An Optimal Transport Framework for Zero-Shot Learning

[1]Wenlin Wang, [2]Hongteng Xu, [1]Guoyin Wang, [3]Wenqi Wang, [1]Lawrence Carin
[1]Duke University, [2]Infinia ML, [3] Facebook
wenlin.wang@duke.edu

Abstract

We present an optimal transport (OT) framework for generalized zero-shot learning (GZSL) of imaging data, seeking to distinguish samples for both seen and unseen classes, with the help of auxiliary attributes. The discrepancy between features and attributes is minimized by solving an optimal transport problem. Specifically, we build a conditional generative model to generate features from seen-class attributes, and establish an optimal transport between the distribution of the generated features and that of the real features. The generative model and the optimal transport are optimized iteratively with an attribute-based regularizer, that further enhances the discriminative power of the generated features. A classifier is learned based on the features generated for both the seen and unseen classes. In addition to generalized zero-shot learning, our framework is also applicable to standard and transductive ZSL problems. Experiments show that our optimal transport-based method outperforms state-of-the-art methods on several benchmark datasets.

1. Introduction

When there is access to abundant labeled examples for image-based data, modern machine learning and deep learning algorithms have demonstrated the ability to learn reliable classifiers [27, 56, 53, 67, 66, 71, 68, 51, 64]. Unfortunately, their ability to generalize to unseen classes typically remains poor. This limitation has motivated significant recent interest in zero-shot learning (ZSL) [55, 33, 62, 44]. By leveraging auxiliary information that may be available for the seen/unseen classes, $e.g.$, attribute vectors and/or textural descriptions of classes [39], ZSL aims to learn new concepts with minor or no supervision ($i.e.$, distinguish data for classes unseen in the training phase).

Although many ZSL methods have been proposed, they often suffer from inherent limitations. A typical problem is "*domain shift*." Many ZSL methods try to establish a mapping between the feature space and the class/attribute space, and predict the unseen classes by finding their closest attribute vectors [2, 33]. However, seen classes and unseen ones often suffer from clear domain differences in high-dimensional space. Accordingly, the mapping learned based on the seen classes may be inapplicable to the unseen classes. Another problem is "*high-bias*." Most methods highly bias towards predicting the seen classes [58, 75], because the training data are purely from the seen classes. Such a phenomenon is important in generalized zero-shot learning (GZSL), where the proposed classifier needs to distinguish samples for both seen and unseen classes.

To overcome the above problems, we propose an optimal-transport-based zero-short learning method. The proposed method is motivated by recent *synthesis* of exemplars for ZSL [58, 40, 75], extended by building a more powerful generative model via optimal transport (OT) [60]. As shown in Figure 1, our method learns a conditional generative model to construct features from attributes, and minimizes the optimal transport distance (also called Wasserstein discrepancy) between the generated features and the real ones. The generative model helps to synthesize samples for unseen classes from given attributes. We apply iterative optimization to jointly learn the generative model and the corresponding optimal transport, and further enhance the discriminative power of the generated data via learning an associated attribute predictor as a regularization. Finally, an additional classifier is trained to distinguish generated data from both seen and unseen classes.

Distinct from existing methods, which minimize the discrepancy between features and attributes on each individual data, our method considers the optimal transport between the *distribution* of features and that of attributes. Matching distributions [57, 75, 72, 12, 69, 80, 50, 79] is found to be more robust than finding correspondences between individual samples, with this beneficial for suppressing the risk of domain shift caused by mismatch. Additionally, the generative model can synthesize instances for unseen classes, with the synthesized data available for training the classifier. Therefore, the *high-bias* problem can be suppressed effectively. Moreover, our method is the first framework conducting ZSL under the primal form of optimal transport (OT), and it provides universal solutions to

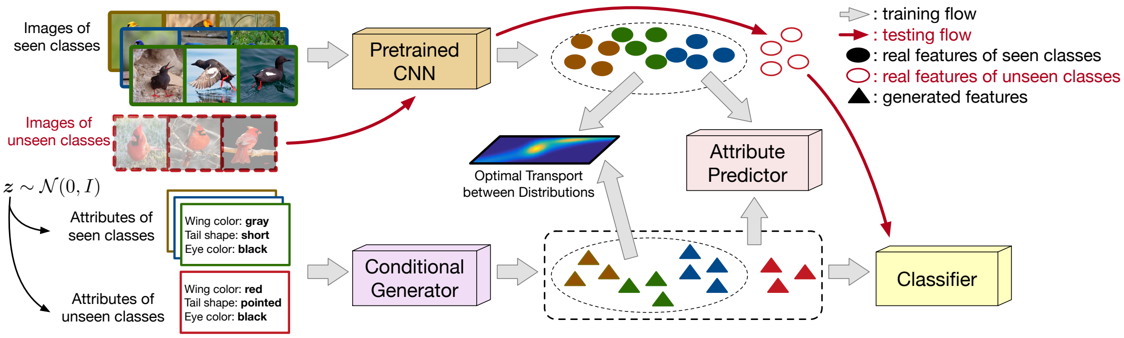

Figure 1. The diagram of proposed method. In the training phase, a generator is learned to synthesize features from attributes, whose distribution approaches to that of real features obtained from a pretrained CNN. After the generator is learned, we further train a classifier based on the generated features for both seen and unseen classes. In the testing phase, the "pretrained CNN + classifier" achieves zero-shot learning, whose pipeline is shown as red arrows.

ZSL problems. It is applicable not only to the generalized ZSL [76] problem, but also to the standard ZSL [33] and transductive ZSL [45]. We analyze the assumptions of our method in depth and investigate its robustness to hyper-parameters. Experimental results demonstrate that the proposed approach outperforms state-of-the-art methods on several image-based benchmark datasets.

2. Proposed Framework

Suppose there are S seen classes and U unseen classes. For the i-th class, $i \in \{1, ..., S+U\}$, we are given a d-dimensional attribute vector [55], denoted $\boldsymbol{a}_i \in \mathbb{R}^d$. All the attribute vectors formulate an attribute matrix $\boldsymbol{A} = [\boldsymbol{a}_i] \in \mathbb{R}^{d \times (S+U)}$. Data from the same class share the same attributes and different classes always have different attributes. The seen classes contain N labeled samples $\mathcal{D}_s = \{(\boldsymbol{x}_n, \boldsymbol{a}_n)\}_{n=1}^{N}$, where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $\boldsymbol{a}_n \in \{\boldsymbol{a}_1, ..., \boldsymbol{a}_S\}$ represent the n-th sample and its attribute/label, respectively. For the U unseen classes, the unlabeled data are denoted $\mathcal{D}_u = \{\boldsymbol{x}_{n+N}\}_{n=1}^{N'}$, whose correspondences with $\{\boldsymbol{a}_{S+1}, ...\boldsymbol{a}_{S+U}\}$ are unknown. In the work considered here, $\boldsymbol{x}_n$ corresponds to features extracted from image n, and such features will be manifested by a deep neural network operating on the image.

