Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification

Sanath Narayan*[1], Akshita Gupta*[1], Fahad Shahbaz Khan[1,3],
Cees G. M. Snoek[2], Ling Shao[1,3]

[1] Inception Institute of Artificial Intelligence, UAE [2] University of Amsterdam
[3] Mohamed Bin Zayed University of Artificial Intelligence, UAE

Abstract. Zero-shot learning strives to classify unseen categories for which no data is available during training. In the generalized variant, the test samples can further belong to seen or unseen categories. The state-of-the-art relies on Generative Adversarial Networks that synthesize unseen class features by leveraging class-specific semantic embeddings. During training, they generate semantically consistent features, but discard this constraint during feature synthesis and classification. We propose to enforce semantic consistency at *all* stages of (generalized) zero-shot learning: training, feature synthesis and classification. We first introduce a feedback loop, from a semantic embedding decoder, that iteratively refines the generated features during both the training and feature synthesis stages. The synthesized features together with their corresponding latent embeddings from the decoder are then transformed into discriminative features and utilized during classification to reduce ambiguities among categories. Experiments on (generalized) zero-shot object and action classification reveal the benefit of semantic consistency and iterative feedback, outperforming existing methods on six zero-shot learning benchmarks. Source code at https://github.com/akshitac8/tfvaegan.

Keywords: Generalized zero-shot classification · Feature synthesis

1 Introduction

This paper strives for zero-shot learning, a challenging vision problem that involves classifying images or videos into new ("unseen") categories at test time, without having been provided any corresponding visual example during training. In the literature [1,34,45,42], this is typically achieved by utilizing the labelled seen class instances and class-specific semantic embeddings (provided as a side information), which encode the inter-class relationships. Different from the zero-shot setting, the test samples can belong to the seen or unseen categories in generalized zero-shot learning [41]. In this work, we investigate the problem of both zero-shot learning (ZSL) and generalized zero-shot learning (GZSL).

Most recent ZSL and GZSL recognition approaches [42,8,43,13,22] are based on Generative Adversarial Networks (GANs) [11], which aim at directly optimizing the divergence between real and generated data. The work of [42] learns a

[*] Equal Contribution. Correspondence: sanath.narayan@inceptioniai.org

GAN using the seen class feature instances and the corresponding class-specific semantic embeddings, which are either manually annotated or word vector [27] representations. Feature instances of the unseen categories, whose real features are unavailable during training, are then synthesized using the trained GAN and used together with the real feature instances from the seen categories to train zero-shot classifiers in a fully-supervised setting. A few works [8,13,25] additionally utilize auxiliary modules, such as a decoder, to enforce a cycle-consistency constraint on the reconstruction of semantic embeddings during training. Such an auxiliary decoder module aids the generator to synthesize semantically consistent features. Surprisingly, these modules are *only* employed during training and discarded during *both* the feature synthesis and ZSL classification stages. Since the auxiliary module aids the generator during training, it is also expected to help obtain discriminative features during feature synthesis *and* reduce the ambiguities among different classes during classification. In this work, we address the issues of enhanced feature synthesis and improved zero-shot classification.

Further, GANs are likely to encounter mode collapse issues [2], resulting in decreased diversity of generated features. While Variational Autoencoders (VAEs) [18] achieve more stable feature generation, the approximate inference distribution is likely to be different from the true posterior [48]. Recently, [43] build on [42] to combine the strengths of VAEs and GANs and introduce an `f-VAEGAN` ZSL framework by sharing the VAE decoder and GAN generator modules. To ensure that the generated features are semantically close to the distribution of real feature, a cycle-consistency loss [49] is employed between generated and original features, during training. Here, we propose to additionally enforce a similar consistency loss on the semantic embeddings during training and further utilize the learned information during feature synthesis and classification.

1.1 Contributions

We propose a novel method, which advocates the effective utilization of a semantic embedding decoder (SED) module at *all* stages of the ZSL framework: training, feature synthesis and classification. Our method is built on a VAE-GAN architecture. (i) We design a *feedback module* for (generalized) zero-shot learning that utilizes SED during both training and feature synthesis stages. The feedback module first transforms the latent embeddings of SED, which are then used to modulate the latent representations of the generator. To the best of our knowledge, we are the first to propose a feedback module, within a VAE-GAN architecture, for the problem of (generalized) zero-shot recognition. (ii) We introduce a *discriminative feature transformation*, during the classification stage, that utilizes the latent embeddings of SED along with their corresponding visual features for reducing ambiguities among object categories. In addition to object recognition, we show effectiveness of the proposed approach for (generalized) zero-shot action recognition in videos.

