Deep Learning From Noisy Image Labels With Quality Embedding

Jiangchao Yao, Jiajie Wang, Ivor W. Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, *Senior Member, IEEE*, and Rui Zhang

***Abstract*— There is an emerging trend to leverage noisy image datasets in many visual recognition tasks. However, the label noise among datasets severely degenerates the performance of deep learning approaches. Recently, one mainstream is to introduce the latent label to handle label noise, which has shown promising improvement in the network designs. Nevertheless, the mismatch between latent labels and noisy labels still affects the predictions in such methods. To address this issue, we propose a probabilistic model, which explicitly introduces an extra variable to represent the trustworthiness of noisy labels, termed as the quality variable. Our key idea is to identify the mismatch between the latent and noisy labels by embedding the quality variables into different subspaces, which effectively minimizes the influence of label noise. At the same time, 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 image datasets. Comprehensive results have demonstrated that CAN outperforms the state-of-the-art deep learning approaches.**

***Index Terms*— Deep learning, noisy image labels, quality embedding.**

Manuscript received September 26, 2017; revised May 4, 2018 and August 6, 2018; accepted October 10, 2018. Date of publication October 24, 2018; date of current version December 12, 2018. This work was supported in part by The High Technology Research and Development Program of China under Grant 2015AA015801, in part by NSFC under Grant 61521062, in part by STCSM under Grant 18DZ2270700, and in part by the Australian Research Council under Grant FT130100746, Grant DP180100106, and Grant LP150100671. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chia-Wen Lin. *(Corresponding authors: Ivor W. Tsang; Ya Zhang.)*

J. Yao is with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, and the Center for Artificial Intelligence, FEIT, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: sunarker@sjtu.edu.cn; jiangchao.yao@student.uts.edu.au).

J. Wang and Y. Zhang are with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ww1024@sjtu.edu.cn; ya_zhang@sjtu.edu.cn).

I. W. Tsang and C. Zhang are with the Center for Artificial Intelligence, FEIT, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: ivor.tsang@uts.edu.au; chengqi.zhang@uts.edu.au).

J. Sun and R. Zhang are with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: junsun@sjtu.edu.cn; zhang_rui@sjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2018.2877939

I. Introduction

A CRUCIAL factor in building accurate visual classification models is editorially labeled image datasets [1]–[4]. 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 [5]–[12]. 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.

Early attempts [13]–[17] leverage noisy labels based on the theory of the noise-free PAC model [18]. The noise is usually modeled by theoretically introducing a global parameter on the human factor [13] or the noise ratio [15]. And the model fitting usually requires the EM optimization. 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. These studies can be summarized into two categories, building the robust loss function [19]–[22] and modeling the latent labels [23]–[29]. The former paradigm explores to build a robust loss function, e.g., the perceptual-consistent loss [20], which is immune to 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.* [24] assume the latent label as ground-truth to inherently supervise the classifier training and on top of it, a noise transition layer is constructed to fit noisy labels. However, since the linear noise transition layer is insufficiently model the noise nature, e.g., flip and outlier, label noise can still go through this layer to degrade the classifier.

In this paper, we follow the latter paradigm and propose to introduce an auxiliary variable to confront noise in the noise transition. Fig. 1 illustrates the core idea of our model. As shown in Fig. 1(a), the latent labels and predictions of the first three cat images shall be approximately consistent due to their content similarity. However, mismatch occurs between the prediction of the second image and the corresponding

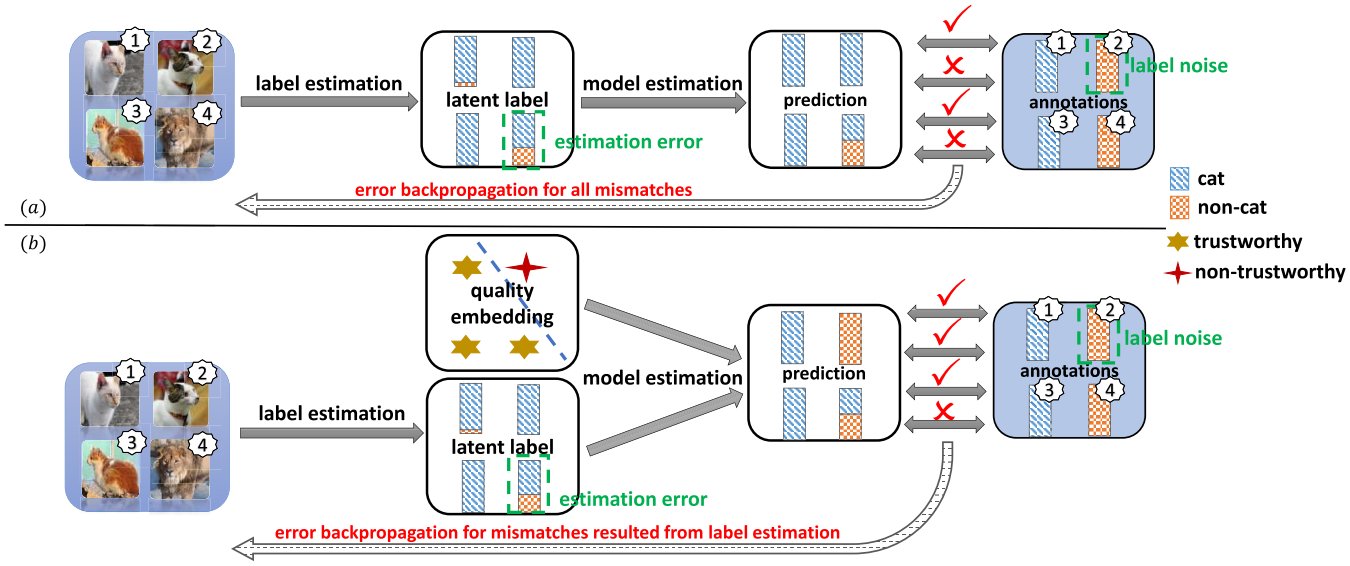


Fig. 1. Analysis about back-propagation in previous methods that model the latent label, as well as our idea to avoid the effect of label noise. (a) All images are forward into the model and the mismatch error caused by both label estimation and label noise are back-propagated. (b) With quality embedding as a control from latent labels to predictions, the negative effect of label noise is reduced in the back-propagation.

annotation due to the label 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 two types of mismatch are coupled during back-propagation, which degrades the training of the classifier. To overcome this issue, in Fig. 1(b), we explicitly introduce an auxiliary variable to model the trustworthiness of noisy labels. In this case, if the auxiliary variable of the second sample is embedded in the non-trustworthy subspace, the latent label can be accordingly disturbed to prevent mismatch error caused by the label noise in back-propagation. 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. Since this 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 way as ***quality embedding***, and term our model as the quality embedding model.

Compared with previous latent-label-based deep learning approaches, a quality variable is specially introduced to model the trustworthiness of noisy labels. By embedding the quality variable into different subspaces, the shortcoming illustrated like Fig.1(a) can be solved as Fig.1(b). 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 paper can be summarized into four parts in the following.

