This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Learning from Noisy Web Data with Category-level Supervision

Li Niu[+], Qingtao Tang[*], Ashok Veeraraghavan[+], and Ashu Sabharwal[+]
[+]Department of Electrical and Computer Engineering, Rice University
[*]Department of Computer Science and Technology, Tsinghua University
[+] {ln7,vashok,ashu}@rice.edu, [*] tqt15@mails.tsinghua.edu.cn

Abstract

Learning from web data is increasingly popular due to abundant free web resources. However, the performance gap between webly supervised learning and traditional supervised learning is still very large, due to the label noise of web data as well as the domain shift between web data and test data. To fill this gap, most existing methods propose to purify or augment web data using instance-level supervision, which generally requires heavy annotation. Instead, we propose to address the label noise and domain shift by using more accessible category-level supervision. In particular, we build our deep probabilistic framework upon variational autoencoder (VAE), in which classification network and VAE can jointly leverage category-level hybrid information. Then, we extend our method for domain adaptation followed by our low-rank refinement strategy. Extensive experiments on three benchmark datasets demonstrate the effectiveness of our proposed method.

1. Introduction

The recent success of image classification is largely fueled by available large-scale image datasets. However, manually annotating large-scale dataset is time-consuming and labor-intensive. So it is unsurprising that learning from web images becomes increasingly popular because of the large amount of freely available web data. However, the labels of web images crawled from public website are very noisy and often inaccurate. Moreover, the data distributions between web data (*i.e.*, source domain) and test data (*i.e.*, target domain) are quite different, which is known as domain shift. Therefore, when applying the classifier learnt on the noisy web training images to the test images, the performance will be significantly degraded. Although abundant research works are intended to tackle the label noise and domain shift [42, 6, 3, 49, 57, 34], webly supervised learning is still struggling to compete with conventional supervised learning. To facilitate webly supervised learning, some works tend to utilize extra supervision to purify or augment the web data by selecting informative web data to label [17] or leveraging strong supervision (*e.g.*, clean images, part landmarks, or bounding boxes) from well-labeled dataset [50, 52]. However, these approaches are in high demand of manual annotations on the instance level, which are generally difficult to acquire.

Compared with instance-level supervision, category-level supervision is more accessible in real world. One of the most prominent category-level supervision is attribute, which is manually designed semantic cue for each category [9, 18] such as the shape (*e.g.*, cylindrical), material (*e.g.*, cloth), and color (*e.g.*, white). When attribute is not available, an alternative choice is the word vector, *i.e.*, a real-valued vector, corresponding to each category name obtained based on free online corpus (*e.g.*, Wikipedia) [1, 10, 39, 54]. Besides, we can also summarize category-level visual information based on free web images despite their inaccurate labels (see Section 6). Therefore, the availability of category-level information motivates us to explore learning from web data with category-level supervision, which is in the middle ground between no extra supervision and instance-level supervision.

In order to cope with label noise with category-level supervision, we opt for autoencoder-like network structure, because autoencoder has been used for outlier detection [37, 46] and its hidden layer can be regulated by prior information, which is suitable for our task. Specifically, we build our probabilistic framework upon a probabilistic variant of autoencoder, *i.e.*, variational autoencoder (VAE) [14, 35], because its provided reconstruction probability density can be easily integrated into our probabilistic framework. However, outlier detection is a non-trivial task for VAE trained on multi-category noisy data. To tackle this issue, we propose a framework named Webly Supervised learning with Category-level Information (WSCI), as illustrated in Figure 1. It can be seen from Figure 1 that our network consists of a classification network in the top flow and variational autoencder (VAE) in the bottom flow, which share common modules (*i.e.*, CNN and encoder) and jointly leverage category-level information. The classification network and VAE influence each other in the following way. On one hand, VAE detects outliers to help learn a more robust classification network by assigning higher weights on the losses of identified non-outliers. On the other hand, the classification network injects relatively accurate discriminative information into the hidden layer to learn a semantic VAE for better outlier detection.

With the aim to further address the domain shift between web data and test data, we extend our WSCI method to WSCI-DA by reconstructing unlabeled test instances via VAE in the training stage. Moreover, we also propose a novel low-rank refinement strategy following WSCI-DA.

2. Related Work

Webly Supervised Image Classification: Recently, abundant research works [7, 3, 27, 19, 30, 31, 28] are intended to address the label noise and domain shift when learning from web data. More recently, several CNN approaches were proposed for webly supervised learning [49, 42, 6, 57, 34, 8]. To fill the performance gap between webly supervised learning and traditional supervised learning, some research works resort to auxiliary information such as selective labeling [17] or extraneous strong supervision [50, 52] (*e.g.*, clean images, part landmarks, or bounding boxes), which involve human annotation on the instance level. In the contrast, we tend to boost the performance of webly supervised learning using more accessible category-level information.

Variational Autoencoder: Variational autoencoder (VAE) [14, 35] is a probabilistic generative model and its technical details will be introduced in Section 3. Several works [2, 41] use VAE for outlier detection, but they did not discuss how to handle the label noise in the multi-category training data. Recently, one promising research direction is to regulate the hidden layer of VAE more heavily such as conditional VAE (CVAE) [40, 45, 53], adversarial autoencoder (AAE) [22], and semi-supervised VAE [13]. Generally speaking, our method also falls into this scope, *i.e.*, regulating the hidden layer of VAE. However, our method regulates the hidden layer of VAE to eliminate the label noise from training data, which has not been explored in the above works [40, 45, 53, 22, 13].

Domain Adaptation: Domain adaptation (DA) methods [51, 26, 25, 29] aim to alleviate the domain shift between the source domain (*i.e.*, training set) and the target domain (*i.e.*, test set). Among existing DA approaches, the closest related works are autoencoder based DA methods [5, 12, 21] and low-rank based DA methods [11, 38]. However, all these methods only focus on domain adaptation while our WSCI-DA method can cope with the label noise and simultaneously address the domain issue when learning from web data. Besides, our proposed low-rank refinement following WSCI-DA is also quite different from [11, 38] because our low-rank reconstruction is based on latent variables on the hidden layer while theirs are based on visual features.

3. Background

In the remainder of this paper, for better representation, we denote a matrix/vector by using a uppercase/lowercase letter in boldface (*e.g.*, $\mathbf{A}$ denotes a matrix and $\mathbf{a}$ denotes a vector). We use $\mathbf{I}$ (*resp.*, $\mathbf{0}$) to denote identity matrix (*resp.*, all-zero vector/matrix). $\mathbf{A}^T$ is used to denote the transpose of $\mathbf{A}$. Moreover, we use $\mathbf{A} \circ \mathbf{B}$ to denote the element-wise product between $\mathbf{A}$ and $\mathbf{B}$.