We validate our approach by performing comprehensive experiments on four commonly used ZSL object recognition datasets: CUB [40], FLO [29], SUN [30] and AWA [41]. Our experimental evaluation shows the benefits of utilizing SED at all stages of the ZSL/GZSL pipeline. In comparison to the baseline, the proposed approach obtains absolute gains of 4.6%, 7.1%, 1.7%, and 3.1% on CUB, FLO, SUN, and AWA, respectively for generalized zero-shot (GZSL) object recognition. In addition to object recognition, we evaluate our method on two (generalized) zero-shot action recognition in videos datasets: HMDB51 [20] and UCF101 [38]. Our approach outperforms existing methods on *all* six datasets. We also show the generalizability of our proposed contributions by integrating them into GAN-based (generalized) zero-shot recognition framework.

2 Related Work

In recent years, the problem of object recognition under zero-shot learning (ZSL) settings has been well studied [16,10,1,9,34,33,45,42]. Earlier ZSL image classification works [16,21] learn semantic embedding classifiers for associating seen and unseen classes. Different from these methods, the works of [1,9,34] learn a compatibility function between the semantic embedding and visual feature spaces. Other than these inductive approaches that rely only on the labelled data from seen classes, the works of [10,33,45] leverage additional unlabelled data from unseen classes through label propagation under a transductive zero-shot setting.

Recently, Generative Adversarial Networks [11] (GANs) have been employed to synthesize unseen class features, which are then used in a fully supervised setting to train ZSL classifiers [42,8,22,43]. A conditional Wasserstein GAN [3] (WGAN) is used along with a seen category classifier to learn the generator for unseen class feature synthesis [42]. This is achieved by using a WGAN loss and a classification loss. In [8], the seen category classifier is replaced by a decoder together with the integration of a cycle-consistency loss [49]. The work of [35] proposes an approach where cross and distribution alignment losses are introduced for aligning the visual features and corresponding embeddings in a shared latent space, using two Variational Autoencoders [18] (VAEs). The work of [43] introduces a `f-VAEGAN` framework which combines a VAE and a GAN by sharing the decoder of VAE and generator of GAN for feature synthesis. For training, the `f-VAEGAN` framework utilizes a cycle-consistency constraint between generated and original visual features. However, a similar constraint is not enforced on the semantic embeddings in their framework. Different from `f-VAEGAN`, other GAN-based ZSL classification methods [8,47,13,25] investigate the utilization of auxiliary modules to enforce cycle-consistency on the embeddings. Nevertheless, these modules are utilized only during training and discarded during both feature synthesis and ZSL classification stages.

Previous works [46,14,23,36] have investigated leveraging feedback information to incrementally improve the performance of different applications, including classification, image-to-image translation and super-resolution. To the best of our knowledge, our approach is the first to incorporate a feedback loop for improved feature synthesis in the context of (generalized) zero-shot recognition (both image and video). We systematically design a feedback module, in a VAE-GAN framework, that iteratively refines the synthesized features for ZSL.

While zero-shot image classification has been extensively studied, zero-shot action recognition in videos received less attention. Several works [19,44,28] study the problem of zero-shot action recognition in videos under transductive setting. The use of image classifiers and object detectors for action recognition under ZSL setting are investigated in [15,26]. Recently, GANs have been utilized to synthesize unseen class video features in [47,25]. Here, we further investigate the effectiveness of our framework for zero-shot action recognition in videos.

