# Deep Learning with Noisy Supervision

Jiangchao Yao
Centre for Artificial Intelligence
University of Technology Sydney

A thesis submitted for the degree of
*Doctor of Philosophy*
April, 2019

# CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

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

This thesis is the result of a Collaborative Doctoral Research Degree program with Shanghai Jiao Tong University

Production Note:

Signature of Student: Signature removed prior to publication.

Date: June, 28th 2019

I would like to dedicate this thesis to my loving parents and family.

# Acknowledgements

I would like to take this good opportunity to appreciate my advisors, several professors, my colleagues, my friends and my family for their significant help during my doctoral study in University of Technology Sydney and Shanghai Jiao Tong University.

First, I am deeply indebted to my supervisor Prof. Ivor W. Tsang, who improves my machine learning taste into a new stage. During my research life in University of Technology Sydney, I am deeply impressed by his broad knowledge in machine learning and his pure altitude toward the research. His patient guides on how to propose a novel idea across or to bridge two different fields and how to generalize its impact to many tasks, really make an great effect on my deep understanding of the research philosophy. Besides, he constantly encourages us to focus on the fundamental problem and make the enough effort to purse the top research in the advance area, which moves me deeply in the journey of my research life. With his help, I found the happiness to do the pure research.

Second, I would like to express my foremost and deepest gratitude to my co-supervisor Prof. Ya Zhang, who plays an important role in showing me how to solve the difficulty in the doctoral life and how to do a significant academic research. In particular, at the beginning of my research career, her vision on the proposal of a novel idea in the research field and the endless labor on teaching me the writing of a regular academic paper helps me build the solid foundation of the research. Besides, her special interesting view on many works breaks through my original shallow sense and motivates the research going deep. It is hard to imagine I could finish this thesis without her high standards, unlimited patience, generous support and guidance.

I have been fortunate to attend the joint degree program between Shanghai Jiao Tong University and University of Technology Sydney for about two years. Wherein I would like to express my sincere appreciation to Prof. Ivor W. Tsang, Prof. Ya Zhang and Prof. Chengqi Zhang, who provides me this opportunity to work in such a great team and around so many brilliant minds. Within these two years, I have met and made

friends with many famous researchers in different areas and learned many useful experiences from them. In particularly, with the help of Bo Han, Yuangang Pan and Yueming lyu, I have broaden the scope of the whole research community and dived in more rigorous solution for the difficulty to be solved. With the guide of Gang Niu and Mingyuan Zhou, I realized what I lack for the top research in detail, which is very important in my development.

Then, it comes to my dear colleagues and friends in my doctoral study. I would like to thank Dr. Zhe Xu for his selfless help in my research, Jiawei Hu, Shuai Xiao, Weichang Wu, Xiaoyu Chen, Xiao Luo and Siyuan Zhou for being good friends, discussions and in their company, Zhiyi Tan and Fei Ye for being close friends and offering me the generous help, lovely juniors Jie Hou, Yichao Xiong, Zhiwei Rao, Xiaohui Zhu, Yan Wang, Yexun Zhang, Hongxiang Cai, Zhuoxiang Chen, Xinyan Yu, Zhonghao Wang, and young brothers Huangjie Zheng, Jiajie Wang, Xu Chen, Jie Chang, Conan Cui, Maosen Li, Hao Wu and many others. I am happy to experience the research life with you all and wish you all have a bright future. I am also grateful to all the other friends in Sydney: Bo Han, Donna Xu, Yuangang Pan, Zhuanghua Liu, Yueming Lyu, Muming Zhao, Xiaoqi Jiang, Yaxin Shi, Yan Zhang, Jing Chai, Weijie Chen, Tao Zheng, Xingrui Yu, Xiaowei Zhou, Jing Li, Yinghua Yao, Xiaofeng Xu, Xiaofeng Cao, Yurui Ming, Guang Li, Xinyuan Chen, Xuan Wang and Razia Osman for their support and company.

Finally, I would like to express my deeply felt gratitude to my parents, my sister, my wife and my daughter. You all always stand behind me and gives me the unconditional support. It is you contribute the happy lifetime to me so that I could purse my research goal without any concerns. This love will be remembered by me forever.

# Abstract

Central to many state-of-the-art classification systems via deep learning is sufficient accurate annotations for training. This is almost the bottleneck of all machine learning algorithms deployed with deep neural networks. The dilemma behind such a phenomenon is essentially the trade-off between the low expensive model design and the low expensive sample collection. For practical purposes to alleviate this issue, learning with noisy supervision is a critical solution in the Big Data era, since the noisily annotated data on the social websites and Amazon Mechanical Turk platforms can be easily acquired. Therefore, in this dissertation, we explore to solve the fundamental problems when training deep neural networks with noisy supervision.

Our first work is to introduce the low expensive noise structure information to overcome the decoupling bias issue existed learning with noise transition. We study the noise effect via a variable whose structure is implicitly aligned by the provided structure knowledge. Specifically, a Bayesian lower bound is deduced as the objective and it naturally degenerates to previous transition models in the case that there is no structure information available. Furthermore, a generative adversarial implementation is given to stably inject the structure information when training deep neural networks. The experimental results show the consistently improvement in the different simulated noises and the real-world scenario.

Our second work targets to substitute the previous ill-posed stochastic approximation to the noise transition with a rigorous stochastic re-allocation regarding the confusion matrix. This work discovers the reason that causes the unstable issue in modeling the noise effect by a neural Softmax layer and introduces a Latent Class-Conditional Noise model to overcome it. In addition, a computational effective dynamic label regression method is deduced for optimization, which stochastic trains the deep neural network and safeguards the noise transition estimation. The proposed method achieves the state-of-the-art results on two toy datasets and two large real-world datasets.

The last work aims to alleviate the difficulty that the ideal assumption on the accurate noise transition is usually not fulfilled and the noise could

still pollute the classifier in the back-propagation. We specially introduce a quality embedding factor to apportion the reasoning in the back-propagation, yielding a quality-augmented class-conditional noise model. On the network implementation, we elaborately design a contrastive-additive layer to infer the latent variable and deduce a stochastic optimization via reparameterization tricks. The results on a noisy web dataset and a noisy crowdsourcing dataset confirm the superiority of our model in the accuracy and interpretability.

# Contents

# List of Figures

# List of Tables

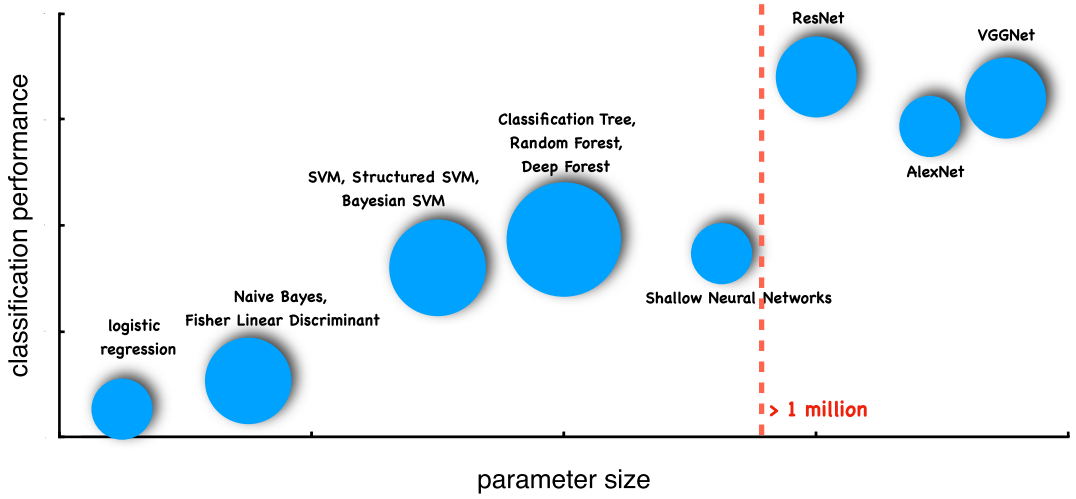
# Chapter 1

# Introduction

## 1.1 Motivation

Image classification, as one fundamental task in computer vision, has been investigated in a long term. The recent progress heavily benefits from the development of deep learning with the computation acceleration like algorithm differentiation [Abadi et al., 2016; Bergstra et al., 2011; Chetlur et al., 2014] and the network architecture evolution like ResNet [He et al., 2016; Huang et al., 2017; Krizhevsky et al., 2012; Sabour et al., 2017; Simonyan and Zisserman, 2014; Szegedy et al., 2015]. These advances make deep neural networks outperform many traditional models with the hand-crafted features. However, the side effect behind the state-of-the-art performance achieved by deep neural networks lies in introducing millons of neural parameters to characterize the feature extracting procedure. As reviewed in Figure 1.1(a), among the popular classification algorithms in the last two decades, the parameter size of the powerful deep neural networks, e.g., even the structure efficient ResNet [He et al., 2016], has been significantly larger than 1 million. To achieve the desired results and avoid overfitting, this large parameter space usually requires a large-scale accurately annotated dataset for training. However, as shown in Figure 1.1(b), collecting the data in such large volume is not an off-the-shelf procedure, which could consume the considerable human labor and expert labor. Especially when we dive into some more professional area like medical diagnostics [Miotto et al., 2017] or the fine-grained visual categorization [Xu et al., 2018], the expensive huge expert labor will be always a bottleneck. This practical needs raise new challenges to us and motivates the corresponding research to alleviate the data collection burden.

One popular proposal to overcome this dilemma is learning with noisy supervision, since it allows the supervision is not always accurate, which extremely saves the expensive expert labor. Simultaneously noisy image datasets, compared to the accurately annotated datasets, sometimes can be inexhaustibly acquired from social




(a)


(b)

Figure 1.1: (a) A simple review of some popular algorithms applied to image classification in the last two decades. We approximately place them into this axis according to the parameter size and the classification performance. (b) The procedure of building one standard image classification dataset.

crowdsourcing websites such as Flickr website[1], which may further save the human annotation labor. For example, the largest multimedia dataset *YFCC100M* [Thomee et al., 2016] contains a huge number of objects with noisy annotations in the daily life and guarantees diversity, granularity and free license for both research and the industrial applications. As demonstrated in [Joulin et al., 2016], such a large-scale dataset, even with noisy annotations, could greatly boost the performance of many vision problems. Formally, deep learning with noisy supervision is a framework that casts the reduce of the accurate annotation burden as a mathematical problem, which incorporates the noise modeling or removal along with training the classifier. In this

[1]https://www.flickr.com/

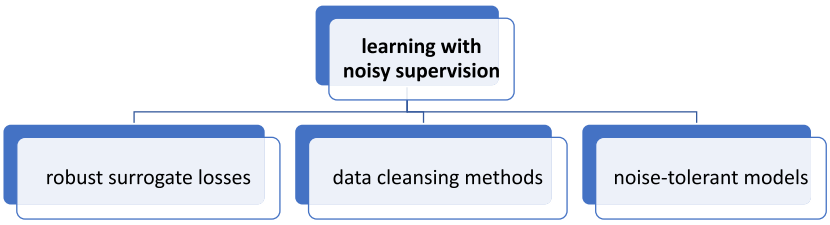



Figure 1.2: The algorithmic tree to depict the traditional methods for learning with noisy supervision.

dissertation, our goal is solving a series of current issues in deep learning with noisy supervision to make a contribution on the progress of applying this technique into practises.

## 1.2 Learning with Noisy Supervision

Learning with noisy supervision is an old problem that can be traced back to three decades ago [Angluin and Laird, 1988]. Hitherto, there are a lot of works [Brodley and Friedl, 1999; Hickey, 1996; Kalai and Servedio, 2005; Malossini et al., 2006; Manwani and Sastry, 2013; Natarajan et al., 2013; Raykar et al., 2010; Sastry et al., 2010; Smyth et al., 1995] from machine learning community that have well investigated this task, including the definitions, the sources and the taxonomy of label noise, the potential consequence and the corresponding methods [Frenay and Verleysen, 2014]. Previous approaches to deal with noisy supervision can be categorized into three families, *robust surrogate losses*, *data cleansing algorithms* and *noise-tolerant models*, which is structured in Figure 1.2. In the following, we gives simple concrete analysis on them.

The first family of methods are to design some surrogate losses that are robust under the empirical risk minimization framework in the presence of the label noise [Beigman and Klebanov, 2009; Manwani and Sastry, 2013; Sastry et al., 2010]. However, the surrogate losses may introduce additional ideal noise-related parameters, e.g., noise rate parameters [Liu and Tao, 2016; Natarajan et al., 2013], which resorts this difficulty to the hyperparameter estimation. Besides, experimental results in the literature show that such surrogate losses seem to be adequate only for simple cases of the label noise and cannot sufficiently prevent the degeneration of classifier [Frenay and Verleysen, 2014]. Therefore, a better choice is to combine the surrogate losses with the other approaches in learning with noisy supervision.

The second family of methods are to cleanse the dataset before using to the training, which mainly include ad-hoc thresholding measures and model prediction-based methods. The former way will set up a heuristic measure, e.g., the model

complexity measure [Gamberger et al., 1996], to remove the samples exceeding the threshold. The latter diverges in the strategies to leverage the prediction of a pre-trained classifier to remove outliers, e.g., classification filtering [Gamberger et al., 1999], voting filtering [Brodley and Friedl, 1999] and partition filtering [Zhu et al., 2003]. Although the label cleansing method is a conservative and safe way to choose a subset, it sometimes removes too many instances and could be an issue in the case of the data scarcity and the imbalance dataset.

The last family of methods explicitly build the classifier together with modeling the label noise, so that the whole data fitting is noise-tolerant. This refers to decoupling the generation process of the noisy data with a generation process of the clean data and a generation process of disturbing clean data [Frenay and Verleysen, 2014]. Generally, most of the proposed methods are probabilistic, which models noise transition from Beta distribution[Daniel Paulino et al., 2003; Gaba, 1993; Gustafson et al., 2001; Rekaya et al., 2001; Winkler et al., 1985; Zhang et al., 2005] or Dirichlet distribution [Liu et al., 2009; Ruiz et al., 2008] on top of the classifier and optimizes by the EM algorithm [Raykar et al., 2010]. While the probabilistic models in this family are a more theoretical way than above two families, the main disadvantage is the increasing complexity and the optimization burden.

## 1.3 Deep Learning with Noisy Supervision

Recently, deep learning as a revolutionary paradigm has improved several machine learning areas to a new stage on the performance of the real-world applications. Specifically, the advances in the last ten years make it more efficient from Alexnet Hinton et al. [2012] to ResNet He et al. [2016], InceptionNet Szegedy et al. [2015] and training a deep learning model is not hard any more to the researchers in NLP, Computer Vision and Speech Recognition with a convenient tool like Tensorflow Abadi et al. [2016]. Nevertheless, the following new challenges raise when applying deep learning methods to learning with noisy supervision, even plenty of approaches regarding learning with noisy supervision have been developed in the last few decades.

1. The first challenge is *the large capacity* of deep neural networks that makes the classifier can memorize any kinds of noise involved in the noisy dataset [Zhang et al., 2016].

2. The second challenge is the *the large parameter space* of deep neural networks that requires the efficient optimization algorithms to train the classifier [Schneider et al., 2018].

The large capacity leads to deep neural networks as the classifier treats the noisy dataset like the clean dataset and absorbs the noise among the dataset almost without rejection.

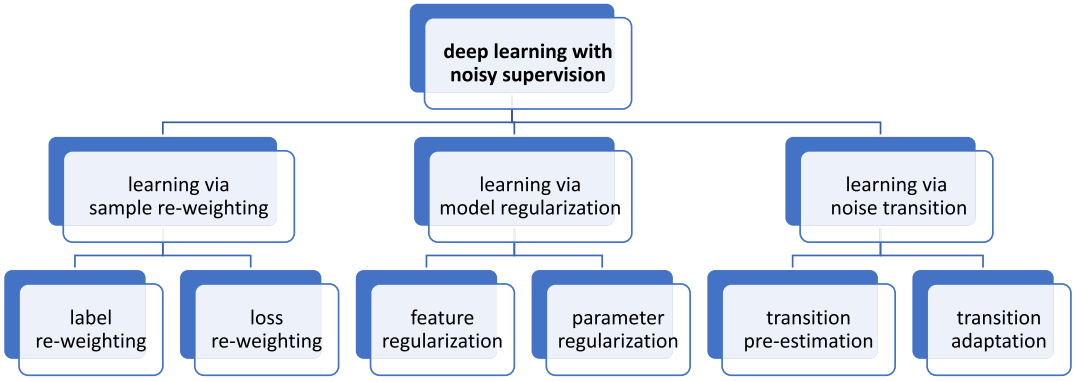



Figure 1.3: The algorithmic tree to depict the methods for deep learning with noisy supervision.

In this case, the robust surrogate losses will not make an effect since the classifier can memorize anything compared to the traditional shallow classifier, e.g., SVM [Suykens and Vandewalle, 1999]. The label cleansing methods similarly fail since the feature topology is not distinguishable after the multiple-layer mapping. As for the large parameter space, it leads to the previous models with complex optimization algorithms are unacceptable on the training time. This excludes the noise-tolerant models based on the expensive EM optimization algorithm.

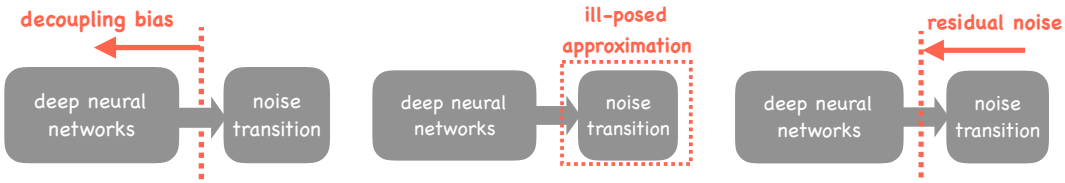
In spite of new challenges, there are still several methods explored for deep learning with noisy supervision. Figure 1.3 summarizes the recent progress in this direction, which mainly has three methodologies, i.e., learning via *sample re-weighting*, learning via *model regularization* and learning via *noise transition*. The first one is to selectively weight the contribution of samples to the training of the classifier, which therein filters the influence of the label noise. Nevertheless, the core drawback is the weights to be learned are related to the sample number, yielding the scalability problem [Jiang et al., 2018] to the very large datasets applied for deep learning. While the collaborative way can alleviate this issue, it usually consumes the auxiliary clean dataset to calibrate the optimization direction [Veit et al., 2017] or lack of the rigorous interpretability [Han et al., 2018]. As for the second methodology, learning via model regularization, it origins from the heuristic conjecture on the memorization order of feature patterns and critically depends on the human experiences to tune the parameter or the learning procedure in different datasets [Arpit et al., 2017; Jindal et al., 2016]. Besides, it is lack of theoretical analysis and interpretation on the noise level and the noise type. Compared to above two methodologies, learning via noise transition is the most promising one in terms of the data scalability, optimization efficiency and theoretical interpretation (Figure 1.4). There are also a lot of research works [Natarajan et al.,


Figure 1.4: The methodological advantage of learning via noise transition in dealing with noisy data. The current methods of the other two methodologies do not possess such three merits simultaneously.

2013, 2017; Raykar et al., 2010; Reid and Williamson, 2010; Vernet et al., 2011] from the machine learning communities can be referred since it is tightly related to the noise-tolerant models. Therefore, we will follow the pace of learning via noise transition in this dissertation and overcome its current shortcomings to improve the performance of deep learning with noisy supervision.

Specifically, as shown in Figure 1.5(a), we first consider the *decoupling bias* issue in learning via noise transition due to the large capacity of deep neural networks. As discussed above, deep neural networks are flexible enough to memorize any kinds of noise after the sufficient training. This cause that the optimization in learning with noise transition unilaterally encourages deep neural networks to memorize all the noisy training samples instead of reasoning the noise to the noise transition. Correspondingly, the noise transition will then degenerate to other undesired minimums, e.g., a meaningless diagonal matrix. One solution is collecting an approximate infinite dataset to squeeze out the noise statistic from deep neural networks and make the noise transition sufficiently represent the underlying disturbance procedure. However, it is usually quite an impractical choice considering the limited storage and computational resources. To better overcome this issue, we introduce the low expensive *structure prior* about the noise transition to the optimization procedure, which is like the reverse action force of the decoupling bias and as much as possible prevents the degeneration induced by the decoupling bias.

Second, we find the stochastic approximation to the noise transition via a Softmax laye [Goldberger and Ben-Reuven, 2017] is *ill-posed* (Figure 1.5(b)). The performance acquired by this method critically depends on the highly tweaking and the model parameters easily get stuck into undesired minimums. By analysis, this is because the training by the standard back-propagation makes the noise transition lack of the global dependency on the whole dataset, and any local mini-batch of samples can unbounded update its parameter. To overcome this problem, we reformulate Class-Conditional Noise model as a *Latent Class-Conditional Noise* model, which treats the noise transition as a global variable to emphasize this dependency. Then, combined with Gibbs sampling, we deduce a new stochastic approximation to the noise transition




(a) (b) (c)

Figure 1.5: The three issues in current learning via noise transition. (a) The decoupling bias issue induced by the large capacity of deep neural networks. (b) The ill-posed approximation issue via the neural Softmax layer. (c) The residual noise effect due to the inaccurate learning of noise transition.

and importantly, the deep neural network is trained in a stochastic way.

Finally, we consider when the accurate noise transition is not closely reached after optimization, what the issue is to the current framework by theory? This is useful for further improvement since the real-world noise could be more complex compared to the popular class-conditional noise. To answer this question, we theoretically deduce that the minimizer of learning with noise transition in this case is equal to the minimizer of the classifier trained on a new noisy dataset disturbed by a residual transition matrix from the clean dataset. That is to say, there is a part of *residual noise* that has not absorbed by the noise transition and has to be memorized by the classifier. Based on this analysis, we propose to a *composite reasoning* way to better apportion the noise effect in learning with noise transition. Specifically, we introduce a quality factor parallel to the latent true label to absorb the noise effect in the original model and propose a contrastive-addtive neural network to implement this new model. Extensive experiments on two real-world noisy datasets demonstrate the superiority of our new structure. In Figure 1.6, we illustrate our research view to three issues and their relationship to be a summary.

## 1.4 Contributions and Organizations

This thesis investigates deep learning with noisy supervision by following the methodology of learning via noise transition. We consider different scenarios to overcome a series of issues existed in the current framework and acquire the constant improvements on the performance. The main contributions of this dissertation can be summarized in the following three perspectives.

- **Theoretical contributions.** By conducting deep learning with noisy supervision, we present a thorough analysis on how to save the huge human labor on

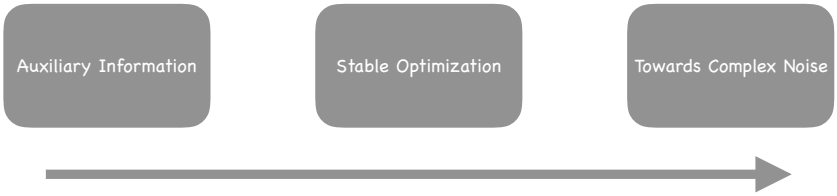



Figure 1.6: The research view of our thesis as well as the relationship of three works. The first one focuses on the scenario where auxiliary information is available; the second one targets to make the training more stable and the last one moves the ideal class-conditional noise to the more complex noise and investigates how to deal with the residual noise effect. Such three works are respectively applicable to different scenarios and also combinable when the corresponding assumption meets.

the accurate annotation of large-scale datasets for deep learning, and the methods to leverage the low expensive noisy datasets to train a robust classifier.

- **Algorithmic contributions.** By analyzing a series of issues existed in the current methodology of learning via noise transition, we propose three orthogonal algorithms to point-to-pointly overcome the decoupling bias issue, the ill-posed approximation issue and the residual noise issue of previous works and achieve the consistent improvements in the model performance.

- **Applicational contributions.** The proposed methods have been evaluated in the widely used benchmarks including toy datasets and real-world datasets, and have shown impressive results. Although we only apply our methods to the classification scenarios, more diverse applications including image detection, image segmentation and other areas can be extended with the moderate modification.

The structure of the remaining chapters in this thesis is organized as follows.

- Chapter 2 briefly reviews the related work of deep learning with noisy supervision, the theory of risk minimization to analyze the classifier and some techniques used in the optimization of deep learning including variational inference and Monte Carlo sampling, which will be used in this dissertation.

- Chapter 3 proposes to utilize the structure information to overcome the decoupling bias issue in learning with noise transition. A structure-aware probabilistic model has been derived to incorporate the structure prior with a GAN-like optimization.

- Chapter 4 targets to improve the previous ill-posed stochastic approximation to the noise transition by extending the previous class-conditional noise model with

a latent class-conditional noise model. A dynamic label regression approach is deduced to guarantee the stochastic training of deep neural networks and simultaneously safeguard the transition update of the noise transition.

- Chapter 5 analyzes the residual noise effect when the transition is not sufficient to capture noise and proposes a composite reasoning way to solve this issue. Specifically, a quality augmented class-conditional noise model have been presented to absorb the residual noise influence parallel to the training of the classifier.

- Chapter 6 concludes this thesis and discusses some possible future directions.

## 1.5 Publications during PhD Study

1. JIANGCHAO YAO, YANFENG WANG, YA ZHANG, JUN SUN, JUN ZHOU. Joint Latent Dirichlet Allocation for Social Tags, IEEE Transactions on Multimedia (TMM), 20(1): 224 - 237, 2018.

2. JIANGCHAO YAO, JIAJIE WANG, IVOR W. TSANG, YA ZHANG, JUN SUN, CHENGQI ZHANG, RUI ZHANG. Deep Learning from Noisy Labels with Quality Embedding, IEEE Transactions on Image Processing (TIP), 28(4): 1909-1922, 2019.

3. JIANGCHAO YAO, YA ZHANG, IVOR W. TSANG, JUN SUN. Safeguarded Dynamic Label Regression for Noisy Supervision, IEEE Transactions on Image Processing (TIP), 2019. (under review)

4. JIANGCHAO YAO, HAO WU, YA ZHANG, IVOR W. TSANG, JUN SUN. Safeguarded Dynamic Label Regression for Noisy Supervision, Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI), 2019.

5. HUANGJIE ZHENG, JIANGCHAO YAO, YA ZHANG, IVOR W. TSANG, JIA WANG. Understanding VAEs in Fisher-Shannon Plane, Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI), 2019.

6. KENAN CUI, XU CHEN, JIANGCHAO YAO, YA ZHANG. Variational Collaborative Learning for User Probabilistic Representation, Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence workshops (AAAI workshops), 2019.

7. BO HAN*, JIANGCHAO YAO*, GANG NIU, MINGYUAN ZHOU, IVOR W. TSANG, YA ZHANG, MASASHI SUGIYAMA. Masking: A New Perspective of Noisy Supervision, Advances In Neural Information Processing Systems (NeurIPS), 2018. (* means equal contribution)

8. JIAJIE WANG, JIANGCHAO YAO, YA ZHANG, RUI ZHANG. Collaborative Learning for Weakly Supervision Object Detection, International Joint Conference on Artificial Intelligence (IJCAI), 2018.

9. JIANGCHAO YAO, YA ZHANG, IVOR W. TSANG, JUN SUN. Discovering User Interests from Social Images, International Conference on Multimedia Modeling (MMM), 2017.

10. HUANGJIE ZHENG, JIANGCHAO YAO, YA ZHANG. Describing Geographical Characteristics from Social Images, International Conference on Multimedia Modeling (MMM), 2017.

11. ZHIWEI RAO, JIANGCHAO YAO, YA ZHANG, RUI ZHANG. Preference Aware Recommendation Based on Categorical Information, International Conference on Machine Learning and Applications (ICMLA), 2016.

12. XIAOYU CHEN, JIANGCHAO YAO, YANFENG WANG, YA ZHANG. Online Learning Algorithm for Collective LDA, International Conference on Machine Learning and Applications (ICMLA), 2016.

13. JIANGCHAO YAO, YA ZHANG, ZHE XU, JUN SUN, JUN ZHOU, XIAO GU. Joint Latent Dirichlet Allocation for Non-IID Social Tags, International Conference on Multimedia and Expo (ICME), 2015.

# Chapter 2

# Background

As discussed in Chapter 1.1, the main goal of this thesis is by investigating deep learning with noisy supervision to save human labor on accurate annotations. This chapter will simply survey the large image sources with noisy supervision for deep learning in the recent years. Then, we provide a comprehensive overview of existing methods from multiple methodologies including sample re-weighting, model regularization and noise transition. Meanwhile, we will enclose a brief review about the advances in optimization and theoretical analysis that is useful to deep learning with noisy supervision.

## 2.1 Large Image Sources with Noisy Supervision

With the rapid development of deep learning in the recent years, a great achievement has been made in computer vision [Krizhevsky et al., 2012], natural language processing [Sutskever et al., 2014] and speech recognition [Hinton et al., 2012]. One important factor behind these success could attribute to the large scale datasets, e.g., ImageNet [Deng et al., 2009] for image recognition, which usually consume considerable human labor on the accurate annotations. To alleviate this cost, several noisy datasets [Bossard et al., 2014; Goeau et al., 2017; Krause et al., 2016; Kuznetsova et al., 2018; Li et al., 2017a; Thomee et al., 2016; Xiao et al., 2015] have been proposed as the low expensive alternatives to the accurately annotated datasets. We simply survey these datasets in computer vision as follows.

- **Food101** [Bossard et al., 2014] is a popular food dataset that leverages computer vision to serve our daily life. It contains the common food categories that are collected from the social website $foodspotting.com$. The challenge of this dataset lies in that the noise may reside among the fine-grained classes, which will confuse the subtle differences between food. Therefore, how we design a

good method to distinguish the noise among fine-grained categories is critical to improve the performance.

- **Clothing1M** [Xiao et al., 2015] is another well studied dataset about several kinds of clothes, which is crawled from many online shopping websites. As discussed in [Xiao et al., 2015], the overall accuracy of this dataset is approximate $61\%$ and some pairs of classes are very confusing with each other (e.g. Knitwear and Sweater), even counterfactual. Thus, the labels of images are not so reliable that challenges current algorithms in the area of deep learning with noisy supervision.

- **YFCC100M** [Thomee et al., 2016] is, up to now, the largest public multimedia dataset that includes photos and videos, harvested from the *Flickr* website. It overcomes a lot of issues in current datasets like modalities, metadata, licensing and principally volume, and guarantees accessible and intact for years to come. The broad range of attributes and various domains encourage us to develop diverse side-information-aware and domain-aware methods to deep learning with noisy supervision.

- **Goldfinch** [Krause et al., 2016] is a dataset for fine-grained recognition challenges. It contains a list of bird, aircraft, Lepidoptera and dog categories with relevant Google image search and Flickr search URLs. Compared to the Food101, the challenge to utilize this dataset is we should not only consider the cross-category noise but also the cross-domain noise, which is a harder task to the noisy fine-grained classification.

- **PlantCLEF** [Goeau et al., 2017] is proposed in the 2017 LifeCLEF Plant identification challenge. In addition to the small part from national botany institutes, a much larger part of this dataset is from botanist blogs, plant lovers web-pages, image hosting websites and on-line plant retailers with a high degree of noise. Deep learning with such a dataset is an useful and important explore to leverage computer to automatically identify and protect the plants in Europe and North America.

- **WebVision** [Li et al., 2017a] is a large web image dataset containing the object category in ImageNet [Deng et al., 2009] to benefit the generalized visual recognition. Especially, it exhibits the comparable or even better generalization ability than the one trained from the ILSVRC 2012 [Deng et al., 2009] dataset when being transferred to new tasks. This opens the explore to the potential of deep learning in the presence of inaccurate supervision.

Table 2.1 summarizes the large image sources with noisy supervision proposed in the recent years and Figure 2.1 illustrates some examples for summarized noisy image

| Dataset | Year | #Category | #Sample | #Label | Granularity | Test |
|---|---|---|---|---|---|---|
| Food101 | 2014 | 101 | 101K | single | micro | ✓ |
| Clothing1M | 2015 | 13 | 1M | single | middle | ✓ |
| YFCC100M | 2016 | >1M | 100M | multiple | macro | × |
| Goldfinch | 2016 | 26,458 | 9.8M | single | micro | ✓ |
| PlantCLEF | 2017 | 10,000 | 1.1M | single | middle | ✓ |
| WebVision | 2017 | 1,000 | 2.4M | single | macro | × |

Table 2.1: A summary of the popular real-world datasets in the area of deep learning with noisy supervision. From left to right, we respectively list the dataset name, the release year, the category number, the sample number, the label number for each sample and the inter-class granularity of the dataset as well as whether there is a open test set available provided by the release organization.

datasets. These datasets are explored in the respective sub-area of deep learning based on different methods to deal with label noise, which promotes to save human labors.