Now we introduce the background knowledge of variational autoencoder (VAE), upon which our probabilistic framework is built. Assume data $\mathbf{x}$ can be generated from latent variable $\mathbf{z}$, then the marginal likelihood of $\mathbf{x}$ can be represented as $p_{\boldsymbol{\theta}_1}(\mathbf{x}) = \int p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z}) p_{\boldsymbol{\theta}_1}(\mathbf{z}) d\mathbf{z}$ with generative parameters $\boldsymbol{\theta}_1$, in which $p_{\boldsymbol{\theta}_1}(\mathbf{z})$ is the prior over latent variable $\mathbf{z}$ and $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})$ is the likelihood of $\mathbf{x}$ given $\mathbf{z}$.

However, $\int p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z}) p_{\boldsymbol{\theta}_1}(\mathbf{z}) d\mathbf{z}$ is intractable over all configurations of latent variables. To solve this issue, variational autoencoder (VAE) [14, 35] introduces approximate posterior $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$ with variational parameters $\boldsymbol{\theta}_2$. Then, instead of maximizing the marginal likelihood $p_{\boldsymbol{\theta}_1}(\mathbf{x})$, VAE proposes to maximize the lowerbound of marginal likelihood $p_{\boldsymbol{\theta}_1}(\mathbf{x})$, *i.e.*, Evidence Lower BOund (ELBO), which is equal to minimizing the following objective function (please refer to [14] for technical details):

$$\mathrm{KL}[q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}) || p_{\boldsymbol{\theta}_1}(\mathbf{z})] - \mathbb{E}_{q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})], \quad (1)$$

in which the first regularizer is a penalty enforcing the learnt $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$ to be close to the given prior $p_{\boldsymbol{\theta}_1}(\mathbf{z})$ based on KL divergence between $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}_1}(\mathbf{z})$, and the second regularizer is the reconstruction error measuring the truthfulness of reconstruction based on the expectation of $\log p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})$ w.r.t. $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$. In summary, the objective function in (1) aims to reduce the reconstruction error as well as the KL divergence between the approximate posterior and prior of latent variables at the same time. Note that it is generally assumed that $p_{\boldsymbol{\theta}_1}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ for simplicity. In this paper, following [14], we assume $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})$ (*resp.*, $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$) to be a multivariate Gaussian with diagonal covariance, *i.e.*, $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \mathrm{diag}(\boldsymbol{\sigma}_x^2))$ (*resp.*, $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_z, \mathrm{diag}(\boldsymbol{\sigma}_z^2))$), which is specified by a probabilistic decoder (*resp.*, encoder) network with model parameters $\boldsymbol{\theta}_1$ (*resp.*, $\boldsymbol{\theta}_2$).

The forward process of VAE consists of three steps: 1) generate approximate posterior $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$ (*i.e.*, $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$) using probabilistic encoder based on $\mathbf{x}$; 2) sample latent variables $\mathbf{z}$ based on $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})$; 3) generate likelihood of $\mathbf{x}$ given $\mathbf{z}$, $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})$ (*i.e.*, $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x$), using probabilistic decoder based on the sampled $\mathbf{z}$. The above procedure can be seen from the bottom flow in Figure 1. Note

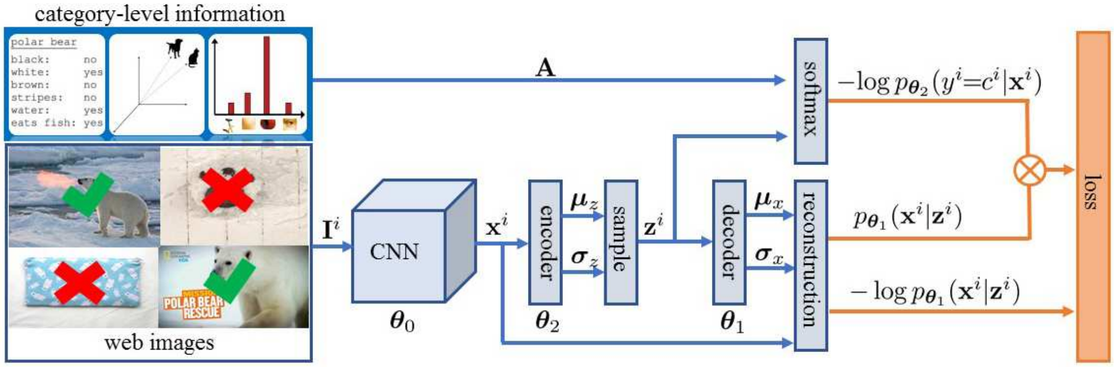

Figure 1: Flowchart of learning from noisy web data with category-level semantic information. The top flow is classification network and the bottom flow is variational autoencoder. Two flows share the common model parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_2$.

that reparameterization trick is generally used in the second step for ease of optimization [14]. Specifically, instead of sampling directly from approximate posterior $q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z};\boldsymbol{\mu}_z,\text{diag}(\boldsymbol{\sigma}_z^2))$, we can use deterministic mapping $\mathbf{z} = g_{\boldsymbol{\theta}_2}(\mathbf{x},\boldsymbol{\epsilon}) = \boldsymbol{\mu}_z + \boldsymbol{\sigma}_z \circ \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$, because it is proved that $\mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\boldsymbol{\mu}_z,\text{diag}(\boldsymbol{\sigma}_z^2))} f(\mathbf{z}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})} f(\boldsymbol{\mu}_z + \boldsymbol{\sigma}_z \circ \boldsymbol{\epsilon}) \approx \frac{1}{L}\sum_{l=1}^{L} f(\boldsymbol{\mu}_z + \boldsymbol{\sigma}_z \circ \boldsymbol{\epsilon}^l)$ with L being the number of samples per training instance. In practice, each instance can be sampled only once (*i.e.*, $L = 1$) per training epoch as long as the batch size is large enough [14].

4. Webly Supervised Learning with Category-level Information

In this section, we build our method named Webly Supervised learning with Category-level Information (WSCI) upon variational autoencoder (VAE), in which the classification network and VAE can jointly leverage category-level information to handle label noise.

