Non-generative Generalized Zero-shot Learning via Task-correlated Disentanglement and Controllable Samples Synthesis

Yaogong Feng, Xiaowen Huang, Pengbo Yang, Jian Yu, Jitao Sang
School of Computer and Information Technology & Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China
{fengyg18, pengboyang, xwhuang, jianyu, jtsang}@bjtu.edu.cn

Abstract

Synthesizing pseudo samples is currently the most effective way to solve the Generalized Zero Shot Learning (GZSL) problem. Most models achieve competitive performance but still suffer from two problems: (1) Feature confounding, the overall representations confound task-correlated and task-independent features, and existing models disentangle them in a generative way, but they are unreasonable to synthesize reliable pseudo samples with limited samples; (2) Distribution uncertainty, that massive data is needed when existing models synthesize samples from the uncertain distribution, which causes poor performance in limited samples of seen classes. In this paper, we propose a non-generative model to address these problems correspondingly in two modules: (1) Task-correlated feature disentanglement, to exclude the task-correlated features from task-independent ones by adversarial learning of domain adaption towards reasonable synthesis; (2) Controllable pseudo sample synthesis, to synthesize edge-pseudo and center-pseudo samples with certain characteristics towards more diversity generated and intuitive transfer. In addition, to describe the new scene that is the limit seen class samples in the training process, we further formulate a new ZSL task named the ‘Few-shot Seen class and Zero-shot Unseen class learning’ (FSZU). Extensive experiments on four benchmarks verify that the proposed method is competitive in the GZSL and the FSZU tasks.

1. Introduction

The explosion of data and the rapid development of deep learning need massive precise but expensive labels. In the real world, these labels are usually sparse/missing. Zero Shot Learning (ZSL) techniques offer a good solution to address such problem, which trains on seen classes and tests on unseen classes (the seen classes and unseen classes are independent). In this paper, we focus on the Generalized ZSL (GZSL) task. It is a realistic setting of ZSL in making predictions on recognizing samples from both classes simultaneously rather than classifying only samples of the unseen classes.

At present, the method of synthesizing pseudo samples for unseen classes has proven to be one of the most effective ways of knowledge transfer to solve the GZSL problem. But there are still two challenging problems: (1) Feature Confounding. Most GZSL models are based on overall representations of samples extracted in pre-trained CNN (e.g., ResNet101 [13]) while the semantic features are class-level attributes or class-level sentence embeddings [21, 40]. The former contains more rich information and inconsistent with human cognition. So it is unreasonable to construct a mapping from visual features to semantic features directly and synthesize reliable pseudo samples based on the confounding visual features. Although some models [7, 23] contribute to extracting more human-consistent-cognition features, they are related to generative models and hard to
guarantee the diversity of sample synthesis with limited real samples.

(2) **Distribution Uncertainty.** The existed methods, especially the generative models, usually require a large amount of data to fit the distribution of real data, and can only synthesize pseudo samples with the uncertain distribution. So these models have poor performances when samples of seen class are few shot.

Figure 1 illustrates the motivation of our method. We first exclude the task-independent feature from horse image to horse object. Then we synthesize zebra’s pseudo samples of target class based on the horse’s task-correlated feature of source class in a non-generative way.

To be specific, we propose a non-generative approach named Task-correlated Disentanglement and Controllable Samples Synthesis (TDCSS) method that handles above issues. The TDCSS mainly consists of two components. (1) **Task-correlated Feature Disentanglement Module.** Our model is based on class semantic features to complete the image classification task. According to whether the visual features are corresponding to class semantics, we disentangle the confounding features into task-correlated features and task-independent features. The task-correlated features are more consistent with class semantics. We introduce the adversarial training of domain adaptation to achieve feature disentanglement. (2) **Controllable Pseudo Samples Synthesis Module.** Based on task-correlated features, we add different offsets to synthesize two types of pseudo samples which are edge-pseudo samples and center-pseudo samples in a non-generative way. For the edge-pseudo samples, we treat them as the adversarial examples in the feature-level and the edge offsets can be seen as perturbations in adversarial examples. For the center-pseudo samples, we make them distributed closer to the center of one class. Both of them guarantee the generative diversity of samples based on the limited seen samples. In addition, synthesizing pseudo samples with certain characteristics contribute to exploring the role of different types of pseudo samples in the knowledge transfer of GZSL.

