Two-Level Adversarial Visual-Semantic Coupling for Generalized Zero-shot Learning

Shivam Chandhok
Indian Institute of Technology
Hyderabad,India

Vineeth N Balasubramanian
Indian Institute of Technology
Hyderabad,India

Abstract

The performance of generative zero-shot methods mainly depends on the quality of generated features and how well the model facilitates knowledge transfer between visual and semantic domains. The quality of generated features is a direct consequence of the ability of the model to capture the several modes of the underlying data distribution. To address these issues, we propose a new two-level joint maximization idea to augment the generative network with an inference network during training which helps our model capture the several modes of the data and generate features that better represent the underlying data distribution. This provides strong cross-modal interaction for effective transfer of knowledge between visual and semantic domains. Furthermore, existing methods train the zero-shot classifier either on generated synthetic image features or latent embeddings produced by leveraging representation learning. In this work, we unify these paradigms into a single model which, in addition to synthesizing image features, also utilizes the representation learning capabilities of the inference network to provide discriminative features for the final zero-shot recognition task. We evaluate our approach on four benchmark datasets i.e. CUB, FLO, AWA1 and AWA2 against several state-of-the-art methods, and show its performance. We also perform ablation studies to analyze and understand our method more carefully for the Generalized Zero-shot Learning task.

1. Introduction

Practical settings require recognition models to have the ability to learn from few labeled samples and be extended to novel classes where data annotation is infeasible. However, deep learning models do not directly adapt to such settings due to their reliance on large amounts of labeled data during training. On the other hand, humans perform well under such conditions due to their capability to transfer semantics and recast information from high-level descriptions to the visual space, enabling them to recognize objects that they have never seen before. Zero-shot learning (ZSL) aims to bridge this gap by providing recognition models with the capability to classify images of novel classes that have not been seen during the training phase. The model is typically given access to semantic description of the novel unseen classes during training (such as embeddings of attributes of the classes) and is expected to recognize unseen class images by knowledge transfer between visual and semantic domains.

Based on the classes that a model sees in the test phase, the ZSL problem is generally categorized into two settings: *conventional* and *generalized zero-shot*. In conventional ZSL, the image features to be recognized at test time belong only to unseen classes. In the generalized ZSL (GZSL) setting, the images at test time may belong to both seen or unseen classes. The GSZL setting is practically more useful and challenging when compared to the conventional setting, since the assumption that images at test time come only from unseen classes need not hold. We aim to address the generalized zero-shot learning problem in this work (although we also show the applicability of our method to conventional ZSL).

A potential approach to address GZSL is to utilize generative models to generate features for unseen classes and reduce the zero-shot problem to a supervised learning problem [21, 29, 15, 8, 30]. Most existing methods in this direction use a unidirectional mapping by generating visual features conditioned on semantic attributes. However, it has been shown that such methods that rely on unidirectional mapping lose out a tight visual-semantic coupling which is crucial for zero-shot recognition [11, 16]. To address this issue, more recent approaches such as [11, 16] have proposed to use bidirectional mapping between visual and semantic domains to enhance zero-shot recognition performance.

In this work, we propose a holistic unified framework that uses a **t**wo-level **a**dversarial visual-semantic **co**upling (and hence named TACO) approach for bidirectional mapping that provides a tight coupling between semantic and visual spaces, and is also expressive enough to capture the complex distributions of the underlying data. Figure 1 summarizes our overall framework. Our key contributions are
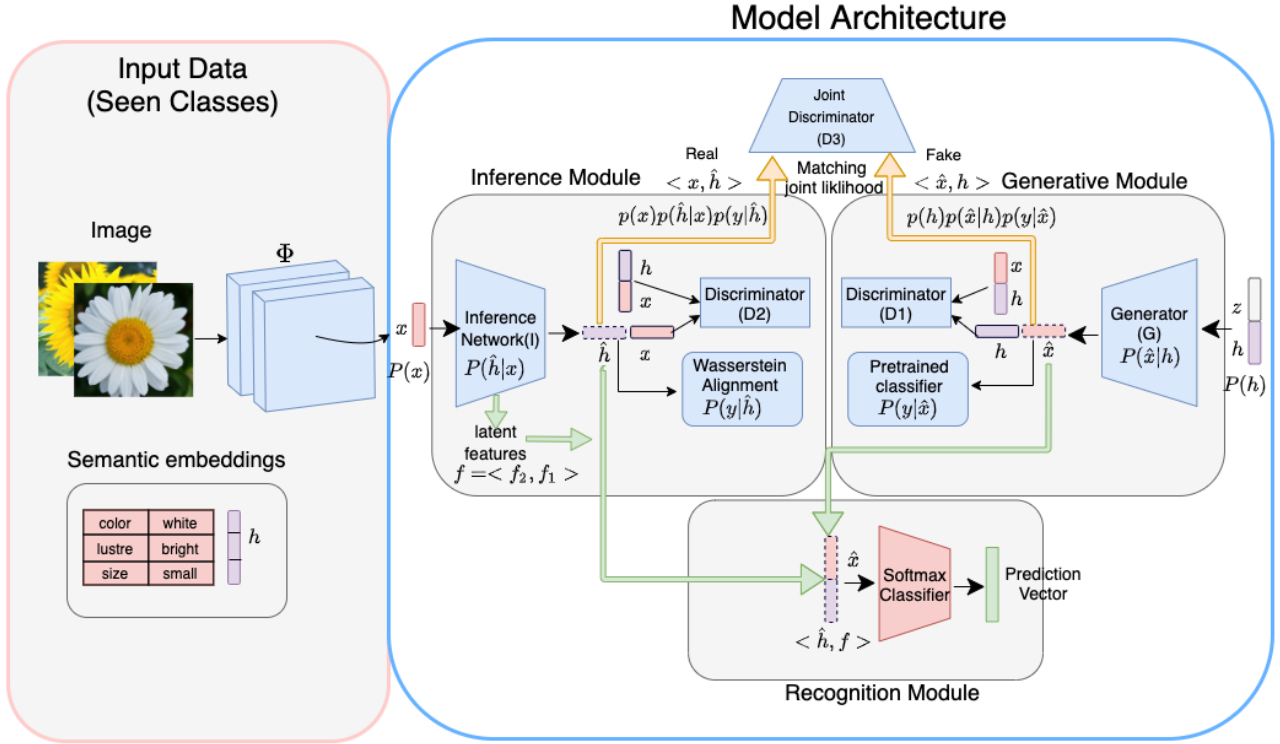
Figure 1: Network architecture for our proposed methodology. The proposed pipeline consists of a Generative module, Inference module, Recognition module and a Joint Discriminator. The model is trained on seen class visual features and semantic attributes. The feature extractor backbone network Φ is used to extract visual features from images. The vectors generated by our model are shown with a dotted outline. The final softmax classifier is trained on synthesized features $\hat{x}$ and representations from the inference network($< \hat{h}, f >$) as shown.