4.1. Semantic VAE for Outlier Detection

We rewrite the objective function of VAE introduced in Section 3 as follows,

$$\text{KL}[q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}_1}(\mathbf{z})] - \mathbb{E}_{q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})], \qquad (2)$$

in which the reconstruction probability density $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z})$ can be used for detecting outliers [2, 41]. Specifically, an instance $\mathbf{x}^i$ is identified as an outlier if $\mathbb{E}_{q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}^i)} p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z})$ is below certain threshold. In ideal cases, VAE should be trained on clean training data so as to learn a normal profile for non-outliers, which is not applicable to our case because our training labels are very noisy. One possible solution is to learn one VAE for each category because the distribution within each category is relatively coherent and the negative effect of outliers can be mitigated to some extent, similar to the explanation in [46]. However, this solution is rather cumbersome especially when the number of categories is very large. As an alternative, we tend to inject category-level semantic information into the hidden layer and learn a semantic VAE. The motivation and details will be elaborated later in this section.

Suppose we have a noisy training set $\mathcal{I} = \{\mathbf{I}^i|_{i=1}^n\}$ with $\mathbf{I}^i$ being the i-th image and n being the number of training images, the visual feature of $\mathbf{I}^i$ (*i.e.*, output of CNN in Figure 1) is denoted as $\mathbf{x}^i$ and its associated label (*resp.*, predicted label variable) is denoted as c^i (*resp.*, y^i). Recall that in each training epoch, we only sample one latent variable for each training instance using reparameterization trick (see Section 3), so we use $\mathbf{z}^i = g_{\boldsymbol{\theta}_2}(\mathbf{x}^i, \boldsymbol{\epsilon}^1)$ to denote the deterministic latent variable of $\mathbf{x}^i$. Then, the objective function in (2) w.r.t. $\mathbf{x}^i$ can be simplified as

$$\text{KL}[q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}^i)||p_{\boldsymbol{\theta}_1}(\mathbf{z})] - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i). \qquad (3)$$

In order to incorporate category-level semantic information, we expect $\mathbf{z}^i$ to represent the semantic embedding of $\mathbf{x}^i$ so that the hidden layer of VAE has specific semantic meanings. By denoting $\mathcal{X}^{\tilde{c}} = \{\mathbf{x}^i|c^i = \tilde{c}\}$ and $\mathcal{Z}^{\tilde{c}} = \{\mathbf{z}^i|\mathbf{x}^i \in \mathcal{X}^{\tilde{c}}\}$, each $\mathcal{Z}^c$ corresponding to each category should be densely distributed. Instead of enforcing the distribution of $\mathcal{Z}^c$ to be close to certain hypothetical distribution, we tend to regulate the hidden layer indirectly, inspired by some recent ZSL approaches [1, 36, 24]. To be exact, in ZSL methods [1, 36, 24], with C categories in total, the classification score of $\mathbf{x}^i$ is calculated by using ${\mathbf{x}^i}^T\mathbf{VA}$, in which $\mathbf{V} \in \mathcal{R}^{d\times m}$ is mapping matrix with m (*resp.*, d) being the dimension of attribute vector (*resp.*, visual feature), and $\mathbf{A} \in \mathcal{R}^{m\times C}$ is category-attribute matrix with the c-th column $\mathbf{a}^c$ being the attribute vector of the c-th category. Analogous to ${\mathbf{x}^i}^T\mathbf{VA}$, we expect ${\mathbf{z}^i}^T\mathbf{A}$ to be consistent with the classification score of $\mathbf{x}^i$ by attaching a classifier on the hidden layer of VAE (see Figure 1). Note that $\mathbf{A}$ used in our experiments is hybrid semantic representation including attribute vector, which will be fully introduced in Section 6. Then, the predicted category probability of $\mathbf{z}^i$, *i.e.*, $p(y^i = c^i|\mathbf{z}^i)$, can be calculated by $\frac{\exp({\mathbf{z}^i}^T \mathbf{a}^{c^i})}{\sum_{\tilde{c}=1}^{C} \exp({\mathbf{z}^i}^T \mathbf{a}^{\tilde{c}})}$. In semantic VAE, as opposed to imposing non-discriminative generic prior $p_{\boldsymbol{\theta}_1}(\mathbf{z})$, we replace the regularizer $\mathrm{KL}[q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}^i)||p_{\boldsymbol{\theta}_1}(\mathbf{z})]$ in (3) with softmax loss $-\log p(y^i = c^i|\mathbf{z}^i)$ and obtain the following loss function:

$$l_{\boldsymbol{\theta}}(\mathbf{x}^i, c^i) = -\log p(y^i = c^i|\mathbf{z}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i), \tag{4}$$

in which $\boldsymbol{\theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ including the CNN parameters $\boldsymbol{\theta}_0$. The $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ in semantic VAE (4) is more reliable for indicating non-outliers than that in VAE (3), which can be explained as follows. According to the analysis in [46], when using gradient descent to minimize the reconstruction error, the learnt model is more capable of detecting outliers if the gradients of non-outliers are more consistent. Considering each $\mathcal{Z}^c$, due to the first regularizer in (4), the distribution of $\mathcal{Z}^c$ in semantic VAE should be more concentrated than that in VAE. Hence, when minimizing the reconstruction error $\sum_{\mathbf{z}^i \in \mathcal{Z}^c} -\log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ via gradient descent, the gradients of non-outliers in semantic VAE should be more consistent than those in VAE. As a result, by injecting discriminative information into the hidden layer, semantic VAE becomes better at outlier detection and the corresponding $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ is a more reliable indicator of $\mathbf{x}^i$ being a non-outlier.

In the next section, we will explore how to learn a robust classifier with the aid of semantic VAE and how the robust classifier can affect semantic VAE in return. In the rest part of this paper, "VAE" means semantic VAE by default and original VAE without semantic information is differentiated with the name "plain VAE" instead.

4.2. Learn Robust Classifier with Label Noise

In this section, we aim to learn a robust classifier by taking advantage of the reconstruction probability density from VAE. Recall that we attach a classifier on the hidden layer of VAE in Section 4.1. Instead of training a totally separate classification network, we tend to modify the attached classifier to account for label noise. Besides sharing model parameters, another benefit of doing so is a more effective VAE, which will be discussed later in this section.