To describe the scene that only has the limited samples on seen class in formulation, we further propose a new ZSL task named ‘Few-shot Seen class and Zero-shot Unseen class learning (FSZU)’. The FSZU is also more reasonable and more practical. In the GZSL task, all classes have strong semantic relationships. So we believe that the ZSL and the Few Shot Learning (FSL) are coexistent. For example, in the deep-space exploration and deep-sea exploration, the machine (detector) always encounter new situations, and the number of seen class samples that humans have obtained is also extremely limited. In this paper, we perform the TDCSS and the similar methods on the new task.

In summary, our main contributions of this work are summarized as follows:

(1) We propose a novel non-generative model that disentangles visual features into task-correlated and task-independent by adversarial training of domain adaptation. And we use the task-correlated features to synthesize two types of pseudo samples as center-pseudo samples and edge-pseudo samples to guarantee the diversity of sample synthesis and the intuitive transfer.

(2) We propose a new zero shot task named ‘Few-shot Seen class and Zero-shot Unseen class learning (FSZU)’, which is more reasonable and more practical compared with the GZSL task in real life.

(3) Extensive experiments in the GZSL task and the FSZU task on four widely used datasets verify the results of the TDCSS are competitive with similar methods.

2. Related Works

2.1. Generalized Zero-Shot Learning

Domain shift is a basic problem in GZSL. It is described as ‘due to having disjoint and potentially unrelated classes, the projection functions learned from the auxiliary dataset/domain are biased when applied directly to the target dataset/domain’ [9].

**GZSL with pseudo sample synthesis.** Many researchers have proposed generative models to synthesize pseudo samples for unseen classes to alleviate this problem. The GAN-based models contribute to increasing the diversity of pseudo samples [22] and preserving semantic consistency [24,30]. The VAE-based models [6,19,34] contribute to preserving semantic consistency of different representations of distribution in hidden layer. Some researchers integrate the VAE and GAN into a unified conditional feature generating model [29,42] to integrate the advantages. There are also some non-generative models [8,11,12,15,26,27] synthesizing pseudo samples. The models [15,27] extract attribute-based features and then combine them to synthesize unseen pseudo samples. The BPL [11] synthesizes pseudo samples based on bidirectional projection learning and linear interpolation. The AGZSL [8] uses the Image Adaptive Semantics to expand the semantic features by using visual features, then it trains the seen class classifier based on the expanded semantic features and trains the unseen class classifier by synthesizing virtual class by sampling and interpolating over seen counterparts. The models [8,11], synthesizing pseudo samples by perturbing or interpolating, are simpler and more likely to transfer class variations from seen class to unseen classes. So we use the non-generative models to synthesize pseudo samples.

**GZSL with representation disentanglement.** Most of GZSL models are based on the overall representations. Researchers use the representation disentanglement to obtain more consistent visual features with semantics. Some re-
searchers disentangle visual features into ‘object + attribute’ features \([2, 28]\) by exploring their respective distributions. For example, the ‘red wine’ can be disentangled into ‘red (attribute)’ + ‘wine (object)’. But these methods require more strictly labeled datasets. Some researchers try to align attribute-based features with their attribute semantic vectors in fine-grained ZSL task \([15, 16]\). But their methods are based on the feature map at the last convolutional layer. Some researchers disentangle visual features according to their understanding of the GZSL. The DLFFZRL \([38]\) disentangles the feature into the semantic latent feature, the non-semantic latent feature, and the non-discriminative latent feature, in which the first two factors are learned by adversarial learning and the last is learned by a hierarchical structure. In addition, SP-AEN \([5]\) disentangles the semantic space into two subspaces for classification and reconstruction respectively. Some researchers also use generative models with random permuting to achieve representation disentanglement \([7, 23]\). In this paper, we disentangle visual features into task-correlated features and task-independent features by using domain adversarial, which is more robust in different tasks.