- To address the shortcoming of existing latent-label-based deep learning approaches, we propose a quality embedding model that introduces a quality variable to represent the trustworthiness of noisy labels. By embedding the quality variable into different subspaces, the negative effect of label noise can be effectively reduced. Simultaneously, the supervision from reliable labels still can be back-propagated normally for training.
- To instantiate the quality embedding model, we design a Contrastive-Additive Noise network. Specially, it consists of two important layers: (1) the contrastive layer estimates the quality variable in the embedding space to reduce noise effect; (2) the additive layer aggregates prior predictions and noisy labels as posterior to train the classifier.
- To tackle the optimization difficulty, we apply the reparameterization tricks and deduce an efficient SGD algorithm, which makes our model scalable to big data.
- We conduct a range of experiments to demonstrate that CAN outperforms existing state-of-the-art deep learning methods on noisy datasets. We further present qualitative analysis about quality embedding, latent label estimation and noise pattern to give a deep insight on our model.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work of learning with noisy labels in deep learning. Then we introduce our quality embedding model, the corresponding instantiation Contrastive-Additive-Noise network as well as its optimization algorithm in Section 3. We validate the efficiency of our method over a range of experiments in Section 4. Section 5 concludes the paper.

II. Related Work

Social websites and crowdsourcing platforms provide us an effective way to gather a large amount of low-cost annotations for images. However, in the visual recognition tasks such as image classification, the noise among labels shall severely degenerate the performance of classification models [30]. To exploit the great value of noisy labels, several noise-aware deep learning methods have been proposed for the image classification task. Here, we briefly review these related works.

A. Robust Loss Function

This line of research aims at designing a robust loss function to alleviate noise effect. For instance, Joulin *et al.* [31] weight the cross-entropy loss with the sample number to balance the emphasis of noise in positive and negative instances. Izadinia *et al.* [21] estimate a global ratio of positive samples to weaken the supervision in the loss function. Reed *et al.* [20] consider the consistency of predictions in similar images and apply bootstrap to the loss function. They substitute the noisy label with a weight combination of the noisy label and the prediction to encourage the consistent output. Recently, Li *et al.* [22] re-weight the noisy label with a soft label learned from side information. They train a teacher network with the clean dataset to compute the soft label by leveraging the knowledge graph. The soft label is then combined with the noisy label in the loss function to pilot student model's learning. Veit *et al.* [29] rectify labels in the cross-entropy loss with a label-correction network trained on the extra clean dataset. While these methods are concerned with modifying the labels in the loss function by re-weighting or rectification, our approach also models the auxiliary trustworthiness of noisy image labels to reduce the noise effect on training.

B. Modeling the Latent Labels

This paradigm targets at modeling the latent labels to train the classifier, and building a transition for adaption from the latent labels to the noisy labels. With the success of deep learning in image recognition, this kind of idea receives considerable attention. Mnih and Hinton [23] first propose a latent variable model on aerial images, which assumes that the noise is symmetric and at random. Based on it, several works [24], [28] 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 and Gupta [6] apply a two-stage approach to model the latent label and learn the translation to the noisy label, in which a clean dataset is used. Different from methods that model label transition in the dataset level, Xiao *et al.* [27] propose a probabilistic graphic model that disturbs 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.* [25] build two parallel classifiers. One classifier deals with image recognition and the other classifier models human's reporting bias. However, it still suffers from the problem mentioned in Fig. 1(a) since similar images have similar latent variables. Although these methods take advantages of deep neural network to model the latent label, the simple transition cannot sufficiently model the label corruption. We go on by unearthing the annotation quality from training data and further utilize it to guide the learning of our model.

TABLE I
NOTATIONS AND THEIR DESCRIPTIONS FREQUENTLY USED IN THIS PAPER

Notation	Description
M	number of training items
K	number of categories
D	dimension of the quality variable
X	image variable
Y	noisy label vector variable
Z	latent label vector variable
S	quality vector variable
W_C	parameter of classifier network
W_N	parameter of noise network
W_Q	parameter of annotation quality network
W_L	parameter of latent label network
m	index of an item
x_m	mth observed image
y_m	mth observed noisy label vector
z_m	mth latent label vector
s_m	mth quality vector
μ	mean of Gaussian distribution
σ	covariance diagonal of Gaussian distribution
λ	regularizaion cofficient
γ_m	mth sample from Gumbel distribution
ζ_m	mth sample from Gaussian distribution
τ	temperature in Gumbel-SoftMax
α	time-varying coefficient

III. QUALITY EMBEDDING MODELS

A. Preliminaries

Consider that we have a noisy image dataset of M items,

$$\mathscr{D} : \{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\},$$

where each tuple in $\mathscr{D}$ consists of one image x_m and its noisy labels y_m. Note that x_m can be the original image or the feature vector extracted from the image. $y_m \in \mathbb{R}^K$ is a K-dimensional binary vector indicating which labels are annotated, and K is the number of categories. In the real-world case, y_m may be corrupted with annotation noise and thus incorrect. We assume the underlying clean label is $z_m \in \mathbb{R}^K$. We introduce s_m, a quality variable embedded in D-dimensional Gaussian space, to represent the annotation quality of y_m. For ease of reference, we list the notations of this paper in Table I.

Formally, it is a multi-label, multi-class classification problem with noise in labels. We target to train a deep classifier from these noisy training samples. There are many other tasks with this similar setting, like weakly supervised object detection and segmentation [9], [32]–[36] with web data.

B. Quality Embedding Model

1) Quality Embedding: In this section, we introduce a quality variable in parallel to the latent label, which jointly transit to the noisy image label. Our probabilistic graphical model is illustrated in Fig. 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

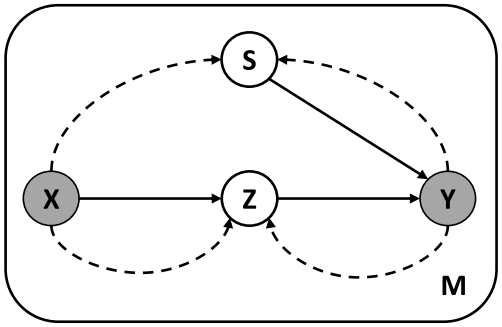


Fig. 2. Quality embedding model for noisy image labels. The shaded nodes as the observed variables are image X and its noisy label vector Y. The latent label vector Z and the quality variable vector S are latent variables. Solid lines and dashed lines represent the generative process and the inference process respectively.

two distributions with $q(Z|X, Y)$ and $q(S|X, Y)$, which play roles of posterior approximation.

According to Fig. 2, since it is usually intractable to directly maximize the log-likelihood, we construct an adjustable evidence lower bound (ELBO) [37]–[39] as follows,

$$\begin{aligned}\ln P(Y|X) &= \sum_{m=1}^{M} \ln P(y_m|x_m) \\ &\geq \sum_{m=1}^{M} \mathbf{E}_{q(z_m|x_m,y_m),q(s_m|x_m,y_m)} \left[\ln P(y_m|z_m, s_m)\right] \\ &\quad - \sum_{m=1}^{M} \mathbf{D_{KL}} \left[q(z_m|x_m, y_m)||P(z_m|x_m)\right] \\ &\quad - \sum_{m=1}^{M} \mathbf{D_{KL}} \left[q(s_m|x_m, y_m)||P(s_m)\right], \end{aligned} \tag{1}$$

where variational distributions $q(z_m|x_m, y_m)$ and $q(s_m|x_m, y_m)$ are to approximate the true distributions of z_m and s_m. The above bound is a relatively good approximation of the marginal log-likelihood, providing a basis for model selection [39]. As the gap between the marginal likelihood and the ELBO becomes zero, the variational distributions approach the true distributions.