To cope with the label noise, we introduce $\tilde{y}^i$ to denote the noisy label variable of $\mathbf{x}^i$, distinctive from its predicted label variable y^i. Besides, we further introduce a hidden variable h^i as binary non-outlier indicator, *i.e.*, $h^i = 1$ if $\mathbf{x}^i$ is a non-outlier and $h^i = 0$ otherwise. Strictly speaking, label noise of web data consists of outlier noise (image belongs to none of the training categories) and label flip noise (image belongs to one of the other training categories) [42]. Nevertheless, based on [50] as well as our own observation, outlier noise is far more dominant than label flip noise, so we simply treat all the noise as outlier noise without treating label flip noise separately in this paper. Now let us consider the conditional probability of noisy label $p_{\boldsymbol{\theta}}(\tilde{y}^i|\mathbf{x}^i, h^i)$. When $h^i = 0$, $\mathbf{x}^i$ does not belong to any training category and could be assigned with any category label randomly, so $p_{\boldsymbol{\theta}}(\tilde{y}^i|\mathbf{x}^i, h^i = 0) = \frac{1}{C}$. When $h^i = 1$, the associated label of $\mathbf{x}^i$ is accurate and we expect y^i predicted by our model to be aligned with $\tilde{y}^i$. Thus, $p_{\boldsymbol{\theta}}(\tilde{y}^i|\mathbf{x}^i, h^i)$ can be represented as

$$p_{\boldsymbol{\theta}}(\tilde{y}^i|\mathbf{x}^i, h^i) = \begin{cases} \frac{1}{C}, & \text{if } h^i = 0, \\ p_{\boldsymbol{\theta}}(y^i|\mathbf{x}^i), & \text{if } h^i = 1. \end{cases} \tag{5}$$

Then, we aim to cope with the label noise by maximizing $\log p(\tilde{y}^i = c^i|\mathbf{z}^i)$ instead of $\log p(y^i = c^i|\mathbf{z}^i)$, so that our assumption on noisy label in (5) can be taken into consideration. Specifically, we replace $\log p(y^i = c^i|\mathbf{z}^i)$ in (4) with $\log p(\tilde{y}^i = c^i|\mathbf{z}^i)$, leading to the new loss function:

$$l'_{\boldsymbol{\theta}}(\mathbf{x}^i, c^i) = -\log p(\tilde{y}^i = c^i|\mathbf{z}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i). \tag{6}$$

With deterministic latent variable $\mathbf{z}^i$ given $\mathbf{x}^i$, $p(\tilde{y}^i|\mathbf{z}^i)$ can be approximated by $p_{\boldsymbol{\theta}_2}(\tilde{y}^i|\mathbf{x}^i)$, so the loss function in (6) can be simplified as

$$\begin{aligned}
l'_{\boldsymbol{\theta}}(\mathbf{x}^i, c^i) &= -\log p_{\boldsymbol{\theta}_2}(\tilde{y}^i = c^i|\mathbf{x}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \\
&= -\log \sum_{h^i} p_{\boldsymbol{\theta}_2}(\tilde{y}_i = c^i, h^i|\mathbf{x}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \\
&= -\log \sum_{h^i} p_{\boldsymbol{\theta}_2}(\tilde{y}^i = c^i|\mathbf{x}^i, h^i) p(h^i|\mathbf{x}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \\
&\leq -\sum_{h^i} p(h^i|\mathbf{x}^i) \log p_{\boldsymbol{\theta}_2}(\tilde{y}^i = c^i|\mathbf{x}^i, h^i) \\
&\quad - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \qquad (7) \\
&= -p(h^i = 1|\mathbf{x}^i) \log p_{\boldsymbol{\theta}_2}(y^i = c^i|\mathbf{x}^i) - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \\
&\quad -(1 - p(h^i = 1|\mathbf{x}^i)) \log \frac{1}{C} \qquad (8) \\
&= -p(\mathbf{x}^i|h^i = 1) \frac{p(h^i = 1)}{p(\mathbf{x}^i)} \left(\log p_{\boldsymbol{\theta}_2}(y^i = c^i|\mathbf{x}^i) + \log C\right) \\
&\quad - \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) + \text{const}, \qquad (9)
\end{aligned}$$

in which (7) is based on Jensen's inequality and (8) is based on the definition in (5). Recall that $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ can be used as an indicator of $\mathbf{x}^i$ being a non-outlier as discussed in Section 4.1, in accordance with the meaning of $p(\mathbf{x}^i|h^i = 1)$. Thus, we approximate $p(\mathbf{x}^i|h^i = 1)$ in (9) with $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$. After omitting the constant, (9) can be rewritten as

$$\begin{aligned}
l'_{\boldsymbol{\theta}}(\mathbf{x}^i, c^i) \propto & -p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i) \left(\log p_{\boldsymbol{\theta}_2}(y^i = c^i|\mathbf{x}^i) + \log C\right) \\
& -\lambda_i \log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i), \qquad (10)
\end{aligned}$$

where $\lambda_i = \frac{p(\mathbf{x}^i)}{p(h^i = 1)}$. Since $p(\mathbf{x}^i)$ and $p(h^i = 1)$ cannot be directly inferred from our model, we simply treat all λ_i's as a common parameter λ. It is worth noting that from the perspective of classification loss, $-p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)\log p_{\boldsymbol{\theta}_2}(y^i = c^i|\mathbf{x}^i)$ in (10) is essentially weighted softmax loss, aiming to assign higher weights to the training instances which are less likely to be outliers.

Now let us switch to the perspective of VAE, the corresponding $\mathbf{z}^i$'s of identified non-outliers (*i.e.*, with larger $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$) are regulated more heavily than those of identified outliers. In this sense, the distribution of each $\mathcal{Z}^c$ is biased towards identified non-outliers from $\mathcal{X}^c$ and thus expected to be closer to the distribution of ground-truth $\bar{\mathcal{Z}}^c$, which is the semantic embedding space of ground-truth non-outliers from $\mathcal{X}^c$. With the distribution of $\mathcal{Z}^c$ closer to that of $\bar{\mathcal{Z}}^c$, the corresponding $\mathbf{z}^i$'s of ground-truth non-outliers from $\mathcal{X}^c$ should be more densely distributed, and thus their gradients when minimizing the reconstruction error should be more consistent. Similar to the discussion in Section 4.1, more consistent gradients of ground-truth non-outliers will contribute to better capability of VAE for outlier detection. Intuitively, by injecting relatively accurate discriminative information into the hidden layer, we can obtain a more effective VAE.