2.2. Adversarial Example and Adversarial Self-Supervised Learning

Recently, extensive experiments \([1, 33, 35, 39]\) have shown that model would have better generalization which has better adversarial robustness. And it achieves higher performance than the naturally trained models in ZSL \([1]\). These works usually use the gradient-based adversarial example generation algorithm, such as FSGM \([10]\) and PGD \([18]\). Some self-supervised learning methods \([14, 20]\) are also upgraded to adversarial self-supervised learning based on the adversarial examples in the way of contrastive learning, which extracts image features that are more consistent with human cognition.

In this paper, we draw on that core idea and further propose the edge-pseudo samples that can be seen as feature-level adversarial examples based on the targeted attack. We also introduce a training mechanism of the adversarial self-supervised to TDSCS to make our model extract more consistent task-correlated features with class semantics.

3. The Proposed Method

3.1. Problem Formulation

In the GZSL task, let \(\{X^s, Y^s\}\) be the dataset with \(S\) seen classes, which contains \(N_s^s\) training samples \(X^s = \{x_{(i)}^s\}_{i=1}^{N_s^s}\) and the corresponding class labels \(Y^s = \{y_{(i)}^s\}_{i=1}^{N_s^s}\). The class labels span from 1 to \(S\), \(y^s \in L^s = \{1, \ldots, S\}\). In the FSZU task, the \(|S|\) is same with GZSL. But the samples on every seen class are much less than GZSL, which contains \(X^s = \{x_{(i)}^s\}_{i=1}^{N_f^s}\) and \(Y^s = \{y_{(i)}^s\}_{i=1}^{N_f^s}\), where \(N_f^s \ll N_s^s\). For the test set which involves unseen classes, the GZSL is the same as the FSZU. Specifically, given another dataset \(\{X^u, Y^u\}\), on which the classes are related to the seen dataset. The dataset has \(U\) unseen classes and consist of \(N_u\) data instances \(X^u = \{x_{(i)}^u\}_{i=1}^{N_u}\) with corresponding labels \(Y^u = \{y_{(i)}^u\}_{i=1}^{N_u}\). The class labels thus range from \(S + 1\) to \(S + U\), \(y^u \in L^u = \{S + 1, \ldots, S + U\}\). The \(L^s \cap L^u = \emptyset\). Each class is associated with a class-level semantic feature, which can be embedding and attribute. And the semantic information can be represented as \(\mathcal{A} = \{a(k)\}_{k=1}^{S+U}\). We denote \(A^s\) and \(A^u\) as the semantic features of seen and unseen classes. In this paper, the model training is achieved in two stages. We split the seen classes into source classes (\(\{X^{ss}, Y^{ss}\}\), \(A^{ss}\)) and target classes (\(\{X^{st}, Y^{st}\}, A^{st}\)) in the first training stage. And we regard the seen classes as source classes and unseen classes as target classes in the second training stage.

3.2. Overall Framework

In this section, we present the details and the training strategy of TDSCS. The overall framework is illustrated in Figure 2. There are two key components: (1) Task-correlated feature disentanglement. We first disentangle the confounding visual features \(X^s\) of source classes to task-correlated features \(h^{cor}\) and task-independent features \(h^{ind}\) by adversarial training of domain adaptation and making sure that both of them are precise, meaningful and independent to each other. The disentangled task-correlated features \(h^{cor}\) are then regarded as more reasonable representations for the sample synthesis. (2) Controllable pseudo samples synthesis. We use the task-correlated features \(h^{cor}\) of source classes to add the Center Offset \(\sigma^{center}\) and the Edge Offsets \(\sigma^{edge}\) respectively, which can synthesize two types of pseudo samples that are center-pseudo samples \(h^{center}\) and edge-pseudo samples \(h^{edge}\). The offsets are outputted by convert networks.

3.3. Task-correlated Feature Disentanglement

This module consists of adversarial training, reconstruction, and mutual minimization.

