L}_{CE}^{d^s} + \beta\mathscr{L}_{KD}^{d^s}(q_{\phi_i}))$ *//where* $\alpha : \beta = 1 : N$
    end for
end for
Return κ for inference

query is reduced by switched training.

5.5 Experiments

We design various experiments to evaluate our approach including comparing with state-of-the-art(SoTA) methods. The comparative study focuses on InD model calibration and OoD data detection. Meanwhile, we assess the integration of our approach with different post-processing techniques for OoD data detection. Next, we study the effectiveness of different data augmentation on our ensemble distillation for model uncertainty reduction. We also conduct ablation study to show

the variability among different distillation schemes. Last but not least, we extend our experiments to ViT[104] variants.

5.5.1 Experiment Setting

InD Model Calibration. We select four popular image classification benchmark datasts, including CIFAR-10 [72], CIFAR-100[72], Tiny-ImageNet[105], and ImageNet[106]. We compare our approach with the four most relevant SoTA approaches, including DE[5], ensemble distribution distillation (EnD^2)[14], batch ensemble (BE)[15], and temperature scaling (TS)[13]. We select four metrics for performance evaluation: 1) classification error (Error); 2) negative log likelihood (NLL); 3)Brier score (Brier); 4) expected calibration error (ECE).

OoD Data Detection. We select six benchmark datasets as OoD datasets for CIFAR-10 and CIFAR-100, including Textures [107], SVHN[108], Places365[66], LSUN-Crop[109], LSUN-Resize[109], and iSUN[109]. For ImageNet, we select ImageNet-O[110] and ImageNet-A[110] as OoD data. We compare our approach with DE[5], EnD^2 [14], and PD-EnD^2[3]. Note that the compared models are the same as the ones from InD ones. We also select three OoD detection strategies, including maximum over softmax probability (MSP) [95], energy score [97], and ODIN [96], to identify OoD data. We select three evaluation metrics: (1) the false positive rate of OoD data when true positive rate of InD data is is 95% (FPR95); (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-recall curve (AUPR).

5.5.2 InD Model Calibration

Table 5.1 shows that our method significantly outperforms the other four SoTA approaches on model calibration metrics. Since stronger data augmentation is incorporated into knowledge dis-

tillation, it is not surprising that the error of EKD is lower than other approaches. NLL and Brier of EKD are also lower.

Table 5.1: Comparison of SoTA on InD calibration with five runs. PD denotes PD-EnD^2[3]. Best performance is bold, * results are reported in the original paper, - denotes no result reported from the paper, and ↓ denotes the less the better.

Methods	Error↓	NLL↓	Brier↓	ECE↓	Error↓	NLL↓	Brier↓	ECE↓
	VGG16 on CIFAR-10				ResNet34 on CIFAR-10			
EKD	**5.7**$_{\pm 0.1}$	**0.17**$_{\pm 0.01}$	**0.08**$_{\pm 0.00}$	**0.7**$_{\pm 0.1}$	**5.6**$_{\pm 0.1}$	**0.17**$_{\pm 0.01}$	**0.08**$_{\pm 0.00}$	**0.5**$_{\pm 0.1}$
DE [5]	6.3$_{\pm \text{NA}}$	0.18$_{\pm \text{NA}}$	0.09$_{\pm \text{NA}}$	1.3$_{\pm \text{NA}}$	5.8 $_{\pm \text{NA}}$	0.18$_{\pm \text{NA}}$	0.09$_{\pm \text{NA}}$	0.9$_{\pm \text{NA}}$
EnD^2[14]	7.3$_{\pm 0.2}$*	0.25$_{\pm 0.01}$*	-	1.0$_{\pm 0.2}$*	7.9$_{\pm \text{NA}}$*	0.26$_{\pm \text{NA}}$*	0.12$_{\pm \text{NA}}$*	1.7$_{\pm \text{NA}}$*
BE [15]	-	-	-	-	6.0$_{\pm 0.2}$*	0.18$_{\pm 0.01}$*	0.09$_{\pm 0.0}$*	0.7$_{\pm 0.1}$*
TS [13]	8.3$_{\pm 0.4}$	0.25$_{\pm 0.02}$	0.12$_{\pm 0.01}$	0.9$_{\pm 0.1}$	6.4$_{\pm 0.3}$	0.20$_{\pm 0.01}$	0.10$_{\pm 0.01}$	0.8$_{\pm 0.1}$
	VGG16 on CIFAR-100				WideResNet28-10 on CIFAR-100			
EKD	**23.8**$_{\pm 0.1}$	**0.84**$_{\pm 0.01}$	**0.32**$_{\pm 0.01}$	**1.1**$_{\pm 0.1}$	**16.6**$_{\pm 0.2}$	**0.59**$_{\pm 0.01}$	**0.23**$_{\pm 0.01}$	**1.5**$_{\pm 0.1}$
DE [5]	25.0$_{\pm \text{NA}}$	0.89$_{\pm \text{NA}}$	0.34$_{\pm \text{NA}}$	2.1$_{\pm \text{NA}}$	17.7$_{\pm \text{NA}}$	0.68$_{\pm \text{NA}}$	0.25$_{\pm \text{NA}}$	2.2$_{\pm \text{NA}}$
EnD^2[14]	27.9$_{\pm 0.3}$*	1.14$_{\pm 0.01}$*	-	4.9$_{\pm 0.5}$*	-	-	-	-
BE [15]	-	-	-	-	18.1$_{\pm 0.3}$*	0.67$_{\pm 0.01}$*	0.26$_{\pm 0.01}$*	2.4$_{\pm 0.0}$*
TS [13]	31.2$_{\pm 0.5}$	1.15$_{\pm 0.03}$	0.43$_{\pm 0.01}$	2.0$_{\pm 0.2}$	20.2$_{\pm 0.5}$	0.79$_{\pm 0.01}$	0.29$_{\pm 0.01}$	3.6$_{\pm 0.1}$
	VGG16 on Tiny ImageNet				WideResNet28-5 on Tiny ImageNet			
EKD	**34.2**$_{\pm 0.1}$	**1.40**$_{\pm 0.01}$	**0.47**$_{\pm 0.01}$	**1.2**$_{\pm 0.3}$	**29.1**$_{\pm 0.2}$	**1.18**$_{\pm 0.04}$	**0.40**$_{\pm 0.02}$	**2.3**$_{\pm 0.2}$
DE [5]	36.6$_{\pm \text{NA}}$	1.51$_{\pm \text{NA}}$	0.52$_{\pm \text{NA}}$	3.8$_{\pm \text{NA}}$	30.1$_{\pm \text{NA}}$	1.24$_{\pm \text{NA}}$	0.43$_{\pm \text{NA}}$	2.9$_{\pm \text{NA}}$
EnD^2[14]	37.6$_{\pm 0.2}$*	1.83$_{\pm 0.02}$*	-	7.2$_{\pm 0.4}$*	-	-	-	-
BE[15]	-	-	-	-	34.0$_{\pm \text{NA}}$*	1.44$_{\pm \text{NA}}$*	0.46$_{\pm \text{NA}}$*	5.9$_{\pm \text{NA}}$*
TS [13]	40.5$_{\pm 0.4}$	1.68 $_{\pm 0.02}$	0.52$_{\pm 0.02}$	3.1$_{\pm 0.1}$	39.3$_{\pm 0.4}$	1.58$_{\pm 0.02}$	0.51$_{\pm 0.02}$	2.1$_{\pm 0.1}$
	ResNet-50 on ImageNet				ResNet-18 on ImageNet			
EKD	**20.5**$_{\pm 0.2}$	**0.82**$_{\pm 0.01}$	**0.29**$_{\pm 0.01}$	**1.1**$_{\pm 0.1}$	**27.0**$_{\pm 0.2}$	**1.06**$_{\pm 0.02}$	**0.37**$_{\pm 0.1}$	**1.7**$_{\pm 0.1}$
DE[5]	21.0$_{\pm \text{NA}}$	0.83$_{\pm \text{NA}}$	0.30$_{\pm \text{NA}}$	2.3$_{\pm \text{NA}}$	27.3$_{\pm \text{NA}}$	1.07$_{\pm \text{NA}}$	0.38$_{\pm \text{NA}}$	2.1$_{\pm \text{NA}}$
PD[3]	23.0$_{\pm 0.1}$*	-	-	1.6$_{\pm 0.1}$*	-	-	-	-
TS [13]	24.5$_{\pm 0.3}$	0.92$_{\pm 0.2}$	0.33$_{\pm 0.2}$	1.5$_{\pm 0.2}$	30.2$_{\pm 0.4}$	1.24$_{\pm 0.03}$	0.41$_{\pm 0.02}$	1.8$_{\pm 0.2}$

However, interestingly, EKD can consistently outperform on ECE. Compared to DE, EKD reduces ECE by half in average. This improvement indicates that ensemble distillation not only imitates the posterior distribution pattern of the ensemble, but also can outperform ensemble when stronger augmentation is carried out. Compared to EnD^2, the performance gain is more significant in larger datasets, like Tiny-Imagent. This implies that simple data augmentation can depict posterior distribution of ensemble model without explicit complex formulation, such as Dirichlet distribution.

The implicit depiction by augmented data query may be more accurate since ensemble may follow more complex multi-modal distribution. Compared to BE, the improvement from EKD indicates that effective data augmentation can outperform adversarial perturbation on ensemble distillation efficiency. Compared to TS, calibration improvement from EKD implies that ensemble plays a prominent role in better calibration.

5.5.3 OoD Data Detection

Next, we evaluate our EKD on OoD data detection. Note that the evaluated models are exactly the same as the models for InD model calibration assessment in Table 5.1 for fair comparison. We compare our approach with DE[5], EnD^2 [14], and PD-EnD^2[3]. The comparison in Table 5.2 shows that our EKD significantly improves AUROC and outperforms DE and EnD^2. Interestingly, the OoD detection of our approach is more independent from model classification error. This implies that the posterior distributions between OoD and InD data are more separated and the confounding is reduced.

For OoD detection, we also extensively evaluate our approach with the integration of the three different OoD detection techniques, including MSP [95], energy score (Energy) [97], and ODIN [96]. The results in Table 5.3 show that our EKD consistently outperforms DE across different benchmarks except the case of ResNet-50 on ImageNet with ODIN. This performance gain further justifies that our EKD better separates OoD from InD data than DE due to model variance reduction. Also, the OoD detection is more improved when EKD is integrated with Energy since both boost OoD and InD distribution separation [97]. For ImageNet, ImageNet-A and ImageNet-O are natural adversarial samples, which are more challenging for OoD detection, and the improvement by distillation is less. For ODIN, it is optimized with image pre-processing with adversarial samples to identify OoD data [96]. When OoD data are adversarial samples, like ImageNet-A

or ImageNet-O, this optimization may be limited and cause significant performance degradation, such as the case of ResNet-50 with ImageNet.

Table 5.2: Comparison of the OoD detection with AUROC (the higher the better) across SoTA methods. The evaluated models are the same as the one from Table 5.1. The best performance is bold and * results are reported in the original paper.

InD	CIFAR-10		CIFAR-100		Tiny-ImageNet		ImageNet	
OoD	LSUN	TIM	LSUN	TIM	LSUN	C100	ImageNet-O	ImageNet-A
EnD^2[14]	92.2*	88.8*	83.8*	77.2*	70.4*	76.7*	48.80*	87.20*
PD[3]	-	-	-	-	-	-	53.20*	86.80*
DE[5]+[95]	92.3	85.4	72.9	78.5	84.8	81.8	54.04	88.36
DE[5]+[97]	93.1	85.9	73.7	79.0	86.6	83.2	54.68	89.24
DE[5]+[96]	93.1	86.2	78.0	80.1	91.5	87.5	54.85	84.38
EKD+[95]	93.2	88.8	66.5	79.3	89.5	86.8	54.84	87.25
EKD+[97]	96.7	**90.3**	85.1	**81.8**	91.4	88.4	**59.80**	**89.47**
EKD+[96]	**97.1**	89.7	**86.2**	79.3	**94.1**	**92.2**	58.81	74.56

Table 5.3: Comparison among a single model (Single), DE with five members, and our EKD with five teachers. The mean performance is reported and the best is bold.