We consider three settings for zero-shot learning: standard, generalized and transductive. In the standard setting [33], we only have access to $\mathcal{D}_s$ and the attribute matrix $\boldsymbol{A}$ in the training phase, and focus on classification of the samples in $\mathcal{D}_u$ in the testing phase. Generalized ZSL [76] requires the final classifier to categorize the samples in the whole $U + S$ classes. Similar to the standard setting, the transductive ZSL [31] also aims to train a classifier for the U unseen classes, but it uses both the labeled data $\mathcal{D}_s$ and the unlabeled data $\mathcal{D}_u$ in the training phase.

We solve these three ZSL tasks within a unified framework. The key of the proposed approach is to learn a conditional generative model (*i.e.*, the "*generator*" in Figure 1), capable of generating representative features for both seen and unseen classes, conditioned on their attributes. Those generated features are used to train a classifier.[1]

The quality of the generated features has a significant influence on the target classifier, which is purely decided by the proposed generator. According to the nature of ZSL problems, it is necessary for an ideal generator to have the following properties:

1. For seen classes, the distribution of their generated features should be close to that of real features.

2. For both seen and unseen classes, the generated features should have discriminative power, e.g., the features from the same attribute have a clustering structure.

The first property means that the generator establishes a reliable mapping from attributes to features. The second property ensures that the features synthesized by the generator are informative to train a classifier. The generator with these two properties can synthesize realistic features and be extended to those unseen classes. Since the generator can synthesize a large quantity of features for unseen classes when training the classifier, the bias between seen and unseen classes can be suppressed. Guided by these two properties, we design an optimal transport-based method to learn the generator.

3. Conditional Feature Generator

3.1. Optimal transport-based loss

The conditional feature generator is denoted $g : \mathcal{A} \mapsto \mathcal{X}$, with $\mathcal{A}$ and $\mathcal{X}$ the attribute and feature space, respectively. Given an attribute vector $\boldsymbol{a} \in \mathcal{A}$, we generate a set of synthetic features via

$$\hat{\boldsymbol{x}} = g([\boldsymbol{a}; \boldsymbol{z}]), \ \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d), \tag{1}$$

[1] In standard and transductive ZSL problems, the classifier is just for the U unseen classes. In the generalized ZSL problem, the classifier is for all $U + S$ classes.

where $\boldsymbol{z} \in \mathbb{R}^d$ is a random variable drawn from a normal distribution, and $[\boldsymbol{a}; \boldsymbol{z}]$ denotes the concatenation of vectors $\boldsymbol{a}$ and $\boldsymbol{z}$. We choose to make the dimension of $\boldsymbol{z}$ the same as that of $\boldsymbol{a}$; while not required, it was found to work well in practice.

For the seen classes, we have N real features $\{\boldsymbol{x}_n\}_{n=1}^N$,[2] and obtain M generated features $\{\hat{\boldsymbol{x}}_m\}_{m=1}^M$ from the attributes of the seen classes via (1). Instead of finding correspondences between the features generated from the attribute space and those in the feature space, we aim to match the empirical *distribution* of the generated features to that of real features (as mentioned in Property 1). We utilize distribution matching because point-wise matching in high-dimensional space is sensitive to fluctuations of samples and to outliers, yielding solutions that often fall into bad local optima with incorrect correspondences. Distribution matching, on the contrary, considers more variability of the data and matches the data globally. In practice, distribution matching has proven to be effective for transferring knowledge from one modality to another [37, 57, 28].

To measure the distance between distributions, the optimal transport (OT) distance [60] is a natural choice. Mathematically, the optimal transport distance between two probability measures $\boldsymbol{\mu}$ and $\boldsymbol{v}$ is defined as [43]:

$$d_c(\boldsymbol{\mu}, \boldsymbol{v}) = \inf_{\pi \in \Pi(\boldsymbol{\mu}, \boldsymbol{v})} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \sim \pi}[c(\boldsymbol{x}, \boldsymbol{x}')] \tag{2}$$

where $\Pi(\boldsymbol{\mu}, \boldsymbol{v})$ is the set of all joint distributions with $\boldsymbol{\mu}$ and $\boldsymbol{v}$ as marginals, and $c(\boldsymbol{x}, \boldsymbol{x}')$ is the cost for moving from $\boldsymbol{x}$ to $\boldsymbol{x}'$. When the cost is defined as the Euclidean distance, $i.e.$, $\|\boldsymbol{x} - \boldsymbol{x}'\|_2$, (2) corresponds to the Wasserstein distance. When the cost is not a metric, which is common in practice, (2) is also called Wasserstein discrepancy.

In our case, both the *empirical* distribution of real features and that of generated features are represented as uniformly distributed, denoted as $\boldsymbol{\mu} = [\frac{1}{N}] \in \mathbb{R}^N$ and $\boldsymbol{v} = [\frac{1}{M}] \in \mathbb{R}^M$, and (2) can be rewritten as

$$\begin{aligned} \mathcal{L}_{OT}(\mathcal{D}_s, \boldsymbol{A}; g) &= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{v})} \sum\nolimits_n \sum\nolimits_m T_{nm} C_{nm} \\ &= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}, \boldsymbol{v})} \mathrm{tr}(\boldsymbol{T}^\top \boldsymbol{C}). \end{aligned} \tag{3}$$

Here, $\Pi(\boldsymbol{\mu}, \boldsymbol{v}) = \{\boldsymbol{T} | \boldsymbol{T}\mathbf{1}_M = \boldsymbol{\mu}, \boldsymbol{T}^\top \mathbf{1}_N = \boldsymbol{v}\}$. The cost $C_{nm} = C(\boldsymbol{x}_n, \hat{\boldsymbol{x}}_m)$ is the distance between the n-th real feature and the m-th generated feature, which can be designed with high flexibility. Since the usage of Euclidean distance in a high dimensional embedding space will lead to a severe *hubness* problem [18], we define the cost as the cosine distance $C_{nm} = 1 - \frac{\boldsymbol{x}_n^\top \hat{\boldsymbol{x}}_m}{\|\boldsymbol{x}_n\|_2 \|\hat{\boldsymbol{x}}_m\|_2}$. $\boldsymbol{T} = [T_{nm}] \in \mathbb{R}^{N \times M}$ is the proposed optimal transport matrix. The element $T_{nm} \geq 0$ denotes the probability that the real feature $\boldsymbol{x}_n$ matches with the generated feature $\hat{\boldsymbol{x}}_m$.