as follows:

(1) Unlike most existing methods that use only a generative module, we augment the generative network with an inference network, and train these two components of our TACO framework together to maximize the joint likelihood of visual and semantic features. Learning the inference network jointly with the generative model helps us capture the underlying modes of the data distribution better [7].

(2) We use a two-level adversarial training strategy to train TACO, where we train both generative and inference modules through respective discriminators. We also use an adversarial joint-maximization loss as an additional supervisory signal to enhance the visual-semantic coupling and facilitate better cross-domain information transfer. This helps our model outperform other bidirectional mapping methods which lack such a mechanism.

(3) We use a novel Wasserstein semantic alignment loss that helps us model the joint distribution of visual and semantic features better, and ensures that the generated semantic features are distributionally aligned with real semantic features, which in turn helps lower loss in the semantic space.

(4) Furthermore, we use the discriminative information in latent layers of the inference network to train our final recognition model. This helps provide the final recognition module with representations from both generative and inference modules, and thus enhances performance when compared to earlier approaches that use only the synthesized image features in the final recognition model.

(5) We perform detailed experimental studies and analysis on Caltech-UCSD Birds (CUB), Oxford Flowers (FLO) and Animals and Attributes (AWA1 and AWA2) datasets. We demonstrate that the proposed method helps in better visual-semantic coupling, and thus obtains state-of-the art performance, outperforming other methods on both fine-grained as well as coarse-grained datasets.

To the best of our knowledge, this is the first effort to employ a joint maximization step in adversarial training to provide deeper visual-semantic coupling for solving GZSL. In addition, the new idea of using an adversarially learned discriminative representation from the latent layers of inference network, along with the generated features from the generator to train the final zero-shot recognition model, significantly improves GZSL performance. The use of a Wasserstein alignment loss to preserve semantics is also the first of its kind to be used in a generative approach to GZSL.

2. Related Work

As stated in Section 1, existing work in ZSL can be broadly divided into work on conventional ZSL and work on generalized ZSL (GZSL). This work focuses on the more challenging GZSL setting, and we focus on presenting related literature in GZSL in this section.

There has been a recent increase in efforts in the field of zero-shot learning with the aim of boosting GZSL performance. The methods proposed so far can be broadly categorized into approaches that learn a projection function based on seen class image features [1, 9, 20, 5, 27, 2, 32, 10, 14, 22] or generative network-based methods which aim to synthesize unseen class features reducing GZSL to a standard supervised problem [29, 21, 11, 16, 23, 12, 31, 8, 30, 24, 4]. We focus on describing recent related methods in more detail in the rest of this section. The authors in [22] first proposed to leverage multi-modal learning by learning a joint embedding of image and textual features for GZSL. They utilized a common representation learning along with cross-domain alignment to map and align visual and semantic features in a common latent space. To alleviate the bias problem, generative methods for GZSL have been proposed. These methods generally combine adversarial loss and classification loss to generate discriminative features for unseen classes conditioned on semantic attributes. Methods like f-clsWGAN [29], CVAE [15], [8] used conditional Generative Adversarial Networks (GANs) or Variational Autoencoders (VAE) for generation of unseen class features. [21] tried combining multi-modal learning with generative GZSL approaches and learned a cross-aligned multi-modal VAE to generate latent features for unseen classes and later trained a softmax classifier on latents from all classes. More recently, [30] proposed to combine the strengths of VAEs and GANs by using the decoder of VAE as a generator. On the other hand, GDAN [11] and DASCN [16] formulates a dual learning framework that uses bidirectional mapping between visual and semantic spaces, and trained the model with adversarial loss and cyclic consistency. All of these efforts showed the need to enforce stronger coupling between the visual domain (images) and the semantic domain (image attributes provided for seen and unseen classes) in different ways. Our work is closest to GDAN [11] and DASCN [16] in this regard, and also comes in the category of methods that learn a bidirectional mapping. However, there many differences as described below. Importantly, our approach unifies ideas from existing approaches.

In DASCN [16], in the formulation of dual GAN, the visual to semantic mapping network never sees real image features and only has access to the features generated by the primal generator. As pointed out in [6], since the generated image features are practically not as good as actual features, this inhibits the ability of the network to make full use of the dual learning paradigm since the flow of information from visual to semantic domains is partial. On the other hand, GDAN [11] uses a regressor to implement dual learning which maps the generated features back to the attribute space. As pointed out in [16], minimizing L2-norm between generated semantic embeddings and real semantic attributes is weak and unreliable to preserve high-level semantics when using Euclidean distance. Furthermore, in the objective of GDAN, the only way the generative network and regressor interact with each other is via an L2-norm based cyclic loss which does not provide a strong coupling between the visual and semantic domains.