2) Variational Mutual Information Regularizer: Although Fig. 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 models can resort to posterior regularizations [40]. Posterior regularizations ensure the desirable expectation and retain the computational efficiency simultaneously, which has been applied in clustering [41], classification [42] and image generation [43]. In this paper, we introduce regularization for variational distributions of Z and S based on mutual information maximization,

$$\begin{aligned}&\max \ I((Z, S); (X, Y)) \\ &\quad \simeq \frac{1}{M} \sum_{m=1}^{M} \mathbf{E}_{q(z_m|x_m,y_m)} \left[\ln q(z_m|x_m, y_m)\right] \\ &\qquad + \frac{1}{M} \sum_{m=1}^{M} \mathbf{E}_{q(s_m|x_m,y_m)} \left[\ln q(s_m|x_m, y_m)\right], \end{aligned} \tag{2}$$

where $I(;)$ means the mutual information of two distributions. The deduction can be found in APPENDIX A. In Eq. (2), maximizing the mutual information is equal to minimizing the entropy of z_m and s_m. This means, for z_m, such posterior regularization forces the label probability $q(z_m|x_m, y_m)$ close to the extreme points, and for s_m, it will favor the distribution $q(s_m|x_m, y_m)$ with a low variance.

3) Objective: Combining Eq. (1) with (2), our objective becomes the maximization of the ELBO along with the mutual information regularizer. Note that, we substitute $\frac{1}{M}$ in Eq. (2) 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 for simplicity.

$$\begin{aligned}\min \ \hat{L} &= -\sum_{m=1}^{M} \mathbf{E}_{q(z_m|x_m,y_m),q(s_m|x_m,y_m)} \left[\ln P(y_m|z_m, s_m)\right] \\ &\quad + \sum_{m=1}^{M} \mathbf{D_{KL}} \left[q(z_m|x_m, y_m)||P(z_m|x_m)\right] \\ &\quad + \sum_{m=1}^{M} \mathbf{D_{KL}} \left[q(s_m|x_m, y_m)||P(s_m)\right] \\ &\quad - \lambda \sum_{m=1}^{M} \mathbf{E}_{q(z_m|x_m,y_m)} \left[\ln q(z_m|x_m, y_m)\right] \\ &\quad - \lambda \sum_{m=1}^{M} \mathbf{E}_{q(s_m|x_m,y_m)} \left[\ln q(s_m|x_m, y_m)\right]. \end{aligned} \tag{3}$$

From Eq. (3), our model mainly differs from previous methods in the following three aspects.

- $P(y_m|z_m, s_m)$ indicates that the transition from the latent label to the noisy label is based on both z_m and s_m while previous methods [23], [24], [28] only depend on z_m.
- Previous works [23]–[25], [27], [28] use the linear transition $P(y_m|z_m)$ while our model applies nonlinear implementation $P(y_m|z_m, s_m)$.
- z_m and s_m are approximated with $q(z_m|x_m, y_m)$ and $q(s_m|x_m, y_m)$ in the posterior perspective while previous works [22], [27], [29] have to require the auxiliary data.

C. Contrastive-Additive Noise Network

In this section, we instantiate our model with a Contrastive-Additive Noise network (CAN) in Fig. 3. 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.

1) Architectures: For the **encoder module**, it is to model the variational distributions, $q(s_m|x_m, y_m)$ and $q(z_m|x_m, y_m)$. Concretely, we first forward x_m to a neural network to generate a prior label judgement $\hat{y}_m$. Then, according to $\hat{y}_m$ and y_m, we model the distribution parameters with two elaborately-designed layers. The neural network for $\hat{y}_m$ can be decided by the type of x_m. If x_m is the original image, then a convolutional neural network can be applied. While if x_m is a feature vector, a fully-connected network can be chosen. In Fig. 3, we take the convolutional neural network as an example. The **sampler module** is the implementation of Monte Carlo

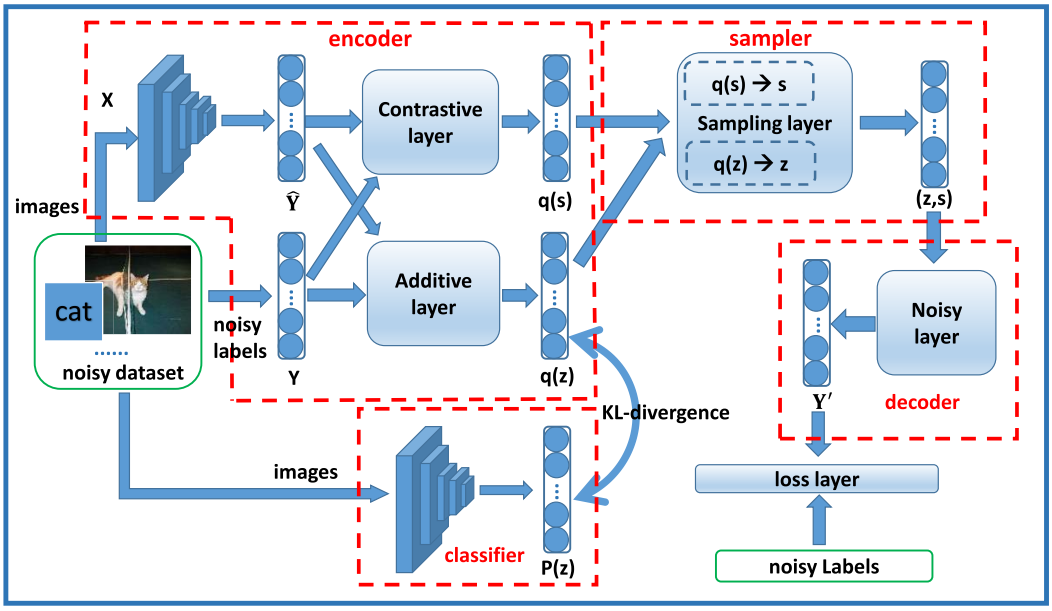


Fig. 3. The network consists of four modules, encoder, sampler, decoder and classifier, which are trained in an end-to-end manner. Encoder tries to learn latent labels and evaluate the quality of noisy labels; sampler is used to generate samples from the encoder outputs; decoder tries to recover noisy labels from samples. Meanwhile, our classifier is learned based on KL-divergence between $q(z)$ and $P(z)$.

sampling on $q(s_m|x_m, y_m)$ and $q(z_m|x_m, y_m)$. It receives the output of the encoder module and samples from the Gumbel and Gaussian distributions to generate a sample set of z_m and s_m, respectively. In the next section, we will discuss this part in details with reparameterization tricks. For the **decoder module**, it is a neural network, $P(y_m|z_m, s_m)$, 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_m to y_m. We consider the nonlinear transition since we have the heterogeneous quality variable s_m. The **classifier module** as our most important target $P(z_m|x_m)$, employs the same network in the encoder module. It is trained based on KL-divergence between $q(z_m|x_m, y_m)$ and $P(z_m|x_m, y_m)$.