The optimal transport distance supplies an unsupervised alignment between the empirical distribution of real features and that of synthetic features generated from attributes. By minimizing this distance, we can learn a generator to synthesize reasonable features consistent with the real data distribution. Compared with other distances, $e.g.$, KL-divergence [19, 70] and maximum mean discrepancy (MMD) [57], optimal transport distance has some advantages. Specifically, KL-divergence is asymmetric and may suffer from a "vanishing gradient" problem when the supports of two distributions are non-overlapped [11] (which is common when our generator is initialized randomly). MMD just considers the similarity of the first order statistic, whose constraints are too loose for distribution matching. The proposed optimal transport distance, however, is applicable to non-overlapped distributions, which suppresses the "vanishing gradient" problem when learning our generator. Additionally, the optimal transport distance imposes more constraints on the generator than MMD does, which accordingly has lower risk of over-fitting.

3.2. Attribute-based regularizer

By minimizing the optimal transport distance, $i.e.$, $\min_g \mathcal{L}_{OT}(\mathcal{D}_s, \boldsymbol{A}; g)$, we ensure that the generator can mimic the distribution of features based on the corresponding attributes. However, there is no guarantee that the generated features are discriminative enough to train a good classifier. To reduce this potential risk, we propose an attribute-based regularizer. In particular, given a feature vector, either the real $\boldsymbol{x} \in \mathcal{D}_s$ or the synthetic $\hat{\boldsymbol{x}}$ from our generator, the proposed regularizer aims to maximize its conditional likelihood given the corresponding attribute. Taking advantage of neighborhood component analysis (NCA) [24], we define the conditional likelihood of a feature $\boldsymbol{x}$ (or $\hat{\boldsymbol{x}}$) given an attribute $\boldsymbol{a}$ as

$$p(\boldsymbol{x}|\boldsymbol{a}) = \frac{\exp(-\gamma^2 d(f(\boldsymbol{x}), \boldsymbol{a}))}{\sum_{i=1}^{S+U} \exp(-\gamma^2 d(f(\boldsymbol{x}), \boldsymbol{a}_i))}, \tag{4}$$

where $f : \mathcal{X} \mapsto \mathcal{A}$ is a predictor estimating the attributes of generated features, and γ^2 is a hyper-parameter tuning the strength of the discriminator power. The discrepancy between the proposed attribute and the predicted one is defined by the cosine similarity $d(f(\boldsymbol{x}), \boldsymbol{a}) = 1 - \frac{f(\boldsymbol{x})^\top \boldsymbol{a}}{\|f(\boldsymbol{x})\|_2 \|\boldsymbol{a}\|_2}$.

Accordingly, we calculate negative log-likelihood of both the labeled features from the seen classes and those generated from both seen and unseen classes, and impose it on our generator as a regularizer:

$$\begin{aligned} &\mathcal{L}_P(\mathcal{D}_s, \boldsymbol{A}; g, f) \\ &= -\mathbb{E}_{\boldsymbol{x}, \boldsymbol{a} \sim \mathcal{D}_s}[\log p(\boldsymbol{x}|\boldsymbol{a})] - \mathbb{E}_{\hat{\boldsymbol{x}}}[\log p(\hat{\boldsymbol{x}}|\boldsymbol{a})] \\ &= -\mathbb{E}_{\boldsymbol{x}, \boldsymbol{a} \sim \mathcal{D}_s}[\log p(\boldsymbol{x}|\boldsymbol{a})] - \mathbb{E}_{\boldsymbol{z}}[\log p(g([\boldsymbol{a}; \boldsymbol{z}])|\boldsymbol{a})]. \end{aligned} \tag{5}$$

By minimizing (5), the generator is endowed discriminative power for the generated features. Further, this regularizer

[2] In this work, we focus on the zero-shot learning problem in image classification. Taking images as input, a pre-trained convolutional neural network outputs the real features.

provides guidance for the learning of the generative model, navigating the generator to construct high-quality instances yielding more rational transport.

Combining the optimal-transport-based loss with the attribute-based regularizer, we optimize the proposed feature generator g associated with the attribute predictor f via

$$\min_{g,f} \mathcal{L}_{OT}(\mathcal{D}_s, \boldsymbol{A}; g) + \beta\mathcal{L}_P(\mathcal{D}_s, \boldsymbol{A}; g, f) \tag{6}$$

where β is a hyper-parameter weighting the optimal transport loss and the attribute predictor.

4. Learning Algorithm

The optimization problem (6) can be solved by an iterative optimization strategy. Specifically, on each iteration we first calculate the optimal transport distance given the cost derived from the current generator g, and then optimize g and f based on estimated optimal transport matrix $\boldsymbol{T}$.

4.1. Calculating optimal transport distance

Given the generator g obtained in the previous iteration, we generate M features $\{\hat{\boldsymbol{x}}_m\}_{m=1}^M$ and calculate the cost matrix $\boldsymbol{C}$ accordingly. The optimal transport distance $\mathcal{L}_{OT}$ can then be calculated by solving (2). Instead of applying linear programming (LP) directly to solve (2), whose complexity is $\mathcal{O}(N^3 \log N)$, in this work we calculate the optimal-transport distance using the Inexact Proximal point method for Optimal Transport (IPOT) [78]. This method finds the optimal transport $\boldsymbol{T}$ iteratively, and in each iteration it imposes a Bregman divergence term D_B to (2). Accordingly, the optimization problem becomes

$$\boldsymbol{T}^{(t+1)} = \min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu},\boldsymbol{v})} \text{tr}(\boldsymbol{T}^\top \boldsymbol{C}) + \lambda D_B(\boldsymbol{T}, \boldsymbol{T}^{(t)}), \tag{7}$$

where $D_B(\boldsymbol{T}, \boldsymbol{T}^{(t)}) = \sum_{n,m} T_{nm} \log \frac{T_{nm}}{T_{nm}^{(t)}}$ calculates the Kullback-Leibler (KL) divergence between the proposed $\boldsymbol{T}$ and its estimation in the t-th iteration; λ controls the significance of the regularizer.

Equation (7) can be rewritten as

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu},\boldsymbol{v})} \text{tr}(\boldsymbol{T}^\top (\boldsymbol{C} - \log \boldsymbol{T}^{(t)})) + \lambda H(\boldsymbol{T}), \tag{8}$$

where $H(\boldsymbol{T}) = \sum_{n,m} T_{nm} \log T_{nm}$ is the entropy term of $\boldsymbol{T}$. Equation (8) is of the same form as the Sinkhorn distance, introduced in [15], which can be solved with near linear complexity. The solution to (8) can be obtained efficiently via the Sinkhorn-Knopp algorithm [54].

The IPOT [78] algorithm is employed in our framework because it has several advantages compared with traditional linear programming-based algorithms [42] and Sinkhorn iteration [15]. First, the per-iteration computational complexity of IPOT is at most comparable to that of Sinkhorn iteration, and is much lower than that of linear programming. Secondly, although both Sinkhorn iteration and IPOT have near-linear convergence, IPOT requires much fewer inner iterations, because it only needs to find inexact proximity in each step. Thirdly, the Sinkhorn iteration algorithm is sensitive to the choice of the regularizer's weight, while the IPOT method is robust to the change of the weight in a wide range. More detailed analysis of the IPOT method can be found in our supplementary file and related references [42, 15, 78].