3 Method

We present an approach, `TF-VAEGAN`, for (generalized) zero-shot recognition. As discussed earlier, the objective in ZSL is to classify images or videos into new classes, which are unknown during the training stage. Different from ZSL, test samples can belong to seen or unseen classes in the GZSL setting, thereby making it a harder problem due to the domain shift between the seen and unseen classes. Let $x \in \mathcal{X}$ denote the encoded feature instances of images (videos) and $y \in \mathcal{Y}^s$ the corresponding labels from the set of M seen class labels $\mathcal{Y}^s = \{y_1, \dots, y_M\}$. Let $\mathcal{Y}^u = \{u_1, \dots, u_N\}$ denote the set of N unseen classes, which is disjoint from the seen class set $\mathcal{Y}^s$. The seen and unseen classes are described by the category-specific semantic embeddings $a(k) \in \mathcal{A}, \forall k \in \mathcal{Y}^s \cup \mathcal{Y}^u$, which encode the relationships among all the classes. While the unlabelled test features $x_t \in \mathcal{X}$ are not used during training in the inductive setting, they are used during training in the transductive setting to reduce the bias towards seen classes. The tasks in ZSL and GZSL are to learn the classifiers $f_{zsl} : \mathcal{X} \to \mathcal{Y}^u$ and $f_{gzsl} : \mathcal{X} \to \mathcal{Y}^s \cup \mathcal{Y}^u$, respectively. To this end, we first learn to synthesize the features using the seen class features x_s and corresponding embeddings $a(y)$. The learned model is then used to synthesize unseen class features $\hat{x}_u$ using the unseen class embeddings $a(u)$. The resulting synthesized features $\hat{x}_u$, along with the real seen class features x_s, are further deployed to train the final classifiers f_{zsl} and f_{gzsl}.

3.1 Preliminaries: `f-VAEGAN`

We base our approach on the recently introduced `f-VAEGAN` [43], which combines the strengths of the VAE [18] and GAN [11] as discussed earlier, achieving impressive results for ZSL classification. Compared to GAN based models, *e.g.*, `f-CLSWGAN` [42], the `f-VAEGAN` [43] generates semantically consistent features by sharing the decoder and generator of the VAE and GAN. In `f-VAEGAN`, the feature generating VAE [18] (`f-VAE`) comprises an encoder $E(x, a)$, which encodes an input feature x to a latent code z, and a decoder $G(z, a)$ (shared with `f-WGAN`, as a conditional generator) that reconstructs x from z. Both E and G are conditioned on the embedding a, optimizing,

$$\mathcal{L}_V = \mathrm{KL}(E(x, a) || p(z|a)) - \mathbb{E}_{E(x,a)}[\log G(z, a)], \tag{1}$$

where KL is the Kullback-Leibler divergence, $p(z|a)$ is a prior distribution, assumed to be $\mathcal{N}(0, 1)$ and $\log G(z, a)$ is the reconstruction loss. The feature generating network [42] (`f-WGAN`) comprises a generator $G(z, a)$ and a discriminator

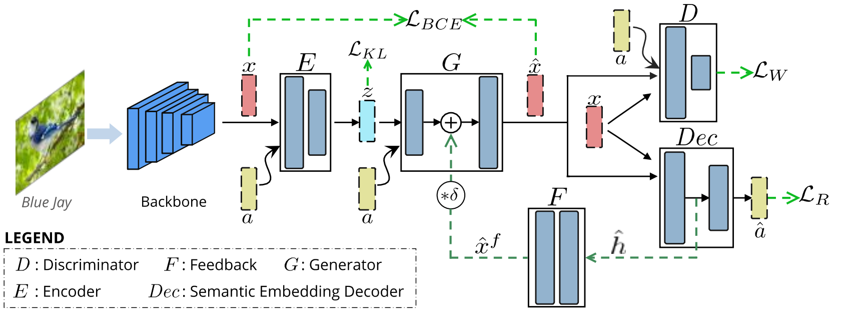


Fig. 1: **Proposed architecture** (Sec 3.2). Given a seen class image, visual features x are extracted from the backbone network and input to the encoder E, along with the corresponding semantic embeddings a. The encoder E outputs a latent code z, which is then input together with embeddings a to the generator G that synthesizes features $\hat{x}$. The discriminator D learns to distinguish between real and synthesized features x and $\hat{x}$, respectively. Both E and G together constitute the VAE, which is trained using a binary cross-entropy loss ($\mathcal{L}_{BCE}$) and the KL divergence ($\mathcal{L}_{KL}$). Similarly, both G and D form the GAN trained using the WGAN loss ($\mathcal{L}_W$). A semantic embedding decoder Dec is introduced (Sec. 3.3) to reconstruct the embeddings $\hat{a}$ using a cycle-consistency loss ($\mathcal{L}_R$). Further, a feedback module F (Sec. 3.4) is integrated to transform the latent embedding $\hat{h}$ of Dec and feed it back to G, which iteratively refines $\hat{x}$.