	Method	MSP			Energy			ODIN (T=1e3)		
		F95↓	AUROC↑	AUP↑	F95↓	AUROC↑	AUP↑	F95↓	AUROC↑	AUP↑
VGG16	Single	64.17	88.94	97.69	40.17	93.13	98.54	34.96	94.01	98.72
CIFAR	DE[5]	55.37	90.03	97.61	48.40	90.63	97.70	25.26	94.76	98.39
10	EKD	**49.74**	**92.70**	**98.51**	**23.59**	**95.78**	**99.10**	**19.82**	**96.43**	**99.23**
ResNet34	Single	67.18	88.08	97.50	66.69	88.17	97.51	41.23	91.09	97.96
CFIAR	DE[5]	58.05	89.01	97.53	52.72	91.00	98.15	32.42	93.78	98.63
10	EKD	**57.71**	**89.29**	**97.72**	**33.01**	**94.09**	**98.73**	**29.25**	**94.56**	**98.80**
VGG16	Single	84.76	72.53	93.09	83.59	72.84	93.10	75.23	78.10	94.49
CIFAR	DE[5]	87.45	71.74	92.94	84.85	74.83	93.81	78.42	79.82	95.18
100	EKD	**80.30**	**74.73**	**93.78**	**63.07**	**85.77**	**96.74**	**61.43**	**86.19**	**96.83**
WRN28-10	Single	79.69	77.71	94.47	78.29	78.01	94.48	59.76	83.64	95.75
CIFAR	DE[5]	76.18	80.73	95.43	74.73	81.18	95.51	54.93	86.43	**96.67**
100	EKD	**71.61**	**82.01**	**95.70**	**65.54**	**84.76**	**96.28**	**52.07**	**86.89**	96.55
ResNet50	Single	73.91	71.16	96.32	66.40	80.20	97.53	70.86	77.43	96.91
IMAGE	DE[5]	67.59	81.13	97.13	65.59	81.96	97.45	**70.85**	**78.16**	**96.92**
-NET	EKD	**66.33**	**81.71**	**97.78**	**64.09**	**83.22**	**97.59**	77.21	71.24	96.83

5.5.4 *Data Augmentation*

We discuss the effectiveness of data augmentation in this section. Different augmentation policies Ωs can yield different distillation efficiencies [111]. We evaluate four augmentation techniques to assess which is more effective to our method. The evaluated techniques include AutoAugment[40], Mixup[1], SimClr[98], and MoCo[99], which are popular in supervised and self-supervised learning. In our study, AutoAugment employs the best policy for each benchmark dataset; Mixup adopts the mix ratio of 0.2 suggested from [100]; SimClr and MoCo (MoCo-V2) pretrained models are used as augmented features for distillation fine-tuning.

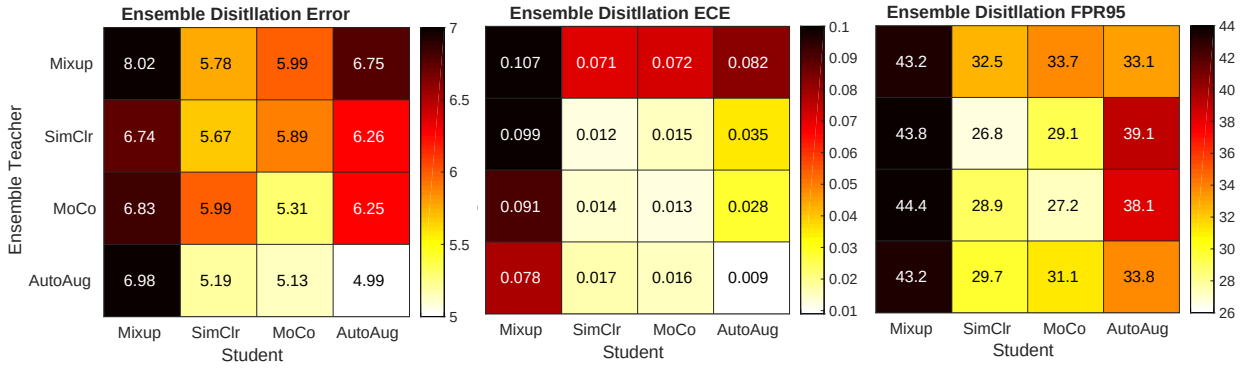


Figure 5.1: Data augmentation interaction between teacher and student models on CIFAR-10 with the size of 32×32. Ensemble distillation error is classification error based upon percentage. The results are reported over the average of three runs of distillations.

We first conduct the experiments on CIFAR-10 across different teachers and students. The interaction in Figure 5.1 reveals that Mixup is not good on either teacher or student due to its largest Error, ECE, and FPR95. This significant degradation mainly results from the linear interpolation on the label. The mix pseudo labels harm model prediction confidence, which affects distillation fidelity to teacher model, especially on uncertainty calibration. To justify this, we conduct hard label based Mixup and the obtained ECE is improved, which is shown in Table 5.4. Figure 5.1 also shows that the augmentation fidelity can help improve calibration. When the teacher and student

Table 5.4: Comparison of data augmentation on InD model calibration and OoD detection. CIFAR-10 images are resized to 224 $\times$ 224. Mixup(hard) denotes mixup with hard label[4]. The best performance is highlighted in bold. Three runs are averaged.

	Methods	Error↓	NLL↓	Brier↓	ECE↓	FPR95↓	AUROC↑	AUPR↑
ResNet50 CIFAR-10	AutoAug	$4.0_{\pm 0.1}$	$0.11_{\pm 0.01}$	$\mathbf{0.05}_{\pm 0.01}$	$\mathbf{0.3}_{\pm 0.1}$	**11.07**	**98.00**	**99.58**
	Mixup	$4.5_{\pm 0.1}$	$0.14_{\pm 0.01}$	$0.06_{\pm 0.01}$	$1.4_{\pm 0.5}$	13.24	97.45	99.38
	Mixup(hard)	$4.4_{\pm 0.1}$	$0.13_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.9_{\pm 0.1}$	13.67	97.42	99.44
	SimClr	$\mathbf{3.4}_{\pm 0.1}$	$\mathbf{0.10}_{\pm 0.01}$	$\mathbf{0.05}_{\pm 0.01}$	$\mathbf{0.3}_{\pm 0.1}$	12.31	97.91	99.56
	MOCO-V2	$\mathbf{3.4}_{\pm 0.1}$	$\mathbf{0.10}_{\pm 0.01}$	$\mathbf{0.05}_{\pm 0.01}$	$0.6_{\pm 0.1}$	12.27	97.68	99.50
ResNet50 ImageNet	AutoAug	$22.3_{\pm 0.2}$	$\mathbf{0.84}_{\pm 0.01}$	$0.30_{\pm 0.02}$	$\mathbf{1.2}_{\pm 0.1}$	65.78	82.01	97.49
	Mixup	$22.8_{\pm 0.0}$	$0.91_{\pm 0.00}$	$0.32_{\pm 0.00}$	$2.1_{\pm 0.3}$	66.13	81.53	97.48
	Mixup(hard)	$22.9_{\pm 0.1}$	$0.91_{\pm 0.01}$	$0.33_{\pm 0.01}$	$1.5_{\pm 0.2}$	66.32	81.99	97.51
	SimClr	$22.9_{\pm 0.0}$	$0.88_{\pm 0.00}$	$0.33_{\pm 0.00}$	$2.1_{\pm 0.0}$	65.66	82.17	97.50
	MOCO-V2	$\mathbf{22.1}_{\pm 0.1}$	$\mathbf{0.84}_{\pm 0.01}$	$\mathbf{0.29}_{\pm 0.02}$	$1.6_{\pm 0.1}$	**65.21**	**82.57**	**97.53**

are both from AutoAugment, Error and ECE both outperform. For AutoAugment outperforming SimClr and MoCo, it is possibly due to that AutoAugment more focuses on the current distillation task while the representation from SimClr or MoCo is more general. Therefore, SimClr and MoCo are both moderately good on teachers and students in Figure 5.1. Especially, they both yield better OoD detection with lower FPR95. This performance benefits from contrastive learning feature, which helps identify OoD data. The features are further enhanced through distillation.

We next extend the experiments to larger scale data. The evaluated benchmarks are 224 $\times$ 224 CIFAR-10 and ImageNet. The ensemble teacher is set to the vanilla model trained with only random cropping and horizontal flipping. For SimClr and MoCo, we employ ImageNet pretrained feature for both CIFAR-10 and ImageNet assessment. From Table 5.4, AutoAugment still shows the best performance on InD model calibration for both CIFAR-10 and ImageNet. SimClr and MoCo both show lower Error, NLL, and Brier due to stronger augmentation features from contrastive learning. AutoAugment still outperforms on ECE. Also, the ECE difference between SimClr or MoCo and AutoAugment is less. It implies the augmentation feature from extra data like ImageNet is helpful for InD model calibration. Interestingly, compared to AutoAugment, SimClr and MoCo exhibit better OoD detection when InD data is ImageNet, but worse for CIFAR-10.

The possible reason is that the pretrained features from ImageNet may contain OoD features and interfere OoD recognition confidence for CIFAR-10 model. Thus, pretrained features from InD data may be a good option for OoD detection improvement.

Table 5.5: Comparison among different distillation strategies. Stochastic Ensemble (ST) [5] and our Self-distillation Ensemble (SD) are compared. Average of outputs (Avg) and switched training (Switch) are compared. The query cost is assessed with one run of feed forward network. The best performance is bold and three runs are averaged.

	Method	Query Cost↓	InD Calibration				OoD Detection		
			Error↓	NLL↓	Brier↓	ECE ↓	F95↓	AUROC↑	AUP↑
ResNet	ST + Avg	$\mathcal{O}(N)$	**4.9**$_{\pm 0.1}$	**0.14**$_{\pm 0.01}$	**0.07**$_{\pm 0.01}$	1.0$_{\pm 0.2}$	**14.22**	**97.25**	**99.39**
18	ST + Switch	$\mathcal{O}(1)$	5.1$_{\pm 0.1}$	0.15$_{\pm 0.01}$	0.08$_{\pm 0.01}$	1.2$_{\pm 0.2}$	14.28	97.23	**99.39**
CIFAR	SD + Avg	$\mathcal{O}(N)$	5.0$_{\pm 0.1}$	0.15$_{\pm 0.01}$	0.08$_{\pm 0.01}$	**0.3**$_{\pm 0.0}$	16.34	96.71	99.22
10	SD + Switch	$\mathcal{O}(1)$	5.2$_{\pm 0.1}$	0.15$_{\pm 0.01}$	0.08$_{\pm 0.01}$	0.4$_{\pm 0.0}$	16.30	96.73	99.21
ResNet	ST + Avg	$\mathcal{O}(N)$	21.9$_{\pm 0.1}$	0.78$_{\pm 0.01}$	0.31$_{\pm 0.01}$	3.6$_{\pm 0.1}$	57.14	87.41	97.08
18	ST + Switch	$\mathcal{O}(1)$	22.2$_{\pm 0.1}$	0.79$_{\pm 0.01}$	0.31$_{\pm 0.01}$	3.8$_{\pm 0.1}$	**56.46**	**87.44**	**97.10**
CIFAR	SD + Avg	$\mathcal{O}(N)$	**21.7**$_{\pm 0.1}$	**0.76**$_{\pm 0.01}$	**0.30**$_{\pm 0.01}$	**1.4**$_{\pm 0.1}$	61.44	86.30	96.83
100	SD + Switch	$\mathcal{O}(1)$	21.8$_{\pm 0.1}$	0.77$_{\pm 0.01}$	**0.30**$_{\pm 0.01}$	1.5$_{\pm 0.1}$	59.41	86.67	96.92
ResNet	ST + Avg	$\mathcal{O}(N)$	**22.3**$_{\pm 0.1}$	**0.85**$_{\pm 0.01}$	**0.30**$_{\pm 0.01}$	1.5$_{\pm 0.1}$	65.50	**82.99**	**97.54**
50	ST + Switch	$\mathcal{O}(1)$	22.6$_{\pm 0.1}$	0.88$_{\pm 0.01}$	0.31$_{\pm 0.01}$	1.7$_{\pm 0.2}$	**65.30**	82.69	97.53
IMAGE	SD + Avg	$\mathcal{O}(N)$	22.4$_{\pm 0.1}$	0.87$_{\pm 0.01}$	0.31$_{\pm 0.01}$	**1.2**$_{\pm 0.1}$	65.75	82.40	97.45
-NET	SD + Switch	$\mathcal{O}(1)$	**22.3**$_{\pm 0.1}$	0.86$_{\pm 0.01}$	**0.30**$_{\pm 0.01}$	**1.2**$_{\pm 0.1}$	65.78	82.01	97.49

5.5.5 Ablation Study of Ensemble Distillation Strategies

We evaluate different distillation strategies for our EKD. Since EKD is optimized with self-distillation for ensemble and switched training for query, we study how they affect distillation. We compare self-distillation with the stochastic scheme [5], and compare switched training [103] with ensemble average.