4.2. Modifications for ZSL problems

Note that (3) provides an unsupervised solution to align the generated data distribution and the empirical data distribution. For the seen classes, however, the correspondence between these data can also be derived by comparing the attribute vectors of the real data with the generator's inputs. To leverage the correspondence provided by attributes, we propose a *stochastic transition* method when learning the optimal transport. In particular, when learning the parameters of our generative model, we have a probability p to learn the optimal transport matrix $\boldsymbol{T}^*$ defined in (7), and $1 - p$ to directly assign the transition matrix $\widetilde{\boldsymbol{T}}$ with supervised signal. Given the real feature $\boldsymbol{x}_n$ associated with the attribute $\boldsymbol{a}_n$ and the synthetic feature $\boldsymbol{x}_m = g([\boldsymbol{a}_m; \boldsymbol{z}_m])$,[3] the element of $\widetilde{\boldsymbol{T}}$ is defined as

$$\widetilde{T}_{mn} = \begin{cases} \frac{1}{\#\{\boldsymbol{a}_n\}\times\#\{\boldsymbol{a}_m\}}, & \text{if } \boldsymbol{a}_n = \boldsymbol{a}_m \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where $\#\{\boldsymbol{a}\}$ counts the number of an attribute appearances. $\widetilde{\boldsymbol{T}}$ is a valid transport matrix in $\Pi(\boldsymbol{\mu}, \boldsymbol{v})$, and provides guidance to align the two distributions in the image feature space. Practically, this method supplies minor improvements for classification, but speeds learning substantially.

4.3. Updating the generative model

Given the optimal transport matrix $\boldsymbol{T}^*$, we further update the generator g and predictor f by

$$\min_{g,f} \text{tr}(\boldsymbol{T}^{*\top} \boldsymbol{C}_g) + \beta\mathcal{L}_P(\mathcal{D}_s, \boldsymbol{A}; g, f), \tag{10}$$

where $\boldsymbol{C}_g$ is the cost matrix parameterized by the generator. This problem can be solved efficiently via stochastic gradient descent. We apply Adam [29] to update g and f. The whole learning process is summarized in Algorithm 1.

4.4. Classifier

Given the generative model learned in the previous section, we are able to generate representative features of both seen and unseen classes by (1). These generated samples, together with the data from the seen classes, can be used to train a classifier, *e.g.*, the simple linear softmax function

[3] Here $\boldsymbol{a}_m \in \{\boldsymbol{a}_1, ..., \boldsymbol{a}_S\}$, which is randomly selected from the attributes of seen classes.

Algorithm 1: Iterative Optimization

```
Input: $\mathcal{D}_s = \{(\boldsymbol{x}_n, \boldsymbol{a}_n)\}_{n=1}^N$, attribute matrix $\boldsymbol{A}$,
    $p = 0.9$, $\gamma^2 = 0.5$, $b = 128$
Output: $g(\cdot)$
while not converge do
    Sample $B_{real} = \{(\boldsymbol{x}_i, \boldsymbol{a}_i)\}_{i=1}^b$ from $\mathcal{D}_s$.
    For seen classes, sample $B_s = \{(\hat{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^b$,
    $\boldsymbol{a}_i \in \{\boldsymbol{a}_1, .., \boldsymbol{a}_S\}$, via (1).
    For unseen classes, sample $B_u = \{(\hat{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^b$,
    $\boldsymbol{a}_i \in \{\boldsymbol{a}_{S+1}, ..., \boldsymbol{a}_{S+U}\}$, via (1).
    Sample $z \sim$ Uniform$[0, 1]$.
    if $z \leq p$ then
        Based on $B_{real}$ and $B_s$, update $\boldsymbol{T} \leftarrow \boldsymbol{T}^*$ via (7)
    else
        $\boldsymbol{T} \leftarrow \tilde{\boldsymbol{T}}$ via (9).
    end if
    Based on $B_s$ and $B_u$, solve (10) via SGD and
    update $\{g, f\}$ accordingly.
end while
```

used in the following experiments. Since the classifier leverages labeled examples from both seen and (synthesized) unseen classes, the corresponding ZSL result is inherently robust against bias towards seen classes.

5. Related Work

Wasserstein GAN (WGAN) [6]. The proposed optimal-transport-based method is akin to a WGAN model [75], with some key differences. WGAN [6] and its variants [26] use Kantorovich-Rubinstein duality to calculate the Wasserstein distance. A constraint for the dual form is that the discriminator must be a 1-Lipschitz function, which may be violated in practice. In our model, we solve the optimal transport problem in its prime form directly. As a result, our method does not play the max-min game like GAN-related work does, and does not need to train an additional discriminator to distinguish real and synthetic features. Additionally, Wasserstain distance is a special case for the optimal transport distance [60]. Our method can be easily adapted to other metrics.

Denoising Auto-Encoder (DAE) [61]. The proposed generator together with an attribute-based regularizer is similar to a DAE [61], by reconstructing the corresponding attribute vector from the noised input attribute. However, our model is specialized for classification tasks. In particular, we introduce a neighborhood component analysis (NCA) [24] loss as the regularizer. Essentially, this regularizer implies that the features should have clustering structure defined by the corresponding attributes. Therefore, the features generated by our model are more discriminative, suitable for training the following classifier.

OT-GAN [48]. Our model is similar to OT-GAN [48] in spirit, learning optimal transport in the primal space, however, different in both modeling and algorithm. OT-GAN [48] is designed for image generation that cannot be directly extended for ZSL problems, while our OT framework serves for ZSL tasks, with numbers of specialized modules, *e.g.*, the stochastic transition method (Sec 4.2) and the attribute regularizer (Sec 3.2). These modules are essential for ZSL. Additionally, OT-GAN [48] applies entropic regularizer and learns optimal transport via the Sinkhorn algorithm [15], while our work uses KL-divergence-based regularizer and the proximal gradient method (*i.e.*, IPOT [78]), which is more robust to hyperparameters and with more stable convergence.

Zero-shot learning methods From the viewpoint of methodology, ZSL methods can be roughly categorized into five types: (i) Learning a mapping from the feature space to the attribute space, and predicting the class of an unseen class test instance by finding its closest class-attribute vector [55, 33, 3, 13]; (ii) Learning a "reverse" projection from the attribute space to the feature space [86, 36], which improves the robustness of nearest-neighbor search; (iii) Representing the classifier for each unseen class as a weighted combination of those for the seen classes, with the combination weights defined by a similarity score of unseen and seen class [85, 9]; (iv) Leveraging a probability distribution for each seen class and extrapolating to those unseen classes [59, 70, 40]. (v) Building on a knowledge graph to predict the image categories [73, 34]. All these types of methods have been widely used in the standard and transductive ZSL problems [21, 35, 45]. Recently, generalized zero-shot learning (GZSL) [10, 76] has been demonstrated to be a more challenging task. Among existing ZSL methods, generative models [30, 25] have achieved significant success, including VAEs [59, 40] and GANs [75]. Our method is derived from generative models and applicable to various ZSL problems. Different from prior work, our framework seeks to generate data by minimizing the Wasserstein distance in the primal space.