$D(x, a)$. The generator $G(z, a)$ synthesizes a feature $\hat{x} \in \mathcal{X}$ from a random input noise z, whereas the discriminator $D(x, a)$ takes an input feature x and outputs a real value indicating the degree of realness or fakeness of the input features. Both G and D are conditioned on the embedding a, optimizing the WGAN loss $\mathcal{L}_W = \mathbb{E}[D(x, a)] - \mathbb{E}[D(\hat{x}, a)] - \lambda\mathbb{E}[(||\nabla D(\tilde{x}, a)||_2 - 1)^2]$. Here, $\hat{x} = G(z, a)$ is the synthesized feature, λ is the penalty coefficient and $\tilde{x}$ is sampled randomly from the line connecting x and $\hat{x}$. The `f-VAEGAN` is then optimized by:

$$\mathcal{L}_{vaegan} = \mathcal{L}_V + \alpha\mathcal{L}_W, \tag{2}$$

where α is a hyper-parameter. For more details, we refer to [43].

Limitations: The loss formulation for training `f-VAEGAN`, contains a constraint (second term in Eq. 1) that ensures the generated visual features are cyclically-consistent, at train time, with the original visual features. However, a similar cycle-consistency constraint is not enforced on the semantic embeddings. Alternatively, other GAN-based ZSL methods [8,47] utilize auxiliary modules (apart from the generator) for achieving cyclic-consistency on embeddings. However, these modules are employed *only* during training and discarded at both feature synthesis and ZSL classification stages. In this work, we introduce a semantic embedding decoder (SED) that enforces cycle-consistency on semantic embeddings and utilize it at *all* stages: training, feature synthesis and ZSL classification. We argue that the generator and SED contain complementary information with respect to feature instances, since the two modules perform inverse transformations in relation to each other. The generator module transforms the semantic embeddings to the feature instances whereas, SED transforms the feature instances to semantic embeddings. Our approach focuses on the utilization of this complementary information for improving feature synthesis and reducing ambiguities among classes (*e.g.*, fine-grained classes) during ZSL classification.

3.2 Overall Architecture

The overall architecture is illustrated in Fig. 1. The VAE-GAN consists of an encoder E, generator G and discriminator D. The input to E are the real features of seen classes x and the semantic embeddings a and the output of E are the parameters of a noise distribution. These parameters are matched to those of a zero-mean unit-variance Gaussian prior distribution using the KL divergence ($\mathcal{L}_{KL}$). The noise z and embeddings a are input to G, which synthesizes the features $\hat{x}$. The synthesized features $\hat{x}$ and original features x are compared using a binary cross-entropy loss $\mathcal{L}_{BCE}$. The discriminator D takes either x or $\hat{x}$ along with embeddings a as input, and computes a real number that determines whether the input is real or fake. The WGAN loss $\mathcal{L}_{W}$ is applied at the output of D to learn to distinguish between the real and fake features.

The focus of our design is the integration of an additional semantic embedding decoder (SED) Dec at both the feature synthesis and ZSL/GZSL classification stages. Additionally, we introduce a feedback module F, which is utilized during training and feature synthesis, along with Dec. Both the semantic embedding decoder Dec and feedback module F collectively address the objectives of enhanced feature synthesis and reduced ambiguities among categories during classification. The Dec takes either x or $\hat{x}$ and reconstructs the embeddings $\hat{a}$. It is trained using a cycle-consistency loss $\mathcal{L}_R$. The learned Dec is subsequently used in the ZSL/GZSL classifiers. The feedback module F transforms the latent embedding of Dec and feeds it back to the latent representation of generator G in order to achieve improved feature synthesis. The SED Dec and feedback module F are described in detail in Sec. 3.3 and 3.4.