Table 5.5 shows that self-distillation can consistently yield significantly lower ECE across all cases. This indicates that self-distillation optimization is critical to model calibration improvement of EKD. Switched training yields comparable or slightly worse performance than average query, but the query cost is significantly reduced. This indicates that switched training can replace ensemble

average for higher query efficiency.

Table 5.6: Comparison on CIFAR-10 and ImageNet distillation results in ViT variant models. The performance is averaged over three runs.

	Method	InD Calibration				OoD Detection		
		Error↓	NLL↓	Brier↓	ECE ↓	F95↓	AUROC↑	AUP↑
CIFAR 10	DeiT[4]	**2.5**$_{\pm 0.1}$	**0.08**$_{\pm 0.01}$	**0.04**$_{\pm 0.00}$	**1.4**$_{\pm 0.1}$	13.62	97.38	99.42
	T2T-DeiT [113]	2.5$_{\pm 0.1}$	**0.08**$_{\pm 0.01}$	**0.04**$_{\pm 0.00}$	1.9$_{\pm 0.1}$	**5.30**	**98.85**	**99.76**
	DE(ResNet18)	3.7$_{\pm NA}$	0.18$_{\pm NA}$	0.06$_{\pm NA}$	2.2$_{\pm NA}$	29.26	95.85	99.16
IMAGE -NET	DeiT[4]	**25.7**$_{\pm 0.1}$	**1.06**$_{\pm 0.00}$	**0.36**$_{\pm 0.01}$	**0.7**$_{\pm 0.1}$	**73.42**	**83.42**	**97.91**
	T2T-DeiT[113]	25.8$_{\pm 0.1}$	1.06$_{\pm 0.00}$	0.37$_{\pm 0.00}$	0.8$_{\pm 0.1}$	74.27	79.36	96.60
	DE(ResNet18)	26.6$_{\pm NA}$	1.07$_{\pm NA}$	0.37$_{\pm NA}$	1.1$_{\pm NA}$	79.03	72.91	96.44

5.5.6 Architecture Extension to ViT

Last, but not least, we further extend our experiment to ViT [104] for performance evaluation. Since ViT variant models demand stronger data augmentation for training, we combine multiple augmentation techniques for ensemble distillation. These techniques include AutoAugment, Mixup, and CutMix[112]. To fit our EKD framework, we select DeiT[4] (DeiT-Tiny) for ViT study. In addition, we include another SoTA ViT variant of T2T[113] (T2T-10) for distillation assessment. We modify T2T-ViT to T2T-DeiT with an extra distillation token for our ensemble distillation. The ensemble teacher consists of five members of ResNet-18. Table 5.6 shows that the improvement still holds for ViT models. Concerning InD model calibration, it shows significant reduction in ECE. Compared to convolution neural networks, ViT variants yields more significant improvement in OoD detection. The possible reason is that data augmentation is stronger, which helps recognize the difference between InD and OoD data.

5.6 Summary

We propose a simple yet effective approach of EKD to reduce model uncertainty by leveraging point estimation. For a better unbiased and more efficient point estimator, we blend ensemble model with knowledge distillation to reduce estimator bias and variance. Stronger data augmentation and switched training are incorporated to enhance distillation efficiency. We conduct extensive experiments and justify that our approach significantly outperforms SoTA methods like DE on InD model calibration and OoD data detection. We also discuss the data augmentation efficiency for EKD and show that AutoAugment yields better InD model calibration while SimClr or MoCo shows better OoD detection when pretrained features come from InD data. We also conduct ablation study to examine EKD, extend EKD to ViT variants, and justify its efficacy.

CHAPTER 6: EXTENSION OF KNOWLEDGE DISTILLATION TO SEMANTIC SEGMENTATION

6.1 Problem Introduction

Deep neural networks (DNNs) have become the "go-to" models in various computer vision tasks, such as image classification, object detection, semantic segmentation. However, recent work found that DNNs are often overconfident when they make mistakes [13], misleading downstream applications. To calibrate DNNs' confidence in prediction, researchers have developed a rich line of works on image classification using regularized training [114, 115, 100, 44], post-hoc processing [13, 116, 117, 118], and Bayesian modeling [5, 119, 120, 45], to name a few.

However, the extensive pursuit of image classification has left it unclear how to calibrate DNNs for other computer vision tasks and how well existing calibration methods generalize to the tasks beyond image classification. In this paper, we conduct a comprehensive study of the calibration of deep semantic segmentation models.

Semantic segmentation tags a semantic label to every pixel in an image. Over the past years, we have witnessed increasingly accurate DNN models [121, 122, 123, 124, 125, 126] for semantic segmentation over various benchmark datasets [127, 128, 129, 130, 131, 132]. The progress benefits many downstream applications, such as medical imaging and diagnostics, autonomous driving, and robotics. While accuracy is essential for the applications, the segmentation models' uncertain-

This chapter contains previously published materials from "ADCNN: Towards learning adaptive dilation for convolutional neural networks", by Jie Yao, Dongdong Wang, Hao Hu, Weiwei Xing, and Liqiang Wang, published in Pattern Recognition, 123, 108369

ties also provide crucial signals, especially for safety-critical applications — by a segmentation model's *uncertainty*, we refer to its confidence in the label it assigns to a pixel. For example, an autonomous driving system can use the uncertainty of a semantic segmentation model (e.g., about drivable areas) to make informed decisions.

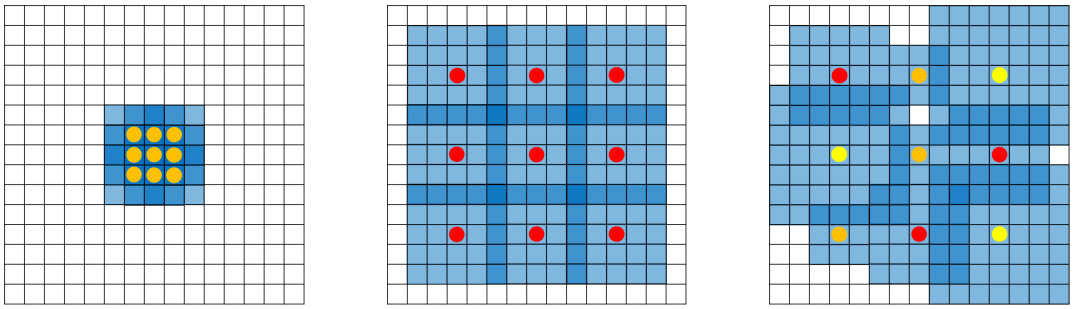
To better understand how to calibrate semantic segmentation models, a thorough study on how semantic segmentation modeling works is imperative. This chapter starts from understanding semantic segmentation modeling from convolution neural network perspective. Then, an improvement algorithm is proposed to increase segmentation prediction accuracy. Given the thorough analysis and study, ensemble knowledge distillation is further extended to semantic segmentation calibration. The investigated models also extend from convolution neural networks to Vision Transformers. Extensive experiments are carried out to justify ensemble knowledge distillation and compare it with other existing calibration methods.

6.2 Semantic Segmentation: Investigation and An Improvement Method

Convolutional kernels are the critical components for Convolutional Neural Networks (CNNs), which have been dominant approaches for majority of computer vision tasks in recent years [133, 134, 135]. Their power relies on the ability of hierarchically representing spatial features over input regions called Receptive Fields (RFs), by stacking a number of convolutional layers into deep structures [136]. Nowadays, among common practices for designing CNN architectures, which usually prefer large RFs in order to achieve superior performances, Dilated Convolutional Kernels (DCKs) serve as a popular choice not only because of their simplicity, but also the effectiveness [137, 138]. Unlike their conventional equivalents, DCKs are able to exponentially enlarge RFs without increasing kernel sizes. CNN models with dilated kernels also report the impressive results on fundamental tasks such as object recognition [139] and semantic segmentation [138]. Moreover,

DCKs perform well in some more specific tasks such as object detection with multi-model [140] and monocular depth estimation [141], demonstrating significant performance gain by employing dilated convolutional kernels.

To further improve the dilated kernels, two obvious problems that universally reside in most of existing dilated CNN structures need to be properly tackled: fixed RF size and manually selected dilation range. First, the dilation value for a convolutional layer is shared across all pixels, which means that every output location has the same size of RF. However, this could be very counter-intuitive: sizes of Region of Interest (ROIs) usually vary dramatically over different positions, and thus, the sizes of RFs are also expected to be adjusted accordingly to encode diverse spatial information. Therefore it is reasonable to believe that a fixed RF across every position is hard to capture such intra and inter sample diversities especially for large-scale, high-resolution image datasets.



(a) No dilation (b) Regular dilation (c) Pixel-wise adaptive dilation

Figure 6.1: Comparison of regular and pixel-wise adaptive dilation. Different colors stand for different dilation.

Second, the mainstream approaches of selecting a dilation value is mainly feature-independent; for each dilated convolution layer, we need to specify dilation values arbitrarily before it can be integrated into the base structure. This usually requires a strong domain knowledge about input and output contexts for hand-crafting; and for many specific tasks, there is no clear guidance available for selecting proper dilation values in practice. In recent years, deformable convolutional neural

networks [142, 143] have been proposed to enhance the transformation modeling capability of CNNs by augmenting the spatial sampling locations in the modules with additional offsets and learning the offsets from the target tasks. However, they set a small value such as 1 for offset as the upper bound, which means that it usually needs to stack deformable convolutional layers to enlarge the RFs and get a better performance. On the other aspect, if we choose a bigger value as the upper bound of the offset, it will degenerate the deformable convolutional layer into an attention mechanism due to some incorrect focus on minute details, which makes learning a proper offset need either a well-prepared dataset or an adequate training process.

In this section, we answer the above challenges by combining the dilation selection with conventional CNN modules and incorporating them into a unified data-driven framework. We propose Adaptive Dilation Convolutional Neural Networks (ADCNN), a simple yet powerful extension for general DCKs, which treats dilation values as learnable weights and can be jointly optimized with other CNN weights in an end-to-end fashion. As shown in Fig. 6.1, in the newly formulated ADCNN kernels, dilation is learned to change at different input positions to reflect input spatial diversity, resulting in dynamic RFs with irregular shapes in a single layer. In practice, there are two major difficulties to overcome.

How to decide the dilation value online. We handle this by regarding the dilation as a function of input at individual pixels. More specifically, the function samples dilation values through certain probability distributions that are conditioned by pixel-wise input features. To solve non-differentiable nature of general sampling process, we approximate it by Gumbel-Softmax [144] as a differentiable estimation to keep ADCNN end-to-end trainable.

What are proper dilation values for inputs. Since there is no clear explanation on how network layers work, we believe that it still remains an open question and can only be answered with valid hypotheses. For ADCNN kernels, we make the assumption that dilation values are related to inter-

layer patterns between convolution layers due to their hierarchical nature. In such cases, RF size at each location is adjusted based on information flows between corresponding inter-layer pixels during forward propagation.

Following the strategies described above, ADCNN-kernels evolve into light-weighted modules that can be easily plugged into various CNN architectures. Moreover, sampling dilation space through inter-layer pattern modeling also demonstrate that adaptive networks can be achieved in a simpler manner without engaging high dimensional spaces. We evaluate the proposed ADCNNs via several fundamental tasks including large-scale, fine-grained visual classification, semantic segmentation and optical flow estimation. Moreover, several ablation studies are performed to examine various properties of ADCNNs. Our experimental results indicate in most cases ADCNNs are able to consistently yield better performances across various popular backbone architectures with trivial cost.