Optimal transport and Wasserstein distance Optimal transport and Wasserstein learning have proven useful in distribution estimation [8], alignment [83] and clustering [1, 81, 16], avoiding over-smoothed intermediate interpolation results. The lower bound of Wasserstein distance has been used as a loss function when learning generative models [14, 6]. The main bottleneck of the application of optimal transport is its high computational complexity. This problem has been greatly eased since the Sinkhorn iterative algorithm was proposed in [15], which applies iterative Bregman projection [7] to approximate Wasserstein distance, and achieves a near-linear time complexity [4]. Many more complicated models have been proposed based on Sinkhorn iteration [23, 49] and its variants,

Table 1. Basic information of the considered datasets.

Dataset	$\boldsymbol{a}$	d	#Image	$\#S$ train/val.	$\#U$
AwA1	Attribute	85	30,475	27/13	10
AwA2	Attribute	85	37,322	27/13	10
CUB	Attribute	312	11,788	100/50	50
SUN	Attribute	102	14,340	580/65	72
ImageNet	Word2Vec	1000	254,000	800/200	360

$e.g.$, Greenkhorn iteration [4] and IPOT [78]. Motivated by these prior work, our framework solves the optimal transport problem with IPOT algorithm [78] and demonstrates its superiority on real datasets in Sec 6.5.

6. Experiments

To evaluate the effectiveness of our method (denoted as **OT-ZSL** for optimal transport-based zero-shot learning), we apply it to generalized ZSL (GZSL), standard ZSL (SZSL) and transductive ZSL (TZSL), and compare it with state-of-the-art methods. Additionally, to investigate the functionality of each module in our method, we also consider a variant of our method, which is trained without the attribute-based regularizer, and denoted **OT-ZSL (w/o $\boldsymbol{f}$)**.

6.1. Image datasets and implementation details

We report results on the following datasets, with associated detailed information found in Table 1.

- **Animals with Attributes (AwA)** [33] This is a coarse-grained dataset containing 30,475 images, with 50 classes and 85 attributes. A standard split of 40 seen classes and 10 unseen classes are provided. Recently, an alternative data split is also available [76]. We refer to the data with the original split as AwA1 [33], and with the new split as AwA2 [76].
- **Caltech-UCSD-Birds-200 (CUB)** [63] This dataset consists of 11,788 fine-grained bird images, with 200 classes in total. A split of 150 unseen and 50 seen classes are provided. Following [58], class attributes are obtained by averaging all the image level attributes.
- **SUN Scene Recognition** [77] SUN contains 14,340 images from 717 scenes annotated with 102 attributes. We follow the most widely employed setting, with 645 seen classes and 72 unseen classes.
- **ImageNet** [47] Following [22], 1000 classes from ILSVRC-20112 [47] are used as the seen classes, while 360 non-overlapped classes of ILSVRC-2010 [17] are used as unseen classes.

Features For AwA1, AwA2, CUB and SUN, we extract 2048-dimensional features from a pre-trained 101-layered ResNet [27]. Their class attributes are the corresponding attribute vectors in these four datasets. For ImageNet, to make a fair comparison to previous work, we maintain the usage of GoogleNet [56], which yields a 1024-dimensional extracted feature. Its class attributes are the semantic word vector obtained from word2vec embeddings [39].

Implementation For the reported experiments, we use the proposed train/test split [76] for each dataset for GZSL, and consider both the standard and the proposed train/test split [76] for SZSL. To make a fair comparison to prior work, only the standard split is evaluated for TZSL. For the network architecture, MLP with ReLU activation is used for both the feature generator and the attribute predictor. In all experiments, we use a single hidden layer with 4,096 hidden units. The noise $\boldsymbol{z}$ is sampled from normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. Finally, for each seen/unseen class, we draw 500 synthetic features from our generator, and train a linear softmax classifier. Following [75], we divided the training data into training set and validation set, as shown in Table 1. We set hyper-parameters empirically based on the performance over the validation set. In all experiments below, we set $p = 0.9$, $\lambda = 0.5$, $\gamma^2 = 0.5$, $\beta = 0.05$, and apply Adam [29], with batch in size of 128 and learning rate in 0.001, to train our model.

6.2. Generalized zero-shot learning

In the GZSL setting, seen and unseen classes are evaluated jointly when testing. We use the data split provided in [76]. Accordingly, the testing data from the seen classes and that from the unseen classes are referred to as $\mathcal{X}_{test}^S$ and $\mathcal{X}_{test}^U$, respectively. Given the classifier trained over all $S + U$ classes, we evaluate the performance of different methods via the following three measurements:

1. A_s: average per-class top-1 accuracy on $\mathcal{X}_{test}^S$.
2. A_u: average per-class top-1 accuracy on $\mathcal{X}_{test}^U$.
3. H: harmonic mean of A_s and A_u, $i.e.$, $H = \frac{2A_sA_u}{A_s+A_u}$.

H evaluates the overall performance of the proposed method on both seen and unseen classes, which is the key criterion of GZSL problem. The results, comparing with several state-of-the-arts, are presented in Table 2. The proposed model achieves superior performance for most of the benchmark datasets, especially on the H measure. This demonstrates that the proposed method keeps a balance between the seen and unseen classes better than alternative approaches [41, 9], avoiding a strong bias towards the seen classes. Among all methods, generative model-based ZSL approaches [40, 58, 75] achieve remarkable success for the GZSL task, showing that learning a power generative model is critical for the generalization to those unseen classes. It can be seen that the proposed model achieves comparable, if not the best, A_u over these benchmark datasets.

Note that even if we purely rely on the optimal transport distance as the objective function, and train the proposed model without the attribute-based regularizer, our method can still achieve comparable learning results to the state-of-the-art methods, as shown in the row of "OT-ZSL (w/o f)" in Table 2. This result demonstrates the effectiveness of the optimal transport framework for ZSL problems. On

Table 2. Comparisons for various methods in the generalized ZSL problem, on the split provided by [76].