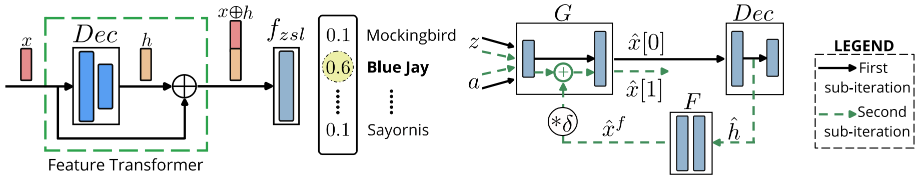
3.3 Semantic Embedding Decoder

Here, we introduce a semantic embedding decoder $Dec : \mathcal{X} \to \mathcal{A}$, for reconstructing the semantic embeddings a from the generated features $\hat{x}$. Enforcing a cycle-consistency on the reconstructed semantic embeddings ensures that the generated features are transformed to the same embeddings that generated them. As a result, semantically consistent features are obtained during feature synthesis. The cycle-consistency of the semantic embeddings is achieved using the ℓ_1 reconstruction loss as follows:

$$\mathcal{L}_R = \mathbb{E}[||Dec(x) - a||_1] + \mathbb{E}[||Dec(\hat{x}) - a||_1]. \tag{3}$$

The loss formulation for training the proposed `TF-VAEGAN` is then given by,

$$\mathcal{L}_{total} = \mathcal{L}_{vaegan} + \beta \mathcal{L}_R, \tag{4}$$



(a) Discriminative Feature Transformation (b) Feedback Module

Fig. 2: (a) **Integration of semantic embedding decoder** Dec at the ZSL/GZSL classification stage. A feature transformation is performed by concatenating ($\oplus$) the input visual features x with the corresponding latent embedding h from SED. The transformed discriminative features are then used for ZSL/GZSL classification.
(b) **Feedback module overview**. First sub-iteration: The generator G synthesizes initial features $\hat{x}[0]$ using the noise z and embeddings a. The initial features are passed through the Dec. Second sub-iteration: The module F transforms the latent embedding h from Dec to $\hat{x}^f$, which represents the feedback to G. The generator G synthesizes enhanced features $\hat{x}[1]$ using the same z and a along with the feedback $\hat{x}^f$.

where β is a hyper-parameter for weighting the decoder reconstruction error.

As discussed earlier, existing GAN-based ZSL approaches [8,47] employ a semantic embedding decoder (SED) *only* during training and discard it during *both* unseen class feature synthesis and ZSL classification stage. In our approach, SED is utilized at *all* three stages of VAE-GAN based ZSL pipeline: training, feature synthesis and classification. Next, we describe importance of SED during classification and later investigate its role during feature synthesis (Sec. 3.4).

Discriminative feature transformation: Here, we describe the proposed discriminative feature transformation scheme to effectively utilize the auxiliary information in semantic embedding decoder (SED) at the ZSL classification stage. The generator G learns a *per-class* "single semantic embedding to many instances" mapping using only the seen class features and embeddings. Similar to the generator G, the SED is also trained using only the seen classes but learns a *per-class* "many instances to one embedding" inverse mapping. Thus, the generator G and SED Dec are likely to encode complementary information of the categories. Here, we propose to use the latent embedding from SED as a useful source of information at the classification stage (see Fig. 2a) for reducing ambiguities among features instances of different categories.

First, the training of feature generator G and semantic embedding decoder Dec is performed. Then, Dec is used to transform the features (real and synthesized) to the embedding space $\mathcal{A}$. Afterwards, the latent embeddings from Dec are concatenated with the respective visual features. Let h_s and $\hat{h}_u \in \mathcal{H}$ denote the hidden layer (latent) embedding from the Dec for inputs x_s and $\hat{x}_u$, respectively. The transformed features are represented by: $x_s \oplus h_s$ and $\hat{x}_u \oplus \hat{h}_u$, where $\oplus$ denotes concatenation. In our method, the transformed features are used to learn final ZSL and GZSL classifiers as,

$$f_{zsl} : \mathcal{X} \oplus \mathcal{H} \rightarrow \mathcal{Y}^u \qquad \text{and} \qquad f_{gzsl} : \mathcal{X} \oplus \mathcal{H} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u. \tag{5}$$

As a result, the final classifiers learn to better distinguish categories using transformed features. Next, we describe integration of *Dec* during feature synthesis.