## 2.2 Deep Learning with Noisy Supervision

Deep learning with noisy supervision, as an important direction to generalize deep learning to more practical applications, is receiving increasing attention. The goal of this task is to learn a classifier from the noisy training dataset such that a good performance can be achieved in the clean test dataset. To implement this, several approaches have been developed respectively in the perspective of sample re-weighting [Jiang et al., 2018; Liu and Tao, 2016; Ma et al., 2018; Reed et al., 2014; Ren et al., 2018], model regularization [Arpit et al., 2017; Azadi et al., 2015; Zhang et al., 2016, 2017] and noise transition [Goldberger and Ben-Reuven, 2017; Misra et al., 2016b; Patrini et al., 2017; Sukhbaatar et al., 2014; Xiao et al., 2015]. In the following, we will respectively review these methods of three methodologies.

### 2.2.1 Learning via Sample Re-weighting

As shown in Figure 2.2, this methodology is by means of a weighting model to dynamically adjust the contribution of each sample to the classifier, which thereby filters the label noise. The current works could be divided into two categories, the label re-weighting and the loss re-weighting. The former focuses on utilizing the classifier itself or the auxiliary knowledge to re-weight the noisy labels to be more reliable supervision. For example, Reed *et al.* [Reed et al., 2014] facilitated the notion of perceptual consistency to linearly combine the label and the prediction as the new supervision, which shows the substantial robustness to label noise. Then, Li

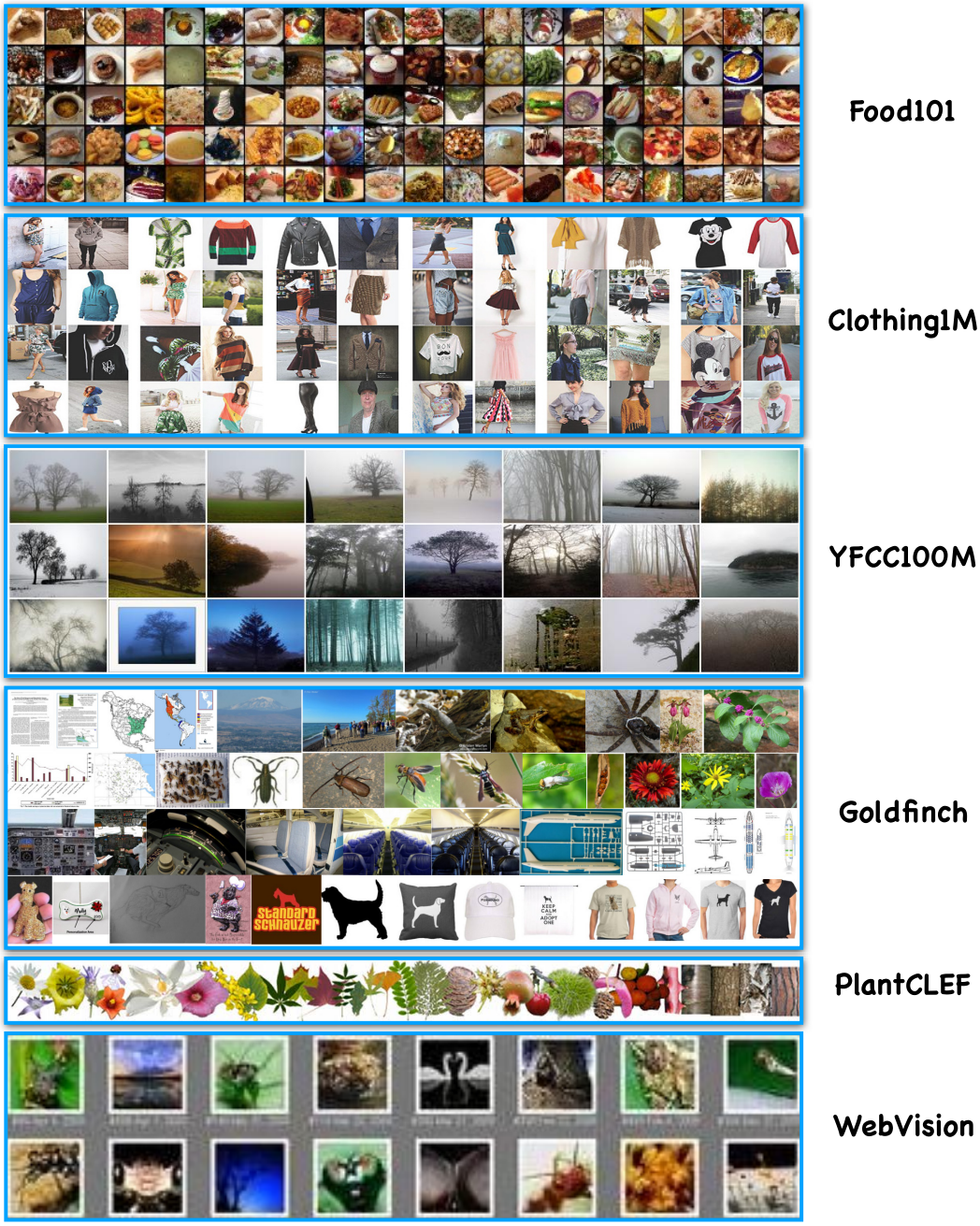



Figure 2.1: The illustration of six large popular real-world image datasets with noisy supervision.

*et al.* [Li et al., 2017b] substituted the prediction with the refined label by the graph distillation. This further avoids the influence of the output degeneration induced by the

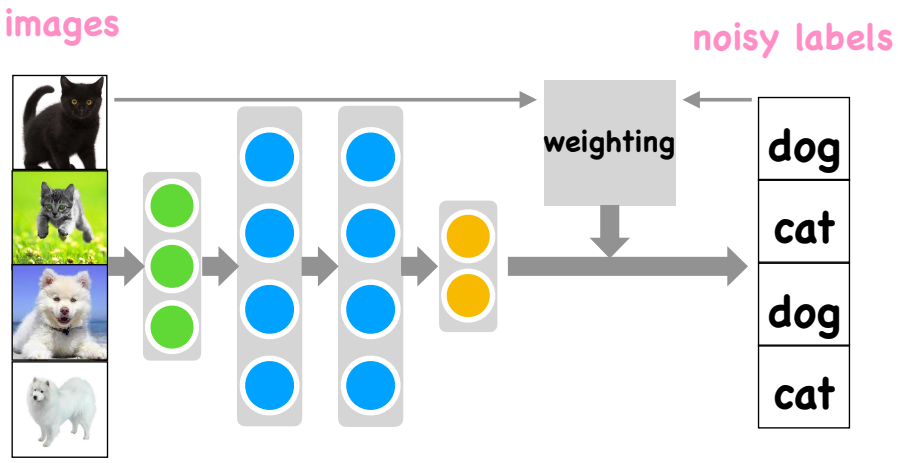



Figure 2.2: The illustration of learning via sample re-weighting

error accumulation from the self-paced training. Ma *et al.* [Ma et al., 2018] moved the focus on the manually tuned weight in Bootstrapping [Reed et al., 2014] and leveraged the local intrinsic dimensionality theory to design an automatic learning strategy.

As for the loss re-weighting, the main intuition is to learn a weight for the global dataset or for each training pair. For instance, Joulin *et al.* [Joulin et al., 2015] weighted the cross-entropy loss with the sample number to balance the emphasis of noise in positive and negative instances. Izadinia *et al.* [Izadinia et al., 2015] estimated a global ratio of positive samples to weaken the supervision in the loss function. However, one weight parameter cannot essentially reason the noise and thus cannot prevent the noise pollution. MonterNet [Jiang et al., 2018] learns a data-driven curriculum for StudentNet (the classifier we target to) so that it focuses on the samples that are correctly annotated. Nevertheless, the weight for each training pair leads to an expensive optimization burden since it cannot be learned in a purely stochastic way. Han [Han et al., 2018; Song and Chai, 2018] explored to use the dual classifiers to collaboratively learn a weight or selection to the loss of each training pair, which alleviates the bottleneck in MentorNet. However, these methods critically depend on the elaborate re-weighting strategy to guarantee the convergence towards at our expectation.

### 2.2.2 Learning via Model Regularization

This type of methods attempt to heuristically introduce the explicit or implicit constraint on the parameter space or the feature space to weaken the noise pollution to the classifier. For example, previous works [Azadi et al., 2015; Zhang et al., 2017] respectively introduce different regularization techniques in the feature space

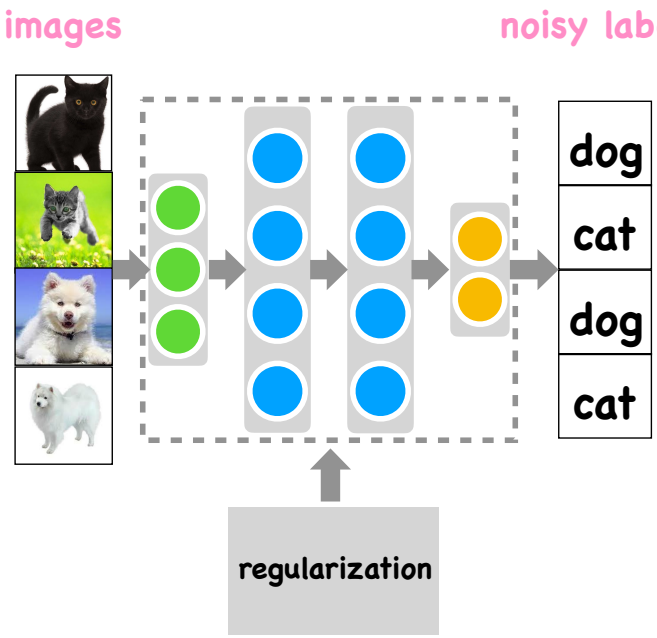



Figure 2.3: The illustration of learning via model regularization.

to prevent the classifier from overfitting on the noise. Zhang *et al.* [Zhang et al., 2016] have shown DNNs can easily memorize the random labels completely, characterizing the challenge to deep learning with noisy labels. Then, Arpit *et al.* [Arpit et al., 2017] investigated the memorization effect of deep neural networks on feature patterns and proposed that deep neural networks first memorized the simple patterns and then the complex patterns which the label noise usually belongs to. Based on this conjecture, they demonstrated both the dropout regularization on the feature space and the $\ell$-norm regularization on the parameter space can efficiently limit the speed of memorization of the classifier. [Tanaka et al., 2018] explicitly introduced a regularization term to prevent the trivial case of assigning all labels to a single class in label correction. To be clear, Figure 2.3 simply depicts the idea of learning via model regularization. Compared to the other two methodologies, this branch of methods are lack of rigorous theoretical support and the training is not guaranteed, while sometimes simple and efficient.

### 2.2.3 Learning via Noise Transition

This branch of research models a noise transition on top of the deep neural network to alleviate the degeneration from label noise, which is illustrated in Figure 2.4. In the early work [Mnih and Hinton, 2012], Mnih *et al.* proposed a latent variable model on aerial images, which first models the symmetric noise along with deep

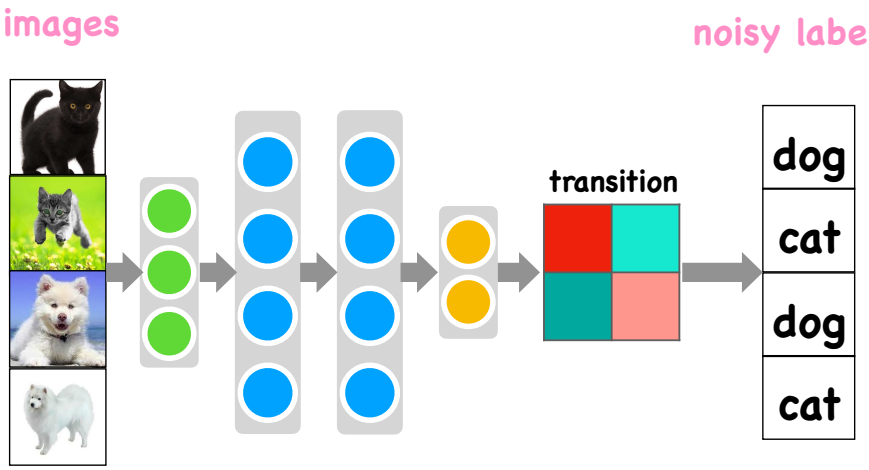



Figure 2.4: The illustration of learning via noise transition

neural networks. Based on it, several works [Jindal et al., 2016; Sukhbaatar et al., 2014] use an linear adaptation layer to model the asymmetric label noise, and add the layer on top of a deep neural network. This transition layer can be deemed as the confusion matrix representing label flip probability. However, the matrix only depends on the distribution of labels but ignore the information of image contents. Chen *et al.* [Chen and Gupta, 2015] applied a two-stage approach to model the latent label and learned the translation to the noisy label, in which a clean dataset was used. Different from methods that model label transition in the dataset level, Xiao *et al.* [Xiao et al., 2015] proposed a probabilistic graphic model that disturbed the label in the image level. However, the model also needs a small part of clean data to learn conditional probability, which may constrains the generalization of the model. To demonstrate the human-centric noisy label exhibits specific structure that can be modeled, Misra *et al.* [Misra et al., 2016a] built two parallel classifiers. One classifier deals with image recognition and the other classifier models human's reporting bias. An important milestone, Patrini *et al.* [Patrini et al., 2017] adopted a two-step solution to pre-estimate the noise transition and then, fixed it to train the classifier. They demonstrated that if the noise transition is accurately estimated, their forward correction method would make the training on the noisy dataset share the same minimizer with that on the clean dataset. Subsequent improvement [Goldberger and Ben-Reuven, 2017] reformulated the noise transition as a Softmax layer and tuned its parameters in the training to purse a better refinement. From the perspective of the generative process, above methods inherit the idea of the *Class-Conditional Noise* (CCN) model [Natarajan et al., 2013], which explicitly models the noise disturbance as the transition from the true label to the noisy label. And most of these works are mainly different from the optimization approaches about the noise transition. Although better performance has been achieved,

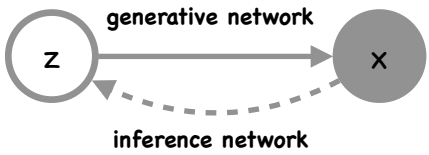



Figure 2.5: Variational auto-encoder. $x$ is the observed samples and $z$ is the hidden variable. The solid line and the dash line respectively represent the generative process and the inference process.

these methods still suffer from some unsolve issues, e.g., decoupling bias and ill-posed stochastic approximation. To exploit the great value of noisy labels, we still need a long way to go and in this thesis, we will continue the explore of this direction.

## 2.3 Advances in Optimization and Theoretical Analysis

### 2.3.1 Variational Inference

Many modern probabilistic models that are combined with deep learning are usually intractable and thus require approximate inference [Zhang et al., 2018a]. Variational inference [Jordan et al., 1999], which casts the posterior estimation as an optimization problem, has been successfully used to solve this type of problems [Blei et al., 2003, 2006, 2012; Ghahramani and Beal, 2000; Hoffman et al., 2010; Wainwright et al., 2008] and also makes a great progress along with deep learning in the recent years [Blei et al., 2017; Higgins et al., 2016; Kingma and Welling, 2014; Mnih and Gregor, 2014; Tran et al., 2017a]. Since we will leverage this technique to solve our probabilistic models in the thesis, a simple review is conducted here.

Variational auto-encoder [Kingma and Welling, 2014] is a represenative method that marries the probabilistic model with deep learning. Similar to the vanilla variational inference, it introduces a variational distribution to approximate the posterior distribution of the latent variable and deduce a *Evidence Lower Bound* (ELBO) via Jensen's Inequality to optimize. The main difference from the vanilla variational inference is both the inference and the generation are parameterized by the inference network and the generative network, which is shown in Figure 2.5. And due to the non-analytical objective, it is encouraged to be optimized via the stochastic gradient plus variance reduction, e.g., reparameterization [Kingma and Welling, 2014; Rezende et al., 2014].

The discrete version of variational auto-encoder [Jang et al., 2017] related to the discrete latent variable model is also developed as it is widely used in the real-world

scenarios e.g., the discrete latent true labels utilized in our models. However, the challenges always remain since the discrete latent variable is not reparameteriable and it is hard to find a suitable control variate[1]. Fortunately, there are a lot of trials on the biased approximate discrete reparameterization [Jang et al., 2017; Maddison et al., 2017], unbiased estimators via neural control variates [Gu et al., 2015; Mnih and Gregor, 2014; Tucker et al., 2017] or collapsed surroages [Rolfe, 2016; Vahdat et al., 2018a,b], which greatly improve the optimization performance.

Other types of variational inference including flow-based variational inference [Dinh et al., 2014, 2016; Kingma et al., 2016], Stein variational inference [Liu, 2017; Liu and Wang, 2016; Ranganath et al., 2016] and implicit variational inference [Shi et al., 2017; Srivastava et al., 2017; Tran et al., 2017b] are all explored in the different sub-directions, which more or less have merits and drawbacks. More details about advances of the development can be found in the survey [Zhang et al., 2018a].

### 2.3.2 Monte Carlo Sampling

Monte Carlo sampling [Levin and Peres, 2017] is another type of methodologies to do approximate inference. Recently, it usually couples with variational inference to optimize deep generative models like variational auto-encoder [Kingma and Welling, 2014], but has the independent improvement in the area of reinforcement learning [Choromanski et al., 2018; Mania et al., 2018; Salimans et al., 2017] and random feature approximations [Rahimi and Recht, 2008; Yu et al., 2016]. Here, we briefly review the special Gibbs sampling [Levin and Peres, 2017] and beyond [Russo et al., 2018].

As shown in Figure 2.6, Gibbs sampling is a Markov chain Monte Carlo algorithm which iterates the sampling among dimensions to sequentially acquire the samples. It can apply to a brand range of machine learning models, e.g., LDA [Blei et al., 2003] and is usually still computational viable even in the complex scenarios. The only one limitation of this technique is there must be the conditional distribution about the sampling dimensions available. In this thesis, we will show how to apply Gibbs sampling to the probabilistic model for deep learning with noisy supervision. Alternative to Gibbs sampling, Laplace approximation [Ritter et al., 2018], Langevin Monte Carlo [Nado et al., 2018] and Bootstrapping [Hjorth, 2017] are all very useful in the approximate inference and have been developed for deep learning [Russo et al., 2018].

[1] https://en.wikipedia.org/wiki/Control_variates

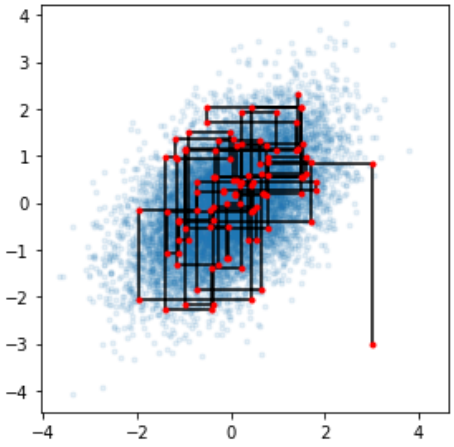



Figure 2.6: The visualization of a 2-D Gibbs sampling for a 2-D Gaussian distribution. The instances are sequentially sampled respectively along x-axis and y-axis in each iteration, which approximates the desired samples from the ideal 2-D Gaussian distribution.

### 2.3.3 Generalization Bound

Deep learning with noisy supervision can be cast as a problem of domain adaptation [Daume III and Marcu, 2006], where the deep neural network is trained in the noisy data domain and then test in the clean data domain. Different from the conventional domain adaptation on the input feature [Long et al., 2013, 2014, 2015, 2016], the domain inconsistency mainly come from the distribution of labels conditioned on the image feature. However, the theory of this area such as the important generalization bound [Bousquet et al., 2004] can still be used for deep learning with noisy supervision. For example, in the early works, a series of bounds [Blitzer et al., 2008; Mansour et al., 2009; Zhang et al., 2012] have been proposed to analyze domain adaptation in different cases, e.g., multiple sources or a part of data from the target source available. Concretely, Blitzer *et al.* [Blitzer et al., 2008] introduced a uniform convergence bound on the true target risk of a model trained to minimize a convex combination of empirical source and target risks, which described an intuitive tradeoff between the quantity of the source data and the accuracy of the target data. Mansour *et al.* [Mansour et al., 2009] proposed a novel distance between distributions, discrepancy distance, which was tailored to adaptation problems with arbitrary loss functions and gave Rademacher complexity bounds for estimating the discrepancy distance from finite samples for different loss functions. With this distance, they derived novel generalization bounds for domain adaptation for a wide family of loss functions. Zhang *et al.* [Zhang et al., 2012] considered domain adaptation with multiple sources and domain adaptation with the combination

of source data and test data, and deduced the corresponding generalization bound under the uniform entropy number. In this thesis, we will apply these theory to analyze deep learning with noisy supervision.

# Chapter 3

# Mask the Decoupling Bias

As discussed in Chapter 1, it is a practical need to learn deep neural classifiers under images with noisy labels. Noisy labels, in the most popular noise model hitherto, are corrupted from ground-truth labels by an unknown *noise transition*. Thus, by estimating and leveraging this transition, we can make the classifiers escape from overfitting those noisy supervision. However, due to the large capacity, deep neural networks are capable enough to memorize almost the whole noisy image datasets as the clean dataset. This yields the decoupling bias in the estimation of noise transition based on either two-step approaches or end-to-end approaches under limited data, which subsequently degenerates the classification performance of the deep neural network in the clean test dataset.

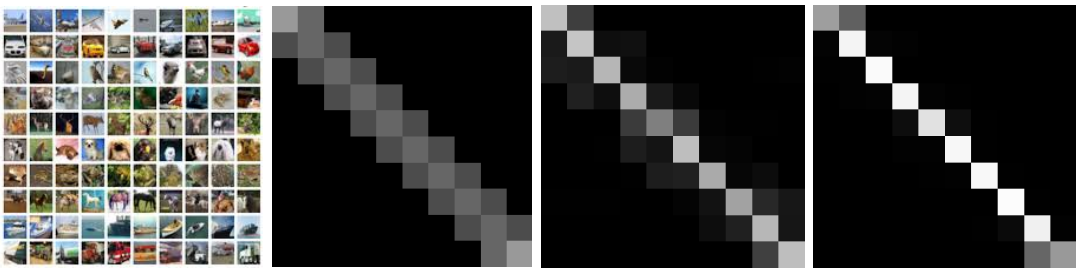
To overcome this problem, in this Chapter, we propose a human-assisted approach called "*Masking*" that conveys human cognition of *invalid class transitions* and naturally speculates the structure of the noise transition matrix. To this end, we derive a *structure-aware probabilistic model* incorporating a structure prior, and solve the challenges from structure extraction and structure alignment. Thanks to Masking, we only need to estimate unmasked noise transition probabilities and the burden of estimation is tremendously reduced. Extensive experiments are conducted on *CIFAR-10* and *CIFAR-100* with three noise structures as well as the industrial-level *Clothing1M* with agnostic noise structure, and the results show that Masking improves the robustness of classifiers significantly.

## 3.1 Introduction

It is always a challenging task to learn from noisy labels [Angluin and Laird, 1988; Azadi et al., 2015; Ma et al., 2018; Reed et al., 2014; Rodrigues and Pereira, 2018], since these labels are systematically corrupted. As the negative effect, noisy labels inevitably degenerate the accuracy of classifiers. And this negative effect becomes more prominent for deep neural networks, since these complex models can almost fully memorize all noisy labels, which correspondingly degenerates their generalization [Zhang et al., 2016] to the clean dataset. Unfortunately, noisy labels are ubiquitous and unavoidable in our daily life, such as web queries [Liu et al., 2011], social-network tagging [Cha and Cho, 2012], crowdsourcing [Welinder et al., 2010], medical images [Dgani et al., 2018], and financial analysis [Aït-Sahalia et al., 2010].

To handle such noisy labels, the recent approaches explore three directions mainly. One direction focuses on training via sample re-weighting, which leverages the sample-weighting bias [Huang et al., 2007] to handle the label noise issue. For example, the MentorNet [Jiang et al., 2018] trains on the "small-loss" samples through a binary weighting strategy. Meanwhile, Decoupling [Malach and Shalev-Shwartz, 2017] trains on "disagreement" samples, for which the predictions of two networks disagree. However, since the data for training are weighted or selected on the fly rather than done at the beginning, it is hard to characterize these biases, and then it is hard to give any theoretical guarantee on the consistency of learning. Another direction develops the model regularization techniques, including parameter regularization and feature regularization. This direction employs the regularization bias to overcome the label noise issue. Explicit parameter regularization is added to the objective function, such as manifold regularization [Belkin et al., 2006] and virtual adversarial training [Miyato et al., 2016]. Implicit feature regularization is designed for training algorithms, such as temporal ensembling [Laine and Aila, 2017] and mean teacher [Tarvainen and Valpola, 2017]. Nevertheless, both approaches introduce a permanent regularization bias, and the learned classier barely reaches the optimum [Domingos and Pazzani, 1997]. The last direction estimates the noise transition matrix without introducing sample re-weighting bias and regularization bias. As an approximation of real-world corruption, noisy labels are theoretically flipped from the ground-truth labels by an unknown noise transition. In this methodology, the accuracy of classifiers can be improved by estimating this transition accurately and we are rewarded with the data scalability, the optimization efficiency and the theoretical interpretability simultaneously.

The previous methods for estimating the noise transition can be roughly summarized into two solutions. One solution estimates the noise transition in advance, and subsequently learns the classifier based on this estimated transition. For example, Patrini et al. [Patrini et al., 2017] leveraged a two-step solution to estimate the noise transition. The benefit is to require limited data only for the estimation procedure.



(a) (b) (c) (d)

Figure 3.1: The quantitative analysis on the degeneration of the noise transition induced by the decoupling bias of deep neural networks. (a) The image samples of the *CIFAR-10* dataset. (b) The simulated noise transition matrix to disturb the original clean labels. (c) The inferred noise transition by the two-step solution in [Patrini et al., 2017]. (d) The inferred noise transition by the noise adaptation layer in [Goldberger and Ben-Reuven, 2017].

Nonetheless, since deep neural networks can almost fully memorize the label noise, their method may not estimate the noise transition accurately under a noise-polluted classifier. The other solution jointly estimates the noise transition matrix and learns the classifier in an end-to-end framework. For instance, on top of the softmax layer, Sukhbaatar et al. [Sukhbaatar et al., 2014] added a constrained linear layer to model the noise transition matrix, while Goldberger et al. [Goldberger and Ben-Reuven, 2017] added a nonlinear softmax layer. The benefit is the generality of their unified learning framework. However, the learning may adapt to other undesired local minimums under a finite dataset. In total, above problems can be attributed to the much larger capacity of deep neural networks compared to that of noise transition. This may induce the "explaining away" [Jordan et al., 1999] bias to deep neural networks in reasoning the noisy dataset. We term this phenomenon as the decoupling bias and give the simple definition as follows.

**Definition 3.1.** *The decoupling bias is the phenomenon when one reasoning module with the limited capacity concatenates on top of deep neural networks to cooperatively explain noisy datasets, the estimation of this module suffers from the degeneration risk.*

Figure 3.1 empirically validates our conjecture about the decoupling bias issue. Specifically, the approximate diagonal matrices in the last two panels of Figure 3.1 mean the previous methods treat the noisy labels directly as the latent labels, which almost has not filtered noise. The reasoning of the noise transition is severely divested by the deep neural networks, i.e., the decoupling bias. Therefore, an important question is, with a finite dataset, can we leverage a constrained end-to-end model to overcome

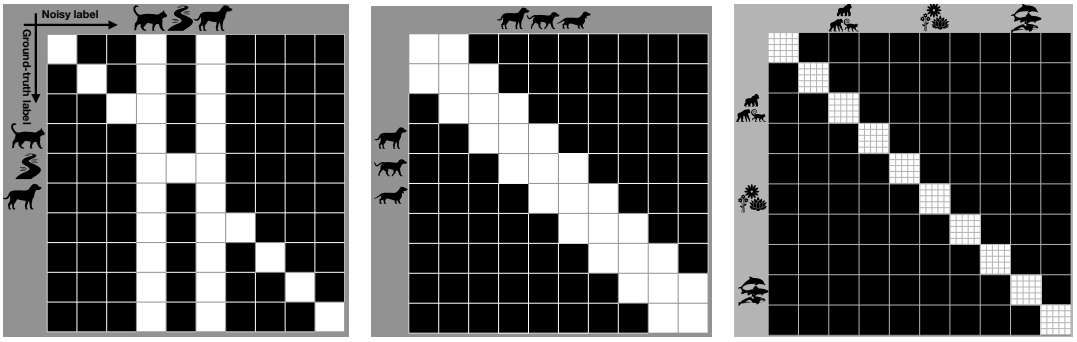
this issue? In this Chapter, we present a human-assisted approach called "*Masking*". Masking conveys human cognition of invalid class transitions (i.e., cat $\nleftrightarrow$ car), and speculates the structure of the noise transition matrix. The structure information can be viewed as a constraint to prevent the degeneration and improve the estimation procedure. Namely, given the structure information, we can focus on estimating the noise transition probability along the structure, which reduces the estimation burden and the degeneration risk largely.

To instantiate our approach, we derive a structure-aware probabilistic model, by incorporating a structure prior. In the realization, we encounter two practical challenges: structure extraction and structure alignment. Specifically, to address the structure extraction challenge, we propose a tempered sigmoid function to simulate the human cognition on the structure of the noise transition matrix. To address the structure alignment challenge, we propose a variant of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] to avoid the difficulty of specifying the explicit distributions. We conduct extensive experiments on two benchmark datasets (*CIFAR-10* and *CIFAR-100*) with three noise structures, and the industrial-level dataset (*Clothing1M* [Xiao et al., 2015]) with agnostic noise structure. The experimental results show our method can improve the robustness of classifiers significantly.

## 3.2 A New Perspective of Noisy Supervision

In this section, we explore the noisy supervision from a new perspective, namely the *structure* of the noise transition matrix. First, we discuss where noisy labels normally come from, and why we can speculate the structure of the noise transition matrix. Then, we present the representative structures of the noise transition matrix. Finally, we discuss the potential of the noise structure to help us prevent the degeneration.

In practice, noisy labels mainly come from the interaction between humans and tasks, such as social-network tagging and crowdsourcing. Assume that the more complex the interaction is, the more efforts human beings spend. Due to cognitive psychology [Michalski et al., 2013], human cognition can mask invalid class transitions, and highlight valid class transitions automatically. This denotes that, human cognition can speculate the structure of the noise transition matrix correspondingly. One effort-saving interaction is that, you tag a cluster of scenery images in social networks, such as beach, prairie, and mountain. However, the foreground of some images appears a dog or a cat, yielding noisy labels [Ren et al., 2018]. Thus, human cognition masks invalid class transitions (i.e., beach $\nleftrightarrow$ mountain), and highlights valid class transitions (i.e., beach $\leftrightarrow$ dog). This noise structure should be the diagonal matrix coupled with two column lines, namely a *column-diagonal* matrix (Figure 3.2(a)), where the column lines correspond to the dog and cat classes, respectively.



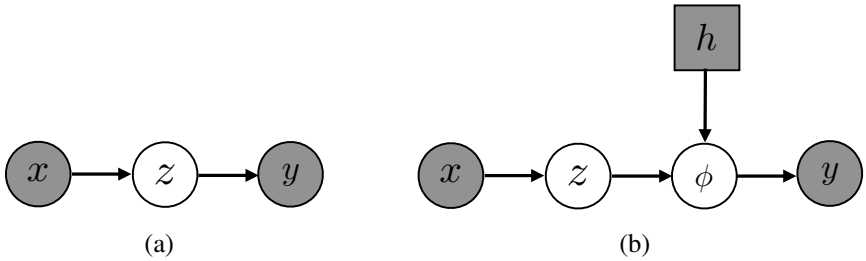

(a) (b) (c)

Figure 3.2: Three types of noise structures: (a) column-diagonal noise structure, (b) tri-diagonal noise structure, (c) block-diagonal noise structure. The vertical axis denotes the class of ground-truth label, while the horizontal axis denotes the class of noisy label. The white cell means the valid class transitions, while the black cell means the invalid class transitions.