Methods	AwA1			AwA2			CUB			SUN		
	A_u	A_s	H	A_u	A_s	H	A_u	A_s	H	A_u	A_s	H
SJE [3]	11.3	74.6	19.6	8.0	73.9	14.4	23.5	59.2	33.6	14.7	30.5	19.8
LATEM [74]	7.3	71.7	13.3	11.5	77.3	20.0	15.2	57.3	24.0	14.7	28.8	19.5
SSE [86]	7.0	80.5	12.9	8.1	82.5	14.8	8.5	46.9	14.4	2.1	36.4	4.0
ESZSL [46]	6.6	75.6	12.1	5.9	77.8	11.0	12.6	63.8	21.0	11.0	27.9	15.8
DEVISE [20]	13.4	68.7	22.4	17.1	74.7	27.8	23.8	53.0	32.8	16.9	27.4	20.9
ALE [2]	16.8	76.1	27.5	14.0	81.8	23.9	23.7	62.8	34.4	21.8	33.1	26.3
SAE [32]	1.8	77.1	3.5	1.1	82.2	2.2	7.8	54.0	13.6	8.8	18.0	11.8
SYNC [9]	8.9	87.3	16.2	10.0	90.5	18.0	11.5	70.9	19.8	7.9	43.3	13.4
CONSE [41]	0.4	**88.6**	0.8	0.5	**90.6**	1.0	1.6	72.2	3.1	6.8	39.9	11.6
SG-GZSL [58]	56.3	67.8	61.5	58.3	68.1	62.8	41.5	53.3	46.7	40.9	30.5	34.9
PSR [5]	-	-	-	20.7	73.8	32.3	24.6	54.3	33.9	20.8	37.2	26.7
Cauchy-Ort [84]	18.3	79.3	29.8	17.6	80.9	29.0	19.9	52.5	28.9	19.8	29.1	23.6
CVAE-ZSL [40]	-	-	47.2	-	-	51.2	-	-	34.5	-	-	26.7
FG [75]	**57.9**	61.4	59.6	-	-	-	**43.7**	57.7	49.7	42.6	**36.6**	**39.4**
OT-ZSL (w/o f)	53.6	68.7	60.2	55.7	69.9	62.0	38.2	59.1	46.4	40.1	34.2	36.9
OT-ZSL	57.2	71.2	**63.5**	**59.7**	72.2	**65.4**	42.9	**61.2**	**50.4**	**43.2**	35.0	38.7

Table 3. Top-1 accuracy in the standard ZSL problem.

Methods	AwA1		AwA2		CUB		SUN	
	S	P	S	P	S	P	S	P
SJE [3]	76.7	65.6	69.5	61.9	55.3	53.9	57.1	53.7
LATEM [74]	74.8	55.1	68.7	55.8	49.4	49.3	56.9	55.3
SSE [86]	68.8	60.1	67.5	61.0	43.7	43.9	54.5	51.5
ESZSL [46]	74.7	58.2	75.6	58.6	55.1	53.9	57.3	54.5
DEVISE [20]	72.9	54.2	68.6	59.7	53.2	52.0	57.5	56.5
ALE [2]	78.6	59.9	80.3	62.5	53.2	54.9	59.1	58.1
SAE [32]	80.6	53.0	80.2	54.1	33.4	33.3	42.4	40.3
SYNC [9]	72.2	54.0	71.2	46.6	54.1	55.6	59.1	56.3
CONSE [41]	63.6	45.6	67.9	44.5	36.7	34.3	44.2	38.8
CVAE-ZSL [40]	-	**71.4**	-	65.8	-	52.1	-	61.7
OT-ZSL (w/o f)	79.4	68.3	79.8	65.1	59.2	55.5	61.6	61.2
OT-ZSL	**84.0**	70.1	**81.5**	**68.8**	**59.7**	**58.6**	**64.3**	**62.9**

"S" represents the standard data split most widely used for each dataset.

"P" is the split provided by [76].

the other hand, the attribute-based regularizer is necessary to further boost the classification accuracy, which brings a non-trivial gain ($2\% \sim 4\%$) for both A_s and A_u.

6.3. Standard zero-shot learning

For the standard zero-shot learning setting, we synthesize samples only from the unseen classes and train the linear softmax classifier accordingly. In the testing phase, this classifier will be used to test the samples from the unseen classes. In terms of the evaluation, the average per-class top-1 accuracy is reported. Experimental results, comparing with existing ZSL approaches under two kinds of data split strategies, are listed in Table 3. We can see that the gain of our model is consistent across all the four benchmark datasets. Similar to GZSL, when training without the attribute-based regularizer, the performance of our method decays slightly, but still beats most of its competitors.

A large-scale experiment over ImageNet is performed as well, and the average per-class top-5 accuracy for various methods is shown in Table 4. The improvement on ImageNet further demonstrates that the superiority of our method is consistent over scales.

Table 4. Top-5 accuracy in the standard ZSL problem of ImageNet.

Methods	Pretrained CNN	Top-5 A_u
DEVISE [20]	GoogleNet	12.8
CONSE [41]	GoogleNet	15.5
SS-VOC [22]	VGG	16.8
VZSL [70]	VGG	23.1
CVAE-ZSL [40]	GoogleNet	24.7
SG-GZSL [58]	GoogleNet	25.4
OT-ZSL	GoogleNet	**27.2**

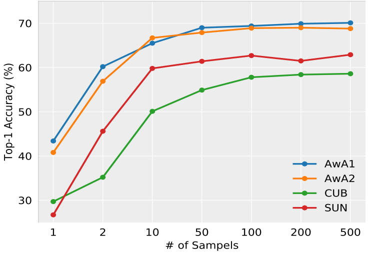

Figure 2. Top-1 Accuracy on the unseen classes with varying number of generated features in standard ZSL.

Additionally, we investigate the sample efficiency of our method in Figure 2. For each dataset, we find that the proposed method can achieve encouraging classification accuracy even if we only generate 100 synthetic features per class to train the classifier.

6.4. Transductive zero-shot learning

In the transductive setting, we use the standard data split for AwA1, AwA2 and CUB. For SUN, we follow the 707/10 data split provided by [32]. To yield a fair comparison rela-

Table 5. Comparisons on A_u in the transductive ZSL problem

Methods	Pretrained CNN	AwA1	AwA2	CUB	SUN
DSRL [82]	VGG	87.2	-	57.1	85.4
SSZSL [52]	VGG	88.6	-	49.9	86.2
SP-ZSR [87]	VGG	92.1	-	55.3	**89.5**
VZSL [70]	VGG	94.8	-	66.5	87.8
EF-ZSL [59]	VGG	85.2	80.8	60.3	64.5
BiDiLEL [65]	VGG/GoogleNet	95.0	-	62.8	-
OT-ZSL (w/o f)	VGG	93.9	94.3	66.8	84.2
OT-ZSL	VGG	**95.8**	**95.0**	67.8	88.1
OT-ZSL (w/o f)	ResNet	93.5	93.6	67.2	85.8
OT-ZSL	ResNet	95.6	94.5	**68.8**	88.7

tive to previous work, VGG [53] features are also included in this section. The training phase of Transductive ZSL is similar to standard ZSL but with additional access to the unseen classes data, without paired label (attribute) information. We can directly use such unlabeled data in our optimal transport framework, *i.e.*, slightly changing Algorithm 1 via sampling B_{real} from $\mathcal{D}_s \cup \mathcal{D}_u$ (line 4) and calculating optimal transport distance based on B_{real}, B_s and B_u (line 9). The testing phase is the same as the standard ZSL setting.