2) Contrastive Layer and Additive Layer: We specially describe these two important layers in the encoder module. Regarding $q(s_m|x_m, y_m)$, 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_m and $\hat{y}_m$ 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 μ and $\log\sigma^2$ like [44], with an another fully-connected layer (function $\mathbf{f_t}(\cdot)$)

$$(\mu(x_m, y_m), \log\sigma^2(x_m, y_m)) = \mathbf{f_t}(\mathbf{f_s}(y_m) - \mathbf{f_s}(\hat{y}_m)).$$

This contrastive layer is built based on the assumption that the quality is related to the difference between y_m and $\hat{y}_m$. 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 $P(y_m|z_m, s_m)$. *This idea has never been explored in previous noise-aware deep learning approaches* [19]–[29].

Regarding the distribution $q(z_m|x_m, y_m)$, 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_m and $\hat{y}_m$ into a latent space, and then feeds their addition into another fully-connected layer plus a sigmoid function (function $\hat{\mathbf{f}}_t$), illustrated as follows,

$$q(z_m|x_m, y_m) = \hat{\mathbf{f}}_t(\mathbf{f_{ns1}}(y_m) + \mathbf{f_{ns2}}(\hat{y}_m)).$$

This design learns a posterior label z_m from y_m and $\hat{y}_m$ by a nonlinear combination with neural network. Previous methods [20]–[22], [29], [31] use a weight in their loss function to linearly combine the noisy label y_m 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, which will be explained in the next section. 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.

D. Optimization

In this section, we will analyze the difficulty in optimization and deduce an SGD algorithm with reparameterization tricks.

1) Reparameterization Tricks: The first term in the RHS of Eq. (3) has no closed form when either $q(z_m|x_m, y_m)$ or $q(s_m|x_m, y_m)$ is not conjugated with $P(y_m|z_m, s_m)$. Let alone we model the distributions with deep neural network in the paper. The general way is by the Monte Carlo sampling. However, Blei *et al.* [38] have shown when the derivative is about $q(z_m|x_m, y_m)$ or $q(s_m|x_m, y_m)$,

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 [44], [45] 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 [45] for z_m and the continuous reparameterization trick [44] for s_m as follows,

$$\begin{aligned} z_{mk} &= g(\gamma_{mk}) \\ &= \frac{\exp\left(\left(\ln q(z_{mk}=1|x_m,y_m)+\gamma_{mk1}\right)/\tau\right)}{\sum_{v=0}^{1}\exp\left(\left(\ln q(z_{mk}=v|x_m,y_m)+\gamma_{mkv}\right)/\tau\right)}, \\ s_m &= f(\zeta_m)=\mu(x_m,y_m)+\sigma^2(x_m,y_m)\odot\zeta_m, \end{aligned}$$

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

$$\begin{aligned} &\mathbf{E}_{q(z_m|x_m,y_m),q(s_m|x_m,y_m)}\left[\ln P(y_m|z_m,s_m)\right] \\ &\qquad \simeq \frac{1}{N}\sum_{n=1}^{N}\ln P(y_m|g(\gamma_m^{(n)}),f(\zeta_m^{(n)})), \end{aligned} \quad (4)$$

where N is the sample number of γ_m and ζ_m for the mth image. Based on Eq. (4), the first term in the RHS of Eq. (3) can be efficiently estimated, even though we set the sample number N equal to 1. The remaining terms in the RHS of Eq. (3) can be explicitly computed, which we present deduction in APPENDIX B. After transforming these terms in Eq. (3), 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 C.

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_m and s_m at the beginning of training, i.e., squeeze out the clean label information for z_m and leave the quality-related information to s_m. In this case, the whole optimization will suffer from the difficulty in convergence;
- The corresponding label order between z_m and y_m may be inconsistent in the optimization. For example, the category in the first dimension of z_m can be corresponding to the category in the second dimension of y_m.

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 C) with the following Eq. (5),

$$\nabla_{\theta_c}\hat{L}_{mod} = (1-\rho(t))\,\nabla_{\theta_c}\hat{L} + \rho(t)\nabla_{\theta_c}\hat{L}_{temp}, \quad (5)$$

where $\nabla_{\theta_c}\hat{L}_{temp}$ is gradient with regard to the cross-entropy loss between z_m and y_m, and $\rho(t) = \exp(-\alpha * t)$, $\alpha > 0$ is a time-varying term. With Eq.(5), 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_m and make the label order of z_m and y_m consistent.

3) Differences From VAE: The optimization procedure can be interpreted from the perspective of auto-encoder. However, our model is different from the conventional variational auto-encoder (VAE) [44] in three aspects.

- VAE is a generative model and usually used in unsupervised learning like image modeling [46]. We first extend it into the discriminative learning with noisy labels.
- As the illustration in Fig.4 (a), the structure of VAE is usually symmetric, that is, observed knowledge is encoded into latent variables and decoded to itself. However, our model in Fig.4 (b) is asymmetric since we introduce X as an auxiliary variable in the encoding-decoding procedure. Simultaneously, a discriminative classifier is involved and 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.

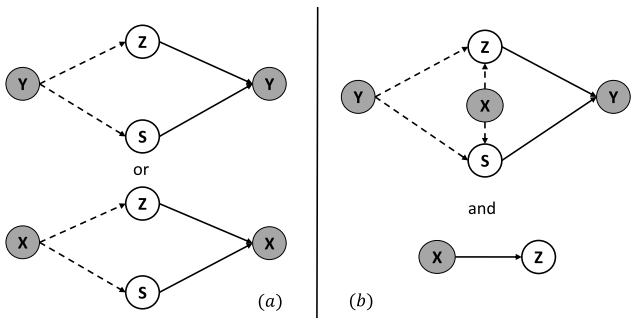


Fig. 4. Difference between the conventional auto-encoder and our model. Solid lines and dashed lines respectively represent generative (decoding) and inference (encoding) procedures. (a) a conventional auto-encoder is symmetric that observed knowledge is used to encode to latent variables and decoded symmetrically. (b) Our model uses an auxiliary variables X in this encoding-decoding procedure and meanwhile learns a discriminative part (X to Z).

IV. 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, hyper-parameter 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.

A. Datasets

We totally have five image datasets used in the experiments.

[1] γ_{mk1}, γ_{mk2} are both sampled by $-\log(-\log U)$, where $U \sim$ Uniform(0,1)

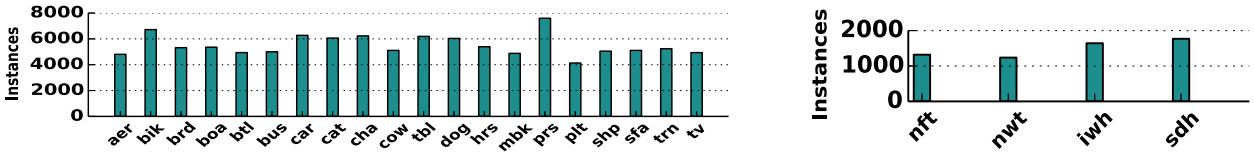


Fig. 5. **Left:** the instance number of each category in *WEB* dataset. **Right:** the instance number of each category in *AMT* dataset.

WEB[2] This dataset is a subset of YFCC100M [47] 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 [48]. The statistics of this dataset are shown in the left panel of Fig. 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 may contain annotation error.

AMT[3] This dataset is collected by Zhou *et al.* [16] from the Amazon Mechanical Turk platform. They submit four breeds of dog images from the Stanford Dog dataset [49] 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. There are 7,354 samples in total and the sample number in each category is between 1k and 2k. All images in this dataset belong to one class. Labels in this dataset may contain annotation error.