Adversarial Training. We aim to disentangle the visual features into \(h^{cor}\) and \(h^{ind}\) in an adversarial way.

In the classification training step, we input visual features \(X^{ss}\) into Feature Extractor Network \(E\), Task-correlated Network \(E_{cor}\) and Task-independent Network \(E_{ind}\) to disentangle the vectors into two factors. Then we train \(E, E_{cor}\) and \(E_{ind}\) in the supervised ways by using the compatibility loss which associates the visual and the semantic. The compatibility score function is parameterized by \(W\), and is typically formulated as the bilinear compatibility function:

\[
\tau_i^k = h_iW a_k
\]
where the $h_i$ can be $h^{cor}$ or $h^{ind}$ of one sample after disentangling. And we further denote the attribute matrix:

$$\tau_i = h_i W A^*$$

where the $A^*$ should be $A$ in the second training stage. We can consider $\tau_i$ as a classification score in the cross-entropy (CE) loss. So we further develop the compatibility loss function [43], which can be formulated as:

$$L_{ce} = \frac{1}{n_b} \sum_{i=1}^{n_b} L(\tau_i, y_i)$$

where the $L(\cdot)$ means the CE loss. The $n_b$ means the size of one batch.

In the adversarial training step, we fix the parameter $W$ of compatibility function and train $E$ and $E_{ind}$ to fool the classifier by minimizing the negative entropy of the predicted class distribution of $h^{ind}$ outputted by $E_{ind}$.

**Reconstruction.** To guarantee the disentangled factors are meaningful, we reconstruct the confounding vector from them. Concretely, we concatenate $h^{cor}$ and $h^{ind}$, and then input it into Reconstructor $R$ to recover the confounding visual features $\hat{\mathcal{X}}^{ss}$. Finally, the reconstruction loss function can be formulated as:

$$L_{rec} = \| \hat{\mathcal{X}}^{ss} - \mathcal{X}^{ss} \|_F^2$$

where the $\hat{\mathcal{X}}^{ss}$ is the reconstructed vector of $\mathcal{X}^{ss}$, we use the $L_{rec}$ to train $R$.

**Mutual Minimization.** We need to make sure two factors are independent of each other. Concretely, we minimize the mutual information [3] between $h^{cor}$ and $h^{ind}$ in an unsupervised way. The mutual information minimization loss function can be formulated as:

$$L_{mine} = \text{Mine}(h^{cor}, h^{ind})$$

$$= H(h^{cor}) - H(h^{cor} | h^{ind})$$

$$= H(h^{ind}) - H(h^{ind} | h^{cor})$$

$$= \sum_{h^{cor}, h^{ind}} p(h^{cor}, h^{ind}) \log \frac{p(h^{cor}, h^{ind})}{p(h^{cor})p(h^{ind})}$$

where $H(\cdot)$ means the Shannon entropy and the $H(h^{cor} | h^{ind})$ means the conditional entropy of $h^{cor}$ given $h^{ind}$. The $p(h^{cor}, h^{ind})$ means the joint probability distribution of $(h^{cor}, h^{ind})$. We use the $L_{mine}$ to train $E$, $E_{cor}$ and $E_{ind}$.

### 3.4. Controllable Pseudo Samples Synthesis

This module is consists of pseudo sample synthesis and adversarial domain classification.
Pseudo Sample Synthesis. Firstly, we use the Center Convert Net $C_{center}$ and the Edge Convert Nets $C_{edge}$ to generate $\hat{h}_{center}$ and $\hat{h}_{edge}$ respectively. The inputs of these networks are the difference between semantic features of target classes and that of source classes. Then the corresponding offsets are further added to $h_{cor}$ to synthesize $\hat{h}_{center}$ and $\hat{h}_{edge}$ of target classes respectively. This process can be formulated as:

$$\hat{h}_{center} = h_{cor} + o_{center}$$  \hspace{1cm} (6)

$$\hat{h}_{edge} = h_{cor} + o_{edge}$$  \hspace{1cm} (7)

where the $o_{center}$ and $o_{edge}$ can be further formulated as:

$$o_{i,j}^{center} = C_{center}(a_i - a_j)$$  \hspace{1cm} (8)

$$o_{i,j}^{edge} = C_{edge}(a_i - a_j)$$  \hspace{1cm} (9)

where $i$ and $j$ are the specific classes that from target classes and source classes respectively.