Another effort-consuming interactions stem from the task annotation on Amazon Mechanical Turk [1]. Even with high-degree efforts, amateur workers may be potentially confused by very similar classes to yield noisy labels, due to their limited expertise. There are two practical cases. The fine-grained case denotes that, the transition from one class to its similar class is continuous (e.g., Australian terrier, Norfolk terrier, Norwich terrier, Irish terrier, and Scotch terrier), and workers make mistakes in the adjacent positions [Deng et al., 2013]. Thus, human cognition can mask invalid class transitions (i.e., Australian terrier $\nleftrightarrow$ Norwich terrier), and highlight valid class transitions (i.e., Norfolk terrier $\leftrightarrow$ Norwich terrier $\leftrightarrow$ Irish terrier). This noise structure should be a *tri-diagonal* matrix (Figure 3.2(b)). The hierarchical-grained case denotes that, the transition among super-classes is discrete (e.g., aquatic mammals and flowers) and impossible, while the transition among sub-classes is continuous (e.g., aquatic mammals contain beaver, dolphin, otter, seal, and whale) and possible [Patrini et al., 2017]. Thus, human cognition can mask invalid class transitions (i.e., aquatic mammals $\nleftrightarrow$ flowers), and highlight valid class transitions (i.e., beaver $\leftrightarrow$ dolphin). This noise structure should be a *block-diagonal* matrix (Figure 3.2(c)), where each block represents a super-class.

Intuitively, the degeneration induced by the decoupling bias of deep neural network will lead to the inaccurate noise transition estimation including the noise structure and the transition probabilities. If we can constrain the degeneration with the help of the noise structure, the noise transition learning can be towards at our desired direction in structure and the transition probabilities can be more accurately estimated. However,

[1] https://www.mturk.com/




(a) (b)

Figure 3.3: Graphical representation of benchmark models and our MASKING model. Assume that, $(x, y)$ denotes the image with the noisy label and $z$ represents the latent ground-truth label. $\phi$ is the noise transition matrix, where $\phi_{ij} = \Pr(y = e^j | z = e^i)$. (a) A benchmark model. (b) our MASKING that models the noise transition matrix by an explicit variable $\phi$. We embed a structure constraint on the variable $\phi$, where the structure information comes from the given human cognition $h$.

how can we instill the structure into the estimation procedure? We discuss this in the following.

## 3.3 Noise Structure against The Decoupling Bias

In the following, we briefly show how benchmark models handle noisy labels and reveal their deficiencies. Then, we show why Masking via the noise structure can solve such issues. After the Masking procedure, we present a straightforward idea to incorporate the noise structure and analyze its dilemma in the implementation.

### 3.3.1 Deficiency of benchmark models

Figure 3.3(a) is a basic model to train a classifier in the presence of noisy supervision. Assuming that $x$ represents a "Dog" image, its latent ground-truth label $z$ should be a "Dog" class. However, its annotated label $y$ belongs to the "Cat" class. In essence, the noisy label $y$ is flipped from the ground-truth label $z$ by an unknown noise transition matrix. Therefore, the current techniques tend to improve the accuracy of classifier $(x \to z)$ by estimating the noise transition matrix $(z \to y)$.

Here, we introduce two benchmark realizations of Figure 3.3(a). The first benchmark model comes from Patrini et al. [Patrini et al., 2017], which uses the anchor set condition [Li et al., 2017a] to independently estimate the noise transition matrix $(z \to y)$. Based on the estimated matrix, they learn the classifier $(x \to z)$ by the strategy of loss corrections. However, when deep neural networks are capable of memorizing most of the noisy data, the decoupling bias can severely degenerate

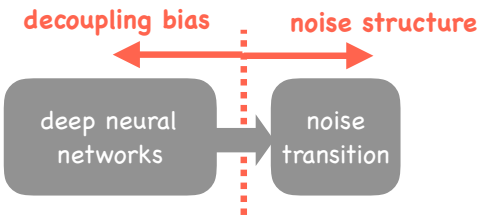



Figure 3.4: The effect of the noise structure and the decoupling bias. The decoupling bias encourages deep neural networks memorize all data instead of reasoning it to the noise transition. The noise structure forces the valid transition happens and as much as possible prevents the invalids transition degeneration, which performs as the reverse action force of the decoupling bias in the optimization.

the estimation of $\phi$. For example, when deep neural networks treat the whole dataset as the clean samples and the loss in the training is minimize to zero. In this case $\phi$ will be a diagonal matrix that is fully no sense in reasoning the noise. The other benchmark model comes from Goldberger and Ben-Reuven [Goldberger and Ben-Reuven, 2017], which unifies the two steps of the first benchmark model into a joint fashion. Specifically, the noise transition matrix ($z \rightarrow y$) modeled by a nonlinear Softmax layer connects the classifier ($x \rightarrow z$) with noisy labels $y$ for the end-to-end training. Under carefully tweaking, it shows promising improvement over [Patrini et al., 2017]. Nevertheless, with a finite dataset, the end-to-end training cannot prevent the decoupling bias and still encourages the deep neural network to absorb the noise compared to the noise transition.

### 3.3.2 Does structure matter?

Therefore, given a finite dataset, can we leverage a constrained end-to-end model to solve the above deficiencies? The answer is affirmative. The reason can be explained intuitively: when human cognition masks the invalid class transitions (i.e., cat $\nleftrightarrow$ car), the structure information is available as the constraint to perform the reverse action force of the decoupling bias in the optimization. The constrained end-to-end model only focuses on estimating the noise transition probability, which alleviate the degeneration of noise transition. The estimation burden will be largely reduced and thus the brute-force estimation can find a good local minimum more easily. We illustrate this in Figure 3.4.

In this Chapter, we summarize this human-assisted approach via the noise structure as "*Masking*". Intuitively, with high probability, Masking means some class transitions will not happen (i.e., cat $\nleftrightarrow$ car), while some class transitions will happen (i.e., beaver $\leftrightarrow$ dolphin). Before delving into our realization, we first present a straightforward idea

to incorporate the structure information into an end-to-end model as follows.

### 3.3.3 Straightforward dilemma

For structure instillation, the most straightforward idea is to add a regularizer on the objective of an end-to-end model. Namely, we can use a regularizer to represent the structure constraint. Due to the regularization effect in the optimization [Goodfellow et al., 2016], the noise transition matrix learned by previous models [Goldberger and Ben-Reuven, 2017; Patrini et al., 2017] will satisfy our expectation regarding the structure. For example, we can leverage the classical Lagrange multiplier [1] in the benchmark model from [Goldberger and Ben-Reuven, 2017] to instantiate this idea.

However, such a deterministic method may not be easily implemented in practice due to three reasons. First, such a class of regularizers requires a suitable distance measure to compute the distance between the learned transition matrix and the prior, and the corresponding regularization parameter. In deep learning scenarios, it is quite hard to justify the choice of a distance measure, e.g., why choosing $L_2$ distance instead of other distance measures for the structure instillation. Second, if we leverage a noisy validation set, we need to construct the unbiased risk estimator for the backward correction. This is impossible, as the inverse of estimated noise transition cannot be accurately computed.

Last but not least, even though we construct a clean validation set, it needs to repeat the training procedure to tune the regularization parameter, which consumes remarkable computational resources. Specifically, it requires a lot of non-trivial trials in the training-validation phase to find an optimal weight. For example, the Residual-52 net will take about one week to be well trained on the WebVision dataset, which consists of millions of images with noisy labels [Li et al., 2017a]. If we consider adding a regularizer to instill structure information, each trial for adjusting the weight is a disaster. To sum up in this thesis, we do not consider this straightforward regularization approach.

## 3.4 When Structure Meets Generative Model

### 3.4.1 The Structure-Aware Probabilistic Model

Based on the previous discussions, we conjecture that the Bayesian method should be a more suitable tool to model the structure information, since the structure information can be explicitly represented as the prior. Following this conjecture, we deduce an end-to-end probabilistic model to incorporate the structure information as shown in Figure 3.3(b).

[1] https://en.wikipedia.org/wiki/Lagrange_multiplier

Compared to benchmark models in Figure 3.3(a), we explicitly model the noise transition matrix with a random variable $\phi$, and we instill the structure information by controlling the prior of its corresponding structure variable, termed as $\phi_o$. Here, we assume there exists a deterministic function $f(\cdot)$ such that $\phi_o = f(\phi)$. The reason that we make such an assumption can be intuitively explained with the observation, once one arbitrary matrix is given, human cognition can certainly describe its structure, e.g., diagonal or tri-diagonal. Thus, there must be a function that can implement the mapping from $\phi$ to its structure $\phi_o$. For clarity, we present an exemplar generative process for "multi-class" & "single-label" classification problem with noisy supervision.

- The latent true label $z \sim P(z|x)$, where $P(z|x)$ is a Categorical distribution [1].
- The noise transition matrix $\phi \sim P(\phi)$ and its structure $\phi_o \sim P(\phi_o)$, where $P(\phi)$ is an implicit distribution modeled by neural networks without the exact form (i.e., multi-Dirac distribution), $P(\phi_o) = P(\phi)\frac{d\phi}{d\phi_o}\big|_{\phi_o=f(\phi)}$, and $f(\cdot)$ is the mapping function from $\phi$ to $\phi_o$.
- The noisy label $y \sim P(y|z,\phi)$, where $P(y|z,\phi)$ models the transition from $z$ to $y$ given $\phi$.

According to the above generative process, we can deduce the following evidence lower bound (ELBO) to approximate the log-likelihood of the noisy data via variational inference, which can be named as MASKING, a structure-aware probabilistic model:

$$
\begin{aligned}
\ln P(y|x) &= \ln \int_{\phi} \sum_{z} P(y|z,\phi)P(z|x)P(\phi)d\phi \\
&= \ln \int_{\phi} \sum_{z} P(y|z,\phi)P(z|x)\frac{Q(\phi)P(\phi)}{Q(\phi)}d\phi \\
&\geq \mathbb{E}_{Q(\phi)}\left[\ln \sum_{z} P(y|z,\phi)P(z|x) - \ln \frac{Q(\phi)}{P(\phi)}\right] \\
&= \mathbb{E}_{Q(\phi)}\left[\ln \sum_{z} P(y|z,\phi)P(z|x) - \ln \frac{Q(\phi_o)\frac{d\phi_o}{d\phi}}{P(\phi_o)\frac{d\phi_o}{d\phi}}\Big|_{\phi_o=f(\phi)}\right] \\
&= \mathbb{E}_{Q(\phi)}\left[\ln \sum_{z} P(y|z,\phi)P(z|x) - \ln \left(Q(\phi_o)/P(\phi_o)\right)\Big|_{\phi_o=f(\phi)}\right],
\end{aligned}
\tag{3.1}
$$

where $Q(\phi)$ is the variational distribution to approximate the posterior of the noise transition matrix $\phi$, and $Q(\phi_o) = Q(\phi)\frac{d\phi}{d\phi_o}\big|_{\phi_o=f(\phi)}$ is the corresponding variational

[1] For the single-label classification, a Categorical distribution is a natural choice. For the multi-label classification, a Multinomial distribution is used, since one example corresponds to multiple labels.

distribution of the structure $\phi_o$. Eq. (3.1) seamlessly unifies previous models and structure instillation, remarked as the following benefits.

**Remark 1.** *The first term inside the expectation in Eq. (3.1) recovers the previous benchmark models, representing the log-likelihood from $x$ to the noisy label $y$. The second term inside the expectation in Eq. (3.1) is for structure instillation, reflecting the inconsistency between the distribution $Q(\phi_o)$ learned from the training data and the structure prior $P(\phi_o)$ provided by human cognition.*

*As a whole, MASKING model benefits from the human guidance (the second term) in the procedure of learning with noisy supervisions (the first term), which avoids the unexpected local minima with incorrect structures in previous works. Moreover, we avoid the difficulty of hyperparameter selection by deducing a Bayesian framework, which does not require the regularization parameter to be tuned.*

*Note that the human cognition may introduce uncertainty. In this case, we just focus on the certain knowledge and use this knowledge as the prior; on the other hand, we dispose the uncertain knowledge by Masking. A special case is when we do not have any transition knowledge, we can unmask the whole matrix, i.e., allowing all possible transitions. This naturally degenerates our MASKING model to the unconstrained S-adapation method.*

### 3.4.2 Towards Principled Realization

To realize the MASKING model, concretely, the second term in Eq. (3.1), we encounter two practical challenges and present the principled solutions as follows.

**Challenge from structure extraction**. One challenge comes from how to specify the mapping function $f(\cdot)$ in Eq. (3.1), which extracts the structure variable $\phi_o$ from the variable $\phi$. Without this step, we cannot compute $Q(\phi_o) = Q(\phi)\frac{d\phi}{d\phi_o}\big|_{\phi_o=f(\phi)}$, and let alone optimize the second term in Eq. (3.1). However, to the best of our knowledge, there is no related work that specifies $f(\cdot)$ in the area of structure extraction.

Here, we explore a principled solution by simulating human cognition on the structure of the noise transition matrix. In terms of the noise transition probability between two classes, human cognition considers the small value (i.e., $0.5\%$) as a sign of the invalid class transition, but the large value (i.e., $20\%$) as a noise pattern [Grigolini et al., 2009]. It indicates that, the larger transition probability is favored when quantifying the noise transition matrix into a structure. Such a procedure is very similar to the thresholding binarization operation with a tempered sigmoid function (Eq. (3.2)). Thus, we can use the following tempered sigmoid function as $f(\cdot)$ to simulate the mapping from $\phi$ to $\phi_o$,

$$f(s) = \frac{1}{1+\exp\left(-\frac{\phi-\alpha}{\beta}\right)}, \quad \text{where } \alpha \in (0,1), \beta \ll 1, \tag{3.2}$$

and we name its output $f(\phi)$ as the masked structure $\phi_o$ from $\phi$. By controlling the location parameter $\alpha$ and the scale parameter $\beta$, Eq. (3.2) quantifies $\phi$ into the structure $\phi_o$ for the second term in Eq. (3.1).

**Challenge from structure alignment**. The second term in Eq. (3.1) is to make the structure learned from the training data (represented by $Q(\phi_o)$), close to the human prior (represented by $P(\phi_o)$). We consider it as structure alignment. The challenge here is how to specify the distributions $P(\phi_o)$ and $Q(\phi_o)$ in Eq. (3.1) to reasonably measure their divergence for the structure alignment. The smaller divergence between $P(\phi_o)$ and $Q(\phi_o)$, the more similar they are.

This question is difficult because, for prior $P(\phi_o)$, we usually provide one or a few limited structure candidates, and human cognition has the sparse empirical certainty [Goodman, 1952] according to a specific noisy data. That is to say, $P(\phi_o)$ should be a distribution that concentrates on one or a few points, e.g., a multi-Dirac distribution or a multi-spike distribution[1]. These distributions are usually quite unstable for optimization, since they are approximately discrete, and easy to cause the computational overflow problem [Rockova and McAlinn, 2017]. Regarding $Q(\phi_o)$, it is equal to specifying $Q(\phi)$ since they are correlated by $\phi_o = f(\phi)$. If we find $Q(\phi)$ such that $\mathbb{E}_{Q(\phi)}\left[\ln Q(\phi_o)/P(\phi_o)\right]$ is analytically computable, we can avoid this challenge. However, such a specification with existing distributions is usually intractable.

Fortunately, we can employ the implicit models to deal with the above dilemma [Tran et al., 2017a]. This is because in the implicit models, the distribution is directly simulated with neural networks plus a random noise, which avoids to specify an explicit distribution. Following this methodology, $Q(\phi)$, $Q(\phi_o)$ and $P(\phi_o)$ in Eq. (3.1) can be implemented like Generative Adversarial Networks (GANs). Specifically, $Q(\phi)$ and the divergence between $Q(\phi_o)$ and $P(\phi_o)$ are parameterized with two neural networks, i.e., one generator and one discriminator, and then play an adversarial game [Goodfellow et al., 2014]. However, different from the original GANs, our model has one extra discriminator (called as reconstructor), since the first term in Eq. (3.1) involves $\phi$, which acts as a second discriminator during the game.

**Generative Adversarial Learning**. Concretely, the corresponding implementation of each term in Eq. (3.1) is illustrated in Figure 3.5, consisting of three modules, generator, discriminator and reconstructor. The generator is responsible for generating a distribution $Q(\phi)$ of the noise transition matrix $\phi$, which serves for both terms in Eq. (3.1). The discriminator implements the function of the second term in Eq. (3.1) to measure the difference $\mathcal{M}(\phi_o, \hat{\phi}_o)$ between the extracted structure $\phi_o$ with Eq. (3.2) and our prior structure $\hat{\phi}_o$. The reconstructor is for the first term in Eq. (3.1), facilitating the classifier prediction $P(z|x)$ and the noise transition matrix $\phi$ to yield noisy labels $y$. In

[1] https://en.wikipedia.org/wiki/Dirac_delta_function; https://en.wikipedia.org/wiki/Spike-and-slab_variable_selection

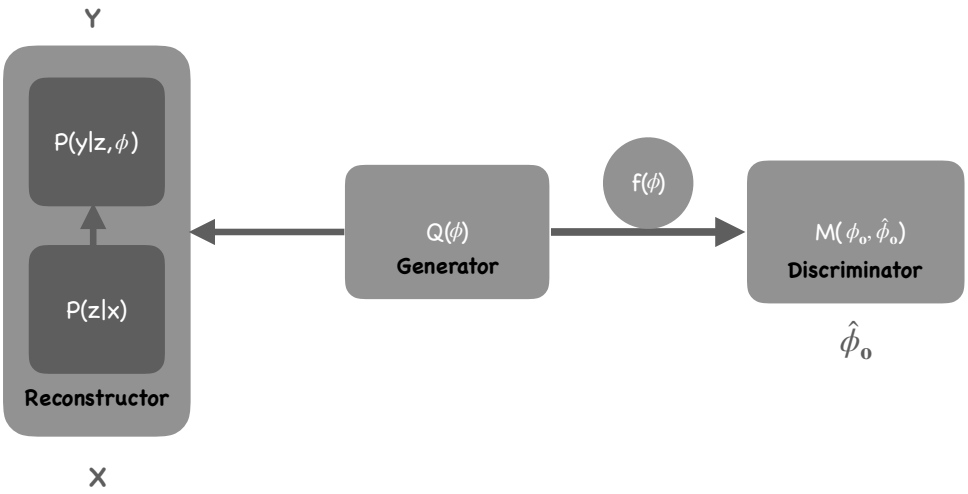



Figure 3.5: A GAN-like implementation to model the structure instillation in our MASKING. The generator is to output the noise transition for the reconstructor and the discriminator. For the former, $\phi$ is directly utlized to model transition combined with the classifier $P(z|x)$ and for the latter, $\phi$ will be extracted the noise structure by $f(\phi)$ to align the human provided structure $\hat{\phi}_o$ via $\mathcal{M}$.

this way, we instill the structure information from human cognition into an end-to-end model. The pseudo algorithm to train the Masking is summarized in Algorithm 1. Note that, for step 6, we use 5 iterations to train $\mathcal{G}$. This is an empirical trick in generative-adversarial learning paradigms. By training $\mathcal{G}$ with more iterations than $\mathcal{C}$ and $\mathcal{D}$, we leave enough patience to learn a good generator, since it is usually difficult to simulate gegeneration compared to the other parts. We can also set larger iterations, but there might be no need to do this in the early stage of $\mathcal{C}$ and $\mathcal{D}$, which is far away from a good minimum. To avoid extra tuning, we just uniformly use 5 iterations.

## 3.5 Experiments

In this section, we will verify the robustness of MASKING from two folds. First, we conduct experiments on two benchmark datasets with three types of noise structure: namely (1) column-diagonal (Figure 3.2(a)); (2) tri-diagonal (Figure 3.2(b)); and (3) block-diagonal (Figure 3.2(c)). Second, we conduct experiments on one industrial-level dataset with agnostic noise structure. The experiments are explained as follows.

**Algorithm 1** The generative adversarial learning to train the MASKING model.

1: **Input** the batch size $M$, the total epoch $L$, the pretrained epoch $L_0$, the prior noise structure $\hat{\phi}_o$, the hyperparameter $\alpha$ and $\beta$.
**for** $l = 1 \rightarrow L$ **do**
  **while** *one whole epoch not full* **do**
    2: Hook a batch of samples.
    **if** $l < L_0$ **then**
      3: Directly train $\mathcal{C}$ on noisy samples.
      4: Train $\mathcal{G}$ and $\mathcal{D}$ jointly based on $\hat{\phi}_o$.
    **else**
      5: Fix $\mathcal{G}$, train $\mathcal{C}$ and $\mathcal{D}$ jointly.
      6: Fix $\mathcal{C}$ and $\mathcal{D}$, train $\mathcal{G}$ by 5 iterations.
    **end**
  **end**
**end**
7: **Output** the classifier $\mathcal{C}$ and the generator $\mathcal{G}$.

### 3.5.1 Experimental Settings

**Datasets.** For the toy experiments, *CIFAR-10* and *CIFAR-100* datasets are used, which respectively consists of 60,000 32x32 color images from 10 and 100 classes. Both datasets consist of 50k samples for training and 10k samples for testing, where each sample is a $32 \times 32$ color image and its label. For *CIFAR-10*, we randomly flip the labels of the training set according to the first two types of noise structure, which has been illustrated in Figure 3.2(a) and Figure 3.2(b) with predefined transition probabilities. For *CIFAR-100*, we implement the similar procedure to generate the noisy data, but follow the last type of noise structure in Figure 3.2(c). All predefined transition probabilities can be found in Table 3.1 (4th row) based on the side-bar. For the real-world test, an industrial-level dataset called *Clothing1M* [Xiao et al., 2015] from online shopping websites (i.e., Taobao.com) is used here, where the ground-truth transition matrix is not available. *Clothing1M* includes mislabeled images of different clothes, such as hoodie, jacket and windbreaker. This dataset consist of 1000k samples for training and 1k samples for testing, where each sample is a $256 \times 256$ color image and its label. According to [Xiao et al., 2015], only about $61.54\%$ labels are reliable. Besides, this dataset contains 50k, 14k and 10k clean samples respectively for auxiliary training, validation and test. Although we cannot know the accurate structure prior for *Clothing1M*, we can distill an approximated structure from the pre-estimated transition matrix [Xiao et al., 2015]. We consider the transition probabilities greater than $0.1$ as the valid transition patterns, and the transition probabilities smaller than $0.01$ as the invalid transition patterns.

**Baselines and Implementations.** We compare MASKING with the state-of-the-

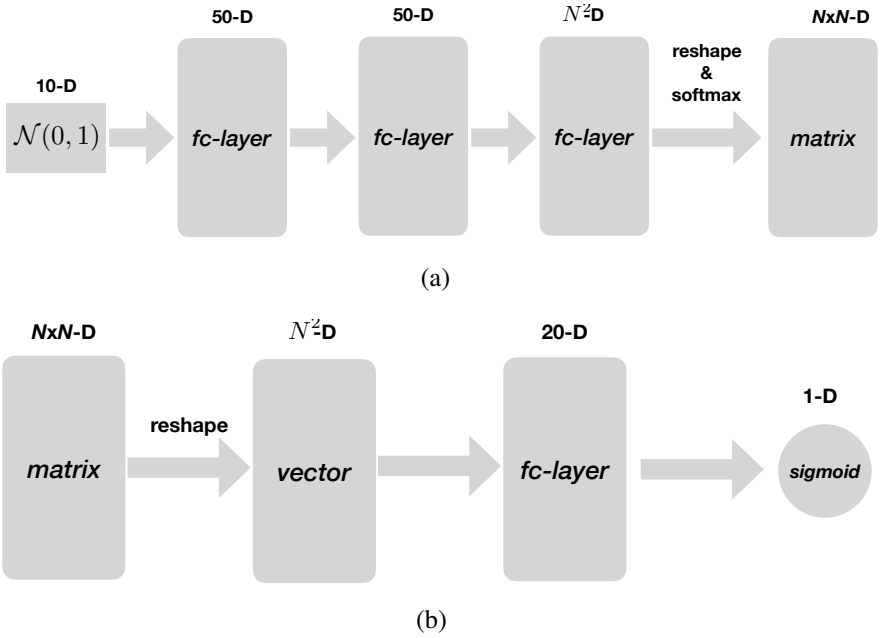
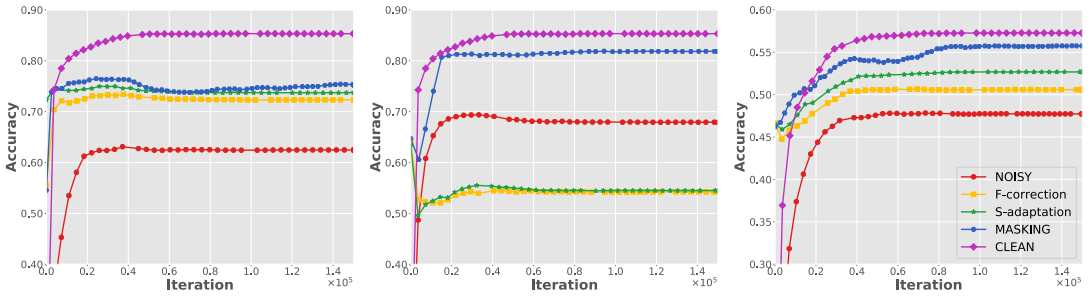



Figure 3.6: The network configurations of the generator module and the discriminator module. (a) The generator. (b) The discriminator. $N$ is the class number.

art & the most related techniques for noisy supervision: (1) forward correction [Patrini et al., 2017] (F-correction) and (2) S-adaptation [Goldberger and Ben-Reuven, 2017]. We also compare it with directly training deep networks on noisy data (marked as (3) NOISY) and clean data (marked as (4) CLEAN). The performance of CLEAN can be viewed as an oracle or an upper bound. The prediction accuracy is used to evaluate the classification performance of each model in the test set. Besides, we qualitatively visualize the noise transition matrix when the training of each model converges, to analyze whether the true noise transition matrix is approached.

All experiments are conducted on a NVIDIA TITAN GPU, and all methods are implemented by Tensorflow. We adopt the same base network as the classifier of all methods, and apply the cross-entropy loss for noisy labels. The stochastic gradient descent optimizer has been used to update the parameter of baselines and the classifier in MASKING. For the generator and the discriminator, the network structure is illustrated in Figure 3.6(a) and Figure 3.6(b). We follow the advice in Gulrajani et al. [Gulrajani et al., 2017] to choose the RMSProp optimizer. For both datasets, the batch size is set to 128 for 15,000 iterations. $\alpha$ and $\beta$ in Eq. (3.2) are respectively set $0.05$ and $0.005$. The estimation of the noise transition matrix in F-correction and the initialization of the adaption layer in S-adaptation follow the strategy in Patrini et al. [Patrini et al., 2017]. Note that we have tried the original initialization way in Goldberger and Ben-Reuven [Goldberger and Ben-Reuven, 2017], but it is not better




(a) (b) (c)

Figure 3.7: Test accuracy vs iterations on benchmark datasets with three types of noise structures, i.e., the column-diagonal structure, the tri-diagonal structure and the block-diagonal structure.

than the way in [Patrini et al., 2017]. Our implementation of MASKING is public available on the website[1]. The learning rate is initialized as 0.1, and decreases with the operation *tf.train.exponential_decay* with the staircase. The corresponding decay factor is set $0.1$. The learning rate for generator and discriminator is fixed with $3e-4$.

### 3.5.2 Results Analysis

We train MASKING and baselines (except CLEAN) on the noisy datasets and validate the performance in the clean test datasets. Figure 3.7 depicts the test accuracy on benchmark datasets with three types of noise transition matrices. From the comparison, we can find that MASKING persistently outperforms F-correction, S-adaptation and NOISY, and the performance is upper bounded by CLEAN which uses the clean dataset counterpart for training. Instead, although F-correction and S-adaptation show improvement compared with directly training on the noisy dataset, i.e., NOISY, they may fall in some case as shown in the middle panel of Figure 3.7. This, in fact, can trace back to the inaccurate noise transition in the optimization that is degenerated by the decoupling bias. It presents the need to solve the decoupling bias issue and demonstrates the value of MASKING. Furthermore, in terms of tri-diagonal and block-diagonal cases, it almost achieves the performance comparable to that of CLEAN.

Besides, Table 3.1 presents the visualization of the noise transition matrix estimated by F-correction, S-adaptation and MASKING, as well as the true noise transition matrix. As can be seen, with the guidance of the prior structure, MASKING infers the noise transition matrix better than two baselines. Specifically, we observe that for the estimation on the tri-diagonal structure, both F-correction and S-adaptation severely

[1] https://github.com/Sunarker/Masking

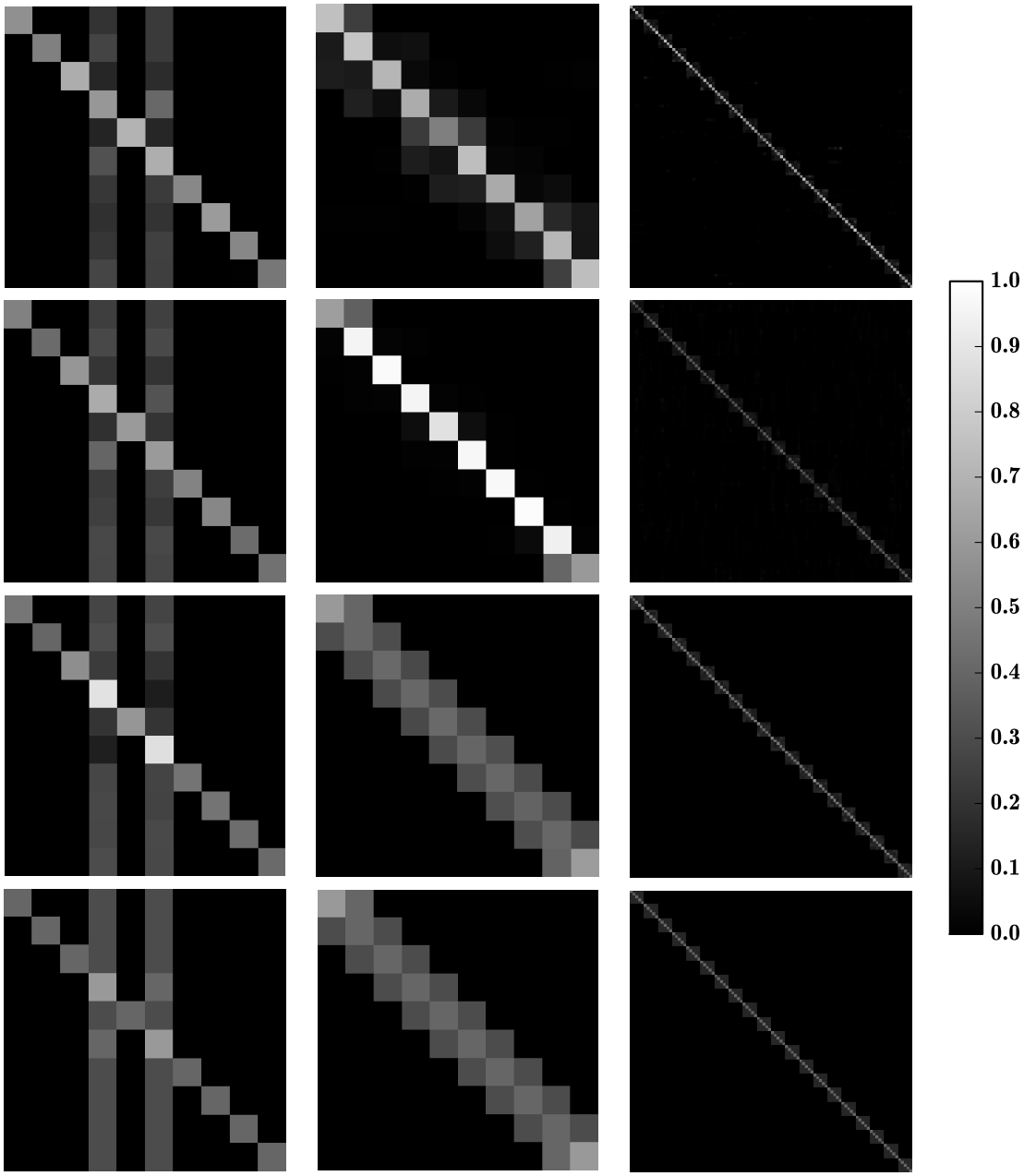



Table 3.1: The estimation of the noise transition matrix by F-correction (1st row), S-adaptation (2nd row) and MASKING (3rd row), and the truth (4th row) in the case of three types of noise transition structure: column-diagonal (1st column), tri-diagonal (2nd column), block-diagonal (3rd column).