V07[4] This dataset is provided for the 20-cateogry classification task in PASCAL VOC Chanllenge 2007 [48]. 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[5] This dataset is provided for the 20-cateogry classification task in PASCAL VOC Chanllenge 2012 [50]. 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[6] This last dataset consists of 4 categories of dogs (same to [16]) in the Stanford Dog dataset [49]. 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.

B. Experimental Setup

For *WEB*, *V07* and *V12* datasets, a 34-layer residual network [4] 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[7] 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 [16]. 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 [45], we vary the temperature τ of the Gumbel-softmax function in Eq. (5) with $\max\left(0.5, \exp(-3\text{x}10^{-5}\text{xStep})\right)$. As for the parameter α in the annealing coefficient ρ, we set it as $4x10^{-1}/M$ where M is the sample number of the dataset. N in the sampler module is set to 1. The regularizer coefficient λ 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. All experiments run 90 epochs. For clarity, please refer to the pseudo algorithm in APPENDIX. D for more details. In evaluation, we adopt Average Precision (AP) and mean Average Precision (mAP) [48], [50] 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 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.

C. Classification Results

1) Training With Real-World Noisy Datasets: To demonstrate the effectiveness of the proposed method in classification, we compare CAN with three state-of-the-art approaches, LearnQ [24], ICNM [25] and Bootstrap [20]. 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.

The classification performance for each category on the *V07TE*, *V12TE* and *SD4TE* datasets is reported in Table. II, III, and IV. From the results in Table II and III, 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

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

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

[4]http://host.robots.ox.ac.uk/pascal/VOC/voc2007/

[5]http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[6]http://vision.stanford.edu/aditya86/ImageNetDogs/

[7]https://github.com/facebook/fb.resnet.torch

TABLE II
CLASSIFICATION RESULTS ON *V07TE*

Model	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
Resnet-N	93.5	85.3	90.1	85.1	51.2	82.3	84.8	91.2	59.3	87.1	72.1	88.7	91.3	88.9	76.1	54.4	87.6	70.0	90.4	61.4	79.5
LearnQ	92.8	86.1	91.0	87.8	50.2	84.9	85.1	90.9	59.2	88.3	71.1	90.1	91.2	88.1	78.3	56.6	89.1	73.1	90.7	64.3	80.4
ICNM	92.5	86.2	90.5	87.9	47.7	84.0	84.8	90.6	59.8	88.3	72.7	89.8	91.5	87.2	77.0	**57.0**	88.9	71.5	91.2	65.7	80.3
Bootstrap	94.0	**88.4**	90.3	88.2	51.7	83.8	86.5	91.0	65.4	88.0	77.4	90.4	91.8	**90.8**	79.8	55.2	92.8	75.2	90.8	66.4	81.9
CAN	**95.5**	87.0	**91.4**	**89.9**	**60.1**	**85.5**	**87.6**	**92.0**	**67.2**	**90.1**	**77.7**	**91.8**	**93.3**	90.6	**82.1**	56.0	**93.6**	**80.7**	**94.5**	**70.6**	**83.8**

TABLE III
CLASSIFICATION RESULTS ON *V12TE*

Model	aer	bik	brd	boa	btl	bus	car	cat	cha	cow	tbl	dog	hrs	mbk	prs	plt	shp	sfa	trn	tv	mAP
Resnet-N	98.4	81.1	92.9	88.7	57.0	87.4	73.2	96.6	63.3	90.0	63.9	94.3	95.0	92.9	76.8	43.8	92.9	67.2	93.1	65.1	80.7
LearnQ	98.4	83.8	93.8	88.5	53.5	87.8	73.7	96.5	64.3	90.6	62.6	94.6	96.1	91.6	78.4	**46.8**	92.8	69.0	94.0	65.4	81.1
ICNM	98.1	82.9	93.6	88.9	53.4	87.7	72.3	96.2	64.7	91.2	66.3	94.2	96.2	91.4	78.0	44.0	93.5	69.3	94.4	66.9	81.2
Bootstrap	98.6	**84.1**	93.6	90.9	56.3	89.8	75.5	96.3	69.8	91.6	69.9	94.4	95.8	93.2	82.2	43.2	92.8	70.9	95.4	67.4	82.6
CAN	**98.8**	**84.1**	**95.3**	**93.2**	**62.1**	**90.8**	**77.0**	**97.9**	**72.6**	**94.4**	**73.5**	**96.1**	**97.7**	**94.3**	**82.4**	45.5	**95.8**	**71.4**	**95.8**	**68.6**	**84.4**

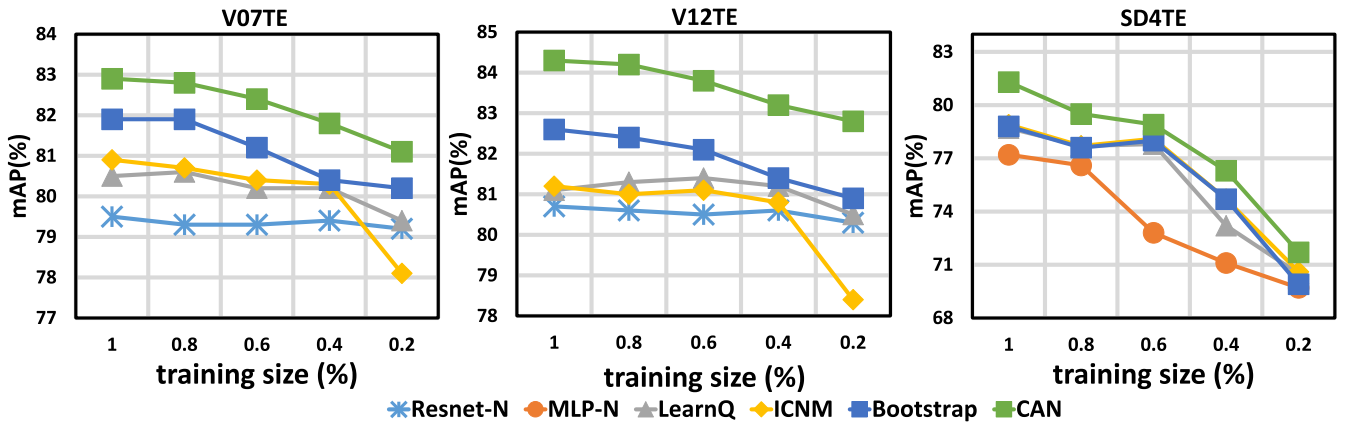


Fig. 6. Classification results with different training sizes. We sample the subsets of *WEB* and *AMT* with five different ratios for training, and evaluate all models on *V07TE*, *V12TE* and *SD4TE* datasets.

TABLE IV
CLASSIFICATION RESULTS ON SD4TE

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**

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. IV, 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 experiments, we have below explanations.

- LearnQ and ICNM, which only introduce the latent label to handle the label noise, cannot prevent noise from degenerating the classifier sufficiently.
- 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 of predictions and noisy labels, which still cannot prevent the error back-propagation from label noise.
- Our approach, which on one hand models the trustworthiness of noisy labels to reduce the noise effect, and on the other hand estimates the latent label in the posterior perspective to train the classifier, shows better classification performance.