For different synthesizing samples, we have the following training process. Firstly, we use $\hat{h}_{center}$ and $\hat{h}_{edge}$ to train the $C_{center}$ and the $C_{edge}$ with classification loss in Eq. 2 and Eq. 3 by labeling as target classes. Furthermore, to guarantee $\hat{h}_{center}$ distribute in the center of classes, we add the additional transfer loss to $\hat{h}_{center}$ to train the $C_{center}$. The transfer loss is based on Eq. 2 and Eq. 3 but using the soft labels that are computed by cosine similarity of semantic features between source classes and target classes. Then, we introduce the adversarial self-supervised [20] that based on $\hat{h}_{edge}$ to train the model. Concretely, for the $\hat{h}_{edge}$, the aforementioned process is similar to FSGM [10] algorithm and the offsets $o_{edge}$ are similar to the perturbations in adversarial examples. We still label $\hat{h}_{center}$ as target classes to train the parameters $W$ of compatibility function by Eq. 2 and Eq. 3 while we label $\hat{h}_{edge}$ as source classes. In this way, we can strengthen the model’s adversarial robustness and further contribute to the generalization of the model. 

Adversarial Domain Classification. We aim to synthesize pseudo samples more consistent with real samples which are achieved by adversarial domain classification. Specifically, we use the Domain Identifier $DI$, which takes $\hat{h}_{center}$ and $h_{cor}$ of $X^{st}$ as input and output the domain label $l_f$ and $(1 - l_f)$ respectively. The loss function can be formulated as:

$$L_{DI} = -l_f log(l_f) + (1 - l_f)log(1 - l_f)$$  \hspace{1cm} (10)

Then the $C_{center}$ is trained by exchanging domain labels of real and pseudo samples to fool the $DI$. The $\hat{h}_{edge}$ distribute between source classes and target classes, so we have not taken them into domain classification.

### 3.5. Optimization and Unseen Samples Prediction

Our model is trained with different losses iteratively. In the second training stage, we regard the seen classes as source classes and unseen classes as target classes. We use the transfer loss to finetune $C_{center}$ firstly and then synthesize $\hat{h}_{center}$ of target classes to finetune $W$ by Eq. 2 and Eq. 3. The first training stage and the second training stage are running alternately in one epoch.

Once the model training is completed, we can project the visual features into semantic space and measure the similarity with the semantic features of all classes in the GZSL task. Specifically, to predict the class label, the location of the maximum compatibility score can be chosen as the predicted label:

$$y = \arg \max_{k \in S + U} \phi(x)T a_k$$  \hspace{1cm} (11)

where $\phi(\cdot)$ includes the $E$, $E_{cor}$ and the $W$.

### 4. Experiments

#### 4.1. Experiments Setting

Datasets. We selected four popular datasets which are Animal with Attribute (AWA1) [21], Animal with Attribute2 (AWA2) [41], Caltech-UCSD Birds-200-2011 (CUB) [40] and Oxford 102 flowers (FLO) [31]. AWA1 and AWA2 are coarse-grained while others are fine-grained.
The semantic features of CUB and FLO are from the CNN-RNN features [32, 44]. Our dataset split is under the PS setting [41]. The details are presented in Table 1.

**Evaluation Metric.** Average Class Accuracy (ACA) is adopted as the evaluation metric in the GZSL and FSZU tasks. We use the average per-class top-1 accuracy of unseen classes \( U \) and seen classes \( S \) to calculate the harmonic mean \( H \):

\[
H = \frac{(2 \times U \times S)}{U + S}
\]  

**Comparison Methods.** Since our model is a non-generative model, we mainly compare our proposed methods against current non-generative models. The main differences between compared methods and our TDCSS are shown in Table 2.