(a) (b) (c)

Figure 3.8: The visualization of (a) the manually provided transition, (b) the candidate transition and (c) the learned noise transition via MASKING. Note that, the candidate transition consists of the bright cells that represents valid transition, black cells that represents invalid transition and the gray cells that represents uncertain transition. Only the former two are used as the prior of MASKING.

| Models | Performance(%) |
|---|---|
| NOISY | 68.9 |
| F-correction | 69.8 |
| S-adaptation | 70.3 |
| MASKING | 71.1 |
| CLEAN | 75.2 |

Table 3.2: Test accuracy on *Clothing1M*.

fail. F-correction pre-estimates a non-ideal matrix, and S-adaptation tunes it worse, while MASKING learns it better. To avoid the performance drop when directly training on the noisy dataset as [Arpit et al., 2017; Jiang et al., 2018], we have configured the dropout layer in deep neural networks.

Regarding the experiments on the real-world *Clothing1M*, the true noise structure are agnostic unlike previous toy experiments. We use the manually provided estimation by [Xiao et al., 2015] as the rough human knowledge and cast it as the noise structure. By setting the thresholds, the transition that has below 0.01 probability is treated as the invalid transition and that has above 0.1 probability is treated as the valid transition, while that between 0.01 and 0.1 probability is treated uncertain transition and will not be used as the structure prior. Figure 3.8 illustrates the noise transition and the structure prior as well as the learned transition. As can be seen, our method almost keeps the noise structure provided by the prior, but the

transition probabilities are different from those provided by Figure 3.8(a). According to the test accuracy of all methods summarized in Table 3.2, MASKING outperforms other methods, demonstrating this constrained adaptation learns more accurate noise transition opposing to the degeneration of the decoupling bias. But compared to results in Figure 3.7, the robustness of MASKING marginally outperforms that of F-correction and S-adaptation. We conjecture two reasons exist: First, the structure prior we used here is not the ground-truth noise structure. Second, the ground-truth noise structure of *Clothing1M* is too complex to be estimated easily. To solve the estimation issue, we can use crowdsourcing to provide labels in practice and invite an expert to find the accurate noise structure. This is much cheaper than letting experts assign labels.

## 3.6 Summary

This chapter presents a Masking approach to overcome the decoupling bias issue of learning via noise transition. This approach conveys human cognition of valid and invalid class transitions and speculates the structure of the noise transition. Given the structure information, we derive a structure-aware probabilistic model (MASKING) that incorporates a structure prior and present an generative adversarial learning implementation. Empirical results on the toy datasets and the real-world dataset demonstrate that our approach can improve the robustness of classifiers obviously in the presence of noisy supervision. However, as the side effect of the assumption, a reliable noise structure prior is usually required if we want to achieve a good performance. In future, the relaxation of this assumption can be a potential direction to explore, since saving human labor to zero is always the pursuit of the intelligence. For example, we can investigate whether we can find an independent way to extract the noise structure automatically, or whether we can extend the MASKING self-corrects the incorrect noise structure when it starts from a unreliable initialization. In summary, Masking as a first explore to overcome the decoupling bias of deep neural networks by the noise structure has opened a heuristic perspective to deep learning with noisy supervision.

# Chapter 4

# A Stable Stochastic Approximation

Compared with learning noise transition via the auxiliary structure in Chapter 3, this Chapter explores an iterative but more stable, efficient method in the perspective of optimization. Specifically, we find previous method that adapts the pre-estimation along with the training progress via a Softmax layer is quite fragile in performance. The parameters in the Softmax layer easily get stuck into local minimums due to the ill-posed stochastic approximation. To address these issues, we propose a Latent Class-Conditional Noise model (LCCN) that naturally embeds the noise transition under a Bayesian framework. By projecting the noise transition into a Dirichlet-distributed space, the learning is constrained on a simplex based on the whole dataset, instead of some ad-hoc parametric space. We then deduce a dynamic label regression method for LCCN to iteratively infer the latent labels, to stochastically train the classifier and to model the noise. Our approach safeguards the bounded update of the noise transition, which avoids previous arbitrarily tuning via a batch of samples. We further generalize LCCN for open-set noisy labels and the semi-supervised setting. We perform extensive experiments with the controllable noise data sets, CIFAR-10 and CIFAR-100, and the agnostic noise data sets, Clothing1M and WebVision17. The experimental results have demonstrated that the proposed model outperforms several state-of-the-art methods.

## 4.1 Introduction

Large scale datasets with editorial labels have driven the success of deep neural networks (DNNs) in computer vision [Krizhevsky et al., 2012], natural language processing [Sutskever et al., 2014], and speech recognition [Hinton et al., 2012]. However, for many real-world applications, it is usually expensive to collect accurately annotated data in large volume. Instead, samples with noisy supervision, as an alternative to alleviate the annotation burden, can be acquired inexhaustibly on the social websites and have shown potential to many applications in the deep learning area [Hu et al., 2017a; Yang et al., 2018a; Zhang et al., 2018b].

The challenging for DNNs in the setting of noisy supervision is that it can easily memorize the clean data as well as the noise data [Arpit et al., 2017]. To overcome the above problem, several methods have been explored from the perspective of model regularization and sample re-weighting, respectively. Arpit *et al.* [Arpit et al., 2017] applied the dropout regularization in DNNs to limit its speed of memorizing noise, which prevents the classifier from noise pollution. Ren *et al.* [Ma et al., 2018] explored dynamically weighting the noisy labels with the corresponding predictions to weaken noise. However, the model regularization and sample re-weighting based methods usually require either a careful hyperparameter setting [Arpit et al., 2017], or auxiliary samples [Ren et al., 2018] or empirical curriculums [Guo et al., 2018].

The study of this Chapter falls into the third popular perspective, learning with noise transition, which places a noise transition on top of the classifier. Early study [Patrini et al., 2017] presents a two-step solution, that is, first pre-estimate the noise transition and then fix it to train the classifier. However, it suffers from the inaccurate pre-estimation via an ideal but impractical anchor set. Subsequent improvement [Goldberger and Ben-Reuven, 2017] uses the stochastic approximation to adapt the noise transition in the form of a Softmax layer along with the training progress. Although it shows promise, the optimization of the Softmax layer depends on highly tweaking and the model parameters easily fall into undesired local minimums. Essentially, such instability is due to the inconsideration of the global dependency in the stochastic approximation, yielding a "local" mini-batch of samples can unbounded update the "global" noise transition in the back-propagation.

To solve this issue, we propose a Latent Class-Conditional Noise model (LCCN) that embeds the noise transition into a Dirichlet-distributed space. Compared to the previous Softmax layer [Goldberger and Ben-Reuven, 2017], LCCN constrains the learning of the noise transition as a global variable depending on the whole dataset. Namely, a "local" mini-batch of samples can only partially affect the "global" estimation of the noise transition. Besides, a dynamic label regression method is deduced to stochastically optimize LCCN. Although it iteratively infers the latent true labels and applies them for the classifier training and the noise modeling, only a small amount of extra computational cost is introduced. We theoretically demonstrate this

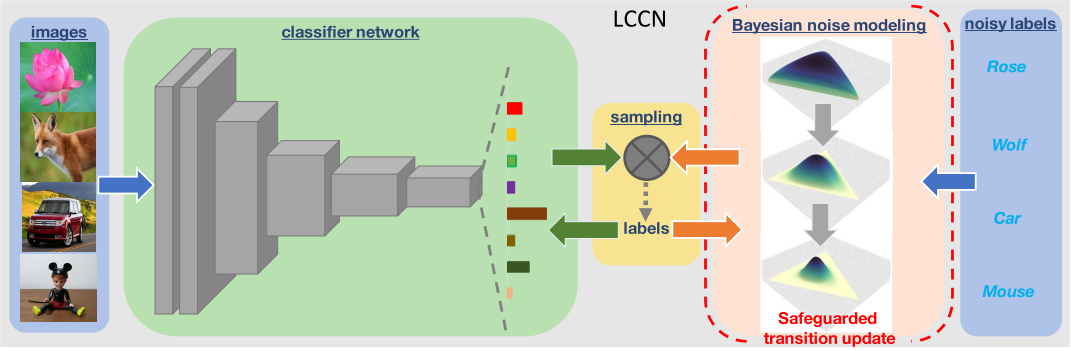



Figure 4.1: Safeguarded dynamic label regression for LCCN. The images and noisy labels are respectively input to the classifier and the safeguarded Bayesian noise modeling to compute the prediction and the conditional transition. Then, the latent true labels are sampled based on their product and then used for the classifier training and the safeguarded Bayesian noise modeling.

method safeguards the bounded update of the noise transition via a mini-batch of samples. Figure 4.1 provides a simple illustration of our safeguarded dynamic label regression for LCCN. As can be seen, images are first input to the classifier to have the prediction of latent true labels. Noisy labels are also forwarded to Bayesian noise modeling to compute the conditional transition of latent true labels. Then, the true labels are sampled based on their product and used to supervise the classifier training and refine the noise modeling. In a nutshell, our main contribution can be summarized into the following four points.

- We propose a Latent Class-Conditional Noise model that embeds the noise transition into a Dirichlet space to emphasize its global dependency, and then deduce a scalable dynamic label regression method for its optimization.
- The theoretical analysis on the convergence of the dynamic label regression, the generalization gap as well as the complexity is provided. Importantly, we prove that our optimization of the noise transition via a batch of samples is bounded to avoid arbitrarily tuning.
- A more general variant of LCCN is further extended in order to handle the open-set noisy labels setting and the semi-supervised learning setting for practical needs.
- We conduct a range of experiments in the small CIFAR-10, CIFAR-100 datasets and the large real-world noisy datasets Clothing1M and WebVision17. The comprehensive results have demonstrated the superior performance of our model compared with existing state-of-the-art methods.

The rest part of this Chapter is organized as follows. Section II briefly reviews the related research of learning with noisy labels in deep learning. Then, we introduce our Latent Class-Conditional model and the dynamic label regression method in Section III, where the corresponding theoretical analysis and the further extension of LCCN is also included. We validate the efficiency of our method over a range of experiments in Section IV. Section V concludes the whole Chapter.

## 4.2 The Proposed Framework

### 4.2.1 Preliminaries

In the c-class classification setting, a collection of $N$ noisy training pairs $\{(x_n, y_n)\}_{n=1}^N$ is given, where $x_n$ is the raw input data or the feature vector and $y_n \in \{1, \dots, K\}$ is the corresponding noisy label. Assume $z_n$ denotes the true label of $x_n$, which is unknown in practice. Then the goal in this task is to train a deep network classifier from the noisy dataset $\{(x_n, y_n)\}_{n=1}^N$ analogous to the one trained from the clean dataset $\{(x_n, z_n)\}_{n=1}^N$, so that a promising performance can be achieved in a clean test dataset. As shown in [Zhang et al., 2016], directly minimizing the following equation will make DNNs memorize both the classification pattern and noise,

$$\hat{f}_\theta = \arg\min_{f_\theta \in \mathcal{F}} -\frac{1}{N} \sum_{n=1}^{N} \ell(y_n, f_\theta(x_n)), \tag{4.1}$$

where $f_\theta$ is from the function class $\mathcal{F}$, which is parameterized by $\theta$ via DNNs, and $\ell$ is the loss function between $y_n$ and the prediction $f_\theta(x_n)$. Eq. (4.1) leads to a bad performance in the clean test dataset since it does not squeeze out the noise influence from $f_\theta$. Therefore, we follows one mainstream of approaches to handle this dilemma, which models a noise transition $\phi$ in simplex $\Delta$ when learning with noisy labels. The objective is then expressed with the following empirical risk minimization problem

$$\hat{f}_\theta, \phi = \arg\min_{f_\theta \in \mathcal{F}, \phi \in \Delta} -\frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \phi \circ f_\theta(x_n)), \tag{4.2}$$

[Patrini et al., 2017] theoretically demonstrate Eq. (4.2) trained with the noisy data shares the same minimizer with Eq. (4.1) trained with the clean data, if $\phi$ is accurately estimated. Unfortunately, it is usually impractical to acquire such a $\phi$ in advance. Thus, subsequent work [Goldberger and Ben-Reuven, 2017] adapts the pre-estimation with a Softmax layer along with the training progress. Although this shows a promising performance, expensive tweaking is required due to the ill-posed stochastic approximation as a simple neural layer.

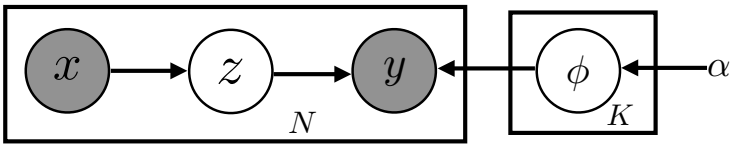



Figure 4.2: Latent Class-Conditional Noise model. $x$ and $y$ are the observed training pair. $z$ is the latent true label. $\phi$ is the unknown noise transition. $\alpha$ is the hyperparameter of the Dirichlet distribution. $N$ is the sample number and $K$ is the class number.

### 4.2.2 Latent Class-Conditional Noise model

In this section, we will present our Latent Class-Conditional Noise model (LCCN). Specifically, it avoids non-trivially tweaking for the fragile performance in [Goldberger and Ben-Reuven, 2017] by modeling $\phi$ in a Bayesian form. The graphical notation is illustrated in Figure 4.2 and the generative procedure is summarized as follows,

- The latent true label $z_n \sim P(\cdot|x_n)$, where $P(\cdot|x_n)$ is a *Categorical* distribution modeled by the deep neural network $f_\theta$ and the given $x_n$ is its input feature.
- The transition vector of the $k$th class $\phi_k \sim Dirichlet(\alpha)$, where $\alpha$ is the parameter of a $Dirichlet$ distribution and $[\phi_1, \cdots, \phi_K]^T$ constitutes the noise transition.
- The observed noisy label $y_n \sim P(\cdot|\phi_{z_n})$, where $P(\cdot|\phi_{z_n})$ is a *Categorical* distribution parameterized by $\phi_{z_n}$.

The general way to solve such a probabilistic model combined with deep learning is amortized variational inference [Kingma and Welling, 2014; Mnih and Gregor, 2014]. However, this way for LCCN will require an approximate Categorical reparameterization [Jang et al., 2017; Maddison et al., 2017] and introduce an unstable Digamma function to optimize. To avoid this issue, we specifically deduce a dynamic label regression method for optimization and demonstrate its safeguarded update on $\phi$.

### 4.2.3 Dynamic Label Regression

In the following, we will deduce a dynamic label regression method to optimize LCCN, which stacks autoencoded Gibbs sampling to infer the latent true labels and loss minimization for parameter learning. It naturally suits the case of LCCN and we show its deduction via a two-step formulation. Note that, in despite of the complex deduction, only a small computational cost is extra introduced, which will be explained in the following section. Simply, the first step is computing the probability of each

$z$ conditional on the others $Z^{\neg}$[1], i.e., $P(z_n|Z^{\neg n})$. Then, with the samples from $P(z_n|Z^{\neg n})$, the classifier training and the noise modeling can be explicitly decoupled as the following loss minimization problem,

$$\begin{cases} \min -\frac{1}{n}\sum_{n=1}^{N} \ell_1(z_n, P(z_n|x_n)) \\ \min -\frac{1}{n}\sum_{n=1}^{N} \ell_2(y_n, P(y_n|z_n)). \end{cases} \tag{4.3}$$

$\ell_1$ is the $\xi$-clipped cross-entropy loss[2] and $\ell_2$ is the likelihood loss. Alternating between the sampling of $P(z_n|Z^{\neg n})$ and the optimization of Eq. (4.3) constructs our final algorithm to learn with noisy supervision. Specifically, when $P(z|x)$ approach the true distribution of clean labels, the classifier training is similar to that on the clean dataset. This yields the asymptotically unbiased estimation as on the clean datasets.

**Autoencoded Gibbs sampling**. The *"autoencoded"* here origins from VAEs that combines variational inference [Wainwright et al., 2008] with deep learning. We claim it here is to express our method combines Gibbs sampling [Levin and Peres, 2017] with deep learning. The deduction of this method is as follows. Firstly, according to the aforementioned generative process, we can easily deduce the posterior of $z$ conditioned on the observed training pair $\{(x_n, y_n)\}_{n=1}^{N}$ and the Dirichlet parameter $\alpha$. This is implemented by factorizing the target conditional probability based on Figure 4.2 and applying the Bayes theorem as follows,

$$\begin{aligned} & P(Z|X, Y; \alpha) \\ &= \int_{\phi} \prod_{k=1}^{K} P(\phi_k; \alpha) \prod_{n=1}^{N} P(z_n|x_n, y_n, \phi) d\phi \\ &= \int_{\phi} \prod_{k=1}^{K} P(\phi_k; \alpha) \prod_{n=1}^{N} \frac{P(z_n|x_n)P(y_n|z_n, \phi)}{P(y_n|x_n)} d\phi \\ &= S * \int_{\phi} \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'}^{K} \alpha_{k'})}{\prod_{k'}^{K} \Gamma(\alpha_{k'})} \prod_{k'}^{K} \phi_{kk'}^{\alpha_{k'}-1} \prod_{n=1}^{N} \phi_{z_n y_n} d\phi, \end{aligned} \tag{4.4}$$

where $S$ represents $\prod_{n=1}^{N} \frac{P(z_n|x_n)}{P(y_n|x_n)}$ to simplify above equation. If we use the notation $N_{(\cdot)(\cdot)}$ to represent the confusion matrix of the noisy dataset, then we have $\sum_{k}^{K}\sum_{k'}^{K} N_{kk'}$=$N$ and $\prod_{n=1}^{N} \phi_{z_n y_n} = \prod_{k}^{K}\prod_{k'}^{K} \phi_{kk'}^{N_{kk'}}$. Putting the later equation into Eq. (4.4) and then using the conjugation characteristic between the Dirichlet distribution and the Multinomial distribution, the following form can be further

[1]Note that $\neg$ means removing the current object statistic from the whole collection of all object statistics.

[2]The probabilistic prediction from the model is clipped between $\xi$ and $1-\xi$ for the computational stability, where $\xi$ is set to $10^{-20}$ in this Chapter.

deduced,

$$
\begin{aligned}
&P(Z|X,Y;\alpha) \\
&= S * \int_{\phi} \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'}^{K} \alpha_{k'})}{\prod_{k'}^{K} \Gamma(\alpha_{k'})} \prod_{k'}^{K} \phi_{kk'}^{N_{kk'}+\alpha_{k'}-1} d\phi \\
&= S * \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'}^{K} \alpha_{k'})}{\prod_{k'}^{K} \Gamma(\alpha_{k'})} \prod_{k=1}^{K} \frac{\prod_{k'}^{K} \Gamma(\alpha_{k'} + N_{kk'})}{\Gamma(\sum_{k'}^{K} (\alpha_{k'} + N_{kk'}))}.
\end{aligned} \tag{4.5}
$$

Unfortunately, Eq. (4.5) is non-analytical and cannot be used to generate the samples of $z$ directly, which can be solved by Gibbs sampling. According to the Gibbs sampling, we need to compute $P(z_n|Z^{\neg n})$ first. And then based on $P(z_n|Z^{\neg n})$, a sequence of observations can be sampled, which are approximately from $P(z_n|x_n, y_n, \phi)$. The following deduction facilitates Eq. (4.5) and $\Gamma(x+1) = x\Gamma(x)$ to acquire the final conditional probability for the autoencoded Gibbs sampling.

$$
\begin{aligned}
P(z_n|Z^{\neg n}, X, Y; \alpha) &= \frac{P(Z|X,Y;\alpha)}{P(Z^{\neg n}|X,Y;\alpha)} \\
&= \frac{P(z_n|x_n)}{P(y_n|x_n)} \frac{\alpha_{y_n} + N_{z_n y_n}^{\neg n}}{\sum_{k'}^{K} (\alpha_{k'} + N_{z_n k'}^{\neg n})} \\
&\propto \underbrace{P(z_n|x_n)}_{\text{Classifier encoder}} \underbrace{\frac{\alpha_{y_n} + N_{z_n y_n}^{\neg n}}{\sum_{k'}^{K} (\alpha_{k'} + N_{z_n k'}^{\neg n})}}_{\text{Conditional transition}}.
\end{aligned} \tag{4.6}
$$

With Eq. (4.6), we can sample a collection of latent true labels $\{z_n\}$. Such samples are then used to solve the optimization problem in Eq. (4.3). Iterating the procedure of Eq. (4.6) and Eq. (4.3), we gradually approach the true latent label, and at the same time train the classifier and estimate the noise transition.

**Lemma 4.1.** *For a reversible, irreducible and aperiodic Markov chain with state space $\Omega$, let $\lambda^*$ be the maximal absolute eigenvalue of the state transition matrix and $\pi$ be the underlying stationary probability measure where $\pi_{min} = \min_{Z\in\Omega} \pi(Z)$. Then, the $\epsilon$-mixing time from the initial arbitrary state to the equilibrium is characterized by the following bounds,*

$$
\frac{\lambda^*}{1-\lambda^*} \ln\left(\frac{1}{2\epsilon}\right) \leq \tau_{mix}(\epsilon) \leq \frac{1}{1-\lambda^*} \ln\left(\frac{1}{\pi_{min}\epsilon}\right), \tag{4.7}
$$

*where $\tau_{mix}(\epsilon) = \min\{t : ||P_t(Z) - \pi||_{TV} \leq \epsilon\}$ and $||\cdot||_{TV}$ is the total variation distance between two probability measures.*

The above lemma indicates [Levin and Peres, 2017] the mixing time of LCCN is at most constantly linear to the inverse of $1 - \lambda^*$. Although it is hard for Gibbs sampling to accurately quantify $\lambda^*$ due to the evolving state transition matrix, the recent work [Johan, 2017] shows Gibbs sampling is efficient enough and almost proportional to the logarithm of the dataset size $N$. In experiments, we will show LCCN is well converged after same epochs as baselines.

In statistical learning theory, the *excess risk* [1] and *the error bound w.r.t.* the expected risk and Bayes risk, are two important quantities to measure model generalization performance. In the setting of noisy labels, such two quantities are bounded by the following generalization bound (see the Appendix A)

$$\Delta_{\mathcal{F}} = \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(z, f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta(x))\right] \right|,$$

where $\mathbf{E}[\cdot]$ and $\mathbf{E}^{(D_N)}[\cdot]$ respectively represents the expectation on the clean data distribution and the empirical estimation with the data whose labels are from the Gibbs sampling. Thus, by analyzing the upper bound of $\Delta_{\mathcal{F}}$, we can then understand which factors affect the generalization performance of LCCN. Specifically, we deduce the following theorem to interpret this.

**Theorem 4.2.** *Assume $f_\theta^*$ and $f_\theta^\dagger$ respectively are the underlying groundtruth labeling functions $\mathcal{X} \to \mathcal{Y}$ of clean test data and data from the Gibbs sampling. Define the composite function class $\mathcal{G} = \{x \mapsto \ell_1(f'_\theta(x), f_\theta(x)) : f'_\theta, f_\theta \in \mathcal{F}\}$. Then, for any probability $\delta > 0$, with probability at least $1 - \delta$,*

$$\Delta_{\mathcal{F}} \leq \Delta + \widehat{\mathcal{R}}(\mathcal{G}) + 3\rho\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2N}} \tag{4.8}$$

*where $\Delta = \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta^*(x) - f_\theta^\dagger(x), f_\theta(x))\right] \right|$, $\widehat{\mathcal{R}}(\mathcal{G})$ is the Rademacher complexity [Bartlett and Mendelson, 2002] of $\mathcal{G}$ and $\rho$ is the maximum of the $\xi$-clipped cross entropy loss, i.e., $-\ln \xi$.*

The above theorem indicates the generalization performance of the classifier learned by LCCN depends upon three factors, i.e., the inherent gap $\Delta$ between the noisy training domain and the clean test domain, the function complexity $\widehat{\mathcal{R}}(\mathcal{G})$ and the sample number $N$. In particular, if LCCN can exactly infer all the true labels of noisy data and eliminate the domain bias, we will have $f_\theta^* = f_\theta^\dagger$ and $\Delta = 0$. In this case, Eq. (4.8) will degenerate to the Rademacher bound [Mansour et al., 2009] after scaling the loss to $[0, 1]$, and equal to the training on the clean data. However, it is usually hard to completely remove the domain bias, since the distribution of the

[1] https://en.wikipedia.org/wiki/Risk_difference

corrected samples could still be different from that of the clean test data. For example, the web training data may contain many outlier classes. Thus, $\Delta$ is an important factor to the generalization performance of LCCN.

### 4.2.4 Safeguarded Transition Update

In this section, we will show that our method safeguards the bounded update of the noise transition by a batch of samples, avoiding the arbitrarily tuning via a Softmax layer.

**Theorem 4.3.** *Suppose $\alpha_i$ is a positive smoothing scalar, $N_i$ is the current sample number of the $i$th category (i=1,...,K), $M_i$ is the sum of the sample numbers newly allocated into (positive) and removed from (negative) the $i$th category after a batch of training samples, and $\widehat{M}_i$ is its absolute sum of such two cases. Then, for the transition vector $\phi_i$ of the $i$th category, its variation via a training batch is characterized by the following equation,*

$$\left|\phi_i^{new} - \phi_i^{old}\right| \leq \frac{|r_i| + \widehat{r}_i}{1 + r_i} \tag{4.9}$$

*where $r_i = \frac{M_i}{N_i + \sum_{j=1}^{K} \alpha_j}$ and $\widehat{r}_i = \frac{\widehat{M}_i}{N_i + \sum_{j=1}^{K} \alpha_j}$. According to the definition, we have $r_i > -1$, $\widehat{r}_i \geq 0$ and $\widehat{r}_i \geq |r_i|$.*

*Proof.* The variation of $\phi_i$ after a training batch is,

$$\begin{aligned}
&\left|\phi_i^{new} - \phi_i^{old}\right| \\
&= \sum_{j=1}^{K} \left|\phi_{ij}^{new} - \phi_{ij}^{old}\right| \\
&= \sum_{j=1}^{K} \left| \frac{N_{ij} + \alpha_j + M_{ij}}{N_i + \sum_{j'=1}^{K} \alpha_{j'} + M_i} - \frac{N_{ij} + \alpha_j}{N_i + \sum_{j'=1}^{K} \alpha_{j'}} \right| \\
&\leq \sum_{j=1}^{K} \frac{\left|(N_i + \sum_{j'=1}^{K} \alpha_{j'}) M_{ij}\right| + \left|(N_{ij} + \alpha_j) M_i\right|}{(N_i + \sum_{j'=1}^{K} \alpha_{j'})(N_i + \sum_{j'=1}^{K} \alpha_{j'} + M_i)} \\
&= \frac{(N_i + \sum_{j'=1}^{K} \alpha_{j'}) \widehat{M}_i + (N_i + \sum_{j'=1}^{K} \alpha_{j'}) \left|M_i\right|}{(N_i + \sum_{j'=1}^{K} \alpha_{j'})(N_i + \sum_{j'=1}^{K} \alpha_{j'} + M_i)} \\
&= \frac{|r_i| + \widehat{r}_i}{1 + r_i}
\end{aligned} \tag{4.10}$$

□

**Algorithm 2** Dynamic Label Regression for LCCN

**Input** a noisy dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, a classifier $P(\cdot|x)$ modeled by DNN $f_\theta$, warming-up steps $\delta$, the running epoch number $L$ and the batch-size $M$.
Directly pretrain the classifier $f_\theta$ on the noisy dataset $\mathcal{D}$.
Compute the warming-up noise transition matrix $\phi'$.
**for** *epoch* $i = 1$ *to* $L$ **do**
    Let step=i$\times\lceil N/M \rceil$+j and hook a batch of samples.
    **if** *step* $< \delta$ **then**
        Substitute the transition in Equation (4.6) with $\phi'$, and then sample $z_n$ for each $x_n$ in the batch.
    **else**
        Sample $z_n$ with Equation (4.6) for the batch.
    **end**
    Update the confusion matrix $N_{(\cdot)(\cdot)}$ based on the existing sampling observations $\{(z_n, y_n)\}$.
    Optimize Equation (4.3) to learn the classifier $f_\theta$ and estimate the noise transition matrix $\phi$.
**end**
**Output** the classifier $f_\theta$ and the noise transition $\phi$.

**Corollary 1.** *Suppose $M$ is the batch size in the training. If it satisfies the condition $M \ll N_i$, we have $\widehat{r}_i < \frac{M}{N_i}$ in a small scale. Then the variation of $\phi_i$ after a training batch will be bounded by $\frac{|r_i|+\widehat{r}_i}{1+r_i} \leq \frac{2\widehat{r}_i}{1-\widehat{r}_i} \approx 2\widehat{r}_i$ in a small scale.*

The core drawback in [Goldberger and Ben-Reuven, 2017] is the noise transition modeled by a Softmax layer can be arbitrarily updated via a batch of samples. This is because the gradients of the parameters estimated by a "local" batch can be arbitrarily large in the backpropagation. Then, the noise transition decided by the "global" dataset might be pushed into a bad local minimum by a batch of some extremely noisy training samples, yielding a serious harm on the classifier. The later experimental analysis in Figure 4.6 will confirm this point. Instead, our dynamic label regression theoretically safeguards the bounded update of the noise transition via a batch of samples. Specifically, with the bounded update, the conditional transition in Equation (4.6) is gradually changing towards at a true distribution when the classifier is well trained. Similarly, with more reliable sampled labels, the classifier is better trained and the noise modeling is refined. Finally, we acquire a virtuous cycle.

### 4.2.5 Complexity Analysis

The learning procedure is summarized in Algorithm 2. Note that, we give the implementation including details like pretraining and warming-up in the experiments.

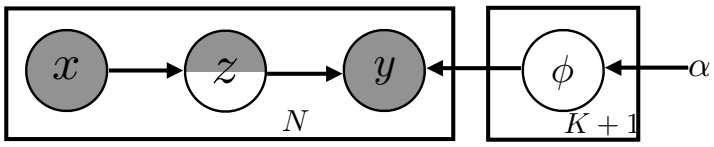



Figure 4.3: The Generalized Latent Class-Conditional Noise model. $x$ and $y$ is the observed training pair. $z$ is the partially observed latent label (the observed samples are for semi-supervised learning). $\phi \in \mathbb{R}^{(K+1)xK}$ is the unknown noise transition (the extra dimension is for outlier learning). $\alpha$ is the hyperparameter of the Dirichlet distribution. $N$ is the sample number and $K$ is the class number.

As we know, the stochastic optimization of a DNN model involves two steps, the forward and backward computations. In each mini-batch update, its time complexity is $\mathcal{O}(M\Lambda)$, where $M$ is the mini-batch size and $\Lambda$ is the parameter size. Here, in Algorithm 2, we additionally add a sampling operation via Eq. (4.6) whose complexity is $\mathcal{O}(M+K^2)$ ($K$ is the class size). Note that, the first term in the RHS of Eq. (4.6) has been computed in the forward procedure. Since $M$ and $K$ is usually significant smaller than $\Lambda$, the extra cost for the sampling is negligible compared to $\mathcal{O}(M\Lambda)$. Besides, the optimization for noise modeling in Eq. (4.3) can be ignored, as this only involves the normalization of a confusion matrix whose complexity is $\mathcal{O}(K^2)$. In total, since the big-O complexity of each mini-batch remains the same, our method is scalable to big data.

## 4.3 Extensions on Outlier and Semi-supervised Learning

In this section, we will extend LCCN to the generalized version in Figure 4.3, which shares the optimization procedure but is more useful in real-world applications.