2) 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 Fig. 6.

From Fig. 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 Fig. 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

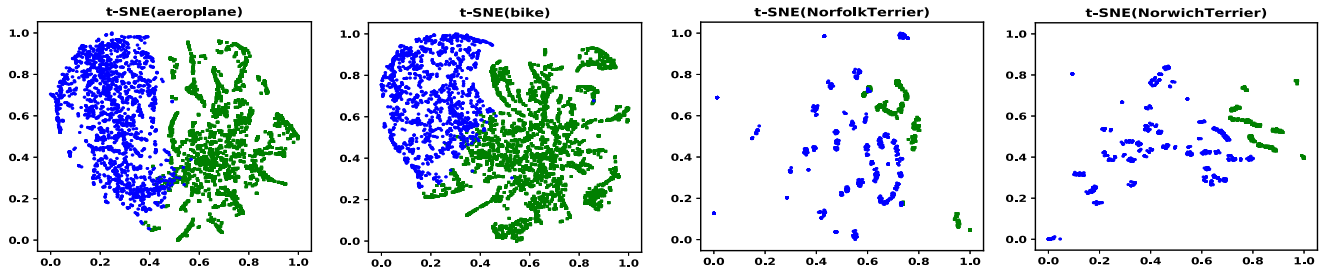


Fig. 7. Quality embedding visualization of two categories in *WEB* dataset and two categories in *AMT* dataset. Better distinguishability of clusters indicates better identifiability of mismatches between the latent labels and the noisy labels. Blue: trustworthy embedding, Green: non-trustworthy embedding.

TABLE V
CLASSIFICATION RESULTS WITH DIFFERENT λ IN CAN

λ	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

(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 Fig. 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.

3) Hyperparameter Sensitivity: To investigate the reliability of CAN with different regularizer coefficients, we set λ to 0, 0.2, 0.5, 1, 5, 10, respectively, to validate its effect. The results are illustrated in Table V. From this table, we find the performance on all datasets first grows to a peak and then gradually decreases with increasing λ. 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 λ between 0 to 1 makes the variational mutual information regularizer collaborates well with KL-divergences.

4) 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. VI.

As shown in Table. VI, when the corruption probability $P_{Noise} = 1.0$, the classification results of all models are almost random. With P_{Noise} varying from 1.0 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. VI 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 may be worse than models without considering noise. (3) CAN shows advantages in different noise levels compared with existing methods.

D. Model Visualization

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

1) 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 quality variable plus the embedding type, a low dimensional visualization of quality embedding can be illustrated with t-SNE package [51].

In Fig. 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. 7, the embedding in each category exhibits two distinguishable clusters. It indicates CAN can identify mismatches between latent 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.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.

TABLE VI
MODEL PERFORMANCE (MAP) WITH QUANTITATIVE NOISE

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

Fig. 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).

2) 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. 8, we present 20 examples 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. 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. 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. 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.

3) 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

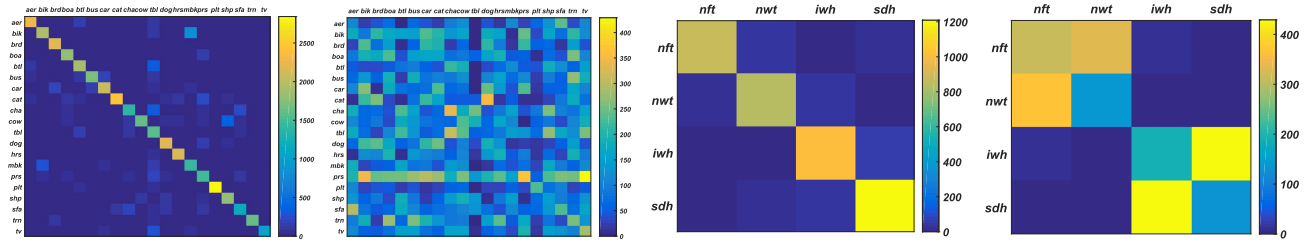


Fig. 9. Transition patterns among labels conditioned on the trustworthy embedding and the non-trustworthy embedding on *WEB* dataset (the first two panels) and *AMT* dataset (the last two panels). Transition conditioned on the trustworthy embedding requires the consistency between the latent label and 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.

two types of embeddings. In Fig. 9, we respectively plot two transition patterns with heatmaps for *WEB* dataset and *AMT* dataset.

As shown for *WEB* dataset in Fig. 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. 9 also have these characteristics. Fig. 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. 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 to TV itself. Similarly in the fourth panel of Fig. 9, the transition on *AMT* usually exists in the appearance-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.

V. Conclusion

In this paper, we present a quality embedding model to learn the classifier from noisy image labels, which effectively avoids the error back-propagated from label noise. To instantiate the model, a Contrastive-Additive Noise network is well-designed. Regarding parameter estimation, we deduce an efficient SGD optimization algorithm by applying recent discrete and continuous reparameterization tricks. 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 paper, we validate our model on image data only, and other types of contents will be further explored.

Appendix A
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.

$$
\begin{aligned}
&I((Z,S):(X,Y)) \\
&\quad = -H(Z,S|X,Y) + H(Z,S) \\
&\quad = \mathbf{E}_{P(X,Y)}\left[\mathbf{E}_{q(Z,S|X,Y)}\left[\ln q(Z,S|X,Y)\underbrace{P(X,Y)}_{\text{prior}}\right]\right] \\
&\qquad - \mathbf{E}_{P(Z,S)}\left[\ln \underbrace{P(Z,S)}_{\text{prior}}\right] \\
&\quad \simeq \frac{1}{M}\sum_{m=1}^{M}\mathbf{E}_{q(z_m,s_m|x_m,y_m)}\left[\ln q(z_m,s_m|x_m,y_m)\right] + \text{const} \\
&\quad \simeq \frac{1}{M}\sum_{m=1}^{M}\mathbf{E}_{q(z_m|x_m,y_m)}\left[\ln q(z_m|x_m,y_m)\right] \\
&\qquad + \frac{1}{M}\sum_{m=1}^{M}\mathbf{E}_{q(s_m|x_m,y_m)}\left[\ln q(s_m|x_m,y_m)\right]. \qquad (6)
\end{aligned}
$$

Appendix B
KL-Divergences and Regularizers

The remaining four terms in the RHS of Eq. (3) can be calculated without sampling. For example, for the latent label z_m, both $q(z_m|x_m,y_m)$ and $P(z_m|x_m)$ are two K-dimensional multinomial probabilities. Their KL-divergence term and regularizer can be simplified by enumerating each dimension. For the quality variable s_m, it is from the D-dimensional Gaussian space $N(\mu(x_m,y_m), \text{diag}(\sigma^2(x_m,y_m)))$ whose parameters are implicitly modeled with network of input x_m and y_m. If we assume its prior $P(s_m)$ is $N(\mathbf{0},\mathbf{1})$ like [44], it is easy to compute their KL-divergence and the regularizer due to the conjugation. In Eq. (7), we give their simplifications

bigeminally.