Table 5 reports results for the transductive setting, with comparison to multiple state-of-the-art baselines. We observe that the proposed method again outperforms the other methods with a non-trivial gain — on average, about 20% improvement is achieved with the access of the unlabeled data. Empirically, VGG [53] features work slightly better than ResNet [27] feature on AwA1 and AwA2. In the optimal-transport framework, the information of the unlabeled data is leveraged effectively, which can significantly improve the classification results. Again, we observe that the attribute-based regularizer helps to improve performance. However, because in the transductive setting we can access the unlabeled data in the training phase, the proposed optimal transport framework can find a mapping between the data and their potential attributes, even if the regularizer is not imposed. As a result, improvements from the regularizer in TZSL are not as significant as those in GZSL/SZSL.

6.5. IPOT vs Sinkhorn

We conduct a study on the appropriateness of using IPOT [78] for our framework. The results are found in Figure 3. On both the AwA2 and CUB datasets, the IPOT algorithm is able to converge much faster and achieve lower losses on the validation sets. One possible reason for these observations is that IPOT is insensitive to the choice of the regularizer's weight. Therefore, IPOT is a more rational choice for our framework.

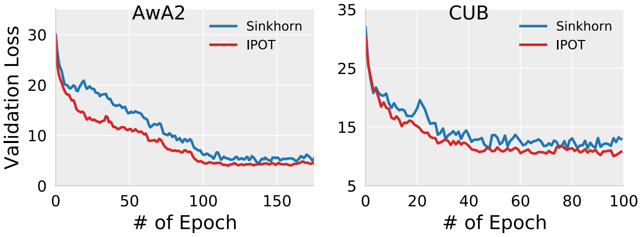

Figure 3. Measuring the validation loss w.r.t. training epoches. $\lambda = 0.5$ is used for IPOT and $\lambda = 0.1$, the most stable choice, for Sinkhorn iteration.

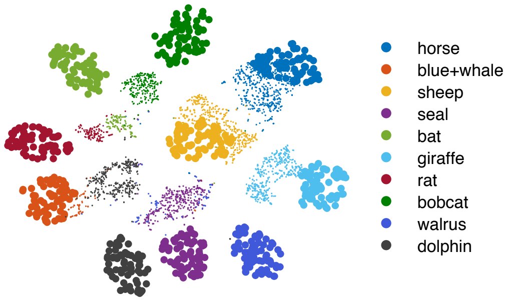

Figure 4. t-SNE visualization of the real image features and the synthetic features for the 10 unseen classes of AwA2. The tiny dot points stand for the real image features while the big circles represent the synthetic features from our generative model.

6.6. Visualization

To further understand our method, for the unseen classes in the AwA2 dataset we take their real features from the pretrained CNN and their synthetic features from our generator, and embed them into a 2D space using t-SNE [38], as shown in Figure 4. The distribution of synthetic features is consistent with that of real features on the clustering structure of classes – for each class its synthetic features are generally close to the real features within the same class. For example, the synthetic features on the class "horse" and "sheep" are well-mixed with the corresponding real features, demonstrating the generalization power of our model. However, it should be noted that the proposed model may make mistakes in some cases, because of the natural similarity between objects, *e.g.*, the generated feature of "blue whale" may look more like the real feature of "dolphin."

7. Conclusions

An optimal transport framework is proposed to address zero-shot learning. In this framework, a conditional generator is learned to map attributes to features. The learning of the generator is driven by minimizing the optimal-transport distance between the distribution of generated features and that of real ones. An attribute predictor is trained simultaneously as the generator's regularizer, encouraging the clustering structure of the generator's outputs. The proposed framework was developed with a focus on generalized zero-shot learning; however, we also demonstrated that it can be readily extended to standard and transductive zero-shot learning. The proposed approach outperforms state-of-the-art methods in all the three ZSL problems, on various datasets.

References

[1] Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *CVPR*, pages 819–826. IEEE, 2013.

[3] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015.

[4] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *arXiv preprint arXiv:1705.09634*, 2017.

[5] Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. In *CVPR*, pages 7603–7612, 2018.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[7] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[8] Emmanuel Boissard, Thibaut Le Gouic, Jean-Michel Loubes, et al. Distributions template estimate with Wasserstein metrics. *Bernoulli*, 21(2):740–759, 2015.

[9] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336, 2016.

[10] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, pages 52–68. Springer, 2016.

[11] Liqun Chen, Shuyang Dai, Yunchen Pu, Erjin Zhou, Chunyuan Li, Qinliang Su, Changyou Chen, and Lawrence Carin. Symmetric variational autoencoder and connections to adversarial learning. In *AISTATS*, pages 661–669, 2018.

[12] Liqun Chen, Guoyin Wang, Chenyang Tao, Dinghan Shen, Pengyu Cheng, Xinyuan Zhang, Wenlin Wang, Yizhe Zhang, and Lawrence Carin. Improving textual network embedding with global attention via optimal transport. *arXiv preprint arXiv:1906.01840*, 2019.

[13] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, pages 1043–1052, 2018.

[14] Nicolas Courty, Rémi Flamary, and Mélanie Ducoffe. Learning Wasserstein embeddings. *arXiv preprint arXiv:1710.07457*, 2017.

[15] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, pages 2292–2300, 2013.

[16] Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *ICML*, pages 685–693, 2014.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[18] Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

[19] Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[20] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.

[21] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *TPAMI*, 37(11):2332–2345, 2015.

[22] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, pages 5337–5346, 2016.

[23] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Sinkhorn-AutoDiff: Tractable Wasserstein learning of generative models. *arXiv preprint arXiv:1706.00292*, 2017.

[24] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In *NIPS*, pages 513–520, 2005.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[26] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[28] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*, pages 5081–5090, 2016.

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

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[31] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015.

[32] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.

[33] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014.

[34] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1576–1585, 2018.

[35] Xin Li, Yuhong Guo, and Dale Schuurmans. Semi-supervised zero-shot classification with label representation learning. In *CVPR*, pages 4211–4219, 2015.

[36] Yan Li, Junge Zhang, Jianguo Zhang, and Kaiqi Huang. Discriminative learning of latent features for zero-shot recognition. In *CVPR*, pages 7463–7471, 2018.

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

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.

[39] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[40] Ashish Mishra, M Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. *arXiv preprint arXiv:1709.00663*, 2017.

[41] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[42] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *CVPR*, pages 460–467. IEEE, 2009.

[43] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. Technical report, 2017.

[44] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[45] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *NIPS*, pages 46–54, 2013.

[46] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, pages 2152–2161, 2015.

[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[48] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.