### 4.3.1 Extension on Outlier Learning

In practise, datasets collected from online websites or real-world scenarios, usually contains the open-set label noise [Wang et al., 2018]. That means data from other distributions might be disturbed as the given class samples involving in the training. Previous class-conditional noise model [Goldberger and Ben-Reuven, 2017; Patrini et al., 2017; Reed et al., 2014; Sukhbaatar et al., 2014; Xiao et al., 2015], mainly focus on the closed-set label perturbation and thus can not handle the outlier classes. For example, it is impossible to estimate the transition matrix via the mentioned two-step formulation in [Patrini et al., 2017] since this requires the selection of the

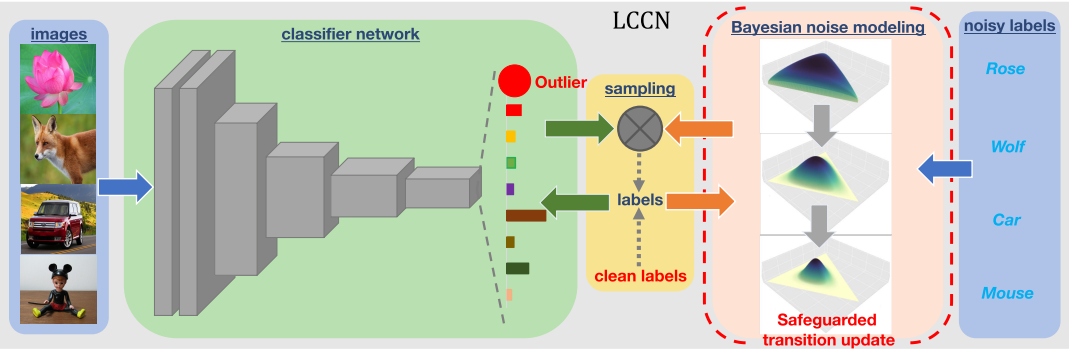



Figure 4.4: Extensions based on the original safeguarded dynamic label regression for LCCN. The images and noisy labels are respectively input to the classifier and the safeguarded Bayesian noise modeling to compute the prediction (including the outlier component) and the conditional transition. When the clean labels are not available as the latent labels, they are sampled based on the product of previous two quantities, and otherwise they keep unchanged. Then, the latent labels composite by both the clean parts and those inferred from sampling are used to train the classifier. In addition, only the latent labels inferred from the Gibbs sampling are used to refine the Bayesian noise model.

representative outlier samples. Similarly, learning the transition matrix by a noise adaptation layer [Goldberger and Ben-Reuven, 2017] still suffers from the instability. Therefore, it is useful to extend LCCN to deal with this open-set noisy label setting. Actually, the modification for LCCN only requires to add an outlier choice for the latent variable $z$, i.e., $z \in \{1, ..., K, K+1\}$, where $K+1$ indexes the collapsed outlier classes and then change the noise transition from $\mathbf{R}^{KxK}$ to $\mathbf{R}^{(K+1)xK}$. The model modification is shown in Figure 4.3 and the the network modification is illustrated in Figure 4.4. Importantly, the above modifications do not alter the aforementioned deduction.

### 4.3.2 Extension on Semi-supervised Learning

It is common to improve the model performance by augmenting the large scale noisy dataset with a small set of clean samples. Many works [Guo et al., 2018; Patrini et al., 2017; Ren et al., 2018; Veit et al., 2017; Xiao et al., 2015] have leveraged such a semi-supervised setting to calibrate the classifier and achieve a better result. In our model, it is naturally compatible with this case, where we can directly utilize the clean labels instead of labels from sampling when they are available. As illustrated in Figure 4.4, this configuration is specifically marked in red in parallel to sampling. The corresponding model modification is indicated in Figure 4.3. In a broad sense,

clean labels can be as accurate as a given category or as weak as a coarse hint that tells outlier or not. In the former case, a standard cross entropy loss can be applied to the classifier; while in the latter case, a collapsed cross entropy loss that defines on outliers *vs.* non-outliers could be applied. Since this is tightly related to the work on domain adaptation [Csurka, 2017], we leave the latter case in the future.

## 4.4 Experiments

The experiments involve both the simulated noisy datasets and the real-world noisy datasets. We verify the performance of our model and state-of-the-art methods.

### 4.4.1 Datasets and Baselines

**Datasets**. We conduct the toy experiments on CIFAR-10, CIFAR-100 and the real-world experiments on Clothing1M and WebVision. The details about the first three datasets can refer to the previous Chapter and here we give the brief introduction of the last one. WebVision[1] [Li et al., 2017a] is a more challenging noisy dataset, which contains more than 2.4 million images. It is crawled from the Internet by using the 1,000 concepts of ILSVRC [Deng et al., 2009] as queries. In addition, a clean validation set which contains 50,000 annotated images, are provided to boost and validate the proposed models in diverse applications. We use the validation set of ImageNet [Deng et al., 2009] as its test set. For the toy experiments without outliers, we inject the asymmetric noise to disturb their labels to form the noisy datasets. Concretely, on CIFAR-10, we set a probability $r$ to disturb the label to its similar class, i.e., truck $\rightarrow$ automobile, bird $\rightarrow$ airplane, deer $\rightarrow$ horse, cat $\rightarrow$ dog. For CIFAR-100, a similar $r$ is set but the label flip only happens in each super-class. The label is randomly disturbed into the next class circularly within the super-classes. For the toy experiments that consider the open-set noisy labels, we randomly select 10,000 samples from the original datasets and shuffle the order of the pixel values as the outliers. In the semi-supervised learning, we utilize the clean labels of the first 5,000 clean samples and the first 500 outlier samples for the training.

**Baselines**. For the toy experiments, we compare *LCCN* with the classifier that is directly trained on the dataset (termed as *CE*), the method *Bootstrapping* proposed in [Reed et al., 2014], the transition based method *Forward* [Patrini et al., 2017] and the method that fine-tunes the transition *S-adaptation* [Goldberger and Ben-Reuven, 2017]. Note that, we choose the hard mode for Bootstrapping, since it is empirically better than the soft mode. Besides, we also use the *MASKING* as one

[1]Due to the reason of time and the computational resource, in this Chapter, we only use the original WebVision 1.0 dataset. The newest version, namely WebVision 2.0 dataset, contains more images and more classes.

baseline in Clothing1M. In the outlier corrupted datasets, we denote the extension of our model that considers the outlier as *LCCN**. In the semi-supervised learning setting, we denote our model as *LCCN+*, meaning the clean samples are used in the training. For the experiments on real-world datasets, we also report the result of *Joint Optimization* [Tanaka et al., 2018] that leverages the auxiliary noisy label distribution and the state-of-the-art result *Forward+* [Patrini et al., 2017] that finetunes on clean samples.

### 4.4.2 Implementation Details

For CIFAR-10 and CIFAR-100, the PreAct ResNet-32 [He et al., 2016] is adopted as the classifier. The image data is augmented by horizontal random flip and 32×32 random crops after padding with 4 pixels. Then, the per-image standardization is used to normalize pixel values. For the optimizer, we utilize SGD with a momentum of 0.9 and a weight decay of $0.0005$. The batch size is set to 128. The training runs totally 120 epochs and is divided into three phases in 40 and 80 epochs. In these three phases, we respectively set the learning rate as 0.5, 0.1 and 0.01. Note that, the reason that we adopt a large learning rate (others may set smaller than 0.001), is that the small learning rate will lead to overfitting on the noisy dataset as claimed in [Arpit et al., 2017]. Following the benchmark in [Patrini et al., 2017], we use CE to initialize the classifier in other baselines and LCCN. For S-adaptation, the following transition is computed to warm-up the transition parameters in the first 80 epochs.

$$\phi'_{ij} = \frac{\Sigma_t 1_{y_t=j} p(z_t = i|x_t)}{\Sigma_t p(z_t = i|x_t)} \tag{4.11}$$

Similarly on CIFAR-10, we use above transition to warm up the sampling procedure in LCCN for the first 20,000 steps. However, on CIFAR-100, we set $\phi'_{ij} = 1[i = j]$ in the warming-up since Equation (4.11) will induce the high sampling variance.

For Clothing1M and WebVision, the ResNet-50 is leveraged as the classifier. We resize the short side of their images to 224 and do the random crop of 224×224. The training images are augmented with the random flip, whiteness and saturation. For the optimizer, we deploy SGD with a momentum of 0.9 with a weight decay of $10^{-3}$. The batch size for Clothing1M is set to 32 and we fix the learning rate as 0.001 to run 5 epochs. For the warming-up transition, we both validate the one [Xiao et al., 2015] from manual annotation and the one estimated by Equation (4.11) for 40,000 steps. Note that, on the large real-world datasets, due to the strong capacity of ResNet-50, it is easy for LCCN to occur the sampling collapsed problem, i.e., the sampled latent label is identical to the noisy label. Thus, we norm Equation (4.6) with a power annealed coefficient $\max\{\exp\left\{(-\frac{\text{step}}{\text{max\_step}} * 0.8)\right\}, 0.5\}$ to introduce the sufficient perturbation in avoid of this issue. On WebVision, the batch size is set to 128, and the learning

| Dataset | | CIFAR-10 | | | | | CIFAR-100 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # | Method \ Noise Ratio | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 1 | CE | 90.10 | 88.12 | 76.93 | 59.01 | 56.85 | 66.15 | 64.31 | 60.11 | 51.68 | 33.37 |
| 2 | Bootstrapping | 90.73 | 88.12 | 76.29 | 57.04 | 56.79 | 66.48 | 64.61 | 63.01 | 55.27 | **34.52** |
| 3 | Forward | 90.86 | 89.03 | 82.47 | 67.11 | 57.29 | 65.43 | 62.72 | 61.28 | 52.64 | 33.82 |
| 4 | S-adaptation | 91.02 | 88.83 | 86.79 | 72.74 | 60.92 | 65.52 | 64.11 | 62.39 | 52.74 | 30.07 |
| 5 | LCCN | **91.35** | **89.33** | **88.41** | **79.48** | **64.82** | **67.83** | **67.63** | **66.86** | **65.52** | 33.71 |
| 6 | CE with the clean data | 91.63 | | | | | 69.41 | | | | |

Table 4.1: The average accuracy (%) over 5 trials on CIFAR-10 and CIFAR-100 with different noise.

| Dataset | | CIFAR-10 | | | | | CIFAR-100 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # | Method \ Noise Ratio | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 1 | CE | 89.13 | 87.06 | 74.63 | 62.29 | **57.07** | 62.94 | 59.73 | 54.71 | 45.57 | 31.74 |
| 2 | Bootstrapping | **90.13** | 84.58 | 74.76 | 54.87 | 55.56 | 63.73 | 60.88 | 59.77 | 40.23 | 31.86 |
| 3 | Forward | 88.63 | 84.97 | 78.47 | 58.23 | 56.52 | 63.69 | 62.63 | 61.86 | 51.47 | **35.71** |
| 4 | S-adaptation | 88.58 | 87.28 | 61.17 | 57.12 | 56.73 | 63.51 | 61.50 | 60.59 | 53.22 | 32.19 |
| 5 | LCCN | 88.63 | 88.06 | 82.15 | 69.48 | 55.12 | 63.97 | 62.84 | 61.79 | 60.34 | 33.52 |
| 6 | LCCN* | 89.59 | **88.43** | **84.34** | **72.33** | 56.28 | **64.71** | **63.05** | **62.48** | **62.02** | 32.37 |
| 7 | LCCN+ | 90.30 | 88.93 | 88.21 | 87.42 | 86.33 | 65.67 | 64.24 | 63.52 | 63.19 | 62.39 |

Table 4.2: The average accuracy (%) over 5 trials on outlier-corrupted CIFAR-10 and CIFAR-100.

rate is initialized with 0.1 is divided by 10 every 30 epochs until 90 epochs. We use the diagonal transition for 10,000 steps of warming-up and then update the confusion matrix to the end, since it contains 1,000 categories. The similar power-annealed strategy for sampling is leveraged. Finally, to fairly compare LCCN+ and Forward+ in semi-supervised learning, we use the similar fine-tuning in [Patrini et al., 2017] to run.

### 4.4.3 Results on CIFAR10 and CIFAR-100

**Classification experiments**. Table 4.1 summarizes the performance of LCCN and baselines on the noisy datasets without outilers. Compared with the baselines, LCCN achieves the best performance at most of noise levels. In particular, even in the large noise rates, our model still acquires the competitive classification accuracy. For example, when $r$=0.7 on CIFAR-10 and $r$=0.4 on CIFAR-100, LCCN reaches $79.48\%$ and $65.52\%$, outperforming the best results of baselines by about $7\%$ and $13\%$ respectively. This demonstrates that our model is significantly better than baselines. Regarding $r$=0.5 on CIFAR-100, the way to disturb the labels [Patrini et al., 2017] leads that there is one undesired minimum, since two classes are mixed into one class by equal quota after injecting noise. In this case, it is hard to say which model can achieve the best result. We include it here for the complete comparison as in other works [Goldberger and Ben-Reuven, 2017; Patrini et al., 2017].

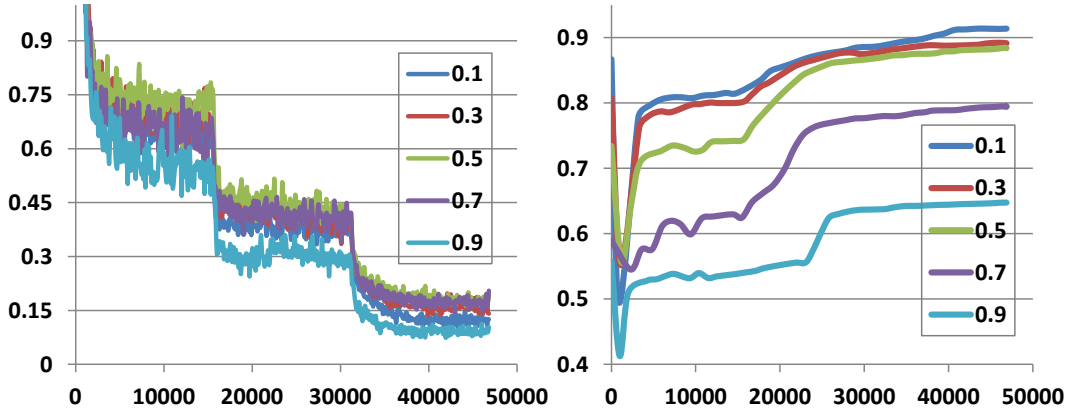



Figure 4.5: The loss curve (the left panel) and the accuracy curve (the right panel) of LCCN in the training procedure on the CIFAR-10 dataset with the different noise rates $r = 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9$.

Table 4.2 presents the results of LCCN and baselines on the outlier-corrupted datasets. Compared to those in Table 4.1, all the methods have a slight performance drop. Nevertheless, LCCN achieved the best performance in such an setting at $r$=0.3, 0.5, 0.7 on CIFAR-10 and $r$=0.1, 0.2, 0.3 and 0.4 on CIFAR-100. The baselines without the outlier detection mechanism usually have a significant degeneration. Specifically, on CIFAR-10, all the other baselines are even not better than CE, i.e., directly training. Instead, LCCN* that considers the outlier achieves a further improvement based on LCCN. Besides, as expected, by adding clean data, LCCN+ performs better than LCCN*, since the classifier is calibrated by the information of the clean data domain. In the scenario of the extreme noise, as marked by the grey color in Table 4.2, this is quite useful and even necessary to guarantee an acceptable performance. In total, according to the quantitative analysis of Table 4.1 and Table 4.2, we demonstrate the superiority of LCCN on toy datasets.

**On convergence visualization**. In Figure 4.5, we trace the training of LCCN on CIFAR-10 to visualize its convergence. As can been in the left panel of Figure 4.5, LCCN has a stable convergence on loss after the given epochs. Besides, we find the loss converges to irregular scales in different noise rates. Concretely, in most cases, i.e., $r$=0.1, 0.3, 0.5, the final training loss increases as $r$ increases, while the loss shows attenuation as $r$ >0.5. It is because in the low-level noise, the model can easily correct the labels via the sampling in LCCN, yielding a small loss. While in the case of the extreme noise, it is more challenging to prevent the model from fitting on noise, which incurs difficulty for optimization and thus achieves a big loss. Furthermore, according to the right penal of Figure 4.5, the test accuracy also approximately converges and persists to the end of the training without performance drop. Actually, it is not a

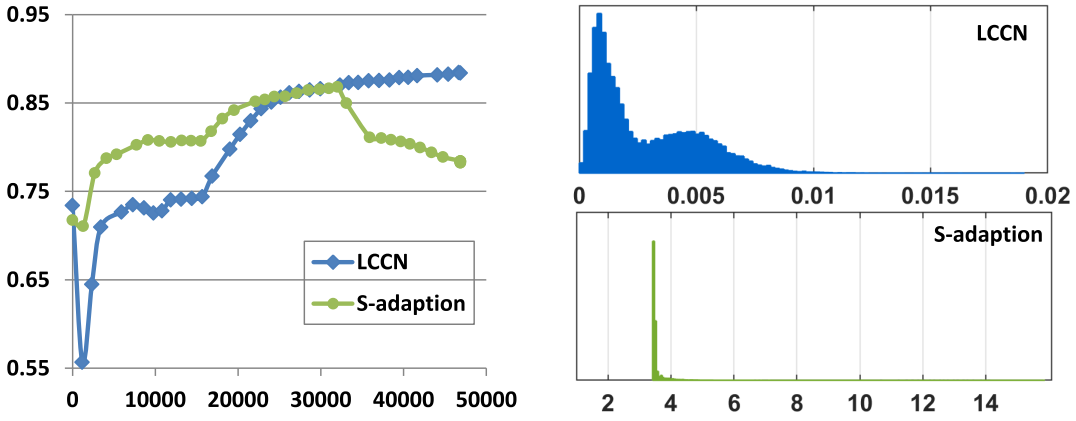



Figure 4.6: The test accuracy of LCCN and S-adaptation in the training on CIFAR-10 with $r$=0.5 and the corresponding histograms for the change of noise transition $\phi$ via a mini-batch of samples.

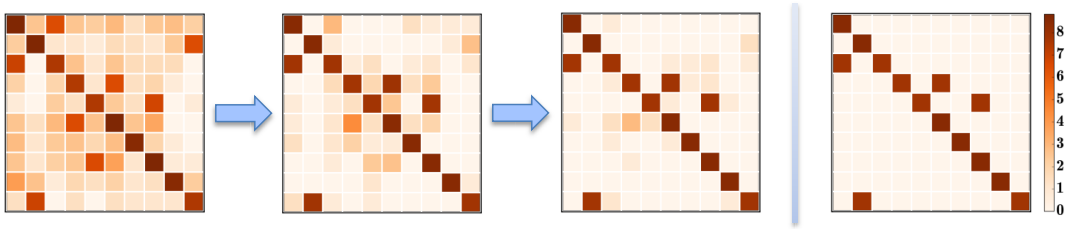



Figure 4.7: The colormap of the confusion matrix on CIFAR-10 with r=0.5. We utilize the log-scale for each element in the confusion matrix for the fine-grained visualization. The left three maps are respectively learned by LCCN at the beginning of the training, the 30,000th step and the end of the training. The most right one is the groundtruth disturbed from the original clean CIFAR-10 dataset.

common phenomenon for previous methods to own this merit, since all baselines tends to overfitting on noise more or less in the final few epochs. This demonstrates the advantages of LCCN in the robust training with the noisy datasets.

**Safeguarded transition update**. To show LCCN safeguards the noise transition update compared to S-adaptation, we compute the statistics about their update of noise transition on CIFAR-10 at $r$=0.5, and illustrate the histogram of changes in Figure 4.6. Firstly, from the left panel of Figure 4.6, we can see that there is a significant performance drop in the training of S-adaptation. The clue to this phenomenon can be found by inspecting the update of noise transition. As shown in the right panel of Figure 4.6, the change magnitude of $\phi$ in S-adaptation is higher than that of LCCN. One is in a large scale ranging from 0 to 16, while the other one is in a very small scale ranging from 0 to 0.02. This leads to S-adaptation suffering from a high risk of

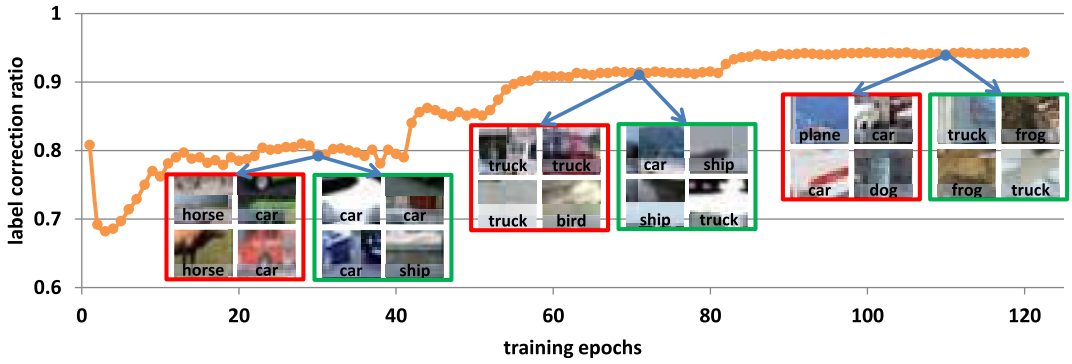



Figure 4.8: The label correction ratio in the training of LCCN on CIFAR-10 with $r$=0.5. We illustrate some negatively corrected samples (the red box) and some positively corrected samples (the green box) based on the high probabilities to be right or wrong at the 30th, the 70th and the 110th epochs.

over-tuning to undesired local minimums in the presence of noise. Instead, according to the histogram, LCCN updates $\phi$ in a safeguarded small scale when approaching to the minimum. In summary, this quantitative analysis confirms the claim of our Theorem 4.3 in the perspective of experiments.

**The latent label and noise analysis**. Figure 5.9 and Figure 4.8 respectively depict the colormap of the confusion matrix and the label correction ratio when training LCCN on CIFAR-10 with $r$=0.5. First, as can be seen in Figure 5.9, the initial confusion matrix does not approach the true matrix and there are many incorrect entries. However, with the training progressing, the matrix is gradually corrected and at the end of training, it is approximately similar to the provided groundtruth. Besides, As shown in Figure 4.8, the ratio of the image with the correct label increases along with the training progress. This reflects LCCN successfully models the class-conditional noise and gradually infer the true labels. Specially, by visualizing the mis-corrected examples in the training process, we can find that the classifier at first make mistakes in even some simple samples, while finally has the wrong classification in only the hard examples. These two figures visualize how dynamic label regression infers the latent label and models the noise.

### 4.4.4 Results on Clothing1M and WebVision

Table 4.3 lists the performance of LCCN and baselines on the large-scale Clothing1M. According to the results, we can see that Forward does not show the significant improvement in this dataset, even though they use the annotated noise transition matrix [Xiao et al., 2015]. And S-adaptation only improves Forward by 0.5%. Joint Optimization that trains the classifier with label correction [Tanaka et al., 2018]

Figure 4.9: Some representative samples in the training set that are considered as the outliers by LCCN*. We intuitively summarize these photos into four categories based on their contents, multiple different objects (RED), implicit categories (GREEN), uncertain types (BLACK) and confusing appearance (BLUE), which are respectively marked by the color of the surrounded boxes. Outliers are relative to LCCN and may contain hard examples of the pre-defined 14 categories.

| # | Method | Accuracy |
|---|---|---|
| 1 | CE | 68.94 |
| 2 | Bootstrapping | 69.12 |
| 3 | Forward | 69.84 |
| 4 | S-adaptation | 70.36 |
| 5 | MASKING | 71.10 |
| 6 | Joint Optimization | 72.16 |
| 7 | LCCN<br>LCCN warmed-up by $\phi$ in [Xiao et al., 2015]<br>LCCN* | 71.63<br>**73.07**<br>**72.80** |
| 8 | CE on the clean data | 75.28 |
| 9 | Forward+ | 80.38 |
| 10 | LCCN+ | **81.25** |

Table 4.3: The average accuracy over 5 trials on Clothing1M.

achieves better results than the other baselines. Nevertheless, this method requires the provided noisy label distribution to prevent degeneration and is not scalable to the large number of classes [Tanaka et al., 2018]. Instead, LCCN that contains both the label correction and Bayesian noise modeling, gets the competitive performance 71.63%. With the warming-up of the auxiliary noise transition [Xiao et al., 2015], it further achieves the best 73.07%. Even although there is no auxiliary information available, our extension LCCN* still outperforms the current state-of-the-art result. This demonstrates the potential of LCCN in handling the real-world noisy dataset. In addition, the results of LCCN+ indicates our model also has the advantages in the semi-supervised learning. Note that, in Table 4.3 LCCN outperforms MASKING. This is

| class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 77 | | 6 | | | 4 | | | | | | | | | T-shirt |
| 2 | 2 | 88 | | 3 | | | | | | | | | | | shirt |
| 3 | 12 | 4 | 48 | 4 | 7 | 7 | | | | | | 5 | | | knitwear |
| 4 | 9 | 15 | 5 | 51 | | | | | | | | 9 | | | chiffon |
| 5 | 11 | 4 | 43 | | 16 | 9 | | | | | | 4 | | | sweater |
| 6 | 3 | | 3 | | | 86 | | | | | | | | | hoodie |
| 7 | | | | | | | 87 | 3 | | 5 | | | | | windbreaker |
| 8 | | | | | | | | 92 | | | | | | | jacket |
| 9 | | | | | | | 2 | | 96 | | | | | | downcoat |
| 10 | | | | | | | 4 | 2 | | 91 | | | | | suit |
| 11 | | | 3 | | | | | | | | 81 | 3 | | | shawl |
| 12 | | | | 3 | | | | | | | | 84 | 3 | | dress |
| 13 | | | | 3 | | | | | | | | 6 | 76 | 3 | vest |
| 14 | 1 | | | | | | | | | | | | | 95 | underwear |
| 15 | 12 | 10 | | 7 | | 9 | 6 | 4 | | 7 | | 14 | 6 | 3 | outlier |

Table 4.4: The learned noise transition on Clothing1M by LCCN.

because in Chapter 3, we have no transition groundtruth avaiable for Clothing1M but adopt the alternative transition manually provided by [3] to extract the mask. The provided transition may not be very accurate compared to the one learnt by LCCN, which yields a slightly lower performance. However, as the claimed in the introduction of this Chapter, we target to iteratively refine the noise transition without considering auxiliary information, namely a different application setting, and thus we have not compared to LCCN with MASKING. In fact, if we re-conduct the experiment for MASKING with the mask extracted from the noise transition learnt by LCCN, a higher performance 73.54 is achieved. This indicates that LCCN can further boost the performance of MASKING. However, a better combination needs to be explored.

In Table 4.4, we present the noise transition matrix learned by LCCN*, where only significant transition probabilities are marked. From this noise transition, we can find the training samples in some classes are very noisy. For example, knitwear and sweater are two classes which transit most of labels to other classes. This is because such two classes usually occur with other categories like jacket or shawl in the common dress collocation, which may incur the label transition according to the visual appearance. Besides, in Table 4.4, we can observe the outlier transition to find which class contains a lot of outliers. Furthermore, to better understand the outliers, we give some represenative samples in Figure 4.9. According to the image contents, we intuitively summarize the outlier into four sub-classes, multiple different objects, implicit categories, uncertain types and confusing appearance. As can be seen, it is usually improper to asign an unique label to these outliers, since some may contain multiple kinds of clothes. And in some challenging cases, e.g., the images in the blue box in Figure 4.9 , the hard example is also considered as the outliers by LCCN*, even

| # | Method | Accuracy@1 | Accuracy@5 |
|---|---|---|---|
| 1 | CE | 58.61 | 80.94 |
| 2 | Bootstrapping | 58.48 | 80.81 |
| 3 | Forward | **58.93** | 81.06 |
| 4 | S-adaptation | 58.00 | 80.16 |
| 5 | LCCN | 58.73 | **81.25** |
| | LCCN* | **59.09** | **81.33** |
| 6 | CE on the clean data | 53.52 | 77.84 |
| 7 | LCCN+ | 59.72 | 80.34 |

Table 4.5: The average accuracy over 5 trials on WebVision.

if the label is correct. Actually, this can be seen as the imperfect sample for training or the potential drawback of our model that requires more explore in the future research.

Table 4.5 shows the performance of LCCN and baselines on a more challenging noisy dataset WebVision. Both Top-1 and Top-5 accuracies are reported in the experiment. According to the results either in the perspective of Top-1 accuracy or Top-5 accuracy, LCCN achieves the best performance. Similarly, LCCN* and LCCN+ respectively have the further refinement based on LCCN. Nevertheless, as can be seen, the results of all methods do not present the significant gap. One possible explanation is that this task couples two challenging sub-tasks, perfectly decoupling the clean samples from the noisy dataset in the 1000 classes and well fitting the clean samples in the 1000 classes. From the current limiting performance of image recognition in ImageNet [Deng et al., 2009], we know either sub-task mentioned above needs a long way to go for a satisfying result in such a large-scale challenging scenario.

## 4.5 Summary

In this Chapter, we present a Latent Class-Conditional Noise model to overcome the unstable issue in previous stochastic approximation of noise transition via a Softmax layer. Besides, a dynamic label regression method is deployed for LCCN to iteratively infer the latent true labels and jointly train the classifier and model the noise. The theoretical analysis on the model convergence and the essential gap between training and test are also provided. Most importantly, we demonstrate that our method safeguards the bounded update of the noise transition to avoid previous arbitrarily tuning via a mini-batch of samples. Finally, we generalize our model to the open-set noisy labels setting and the semi-supervised learning setting. However, although we have shown the advantages of LCCN in a range of experiments with the generalized noisy supervision, other specific settings that considers more complex noise, e.g.,

image content based noise, could be explored. Besides, it is important to explore more effective models on the large scale noisy dataset. To the end, more works based on LCCN can be extended to train with noisy datasets.

# Chapter 5

# Towards Real-World Complex Noise

Although Chapter 3 and Chapter 4 present two approaches to learn a more accurate noise transition, it may not be ideal to achieve this ultimate goal on the real-world datasets. Once in this case, the residual noise that is not reasoned by noise transition can still break through the isolation between the classifier and the noisy labels, and degenerate the performance of the deep neural network. To address this issue, we propose a composite reasoning way to better divert the noise influence. Specifically, a quality-augmented probabilistic model is introduced, in which an extra variable, termed as the quality variable, is modeled to represent the trustworthiness of noisy labels. Our key idea is to identify the mismatch between the latent and noisy labels by embedding the quality variables into different subspaces, which effectively absorbs the influence of label noise. At the same time, the reliable labels are still able to be applied for training. To instantiate the model, we further propose a Contrastive-Additive Noise network (CAN), which consists of two important layers: (1) the contrastive layer that estimates the quality variable in the embedding space to reduce the influence of noisy labels; and (2) the additive layer that aggregates the prior prediction and noisy labels as the posterior to train the classifier. Moreover, to tackle the challenges in optimization, we deduce an SGD algorithm with the reparameterization tricks, which makes our method scalable to big data. We validate the proposed method on a range of noisy real-world image datasets. Comprehensive results have demonstrated that CAN outperforms the state-of-the-art deep learning approaches.

Generally, LCCN and MASKING can be also augmented with a quality variable to better reduce the residual noise effect. However, both of them only limits to the multi-class single-label classification problem since they model one noise transition on the simplex. In CAN, the noise transition will be implicitly modeled via a multi-layer MLP that can input/output a Softmax/Sigmoid prediction, which flexibly deals with more practical multi-class multi-label classification problems. To the end, we do not consider the limited extension and comparison with LCCN and MASKING in this Chapter, and put the focus on the real-world complex noise settings.

## 5.1 Introduction

A crucial factor in building accurate visual classification models is editorially labeled image datasets [He et al., 2016; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015]. However, collecting such data in large volume can be prohibitive in many real-world applications. Non-editorial means such as social tagging and crowdsourcing, have been explored as the efficient alternatives [Chen and Gupta, 2015; Divvala et al., 2014; Hu et al., 2017b; Krishna et al., 2017; Li and Tang, 2017; Wang et al., 2014b; Yang et al., 2018b; Zhang and Ye, 2009]. For example, a plethora of images with tags are posted on Flickr every day and become a valuable resource of labeled images for visual recognition. However, these social tags are usually highly noisy and cannot be directly used as image labels. As a result, many efforts have been devoted to learning with noisy labels.