6.2.1 Understanding of Semantic Segmentation from CNN

Content-Adaptive Networks. This research direction is focused on building dynamic internal structures via data-driven approaches to better leverage larger spatial variations from inputs. A set of related techniques tend to develop differentiable approximations for traditional image-adaptive filters and integrate them as end-to-end trainable layers for CNN models. For example, Jampani et al. [145] include bilateral filters [146, 147] in CNN models as a layer to generalize the parameterization and derive a gradient descent algorithm so the filter parameters can be learned from data; Wang et al. [148] and Wu et al. [149] introduce their trainable version of non-local means filters [150] and guided filters [151], respectively. These approaches conduct content-adaptive enhancements in separate layers without interacting with convolution kernels. Another set of techniques propose the idea of directly generating kernel weights based on layer inputs and extend it with

attention mechanism as well as other task-specific improvements. For example, Xue et al. [152] proposed Cross Convolutional Network which encodes image and motion information as feature maps and convolutional kernels to aid in synthesizing future frames; Jia et al. [153] proposed the Dynamic Filter Network, where filters are generated dynamically conditioned on an input in a sample-specific way; Su et al. [154] proposed a pixel-adaptive convolution (PAC) operation in which the filter weights are multiplied with a spatially varying kernel that depends on learnable, local pixel features; Wu et al. [155] proposed a dynamic filtering strategy with large sampling field for ConvNets (LS-DFN) to learn dynamic position-specific kernels and takes advantage of very large receptive fields and local gradients. In recent years, a new kind of dynamic convolutional network, which is Deformable Convolutional Networks [142], has been proposed to enhance the transformation modeling capability of CNNs. Based on the idea of augmenting the spatial sampling locations in the modules with additional offsets, Deformable Convolutional kernels learn such offsets from the target tasks, without additional supervision. Following the similar direction, Zhu et al. [143] proposed a reformulation of Deformable ConvNets that improves its ability to focus on pertinent image regions.However, most of them rely on additional modules with large kernel sizes, being incapable of scaling up to more general structures.

Dynamic Receptive Fields. Comparing to the above approaches to build content-adaptive networks, there is much less work aiming at enabling the content-aware ability via adjusting receptive fields (RFs). Majority of RF-related researches focus on how to effectively enlarge RFs in order to achieve better performance. Among them, dilated convolution kernels [137], which support exponential expansion of the receptive field without loss of resolution or coverage, become a popular choice as it can exponentially increase RF sizes while maintaining small kernel sizes. However, this could also lead to negative impacts, such as sparsity and “gridding" effect [138]. Unlike static RFs produced by dilation, recent works such as Dai et al. [142] and Zhu et al. [143] argue that RFs should be more diverse in order to capture rich spatial variations. They propose deformable

CNNs that learn to adjust the positions for convolving, resulting in free-form RFs that are totally data-dependent. Besides, Shelhamer et al. [156] attempt to create diverse yet controllable RFs by composing the structured Gaussian kernels and unstructured ordinary convolution kernels.

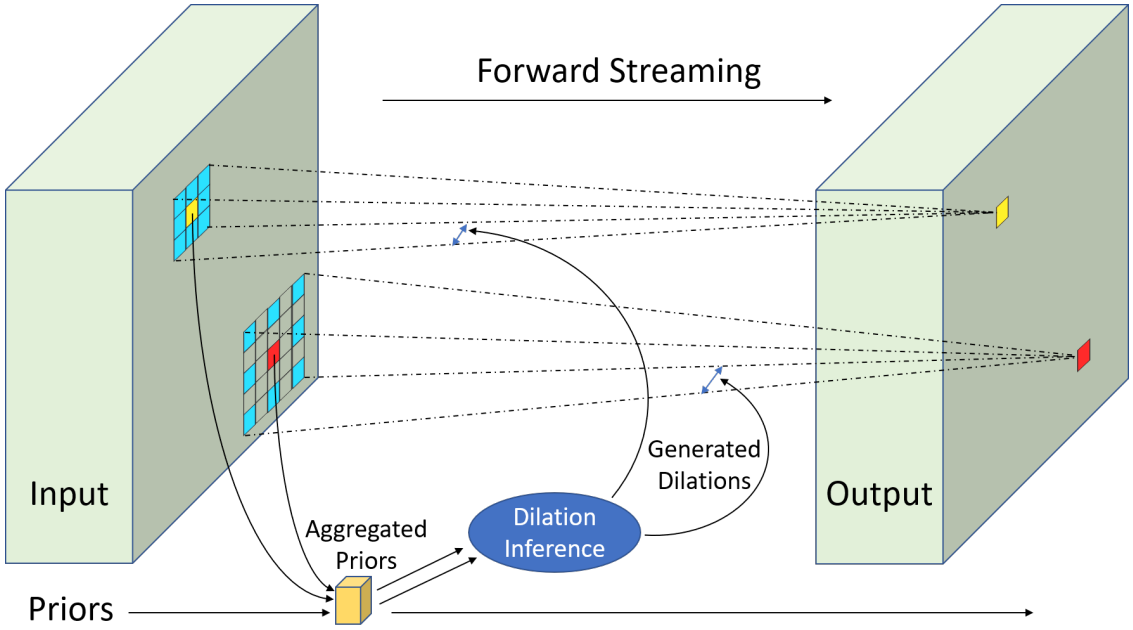


Figure 6.2: Overview of an ADCNN kernel.

6.2.2 *Pixel-wise Adaptive Dilated Convolution*

We elaborate the proposed approach for extending conventional dilated convolution kernels into ADCNN kernels. Without loss of generality, we assume all the convolutions in the rest of this paper are 2D operations. Suppose we are considering the $(l-1)$-th layer, whose input is $\mathbf{X}^{l-1}$ with $\mathbf{X}^{l-1} \in \mathbb{R}^{w^{l-1} \times h^{l-1}}$. w^{l-1}, h^{l-1} are the width and height of the input x^{l-1} respectively. $\mathscr{K}_{\mathbf{W};d}$ is a dilated convolutional kernel with dilation value d and weights $\mathbf{W}$. The output of convolution between $\mathscr{K}$ and $\mathbf{X}$ is

$$\mathbf{Y}^{l}_{i,j} = \sum_{m=0}^{K} \sum_{n=0}^{K} \mathbf{w}_{m,n} \times \mathbf{X}^{l-1}_{i+dm,j+dn} \tag{6.1}$$

where K is the kernel size and i, j are coordinates for dimensions w and h, respectively. Apparently, d is a constant variable independent to i and j. Our goal is to convert d into a function $\mathscr{D}_{i,j}$ such that the output of $\mathscr{D}_{i,j}$ could be aware of location-specific contents. More specifically, we treat $\mathscr{D}_{i,j}$ as an inference process that generates dilation values by sampling from position-dependent hidden distributions. Fig. 6.2 sketches the basic idea of a ADCNN kernel.

Dilation inference Sampling dilation values directly from categorical distributions is straightforward. However, gradients are unable to backpropagate through sampled nodes in such cases, making the entire training process intractable. Inspired by [157, 158] and [159], we employ Gumbel-Softmax (GS) [144, 160] as $\mathscr{D}_{i,j}$ to approximate the inference of discrete dilation values. Suppose that there are D valid options for dilation value, and $\mathbf{d}_{i,j} \in [0,1]^D$ is the estimation of one-hot vector that corresponds to the dilation value at position (i, j), then sampling $\mathbf{d}_{i,j} \sim \mathrm{GS}(\mathbf{h}_{i,j})$ can be achieved by

$$\mathbf{d}_{i,j} = \mathscr{D}_{i,j}(\mathbf{h}) = \frac{\exp((\mathbf{h}_{i,j} + \mathbf{g}_{i,j})/\tau)}{\sum \exp((\mathbf{h}_{i,j} + \mathbf{g}_{i,j})/\tau)} \tag{6.2}$$

where $\sum$ means summation of all tensor elements here; $\mathbf{h}, \mathbf{h}_{i,j}$ are content-related hidden priors and their subtensors at each positions, respectively; $\mathbf{g}_{i,j} \in \mathbb{R}^D$ are i.i.d. samples drawn from the $\mathrm{Gumbel}(0,1)$ distribution and τ controls how much the GS is close to a true categorical distribution.

Hidden Prior Generation As mentioned in Section 6, we believe dilation adaptation should be governed by feature hierarchy, hence build up our dilation inference mechanism upon inter-layer pattern modeling to capture dependencies between abstraction levels. Inspired by [161, 162] and [163], we consider aggregation as a feasible way and will generate hidden priors $\mathbf{h}$ through sequentially aggregating multiple $\mathbf{Y}$ from hierarchical layers. Let l denote the newly added layer index, there are several aggregation options for inter-layer patterns modeling.

Recurrent Aggregation. A straightforward way for sequential aggregation can be written as

$$\mathbf{h}_{i,j}^{l} = f(\mathbf{W}_{h}^{l}\mathbf{h}_{i,j}^{l-1} + \mathbf{U}_{h}^{l}\mathbf{Y}_{i,j}^{l-1}) \tag{6.3}$$

where $\mathbf{W}_{h}^{l}$ and $\mathbf{U}_{h}^{l}$ are 1×1 kernels weights with output channel of D; $f(\cdot)$ is a non-linear activation function. In this case, $\mathbf{h}_{i,j}^{l}$ continuously accumulates information from each layer as l goes deeper, implying layers are highly dependent on each other to mutually decide proper RF sizes.

Gated Aggregation. To model inter-layer pattern smarter, we introduce a gate variable $\mathbf{a}_{h}^{l}$ to modulate information from each layer in a data-driven manner. We use a similar way to [164] and [165] for computing $\mathbf{a}_{h}^{l}$, with which the entire aggregation can be formulated as following

$$\mathbf{h}_{i,j}^{l} = f(\mathbf{a}_{h}^{l} \circ (\mathbf{W}_{h}^{l}\mathbf{h}_{i,j}^{l-1}) + (1 - \mathbf{a}_{h}^{l}) \circ (\mathbf{U}_{h}^{l}\mathbf{Y}_{i,j}^{l-1})) \tag{6.4}$$

$$\mathbf{a}_{h}^{l} = \sigma(\mathbf{W}_{a}^{l}\mathbf{h}_{i,j}^{l-1} + \mathbf{U}_{a}^{l}\mathbf{Y}_{i,j}^{l-1}) \tag{6.5}$$

where $\sigma(\cdot)$ is the sigmoid activation and $\circ$ means element-wise multiplication. In this way, layers are not strictly dependent on their hierarchical order and will impact dilation sampling in a more complicated way.

Markov Aggregation. An important extreme case of Recurrent Aggregation, Markov Aggregation sets the kernel weights $\mathbf{W}_{h}^{l}$ from equation (6.3) to $\mathbf{0}$.

$$\mathbf{h}_{i,j}^{l} = f(\mathbf{U}_{h}^{l}\mathbf{Y}_{i,j}^{l-1}) \tag{6.6}$$

Similar to the Markov model [166], this means RF sizes are dominated by the last layer. No other inter-layer patterns need to be aggregated for multiple hierarchical layers.

6.2.3 Dilation Adaption vs. Kernel Adaption

To better understand advantages of proposed adapted dilation, it is worth comparing ADCNN with other related approaches. Recently, there are several works [152, 153, 155, 142, 143] also targeting on learning dynamic kernels based on different input contents. We give their approaches a unified name called kernel adaption, since they achieve content-awareness via directly manipulating the kernel space. For example, the modulated deformable convolution[143] can be expressed as

$$y(p) = \sum_{k=1}^{K} w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k \tag{6.7}$$

where x is the input feature map and y is the output feature map at location p. Δp_k and Δm_k are the learnable offset and modulation scalar for the k-th location, respectively. This method changes the shape of convolutional kernel by using the offsets and learning these offsets from the target task. More specifically, kernel adaption tends to learn a mapping function $\mathscr{F}$ such that $\mathbf{W}_{m,n} = \mathscr{F}_{m,n}(\mathbf{X})$, where m and n are the pixel index of the convolutional kernel, respectively.