[49] Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngole, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

[50] Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Lawrence Carin, and Ricardo Henao. Nash: Toward end-to-end neural architecture for generative semantic hashing. *arXiv preprint arXiv:1805.05361*, 2018.

[51] Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*, 2018.

[52] Seyed Mohsen Shojaee and Mahdieh Soleymani Baghshah. Semi-supervised zero-shot learning by a clustering-based approach. *arXiv preprint arXiv:1605.09016*, 2016.

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

[54] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[55] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[57] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. *arXiv preprint arXiv:1703.05908*, 2017.

[58] V Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, 2018.

[59] Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 792–808. Springer, 2017.

[60] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[61] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11(Dec):3371–3408, 2010.

[62] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.

[63] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[64] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*, 2018.

[65] Qian Wang and Ke Chen. Zero-shot visual recognition via bidirectional latent embedding. *IJCV*, 124(3):356–383, 2017.

[66] Wenlin Wang, Changyou Chen, Wenlin Chen, Piyush Rai, and Lawrence Carin. Deep metric learning with data summarization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 777–794. Springer, 2016.

[67] Wenlin Wang, Changyou Chen, Wenqi Wang, Piyush Rai, and Lawrence Carin. Earliness-aware deep convolutional networks for early time series classification. *arXiv preprint arXiv:1611.04578*, 2016.

[68] Wenlin Wang, Zhe Gan, Wenqi Wang, Dinghan Shen, Jiaji Huang, Wei Ping, Sanjeev Satheesh, and Lawrence Carin.

Topic compositional neural language model. *arXiv preprint arXiv:1712.09783*, 2017.

[69] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*, 2019.

[70] Wenlin Wang, Yunchen Pu, Vinay Kumar Verma, Kai Fan, Yizhe Zhang, Changyou Chen, Piyush Rai, and Lawrence Carin. Zero-shot learning via class-conditioned deep generative models. *arXiv preprint arXiv:1711.05820*, 2017.

[71] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9329–9338, 2018.

[72] Wenlin Wang, Chenyang Tao, Zhe Gan, Guoyin Wang, Liqun Chen, Xinyuan Zhang, Ruiyi Zhang, Qian Yang, Ricardo Henao, and Lawrence Carin. Improving textual network learning with variational homophilic embeddings. *arXiv preprint arXiv:1909.13456*, 2019.

[73] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *CVPR*, pages 6857–6866, 2018.

[74] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, pages 69–77, 2016.

[75] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018.

[76] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. *arXiv preprint arXiv:1703.04394*, 2017.

[77] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010.

[78] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for wasserstein distance. *arXiv preprint arXiv:1802.04307*, 2018.

[79] Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. Distilled wasserstein learning for word embedding and topic modeling. In *Advances in Neural Information Processing Systems*, pages 1716–1725, 2018.

[80] Qian Yang, Zhouyuan Huo, Dinghan Shen, Yong Cheng, Wenlin Wang, Guoyin Wang, and Lawrence Carin. An end-to-end generative architecture for paraphrase generation. EMNLP, 2019.

[81] Jianbo Ye, Panruo Wu, James Z Wang, and Jia Li. Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *TSP*, 65(9):2317–2332, 2017.

[82] Meng Ye and Yuhong Guo. Zero-shot classification with discriminative semantic representation learning. In *CVPR*, 2017.

[83] Yoav Zemel and Victor M Panaretos. Fréchet means and Procrustes analysis in Wasserstein space. *arXiv preprint arXiv:1701.06876*, 2017.

[84] Hongguang Zhang and Piotr Koniusz. Zero-shot kernel learning. In *CVPR*, pages 7670–7679, 2018.

[85] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015.

[86] Ziming Zhang and Venkatesh Saligrama. Learning joint feature adaptation for zero-shot recognition. *arXiv preprint arXiv:1611.07593*, 2016.

[87] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, pages 6034–6042, 2016.

A. IPOT

A detailed description of the IPOT algorithm used in our framework is summarized in Algorithm 2, where “$\odot$” is the Hadamard product and “$\div$” represents element-wise division. It is notable that this method works well with batch-based optimization, $i.e.$, N in Algorithm 2 can represent either the size of the whole dataset or the size of a batch.

Specifically, compared with the Sinkhorn iteration algorithm, IPOT *changes* the objective function by adding an entropy regularizer on $\boldsymbol{T}$. Although such a modification converts the optimal transport problem to be strictly convex, its success is highly dependent on the choice of regularizer weight. On one hand, if the weight is too large, Sinkhorn iteration only obtains an over-smoothed $\boldsymbol{T}$ with a large number of iterations. On the other hand, if the weight is too small, Sinkhorn iteration suffers from numerical-stability issues. IPOT, in contrast, solves the original optimal transport problem. In particular, the regularizer in (7) just controls the learning process and its weight λ mainly affects the convergence rate. If we reduce its weight λ with respect to the number of iterations, the final result $\boldsymbol{T}^*$ will be equal to that obtained by solving (3) directly. Additionally, because the weight λ mainly affects convergence rate, we can choose it in a wide range to achieve better numerical stability than Sinkhorn iteration.

Algorithm 2: IPOT Algorithm

```
Input: Real features $\{\boldsymbol{x}_n\}_{n=1}^{N}$, generated features
    $\{\hat{\boldsymbol{x}}_m\}_{m=1}^{M}$, $\lambda = 0.5$, $\boldsymbol{\mu} = [\frac{1}{N}]$, $\boldsymbol{v} = [\frac{1}{M}]$.
Output: Optimal transport $\boldsymbol{T}^*$
Calculate $\boldsymbol{C} = [C_{nm}]$, with $C_{nm} = 1 - \frac{\boldsymbol{x}_n^\top \hat{\boldsymbol{x}}_m}{\|\boldsymbol{x}_n\|_2 \|\hat{\boldsymbol{x}}_m\|_2}$.
$\boldsymbol{G} = \exp(-\frac{\boldsymbol{C}}{\lambda})$.
Initialize $\boldsymbol{a} = \boldsymbol{\mu}$, $\boldsymbol{T}^{(1)} = \boldsymbol{\mu}\boldsymbol{v}^\top$
for $t = 1, ..., T$ do
    $\boldsymbol{K} = \boldsymbol{G} \odot \boldsymbol{T}^{(t)}$
    Sinkhorn-Knopp Algorithm:
    for $j = 1, ..., J$ do
        $\boldsymbol{b} = \frac{\boldsymbol{v}}{\boldsymbol{K}^\top \boldsymbol{a}}$ and $\boldsymbol{a} = \frac{\boldsymbol{\mu}}{\boldsymbol{K}\boldsymbol{b}}$.
    end for
    $\boldsymbol{T}^{(t+1)} = \mathrm{diag}(\boldsymbol{a})\boldsymbol{K}\mathrm{diag}(\boldsymbol{b})$
end for
$\boldsymbol{T}^* = \boldsymbol{T}^{(T+1)}$
```