In contrast, in our formulation, adversarial learning is introduced in a two-level fashion, where both generative and inference modules are first adversarially learned (see Figure 1). We subsequently then introduce a new adversarial joint maximization loss which specifically aims to maximize the joint probability of visual and semantic features. This is achieved through a joint discriminator, which has a slightly different formulation from a traditional discriminator (as used in GDAN [11] and DASCN [16]). As pointed in [6], learning a regressor by minimizing reconstruction loss performs poorly when compared to learning an inference network jointly. This helps our model generates features that better represent the underlying distribution of unseen classes. Besides, our design provides our inference network with access to real image features, which facilitates stronger cross-domain coupling with improved representations. Also, our Wasserstein semantic alignment loss enables us to preserve semantics and alleviate semantic loss better than L2 loss. To show the benefits of our method over GDAN and DASCN, we directly compare with them (as well as many other recent methods) on four different GZSL learning benchmark datasets, and show that our method provides the new state-of-the-art for GZSL.

3. Proposed Methodology

We begin our description of the proposed methodology by defining the problem setting and notations, followed by descriptions of each module of our framework.

Problem Setting: Given the dataset $\mathcal{D} = \{D^{Tr}, D^{Ts}\}$, the aim of generalized zero-shot learning is to correctly classify images from both seen and unseen classes during the test phase. Here, $D^{Tr} = \{(\mathbf{x}, y, \mathbf{h}_y) | \mathbf{x} \in \mathcal{X}^s, y \in \mathcal{Y}^s, \mathbf{h}_y \in \mathcal{A}\}$ is the training set, where $\mathbf{x}$ is an image feature, $\mathcal{X}^s$ is the feature space of seen classes, y is the label corresponding to $\mathbf{x}$, $\mathcal{Y}^s$ is the set of labels for seen classes, $\mathbf{h}_y$ is the semantic attribute vector for class y. Similarly, $D^{Ts} = \{(\mathbf{x}, y, \mathbf{h}_y) | \mathbf{x} \in \mathcal{X}^u, y \in \mathcal{Y}^u, \mathbf{h}_y \in \mathcal{A}\}$ is the test set, where $\mathcal{X}^u$ represents the set of image features from unseen classes, $\mathcal{Y}^u$ represents the set of labels of unseen classes such that $\mathcal{Y}^u \cap \mathcal{Y}^s = \emptyset$. Note that image features are typically extracted using a feature extractor backbone Φ as shown in Figure 1. The GZSL task can be formally seen as learning the optimal parameters of a classifier $f : \mathbf{x} \to \mathcal{Y}^u \cup \mathcal{Y}^s$ where x denote the

3101

image features and $\mathcal{Y}^s$, $\mathcal{Y}^u$ denote the set of labels of seen and unseen classes respectively.

Mathematical Framework: GZSL can be formulated as a problem of modeling joint probability $P(\mathbf{x}, \mathbf{h}, y)$ where $\mathbf{x}$, $\mathbf{h}$ and y are as described above. The joint probability $P(\mathbf{x}, \mathbf{h}, y)$ can be factored in two ways:

$$\mathbf{F1} : p(\mathbf{x}, \mathbf{h}, y) = p(\mathbf{x})p(\mathbf{h}|\mathbf{x})p(y|\mathbf{h}) \qquad (1)$$

$$\mathbf{F2} : p(\mathbf{x}, \mathbf{h}, y) = p(\mathbf{h})p(\mathbf{x}|\mathbf{h})p(y|\mathbf{x}) \qquad (2)$$

While most existing work consider one of these factorizations in their approach, we use both the factorizations, and model each of them using separate modules, viz. the *generative module* and *inference module* as shown in Figure 1.

For modeling **F1**, we have the marginal $\Pr(\mathbf{x})$ since we have access to the visual features $\mathbf{x}$ of seen classes. The second term $\Pr(\mathbf{h}|\mathbf{x})$, the conditional probability of $\mathbf{h}$ given the input $\mathbf{x}$, is modeled by an inference network. Thus, Eqn 1 can be written as:

$$\Pr(\mathbf{x}, \mathbf{h}, y) = \Pr(\mathbf{x}) \Pr(\hat{\mathbf{h}}|\mathbf{x}) \Pr(y|\hat{\mathbf{h}}) \qquad (3)$$

where $\hat{\mathbf{h}}$ is the output of the inference module. Now, the term $\Pr(y|\hat{\mathbf{h}})$ can be factorized as:

$$\Pr(y|\hat{\mathbf{h}}) = \Pr(y|\mathbf{h}) \Pr(\mathbf{h}|\hat{\mathbf{h}}) \qquad (4)$$

Here, the factor $\Pr(y|\mathbf{h})$ can be taken as unity, since a known $\mathbf{h}$ has a direct map to the label. We attempt to use a Wasserstein alignment to ensures that the $\hat{\mathbf{h}}$ is close to the original $\mathbf{h}$ in this work. Hence, the joint distribution in factorization **F1** can now be written as:

$$\mathcal{P}_{joint} = \Pr(\mathbf{x}) \Pr(\hat{\mathbf{h}}|x) \Pr(\mathbf{h}|\hat{\mathbf{h}}) \qquad (5)$$

This joint distribution is modeled by the *Inference Module* (see Figure 1 left), which we discuss in detail in Section 3.2.

For modeling **F2**, we are provided with the marginal $\Pr(\mathbf{h})$, since we have access to the attributes $\mathbf{h}$ of seen classes. The second term $\Pr(\mathbf{x}|\mathbf{h})$, the conditional probability of $\mathbf{x}$ given $\mathbf{h}$, is modeled by a generative network. $\Pr(y|\mathbf{x})$ refers to the extra classification constraint that is present in the loss formulation of our generative network (see Eqn 9). Thus, Eqn 2 reduces to:

$$\mathcal{P}_{joint} = \Pr(\mathbf{h}) \Pr(\hat{\mathbf{x}}|\mathbf{h}) \Pr(y|\hat{\mathbf{x}}) \qquad (6)$$

where $\hat{\mathbf{x}}$ is the output of the generative network. This factorization is modeled by the *Generative Module* (see Figure 1 right), which we discuss in detail in Section 3.1.