$$
\begin{aligned}
&\mathbf{D_{KL}}\left[q(z_m|x_m, y_m)||P(z_m|x_m)\right] \\
&\quad - \lambda \mathbf{E}_{q(z_m|x_m,y_m)}\left[\ln q(z_m|x_m, y_m)\right] \\
&= \sum_{k=1}^{K}\sum_{z_{mk}} q(z_{mk}|x_m, y_m) \ln \frac{q(z_{mk}|x_m, y_m)^{1-\lambda}}{P(z_{mk}|x_m)}, \\
&\mathbf{D_{KL}}\left[q(s_m|x_m, y_m)||P(s_m)\right] - \lambda \mathbf{E}_{q(s_m|x_m,y_m)}\left[\ln q(s_m|x_m, y_m)\right] \\
&= -\frac{1}{2}\sum_{d=1}^{D}\Big((1-\lambda)\ln\sigma_d^2(x_m, y_m) - \sigma_d^2(x_m, y_m)\Big) \\
&\quad + \frac{1}{2}\mu(x_m, y_m)^T\mu(x_m, y_m) + \text{const.} \qquad (7)
\end{aligned}
$$

Appendix C
Stochastic Variational Gradient

Assume θ_{de}, θ_{add}, θ_{con}, θ_c and θ_{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.

$$
\begin{aligned}
&\nabla_{\theta_{de}}\hat{L} \\
&= -\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\nabla_{\theta_{de}}\ln P(y'_m|g(\gamma_m^{(n)}), f(\zeta_m^{(n)})) \\
&\nabla_{\theta_{add}}\hat{L} \\
&= -\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\nabla_g P(y'_m|g(\gamma_m^{(n)}), f(\zeta_m^{(n)}))\nabla_{\theta_{add}} g(\gamma_m^{(n)}) \\
&\quad + \sum_{m=1}^{M}\sum_{k=1}^{K}\left(\ln\frac{q(z_{mk1}|x_m, y_m)^{1-\lambda}\,(1-P(z_{mk1}|x_m))}{(1-q(z_{mk1}|x_m, y_m))^{1-\lambda}\,P(z_{mk1}|x_m)}\right) \\
&\quad \nabla_{\theta_{add}} q(z_{mk1}|x_m, y_m) \\
&\nabla_{\theta_{con}}\hat{L} \\
&= -\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\nabla_f P(y'_m|g(\gamma_m^{(n)}), f(\zeta_m^{(n)}))\nabla_{\theta_{add}} f(\zeta_m^{(n)}) \\
&\quad + \sum_{m=1}^{M}\frac{1}{2}\nabla_{\theta_{add}}\sum_{d=1}^{D}\Big(\sigma_d^2(x_m, y_m) - (1-\lambda)\ln\sigma_d^2(x_m, y_m)\Big) \\
&\quad + \sum_{m=1}^{M}\frac{1}{2}\nabla_{\theta_{add}}\Big(\mu(x_m, y_m)^T\mu(x_m, y_m)\Big) \\
&\nabla_{\theta_c}\hat{L} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\frac{P(z_{mk1}|x_m) - q(z_{mk1}|x_m, y_m)}{P(z_{mk1}|x_m)(1-P(z_{mk1}|x_m))}\nabla_{\theta_c}P(z_{mk1}|x_m) \\
&\nabla_{\theta_{pri}}\hat{L} \\
&= \sum_{m=1}^{M}\sum_{k=1}^{K}\nabla_{\theta_c}(y_{mk}\ln P(\hat{y}_m|x_m) \\
&\quad + (1-y_{mk})\ln(1-P(\hat{y}_m|x_m))) \qquad (8)
\end{aligned}
$$

where z_{mk1} is abbreviation of $z_{mk} = 1$ to avoid notation clutters.

Algorithm 1 Contrastive-Additive Neural network

Input: $\mathscr{D}$ with M samples, parameters $\{\theta_{de}, \theta_{add}, \theta_{con}, \theta_c, \theta_{pri}\}$, learning rate κ, temperature τ, annealing coefficient ρ, regularizer coefficient λ, batch size B, epoch number N, step counter s.
Initial: $\kappa = 0.01$, $\rho = 1.0$, $\lambda = 0.3$, $B = 50$, $N = 90$, $s = 0$.
for $n = 1$ **to** N **do**
  for $i = 1$ **to** $\lceil \frac{M}{B} \rceil$ **do**
    Sample a batch $\widehat{\mathscr{D}}$ from $\mathscr{D}$.
    $\tau = \max(0.5, \exp(-3 * 10^{-5} * s))$.
    forward
      Compute $P(\hat{y}_m|x_m)$ and treat as $\hat{y}_m$.
      Compute $q(s_m|\hat{y}_m, y_m)$ via the contrastive layer.
      Compute $q(z_m|\hat{y}_m, y_m)$ via the additive layer.
      Compute $P(y_m|x_m)$ via the classifier.
      Sample s_m and z_m via the sampler module.
      Compute $P(y'_m|x_m)$ via the decoder module.
    end
    backward
      Update θ_{de}, θ_{add}, θ_{con}, θ_{pri} with Eq. (8).
      Update the classifier parameter θ_c with Eq. (5).
    end
    $s = s + 1$.
    $\rho = \exp(-4 * 10^{-1}/M * s)$.
  end for
  if $n\%30 == 0$ **then**
    $\kappa = \kappa * 0.1$.
  end if
end for

Appendix D
Pseudo Training Algorithm

Assuming we have one multi-label multi-class noisy dataset

$$\mathscr{D} : \{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\},$$

we target to train a classifier with CAN. The training procedure is simply summarized in Algorithm 1.

According to ur 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 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.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[3] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[5] S. K. Divvala, A. Farhadi, and C. Guestrin, "Learning everything about anything: Webly-supervised visual concept learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3270–3277.

[6] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1431–1439.

[7] R. Krishna *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[8] Z. Li and J. Tang, "Weakly supervised deep matrix factorization for social image understanding," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 276–288, Jan. 2017.

[9] L. Wang, G. Hua, J. Xue, Z. Gao, and N. Zheng, "Joint segmentation and recognition of categorized objects from noisy Web image collection," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4070–4086, Sep. 2014.

[10] J. Yang, X. Sun, Y.-K. Lai, L. Zheng, and M.-M. Cheng, "Recognition from Web data: A progressive filtering approach," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5303–5315, Nov. 2018.

[11] J. Zhang and L. Ye, "Content based image retrieval using unclean positive examples," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2370–2375, Oct. 2009.

[12] 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 Trans. Image Process.*, vol. 26, no. 10, pp. 4871–4884, Oct. 2017.

[13] V. C. Raykar *et al.*, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, Apr. 2010.

[14] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1196–1204.

[15] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2014.

[16] D. Zhou, S. Basu, Y. Mao, and J. C. Platt, "Learning from the wisdom of crowds by minimax entropy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2195–2203.

[17] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[18] S. A. Goldman and R. H. Sloan, "Can PAC learning algorithms tolerate random attribute noise?" *Algorithmica*, vol. 14, no. 1, pp. 70–84, 1995.

[19] S. Azadi, J. Feng, S. Jegelka, and T. Darrell. (2016). "Auxiliary image regularization for deep CNNs with noisy labels." [Online]. Available: https://arxiv.org/abs/1511.07069

[20] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. (2014). "Training deep neural networks on noisy labels with bootstrapping." [Online]. Available: https://arxiv.org/abs/1412.6596

[21] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, and A. Hertzmann, "Deep classifiers from image tags in the wild," in *Proc. Workshop Commun.-Org. Multimodal Mining, Opportunities Novel Solutions*, Oct. 2015, pp. 13–18.