To this end, we aim to minimize the loss function in (10), which is the upper bound of (6), by training an end-to-end system as illustrated in Figure 1. However, we encounter two practical issues during training. The first issue is that for high-dimension multivariate Gaussian distribution, the value of probability density function is usually too large or too small, leading to numerical problems. Therefore, instead of directly computing $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$, we first compute $\log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$, then subtract the maximum value for each training batch, and finally calculate the exponential to recover $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$. Then we can obtain normalized $p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ in the range of $[0,1]$, denoted as $\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$. The above trick is equivalent to multiplying the first term in (10) with a constant for each training batch, which can be approximately absorbed into the trade-off parameter λ, leading to the new parameter $\tilde{\lambda}$. The second issue is that $\log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$'s suffer from so high variance that in each training batch, only one $\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ is 1 while all the others are close to 0. To circumvent this problem, we fix $\boldsymbol{\sigma}_x$ as $\mathbf{1}$ so that $p_{\boldsymbol{\theta}_1}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_x, \mathbf{I})$ to make $\log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$'s more stable. Finally, we arrive at our optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} & -\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)\left(\log p_{\boldsymbol{\theta}_2}(y^i = c^i|\mathbf{x}^i) + \log C\right) \\ & -\tilde{\lambda}\log p_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i), \end{aligned} \tag{11}$$

in which n is the number of training instances.

In the testing stage, given a test instance $\mathbf{x}^i$, we use $\mathbb{E}_{q_{\boldsymbol{\theta}_2}(\mathbf{z}|\mathbf{x}^i)} p(y^i|\mathbf{z}) \approx \frac{1}{L}\sum_{l=1}^{L} p(y^i|\mathbf{z}^{i,l})$ with $\mathbf{z}^{i,l} = g_{\boldsymbol{\theta}_2}(\mathbf{x}^i, \boldsymbol{\epsilon}^l)$ (see Section 3) for prediction. Specifically, we pass each test instance through the classification network for 5 times (*i.e.*, $L = 5$) and average the predicted category probabilities as its final category probability.

5. Extension for Domain Adaptation

In order to reduce the domain shift between source domain (*i.e.*, web data) and target domain (*i.e.*, test data), we extend our WSCI method to WSCI-DA by reconstructing the target domain instances using the same variational autoencoder (VAE) as for the source domain instances, and the reason can be explained as follows. Unlike low-level visual features, intermediate-level semantic embeddings are more insusceptible to hidden factors (*e.g.*, pose and illumination) and thus more domain invariant. Therefore, the data distribution difference between source domain and target domain can be disentangled in the hidden layer by reconstructing the data from both domains [5, 15]. By denoting the objective function in (11) as $\sum_{i=1}^{n} F(\mathbf{x}^i, y^i; \boldsymbol{\theta})$, our WSCI-DA method can be formulated as

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} F(\mathbf{x}^i, y^i; \boldsymbol{\theta}) + \sum_{i=1}^{\hat{n}} \left(-\tilde{\lambda}\log p_{\boldsymbol{\theta}_1}(\hat{\mathbf{x}}^i|\hat{\mathbf{z}}^i)\right), \tag{12}$$

where $\{\hat{\mathbf{x}}^i|_{i=1}^{\hat{n}}\}$ is the test set and $\hat{\mathbf{z}}^i$ is the deterministic latent variable of $\hat{\mathbf{x}}^i$.

Given that our WSCI-DA method in (12) does not consider the relation of semantic embeddings between source domain instances and target domain instances, we additionally propose a low-rank refinement strategy following our WSCI-DA method. In particular, after the training process based on (12), we can obtain the semantic embeddings of source domain instances (*i.e.*, $\hat{\mathbf{z}}^i$'s) and target domain instances (*i.e.*, $\mathbf{z}^i$'s) based on the trained model. Then, we assume that the semantic embeddings of target domain instances can be linearly reconstructed based on those of source domain instances, and the reconstruction matrix should be low-rank because the reconstruction coefficients corresponding to the target domain instances from the same category should be grouped together. Formally, with the semantic embedding of the i-th target (*resp.*, source) domain instance being $\hat{\mathbf{z}}^i \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_z^i, \text{diag}(\hat{\boldsymbol{\sigma}}_z^{i^2}))$ (*resp.*, $\mathbf{z}^i \sim \mathcal{N}(\boldsymbol{\mu}_z^i, \text{diag}(\boldsymbol{\sigma}_z^{i^2}))$), we assume $\hat{\mathbf{z}}^i$ can be linearly reconstructed based on $\mathbf{z}^i$'s, *i.e.*, $\hat{\mathbf{z}}^i = \sum_{j=1}^{n} s_{ji}\mathbf{z}^j$. Then, by simply assuming $\mathbf{z}^i$'s are mutually independent, it can be easily known that $\hat{\boldsymbol{\mu}}_z^i = \sum_{j=1}^{n} s_{ji}\boldsymbol{\mu}_z^j$ and $\hat{\boldsymbol{\sigma}}_z^{i^2} = \sum_{j=1}^{n} s_{ji}^2\boldsymbol{\sigma}_z^{j^2}$. Let us define the reconstruction matrix $\mathbf{S}$ with each entry being s_{ji}, $\mathbf{Z}^\mu$ (*resp.*, $\hat{\mathbf{Z}}^\mu$) with the i-th column being $\boldsymbol{\mu}_z^i$ (*resp.*, $\hat{\boldsymbol{\mu}}_z^i$), and $\mathbf{Z}^\sigma$ (*resp.*, $\hat{\mathbf{Z}}^\sigma$) with the i-th column being $\boldsymbol{\sigma}_z^{i^2}$ (*resp.*, $\hat{\boldsymbol{\sigma}}_z^{i^2}$). Then, we assume $\mathbf{S}$ to be low-rank to capture the intrinsic relatedness of target domain instances, considering that the columns in $\mathbf{S}$ corresponding to the target domain instances within the same category should be similar with each other. In analogy to low-rank representation (LRR) [20], we can reach the following optimization problem:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{E}^{\mu},\mathbf{E}^{\sigma}} \quad & \|\mathbf{S}\|_* + \|\mathbf{E}\|_{2,1}, \\ \text{s.t.} \quad & \hat{\mathbf{Z}}^{\mu} = \mathbf{Z}^{\mu}\mathbf{S} + \mathbf{E}^{\mu}, \\ & \hat{\mathbf{Z}}^{\sigma} = \mathbf{Z}^{\sigma}(\mathbf{S} \circ \mathbf{S}) + \mathbf{E}^{\sigma}, \end{aligned} \tag{13}$$

where $\|\mathbf{S}\|_*$ is the nuclear norm [33] (the convex approximation of rank function) enforcing $\mathbf{S}$ to be low-rank, and $\|\mathbf{E}\|_{2,1}$ is $L_{2,1}$ norm [55] of reconstruction error matrix $\mathbf{E} = [\mathbf{E}^{\mu}; \mathbf{E}^{\sigma}]$, which enforces $\mathbf{E}$ to be column-sparse to tolerate larger reconstruction errors for "noisy" target domain instances. Note that we employ $L_{2,1}$ norm on vertically stacked $\mathbf{E}^{\mu}$ and $\mathbf{E}^{\sigma}$ to ensure the consistency of sparse columns in $\mathbf{E}^{\mu}$ and $\mathbf{E}^{\sigma}$. The optimization problem in (13) can be solved using inexact Augmented Lagrange Multiplier (ALM) [4].