**Implementation Details.** We utilize the 2048D visual features extracted by pre-trained ResNet-101 [13]. The \( E \), \( E_{cor} \) (\( E_{cor} \)), and \( DI \) consist of two-layer fully connected (FC) neural networks, in which the output units are 1800, 1024, and 2 respectively. The \( C \), \( W \), and \( R \) are three-layer FC neural networks, in which the hidden units are 1024, 512, and 1800. The output units of \( C \) are 1024. We use the LeakyReLU as the activation function for \( DI \) while the ReLU for others. Our model is implemented with PyTorch and optimized by ADAM optimizer. We set the learning rate as \( 2 \times 10^{-4} \) in the first training stage and \( 1 \times 10^{-5} \) in the second training stage, epoch as 1500 for most. And in every epoch, we iterate 30 batches in the first training stage and 10 in the second training stage. Because of the limitation of sample size on every class, the batch size is 32 for the FLO dataset and 64 for the other datasets. For source/target split in the first stage, we set the number of target classes to 2 for CUB and 1 for others.

### 4.2. Evaluations in GZSL Setting

The classification performances in the GZSL tasks are shown in Table 3. We observe that the TDCSS achieves competitive results on four datasets.

Compared with non-generative models, the \( H \) value of our model increases from 47.3% to 60.9% on AWA1, from 64.7% to 66.1% on AWA2, and from 64.5% to 66.2% on FLO. Specifically, the TCN contains the soft labels to quantify the transfer process in GZSL, which we add to \( \hat{h}_{center} \) synthesis process. It can be concluded that our model achieves improvement in addition to the transfer loss. The SP-AEN and AREN are based on tensor-level features. For the SP-AEN that tries to disentangle features, the experimental results prove our model is more effective. For the AREN+CS, our model is still competitive except on CUB dataset. However, the mechanism of calibrated stacking (CS) [4] helps the AREN achieve a great improvement in the GZSL task. But it is a post-processing operation and very susceptible to the influence of the parameter values that are manually set by the researchers. Our model does not use the CS but with comparable results, which shows TDCSS is more robust. For the AGZSL synthesized samples but with no representation disentanglement, which proves the effectiveness of representation disentanglement of our model.

Compared with generative models, the \( H \) value of our model increases from 60.9% to 66.1% on AWA2 and from 64.6% to 66.2% on FLO. However, the pseudo samples synthesized by our model have certain characteristics, which can further explore the role of different types of pseudo samples in GZSL knowledge transfer. The further detailed experiments are shown in section 4.3. The DLFZRL disentangles features by generative networks, our model also achieves comparable results generally but in a simpler concept and method.

---

Table 3. Performance (in %) comparisons for GZSL in terms of unseen accuracy (u), seen accuracy (s), and their harmonic mean (H).

<table>
<thead>
<tr>
<th>Methods</th>
<th>AWA1</th>
<th>AWA2</th>
<th>CUB</th>
<th>FLO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>u</td>
<td>s</td>
<td>H</td>
<td>u</td>
</tr>
<tr>
<td>DEM [45]</td>
<td>32.8</td>
<td>84.7</td>
<td>47.3</td>
<td>30.5</td>
</tr>
<tr>
<td>RELATION NET [37]</td>
<td>31.4</td>
<td>91.3</td>
<td>46.7</td>
<td>30.0</td>
</tr>
<tr>
<td>DCN [25]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>25.5</td>
</tr>
<tr>
<td>TCN [17]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>61.2</td>
</tr>
<tr>
<td>SP-AEN [5]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>23.3</td>
</tr>
<tr>
<td>AREN+CS [43]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>54.7</td>
</tr>
<tr>
<td>AGZSL [8]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>46.6</td>
</tr>
<tr>
<td>f-VAEGAN-D2 [42]</td>
<td>57.6</td>
<td>70.6</td>
<td>63.5</td>
<td>-</td>
</tr>
<tr>
<td>DLFZRL [38]</td>
<td>-</td>
<td>-</td>
<td>61.2</td>
<td>-</td>
</tr>
<tr>
<td>TDCSS</td>
<td>54.4</td>
<td>69.8</td>
<td>60.9</td>
<td>59.2</td>
</tr>
</tbody>
</table>

Table 4. Ablation study (in %) of the TDCSS components on AWA2 datasets in terms of unseen accuracy (u), seen accuracy (s), and their harmonic mean (H).