3.4 Feedback Module

The baseline `f-VAEGAN` does not enforce cycle-consistency in the attribute space and directly synthesizes visual features $\hat{x}$ from the class-specific embeddings a via the generator (see Fig. 3a). This results in a semantic gap between the real and synthesized visual features. To address this issue, we introduce a feedback loop that iteratively refines the feature generation (see Fig. 3b) during both the training and synthesis stages. The feedback loop is introduced from the semantic embedding decoder *Dec* to the generator G, through our feedback module F (see Fig. 1 and Fig. 2b). The proposed module F enables the effective utilization of *Dec* during both training and feature synthesis stages. Let g^l denote the l^{th} layer output of G and $\hat{x}^f$ denote the feedback component that additively modulates g^l. The feedback modulation of output g^l is given by,

$$g^l \leftarrow g^l + \delta \hat{x}^f, \tag{6}$$

where $\hat{x}^f = F(h)$, with h as the latent embedding of *Dec* and δ controls the feedback modulation. To the best of our knowledge, we are the first to design and incorporate a feedback loop for zero-shot recognition. Our feedback loop is based on [36], originally introduced for image super-resolution. However, we observe that it provides sub-optimal performance for zero-shot recognition due to its less reliable feedback during unseen class feature synthesis. Next, we describe an improved feedback loop with necessary modifications for zero-shot recognition.

Feedback module input: The adversarial feedback employs a latent representation of an unconditional discriminator D as its input [36]. However, in the ZSL problem, D is conditional and is trained with an objective to distinguish between the real and fake features of the seen categories. This restricts D from providing reliable feedback during unseen class feature synthesis. In order to overcome this limitation, we turn our attention to semantic embedding decoder *Dec*, whose aim is to reconstruct the class-specific semantic embeddings from features instances. Since *Dec* learns class-specific transformations from visual features to the semantic embeddings, it is better suited (than D) to provide feedback to generator G.

Training strategy: Originally, the feedback module F is trained in a two-stage fashion [36], where the generator G and discriminator D are first fully trained, as in the standard GAN training approach. Then, F is trained using a feedback from D and freezing G. Since, the output of G improves due to the feedback from F, the discriminator D is continued to be trained alongside F, in an adversarial manner. In this work, we argue that such a two-stage training strategy is sub-optimal for ZSL, since G is always fixed and not allowed to improve its feature synthesis. To further utilize the feedback for improved feature synthesis, G and F are trained alternately in our method. In our alternating training strategy, the generator training iteration is unchanged. However, during the training iterations of F, we perform two sub-iterations (see Fig. 2b).

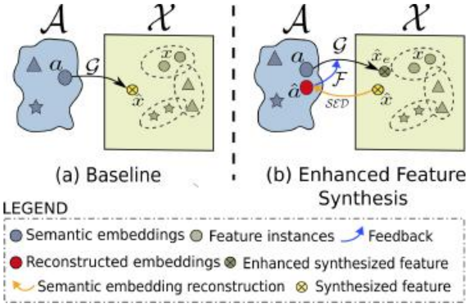


Fig. 3: **Conceptual illustration** between the baseline (a) and our feedback module designed for enhanced feature synthesis (b), using three classes (★, ▲ and ●). The baseline learns to synthesize features $\hat{x}$ from class-specific semantic embeddings a via generator G, without enforcing cycle-consistency in the attribute space. As a consequence, a semantic gap is likely to exist between the synthesized and real x features. In our approach, cycle-consistency is enforced using SED. Further, the disparity between the reconstructed embeddings $\hat{a}$ and a is used as a *feedback signal* to reduce the semantic gap between $\hat{x}$ and x, resulting in enhanced synthesized features $\hat{x}_e$.

First sub-iteration: The noise z and semantic embeddings a are input to the generator G to yield an initial synthesized feature $\hat{x}[0] = G(z, a)$, which is then passed through to the semantic embedding decoder *Dec*.
Second sub-iteration: The latent embedding $\hat{h}$ from *Dec* is input to F, resulting in an output $\hat{x}^f[t] = F(\hat{h})$, which is added to the latent representation (denoted as g^l in Eq. 6) of G. The same z and a (used in the first sub-iteration) are used as input to G for the second sub-iteration, with the additional input $\hat{x}^f[t]$ added to the latent representation g^l of generator G. The generator then outputs a synthesized feature $\hat{x}[t+1]$, as,

$$\hat{x}[t+1] = G(z, a, \hat{x}^f[t]). \tag{7}$$

The refined feature $\hat{x}[t+1]$ is input to D and *Dec*, and corresponding losses are computed (Eq. 4) for training. In practice, the second sub-iteration is performed only once. The feedback module F allows generator G to view the latent embedding of *Dec*, corresponding to current generated features. This enables G to appropriately refine its output (feature generation) iteratively, leading to an enhanced feature representation.