Many early attempts [Frenay and Verleysen, 2014; Liu and Tao, 2016; Natarajan et al., 2013; Raykar et al., 2010; Zhou et al., 2012] leverage noisy labels based on the theory of the noise-free PAC model [Goldman and Sloan, 1995]. The noise is usually modeled by theoretically introducing a global parameter on the human factor [Raykar et al., 2010] or the noise ratio [Liu and Tao, 2016], and the model fitting usually requires the EM optimization. However, due to the high computational cost of EM algorithms, it is almost impractical to combine them with deep neural networks. Thus, to train deep neural networks with noisy image labels, more recent studies resort to approximate modeling to relax the optimization burden. The popular studies includes constructing the robust training by sample re-weighting or regularization [Azadi et al., 2015; Izadinia et al., 2015; Li et al., 2017b; Reed et al., 2014] and modeling the noise transition [Jindal et al., 2016; Misra et al., 2016a; Mnih and Hinton, 2012; Patrini et al., 2017; Sukhbaatar et al., 2014; Veit et al., 2017; Xiao et al., 2015]. For example, we can use the perceptual-consistent loss [Reed et al., 2014] to automatically immunize the label noise. However, the undesired non-trivial hyperparameter selection is also introduced, since it is agnostic to automatically choose the manual parameter for the real-world noise. The latter paradigm tries to recover the latent labels to train the classifier and isolate the noise with a noise transition procedure. As a representative example, Sukhbaatar *et al.* [Sukhbaatar et al., 2014] assume the latent label as ground-truth to inherently supervise the classifier training and on top of it, a noise transition is constructed to fit noisy labels. However, in the real-world complex noise scenario, the noise transition is usually not sufficiently to model the noise nature, e.g., flip and outlier. Then, the noise can still go through the transition to degrade the classifier. We term this remaining noise as *the residual noise* and define it as follows,

**Definition 5.1.** *The residual noise is when the noise transition is not sufficiently to capture the whole real-world complex noise pattern, there is extra noise that goes through the transition and then degenerates the performance of the classifier.*

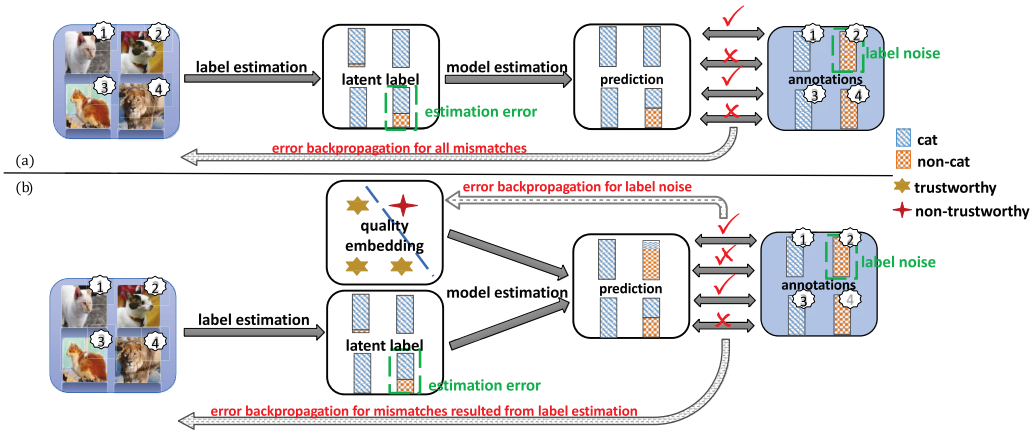



Figure 5.1: Analysis about back-propagation in previous methods that model the noise transition, as well as our idea to avoid the effect of residual label noise. (a) All images are forward into the model and the mismatch error caused by both label estimation and label noise are back-propagated. (b) With quality embedding as a control from latent labels to predictions, the negative effect of label noise is reduced in the back-propagation by diverting the reasoning to the quality embedding branch.

In this Chapter, we follow the paradigm of noise transition and propose a composite reasoning way to resist the residual noise. Figure. 5.1 simply illustrates our core idea. As shown in Figure. 5.1a, the latent labels of the first three cat images shall be approximately consistent due to their content similarity. However, mismatch may occur between the prediction of the second image and the corresponding annotation if the noise transition is not sufficient to reason the noise. On the other hand, the inconsistency between the latent label and observed annotation of the fourth image are due to the error in prediction. These above two types of mismatch are coupled during back-propagation, which degrades the training of the classifier. To overcome this issue, in Figure. 5.1b, we explicitly introduce an auxiliary variable to model the trustworthiness of noisy labels. By means of this reasoning branch, if the auxiliary variable of the second sample is embedded in the non-trustworthy subspace, the prediction can be correspondingly adapted to absorb the mismatch error caused by the label noise. Simultaneously, for the fourth image whose auxiliary variable is in the trustworthy subspace, the latent label still normally transits to the final prediction causing the mismatch. Then supervision from the correct annotations is normally fed back for training. In summary, we use such a composite reasoning idea to overcome the residual noise effect when the transition is not sufficiently accurately estimated. Since this auxiliary variable is related to the reliability of labels, we bind it with the "quality" meaning and call it the *quality variable*. Besides, as we learn it in a latent space, we call this approach *quality embedding*. In Figure 5.2, a quality-augmented

probabilistic model is proposed to formulate Figure 5.1b.

Compared with previous latent-label-based deep learning approaches, a quality variable is specially introduced to absorb the effect of residual label noise. By embedding the quality variable into different subspaces, the shortcoming illustrated like Fig.5.1a can be solved as Fig.5.1b. Further, we instantiate our model by designing a Contrastive-Additive Noise network (CAN), which is specially tailored for this application and different from previous structures. The major contribution in this Chapter can be summarized into four parts in the following.

- To overcome the residual noise issue when the noise transition is not sufficiently accurately estimated, we propose a composite reasoning way, concretely, a quality augmented probabilistic model to deep learning with noisy supervision. Through embedding the quality variable into different subspaces, the negative effect of residual label noise can be effectively reduced. Simultaneously, the supervision from reliable labels still can be back-propagated normally for training.

- To tackle the optimization difficulty, we apply the reparameterization tricks and deduce an efficient SGD algorithm, which makes our model scalable to big data.

- To instantiate the quality augmented probabilistic model, we design a Contrastive-Additive Noise network. Specially, it includes two important layers: (1) the contrastive layer estimates the quality variable in the embedding space to reduce the residual noise effect; (2) the additive layer aggregates prior predictions and noisy labels as posterior to train the classifier.

- We conduct a range of experiments to demonstrate that CAN outperforms existing state-of-the-art deep learning methods on noisy datasets. Furthermore, we present qualitative analysis about quality embedding, latent label estimation and noise pattern to give a deep insight on our model.

The rest of this Chapter is organized as follows. Section 2 theoretically analyzes the dilemma when the noise transition is not accurate and introduce our quality-augmented model. Then, we introduce its optimization algorithm as well as the corresponding instantiation Contrastive-Additive-Noise network in Sections 3, 4. We validate the efficiency of our method over a range of experiments in Section 5. Section 6 summarizes the Chapter.

## 5.2 The Quality Augmented Probabilistic models

### 5.2.1 Preliminaries

Consider that we have a noisy image dataset of $N$ items,

$$\mathcal{D} : \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\},$$

where each tuple in $\mathcal{D}$ consists of one image $x_n$ and its noisy labels $y_n$. Note that $x_n$ can be the original image or the feature vector extracted from the image. $y_n \in \mathbb{R}^K$ is a $K$-dimensional binary vector indicating which labels are annotated, and $K$ is the number of categories. In the real-world case, $y_n$ may be corrupted with annotation noise and thus incorrect. We assume the underlying clean label is $z_n \in \mathbb{R}^K$. We introduce $s_n$, a quality variable embedded in $D$-dimensional Gaussian space, to represent the annotation quality of $y_n$. Formally, it is a multi-label, multi-class classification problem with noise in labels. We target to train a deep classifier from noisy training samples. There are many tasks like this, e.g., weakly supervised object detection and segmentation [Bilen and Vedaldi, 2016; Khoreva et al., 2017; Lu et al., 2017; Wang et al., 2014a,b; Zhang et al., 2015].

### 5.2.2 Residual Noise Effect

In this section, we will theoretically analyze the dilemma when the noise transition is not sufficiently accurately estimated in [Goldberger and Ben-Reuven, 2017; Patrini et al., 2017]. Assume the non-singular $T$ and $\widehat{T}$ ($T \neq \widehat{T}$) are the ground-truth noise transition matrix and the estimated noise transition matrix respectively. $\ell$ is the proper composite cross-entropy loss with an aid link function $\psi : \Delta^{c-1} \rightarrow \mathbb{R}^c$ [Reid and Williamson, 2010; Vernet et al., 2011] and $h(\cdot)$ is the logits output of deep neural network. As discussed in [Patrini et al., 2017], the composite loss has some useful characteristics,

$$\arg\min_h \mathbf{E}_{x,y} \left[\ell_\psi \left(y, h(x)\right)\right] = \arg\min_h \mathbf{E}_{x,y} \left[\ell \left(y, \psi^{-1} \left(h(x)\right)\right)\right] = \psi \left(p(y|x)\right) \tag{5.1}$$

Based on this, we deduce the following proposition to describe the residual noise effect of the inaccurate noise transition and compared with the Theorem 2 in [Patrini et al., 2017].

**Proposition 1.** *Assume $y$ and $\tilde{y}$ are the labels respectively disturbed by the ground-truth noise transition $T$ and the residual noise transition $T\widehat{T}^{-1}$ from the clean label $z$. Then the minimizer of the loss via $\widehat{T}$ in the presence of the noisy label $y$ is same to the minimizer of the loss via $T\widehat{T}^{-1}$ in the presence of another noisy label $\tilde{y}$, but is not*

*equal to the minimizer of the loss in the presence of the clean label $z$.*

$$\begin{aligned}\arg\min_h \mathbf{E}_{x,y}\left[\ell\left(y,\ \widehat{T}^\top\psi^{-1}\left(h(x)\right)\right)\right] &= \arg\min_h \mathbf{E}_{x,\tilde{y}}\left[\ell\left(y,\ \left(T\widehat{T}^{-1}\right)^\top\psi^{-1}\left(h(x)\right)\right)\right] \\ &\neq \arg\min_h \mathbf{E}_{x,z}\left[\ell_\psi\left(y,\ h(x)\right)\right]\end{aligned} \tag{5.2}$$

*Proof.* Above proposition can be demonstrated by means of the characteristics of the composite loss as follows,

$$\begin{aligned}\arg\min_h \mathbf{E}_{x,y}\left[\ell\left(y,\ \widehat{T}^\top\psi^{-1}\left(h(x)\right)\right)\right] &= \psi\left(\left(\widehat{T}^{-1}\right)^\top P(y|x)\right) \\ &= \psi\left(\left(T\widehat{T}^{-1}\right)^\top P(z|x)\right) \\ &= \arg\min_h \mathbf{E}_{x,\tilde{y}}\left[\ell\left(y,\ \left(T\widehat{T}^{-1}\right)^\top\psi^{-1}\left(h(x)\right)\right)\right].\end{aligned} \tag{5.3}$$

Since $T \neq \widehat{T}$, we have $\psi\left(\left(T\widehat{T}^{-1}\right)^\top P(z|x)\right) \neq \psi\left(P(z|x)\right) = \arg\min_h \mathbf{E}_{x,z}\left[\ell_\psi\left(y,\ h(x)\right)\right]$. □

This proposition indicates that when the noise transition is not accurately estimated, i.e., $\widehat{T} \neq T$, the minimization under such an $\widehat{T}$ just make the classifier drift to the training on another noisy dataset, which is characterized by the residual noise transition $T\widehat{T}$. In the real-world noise scenario, the ground-truth transition should be more complex than a linear $T$ and also cannot be accurately modeled by $\widehat{T}$. Then similar phenomenon occurs, which we term as the *residual noise effect*.

### 5.2.3 The Quality Augmented Probabilstic Model

In this section, we introduce a quality variable in parallel to the latent label, which diverts the noise influence. Our probabilistic graphical model is illustrated in Figure 5.2. In the generative process, the latent label vector $Z$ purely depends on the instance $X$. We model this dependency with $P(Z|X)$. However, the noisy label vector $Y$ is generated based on both the annotation quality $S$ and the latent label $Z$, which we model with $P(Y|S,Z)$. In the inference process, both the distributions of $Z$ and $S$ are all modeled based on $X$ and $Y$. We respectively represent these two distributions with $q(Z|X,Y)$ and $q(S|X,Y)$, which play roles of posterior approximation.

According to Figure 5.2, since it is usually intractable to directly maximize the log-likelihood, we build an adjustable evidence lower bound (ELBO) [Wainwright et al.,

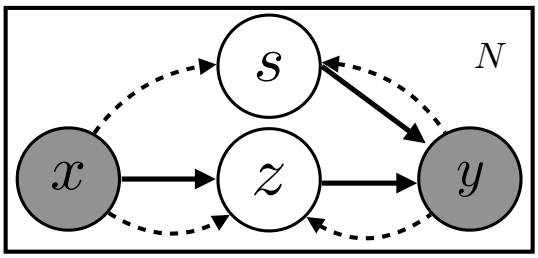



Figure 5.2: The quality augmented probabilistic model. The shaded nodes are the observed image $X$ and its noisy label vector $Y$. The latent label vector $Z$ and the quality variable vector $S$ are latent variables. Solid lines and dashed lines represent the generative process and the inference process.

2008] as follows,

$$\begin{aligned}\ln P(Y|X) &= \sum_{n=1}^{N} \ln P(y_n|x_n) \\ &\geq \sum_{n=1}^{N} \mathbf{E}_{q(z_n|x_n,y_n),q(s_n|x_n,y_n)} \left[\ln P(y_n|z_n,s_n)\right] \\ &\quad - \sum_{n=1}^{N} \mathbf{D_{KL}} \left[q(z_n|x_n,y_n)||P(z_n|x_n)\right] \\ &\quad - \sum_{n=1}^{N} \mathbf{D_{KL}} \left[q(s_n|x_n,y_n)||P(s_n)\right],\end{aligned} \tag{5.4}$$

where variational distributions $q(z_n|x_n,y_n)$ and $q(s_n|x_n,y_n)$ are to approximate the true distributions of $z_n$ and $s_n$. The above bound is a relatively good approximation of the marginal log-likelihood, providing a basis for model selection [Blei et al., 2017]. As the gap between the marginal likelihood and the ELBO becomes zero, the variational distributions approach the true distributions.

### 5.2.4 Variational mutual information regularizer

Although Figure 5.2 defines the structure of our probabilistic model, it may be hard to converge to the desirable optimal, since modeling the distribution with neural networks introduces too much flexibility. This common problem in Bayesian learning can resort to posterior regularizations [Ganchev et al., 2010]. Posterior regularizations ensure the desirable expectation and retain the computational efficiency simultaneously, which has been applied in clustering [Krause et al., 2010], classification [Zhu et al., 2014]

and image generation [Chen et al., 2016]. In this thesis, we use mutual information maximization to regularize variational distributions of $Z$ and $S$,

$$\max I\left((Z,S);(X,Y)\right) \simeq \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{E}_{q(z_n|x_n,y_n)}\left[\ln q(z_n|x_n,y_n)\right]+\mathbf{E}_{q(s_n|x_n,y_n)}\left[\ln q(s_n|x_n,y_n)\right]\right), \tag{5.5}$$

where $I(;)$ means the mutual information of two distributions. The deduction can be found in Appendix 6.5. In Eq. (5.5), maximizing the mutual information is equal to minimizing the entropy of $z_n$ and $s_n$. This means, for $z_n$, such posterior regularization forces the label probability $q(z_n|x_n,y_n)$ close to the extreme points, and for $s_n$, it will favor the distribution $q(s_n|x_n,y_n)$ with a low variance.

### 5.2.5 Objective

Combining Eq. (5.4) with (5.5), our objective becomes the maximization of the ELBO along with the mutual information regularizer. Note that, we substitute $\frac{1}{N}$ in Eq. (5.5) with a coefficient $\lambda \in [0,+\infty]$ to weight the regularization effect in the optimization. Instead of maximization, we re-write our goal as the following minimization problem.

$$\begin{aligned}\min\ \hat{L} = &-\sum_{n=1}^{N}\mathbf{E}_{q(z_n|x_n,y_n),q(s_n|x_n,y_n)}\left[\ln P(y_n|z_n,s_n)\right]\\ &+\sum_{n=1}^{N}\mathbf{D_{KL}}\left[q(z_n|x_n,y_n)||P(z_n|x_n)\right]+\sum_{n=1}^{N}\mathbf{D_{KL}}\left[q(s_n|x_n,y_n)||P(s_n)\right]\\ &-\lambda\sum_{n=1}^{N}\mathbf{E}_{q(z_n|x_n,y_n)}\left[\ln q(z_n|x_n,y_n)\right]-\lambda\sum_{n=1}^{N}\mathbf{E}_{q(s_n|x_n,y_n)}\left[\ln q(s_n|x_n,y_n)\right].\end{aligned} \tag{5.6}$$

From Eq. (5.6), our model differs from previous methods in the following three aspects.

- $P(y_n|z_n,s_n)$ indicates that the transition from the latent label to the noisy label is based on both $z_n$ and $s_n$ while previous methods [Jindal et al., 2016; Mnih and Hinton, 2012; Sukhbaatar et al., 2014] only depend on $z_n$.
- Previous works [Jindal et al., 2016; Misra et al., 2016a; Mnih and Hinton, 2012; Sukhbaatar et al., 2014; Xiao et al., 2015] use the linear transition $P(y_n|z_n)$ while our model applies nonlinear implementation $P(y_n|z_n,s_n)$.
- $z_n$ and $s_n$ are approximated with $q(z_n|x_n,y_n)$ and $q(s_n|x_n,y_n)$ in the posterior perspective while previous works [Li et al., 2017b; Veit et al., 2017; Xiao et al., 2015] have to acquire the auxiliary data.

## 5.3 Optimization

In this section, we will analyze the difficulty in optimization and deduce an SGD algorithm with reparameterization tricks for our quality augmented probabilistic model.

### 5.3.1 Reparameterization tricks

The first term in the RHS of Eq. (5.6) has no closed form when either $q(z_n|x_n, y_n)$ or $q(s_n|x_n, y_n)$ is not conjugated with $P(y_n|z_n, s_n)$. Let alone we model the distributions with deep neural network in this Chapter. The general way is by the Monte Carlo sampling. However, Paisley *et al.* [Blei et al., 2012] have shown when the derivative is about $q(z_n|x_n, y_n)$ or $q(s_n|x_n, y_n)$, the sampling estimation will present high variance. In this case, a large number of samples will be required to have an accurate estimation, which may lead to the high GPU load and the computational burden. Fortunately, reparameterization tricks [Jang et al., 2016; Kingma and Welling, 2014] are explored to overcome this difficulty in the recent years. They have shown promise in discrete and continuous representation learning.

Simply, the idea behind the reparameterization tricks is to decouple the integral variate as one parameter-related part and one parameter-free variate. After integral by substitution, the Monte Carlo sampling on this parameter-free variate will have a small variance. Based on the type of the variable, we apply the discrete reparameterization trick [Jang et al., 2016] for $z_n$ and the continuous reparameterization trick [Kingma and Welling, 2014] for $s_n$ as follows,

$$
\begin{aligned}
z_{nk} = g(\gamma_{nk}) &= \frac{\exp\left(\left(\ln q(z_{nk}=1|x_n, y_n) + \gamma_{nk1}\right)/\tau\right)}{\sum_{v=0}^{1}\exp\left(\left(\ln q(z_{nk}=v|x_n, y_n) + \gamma_{nkv}\right)/\tau\right)}, \\
s_n = f(\zeta_n) &= \mu(x_n, y_n) + \sigma^2(x_n, y_n) \odot \zeta_n,
\end{aligned}
$$

where $\tau$ is a temperature to control the discreteness of samples, $\gamma_{nk} \sim \mathbf{Gumbel}(\mathbf{0}, \mathbf{1})$[1] and $\zeta_n \sim \mathbf{N}(\mathbf{0}, \mathbf{1})$ are the parameter-free variates, $q(z_n|x_n, y_n)$, $\mu(x_n, y_n)$ and $\sigma(x_n, y_n)$ are parameter-related parts. With above reparameterization tricks, we have the following low-variance sampling estimation,

$$
\mathbf{E}_{q(z_n|x_n,y_n),q(s_n|x_n,y_n)}\left[\ln P(y_n|z_n, s_n)\right] \simeq \frac{1}{O}\sum_{o=1}^{O}\ln P(y_n|g(\gamma_n^{(o)}), f(\zeta_n^{(o)})), \tag{5.7}
$$

where $N$ is the sample number of $\gamma_n$ and $\zeta_n$ for the $n$th image. Based on Eq. (5.7), the first term in the RHS of Eq. (5.6) can be efficiently estimated, even though we set the sample number $N$ equal to 1. The remaining terms in the RHS of Eq. (5.6) can be

[1] $\gamma_{nk1}$, $\gamma_{nk2}$ are both sampled by $-\log(-\log U)$, where $U \sim$ Uniform(0,1)

explicitly computed, which we present deduction in Appendix 6.6. After transforming these terms in Eq. (5.6), the objective is derived with regard to parameters of all distributions. We learn the parameter of each distribution with an SGD algorithm, even if they are all modeled with deep networks. The gradients are summarized in Appendix 6.7.

### 5.3.2 Annealing optimization

Although we have gradients for CAN, there are two undesirable problems as follows:

- It is non-trival to exactly decouple the information from back-propagation for $z_n$ and $s_n$ at the beginning of training, i.e., squeeze out the clean label information for $z_n$ and leave the quality-related information to $s_n$. In this case, the whole optimization will suffer from the difficulty in convergence;
- The corresponding label order between $z_n$ and $y_n$ may be inconsistent in the optimization. For example, the category in the first dimension of $z_n$ can be corresponding to the category in the second dimension of $y_n$.

To avoid these two problems, we asymmetrically inject auxiliary information to the optimization procedure in an annealing way by substituting the gradient for the classifier $\nabla_{\theta_c}\hat{L}$ (see Appendix 6.7) with the following modified gradient,

$$\nabla_{\theta_c}\hat{L}_{nod} = (1-\rho(t))\,\nabla_{\theta_c}\hat{L} + \rho(t)\nabla_{\theta_c}\hat{L}_{temp}, \tag{5.8}$$

where $\nabla_{\theta_c}\hat{L}_{temp}$ is gradient with regard to the cross-entropy loss between $z_n$ and $y_n$, and $\rho(t) = \exp(-\alpha * t)$, $\alpha > 0$ is a time-varying term. With Eq.(5.8), the classifier gradient is initially decided by $\nabla_{\theta_c}\hat{L}_{temp}$, and is progressively balanced by $\nabla_{\theta_c}\hat{L}$ as $t$ increases. It guarantees the decoupling procedure from the back-propagation with an asymmetrical constraint to $z_n$ and make the label order of $z_n$ and $y_n$ consistent.

### 5.3.3 Differences from VAE

The optimization can be interpreted from the perspective of variational auto-encoder (VAE) [Kingma and Welling, 2014]. However, our model is different from the conventional VAE in three aspects.

- VAE is a generative model and usually used in unsupervised learning like image modeling [van den Oord et al., 2016]. We first extend it into discriminative learning with noisy labels.
- As the illustration in Fig.5.3 (a), the structure of VAE is usually symmetric, that is, observed knowledge is encoded into latent variables and decoded to

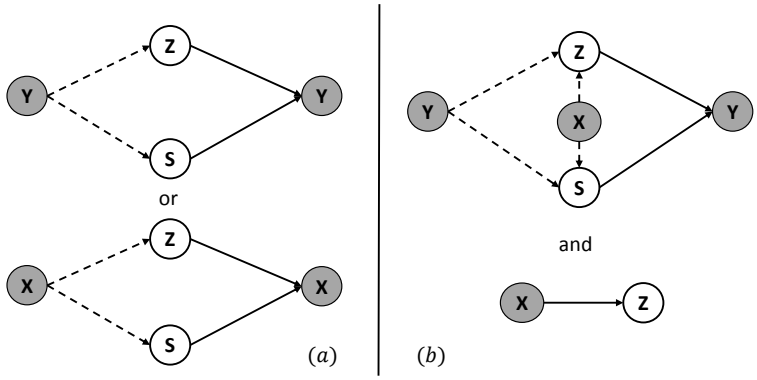



Figure 5.3: Difference between the conventional auto-encoder and our model. Solid lines and dashed lines respectively represent generative (decoding) and inference (encoding) procedures. (a) The conventional variational auto-encoder is symmetric that the observed knowledge is used to encode to the latent variables and decoded symmetrically. (b) Our model uses an auxiliary variables $X$ in this encoding-decoding procedure and meanwhile learns a discriminative model from $X$ to $Z$.

itself. However, our model in Fig.5.3 (b) is asymmetric since we introduce $X$ as an auxiliary variable in the encoding-decoding procedure. Simultaneously, a discriminative classifier will be involved and is jointly optimized.

- We propose the novel neural-layer design to model $Z$ and $S$, which is specially tailored for our application.

These improvements have never been used in previous VAEs before this work is published.

## 5.4 Contrastive-Additive Noise Network

In this section, we instantiate our model with a Contrastive-Additive Noise network (CAN) in Fig. 5.4. Simply, CAN consists of four modules, encoder, sampler, decoder and classifier, which are corresponding to the different parts of our model respectively. In the following, we describe the design in details.

### 5.4.1 Architectures

For the **encoder module**, it is to model the variational distributions, $q(s_n|x_n, y_n)$ and $q(z_n|x_n, y_n)$. Concretely, we first forward $x_n$ to a neural network to generate a prior label judgement $\hat{y}_n$. Then, according to $\hat{y}_n$ and $y_n$, we model the distribution parameters with two elaborately-designed layers. The neural network for $\hat{y}_n$ can be

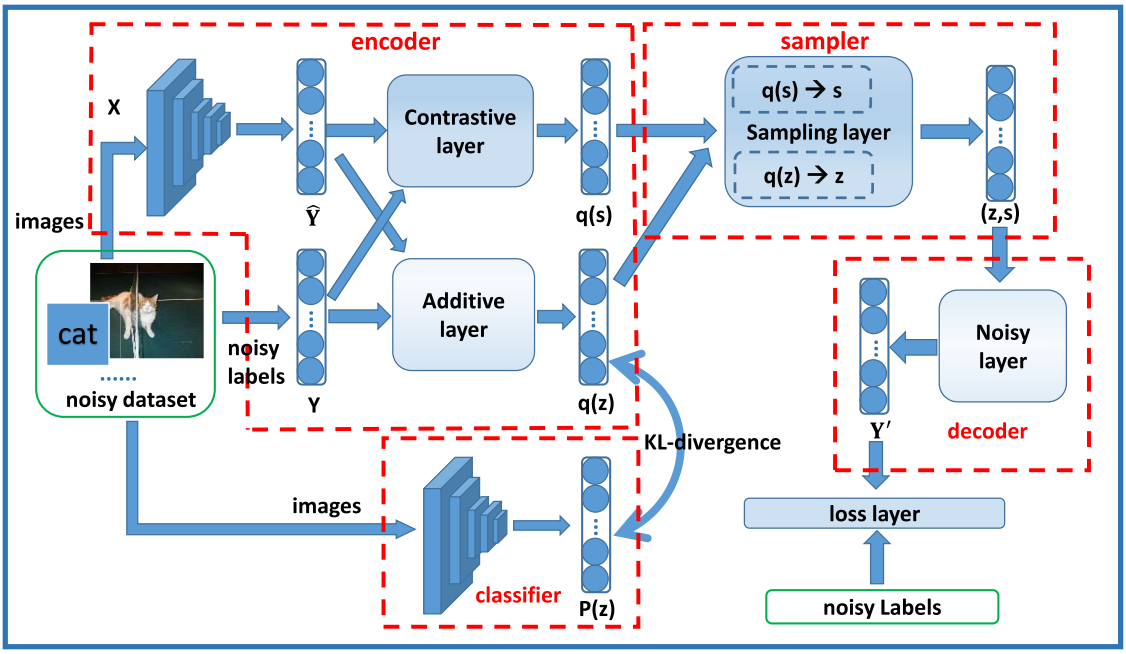



Figure 5.4: The end-to-end network architecture, which consists of four modules, encoder, sampler, decoder and classifier. Encoder tries to learn latent labels and evaluate the quality of noisy labels; sampler is used to generate samples from the encoder outputs; decoder tries to recover noisy labels from samples. Meanwhile, our classifier is learned based on KL-divergence between $q(z)$ and $P(z)$.

decided by the type of $x_n$. If $x_n$ is the original image, then a convolutional neural network can be applied. While if $x_n$ is a feature vector, a fully-connected network can be chosen. In Fig. 5.4, we take the convolutional neural network as an example. The **sampler module** is the implementation of Monte Carlo sampling on $q(s_n|x_n, y_n)$ and $q(z_n|x_n, y_n)$. It receives the output of the encoder module and samples from the Gumbel and Gaussian distributions to generate a sample set of $z_n$ and $s_n$, respectively. In the next section, we will discuss this part in details with reparameterization tricks. For the **decoder module**, it is a neural network, $P(y_n|z_n, s_n)$, which consists of two groups of (linear, ReLU) layers, following with a Sigmoid layer. It uses the output of **sampler** to recover noisy labels. Previous works usually build a linear transition from $z_n$ to $y_n$. We consider the nonlinear transition since we have the heterogeneous quality variable $s_n$. The **classifier module** as our most important target $P(z_n|x_n)$, employs the same network for $\hat{y}_n$ in the encoder module. It is trained based on KL-divergence between $q(z_n|x_n, y_n)$ and $P(z_n|x_n, y_n)$.

### 5.4.2 Contrastive layer and additive layer

We specially describe these two important layers in the encoder module. Regarding $q(s_n|x_n, y_n)$, it is a $D$-dimensional Gaussian distribution and both mean and variance need to be modeled. We exploit the **contrastive layer** to implement the estimation. It forwards $y_n$ and $\hat{y}_n$ to a shared fully-connected layer with the ReLU ($\mathbf{f_s}(\cdot)$) and then transforms their difference to a 2D-dimensional vector, the concatenation of $\mu$ and $\log \sigma^2$ like [Kingma and Welling, 2014], with an another fully-connected layer (function $\mathbf{f_t}(\cdot)$)

$$(\mu(x_n, y_n), \log \sigma^2(x_n, y_n)) = \mathbf{f_t}(\mathbf{f_s}(y_n) - \mathbf{f_s}(\hat{y}_n)).$$

This contrastive layer is built based on the assumption that the quality is related to the difference between $y_n$ and $\hat{y}_n$. We evaluate their difference in a latent space with $\mathbf{f_s}(\cdot)$ and decide which subspace it is embedded in via $\mathbf{f_t}(\cdot)$. This embedding mechanism makes us identify the label quality explicitly and subsequently helps to reduce the noise effect in $P(y_n|z_n, s_n)$. This idea has never been explored in previous noise-aware deep learning approaches [Azadi et al., 2015; Izadinia et al., 2015; Jindal et al., 2016; Li et al., 2017b; Misra et al., 2016a; Mnih and Hinton, 2012; Patrini et al., 2017; Reed et al., 2014; Sukhbaatar et al., 2014; Veit et al., 2017; Xiao et al., 2015].