Compared with kernel adaption, ADCNN kernels do so through a more indirect way of engaging dilation rate. Instead of kernel space, dilation function $\mathscr{D}$ sets the target on a dilation space, which contains all D possible dilation values. Theoretically, mapping inputs to dilation space rather than kernel space could have several benefits.

Low dimensional vs. high dimensional complexity. It is easy to see from previous discussions that the dimension of dilation space equals to the number of all dilation options D, while kernel space needs to keep a dimension of $C^{l-1} \times C^l$ such that it can be consistent with input and output channel size. Practically speaking, there is no need to keep a large group of dilation candidates due to their ability of exponentially enlarging RFs [137, 138]. Meanwhile, channel size usually increases dramatically as network goes deeper in order to capture more complicated high level

abstractions. These facts make D significantly smaller than $C^{l-1} \times C^l$ and leads to an easier learning process with less need of worrying about feature sparsity. Besides, low dimensional complexity also allows ADCNN kernels to be deployed to a wider level range of layers.

Dilation space sharing vs. kernel space orthogonality. Basically, kernel adaption generates kernel values using a single function for a convolution layer. So generated kernels could be highly correlated with each other. However, recent work [167] indicates spaces regularized by orthogonality constrains lead to better results and more stable training process. Therefore, it is hard to balance kernel generation and space orthogonality at the same time. Unlike kernel adaption approaches, ADCNNs mainly rely on dilation spaces, which are not only separated from individual kernel spaces but also can be shared by all convolution layers of a CNN. This means inter-layer patterns are easier to be carried over multiple layers and are able to be more coherently propagated into deeper layers through shared dilation space. Thus compared to kernel adaption, it is expected that ADCNN kernels could be aware of different input contents without interfering the orthogonality among kernel spaces.

6.2.4 ADCNNs for Semantic Segmentation

Since the proposed ADCNN module is highly related to RF adaptation, dense prediction tasks could be ideal to test its effectiveness. Thus, we first evaluate ADCNNs through semantic segmentation to explore their properties from various aspects. We will show that ADCNNs is designed for general purpose and can be applied to solve more problems in later sections.

Experiments and Assessment. We implement ADCNNs with various backbone architectures via PyTorch library [168]. In the following sections, unless otherwise specified, we will employ VGG-16 [169] as backbone net and follow the same training protocol of FCN-8s [170] as task specific framework for evaluation. All ADCNN kernels will follow Markov Aggregation with

three available dilation options $\{1,2,4\}$ ($D = 3$). We selected the dataset of Pascal VOC 2012 [171] and report mean Intersection over Union (mIoU) on its validation set as evaluation results. All the models are optimized via Adam optimizer [172].

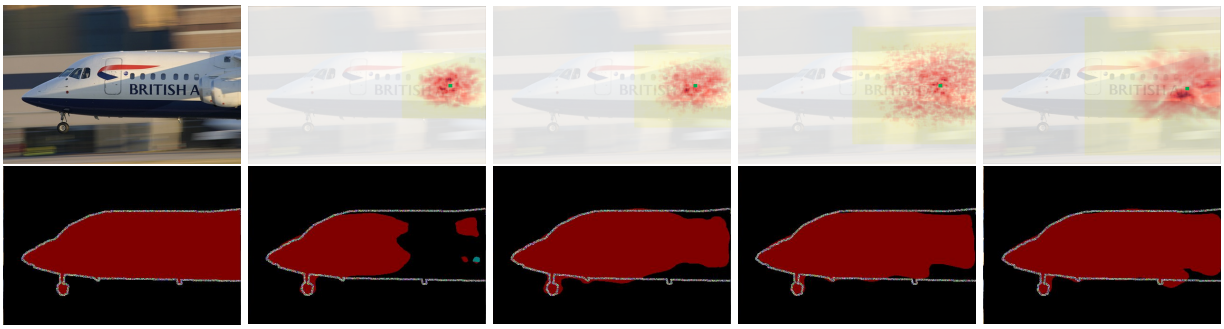
Table 6.1: mIoU for feature level study. $\sigma^2(\mathbf{d}_{i,j})$ is variance of pixel dilation sampling.

conv3	conv4	conv5	$\sigma^2(\mathbf{d}_{i,j})$	mIoU
✓			1.96×10^{-4}	63.9
	✓		1.84×10^{-4}	64.7
		✓	4.01×10^{-6}	**66.5**
✓	✓		2.45×10^{-4}	65.4
	✓	✓	1.24×10^{-4}	66.1
✓	✓	✓	1.93×10^{-4}	65.9

Feature Level Study. In this section, we conduct several experiments to answer the question: Which convolution level is suitable for ADCNN kernels? For example, considering the convolution blocks, conv3, conv4 and conv5, of a VGG-16 backbone network, if either one is evolved into ADCNN kernel, then which one can yield largest RF on the top layer (conv5-3 in this case) after training? Although for static dilation, RF size of conv5-3 should be the same no matter which block is dilated, this might not hold for ADCNN kernels with multiple dilation candidates, since dilation values are subject to various level of sensitivities due to hierarchical representations. To confirm this, we investigate several cases including both individual and combined ADCNN kernels.

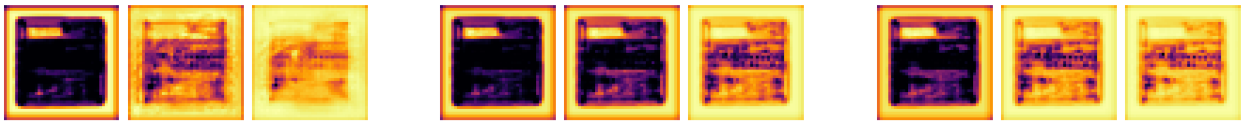
Table 6.1 summarizes the mIoU for different cases. When only one block is modified, mIoU increases when the feature level for ADCNN changes from low to high. This matches our expectation that ADCNN kernels for higher level features perform better than ADCNN kernels in lower level, as low-level ADCNN kernels are more sensitive to local variances and tend to focus on capturing information in a smaller region; while high-level kernels are usually related to complicated and abstract concepts, leading them to be more responsive for larger input regions. To further support such a claim, we visualize both RFs and Effective RFs (ERFs) [136] for a randomly picked image

and put them along with their segmentation results in Fig. 6.3. As we can see, both RFs and ERFs continuously expand their sizes as feature level for ADCNN goes higher; meanwhile, visually better segmentation results can be achieved with larger RFs and ERFs. This provides us a supportive example that encourages ADCNN extension for higher feature level in practice.



(a) Image & Ground Truth (b) conv3 only (c) conv4 only (d) conv5 only (e) conv3 + conv4 + conv5

Figure 6.3: The top row indicates the input image and its visualized RFs and ERFs on conv5-3 layer of LSD-VGG16 with different conv blocks modified. Patches means RFs and red dots inside are ERFs. The bottom row shows the ground truth and corresponding segemtation results. GT stands for groundtruth.



(a) Markov Pattern (b) Gated Pattern (c) Recurrent Pattern

Figure 6.4: Mathematical expectation of dilation sampling at each pixel for individual sub-layers (from left to right: conv5-1 to conv5-3). Brighter color means higher dilation and vise versa. The input is the same as the one in Figure 6.1.

Besides, we also test several cases of combining multiple extended blocks into more complicated ADCNN architectures (the last three lines of Table 6.1). To our surprise, stacking additional ADCNN-blocks may result in inferior performances to single block even with better ERF. We further investigate possible explanations by calculating the variances of dilation sampling for each cases. We find performances always decrease when conv5 is combined with more ADCNN-blocks,

along with notable variance increments. Such increments brought by additional sampling might be the reason for performance downgrading as they make the entire structure more unstable.

Table 6.2: Aggregation study on different backbones and varied tasks

Task	Semantic Segmentation	
Backbone	VGG-16	ResNet-101
Vanilla Non-Aggregation	64.7	75.1
ADCNN Markov Aggregation	**66.5**	**77.2**
ADCNN Gated Aggregation	65.5	76.7
ADCNN Recurrent Aggregation	65.3	75.6

Pattern Aggregation Study. Now we focus on studying the impacts brought by each pattern aggregation strategies described in Section 6. As suggested from Section 6, we only extend conv5 block of a VGG-16 backbone into ADCNN kernels to avoid too much dilation sampling. All three (conv5-1, conv5-2 and conv5-3) sub-layers are upgraded with ADCNN kernels and connected as each aggregation asks. We also include ResNet-101 [173] combined with DeeplabV3+ [174] as an additional backbone to see if skip connections may result in different impacts.

Table 6.3: Performance of ADCNN-ResNet-101 on the CityScapes validation set.

Backbone	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain
ADCNN-ResNet-101	**0.984**	**0.867**	**0.934**	0.610	**0.654**	**0.668**	**0.737**	**0.817**	**0.930**	0.653
ResNet-101	0.983	0.860	0.931	**0.625**	0.638	0.648	0.726	0.801	0.929	**0.659**
Backbone	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
ADCNN-ResNet-101	**0.954**	**0.840**	**0.674**	**0.956**	**0.810**	**0.919**	0.808	**0.722**	**0.796**	**0.807**
ResNet-101	0.953	0.833	0.658	0.953	0.797	0.912	**0.815**	0.720	0.787	0.801

The results are concluded in Table 6.2. Basically, all three strategies have better results than backbone networks. However, for both cases Markov Aggregation always yields a better result than other two options. To further dig up the roots behind such phenomenon, in Fig. 6.4, we calculate and visualize the mathematical expectations at each pixel for all three sub-convolution layers of ADCNN-VGG16. We can see that during the streaming from conv5-1 to conv5-3, ADCNN with

Markov Aggregation is more likely to choose larger dilation everywhere without carrying spatial patterns of input; while both Gated and Recurrent Aggregation are more willing to adjust RF sizes according to spatial structures from input and reserve some spatial clues for dilation sampling. In such cases, information aggregated by lower level features could be too local-sensitive, forcing next layer to put its RF in a smaller region in order to capture such local variations. Thus, our results for semantic segmentation indicate Markov Aggregation is the best option among the three without overly aggregating inter-layer patterns.

Table 6.4: ADCNN-FCN8s IoUs on VOC-2012 across all classes

Backbone	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
ADCNN-FCN8s	**0.914**	**0.833**	**0.388**	0.751	**0.627**	0.740	**0.802**	0.744	**0.805**	0.252	**0.805**
FCN8s	0.908	0.798	0.363	**0.776**	0.581	**0.742**	0.775	**0.749**	0.799	**0.292**	0.712
Backbone	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	mIoU
ADCNN-FCN8s	**0.474**	**0.724**	**0.729**	**0.783**	**0.791**	**0.510**	**0.729**	**0.370**	0.773	**0.600**	**0.665**
FCN8s	0.375	0.684	0.673	0.765	0.780	0.490	0.760	0.344	**0.789**	0.572	0.647

Table 6.5: ADCNN-ResNet-101 IoUs on VOC-2012 across all classes

Backbone	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
ADCNN-ResNet-101	**0.932**	**0.838**	**0.393**	0.848	0.622	**0.756**	0.908	**0.848**	**0.918**	**0.373**	**0.874**
ResNet-101	0.922	0.770	0.388	**0.853**	**0.626**	0.698	**0.913**	0.836	0.886	0.225	0.835
Backbone	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	mIoU
ADCNN-ResNet-101	**0.584**	**0.879**	**0.851**	0.805	**0.833**	**0.554**	**0.852**	**0.534**	**0.835**	0.648	**0.772**
ResNet-101	0.568	0.862	0.791	**0.810**	0.815	0.452	0.764	0.461	0.824	**0.691**	0.751