To provide strong coupling between the generative and inference modules, we also train the generative and inference networks by maximizing the joint probability and matching Eqns 5 and 6, which we describe later in this section. To this end, we introduce a joint discriminator D_3 (see Figure 1 top), whose goal is to match the joint probability by discriminating between a combination of generated and real features i.e $<$*real image feature, generated semantic feature*$>$ and $<$*generated image feature, real semantic feature*$>$ (as shown in Figure 1). This is different from a vanilla discriminator (such as D_1,D_2 in the figure) which only discriminates between generated and real features.

3.1. Generative Module

Given the training data D^{Tr}, the objective of the generative module is to learn a feature generating model conditioned on the semantic attribute vectors, i.e., it should be able to generate discriminative image features that represent the underlying data distribution well. We follow the formulation in [29] for our baseline generative network. The input to generator G and the discriminator D_1 are semantic attributes $\mathbf{h}$ and image features $\mathbf{x}$ of seen classes. The generator G learns a mapping $G(\mathbf{z}, \mathbf{h}) : \mathcal{Z} \times \mathcal{A} \to \hat{\mathcal{X}}$ by taking a random Gaussian noise vector $\mathbf{z}$ concatenated with the attribute vector $\mathbf{h}$ as input and generating image features $\hat{\mathbf{x}}$. We use a Wasserstein GAN [3] for this purpose with the loss given by:

$$\mathcal{L}_{WGAN_1} = \mathbb{E}[D_1(\mathbf{x}, \mathbf{h}(y))] - \mathbb{E}[D_1(\hat{\mathbf{x}}, \mathbf{h}(y))]- \qquad (7)$$
$$\lambda \mathbb{E}[(||\nabla_{\tilde{\mathbf{x}}} D_1(\tilde{\mathbf{x}}, \mathbf{h}(y))||_2 - 1)^2],$$

where $\hat{\mathbf{x}} = G(\mathbf{z}, \mathbf{h}(y))$ is the generated feature, $\tilde{\mathbf{x}} = \alpha \mathbf{x} + (1-\alpha)\hat{\mathbf{x}}$ with $\alpha \sim U(0, 1)$, and λ is a weighting coefficient. In order to ensure that the features generated from the network are discriminative, in addition to the adversarial loss, the generated features are required to minimize the classification loss evaluated over a softmax classifier pretrained on seen class features as in [29]:

$$\mathcal{L}_{CLS} = -\mathbb{E}_{\hat{\mathbf{x}} \sim p_{\hat{\mathbf{x}}}}[\log P(y|\hat{\mathbf{x}}; \theta)] \qquad (8)$$

The final loss for the generative module is then given by:

$$L_{gen} = \min_G \max_D \mathcal{L}_{WGAN_1} + \beta \mathcal{L}_{CLS} \qquad (9)$$

where β is a hyperparameter weighting the classifier.

3.2. Inference Module

It has been shown [7] that augmenting a generative model with an inference network enhances the ability of the model to capture different modes of the data distribution well and generate samples which better represent the underlying distribution. Since the performance of GZSL depends greatly on the quality of synthesized features, we hypothesize that learning an inference network jointly with the generative model will help our proposed TACO model to express the actual data distribution well and generalize to unseen classes better.

The inference network and the discriminator D_2 together form the Inference Module of our TACO framework as shown in Figure 1. The goal of the inference network $I(\mathbf{x}) : \mathcal{X} \to \hat{\mathcal{A}}$ is to learn a mapping from image/visual space to semantic space. The module learns a network that

maps input image features to corresponding semantic attributes i.e. it attempts to infer the semantic attributes that generated the image in the generation module. The discriminator $D_2 : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ outputs a real value. This is also learned through a WGAN, and the loss function is given by:

$$\mathcal{L}_{WGAN_2} = \mathbb{E}[D_2(\mathbf{h}(y), \mathbf{x})] - \mathbb{E}[D_2(\hat{\mathbf{h}}(y), \mathbf{x})] - \quad (10)$$
$$\lambda \mathbb{E}[\left(||\nabla_{\mathbf{h}_{\tilde{(y)}}} D_2(\tilde{\mathbf{h}}(y), \mathbf{x})||_2 - 1 \right)^2],$$

where $\hat{\mathbf{h}} = I(\mathbf{x})$ is the generated semantic attribute, $\tilde{\mathbf{h}} = \alpha \mathbf{h} + (1 - \alpha) \hat{\mathbf{h}}$ with $\alpha \sim U(0,1)$, and λ is a weighting coefficient.

In addition, we use a Wasserstein metric-based alignment loss at the output of the inference network. This aims to ensure that the distribution of class centers of output semantic attributes $\hat{\mathbf{h}}$ aligns with the distribution of ground truth $\mathbf{h}$, enabling us to preserve semantic information better. The overall loss function for this module is then given by:

$$L_{inf} = \min_I \max_D \mathcal{L}_{WGAN_2} + \gamma \mathcal{L}_{wasserstein} \quad (11)$$

where $\mathcal{L}_{wasserstein}$ is computed using the sinkhorn distance as in [25], and γ is a weighting coefficient. More details of this alignment loss term is provided in the Supplementary Section.