Regarding the distribution $q(z_n|x_n, y_n)$, it consists of $K$ Bernoulli distributions and thus $K$ probabilities needed to be modeled. We design an **additive layer** to learn these parameters. It internally uses two non-shared fully-connected layers ($\mathbf{f_{ns1}}$ and $\mathbf{f_{ns2}}$) to transform $y_n$ and $\hat{y}_n$ into a latent space, and then feeds their addition into another fully-connected layer plus a sigmoid function (function $\hat{\mathbf{f}}_\mathbf{t}$), illustrated as follows,

$$q(z_n|x_n, y_n) = \hat{\mathbf{f}}_\mathbf{t}(\mathbf{f_{ns1}}(y_n) + \mathbf{f_{ns2}}(\hat{y}_n)).$$

**Algorithm 3** Contrastive-Additive Neural network

1: **Input** $\mathcal{D}$ with $N$ samples, parameters $\{\theta_{de}, \theta_{add}, \theta_{con}, \theta_c, \theta_{pri}\}$, learning rate $\kappa$, temperature $\tau$, annealing coefficient $\rho$, regularizer coefficient $\lambda$, batch size $B$, epoch number $N$, step counter $s$.
2: **Initial** $\kappa = 0.01$, $\rho = 1.0$, $\lambda = 0.3$, $B = 50$, $s = 0$.
**for** $s = 1$ ***to*** $MaxSteps$ **do**
  3: Sample a batch $\widehat{\mathcal{D}}$ from $\mathcal{D}$.
  4: $\tau = \max(0.5, \exp(-3 * 10^{-5} * s))$.
  **Forward**
    5: Compute $P(\hat{y}_n|x_n)$ and treat as $\hat{y}_n$.
    6: Compute $q(s_n|\hat{y}_n, y_n)$ via the contrastive layer.
    7: Compute $q(z_n|\hat{y}_n, y_n)$ via the additive layer.
    8: Compute $P(y_n|x_n)$ via the classifier.
    9: Sample $s_n$ and $z_n$ via the sampler module.
    10: Compute $P(y'_n|x_n)$ via the decoder module.
  **Backward**
    11: Update $\theta_{de}$, $\theta_{add}$, $\theta_{con}$, $\theta_{pri}$ with Eq. (6.7).
    12: Update the classifier parameter $\theta_c$ with Eq. (5.8).
  13: $\rho = \exp(-4 * 10^{-1}/s)$.
  **if** $s\%MaxSteps == 0$ **then**
    14: $\kappa = \kappa * 0.1$.
  **end**
**end**

This design learns a posterior label $z_n$ from $y_n$ and $\hat{y}_n$ by a nonlinear combination with neural network. Previous methods [Izadinia et al., 2015; Joulin et al., 2015; Li et al., 2017b; Reed et al., 2014; Veit et al., 2017] use a weight in their lost function to linearly combine the noisy label $y_n$ with the "soft" label from the prediction, the clean dataset or other side information. They usually need non-trivial tuning manually, while we resort to a learning procedure via neural network automatically.

The whole network can be trained in an end-to-end manner. During training, the noise effect is reduced by the branch of the quality variable, and simultaneously the posterior label is estimated by the additive layer to guarantee a more reliable training. The training procedure is simply summarized in Algorithm 3.

According to our experience, two parameters are very important in CAN. First, the parameter in the annealing coefficient should be set to make $q(z|x, y)$ is relatively close to the noisy labels in the early phase so that we are able to inject sufficient preference in CAN to decouple information between $z$ and $s$ from $y$. Then in the later phase, CAN can flexibly incorporate the noise for $s$ with the contrastive layer and make the label correction for $z$ by the additive layer. Second, the coefficient of variation mutual

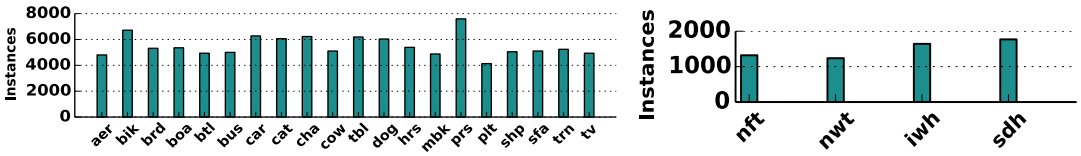



Figure 5.5: The instance number of each class on the *WEB* dataset (left) and the *AMT* dataset (right).

regularizer is important to force $s$ with low variance and $z$ to the extreme point, i.e., more confident on prediction 0 or 1. But we should set it not very big since the gradient from this variational mutual regularizer may affect the label correction procedure.

## 5.5 Experiments

In this section, we conduct the quantitative and qualitative experiments to show the superiority of CAN in classification. Specifically, we compare CAN with state-of-the-art methods, investigate its performance with varying training sizes, hyperparameter sensitivity and artificial noise. To present a deep insight on how CAN works, we analyze the quality embedding, latent label estimation and noise transition in the network.

### 5.5.1 Datasets

There are totally five image datasets that are used in the experiments.

**WEB**[1] This dataset is a subset of YFCC100M [Thomee et al., 2015] collected from the social image-sharing website. It is formed by randomly selecting images from YFCC100M, which belong to the 20 categories of the PASCAL VOC [Everingham et al., 2007]. The statistics of this dataset are shown in the left panel of Fig. 5.5. There are 97,836 samples in total and the sample number in each category ranges from 4k to 8k. Most of images in this dataset belong to one class and about 10k images have two or more. Labels in this dataset contain noise.

**AMT**[2] This dataset is collected by Zhou *et al.* [Zhou et al., 2012] from the Amazon Mechanical Turk platform. They submit four breeds of dog images from the Stanford Dog dataset [Khosla et al., 2011] to Turkers and acquire their annotations. To ease the classification, Zhou *et al.* also provide a 5376-dimensional feature for each image. The statistics of this dataset is illustrated in the right panel of Fig. 5.5. There are 7,354 samples in total and the sample number in each category is between 1k and 2k. All

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67

[2] https://www.microsoft.com/en-us/research/publication/learning-from-the-wisdom-of-crowds-by-minimax-entropy/

images in this dataset belong to one class. Labels in this dataset may contain annotation error.

**V07**[1] This dataset is provided for the 20-cateogry classification task in PASCAL VOC Chanllenge 2007 [Everingham et al., 2007]. It consists of two subsets: trainging (*V07TR*) and test (*V07TE*). There are 5,011 samples in *V07TR* and 4,592 samples in *V07TE*. All labels in this dataset are clean.

**V12**[2] This dataset is provided for the 20-cateogry classification task in PASCAL VOC Chanllenge 2012 [Everingham et al., 2012]. It consists of two subsets: trainging (*V12TR*) and test (*V12TE*). There are 11,540 samples in *V12TR* and 10,991 samples in *V12TE*. All labels in this dataset are clean.

**SD4**[3] This last dataset consists of 4 categories of dogs (same to [Zhou et al., 2012]) in the Stanford Dog dataset [Khosla et al., 2011]. It is a fine-grained categorization dataset and there are 837 samples in total. We randomly partition samples into training (*SD4TR*) and test (*SD4TE*) by $3:1$ to use. All labels in this dataset are clean.

### 5.5.2 Experimental Setup

For *WEB*, *V07* and *V12* datasets, a 34-layer residual network [He et al., 2016] is used as the initialization of the convolutional networks in CAN, and this configuration is applied to all baselines for fair comparisons. In the training phase, we first resize the short side of each image to 224 and then follow the transformations in the residual network[4] to preprocess images. In the test phase, we average the results of six-crop images as the final prediction. For *AMT* and *SD4* datasets, we directly use the features provided by [Zhou et al., 2012]. Hence, one 3-layer perception network (5376→1024, ReLU, 1024→30, ReLU, 30→4) is adopted as the substitution of the convolutional networks in CAN. Following [Jang et al., 2016], we vary the temperature $\tau$ of the Gumbel-softmax function in Eq. (5.8) with $\max(0.5, \exp(-3\text{x}10^{-5}\text{xStep}))$. As for the parameter $\alpha$ in the annealing coefficient $\rho$, we set it as $4x10^{-1}/N$ where $N$ is the sample number of the dataset. $O$ in the sampler module is set to 1. The regularizer coefficient $\lambda$ is important for the performance of CAN, which we will first discuss it, and here we empirically set to 0.3 for the first experiment in the following section. The batch size is set to 50 and the learning rate starting from 0.01 is divided by 10 every 30 epochs. In evaluation, we adopt Average Precision (AP) and mean Average Precision (mAP) [Everingham et al., 2007, 2012] as metrics.

In the following sections of "model comparision", "impact of training size" and "hyperparameter sensitivity", we train all models on *WEB* and *AMT* datasets and test them on *V07TE*, *V12TE* and *SD4TE* datasets. Note that, models trained on *WEB* dataset

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2007/
[2] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
[3] http://vision.stanford.edu/aditya86/ImageNetDogs/
[4] https://github.com/facebook/fb.resnet.torch

| Model | Resnet-N | LearnQ | ICNM | Bootstrap | CAN |
|---|---|---|---|---|---|
| aeroplane | 93.5 | 92.8 | 92.5 | 94.0 | **95.5** |
| bicycle | 85.3 | 86.1 | 86.2 | **88.4** | 87.0 |
| bird | 90.1 | 91.0 | 90.5 | 90.3 | **91.4** |
| boat | 85.1 | 87.8 | 87.9 | 88.2 | **89.9** |
| bottle | 51.2 | 50.2 | 47.7 | 51.7 | **60.1** |
| bus | 82.3 | 84.9 | 84.0 | 83.8 | **85.5** |
| car | 84.8 | 85.1 | 84.8 | 86.5 | **87.6** |
| cat | 91.2 | 90.9 | 90.6 | 91.0 | **92.0** |
| chair | 59.3 | 59.2 | 59.8 | 65.4 | **67.2** |
| cow | 87.1 | 88.3 | 88.3 | 88.0 | **90.1** |
| table | 72.1 | 71.1 | 72.7 | 77.4 | **77.7** |
| dog | 88.7 | 90.1 | 89.8 | 90.4 | **91.8** |
| horse | 91.3 | 91.2 | 91.5 | 91.8 | **93.3** |
| motorbike | 88.9 | 88.1 | 87.2 | **90.8** | 90.6 |
| person | 76.1 | 78.3 | 77.0 | 79.8 | **82.1** |
| plant | 54.4 | 56.6 | **57.0** | 55.2 | 56.0 |
| sheep | 87.6 | 89.1 | 88.9 | 92.8 | **93.6** |
| sofa | 70.0 | 73.1 | 71.5 | 75.2 | **80.7** |
| train | 90.4 | 90.7 | 91.2 | 90.8 | **94.5** |
| tv | 61.4 | 64.3 | 65.7 | 66.4 | **70.6** |
| mAP | 79.5 | 80.4 | 80.3 | 81.9 | **83.8** |

Table 5.1: Classification results on the 20 categories of *V07TE*.

are evaluated on both *V07TE* and *V12TE* datasets since they have same categories. And models trained on *AMT* dataset are ony evaluated on *SD4TE* dataset. For the "artificial noise" section, we first quantitatively add noise to *V07TR*, *V12TR* and *SD4TR* datasets, and then train all models. Finally, we test them on *V07TE*, *V12TE* and *SD4TE* datasets.

### 5.5.3 Classification results

**Training with real-world noisy datasets**. To demonstrate the effectiveness of CAN, we compare it with three state-of-the-art approaches, LearnQ [Sukhbaatar et al., 2014], ICNM [Misra et al., 2016a] and Bootstrap [Reed et al., 2014]. Besides, two baselines Resnet-N and MLP-N are added, which directly train the 34-layer residual network and the 3-layer perception network on *WEB* dataset and *AMT* dataset. Since we configure the same structure in the classifier of CAN and baselines, the possible impact in improvement from the special classification network is eliminated. The main difference between CAN and baselines is that there are extra encoding-decoding parts to incorporate noise and learn the posterior labels in the training.

| Model | Resnet-N | LearnQ | ICNM | Bootstrap | CAN |
|---|---|---|---|---|---|
| aeroplane | 98.4 | 98.4 | 98.1 | 98.6 | **98.8** |
| bicycle | 81.1 | 83.8 | 82.9 | **84.1** | **84.1** |
| bird | 92.9 | 93.8 | 93.6 | 93.6 | **95.3** |
| boat | 88.7 | 88.5 | 88.9 | 90.9 | **93.2** |
| bottle | 57.0 | 53.5 | 53.4 | 56.3 | **62.1** |
| bus | 87.4 | 87.8 | 87.7 | 89.8 | **90.8** |
| car | 73.2 | 73.7 | 72.3 | 75.5 | **77.0** |
| cat | 96.6 | 96.5 | 96.2 | 96.3 | **97.9** |
| chair | 63.3 | 64.3 | 64.7 | 69.8 | **72.6** |
| cow | 90.0 | 90.6 | 91.2 | 91.6 | **94.4** |
| table | 63.9 | 62.6 | 66.3 | 69.9 | **73.5** |
| dog | 94.3 | 94.6 | 94.2 | 94.4 | **96.1** |
| horse | 95.0 | 96.1 | 96.2 | 95.8 | **97.7** |
| motorbike | 92.9 | 91.6 | 91.4 | 93.2 | **94.3** |
| person | 76.8 | 78.4 | 78.0 | 82.2 | **82.4** |
| plant | 43.8 | **46.8** | 44.0 | 43.2 | 45.5 |
| sheep | 92.9 | 92.8 | 93.5 | 92.8 | **95.8** |
| sofa | 67.2 | 69.0 | 69.3 | 70.9 | **71.4** |
| train | 93.1 | 94.0 | 94.4 | 95.4 | **95.8** |
| tv | 65.1 | 65.4 | 66.9 | 67.4 | **68.6** |
| mAP | 80.7 | 81.1 | 81.2 | 82.6 | **84.4** |

Table 5.2: Classification results on the 20 categories of *V12TE*.

The classification performance for each category on the *V07TE*, *V12TE* and *SD4TE* datasets is reported in Table. 5.1, Table 5.2, and Table 5.3. From the results in Table 5.1 and Table 5.2, we find CAN outperforms all baselines in terms of mAP and show improvement almost in all categories. For example, on *V07TE* dataset, CAN achieves 83.8% mAP, which outperforms Resnet-N by 4.3% mAP and the best baseline Bootstrap by 1.9% mAP. In the challenging categories such as "bottle", "chair" and "sofa", it also achieves significant improvement. Although the results of LearnQ, ICNM and Bootstrap are better than those of Resnet-N, the improvement is still limited. Similarly in Table. 5.3, CAN outperforms the baselines by at least 2.8% mAP while LearnQ, ICNM and Bootstrap only improve about 1.6% mAP compared with MLP-N. Based on above results, we have the following explanations.

- LearnQ and ICNM, which only introduce the noise transition to handle the label noise, cannot prevent the residual noise effect from degenerating the classifier.
- Bootstrap shares the similar idea with CAN in the aspect of estimating the posterior label for training. But its loss function uses the linear combination

| Model | nft | nwt | iwh | swh | mAP |
|---|---|---|---|---|---|
| MLP-N | 78.1 | 73.2 | 80.9 | 76.5 | 77.2 |
| LearnQ | 80.5 | 73.7 | 83.0 | 77.7 | 78.7 |
| ICNM | 80.5 | 72.8 | **83.9** | 78.3 | 78.9 |
| Bootstrap | 80.7 | 72.5 | 83.7 | 78.1 | 78.8 |
| CAN | **82.0** | **79.0** | 81.8 | **83.8** | **81.7** |

Table 5.3: Classification results on 4 categories of SD4TE.

of predictions and noisy labels, which has no branch to reason the label noise.

- Our approach, which on one hand models the trustworthiness of noisy labels to absorb the residual noise, and on the other hand estimates the latent label in the posterior perspective to train the classifier, shows better performance.

**Impact of training size**. To explore the reliability of the proposed method when the training size changes, we compare CAN with other methods on different scales of datasets. We randomly sample different ratios of subsets in *WEB* and *AMT* datasets for training, and illustrate results of all methods on *V07TE*, *V12TE* and *SD4TE* in Figure 5.6.

From Figure 5.6, the results of all methods on these datasets decline with the decrease of the training size. However, CAN performs better than other models persistently. For instance, in the left panel of Figure 5.6, when the training size accounts at 20%, CAN achieves 81.0% mAP on the *V07TE* dataset, while ICNM and LearnQ are even worse than the most simple Resnet-N (79.4% mAP). Similar clues can be found in the middle and right panels. These results demonstrate the reliability of CAN on different scales of datasets. In Figure 5.6, we also find the decline trend on *SD4TE* dataset is more significant than that on *V07TE* and *V12TE* datasets. This is because that even if the 20% subset, there are still about 20k samples for training in *WEB* dataset. But there are only about 1.6k samples remaining in *AMT* dataset, which may lack enough knowledge to learn the classifier in the training.

**Hyperparameter sensitivity**. To investigate the reliability of CAN with different regularizer coefficients, we set $\lambda$ to 0, 0.2, 0.5, 1, 5, 10, respectively, to validate its effect. The results are illustrated in Table 5.4. From this table, we find the performance on all datasets first grows to a peak and then gradually decreases with increasing $\lambda$. For example, CAN achieves 85.2% mAP on *V12TE* dataset when $\lambda$=0.2, but significantly decreases to below 76.6% mAP when $\lambda$=10. This indicates: (1) the regularizer in the proper degree encourages our model to find a good solution; (2) too strong regularization may induce the solution deviated from the optimal. Empirically, setting $\lambda$ between 0 to 1 makes the variational mutual information regularizer collaborates well with KL-divergences.

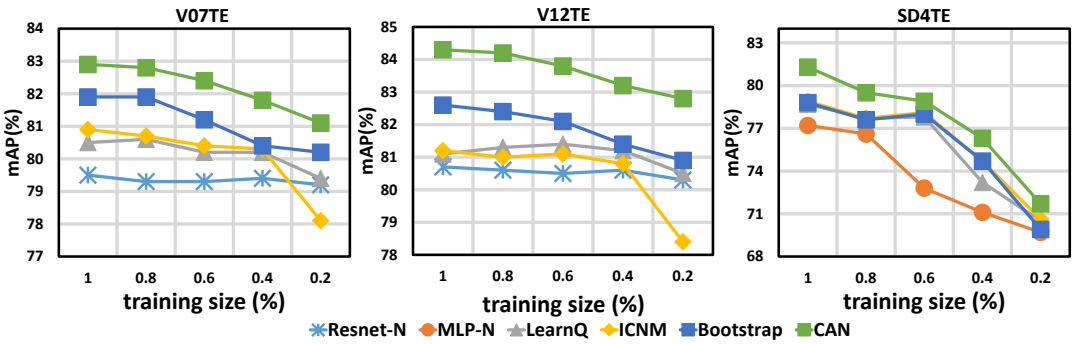



Figure 5.6: Classification results with different training sizes. We sample the subsets of *WEB* and *AMT* with five different ratios for training, and evaluate all models on *V07TE*, *V12TE* and *SD4TE*.

| $\lambda$ | 0 | 0.2 | 0.5 | 1 | 2 | 5 | 10 |
|---|---|---|---|---|---|---|---|
| *V07TE* | 82.9 | 83.5 | **84.8** | 83.6 | 80.7 | 78.8 | 77.0 |
| *V12TE* | 84.3 | **85.2** | 84.1 | 83.0 | 80.8 | 78.3 | 76.6 |
| *SD4TE* | 78.6 | **80.7** | 80.4 | 79.9 | 76.4 | 73.9 | 71.3 |

Table 5.4: Classification results with different regularization coefficient $\lambda$ in CAN.

#### 5.5.3.1 Controlled experiments with artificial noise

In previous sections, all models are trained on *WEB* and *AMT* with given noise, which does not exhibit the characteristics in different noise levels. To show the superiority of CAN, we quantitatively add noise on *V07TR*, *V12TR* and *SD4TR* datasets for training, and then compare the classification performance of all models on the *V07TE*, *V12TE* and *SD4TE* datasets. The way to add noise to datasets is by setting a corruption probability $P_{noise}$ to randomly decide whether to shuffle elements of each clean label vector or not. We list the model performance in different $P_{noise}$ settings in Table. 5.5.

As shown in Table. 5.5, when the corruption probability $P_{Noise}$=1.0, all labels are random noise and all models are trained to make a random guess. In this case, it is meaningless to make a comparision among models. With $P_{Noise}$ varying from 0.8 to 0, all models show an improvement, since there are some clean samples available for training. Specially when $P_{Noise}$ is set to 0.8, 0.6, 0.4, 0.2, CAN robustly outperforms other baselines. However, when the training data becomes purely clean, i.e., $P_{Noise}$=0, all noise-aware models are worse than Resnet-N and MLP-N. Table. 5.5 indicates: (1) The performance of all existing models is strongly-related to the noise level in the datasets. All noise-aware models perform badly in the heavy noise. (2) When the training data is clean, noise-aware models including ours cannot degenerate to the

| Dataset | $P_{noise}$ | 1.0 | 0.8 | 0.6 | 0.4 | 0.2 | 0.0 |
|---|---|---|---|---|---|---|---|
| *V07TE* | Resnet-N | 6.4 | 33.4 | 53.0 | 70.2 | 78.2 | **86.8** |
| | LearnQ | 9.1 | 28.0 | 56.4 | 72.0 | 80.1 | 85.4 |
| | ICNM | **9.2** | 28.5 | 57 | 71.6 | 79.6 | 85.4 |
| | Bootstrap | 8.9 | 30.1 | 59.3 | 73.3 | 81.0 | 85.5 |
| | CAN | 8.6 | **36.1** | **63.2** | **79.4** | **83.6** | 85.3 |
| *V12TE* | Resnet-N | 5.2 | 26.6 | 49.2 | 69.0 | 80.0 | **89.7** |
| | LearnQ | 8.4 | 23.7 | 49.7 | 70.3 | 81.3 | 88.3 |
| | ICNM | 8.4 | 23.8 | 49.6 | 70.5 | 81.4 | 88.3 |
| | Bootstrap | 8.2 | 25.1 | 51.8 | 72.6 | 82.2 | 88.5 |
| | CAN | **10.5** | **28.0** | **55.3** | **78.4** | **84.5** | 87.3 |
| *SD4TE* | MLP-N | 29.6 | 41.6 | 51.5 | 73.4 | 86.1 | **96.4** |
| | LearnQ | 26.9 | 39.6 | 60.4 | 72.7 | 89.0 | 95.9 |
| | ICNM | 27.0 | 39.7 | 60.8 | 73.1 | 89.2 | 95.8 |
| | Bootstrap | 27.8 | 38.6 | 58.7 | 73.5 | 89.3 | 96.2 |
| | CAN | **30.1** | **49.7** | **63.9** | **77.1** | **91.1** | 94.3 |

Table 5.5: Classification results with different levels of artificial noise on *V07TE*, *V12TE* and *SD4TE*.

original Resnet-N or MLP-N with the additional parameters or the hard-coded learning mechanism, yielding a worse performance. This is a potential shortcoming that needs to solved in the future. (3) CAN shows advantages in different noise levels compared with existing methods.

### 5.5.4 Model Visualization

To give a deep insight on how CAN works, here we will present the qualitative analysis about quality embedding, latent label estimation and noise transition in CAN.

**Quality embedding**. The quality variable is estimated in the embedding space by the contrastive layer. To visualize this mechanism, we respectively forward all the training samples into CAN to compute their quality embedding. By comparing the consistency between the prior prediction (thresholded by 0.5) and the noisy label, we then binarize each embedding as *trustworthy embedding* or *non-trustworthy embedding*. If we only consider the Gaussian mean of each $s$ plus the embedding type, a low dimensional visualization of quality embedding can be illustrated with t-SNE package [Van Der Maaten, 2014].

In Fig. 5.7, two exemplar categories "aeroplane" and "bike" in *WEB* dataset, and two exemplar categories "Norfolk Terrier" and "Norwich Terrier" in *AMT* dataset, are presented. As shown in Fig. 5.7, the embedding in each category exhibits two distinguishable clusters. It indicates CAN can identify mismatches between latent

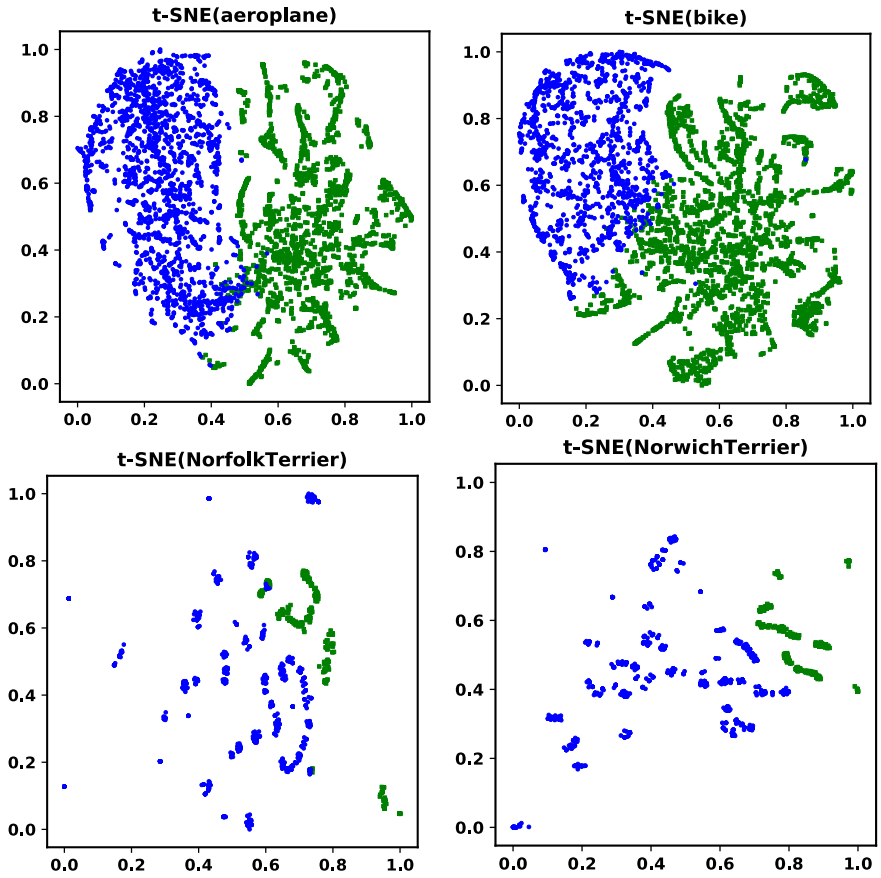



Figure 5.7: Quality embedding visualization of two categories on *WEB* and two categories on *AMT*. Better distinguishability of clusters indicates better identifiability of mismatches between latent labels and noisy labels. Blue: trustworthy embedding, Green: non-trustworthy embedding.

labels and noisy labels, and selectively embed the quality variable to different subspace based on the training samples. Thus the label noise can be effectively reduced with the auxiliary of the quality variable. Besides, we find the embedding for the first two categories is better than that for the last two categories in Fig.5.7. It is because the categories in *WEB* and *AMT* datasets are notably different in number and diversity of training samples. For example, there are about 4,200 different images and annotations in the "aeroplane", while there are only about 200 different images and 1,300 annotations in the "Norfolk Terrier". Thus embedding in the first two categories is uniformly distributed but in the last two categories is discretely cluttered.

**Latent label estimation**. The latent label is estimated in the posterior perspective by the additive layer. To visualize this estimation, we forward all the training samples into CAN to compute output of the additive layer. In Fig. 5.8, we present 20 examples

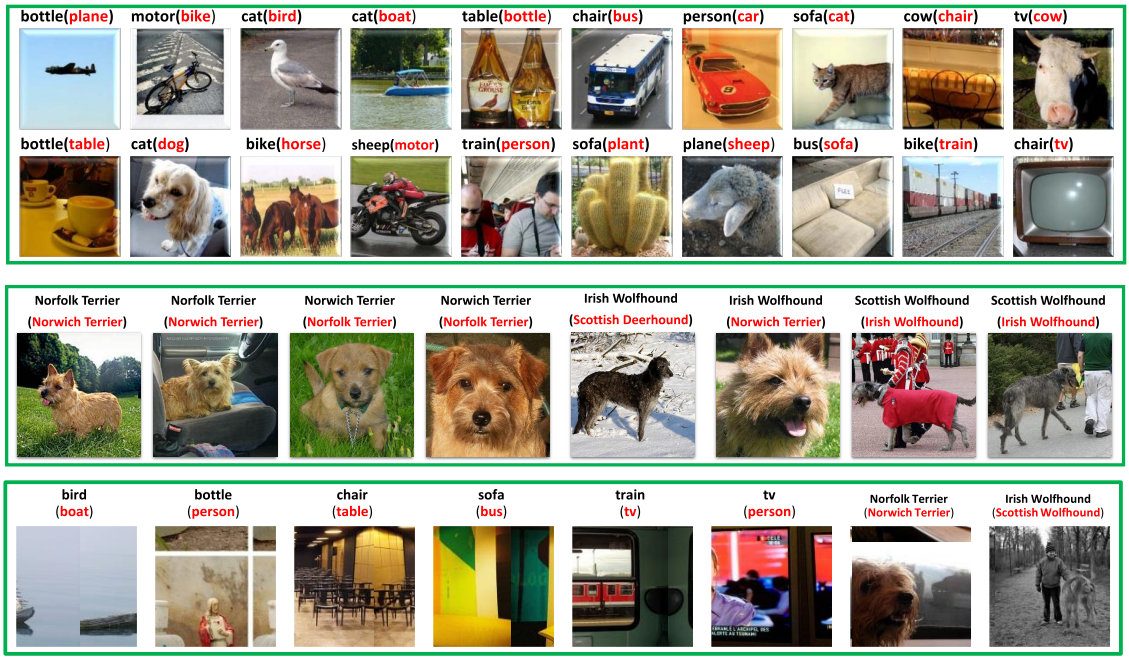



Figure 5.8: Exemplars on latent label estimation of *WEB* dataset (the first two rows) and *AMT* dataset (the third row) as well as some failures (the fourth row). We forward the noisy label (black word in title) and the image into CAN and compute the latent label (red word in title).

of *WEB* dataset and 8 examples of *AMT* dataset as well as some failures of the label estimation in CAN. According to the first three rows in Fig. 5.8, we observe that: (1) the annotations in *WEB* dataset can be totally irrelevant to the image content, e.g., "bottle" for the first "aeroplane" image; (2) In *AMT*, the Turkers also assign the wrong labels to the fine-grained images. The former error is usually from the batch annotation function provided by the Flickr website. The latter error is usually from the limited professional knowledge of Turkers. Nevertheless, from the estimation, we find our additive layer still successfully rectifies the wrong labels. Thus based on these latent labels for training, CAN achieves the better performance than other baselines.

Besides, some failures of label estimation in CAN have been presented in the fourth row of Fig. 5.8. From these examples, we find that CAN fails in the label judgement of some hard examples. Such examples are very challenging due to the target in the small scale, the partial-visible appearance, or surrounded by the specific context. For example, for the first image of the fourth row in Fig. 5.8, the bird is too small to distinguish and the wood shape in the water is very similar to the boat, which leads CAN predicts it as "boat". This indicates CAN cannot make a good distinction between the extremely hard examples and wrong examples. We leave this issue in the future explore.

**Noise transition**. To explore how the quality embedding intermediates the mismatch between latent labels and noisy labels, we investigate the transition patterns between latent labels and noisy labels. Firstly, we forward all the training samples to CAN to compute quality embeddings and latent labels. Secondly, we utilize K-means to binarize quality embeddings (only consider Gaussian mean) into trustworthy embedding and non-trustworthy embedding. Thirdly, we count transitions from latent labels to noisy labels conditioned on two types of embeddings. In Fig. 5.9, we respectively plot two transition patterns with heatmaps for *WEB* dataset and *AMT* dataset.

As shown for *WEB* dataset in Fig. 5.9, the diagonal of the transition pattern conditioned on the trustworthy embedding is dominant. In this case, noisy labels are considered to be reliable and thus transition should mainly happen among same labels. However, the transition patterns conditioned on non-trustworthy embedding is diffusing. Because in this case, noisy labels are considered not correct and transition usually happen between different labels. Similarly, transition patterns on *AMT* dataset in Fig. 5.9 also have these characteristics. Fig. 5.9 indicates CAN is based on quality embedding to automatically disturb the latent label to match the noisy label.