Table 6.6: ADCNN-DRN-54 IoUs on VOC-2012 across all classes

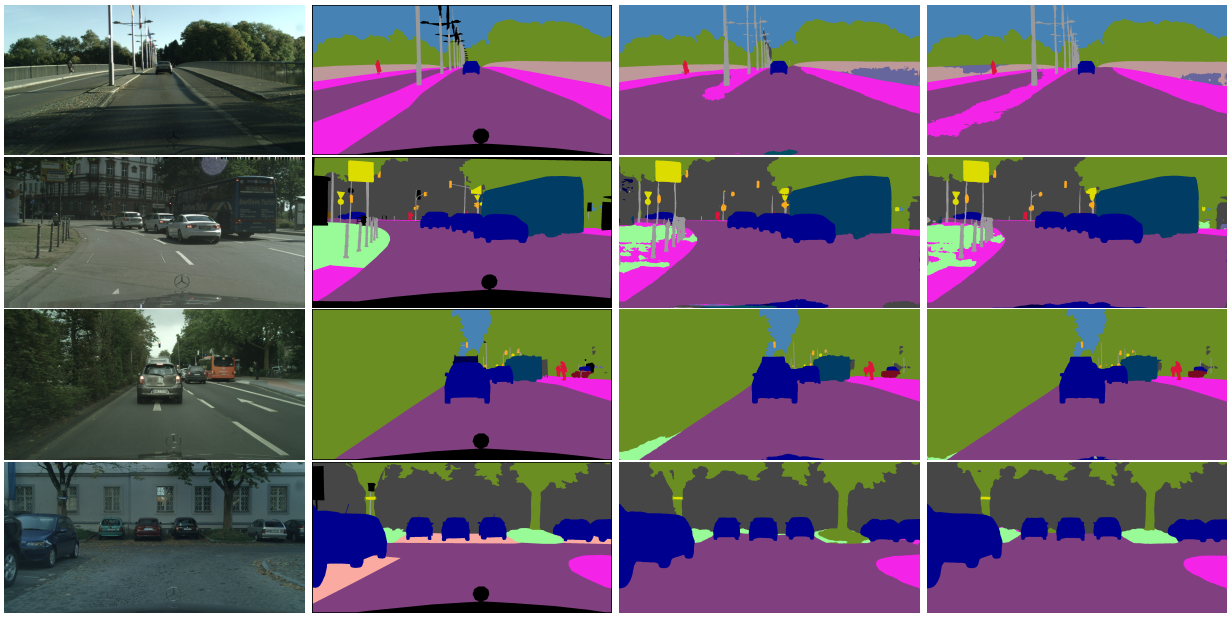
Backbone	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow
ADCNN-DRN-54-D	**0.927**	**0.823**	**0.384**	0.845	**0.668**	**0.729**	**0.915**	0.838	0.852	0.294	**0.876**
DRN-54-D	0.921	0.799	0.345	**0.846**	0.660	0.723	0.868	**0.848**	**0.884**	**0.313**	0.820
Backbone	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor	mIoU
ADCNN-DRN-54-D	**0.568**	0.839	**0.836**	**0.814**	**0.813**	**0.491**	**0.805**	0.434	**0.781**	**0.693**	**0.772**
DRN-54-D	0.528	**0.840**	0.801	0.805	0.800	0.475	0.739	**0.492**	0.750	0.675	0.754

Performance Boosting for Backbone Architectures. Finally, we verify ADCNNs can be easily combined various popular base architectures to further improve their performance. In addition to VGG-16, we also employ another four representative architectures, ResNet-101 [173], Dilated

Table 6.7: Semantic Segmentation Experiments on validation sets of VOC 2012 and Cityscapes

Task	Method	mIoU	
		regular	ADCNN
Pascal VOC 2012	VGG-16+FCN-32s	62.8	**65.1**
	VGG-16+FCN-8s	64.7	**66.5**
	ResNet-101+Deeplabv3+	75.1	**77.2**
	Xception+Deeplabv3+	73.5	**74.4**
	DRN-D-54+Deeplabv3+	75.4	**77.2**
Cityscape	MobileNetv2+Deeplabv3+	70.3	**71.5**
	Xception+Deeplabv3+	77.5	**79.0**
	ResNet-101+Deeplabv3+	80.1	**80.7**

Residual Networks (DRN) [138], Xception [175] and MobileNet-v2 [176], as additional backbone nets. We combined these base structures with FCN [170] and Deeplabv3+ [174] framework and evaluate them on CityScapes [177], a more challenging dataset.



(a) Image (b) Groundtruth (c) ResNet-101 (d) ResNet-101 + ADCNN

Figure 6.5: Semantic segmentation results on CityScapes dataset.

We report mIoUs for each backbone network and corresponding ADCNN in Table 6.8, respec-

Table 6.8: Semantic Segmentation Experiments on validation sets of VOC 2012 and CityScapes

Task	Method	mIoU	
		regular	ADCNN
Pascal VOC 2012	VGG-16+FCN-32s	62.8	**65.1**
	VGG-16+FCN-8s	64.7	**66.5**
	ResNet-101+Deeplabv3+	75.1	**77.2**
	Xception+Deeplabv3+	73.5	**74.4**
	DRN-D-54+Deeplabv3+	75.4	**77.2**
CityScapes	MobileNetv2+Deeplabv3+	70.3	**71.5**
	Xception+Deeplabv3+	77.5	**79.0**
	ResNet-101+Deeplabv3+	80.1	**80.7**

tively, along with other state-of-the-art results for comparison. From these results we can see ADCNNs could always yield better results for every backbone structure on both datasets, exhibiting strong robustness and versatility. We also visualize part of segmentation results in Fig. 6.5, which coincides with mIoU that ADCNNs have more correctly labeled pixels and more details preserved. And the results on class Iou of Cityscapes is shown in Table 6.3. We report more class IoUs, which are included in Table 6.4, Table 6.5 and Table 6.6. Each IoU is reported based on each segmentation class. The final mean of IoU (mIoU) over all classes is reported in the end column of each table. The higher IoUs are bold.

6.2.5 Summary

In this paper we formulate the dilation as a learnable weight for convolution kernels such that its value can be dynamically decided during the running time. This leads to ADCNNs, a light-weighted, end-to-end trainable framework that allows their kernels to adjust pixel-wise RFs in a data-driven manner. To infer proper dilation values based on feature hierarchy, we model inter-layer patterns via several sequential aggregation strategies. Our studies on semantic segmentation explore various properties of ADCNNs. Results indicate better performance can be achieved with

all three aggregation strategies when ADCNN kernels are with higher feature levels, and dilation boundary can be learned to avoid overlarge RFs. We also demonstrate ADCNNs can consistently boost performances over several popular backbone architectures, and be a valuable option for more general visual tasks such as large-scale and fine-grained image classifications.

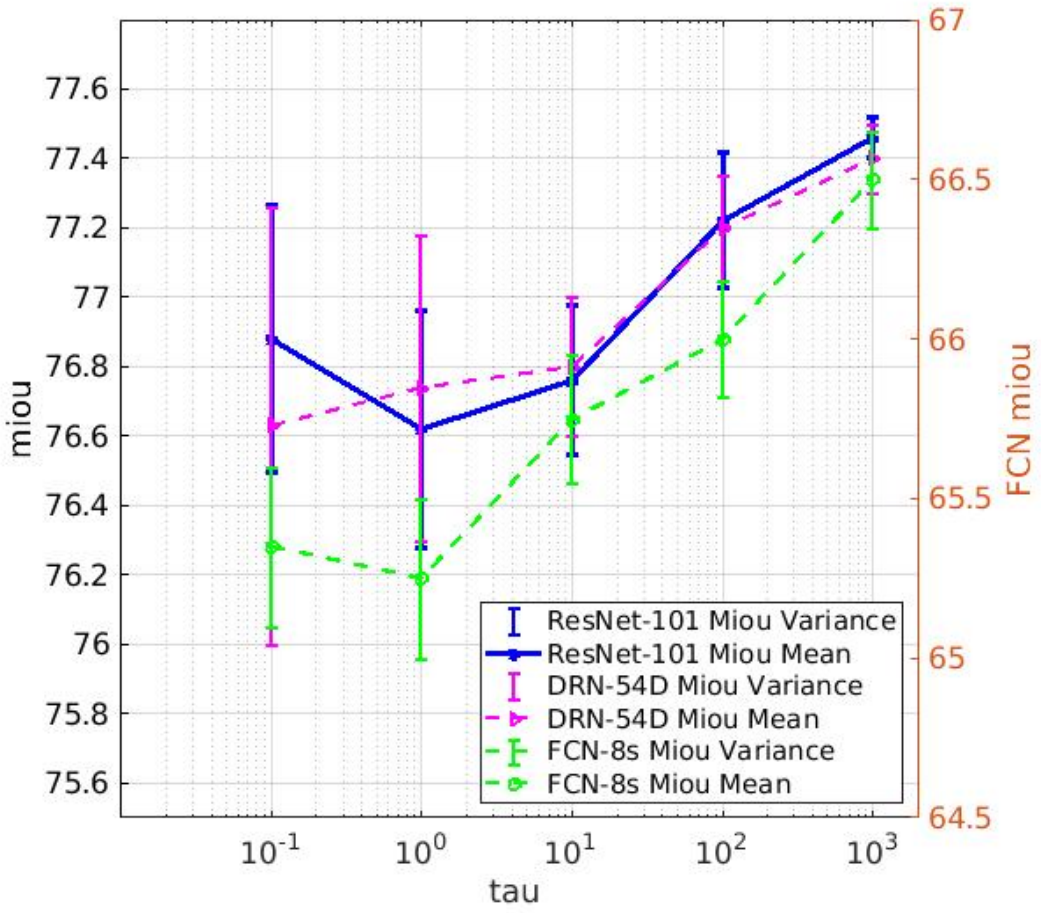


Figure 6.6: The sensitivity analysis on τ by performing semantic segmentation task on VOC 2012 validation set with three backbone nets. The mean and variance at each τ value are computed by repeating 5 times with same settings.

6.3 Calibrating Semantic Segmentation Models through Ensemble Distillation

Given the investigation study on semantic segmentation modeling, this dissertation further explores semantic segmentation calibration with knowledge distillation. The problem is formulated by the optimization on both model accuracy and model reliability. To achieve this goal, the proposed approach has to integrate accurate training with calibration techniques. The proposed framework is ensemble knowledge distillation since both higher accuracy and better calibration can derived. In addition to ensemble modeling, some existing post-hoc calibration techniques are employed to compare ensemble knowledge distillation and justify its performance.

6.3.1 Preliminaries

Similar to image classification, semantic segmentation can be formulated to a multiclass classification problem with a deep neural network. Let $x \in X$ and $y \in Y$ denote an input and its label, respectively. A deep neural network $h(x) = (\hat{y}, \hat{p})$ yields $\hat{y}$ as the predicted label with confidence $\hat{p}$. We expect a well-calibrated model to provide accurate prediction when its confidence is high.

ECE: There are several metrics to measure a model's calibration, and one of the most popular and accepted metrics is *expected calibration error (ECE)*[178], which reflects the gap between predictive confidence and accuracy. The formal definition with continuous variable is as follows.

$$ECE = \mathbb{E}\left[|P(\hat{y} = y|\hat{p}) - \hat{p}|\right], \tag{6.8}$$

where $\hat{y}$ is the predicted label, y is the true label, $\hat{p}$ is the model confidence or probability in its prediction, and $P(\hat{y} = y|\hat{p})$ is the probability for correct prediction. The expectation is about the discrepancy between accuracy and confidence. A perfectly calibrated model has zero ECE.

In practical problems, statistical binning is used to quantize continuous variables and estimate eq. (6.8) by equally binning the probability interval,

$$\widehat{ECE} = \sum_{i=1}^{m} \frac{|B_i|}{n} |\text{acc}(B_i) - \text{conf}(B_i)|, \tag{6.9}$$

where n is the number of samples, m is the number of bins, B_i denotes a set of samples falling into it, and $\text{acc}(B_i)$ and $\text{conf}(B_i)$ are accuracy and confidence averaged over the samples in the bin. $\widehat{ECE}$ can be visualized with the gaps in reliability diagram [179].

ECE in semantic segmentation: We extend ECE to semantic segmentation by considering each pixel as a sample. Instead of pooling all pixels of different images into a set, we calculate ECE over each image first before taking an average across images,

$$ECE = \frac{1}{N} \sum_{I=1}^{N} ECE_I, \tag{6.10}$$

where I is an inference image, and N is the total number of images. ECE_I may significantly vary across images.

6.3.2 Existing Calibration Methods

Temperature Scaling is a simple but effective approach for multi-classification model calibration [13]. The calibration is carried out with a single temperature parameter to scale logits for overfitting problem resolution.

$$\hat{p} = \sigma_{SM}(z/T), \tag{6.11}$$

where T is a scaler of temperature to scale logit vector z.