3.3. Adversarial Joint Maximization

Existing zero-shot approaches have hitherto not considered the use of a third joint maximization step to improve visual-semantic coupling and cross-domain knowledge transfer. In our proposed TACO framework, we introduce the use of a joint maximization loss as an additional supervisory signal to enhance visual-semantic interaction. We match the joint probability of visual-semantic features using a joint discriminator D_3 as shown in Figure 1. The joint discriminator is formulated as follows:

$$\mathcal{L}_{joint-discriminator} = \mathbb{E}[D_3(\mathbf{x}, \hat{\mathbf{h}}(y))] - \mathbb{E}[D_3(\hat{\mathbf{x}}, \mathbf{h}(y))] -$$
$$\lambda \mathbb{E}[\left(||\nabla_{\tilde{x}} D_3(\tilde{\mathbf{x}}, \mathbf{h}\tilde{(y)}), \nabla_{\tilde{\mathbf{h}}} D_3(\tilde{\mathbf{x}}, \mathbf{h}\tilde{(y)})||_2 - 1 \right)^2] \quad (12)$$

where $\hat{\mathbf{h}} = I(\mathbf{x})$, $\hat{x} = G(\mathbf{z}, \mathbf{h})$ are the generated semantic attributes and image features respectively. $\tilde{\mathbf{x}} = \alpha \mathbf{x} + (1 - \alpha) \hat{\mathbf{x}}$ and $\tilde{\mathbf{h}} = \alpha \hat{\mathbf{h}} + (1 - \alpha) \mathbf{h}$ with $\alpha \sim U(0,1)$, as before, and λ is a weighting coefficient. Note that the joint discriminator is formulated differently form vanilla discriminators D_1, D_2, as mentioned earlier. D_3 aims to discriminate between the pairs $< \mathbf{x}, \hat{\mathbf{h}} >$ and $< \hat{\mathbf{x}}, \mathbf{h} >$ which enables it to match $(p(\mathbf{x})p(\hat{\mathbf{h}}|\mathbf{x}))$ and $(p(\mathbf{h})p(\hat{\mathbf{x}}|\mathbf{h}))$ (also shown in Figure 1 top).

On the other hand, while D_3 aims to optimizes Eqn 12, the generator and inference networks jointly maximize:

$$\mathcal{L}_{joint-max} = -(E[D_3(\mathbf{x}, \hat{\mathbf{h}}(y))] - \mathbb{E}[D_3(\hat{\mathbf{x}}, \mathbf{h}(y))]) \quad (13)$$

Note that the difference between Eqns 13 and 12 is only a regularizer term which is added to improve $D3$. In summary, the generative and inference modules are jointly trained to optimize the final objective, given as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{gen}} + \alpha_1 \mathcal{L}_{\text{inf}} + \alpha_2 \mathcal{L}_{\text{joint-max}} \quad (14)$$

3.4. Recognition module

In previously proposed zero-shot methods, once the generative model is trained, it is used to generated images features for the unseen classes. However, the synthesized image features by themselves might not be discriminative enough to get best GZSL performance. In order to address this issue and obtain highly discriminative features for training the classifier in the recognition module of our TACO framework, we train the final classifier using the adversarially learned representation in the intermediate layers and the output of the inference network.

To this end, we combine seen class image features from D^{Tr} and synthesized features for unseen classes from the trained generator into a set, D. We then pass the image features in D through the pre-trained inference network and get the output $\hat{\mathbf{h}}$, as well as internal intermediate features from the inference network. These are concatenated and used to train the final classifier. For fair comparison with other methods, we use a single layered softmax classifier (as used in most earlier efforts).

At test time, for an input image feature $\mathbf{x}_{test}$, we pass it through the inference network and get the internal feature vectors $\mathbf{f}$ and output $\hat{\mathbf{h}}$. We concatenate these with the test image vector (output of generator module), and pass it to the softmax classifier, given by:

$$f(x) = \arg \max_{y \in \tilde{\mathcal{Y}}} P(y| < \mathbf{x}_{test}, \mathbf{h}, \mathbf{f} >; \Theta') \quad (15)$$

where $\tilde{\mathcal{Y}} = \mathcal{Y}^s \cup \mathcal{Y}^u$ for GZSL, and Θ' are the parameters of our overall architecture.

4. Experiments and Results

In this section, we conduct extensive experiments on four public benchmark datasets under the generalized zero-shot learning setting. We compare our model with several baselines and state-of-the-art methods on four benchmark datasets: **CUB** [26], **FLO** [17], **AWA1** [13] and **AWA2** [13]. Among these datasets, AWA1 and AWA2 are coarse-grained, while FLO and CUB are fine-grained datasets. We follow the standard training/validation/testing splits and evaluation protocols, as in [28]. Following the protocol in [28], we use ResNet101 as the feature extractor backbone network for fair comparison. We denote this backbone by Φ_1 henceforth, for convenience. Recently, [18] proposed to transform the ResNet-101 image features and use a 1024-dimensional intermediate representation as input features to overcome hubness and preserve