<table>
<thead>
<tr>
<th>Setting</th>
<th>AWA2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>u</td>
</tr>
<tr>
<td>TDCSS w/o TFD</td>
<td>52.7</td>
</tr>
<tr>
<td>TDCSS w/o EPS</td>
<td>44.5</td>
</tr>
<tr>
<td>TDCSS w/o CPS</td>
<td>34.9</td>
</tr>
<tr>
<td>TDCSS</td>
<td>59.2</td>
</tr>
</tbody>
</table>
4.3. Ablation Study

We take the AWA2 dataset into the ablation analysis and aim to demonstrate that the main components of the TD-CSS both contribute to the final performance. We also observe the role of different types of pseudo samples in the knowledge transfer of the GZSL task. The best performance is achieved when TFD (Task-correlated Feature Disentanglement), EPS (Edge-Pseudo Samples) and CPS (Center-Pseudo Samples) are both applied. We have the following main findings:

(1) Compared with the model w/o TFD, it can be shown that disentanglement has little effect on the recognition of seen classes in the model, but has a greater impact on unseen classes. It proves that the disentanglement module, which extracts more consistent visual features with class semantics, is of great help to the knowledge transfer from the seen class to the unseen class.

(2) Compared with the model w/o EPS, the experiments show that our whole model has a certain improvement in the accuracy of the seen and unseen classes, indicating that $h_{edge}$ with adversarial self-supervised training contribute to the consistency between $h_{cor}$ and class semantics, and further improving the robustness and generalization of the model.

(3) Compared with the model w/o CPS, on the one hand, the precision on seen classes of the whole TDCSS is worse than it. It has demonstrated that $h_{edge}$ can further perfect the classification boundary for seen classes. On the other hand, the precision on unseen classes has significantly declined, it has proved $h_{center}$ play a key role in the knowledge transfer from the seen classes to the unseen classes in the GZSL task.

4.4. Qualitative Analysis

We take the AWA2 dataset into the following qualitative analysis.

**Task-correlated features visualization** We visualize $h_{cor}$ and $h_{ind}$ in Figure 4 (a) and (b) to validate the properties of the disentanglement. It shows that $h_{cor}$ are much more discriminative than $h_{ind}$. But $h_{ind}$ remain some discriminative, which we guess that some characteristics are not annotated in semantics. We further visualize $h_{cor}$ by saliency maps [36] which compute the gradient of the output of $E_{cor}$ against the original image inputted in the backbone network. The results are shown in Figure 3. We can observe that the saliency maps focus on the task-correlated information, especially the objects. And the task-independent information is effectively filtered. But there are also some failed samples that animals blend with their surroundings.

**Distribution visualization of pseudo samples** To demonstrate that our method can synthesize two types of pseudo samples effectively, we randomly select two classes and visualize the distributions of part samples. As Figure 4 (c) and (d) shows, we can observe that the distributions of $h_{center}$ are closer to real samples than that of $h_{edge}$ on the whole, which are consistent with their characteristics.
Table 5. Performance (in %) comparisons for FSZU in terms of unseen accuracy (u), seen accuracy (s), and their harmonic mean (H).