Logistic Scaling is an extension of temperature scaling a.k.a. vector scaling [13]. The scaling

model is formulated with linear transformation for more complex calibration map.

$$\hat{p} = \sigma_{SM}(w \odot z + b), \tag{6.12}$$

where w and b are two vectors to scale the logit vector z.

Dirichlet Scaling is the extension of logistic scaling and derived with probability output distribution [117], which enriches calibration map for better optimization with Dirichlet distribution. We adopt linear paramenterization [117] for Dirichlet scaling, which is formulated as follows.

$$\hat{p} = \sigma_{SM}(W \cdot log(\sigma_{SM}(z)) + b), \tag{6.13}$$

where W is a matrix and b is a vector for linear parametrisation of the probability $\sigma_{SM}(z)$.

Ensembling is proposed to solve medical image segmentation uncertainty prediction [180]. It carries out calibration improvement with simple average of ensemble model, a simplified version of Bayesian inference [45]. Also, deep ensemble is a strong baseline for image classification calibration [5, 45]. We compare its performance with other post-hoc approaches.

$$\hat{p} = \frac{1}{N}\Sigma_{n=1}^{N}\hat{p}^n, \tag{6.14}$$

where N is the number of ensemble members.

6.3.3 Experiments

Models. We consider five recent state-of-the-art semantic segmentation models. Our selection of models covers CNN and ViT architectures.

1. **SegFormer**[126] is a ViT-based encoder-decoder model based upon lightweight multilayer perception (MLP) decoders with pyramid architecture.
2. **Segmenter**[125] is an encoder-decoder model based exclusively on Transformer.
3. **Knet** [123] is a unified segmentation decoder module which includes two important variants, Knet-DeepLab and Knet-SWIN. Knet-DeepLab consists of ResNet-50 [69] backbone and DeepLab-V3 [121] decoder. Knet-SWIN is contructed with SWIN [181] backbone and UperNet [182] decoder.
4. **ConvNeXt** [183] is a CNN backbone, but shaped to ViT-like architecture for better scaling. We select ConvNeXt backbone with UperNet [182] decoder as a segmentation model.

Dataset. We examine existing calibration methods across six important benchmarks from various applications. Figure 6.7 illustrates the example images from benchmarks.

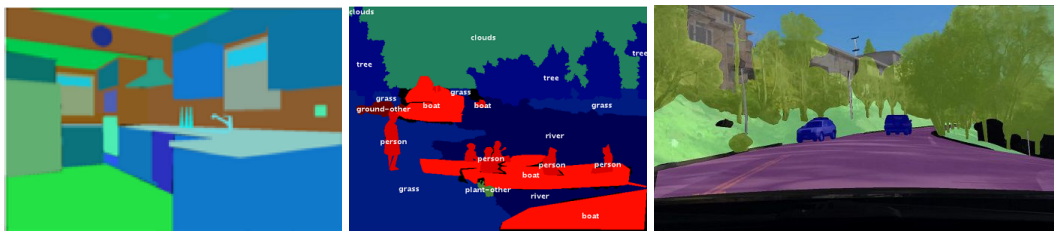


Figure 6.7: Segmented examples from ADE20K, COCO-164K, and BDD100K (left to right).

1. **Scene and stuff segmentation.** We use ADE20K [127] and COCO-164K [128] as large-scale segmentation benchmark. ADE20K contains 150 object and stuff classes with 20,210/2,000 images in the training/validation set. COCO-164K, $\sim$164K images for 91 stuff and 80 thing classes, includes 118K/5K images in the training/validation set.
2. **Autonomous driving.** We choose BDD-100K [129], a latest benchmark for urban driving scene segmentation. BDD-100K contains 7K/1K 1280$\times$720 images in 19 classes for training/validation set. We select the validation set of CityScapes [184] as test set from target domain for domain-shift calibration assessment. CityScapes has 2975/500 1024$\times$2048 images for training/validation set. BDD-100K and CityScapes share the same label space.

Training. All training settings are unified for fair comparison. For ADE20K, all models except Knet-DeepLab are trained with 640×640 crop size. For BDD100K, the crop size is 512×1024. For other benchmarks, the crop size is 512×512. The batch size is set to 8. For data augmentation, stronger random crop and longer training are used to carry out ensemble knowledge distillation.

6.3.4 Results

We present ECEs for five models and compare ensemble knowledge distillation with four calibrators. Table 6.9 shows that majority of calibrators yield very limited improvement in model calibration. However, surprisingly, ensemble knowledge distillation consistently outperforms these scaling approaches. This calibration gain is attributed to both calibration property derivation form ensemble model and more reduction on mispredictions. Moreover, this improvement is more significant for scene and object benchmarks such as ADE20K and COCO-164K than street view datasets like BDD-100K. This observation is related to scenery variation. When variation in scene is large, background and object identification is more challenging for models, and thus, confidence becomes lower. This phenomena can be alleviated by distillation to some extent, but the improvement is still limited.

Furthermore, we find that BDD100K yields more calibrated models than other benchmarks. We associate it with larger crop size (512×1024) and higher model accuracy. We also find that ensembling exhibits weaker calibration. We connect it with model accuracy since we restrain the ensemble accuracy for fair comparison, which limits its calibration performance. Moreover, across models, Segmenter exhibits better calibration. Since it is a Transformer-exclusive model, we link it to different spatial inductive bias which is speculated in [185].

This experiment reveals that ensemble knowledge distillation successfully inherits scaling competence of ensemble model, and furthermore, improves the ensembling calibration through stronger

data augmentation. Stronger data augmentation enhances model generalization to unseen data, which both reduces mispredictions and corrects overconfidence on these unseen samples. It is effectively integrated with ensemble modeling through knowledge distillation and further boosts model performance. This is the speculation on the observation of ensemble knowledge distillation for higher accuracy and lower calibration errors.

Table 6.9: Segmentation accuracy (mIoU) and calibration error (ECE) on different benchmarks. TempS, LogS, DirS, Ens., and EKD denote temperature, logistic, Dirichlet, ensembling, and ensemble knowledge distillation, respectively. For ensembling, we achieve three models with reduced size for comparable mIoU.

Dataset	Model	mIoU	Uncal	TempS	LogS	DirS	Ens.	mIoU	EKD
ADE20K	SegFormer-B5 [126]	49.13	0.111	0.109	0.110	0.110	0.109	49.70	**0.100**
ADE20K	Segmenter-L [125]	51.65	0.087	0.086	0.086	0.087	0.086	52.01	**0.080**
ADE20K	Knet-DeepLab [123]	45.06	0.111	0.105	0.107	0.106	0.110	45.74	**0.101**
ADE20K	Knet-SWIN-L [123]	52.46	0.098	0.094	0.093	0.097	0.096	52.79	**0.089**
ADE20K	ConvNeXt-L [183]	53.16	0.097	0.092	0.094	0.091	0.094	53.43	**0.089**
COCO-164K	SegFormer-B5 [126]	45.78	0.151	0.149	0.141	0.151	0.149	46.01	**0.140**
COCO-164K	Segmenter-L [125]	47.09	0.152	0.149	0.149	0.151	0.150	47.40	**0.142**
COCO-164K	Knet-DeepLab [123]	37.24	0.170	0.170	0.168	0.171	0.169	37.61	**0.161**
COCO-164K	Knet-SWIN-L [123]	46.49	0.161	0.159	0.161	0.160	0.160	46.77	**0.151**
COCO-164K	ConvNeXt-L [183]	46.48	0.160	0.157	0.158	0.159	0.159	46.69	**0.151**
BDD100K	SegFormer-B5 [126]	65.08	0.064	0.055	0.054	0.053	0.059	65.70	**0.051**
BDD100K	Segmenter-L [125]	61.33	0.055	0.045	0.043	0.042	0.052	61.81	**0.041**
BDD100K	Knet-DeepLab [123]	62.89	0.060	0.049	0.047	0.048	0.057	63.10	**0.045**
BDD100K	Knet-SWIN-L [123]	67.59	0.065	0.055	0.054	0.054	0.063	67.99	**0.053**
BDD100K	ConvNeXt-L [183]	67.26	0.064	0.054	0.053	0.056	0.063	67.91	**0.052**

6.3.5 Summary

This chapter focuses on extension of the research about accuracy and reliability to semantic segmentation tasks. It starts with investigation on modeling of semantic segmentation task and understanding of its mechanism from convolution modeling. Given the understanding, an algorithm is proposed based upon adaptive dilation across images which improves perceiving dynamics on

receptive field. This algorithm increases segmentation accuracy compared to other state-of-the-art convolution models. Through further analysis on different information aggregation pattern, the study finds that within network, connecting layers yield more correlated information flow which helps segmentation modeling. After this study, model calibration is carried out through knowledge distillation on ensemble model with stronger data augmentation. Large scale datasets are evaluated and used for comparing ensemble knowledge distillation with other calibration approaches, such as ensemble model. The results show that ensemble knowledge distillation outperforms on both accuracy of mIoU and the calibration error of ECE. This justifies that knowledge distillation enables improving semantic segmentation deep neural network training on both accuracy and calibration.

CHAPTER 7: CONCLUSION

7.1 Overall Summary

This dissertation explores how to improve deep neural network training with knowledge distillation. As an effective compression approach, knowledge distillation has been widely used to solve deep neural network reduction problems. The primary idea of knowledge distillation is to obtain a smaller neural network which can imitate the outputs of the large model. The smaller model is named student model while the larger one is teacher model. Given the optimization based upon imitation training with knowledge distillation, the student network is capable of retaining the teacher's accuracy while the model size is significantly reduced. Moreover, its effectiveness is not only reflected on model reduction, but more research finds knowledge distillation enables better model generalization as well. Furthermore, there are still problems for applying knowledge distillation to solve network training. This dissertation attempts to tackle some of these challenges to promote application of knowledge distillation.

According to the literature review on knowledge distillation and its related topics, including accurate model training, model reliability improvement, data augmentation, ensemble modeling, and semantic segmentation, this dissertation addresses four questions:

1. How to efficiently distill a blackbox teacher model in a data-efficient manner? Given formulated blackbox knowledge distillation, this dissertation attempts to solve the problem with constraints of limited labeled data, restricted model query, and limited computation source (cf. Chapter 3).
2. How to accurately predict time series trend through calibrating physics-based model with limited observation data? Inspired from blackbox knowledge distillation, this dissertation

proposes a new method to render calibration possible and reliable when observation data are limited (cf. Chapter 4).

3. How to derive an accurate and well-calibrated deep neural network? This dissertation integrates ensemble modeling and knowledge distillation to obtain a well-calibrated model while accuracy is also boosted over distillation (cf. Chapter 5).
4. How to extend the proposed methods based upon knowledge distillation to semantic segmentation task? This dissertation starts with an investigation on semantic segmentation from perspective of convolution neural network. An algorithm is proposed to improve semantic segmentation accuracy. Given the understanding, ensemble knowledge distillation is used to further improve semantic segmentation calibration (cf. Chapter 6).

From extensive experiments across different latest benchmarks, all proposed methods are justified to outperform corresponding state-of-the-art approaches. Given the results, more detailed analysis on observation data is conducted to provide more insights for further improvement in proposed approaches. The experiments for ablation study also reveal more information on the effectiveness of proposed approaches and facilitate the application of the proposed method to other real-world problems.

7.2 Future Work

7.2.1 Distillation Calibration on Temporal Modeling

Current work focuses on spatial model distillation and calibration. With rapid development of deep learning techniques, more applications are developed with temporal modeling. Similar to spatial modeling, temporal model can also encounter the problems like data limitation and severe miscalibation. It is critical to tackle these challenges before getting these temporal-model-based

applications into practice. Accordingly, one future research direction from this dissertation is to extend the proposed distillation methods to temporal modeling, such as temporal model calibration. Due to different modeling strategies, the optimization approaches will differ, especially for data augmentation. How to design an effective and efficient augmentation pipeline to improve temporal model calibration? How to distill temporal model with lower cost but yielding higher accuracy? Would semantic segmentation be easily solved with current modeling approaches in a temporal manner? These questions will be explored in future. Over the exploration, well-designed experiments on challenged benchmarks will be carried out to ensure the findings are reliable.