Dataset	CUB			FLO			AWA1			AWA2		
Methods	U	S	H	U	S	H	U	S	H	U	S	H
DEM(CVPR'17)[32]	19. 6	57. 9	29. 2	-	-	-	32. 8	**84. 7**	47. 3	30. 5	**86. 4**	45. 1
ZSKL(CVPR'18)[10]	21. 6	52. 8	30. 6	-	-	-	18. 3	79. 3	29. 8	18. 9	82. 7	30. 8
DCN(NIPS'18)[14]	28. 4	60. 7	38. 7	-	-	-	-	-	-	25. 5	**84. 2**	39. 1
ALE(TPAMI'13)[1]	23. 7	62. 8	34. 4	13. 3	61. 6	21. 9	16. 8	76. 1	27. 5	**81. 8**	14. 0	23. 9
DEVISE(NIPS'13)[9]	23. 8	53. 0	32. 8	9. 9	44. 2	16. 2	13. 4	68. 7	22. 4	74. 7	17. 1	27. 8
ESZSL(ICML'15)[20]	12. 6	63. 8	21. 0	11. 4	56. 8	19. 0	6. 6	75. 6	12. 1	77. 8	5. 9	11. 0
SYNC(CVPR'16)[5]	11. 5	**70. 9**	19. 8	-	-	-	8. 9	**87. 3**	16. 2	**90. 5**	10. 0	18. 0
LATEM(CVPR'16)[27]	15. 2	57. 3	24. 0	-	-	-	7. 3	71. 7	13. 3	77. 3	11. 5	20. 0
SJE(CVPR'15)[2]	23. 5	59. 2	33. 6	13. 9	47. 6	21. 5	**74. 6**	11. 3	19. 6	73. 9	8. 0	14. 4
CLSWGAN(CVPR'18)[29]	43. 7*	57. 7*	49. 7*	59. 0	73. 8	**65. 6**	-	-	-	57. 9	61. 4	59. 6
CADA-VAE(CVPR'19)[21]	53. 5	51. 6	52. 4*	-	-	-	**72. 8**	57. 3	**64. 1**	75. 0	55. 8	63. 9
VSE(CVPR'19)[19]	39.5*	68.9*	50.2*	-	-	-	-	-	-	45. 6	88. 7	60. 2
GZLOCD(CVPR'20)[12]	44. 8*	59. 9*	51. 3*	-	-	-	-	-	-	59. 5	73. 4	65. 7
GDAN(NIPS'19)[11]	39. 3*	66. 7*	49. 5*	-	-	-	-	-	-	32. 1	67. 5	43. 5
DASCN(NIPS'19)[16]	45. 9*	59. 0*	51. 6*	-	-	-	59. 3	68. 0	63. 4	-	-	-
SGAL(NIPS'19)[31]	40. 9*	55. 3*	47. 0*	-	-	-	52. 7	75. 7	62. 2	55. 1	81. 2	65. 6
SE-GZSL(CVPR'18)[23]	41. 5	53. 3	46. 7	-	-	-	56. 3	67. 8	61. 5	58. 3	68. 1	62. 8
CycWGAN(ECCV'18)[8]	47. 9	59. 3	53. 0	61. 6	69. 2	65. 2	59. 6	63. 4	59. 8	59. 6	63. 4	59. 8
f-VAEGAN(CVPR'19)[30]	48. 4	60. 1	53. 6	56. 8	**74. 9**	64. 6	-	-	-	57. 6	70. 6	63. 5
ZSML(AAAI'20)[24]	60. 0	52. 1	55. 7	-	-	-	57. 4	71.1	63.5	58. 9	74. 6	65. 8
TACO-GZSL	**61.2**	57.7	**59.4**	60. 6	81. 1	**69. 4**	60.5	71,9,	**65.7**	59,4,	74,2,	**66.0**
TACO-GZSL(312)	51.8*	60. 0*	55. 6*	60. 6	81. 1	**69. 4**	60.5	71,9,	**65.7**	59,4,	74,2,	**66.0**
TACO-GZSL(using Φ_2)	64.7	65.9	65.35	-	-	-	-	-,	-	62.6	75.6	**68.5**
TACO-GZSL(312)(using Φ_2)	55.6*	**67.50***	**61.0***	-	-	-	-	-,	-	62.6	75.6	**68.5**

Table 1: GZSL performance comparison with several baseline and state-of-the-art methods. For fair comparison, all results reported here are *without fine-tuning* the backbone ResNet101 feature extractor. We measure Top-1 accuracy on Unseen(U), Seen(S) classes and their Harmonic mean(H). Best results are highlighted in bold. * indicates result on CUB dataset with only 312 dim attributes (included for fair comparison with other work that use this setting)

semantic relations. To show the generalizability of our approach, we also evaluate our model on features provided by [18]. The feature extractor network corresponding to these 1024-dimensional features is denoted as Φ_2, henceforth. We use the class attributes/embeddings provided in [28] for each dataset, which represents the **h**(.) in our approach (Fig 1). For CUB and FLO datasets, we use additional 1024-dimensional character-based CNN-RNN features as in [30][4], unless explicitly stated otherwise. We use the average-per-class top 1 accuracy metric to evaluate our model on a set of classes, as generally followed [28]. In order to evaluate and compare our model in the GZSL setting, we report the harmonic mean of our model performance on both seen and unseen classes[28]. For fair comparison, we follow the architecture used in [30] for all our components. Due to space constraints, we provide the details of each component in the supplementary material.

Results: Table 1 shows the performance comparison of our model with multiple baselines and state-of-the-art methods in GZSL. The table is divided into two sections, which show the performance of non-generative (top) and generative GZSL approaches (bottom) respectively. For fair comparison, all results reported are without fine-tuning the backbone ResNet-101 network. It can be clearly seen that our methodology consistently outperforms other approaches across all four datasets. Note that GZSL is a challenging problem and most existing methods have not been able to maintain consistently high performance on both fine-grained and coarse-grained datasets. Thanks to the joint-maximization loss and Wasserstein alignment which enables our model to facilitate better cross-domain coupling and learn a useful discriminative representation, our method is able to outperform even other approaches which utilize bidirectional mapping i.e [11, 16] across all datasets with varying granularity. We also note that our method consistently outperforms approaches like [8, 11, 16] which use a cyclic consistency to model visual-semantic interaction.

In Table 1, we also show the results for our method using Φ_2 as the feature extractor backbone for CUB (fine-grained) and AWA2 (coarse-grained) datasets. We see that GZSL performance increases from 59.4% to 65.35% in CUB, and 66.0% to 68.5% in AWA2. This shows that our model generalizes well to different feature extractor backbones and is not specific to learning from only ResNet-101 features.