After obtaining $\mathbf{S}$ by solving (13), we update $\hat{\mathbf{Z}}^{\mu}$ (*resp.*, $\hat{\mathbf{Z}}^{\sigma}$) by $\hat{\mathbf{Z}}^{\mu} = \mathbf{Z}^{\mu}\mathbf{S}$ (*resp.*, $\hat{\mathbf{Z}}^{\sigma} = \mathbf{Z}^{\sigma}(\mathbf{S} \circ \mathbf{S})$), and generate $\hat{\mathbf{z}}^i$ based on updated $\{\hat{\boldsymbol{\mu}}_z^i, \hat{\boldsymbol{\sigma}}_z^i\}$ with reparameterization trick, followed by the same prediction strategy as in Section 4.

6. Category-level Semantic Representation

In this section, we introduce the category-level semantic representation used in our experiments (matrix $\mathbf{A}$ in Section 4), which consists of three types of information. Two common types of category-level semantic information are attribute and word vector. To overcome the drawbacks of attribute (*e.g.*, not free) and word vector (*e.g.*, free yet not visually grounded), we propose a third type of category-level semantic information called visual encoding, which is both free and visually grounded.

Attribute: Attribute representation [18, 9] for each category is a high-level description in the form of semantic cues, such as the shape (*e.g.*, cylindrical), material (*e.g.*, cloth), and color (*e.g.*, white). Such attribute representations are acquired based on expertise from human experts and thus not freely available.

Word Vector: Each word can be represented by a real-valued vector (*e.g.*, Word2Vec [23] and GloVe [32]) using the linguistic model trained on free online corpus (*e.g.*, Wikipedia). Recently, word vectors have been used for image classification in zero-shot learning (ZSL) by using the word vector of category name as the intermediate semantic representation of each category [1, 10, 39, 54]. However, word vector focus on linguistic regularities and patterns, and hence may lack visual grounding [16].

Visual Encoding: We propose a free and visually grounded encoding method, which encodes each category as visual bag-of-word representation by using Gaussian Mixture Model (GMM) as the visual codebook. Specifically, we first generate region proposals for each training image using existing method and then train a K-component GMM based on sampled region proposals. We define the probability that the i-th region proposal belongs to the j-th Gaussian model as $\gamma_i(j)$. With $\boldsymbol{\gamma}_i = [\gamma_i(1), \ldots, \gamma_i(K)]$, we calculate the average of $\boldsymbol{\gamma}_i$'s of all the region proposals from the images belonging to the c-th category as the encoding vector of the c-th category, which is denoted as $\bar{\boldsymbol{\gamma}}_c$. However, there are two issues with the obtained visual encoding vector $\bar{\boldsymbol{\gamma}}_c$: 1) region proposals from one category may be very noisy due to inaccurate image labels; 2) some components in GMM (*i.e.*, visual words in the codebook) are category-invariant and commonly shared by all categories (*e.g.*, visual words that fall in the background), which renders the visual encoding vector less discriminative. For the first issue, we reset the entries with small values. For the second issue, we learn a project matrix to maximize the category separation and reduce the dimension of $\bar{\boldsymbol{\gamma}}_c$ to $\tilde{K}$.

In practice, we can concatenate available types of semantic representations of each category as the hybrid semantic representation of that category. More experimental details will be introduced in Section 7.

7. Experiments

In this section, we evaluate our WSCI method for image classification on three benchmark datasets and also provide some showcases for identified outliers/non-outliers.

Datasets: Since attribute vector is included as part of our hybrid semantic information, we conduct experiments on three popular benchmark datasets: AwA2, CUB, and SUN Attribute, which are associated with attribute vector for each category. For each benchmark dataset, we use the entire dataset as test set while crawling 500 images from Google image website for each category as training set.

1) AwA2 [47]: Animals with Attributes 2 (AwA2) dataset releases more images than its previous version AwA. Specifically, AwA2 consists of 37322 images of 50 animal categories. The provided category-attribute matrix contains 85 numeric attribute values for all 50 categories.

2) CUB [44]: Caltech-UCSD Bird (CUB) has in total $11,788$ images distributed in 200 bird species. The CUB dataset contains a 312-dim binary human specified attribute vector for each image, so we average the attribute vectors of the images within each category and use the averaged attribute vector for that category.

3) SUN Attribute [48]: Scene UNderstanding (SUN) attribute dataset has 717 scene categories with 20 images in each category. Similar to CUB, we calculate the averaged 102-dim attribute vector for each category.

4) Google image dataset: We construct the web training set by ourselves. Particularly, for each benchmark test set (*i.e.*, AwA2, CUB, and SUN), we use the category names as queries to collect the top ranked 500 images from Google image website for each category after performing PCA-based near-duplicate removal [56].

Category-level Semantic Representation: As discussed in Section 6, we employ three types of category-level infor-

Table 1: Accuracies (%) of different methods on three datasets. The best results are highlighted in boldface.

Dataset	AwA2	CUB	SUN	Avg
CNN	84.02	72.24	35.91	64.06
bootstrap [34]	85.71	73.63	37.36	65.57
Chen and Gupta [6]	85.53	74.92	38.33	66.26
Sukhbaatar *et al.* [42]	86.11	73.51	38.61	66.08
Xiao *et al.* [49]	86.41	75.02	40.56	67.33
RGT+AT+R [57]	86.52	73.75	38.03	66.10
WSCI_sim1	86.15	74.17	37.91	66.08
WSCI_sim2	88.88	76.28	41.23	68.80
WSCI (w/o ve)	90.52	76.86	41.97	69.78
WSCI	**91.14**	**77.34**	**42.26**	**70.25**

mation: attribute, word vector, and visual encoding. In the following, we provide the details of extracting these three types of information: (1) for attribute, we use the 85-dim (*resp.*, 312-dim and 102-dim) continuous attribute vector associated with the AwA2 (*resp.*, CUB and SUN) dataset as mentioned above; (2) for word vector, we train GloVe [32] language model based on the latest Wikipedia corpus, with the dimension of word vector set as 500. Then, we can obtain the word vector corresponding to each category name. For the category names with more than one word, we average the word vectors corresponding to all the words appearing in the category name as the final word vector of that category; (3) for visual encoding, we set K (*resp.*, $\tilde{K}$) as 256 (*resp.*, 128). At last, we concatenate three types of information as hybrid semantic representation, leading to a 713-dim (*resp.*, 940-dim and 730-dim) vector for each category on the AwA2 (*resp.*, CUB and SUN) dataset.