7.2.2 On Calibration of Semantic Segmentation Models

Semantic segmentation is a core fundamental computer vision task for a variety of applications, such as autonomous driving. Although the current work from this dissertation provides a solution to efficiently improving semantic segmentation on both accuracy and calibration, the optimization approaches are still not fully explored and there is a possibility to further enhance calibration. Will post-hoc calibration help reduce miscalibration? If so, how to develop a simple yet effective approach to calibrate semantic segmentation models? Is that possible to derive a unified and general calibration solution to both spatial and temporal segmentation modeling? These remaining questions will be studied in future. More extensive experiments on more challenged benchmarks will be conducted to justify speculations or proposed methods. A more systematic study will be conducted to understand this topic.

LIST OF REFERENCES

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

[2] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[3] Max Ryabinin, Andrey Malinin, and Mark Gales. Scaling ensemble distribution distillation to many classes with proxy targets. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[5] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

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

[7] Akisato Kimura, Zoubin Ghahramani, Koh Takeuchi, Tomoharu Iwata, and Naonori Ueda. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039*, 2018.

[8] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

[9] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751, 2019.

[10] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018.

[11] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *International Conference on Machine Learning*, pages 1602–1611, 2018.

[12] 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. *arXiv preprint arXiv:1905.08094*, 2019.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[14] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2019.

[15] Giung Nam, Jongmin Yoon, Yoonho Lee, and Juho Lee. Diversity matters when learning from ensembles. *Advances in Neural Information Processing Systems*, 34, 2021.

[16] Muhamad Risqi U Saputra, Pedro PB de Gusmao, Yasin Almalioglu, Andrew Markham, and Niki Trigoni. Distilling knowledge from a deep pose regressor network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 263–272, 2019.

[17] Chaoyang Wang, Chen Kong, and Simon Lucey. Distill knowledge from nrsfm for weakly supervised 3d pose learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 743–752, 2019a.

[18] Xuecheng Nie, Yuncheng Li, Linjie Luo, Ning Zhang, and Jiashi Feng. Dynamic kernel distillation for efficient pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6942–6950, 2019.

[19] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1013–1021, 2019.

[20] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3573–3582, 2019.

[21] Jiajun Deng, Yingwei Pan, Ting Yao, Wengang Zhou, Houqiang Li, and Tao Mei. Relation distillation networks for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7023–7032, 2019.

[22] Mohammad Tavakolian, Hamed R Tavakoli, and Abdenour Hadid. Awsd: Adaptive weighted spatiotemporal distillation for video representation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8020–8029, 2019.

[23] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5589–5597, 2018.

[24] Chuang Gan, Hang Zhao, Peihao Chen, David Cox, and Antonio Torralba. Self-supervised moving vehicle tracking with stereo sound. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7053–7062, 2019.

[25] Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1355–1364, 2019.

[26] Jogendra Nath Kundu, Nishank Lakkakula, and R Venkatesh Babu. Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1436–1445, 2019.

[27] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019.

[28] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. *arXiv preprint arXiv:1904.09149*, 2019.

[29] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. *arXiv preprint arXiv:1904.01866*, 2019.

[30] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4794–4802, 2019.

[31] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5007–5016, 2019.

[32] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[33] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[34] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[35] Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2020.

[36] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[37] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

[38] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*, 2017.

[39] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018a.

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

[41] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019a.

[42] Zakaria Laskar and Juho Kannala. Data-efficient ranking distillation for image retrieval. *arXiv preprint arXiv:2007.05299*, 2020.

[43] Hossein Mobahi, Mehrdad Farajtabar, and Peter L Bartlett. Self-distillation amplifies regularization in hilbert space. *arXiv preprint arXiv:2002.05715*, 2020.

[44] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[45] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

[46] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7047–7058, 2018.

[47] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[48] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3714–3722, 2019b.

[49] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[50] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[51] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

[52] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[53] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[54] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *arXiv preprint arXiv:1905.00441*, 2019.

[55] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

[56] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

[57] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017a.

[58] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[59] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[60] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

[61] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b.

[62] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[63] 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.

[64] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[65] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[66] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2017a.

[67] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20:5, 2015.

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

[69] 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, 2016a.

[70] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

[72] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[73] William Ogilvy Kermack and Anderson G McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[74] Michael A Benjamin, Robert A Rigby, and D Mikis Stasinopoulos. Generalized autoregressive moving average models. *Journal of the American Statistical association*, 98(461): 214–223, 2003.

[75] Shihao Yang, Mauricio Santillana, and Samuel C Kou. Accurate estimation of influenza epidemics using google search data via argo. *Proceedings of the National Academy of Sciences*, 112(47):14473–14478, 2015.

[76] Shihao Yang, Mauricio Santillana, John S Brownstein, Josh Gray, Stewart Richardson, and SC Kou. Using electronic health records and internet search information for accurate influenza forecasting. *BMC infectious diseases*, 17(1):332, 2017.

[77] Joseph T Wu, Kathy Leung, and Gabriel M Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in wuhan, china: a modelling study. *The Lancet*, 395(10225):689–697, 2020.

[78] Zixin Hu, Qiyang Ge, Li Jin, and Momiao Xiong. Artificial intelligence forecasting of covid-19 in china. *arXiv preprint arXiv:2002.07112*, 2020.

[79] Zifeng Yang, Zhiqi Zeng, Ke Wang, Sook-San Wong, Wenhua Liang, Mark Zanin, Peng Liu, Xudong Cao, Zhongqiang Gao, Zhitong Mai, et al. Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions. *Journal of Thoracic Disease*, 12(3):165, 2020.

[80] Simon James Fong, Gloria Li, Nilanjan Dey, Rubén González Crespo, and Enrique Herrera-Viedma. Composite monte carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction. *Applied Soft Computing*, page 106282, 2020.

[81] Lijing Wang, Jiangzhuo Chen, and Madhav Marathe. Defsi: Deep learning based epidemic forecasting with synthetic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9607–9612, 2019b.

[82] Fred Brauer. Mathematical epidemiology: Past, present, and future. *Infectious Disease Modelling*, 2(2):113–127, 2017.

[83] Shweta Bansal, Bryan T Grenfell, and Lauren Ancel Meyers. When individual behaviour matters: homogeneous and network models in epidemiology. *Journal of the Royal Society Interface*, 4(16):879–891, 2007.

[84] Stanley J Farlow. *Partial differential equations for scientists and engineers*. Courier Corporation, 1993.

[85] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 20(5):533–534, 2020.

[86] Irani Thevarajan, Thi HO Nguyen, Marios Koutsakos, Julian Druce, Leon Caly, Carolien E van de Sandt, Xiaoxiao Jia, Suellen Nicholson, Mike Catton, Benjamin Cowie, et al. Breadth of concomitant immune responses prior to patient recovery: a case report of non-severe covid-19. *Nature medicine*, 26(4):453–455, 2020.

[87] Ying Liu, Albert A Gayle, Annelies Wilder-Smith, and Joacim Rocklöv. The reproductive number of covid-19 is higher compared to sars coronavirus. *Journal of travel medicine*, 2020a.

[88] Johannes Bracher, Evan L Ray, Tilmann Gneiting, and Nicholas G Reich. Evaluating epidemic forecasts in an interval format. *arXiv preprint arXiv:2005.12881*, 2020.

[89] Evan L Ray, Nutcha Wattanachit, Jarad Niemi, Abdul Hannan Kanji, Katie House, Estee Y Cramer, Johannes Bracher, Andrew Zheng, Teresa K Yamana, Xinyue Xiong, et al. Ensemble forecasts of coronavirus disease 2019 (covid-19) in the us. *MedRXiv*, 2020.

[90] Yarin Gal et al. Uncertainty in deep learning.

[91] Zoubin Ghahramani and Carl Rasmussen. Bayesian monte carlo. *Advances in neural information processing systems*, 15, 2002.

[92] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how*, volume 488. Springer, 2005.

[93] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[94] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *arXiv preprint*, 2017.

[95] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[96] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[97] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020b.

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

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

[100] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019.

[101] Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. Combining ensembles and data augmentation can harm your calibration. *arXiv preprint arXiv:2010.09875*, 2020.

[102] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[103] Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Interspeech*, pages 3697–3701, 2017.

[104] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[105] Stanford CS231N. Tiny imagenet visual recognition challenge, 2016.

[106] 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.

[107] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3606–3613. IEEE, 2014.

[108] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[109] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[110] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[111] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34, 2021.

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

[113] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[114] Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. Energy-based open-world uncertainty modeling for confidence calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9302–9311, 2021.

[115] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

[116] Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. *arXiv preprint arXiv:2006.13092*, 2020.

[117] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

[118] Xingchen Ma and Matthew B Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In *International Conference on Machine Learning*, pages 7235–7245. PMLR, 2021.

[119] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[120] Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. In *Uncertainty in Artificial Intelligence*, pages 1169–1179. PMLR, 2020.

[121] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018a.

[122] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic seg-

mentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

[123] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *NeurIPS*, 2021a.

[124] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2021.

[125] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[126] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

[127] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. CVPR*, 2017b.

[128] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[129] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[130] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[131] Adam Van Etten, Daniel Hogan, Jesus Martinez Manso, Jacob Shermeyer, Nicholas Weir, and Ryan Lewis. The multi-temporal urban development spacenet dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2021.

[132] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein. Brain tumor segmentation and radiomics survival prediction: Contribution to the brats 2017 challenge. In *International MICCAI Brainlesion Workshop*, pages 287–297. Springer, 2017.

[133] Krishna Wadhwani and Suyash P Awate. Controllable image generation with semi-supervised deep learning and deformable-mean-template based geometry-appearance disentanglement. *Pattern Recognition*, page 108001, 2021.

[134] Zhuoyi Zhang, Yifeng Zhang, Xu Cheng, and Guojun Lu. Siamese network for object tracking with multi-granularity appearance representations. *Pattern Recognition*, page 108003, 2021b.

[135] Parnian Afshar, Farnoosh Naderkhani, Anastasia Oikonomou, Moezedin Javad Rafiee, Arash Mohammadi, and Konstantinos N Plataniotis. Mixcaps: A capsule network-based mixture of experts for lung nodule malignancy prediction. *Pattern Recognition*, 116:107942, 2021.

[136] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[137] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[138] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[139] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[140] Jie Xu, Wei Wang, Hanyuan Wang, and Jinhong Guo. Multi-model ensemble with rich spatial information for object detection. *Pattern Recognition*, 99:107098, 2020.

[141] Bo Li, Yuchao Dai, and Mingyi He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 83:328–339, 2018.

[142] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[143] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[144] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[145] Varun Jampani, Martin Kiefel, and Peter V Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4452–4461, 2016.

[146] Volker Aurich and Jörg Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Mustererkennung 1995*, pages 538–545. Springer, 1995.

[147] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Iccv*, volume 98, page 2, 1998.

[148] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[149] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1838–1847, 2018a.

[150] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[151] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012.

[152] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems*, pages 91–99, 2016.

[153] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016.

[154] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11166–11175, 2019.

[155] Jialin Wu, Dai Li, Yu Yang, Chandrajit Bajaj, and Xiangyang Ji. Dynamic sampling convolutional neural networks. *arXiv preprint arXiv:1803.07624*, 2018b.

[156] Evan Shelhamer, Dequan Wang, and Trevor Darrell. Blurring the line between structure and learning to optimize and adapt receptive fields. *arXiv preprint arXiv:1904.11487*, 2019.

[157] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[158] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590, 2017b.

[159] Hao Hu, Liqiang Wang, and Guo-Jun Qi. Learning to adaptively scale recurrent neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[160] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[161] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[162] 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.

[163] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[164] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[165] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[166] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[167] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *Advances in Neural Information Processing Systems*, pages 4261–4271, 2018.

[168] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.

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

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

[171] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

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

[173] 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, 2016b.

[174] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018b.

[175] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[176] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[177] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016a.

[178] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[179] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[180] Alireza Mehrtash, William M Wells, Clare M Tempany, Purang Abolmaesumi, and Tina Kapur. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE transactions on medical imaging*, 39(12):3868–3878, 2020.

[181] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[182] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[183] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[184] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016b.

[185] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021.