Visualizing Generated Features: To visualize the unseen class image features generated by our method, we sample 200 synthesized feature vectors for each unseen class for the FLO dataset and plot them using t-SNE as shown in Fig 3. In addition, we also show the mean of all synthesized features (represented by ■) and the mean of real unseen class features (represented by ×). It is evident that for most classes, the center/mean of generated synthetic features co-
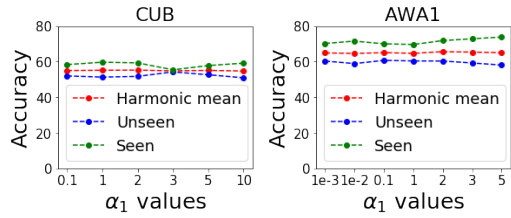
(a) Study of Seen class accuracy, Unseen class accuracy and Harmonic mean by varying α_1 for fine- and coarse-grained datasets
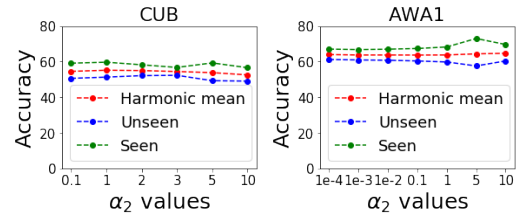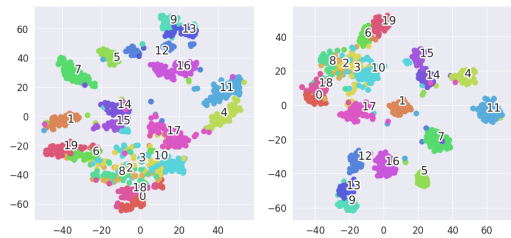

(b) Study of Seen class accuracy, Unseen class accuracy and Harmonic mean by varying α_2 for fine- and coarse-grained datasets
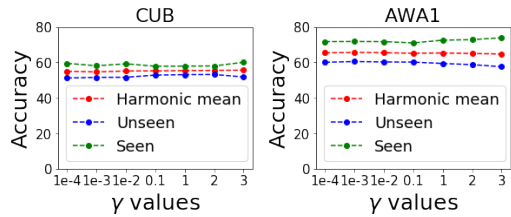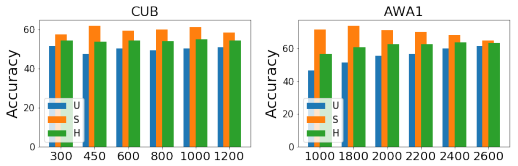

(c) Study of Seen class accuracy, Unseen class accuracy and Harmonic mean by varying γ for fine- and coarse-grained datasets
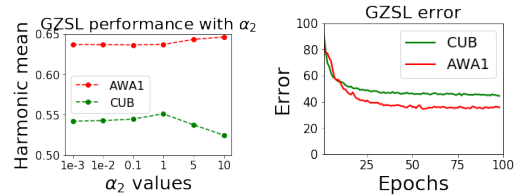
(d) *(Left)* Harmonic mean accuracy for varying α_2 (plotted on smaller scale for clarity); *(Right)* GZSL error trajectory over epochs for proposed method

(e) t-SNE visualisations of unseen class features on FLO dataset, which are input to softmax layer of zero-shot classifier (recognition module): *(Left)* f-VAEGAN; *(Right)* Ours

(f) Variation in GZSL performance (S=seen class accuracy; U=unseen class accuracy; H=harmonic mean) with number of synthesised features for unseen classes

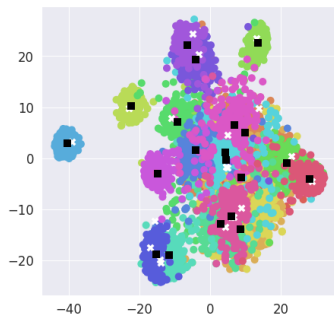Figure 2: Ablation Studies and Analysis

Figure 3: t-SNE visualization of synthesized image features for unseen classes for FLO dataset. ■ = mean centers of synthesized features; × (in white) = centres of actual unseen class features *(Best viewed in color, when zoomed in)*

incides or is very close to the center of actual unseen class image features. This verifies that our model captures the modes of the underlying distribution well. Furthermore, it can be seen that the features of unseen classes form distinct clusters for most cases which shows the discriminative ability of the generated features.

We note that both at training and test times, our method has time complexity comparable to any other recent GZSL method, and no additional overheads.

5. Ablation Studies and Analysis

We show several ablation studies to show the usefulness of different components as well as the sensitivity of our method to hyperparameter choices in this section. Similar to [21, 11, 31, 12], we use 312-dimensional semantic attributes for our ablation studies on CUB in these studies.

Relevance of Each Component in our Approach: Table 2 shows the performance enhancement that each module of our overall architecture brings to GZSL performance.

- $S1$ corresponds to using only the baseline generative module as explained in Section 3.1.
- $S2$ corresponds to the use of baseline generative module, along with the inference module trained using joint maximization loss. Both $S1$ and $S2$ have the final recognition module trained solely on synthesized image features (i.e $\hat{x}$) without any latent representations from the inference module.
- $S3$ denotes the use of $S2$, along with the use of output of inference network and latent representations from

the intermediate layers of the inference module in the recognition module.

- Lastly, $S4$ denotes our complete model, with the Wasserstein alignment loss added to $S3$.

Model	CUB	AWA1
$S1$ = Baseline Generative Module	51.9	61.1
$S2 = S1$ + Inference module + Joint maximization	52.7	62.5
$S3 = S2$ + Additional features for recognition module	54.4	65.4
$S4 = S3$ + Wasserstein alignment	**55.6**	**65.7**

Table 2: Ablation study of different components of our framework on CUB and AWA1. Result reported is harmonic mean accuracy.

We draw the following conclusions from Table 2. Training the generative module along with the inference network and joint maximization improves performance for both fine-grained and coarse-grained datasets. The improvement is higher for the coarse-grained dataset since visual-semantic knowledge transfer becomes more important when classes are farther apart. Utilizing features from inference network gives a strong boost in GZSL performance for both datasets, showing the importance of good features at final recognition time. Lastly, the Wasserstein alignment also adds improvement for both datasets, although more significantly in fine-grained datasets (CUB). We hypothesize that this is because the classes are close in such datasets, and aligning the distribution of semantic attributes appropriately leads to clearer decision boundary between classes.