3.5 (Generalized) Zero-Shot Classification

In our `TF-VAEGAN`, unseen class features are synthesized by inputting respective embeddings $a(u)$ and noise z to G, given by $\hat{x}_u = G(z, a(u), \hat{x}^f[0])$. Here, $\hat{x}^f[0]$ denotes feedback output of F, computed for the same $a(u)$ and z. The synthesized unseen class features $\hat{x}_u$ and real seen class features x_s are further input to *Dec* to obtain their respective latent embeddings, which are concatenated with input features. In this way, we obtain transformed features $x_s \oplus h_s$ and $\hat{x}_u \oplus \hat{h}_u$, which are used to train ZSL and GZSL classifiers, f_{zsl} and f_{gzsl}, respectively. At inference, test features x_t are transformed in a similar manner, to obtain $x_t \oplus h_t$. The transformed features are then input to classifiers for final predictions.

4 Experiments

Datasets: We evaluate our `TF-VAEGAN` framework on four standard zero-shot object recognition datasets: Caltech-UCSD-Birds [40] (CUB), Oxford Flowers [29]

(FLO), SUN Attribute [30] (SUN), and Animals with Attributes2 [41] (AWA2) containing 200, 102, 717 and 50 categories, respectively. For fair comparison, we use the *same* splits, evaluation protocols and class embeddings as in [41].

Visual features and embeddings: We extract the average-pooled feature instances of size 2048 from the ImageNet-1K [6] pre-trained ResNet-101 [12]. For semantic embeddings, we use the class-level attributes for CUB (312-d), SUN (102-d) and AWA2 (85-d). For FLO, fine-grained visual descriptions of image are used to extract 1024-d embeddings from a character-based CNN-RNN [32].

Implementation details: The discriminator D, encoder E and generator G are implemented as two-layer fully-connected (FC) networks with 4096 hidden units. The dimensions of z and a are set to be equal ($\mathbb{R}^{d_z} = \mathbb{R}^{d_a}$). The semantic embedding decoder Dec and feedback module F are also two-layer FC networks with 4096 hidden units. The input and output dimensions of F are set to 4096 to match the hidden units of Dec and G. For transductive setting, an unconditional discriminator $D2$ is employed for utilizing the unlabelled feature instances during training, as in [43]. Since the corresponding semantic embeddings are not available for unlabelled instances, only the visual feature is input to $D2$. Leaky ReLU activation is used everywhere, except at the output of G, where a *sigmoid* activation is used for applying BCE loss. The network is trained using the Adam optimizer with 10^{-4} learning rate. Final ZSL/GZSL classifiers are single layer FC networks with output units equal to number of test classes. Hyper-parameters α, β and δ are set to 10, 0.01 and 1, respectively. The gradient penalty coefficient λ is initialized to 10 and WGAN is trained, similar to [3].

4.1 State-of-the-art Comparison

Tab. 1 shows state-of-the-art comparison on four object recognition datasets. Results for inductive (IN) and transductive (TR) settings are obtained without any fine-tuning of the backbone network. For inductive (IN) ZSL, the `Cycle-WGAN` [8] obtains classification scores of 58.6%, 70.3%, 59.9%, and 66.8% on CUB, FLO, SUN and AWA, respectively. The `f-VAEGAN` [43] reports classification accuracies of 61%, 67.7%, 64.7%, and 71.1% on the same datasets. Our `TF-VAEGAN` outperforms `f-VAEGAN` on *all* datasets achieving classification scores of 64.9%, 70.8%, 66.0%, and 72.2% on CUB, FLO, SUN and AWA, respectively. In the transductive (TR) ZSL setting, `f-VAEGAN` obtains *top-1* classification (**T1**) accuracies of 71.1%, 89.1%, 70.1%, and 89.8% on the four datasets. Our `TF-VAEGAN` outperforms `f-VAEGAN` on *all* datasets, achieving classification accuracies of 74.7%, 92.6%, 70.9%, and 92.1% on CUB, FLO, SUN and AWA, respectively. Similarly, our `TF-VAEGAN` also performs favourably compared to existing methods on all datasets for both inductive and transductive GZSL settings. Utilizing unlabelled instances during training, to reduce the domain shift problem for unseen classes, in the transductive setting yields higher results compared to inductive setting.