Network Architecture: Our network consists of a CNN model, a VAE model, and a softmax classification model, as shown in Figure 1. For the CNN model, we adopt Inception-V3 [43], which outputs 2048-dim visual feature. For the VAE model, we implement both encoder and decoder as multiple layer perceptron (MLP) with one hidden layer. The dimension of the hidden layer in MLP is set as 1500, which is approximately $\frac{d+m}{2}$ with d being the dimension of Inception-V3 output (*i.e.*, 2048) and m being the dimension of category-level semantic representation. The entire network is implemented using TensorFlow, based on which we use Adam optimizer with batch size being 64 and exponentially decaying learning rate initialized as 0.001.

Parameter: The objective function in (11) has one hyperparameter $\tilde{\lambda}$, which is empirically set as 10^{-4} for all datasets. In our experiments, we observe that our methods are relatively robust when setting $\tilde{\lambda}$ in certain range (*e.g.*, [10^{-6},10^{-4}]).

Baselines: We compare with three sets of baselines: basic CNN, Webly Supervised Learning (WSL) methods, and simplified versions of our method.

1) For basic CNN, we train Inception-V3 without considering label noise, which is referred to as CNN in Table 1.

2) For WSL methods, we compare with the following recent deep learning approaches: Chen and Gupta [6], Sukhbaatar *et al.* [42], Xiao *et al.* [49], bootstrap [34], and RGT+AT+R [57]. For Chen and Gupta [6] and Xiao *et al.* [49], since we do not have clean images to estimate the confusion matrix, we calculate the category similarities based on semantic representations as the confusion matrix. In this way, category-level information is utilized in [6, 49]. For Xiao *et al.* [49] and RGT+AT+R [57], they use partial clean data when training the network, which is not available under our setting, so we only utilize noisy web data to train their models. For all the deep learning baselines mentioned above, we use Inception-V3 as the basic network structure for fair comparison.

3) For simplified versions of our WSCI method, we first split the flowchart in Figure 1 into two separate flows with joint loss at the end, in which the top flow is classification network based on $\mathbf{x}$ and the bottom flow is a plain VAE detached from the classifier. We use the same loss function as in (11) and refer to this simplified version as WSCI_sim1 in Table 1. Based on WSCI_sim1, we replace VAE with semantic VAE by attaching a classifier on the hidden layer, which can utilize category-level information $\mathbf{A}$. Note that the attached classifier has an additional standard classification loss, which is independent on the classification network in the top flow. This simplified version is referred to as WSCI_sim2. Another simplified version is nearly the same as full-fledged WSCI except that we exclude our proposed visual encoding from hybrid semantic representation, which is referred to as WSCI (w/o ve).

Experimental Results: The experimental results are summarized in Table 1, in which the results on the SUN dataset are much worse than those on the other two datasets. This is because the SUN dataset has far more categories and the web images of scene categories are more noisy than those of animal categories. We observe that the baselines [34, 6, 42, 49, 57] achieve better results than basic CNN, because they cope with label noise using different techniques. We also observe that WSCI_sim2 outperforms WSCI_sim1, which shows it is helpful to utilize category-level information. Moreover, WSCI is better than WSCI_sim2, indicating the advantage for classification network and VAE to jointly leverage category-level information. Another observation is that WSCI is slightly better than WSCI (w/o ve), which validates the effectiveness of our proposed visual encoding vector as part of hybrid semantic representation. We also observe that in the absence of manually annotated attribute vector, our method WSCI (w/o ve) still outperforms all the baselines, which proves the superiority of our method even only using free category-level semantic information. Finally, our WSCI method achieves the best results on all three datasets. This again

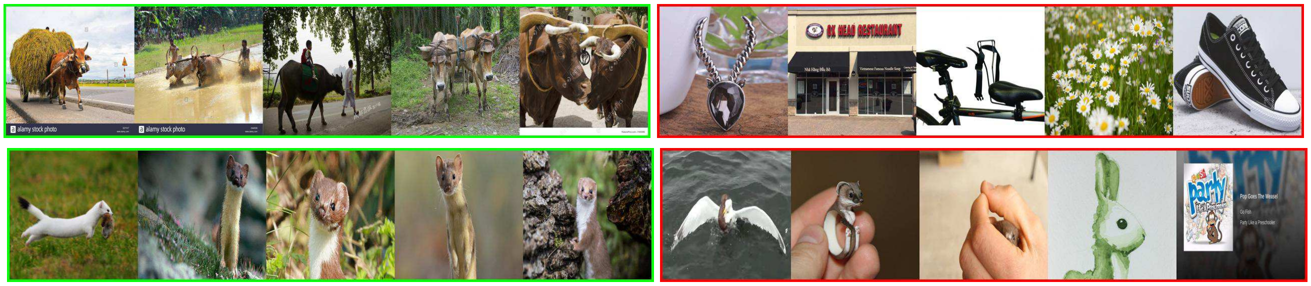

Figure 2: The first (*resp.*, second) row contains the web training images from the category "ox" (*resp.*, "weasel"). The green (*resp.*, red) boxes on the left (*resp.*, right) column group the top 5 images with highest (*resp.*, lowest) $\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$, which indicates the identified non-outliers (*resp.*, outliers).

demonstrates the superiority of our VAE based method, in which VAE and classification network can jointly utilize category-level supervision.

Qualitative Analysis: Recall that we use the reconstruction probability density $\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ in (11) as the indicator of $\mathbf{x}^i$ being a non-outlier. In order to qualitatively verify the capability of our semantic VAE to detect outliers, we take the categories "ox" and "weasel" from the dataset AwA2 as examples (see the top and bottom row in Figure 2). In particular, we show the top 5 web training images from each category with highest (*resp.*, lowest) $\tilde{p}_{\boldsymbol{\theta}_1}(\mathbf{x}^i|\mathbf{z}^i)$ in the green (*resp.*, red) boxes. From Figure 2, it can be seen that the top outliers and non-outliers are successfully identified, contributing to a more robust classifier. We have similar observations on the other categories from the other datasets.