[22] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. (2017). "Learning from noisy labels with distillation." [Online]. Available: https://arxiv.org/abs/1703.02391

[23] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 567–574.

[24] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. (2015). "Training convolutional networks with noisy labels." [Online]. Available: https://arxiv.org/abs/1406.2080

[25] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2930–2939.

[26] G. Patrini, A. Rozza, A. Menon, R. Nock, and L. Qu. (2016). "Making neural networks robust to label noise: A loss correction approach." [Online]. Available: https://arxiv.org/abs/1609.03683

[27] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2691–2699.

[28] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 967–972.

[29] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6575–6583.

[30] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, 2010.

[31] A. Joulin, L. Van Der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 67–84.

[32] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 431–445.

[33] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2846–2854.

[34] W. Zhang, S. Zeng, D. Wang, and X. Xue, "Weakly supervised semantic segmentation for social images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2718–2726.

[35] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1665–1674.

[36] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, "Learning from weak and noisy labels for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 486–500, Mar. 2017.

[37] M. I. Jordan and M. J. Wainwright, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2008.

[38] D. M. Blei, M. I. Jordan, and J. W. Paisley, "Variational Bayesian inference with stochastic search," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, J. Langford and J. Pineau, Eds. New York, NY, USA, 2012, pp. 1367–1374.

[39] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Statist. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[40] K. Ganchev, J. Gracça, J. Gillenwater, and B. Taskar, "Posterior regularization for structured latent variable models," *J. Mach. Learn. Res.*, vol. 11, pp. 2001–2049, Jul. 2010.

[41] A. Krause, P. Perona, and R. G. Gomes, "Discriminative clustering by regularized information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 775–783.

[42] J. Zhu, N. Chen, and E. P. Xing, "Bayesian inference with posterior regularization and applications to infinite latent svms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1799–1847, 2014.

[43] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 2172–2180.

[44] D. P. Kingma and M. Welling. (2015). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114

[45] E. Jang, S. Gu, and B. Poole. (2016). "Categorical reparameterization with gumbel-softmax." [Online]. Available: https://arxiv.org/abs/1611.01144

[46] A. Van Den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with pixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4790–4798.

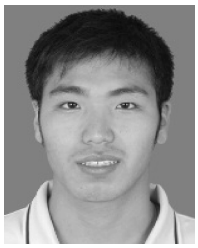
[47] B. Thomee *et al.*, "The new data and new challenges in multimedia research," *Commun. ACM*, vol. 59, no. 2, pp. 64–73, 2015.

[48] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

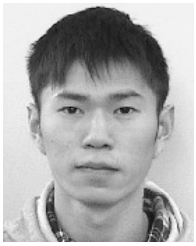
[49] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. Workshop Fine-Grained Vis. Categorization*, Colorado Springs, CO, USA, Jun. 2011, pp. 1–2.

[50] M. Everingham, S. M. Ali Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.

[51] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014.



Jiangchao Yao received the B.S. degree in information engineering from the South China University of Technology, Guangzhou, China, in 2013. He is currently pursuing the dual Ph.D. degree under the supervision of Y. Zhang with Shanghai Jiao Tong University and under the supervision of I. W. Tsang with the University of Technology Sydney. His research interests include representation learning and multimedia analysis.



Jiajie Wang received the B.S. degree in information engineering from the Nanjing University of Posts and Telecommunications, China, in 2016. He is currently pursuing the master's degree in information and communication engineering with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China. His research is focused on computer vision and weakly supervised learning.

Ivor W. Tsang is currently a Professor of artificial intelligence at the University of Technology Sydney (UTS). He is also the Research Director of the UTS Flagship Research Centre for Artificial Intelligence. His research focuses on transfer learning, feature selection, crowd intelligence, big data analytics for data with extremely high dimensions in features, samples and labels, and their applications to computer vision and pattern recognition. He has authored more than 170 research papers published in top-tier journal and conference papers. According to Google Scholar, he has more than 10,000 citations and his H-index is 50. In 2009, he was conferred the 2008 Natural Science Award (Class II) by the Ministry of Education, China, which recognized his contributions to kernel methods. In 2013, he received the prestigious Australian Research Council Future Fellowship for his research regarding Machine Learning on Big Data. In addition, he had received the prestigious IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2007, the 2014 IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010.

Ya Zhang received the bachelor's degree from Tsinghua University, China, and the Ph.D. degree in information sciences and technology from Pennsylvania State University. Before joining Shanghai Jiao Tong University, she was a Research Manager at Yahoo! Labs, Sunnyvale, CA, USA, where she leaded a research and development team of researchers with strong background in data mining and machine learning to improve the web search quality of Yahoo international markets. Prior to joining Yahoo, she was an Assistant Professor at the University of Kansas with research focus on machine learning applications in bioinformatics and information retrieval. She is currently a Professor with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University. She has published more than 70 refereed papers in prestigious international conferences and journals, including TPAMI, TIP, TNNLS, ICDM, CVPR, ICCV, ECCV, and ECML. She currently holds five U.S. patents and four China patents, and has nine patents pending in the areas of multimedia analysis. She is appointed as the Chief Expert for the project Research of Key Technologies and Demonstration for Digital Media Self-organizing under the 863 program by the Ministry of Science and Technology of China. Her research interest is mainly on machine learning and data mining with applications to multimedia information retrieval, social network analysis, and intelligent information system.

Jun Sun received the B.S. degree from the University of Electronic Sciences and Technology of China, Chengdu, China, in 1989, and the Ph.D. degree from Shanghai Jiao Tong University, in 1995, all in electrical engineering. In the past five years, he has been responsible for several national projects in DTV and IPTV fields. He is currently a Professor and a Ph.D. Advisor with Shanghai Jiao Tong University. He has published over 50 technical papers in the area of digital television and multimedia communications. His research interests include digital television, image communication, and video encoding. He received the 2nd Prize of the National Sci. & Tech. Development Award in 2003 and 2008, respectively.

Chengqi Zhang (SM'95) received the Ph.D. degree from the University of Queensland, Brisbane, Australia, in 1991, and the D.Sc. degree (higher doctorate) from Deakin University, Geelong, Australia, in 2002. Since 2001, he has been a Professor of information technology with the University of Technology Sydney, Australia, and has been the Executive Director of UTS Data Science since 2016. Since 2005, he has been the Chairman of the Australian Computer Society National Committee for Artificial Intelligence. He has published more than 200 research papers, including several in first-class international journals, such as *Artificial Intelligence*, and the IEEE and *ACM Transactions*. He has published six monographs and edited 16 books, and has attracted 11 Australian Research Council grants. His research interests mainly focus on data mining and its applications. He has served as an Associate Editor for three international journals, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008; and as general chair, PC chair, or organizing chair for five international conferences, including ICDM 2010 and WI/IAT 2008. He was/is general co-chair of KDD 2015 in Sydney, the local arrangements chair of IJCAI-2017 in Melbourne, a fellow of the Australian Computer Society.

Rui Zhang received the B.S. and M.S. degrees from the Hefei University of technology, Hefei, China, in 1995 and 1999, respectively, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. Since 1999, she was at the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University. Her research interests include image communication, image processing, and DTV.