<table>
<thead>
<tr>
<th></th>
<th>All data</th>
<th>Num = 10</th>
<th>Num = 5</th>
<th>Num = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>u</td>
<td>s</td>
<td>H</td>
<td>u</td>
</tr>
<tr>
<td>Disentangled-VAE [23]</td>
<td>50.9</td>
<td>79.8</td>
<td>62.2</td>
<td>50.8</td>
</tr>
<tr>
<td>SDGZSL [7]</td>
<td>74.4</td>
<td>63.6</td>
<td>68.6</td>
<td>47.1</td>
</tr>
<tr>
<td>AGZSL [8]</td>
<td>46.6</td>
<td>74.2</td>
<td>57.3</td>
<td>18.3</td>
</tr>
<tr>
<td>TDCSS</td>
<td>59.2</td>
<td>74.9</td>
<td>66.1</td>
<td>56.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>All data</th>
<th>Num = 10</th>
<th>Num = 5</th>
<th>Num = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>u</td>
<td>s</td>
<td>H</td>
<td>u</td>
</tr>
<tr>
<td>Disentangled-VAE [23]</td>
<td>52.1</td>
<td>54.2</td>
<td>53.1</td>
<td>44.7</td>
</tr>
<tr>
<td>SDGZSL [7]</td>
<td>61.2</td>
<td>65.3</td>
<td>63.2</td>
<td>39.4</td>
</tr>
<tr>
<td>AGZSL [8]</td>
<td>42.1</td>
<td>48.1</td>
<td>44.9</td>
<td>29.3</td>
</tr>
<tr>
<td>TDCSS</td>
<td>44.2</td>
<td>62.8</td>
<td>51.9</td>
<td>40.1</td>
</tr>
</tbody>
</table>

Figure 5. The experimental results under different numbers of target classes when the sample size of each class is 2.

4.5. Evaluations in FSZU Setting

For compared methods, we select two generative models that disentangle the overall representations into two factors by random permuting. The Disentangled-VAE [23] consists of two parallel VAEs and each with two branches. The SDGZSL [7] consists of VAE, AE, and the RELATION NET [37], while the VAE is used for data enhancement. We also select the non-generative AGZSL that synthesizes pseudo samples in mixup interpolation for comparison.

For experimental settings, we select AWA2 and CUB datasets, that appeared simultaneously in the Disentangled-VAE, SDGZSL, and AGZSL. And we reduce the sample size of seen classes (the sizes are set to 10, 5, and 2) to stimulate the FSZU. It should be pointed out that the Disentangled-VAE is reproduced by us based on Python 3.6 and Pytorch 1.0.1. For SDGZSL and AGZSL, we use the codes that have been released on Github. The performances in the FSZU tasks are shown in Table 5.

Compared with generative models, we achieve 22.4% improvement in $H$ value on average for AWA2 and 8.8% for CUB. These experimental results reflect that our model can still work effectively in the new task. The generative model, especially SDGZSL, has excellent performances in GZSL but degrade sharply with the decrease in sample size, which shows more non-robust compared with our model. And our model can synthesize more diverse pseudo samples based on the limited seen class samples in a non-generative way.

Compared with the non-generative model that synthesizes pseudo samples, we achieve 117.9% improvement in $H$ value on average for AWA2 and 65.5% for CUB. It shows that the representation disentanglement before synthesizing samples is reasonable and important. The results in the FSZU also show that the mixup interpolation of the AGZSL will lead the diversity of the synthesis samples to decrease sharply as the sample size decreases.

In the task of FSZU, we can improve the model’s performance by increasing the sample size of the target class with increasing the number of target classes. The results are shown in Figure 5. It shows that within a certain range, the accuracy will increase as the number of target classes increases, but too many target classes will cause the performance to decrease. It is a corollary that too many synthesized pseudo samples would be the leading data in the training process and further distraction the model from recognizing real samples.

5. Conclusions

In this paper, we propose a non-generative model, TPDSS, to perform the task-correlated feature disentanglement and diversity pseudo samples synthesis in the GZSL and the FSZU tasks. For disentanglement, the TPDSS uses the adversarial training of domain adaptation to achieve it. For synthesis, the TPDSS synthesizes diverse pseudo samples with certain characteristics. The above mechanisms make our model achieve competitive performances in different tasks and help people intuitively understand the role of different types of pseudo samples in the knowledge transfer of ZSL.

Acknowledgments

This work is supported by the National Key R&D Program of China (Grant No. 2018AAA0100604), the National Natural Science Foundation of China (Grant No. 61832004, 61632002), and Beijing Natural Science Foundation (No. JQ20023).
References