The transition pattern conditioned on non-trustworthy embedding usually reflects the real-world noise. Some interesting patterns can be found. For instance, according to the second panel of Fig. 5.9, "plt" class has less transition to other classes while the transition between "prs" and "tv" has high value. It means: (1) people who upload the "pottedplant" images to social websites almost do not annotate it wrong; (2) for "tv" images, some people focus on persons in the TV program, and others may pay attention

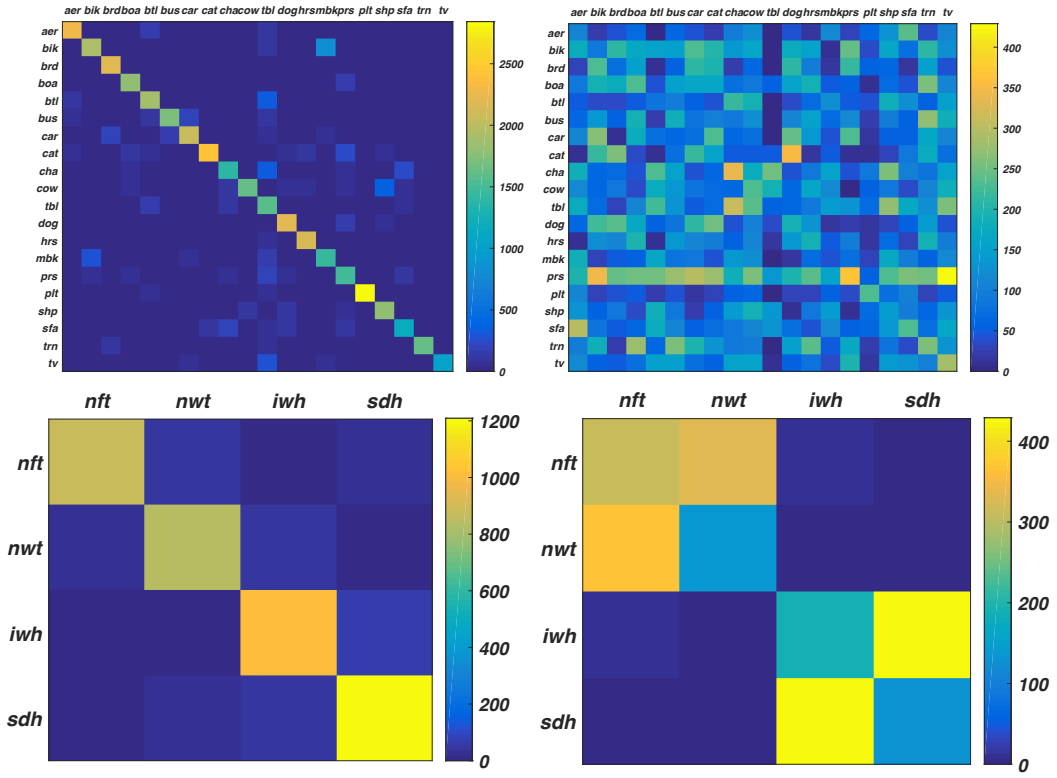



Figure 5.9: Transition patterns among labels conditioned on the trustworthy embedding and the non-trustworthy embedding on *WEB* dataset (the first two panels) and *AMT* dataset (the last two panels). Transition conditioned on the trustworthy embedding requires the consistency between the latent label and the noisy label, and thus concentrates on the diagonal. Transition conditioned on the non-trustworthy embedding identifies the mismatch between the latent label and the noisy label, and thus diffuses from the diagonal.

to TV itself. Similarly in the fourth panel of Fig. 5.9, the transition on *AMT* usually exists in the appear-similar dogs, i.e., "Norfolk Terrier" and "Norwich Terrier", "Irish Wolfhound" and "Scottish Wolfhound". It reflects that it is more difficult to distinguish these two breeds of dogs than other pairs in some sense.

## 5.6 Summary

In this Chapter, we present a quality augmented probabilistic model to learn the classifier from noisy image labels, which effectively avoids the residual noise effect. Regarding parameter estimation, we deduce an efficient SGD optimization algorithm by applying recent discrete and continuous reparameterization tricks. To instantiate the model, a Contrastive-Additive Noise network is well-designed. We demonstrate our

model outperforms other noise-aware deep learning methods on some noisy training datasets. Simultaneously, detailed visualization on three key parts is presented to give a deep insight on our model. In this Chapter, we validate our model on image data only, and other types of contents will be further explored.

# Chapter 6

# Conclusion

In this thesis, we focus on deep learning with noisy supervision to save expensive human labor on the accurate annotations in the Big data era. Specifically, we consider a series of orthogonal issues existing in the current methodology of learning via noise transition and propose the corresponding solutions. The summarization of the proposed method will be given in this Chapter and meanwhile we will discuss some future directions on the algorithms and the applications.

## 6.1 Summary of Contributions

Chapter 3 investigates the decoupling bias issue in learning via noise transition due to the large capacity of deep neural networks. We propose to leverage the structure knowledge to prevent the resulting degeneration and formulate this idea into a structure-aware probabilistic model. We term our model as "MASKING" to express it constrains the learning of noise transition consistent with the human recognition on valid and invalid noise transitions, and reduce the optimization burden to the exact transition probabilities. Furthermore, a generative adversarial learning framework is introduced to overcome the implementation challenges. The empirical results on the toy datasets and one industrial dataset confirm the promising performance of the proposed method to alleviate the decoupling bias.

Chapter 4 starts from the stochastic approximation perspective to solve the unstable learning of noise transition via a Softmax layer. Specifically, we find this is due to the lack of consideration of the global dependency relationship, which leads to a mini-batch of samples can arbitrarily update the "global" noise transition. Motivated by this, we reformulate the original class-conditional noise model as the latent class-conditional noise model, which emphasizes the noise transition depends on the whole dataset. Furthermore, we deduce a dynamic label regression method to stochastically train the classifier and learn the noise transition, which safeguards the stable update.

Extensive experiments on toy datasets and two real-world datasets demonstrate our method outperforms the current state-of-the-art baselines.

Chapter 5 theoretically analyzes the residual noise effect when the noise transition is not accurately estimated, which makes the minimizer of the classifier drift to that of training under another noisy dataset. We propose a quality-augmented probabilistic model in the composite reasoning way to overcome this residual noise influence. Concretely, a quality variable modeling the trustworthy of each training sample is introduced in parallel to the latent label. This leads to the residual noise can be absorbed by the quality variable and the classifier is normally supervised by the reliable label. We optimize this model via the reparameterization tricks and elaborate a contrastive-additive neural network to instantiate our model. Quantitative and qualitative results on one web noisy dataset and one crowdsourcing noisy dataset show the superiority and interpretability of our model.

## 6.2 Future Works

Although we have present some useful methods to improve the performance of deep learning with noisy supervision, there is still a way to go in this area. The potential works on algorithm developments and application developments deserve further explore in the future, which are summarized as follows.

- **High dimension issue**. One common drawback of the methods in Chapter 3 and Chapter 4 is the difficulty when the number of classes become very large, e.g., 10000. In this case, the structure knowledge is difficult to acquire from the human and the method to automatically extract the structure will be critical to applying MASKING. Similarly, for a large number of classes, the Gibbs sampling will suffer from the unreliable labels due to the high variance, which may make the model converge to the undesired local minimums. How to apply other efficient sampling will be useful to improve the performance of dynamic label regression.

- **More practical applications**. Deep learning with noisy supervision does not limit to the classification issue. Other areas close to this setting, e.g., weakly supervised object detection and segmentation, can be explored in the context of noisy supervision. Besides, this is practical and useful since the data collected from the real-world environment is usually imperfect. For example, the edges of the social graph may represent the fake correlation and the reward of the reinforcement learning may not reflect the real need.

# Deduction in Chapter 4

## 6.3 Proof of $\Delta_{\mathcal{F}}$ regarding to the two quantities

By definition, the excess risk of the estimated model $\hat{f}_\theta$ is directly bounded by the absolute supremum $\Delta_{\mathcal{F}}$ as follows,

$$\begin{aligned} &\mathbf{E}\left[\ell_1(z, \hat{f}_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, \hat{f}_\theta(x))\right] \\ &\quad \leq \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(z, f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta(x))\right] \right|. \end{aligned} \tag{6.1}$$

Assume the Bayes optimal classifier is $f_\theta^*$. Since $\hat{f}_\theta$ minimizes the empirical loss on $D_N$, we have the following inequality,

$$\mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta^*(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, \hat{f}_\theta(x))\right] \geq 0.$$

Then, for the error bound *w.r.t.* the expected risk and the Bayes risk, we can deduce with above inequalities as follows,

$$\begin{aligned} &\mathbf{E}\left[\ell_1(z, \hat{f}_\theta(x))\right] - \mathbf{E}\left[\ell_1(z, f_\theta^*(x))\right] \\ &\leq \mathbf{E}\left[\ell_1(z, \hat{f}_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, \hat{f}_\theta(x))\right] \\ &\quad - \left(\mathbf{E}\left[\ell_1(z, f_\theta^*(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta^*(x))\right]\right) \\ &\leq 2 * \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(z, f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta(x))\right] \right|. \end{aligned} \tag{6.2}$$

Eq. (6.1) and Eq. (6.2) show the supremum of both the excess risk and the error bound are characterized by $\Delta_{\mathcal{F}}$. Thus, we can universally analyze the corresponding upper bound of $\Delta_{\mathcal{F}}$ to investigate the generalization performance of LCCN.

## 6.4 Proof of Theorem 4.2

Assume $f_\theta^*$ and $f_\theta^\dagger$ are the underlying groundtruth labeling functions $\mathcal{X} \to \mathcal{Y}$ of clean test data and data from the Gibbs sampling respectively. Then, the $\Delta_{\mathcal{F}}$ can be reformulated by applying these notations and further deduced as follows,

$$
\begin{aligned}
&\sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(z, f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(z, f_\theta(x))\right] \right| \\
&= \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta^*(x), f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(f_\theta^\dagger(x), f_\theta(x))\right] \right| \\
&\leq \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta^*(x), f_\theta(x))\right] - \mathbf{E}\left[\ell_1(f_\theta^\dagger(x), f_\theta(x))\right] \right| \\
&+ \sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta^\dagger(x), f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(f_\theta^\dagger(x), f_\theta(x))\right] \right| \\
&\leq \underbrace{\sup_{f_\theta \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta^*(x) - f_\theta^\dagger(x), f_\theta(x))\right] \right|}_{\Delta} \\
&+ \underbrace{\sup_{f_\theta, f_\theta' \in \mathcal{F}} \left| \mathbf{E}\left[\ell_1(f_\theta'(x), f_\theta(x))\right] - \mathbf{E}^{(D_N)}\left[\ell_1(f_\theta'(x), f_\theta(x))\right] \right|}_{\Delta_{disc}}
\end{aligned}
\tag{6.3}
$$

The second term $\Delta_{disc}$ in the right-hand side of Eq. (6.3) is the popular discrepancy distance. It has been demonstrated by the following Rademacher bound [Mansour et al., 2009] for any probability $\delta > 0$,

$$
\Delta_{disc} \leq \widehat{\mathcal{R}}(\mathcal{G}) + 3\rho\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2N}},
$$

where $\mathcal{G}$ is defined by the composite functional class $\{x \mapsto \ell_1(f_\theta'(x), f_\theta(x)) : f_\theta', f_\theta \in \mathcal{F}\}$ and $N$ is the sample number. Then, combined with the given Rademacher bound, we finally proof the Theorem 4.2. i.e., for any probability $\delta$, the generalization bound of the models for learning with noisy labels is

$$
\Delta_{\mathcal{F}} \leq \Delta + \widehat{\mathcal{R}}(\mathcal{G}) + 3\rho\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2N}}. \tag{6.4}
$$

This bound theoretically points out three important factors to affect our model generalization performance, i.e., domain gap, the function complexity and the support sample number.

# Deduction in Chapter 5

## 6.5 Variational mutual information regularizers

Assume $I(:)$ and $H(:)$ respectively represent the mutual information of two random variables and the entropy of the random variable. Then we have the following deduction on the variational mutual information regularizers.

$$
\begin{aligned}
& I((Z,S):(X,Y)) \\
& = -H(Z,S|X,Y) + H(Z,S) \\
& = \mathbf{E}_{P(X,Y)}\left[\mathbf{E}_{q(Z,S|X,Y)}\left[\ln q(Z,S|X,Y)\underbrace{P(X,Y)}_{\text{prior}}\right]\right] \\
& \quad - \mathbf{E}_{P(Z,S)}\left[\ln \underbrace{P(Z,S)}_{\text{prior}}\right] \\
& \simeq \frac{1}{N}\sum_{n=1}^{N}\mathbf{E}_{q(z_n,s_n|x_n,y_n)}\left[\ln q(z_n,s_n|x_n,y_n)\right] + \text{const} \\
& \simeq \frac{1}{N}\sum_{n=1}^{N}\mathbf{E}_{q(z_n|x_n,y_n)}\left[\ln q(z_n|x_n,y_n)\right] \\
& \quad + \frac{1}{N}\sum_{n=1}^{N}\mathbf{E}_{q(s_n|x_n,y_n)}\left[\ln q(s_n|x_n,y_n)\right].
\end{aligned}
\tag{6.5}
$$

## 6.6 KL-divergences and regularizers

The remaining four terms in the RHS of Eq. (5.6) can be calculated without sampling. For example, for the latent label $z_n$, both $q(z_n|x_n,y_n)$ and $P(z_n|x_n)$ are two $K$-dimensional multinomial probabilities. Their KL-divergence term and regularizer can be simplified by enumerating each dimension. For the quality variable $s_n$, it is from

the $D$-dimensional Gaussian space $N(\mu(x_n, y_n), \text{diag}(\sigma^2(x_n, y_n)))$ whose parameters are implicitly modeled with network of input $x_n$ and $y_n$. If we assume its prior $P(s_n)$ is $N(\mathbf{0}, \mathbf{1})$ like [Kingma and Welling, 2014], it is easy to compute their KL-divergence and the regularizer due to the conjugation. In Eq. (6.6), we give their simplifications bigeminally.

$$\begin{aligned}
&\mathbf{D_{KL}}\left[q(z_n|x_n, y_n)||P(z_n|x_n)\right] - \lambda\mathbf{E}_{q(z_n|x_n,y_n)}\left[\ln q(z_n|x_n, y_n)\right] \\
&= \sum_{k=1}^{K}\sum_{z_{nk}} q(z_{nk}|x_n, y_n) \ln \frac{q(z_{nk}|x_n, y_n)^{1-\lambda}}{P(z_{nk}|x_n)}, \\
&\mathbf{D_{KL}}\left[q(s_n|x_n, y_n)||P(s_n)\right] - \lambda\mathbf{E}_{q(s_n|x_n,y_n)}\left[\ln q(s_n|x_n, y_n)\right] \\
&= -\frac{1}{2}\sum_{d=1}^{D}\left((1-\lambda)\ln\sigma_d^2(x_n, y_n) - \sigma_d^2(x_n, y_n)\right) \\
&\quad + \frac{1}{2}\mu(x_n, y_n)^T\mu(x_n, y_n) + \text{const.}
\end{aligned} \tag{6.6}$$

## 6.7 Stochastic variational gradient

Assume $\theta_{de}$, $\theta_{add}$, $\theta_{con}$, $\theta_c$ and $\theta_{pri}$ respectively represent parameters of the decoder $f(Y'|Z, S)$, the additive layer $f(Z|\hat{Y}, Y)$, the contrastive layer $f(S|\hat{Y}, Y)$, the classifier $f(Y|X)$ and the prior prediction network $f(\hat{Y}|X)$. We can derive their

gradients as the following equations.

$$
\begin{aligned}
\nabla_{\theta_{de}}\hat{L} &= -\sum_{n=1}^{N}\frac{1}{O}\sum_{o=1}^{O}\nabla_{\theta_{de}}\ln P(y'_n|g(\gamma_n^{(o)}),f(\zeta_n^{(o)})) \\
\nabla_{\theta_{add}}\hat{L} &= -\sum_{n=1}^{N}\frac{1}{O}\sum_{o=1}^{O}\nabla_g P(y'_n|g(\gamma_n^{(o)}),f(\zeta_n^{(o)}))\nabla_{\theta_{add}}g(\gamma_n^{(o)}) \\
&\quad+\sum_{n=1}^{N}\sum_{k=1}^{K}\left(\ln\frac{q(z_{nk1}|x_n,y_n)^{1-\lambda}\left(1-P(z_{nk1}|x_n)\right)}{\left(1-q(z_{nk1}|x_n,y_n)\right)^{1-\lambda}P(z_{nk1}|x_n)}\right) \\
&\quad\nabla_{\theta_{add}}q(z_{nk1}|x_n,y_n) \\
\nabla_{\theta_{con}}\hat{L} &= -\sum_{n=1}^{N}\frac{1}{O}\sum_{o=1}^{O}\nabla_f P(y'_n|g(\gamma_n^{(o)}),f(\zeta_n^{(o)}))\nabla_{\theta_{add}}f(\zeta_n^{(o)}) \\
&\quad+\sum_{n=1}^{N}\frac{1}{2}\nabla_{\theta_{add}}\sum_{d=1}^{D}\left(\sigma_d^2(x_n,y_n)-(1-\lambda)\ln\sigma_d^2(x_n,y_n)\right) \\
&\quad+\sum_{n=1}^{N}\frac{1}{2}\nabla_{\theta_{add}}\left(\mu(x_n,y_n)^T\mu(x_n,y_n)\right) \\
\nabla_{\theta_c}\hat{L} &= \sum_{n=1}^{N}\sum_{k=1}^{K}\frac{P(z_{nk1}|x_n)-q(z_{nk1}|x_n,y_n)}{P(z_{nk1}|x_n)(1-P(z_{nk1}|x_n))}\nabla_{\theta_c}P(z_{nk1}|x_n) \\
\nabla_{\theta_{pri}}\hat{L} &= \sum_{n=1}^{N}\sum_{k=1}^{K}\nabla_{\theta_c}\left(y_{nk}\ln P(\hat{y}_n|x_n)+(1-y_{nk})\ln\left(1-P(\hat{y}_n|x_n)\right)\right)
\end{aligned}
\tag{6.7}
$$

where $z_{nk1}$ is abbreviation of $z_{nk}=1$ to avoid notation clutters.

# References


Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *Symposium on Operating Systems Design and Implementation*, volume 16, pages 265–283, 2016. 1, 4

Y. Aït-Sahalia, J. Fan, and D. Xiu. High-frequency covariance estimates with noisy and asynchronous financial data. *Journal of the American Statistical Association*, 105(492):1504–1517, 2010. 23

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2 (4):343–370, 1988. 3, 23

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 2017. 5, 13, 16, 38, 41, 53

Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. Auxiliary image regularization for deep cnns with noisy labels. In *International Conference on Learning Representation*, 2015. 13, 15, 23, 63, 74

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 47

Eyal Beigman and Beata Beigman Klebanov. Learning with annotation noise. In *International Joint Conference on Natural Language Processing*, pages 280–287, 2009. 3

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006. 23

James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, David Warde-Farley, Ian Goodfellow, Arnaud Bergeron, et al. Theano: Deep learning on gpus with python. In *BigLearning Workshop in Advances in Neural Information Processing Systems*, volume 3, pages 1–48, 2011. 1

Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 66

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 18, 19

David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006. 18

David M. Blei, Michael I. Jordan, and John W. Paisley. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning*, pages 1367–1374, New York, NY, USA, 2012. ACM. 18, 70

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (just-accepted), 2017. 18, 68

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 129–136. 2008. 20

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014. 11

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, pages 169–207. Springer, 2004. 20

Carla E. Brodley and Mark A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999. 3, 4

Y. Cha and J. Cho. Social-network analysis using topic models. In *Special Interest Group on Information Retrieval*, 2012. 23

Xi Chen, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information

maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180. Curran Associates, Inc., 2016. 69

Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *IEEE International Conference on Computer Vision*, pages 1431–1439, 2015. 17, 63

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014. 1

Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard E Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pages 969–977, 2018. 19

Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. *CoRR*, abs/1702.05374, 2017. 52

Carlos Daniel Paulino, Paulo Soares, and John Neuhaus. Binomial regression with misclassification. *Biometrics*, 59(3):670–675, 2003. 4

Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006. 20

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 11, 12, 52, 60

J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 26

Y. Dgani, H. Greenspan, and J. Goldberger. Training a neural network based on unreliable human annotation of medical images. In *IEEE International Symposium on Biomedical Imaging*, 2018. 23

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representation*, 2014. 19

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representation*, 2016. 19

Santosh K. Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 63

P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997. 23

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 results, 2007. 76, 77

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 results, 2012. 77

B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014. 3, 4, 63

Anil Gaba. Inferences with an unknown noise level in a bernoulli process. *Management Science*, 39(10):1227–1237, 1993. 4

Dragan Gamberger, Nada Lavrač, and Sašo Džeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *International Workshop on Algorithmic Learning Theory*, pages 199–212. Springer, 1996. 4

Dragan Gamberger, Nada Lavrac, and Ciril Groselj. Experiments with noise filtering in a medical domain. In *International Conference on Machine Learning*, pages 143–151, 1999. 4

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul): 2001–2049, 2010. 68

Zoubin Ghahramani and Matthew J Beal. Variational inference for bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems*, pages 449–455, 2000. 18

Herve Goeau, Pierre Bonnet, and Alexis Joly. Plant identification based on noisy web data: the amazing performance of deep learning. In *CLEF: Conference and Labs of the Evaluation Forum*, pages 1–13, 2017. 11, 12

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representation*, 2017. xi, 6, 13, 17, 24, 28, 29, 35, 41, 43, 44, 49, 50, 51, 52, 54, 66

Sally A. Goldman and Robert H. Sloan. Can pac learning algorithms tolerate random attribute noise? *Algorithmica*, 14(1):70–84, 1995. 63

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 25, 32

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. 29

N. Goodman. Sense and certainty. *The Philosophical Review*, pages 160–167, 1952. 32

P. Grigolini, G. Aquino, M. Bologna, M. Luković, and B. West. A theory of 1/f noise in human cognition. *Physica A: Statistical Mechanics and its Applications*, 388(19): 4192–4204, 2009. 31

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. In *International Conference on Learning Representation*, 2015. 19

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 2017. 35

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *European Conference on Computer Vision*, September 2018. 41, 51

Paul Gustafson, Nhu D Le, and Refik Saskin. Case–control analysis with partial knowledge of exposure misclassification probabilities. *Biometrics*, 57(2):598–609, 2001. 4

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 2018. 5, 15

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 4, 53, 63, 77

Ray J Hickey. Noise modelling and evaluating learning from examples. *Journal of Artificial Intelligence Research*, 82:157–179, 1996. 3

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 18

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6): 82–97, 2012. 4, 11, 41

JS Urban Hjorth. *Computer intensive statistical methods: Validation, model selection, and bootstrap*. Routledge, 2017. 19

Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 856–864, 2010. 18

M. Hu, Y. Yang, F. Shen, L. Zhang, H. T. Shen, and X. Li. Robust web image annotation via exploring multi-facet and structural knowledge. *IEEE Transactions on Image Processing*, 26(10):4871–4884, 2017a. 41

Mengqiu Hu, Yang Yang, Fumin Shen, Luming Zhang, Heng Tao Shen, and Xuelong Li. Robust web image annotation via exploring multi-facet and structural knowledge. *IEEE Transactions on Image Processing*, 26(10):4871–4884, 2017b. 63

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 1

J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, 2007. 23

Hamid Izadinia, Bryan C Russell, Ali Farhadi, Matthew D Hoffman, and Aaron Hertzmann. Deep classifiers from image tags in the wild. In *The Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*, pages 13–18, 2015. 15, 63, 74, 75

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representation*, 2016. 70, 77

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representation*, 2017. 18, 19, 44

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels. In *International Conference on Computer Vision and Pattern Recognition*, 2018. 5, 13, 15, 23, 38

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *International Conference on Data Mining*, 2016. 5, 17, 63, 69, 74

Jonasson Johan. Fast mixing for latent dirichlet allocation. *arXiv preprint arXiv:1701.02960v2*, 2017. 47

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2): 183–233, 1999. 18, 24

Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. *Learning Visual Features from Large Weakly Supervised Data*. Springer International Publishing, 2015. 15, 75

Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84, 2016. 2

Adam Tauman Kalai and Rocco A Servedio. Boosting in the presence of noise. *Journal of Computer and System Sciences*, 71:266–290, 2005. 3

Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 66

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization in IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 76, 77

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representation*, 2014. 18, 19, 44, 70, 71, 74, 93

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016. 19

Andreas Krause, Pietro Perona, and Ryan G. Gomes. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems*, pages 775–783. Curran Associates, Inc., 2010. 68

Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *European Conference on Computer Vision*, pages 301–320, 2016. 11, 12

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 63

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. 1, 11, 41, 63

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 11

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representation*, 2017. 23

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. 19, 45, 47

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017a. 11, 12, 27, 29, 52

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *International Conference on Computer Vision*, 2017b. 14, 63, 69, 74, 75

Zechao Li and Jinhui Tang. Weakly supervised deep matrix factorization for social image understanding. *IEEE Transactions on Image Processing*, 26(1):276–288, 2017. 63

Juxin Liu, Paul Gustafson, Nicola Cherry, and Igor Burstyn. Bayesian analysis of a matched case–control study with expert prior information on both the misclassification of exposure and the exposure–disease association. *Statistics in Medicine*, 28(27):3411–3423, 2009. 4

Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3115–3123, 2017. 19

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016. 19

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (3):447–461, 2016. 3, 13, 63

W. Liu, Y. Jiang, J. Luo, and S. Chang. Noise resistant graph ranking for improved web image search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 23

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2200–2207, 2013. 20

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417, 2014. 20

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 20

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016. 20

Zhiwu Lu, Zhenyong Fu, Tao Xiang, Peng Han, Liwei Wang, and Xin Gao. Learning from weak and noisy labels for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):486–500, 2017. 66

Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, 2018. 13, 15, 23, 41

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representation*, 2017. 19, 44

E. Malach and S. Shalev-Shwartz. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*, 2017. 23

Andrea Malossini, Enrico Blanzieri, and Raymond T Ng. Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, 22:2114–2121, 2006. 3

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1800–1809. 2018. 19

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory*, 2009. 20, 47, 91

Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151, 2013. 3

R. Michalski, J. Carbonell, and T. Mitchell. *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media, 2013. 25

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6):1236–1246, 2017. 1

Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2016a. 17, 63, 69, 74, 78

Ishan Misra, C. Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels. In *International Conference on Computer Vision and Pattern Recognition*, 2016b. 13

T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. In *International Conference on Learning Representation*, 2016. 23

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on International Conference on Machine Learning*, pages II–1791–II–1799, 2014. 18, 19, 44

Volodymyr Mnih and Geoffrey Hinton. Learning to label aerial images from noisy data. In *International Conference on Machine Learning*, 2012. 16, 63, 69, 74

Zachary Nado, Jasper Snoek, Roger Grosse, David Duvenaud, Bowen Xu, and James Martens. Stochastic gradient langevin dynamics that exploit neural network structure. In *International Conference on Learning Representation*, 2018. 19

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013. 3, 5, 17, 63

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research*, 18(Jan):5666–5698, 2017. 6

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *International Conference on Computer Vision and Pattern Recognition*, 2017. xi, 13, 17, 23, 24, 26, 27, 28, 29, 35, 36, 41, 43, 50, 51, 52, 53, 54, 63, 66, 74

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008. 19

Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016. 19

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010. 3, 4, 6, 63

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representation*, 2014. 13, 15, 23, 50, 52, 63, 74, 75, 78

Mark D Reid and Robert C Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11(Sep):2387–2422, 2010. 6, 66

R Rekaya, KA Weigel, and D Gianola. Threshold model for misclassified binary responses with applications to animal breeding. *Biometrics*, 57(4):1123–1129, 2001. 4

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, 2018. 13, 25, 41, 51

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and approximate inference in deep generative models. *International Conference on Machine Learning*, 2014. 18

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representation*, 2018. 19

V. Rockova and K. McAlinn. Dynamic variable selection with spike-and-slab process priors. *arXiv:1708.00085*, 2017. 32

F. Rodrigues and F. Pereira. Deep learning from crowds. In *Association for the Advancement of Artificial Intelligence*, 2018. 23

Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016. 19

M Ruiz, FJ Girón, CJ Pérez, J Martín, and C Rojano. A bayesian model for multinomial sampling with misclassified data. *Journal of Applied Statistics*, 35(4): 369–382, 2008. 4

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018. 19

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017. 1

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017. 19

PS Sastry, GD Nagendra, and Naresh Manwani. A team of continuous-action learning automata for noise-tolerant learning of half-spaces. *IEEE Transactions on Systems, Man, and Cybernetics*, 40:19–28, 2010. 3

Frank Schneider, Lukas Balles, and Philipp Hennig. Deepobs: A deep learning optimizer benchmark suite. In *International Conference on Learning Representation*, 2018. 4

Jiaxin Shi, Shengyang Sun, and Jun Zhu. Kernel implicit variational inference. In *International Conference on Learning Representation*, 2017. 19

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *IEEE Conference on Learning Representation*, 2014. 1, 63

Padhraic Smyth, Usama M Fayyad, Michael C Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, pages 1085–1092, 1995. 3

Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1837–1846, 2018. 15

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017. 19

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *International Conference on Learning Representation*, 2014. 13, 17, 24, 50, 63, 69, 74, 78

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014. 11, 41

Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999. 5

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1, 4, 63

D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. Joint optimization framework for learning with noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 16, 53, 57, 58

A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017. 23

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li Jia Li. The new data and new challenges in multimedia research. *Communications of the ACM*, 59(2):64–73, 2015. 76

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 2, 11, 12

D. Tran, R. Ranganath, and D. Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, 2017a. 18, 32

Dustin Tran, Rajesh Ranganath, and David Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533, 2017b. 19

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017. 19

Arash Vahdat, Evgeny Andriyash, and William Macready. Dvae#: Discrete variational autoencoders with relaxed boltzmann priors. In *Advances in Neural Information Processing Systems*, pages 1869–1878, 2018a. 19

Arash Vahdat, William Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. Dvae++: Discrete variational autoencoders with overlapping transformations. In *International Conference on Machine Learning*, pages 5042–5051, 2018b. 19

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 71

Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014. 82

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 51, 63, 69, 74, 75

Elodie Vernet, Mark D Reid, and Robert C Williamson. Composite multiclass losses. In *Advances in Neural Information Processing Systems*, pages 1224–1232, 2011. 6, 66

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008. 18, 45, 67

Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. In *European Conference on Computer Vision*, pages 431–445. Springer, 2014a. 66

Le Wang, Gang Hua, Jianru Xue, Zhanning Gao, and Nanning Zheng. Joint segmentation and recognition of categorized objects from noisy web image collection. *IEEE Transactions on Image Processing*, 23(9):4070–4086, 2014b. 63, 66

Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S. Xia. Iterative learning with open-set noisy labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 50

P. Welinder, S. Branson, P. Perona, and S. Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, 2010. 23

Robert L Winkler, JM Bernardo, MH DeGroot, DV Lindley, and AFM Smith. Information loss in noisy and dependent processes. *Bayesian Statistics*, 2:559–570, 1985. 4

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *International Conference on Computer Vision and Pattern Recognition*, 2015. 11, 12, 13, 17, 25, 34, 38, 50, 51, 53, 57, 58, 63, 69, 74

Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1100–1113, 2018. 1

J. Yang, X. Sun, Y. Lai, L. Zheng, and M. Cheng. Recognition from web data: A progressive filtering approach. *IEEE Transactions on Image Processing*, 27(11): 5303–5315, 2018a. 41

Jufeng Yang, Xiaoxiao Sun, Yu-Kun Lai, Liang Zheng, and Ming-Ming Cheng. Recognition from web data: A progressive filtering approach. *IEEE Transactions on Image Processing*, 2018b. 63

Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016. 19

Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3320–3328. 2012. 20

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018a. 18, 19

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representation*, 2016. 4, 13, 16, 23, 43

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representation*, 2017. 13, 15

J. Zhang and L. Ye. Content based image retrieval using unclean positive examples. *IEEE Transactions on Image Processing*, 18(10):2370–2375, 2009. 63

Wei Zhang, Sheng Zeng, Dequan Wang, and Xiangyang Xue. Weakly supervised semantic segmentation for social images. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. 66

Wensheng Zhang, Romdhane Rekaya, and Keith Bertrand. A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. *Bioinformatics*, 22(3):317–325, 2005. 4

Y. Zhang, X. Chen, J. Li, W. Teng, and H. Song. Exploring weakly labeled images for video object segmentation with submodular proposal selection. *IEEE Transactions on Image Processing*, 27(9):4245–4259, 2018b. 41

Denny Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012. 63, 76, 77

Jun Zhu, Ning Chen, and Eric P Xing. Bayesian inference with posterior regularization and applications to infinite latent svms. *Journal of Machine Learning Research*, 15: 1799, 2014. 68

Xingquan Zhu, Xindong Wu, and Qijun Chen. Eliminating class noise in large datasets. In *International Conference on Machine Learning*, pages 920–927, 2003. 4