Hyperparameter Choices: In Figures 2a, 2b, and 2c, we plot the variation in seen class, unseen class and harmonic mean performance with change in hyperparameters α_1, α_2, and γ respectively - for both coarse-grained and fine-grained datasets. In Figure 2b, the best performance is obtained at $\alpha = 10$ for the coarse-grained dataset AWA1, and at $\alpha = 1$ for the fine-grained dataset CUB. For more careful analysis, the variation of harmonic mean accuracy is also shown in Figure 2d(left) where we plot the GZSL performance on a smaller scale for the sake of clarity. It can be seen that the performance increases with an increase in α_2 for AWA1 (coarse-grained), while it decreases after a certain point for CUB (fine-grained). This behavior is expected since a higher value of α_2 is required in case of coarse-grained datasets, when compared with fine-grained datasets, as it is more difficult to learn from seen classes and generalize to unseen classes for coarse-grained datasets due to classes being more different. In case of Figures 2a and 2c, we note that higher values of α_1 and γ provide the best performance for both CUB and AWA1, showing the importance of the proposed terms. The improvement of higher values of α_1 and γ is greater in AWA1, which we ascribe to the same reason described above for α_2. Fig 2f shows the performance of our model with varying number of synthesized examples for unseen classes. The trend for harmonic mean is stable for fine and coarse-grained datasets.

Stability and Generalization: Training Generative Adversarial Networks is in general known to be difficult due to an inherently unstable training procedure. In Figure 2d(right), we show the training error trajectories over epochs for CUB and AWA1 datasets. We see that the training error smoothly decreases with a stable trend, and reaches convergence within 100 epochs for both fine-grained as well as coarse-grained datasets.

Usefulness of Latent Features in Recognition Module: In order to study the usefulness of using latent feature representations from intermediate layers of the inference network, we plot the representation before the softmax activation layer of the zero-shot classifier (recognition module) of our method and f-VAEGAN-D2, a recent state-of-the-art method, for the FLO dataset in Fig 2e. We visualize the representations for unseen classes (20 classes), since visualizing the seen classes (82 classes) can be cluttered due to their high number. Notice that the clusters for our method (right subfigure) are more compact than those of f-VAEGAN-D2 (left subfigure) for almost all classes. The clusters in f-VAEGAN-D2 show features from one class potentially leaking into other classes, which can result in misclassification. This is however improved in our approach.

6. Conclusions

In this work, we propose a unified approach for the generalized zero-shot learning problem that uses a two-level adversarial learning strategy for tight visual-semantic coupling. We use adversarial learning at the level of individual generative and inference modules, as well as use a separate joint maximization constraint across the two modules. In addition, we also show that using the latent representation of intermediate layers of the inference network improves recognition performance. This helps our model unify existing latent representation and generative approaches in a single pipeline. Our contributions in this framework enable us to capture the several modes of the data distribution better and improve GZSL performance by providing stronger visual-semantic coupling. We conduct extensive experiments on four benchmark datasets and demonstrate the value of the proposed method across these fine-grained and coarse-grained datasets. Our future work will include coming up with other ways of performing the joint maximization, as well as considering alignments beyond Wasserstein alignment to improve GZSL performance.

7. Acknowledgement

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label embedding for image classification. *TPAMI*, 2016.

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

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *ICML*, 2017.

[4] Mert Bulent and Ramazan Gokberk Cinbis Sariyildiz. Gradient matching generative networks for zero-shot learning. *CVPR*, 2016.

[5] S. Changpinyo, W.-L. Chao, B.; Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *CVPR*, 2016.

[6] Jeff Donahue, Philipp Krhenbhl, and Trevor Darrell. Adversarial feature learning. *ICLR*, 2017.

[7] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *ICLR*, 2017.

[8] Rafael Felix, Vijay BG Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. *In Proceedings of the European Conference on Computer Vision*, 2018.

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, M. A. Ranzato J. Dean, and T. Mikolov. Devise: A deep visual-semantic embedding model. *NIPS*, 2013.

[10] Zhang H. and P. Koniusz. Zero-shot kernel learning. *CVPR*, 2018.

[11] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. *CVPR*, 2019.

[12] Rohit Keshari, Richa Singh, and Mayank Vatsa. Generalized zero-shot learning via over-complete distribution. *CVPR*, 2020.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 2009.

[14] S. Liu, M. Long, Wang;, and M. I. Jordan. Generalized zero shot learning with deep calibration network. *NIPS*, 2018.

[15] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. *CVPRW*, 2018.

[16] Jian Ni, Shanghang Zhang, and Haiyong Xie. Dual adversarial semantics-consistent network for generalized zero-shot learning. *NeurIPS,*, 2019.

[17] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. *ICCVGI*, 2008.

[18] Akanksha Paul, Narayanan C. Krishnan, and Prateek Munjal. Semantically aligned bias reducing zero shot learning. *CVPR*, 2019.

[19] Zhu; Pengkai, H. Wang, and V. Saligrama. Generalized zero-shot recognition based on visually semantic embedding. *CVPR*, 2019.

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

[21] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. *CVPR*, 2019.

[22] Y.-H. H. Tsai, L.-K. Huang, and R. Salakhutdinov. Learning robust visual-semantic embeddings. *ICCV*, 2017.

[23] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. *CVPR*, 2018.

[24] Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. Meta-learning for generalized zero-shot learning. *AAAI*, 2020.

[25] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. *NIPS*, 2019.

[26] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds. 2010.

[27] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. *CVPR*, 2016.

[28] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018.

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

[30] Y. Xian, S. Sharma, B. Schiele, and Z. Akata. A feature generating framework for any-shot learning. *CVPR*, 2019.

[31] Hyeonwoo Yu and Beomhee Lee. Zero-shot learning via simultaneous generating and learning. *NIPS*, 2019.

[32] L. Zhang, T. Xiang, , and S Gong. Learning a deep embedding model for zero-shot learning. *CVPR*, 2017.