Extension for Domain Adaptation: We additionally conduct experiments under the domain adaptation setting, in which unlabeled test instances are used in the training stage. We compare with two sets of baselines: Domain Adaptation (DA) methods and Webly Supervised Learning (WSL) methods which can address the domain issue. For DA methods, we compare with several closely related DA methods based on autoencoder/VAE or low-rank techniques: mSDA [5], BAE [12], VFAE [21], RDALR [11], and LTSL [38]. For DA-related WSL methods, we compare with Bergamo and Torresani [3] and WSDG [27], which can cope with the label noise and address the domain shift at the same time. Note that the methods in [11, 38, 5, 12, 3, 27] are all feature-based approaches. For fair comparison, we extract visual features from Inception-v3 retrained on our training sets, in which case feature-based methods can achieve at least comparable results with CNN in Table 1. For our WSCI-DA method, we report both results with or without low-rank refinement.

The experimental results are reported in Table 2, from which we observe that the DA baselines and DA-related WSL baselines achieve better results than CNN, which indicates the necessity of addressing the domain shift between web data and test data. We also observe that WSCI-DA is better than WSCI, which shows it is beneficial to reconstruct unlabeled test instances using VAE. Besides, WSCI-DA (refinement) further improves WSCI-DA, which demonstrates the effectiveness of our low-rank refinement. Finally, our WSCI-DA (refinement) method achieves superior performance compared with all the baselines on all three datasets, which indicates the advantage of our probabilistic framework to handle label noise and domain issue with the aid of category-level supervision.

Table 2: Accuracies (%) of different methods with domain adaptation on three datasets. The best results are highlighted in boldface.

Dataset	AwA2	CUB	SUN	Avg
CNN	84.02	72.24	33.76	63.34
RDALR [11]	85.89	73.08	34.63	64.53
LTSL [38]	86.63	74.39	35.13	65.38
mSDA [5]	84.63	72.79	34.32	63.91
BAE [12]	84.87	73.80	36.55	65.07
VFAE [21]	86.91	75.15	36.56	66.20
Bergamo *et al.* [3]	87.99	75.96	37.33	67.09
WSDG [27]	86.56	75.10	36.71	66.12
WSCI	91.14	77.34	39.59	69.36
WSCI-DA	93.17	78.70	41.39	71.08
WSCI-DA (refinement)	**94.28**	**80.83**	**42.70**	**72.60**

8. Conclusion

In this paper, we have studied addressing the label noise and domain shift by using category-level supervision when learning from web data. Extensive experiments on three benchmark datasets have demonstrated the effectiveness of our proposed methods.

Acknowledgement

This work is supported by NIH 7000000356 and NSF IIS-1652633. This work is also supported by the NGA NHARP program HM0476-15-1-0007.

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015. 1, 3, 6

[2] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. Technical report, Technical Report, 2015. 2, 3

[3] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *NIPS*, 2010. 1, 2, 8

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 6

[5] M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *ICML*, 2012. 2, 5, 8

[6] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015. 1, 2, 7

[7] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013. 2

[8] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 2

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 6

[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. DeViSE: A deep visual-semantic embedding model. In *NIPS*, 2013. 1, 6

[11] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *CVPR*, 2012. 2, 8

[12] M. Kan, S. Shan, and X. Chen. Bi-shifting auto-encoder for unsupervised domain adaptation. In *ICCV*, 2015. 2, 8

[13] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, 2014. 2

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2014. 1, 2, 3

[15] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *CVPR*, 2017. 5

[16] S. Kottur, R. Vedantam, J. M. F. Moura, and D. Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *CVPR*, 2016. 6

[17] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*, 2016. 1, 2

[18] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *T-PAMI*, 36(3):453–465, 2014. 1, 6

[19] W. Li, L. Niu, and D. Xu. Exploiting privileged information from web data for image categorization. In *ECCV*, 2014. 2

[20] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *T-PAMI*, 35(1):171–184, 2013. 5

[21] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015. 2, 8

[22] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 2

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 6

[24] L. Niu, J. Cai, and A. Veeraraghavan. Zero-shot learning via category-specific visual-semantic mapping. *arXiv preprint arXiv:1711.06167*, 2017. 3

[25] L. Niu, J. Cai, and D. Xu. Domain adaptive fisher vector for visual recognition. In *ECCV*, 2016. 2

[26] L. Niu, W. Li, and D. Xu. Multi-view domain generalization for visual recognition. In *ICCV*, 2015. 2

[27] L. Niu, W. Li, and D. Xu. Visual recognition by learning from web data: A weakly supervised domain generalization approach. In *CVPR*, 2015. 2, 8

[28] L. Niu, W. Li, and D. Xu. Exploiting privileged information from web data for action nd event recognition. *IJCV*, 118(2):130–150, 2016. 2

[29] L. Niu, W. Li, D. Xu, and J. Cai. An exemplar-based multi-view domain generalization framework for visual recognition. *T-NNLS*, 2016. 2

[30] L. Niu, W. Li, D. Xu, and J. Cai. Visual recognition by learning from web data via weakly supervised domain generalization. *T-NNLS*, 28(9):1985–1999, 2017. 2

[31] L. Niu, X. Xu, L. Chen, L. Duan, and D. Xu. Action and event recognition in videos by learning from heterogeneous web sources. *T-NNLS*, 28(6):1290–1304, 2017. 2

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 6, 7

[33] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010. 6

[34] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *ICLR*, 2015. 1, 2, 7

[35] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *ICML*, 2014. 1, 2

[36] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 3

[37] M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *MLSDA*, 2014. 1

[38] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *IJCV*, 2014. 2, 8

[39] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 1, 6

[40] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *NIPS*, 2015. 2

[41] S. Suh, D. H. Chae, H.-G. Kang, and S. Choi. Echo-state conditional variational autoencoder for anomaly detection. In *IJCNN*, 2016. 2, 3

[42] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. *ICLR*, 2015. 1, 2, 4, 7

[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 7

[44] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 6

[45] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 2

[46] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *ICCV*, 2015. 1, 3, 4

[47] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017. 6

[48] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[49] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 1, 2, 7

[50] Z. Xu, S. Huang, Y. Zhang, and D. Tao. Webly-supervised fine-grained visual categorization via deep domain adaptation. *T-PAMI*, 2016. 1, 2, 4

[51] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*, 2014. 2

[52] Z. Xu, L. Zhu, and Y. Yang. Few-shot object recognition from machine-labeled web images. *CVPR*, 2017. 1, 2

[53] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016. 2

[54] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013. 1, 6

[55] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 6

[56] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016. 6

[57] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. Reid. Attend in groups: a weakly-supervised deep learning framework for learning from web data. *CVPR*, 2017. 1, 2, 7