Some previous works, including `f-VAEGAN` [43] have reported results with fine-tuning the backbone network only using the seen classes (without violating the ZSL condition). Similarly, we also evaluate our `TF-VAEGAN` by utilizing fine-tuned backbone features. Tab. 1 shows the comparison with existing fine-tuning

Table 1: **State-of-the-art comparison on four datasets.** Both inductive (IN) and transductive (TR) results are shown. The results with fine-tuning the backbone network using the seen classes only (without violating ZSL), are reported under fine-tuned inductive (FT-IN) and transductive (FT-TR) settings. For ZSL, results are reported in terms of average *top-1* classification accuracy (**T1**). For GZSL, results are reported in terms of *top-1* accuracy of unseen (u) and seen (s) classes, together with their harmonic mean (**H**). Our `TF-VAEGAN` performs favorably in comparison to existing methods on *all* four datasets, in all settings (IN, TR, FT-IN and FT-TR), for *both* ZSL and GZSL.

		Zero-shot Learning				**Generalized Zero-shot Learning**											
		CUB	**FLO**	**SUN**	**AWA**	**CUB**			**FLO**			**SUN**			**AWA**		
		T1	**T1**	**T1**	**T1**	u	s	**H**	u	s	**H**	u	s	**H**	u	s	**H**
IN	`f-CLSWGAN` [42]	57.3	67.2	60.8	68.2	3.7	57.7	49.7	59.0	73.8	65.6	42.6	36.6	39.4	57.9	61.4	59.6
	`Cycle-WGAN` [8]	58.6	70.3	59.9	66.8	47.9	59.3	53.0	61.6	69.2	65.2	**47.2**	33.8	39.4	59.6	63.4	59.8
	`LisGAN` [22]	58.8	69.6	61.7	70.6	46.5	57.9	51.6	57.7	83.8	68.3	42.9	37.8	40.2	52.6	**76.3**	62.3
	`TCN` [17]	59.5	-	61.5	71.2	52.6	52.0	52.3	-	-	-	31.2	37.3	34.0	**61.2**	65.8	63.4
	`f-VAEGAN` [43]	61.0	67.7	64.7	71.1	48.4	60.1	53.6	56.8	74.9	64.6	45.1	38.0	41.3	57.6	70.6	63.5
	`Ours: TF-VAEGAN`	**64.9**	**70.8**	**66.0**	**72.2**	**52.8**	**64.7**	**58.1**	**62.5**	**84.1**	**71.7**	45.6	**40.7**	**43.0**	59.8	75.1	**66.6**
TR	`ALE-tran` [41]	54.5	48.3	55.7	70.7	23.5	45.1	30.9	13.6	61.4	22.2	19.9	22.6	21.2	12.6	73.0	21.5
	`GFZSL` [39]	50.0	85.4	64.0	78.6	24.9	45.8	32.2	21.8	75.0	33.8	0.0	41.6	0.0	31.7	67.2	43.1
	`DSRL` [45]	48.7	57.7	56.8	72.8	17.3	39.0	24.0	26.9	64.3	37.9	17.7	25.0	20.7	20.8	74.7	32.6
	`f-VAEGAN` [43]	71.1	89.1	70.1	89.8	61.4	65.1	63.2	78.7	87.2	82.7	60.6	41.9	49.6	84.8	88.6	86.7
	`Ours: TF-VAEGAN`	**74.7**	**92.6**	**70.9**	**92.1**	**69.9**	**72.1**	**71.0**	**91.8**	**93.2**	**92.5**	**62.4**	**47.1**	**53.7**	**87.3**	**89.6**	**88.4**
FT-IN	`SBAR-I` [31]	63.9	-	62.8	65.2	55.0	58.7	56.8	-	-	-	**50.7**	35.1	41.5	30.3	**93.9**	46.9
	`f-VAEGAN` [43]	72.9	70.4	65.6	70.3	63.2	75.6	68.9	63.3	92.4	75.1	50.1	37.8	43.1	**57.1**	76.1	65.2
	`Ours: TF-VAEGAN`	**74.3**	**74.7**	**66.7**	**73.4**	**63.8**	**79.3**	**70.7**	**69.5**	**92.5**	**79.4**	41.8	**51.9**	**46.3**	55.5	83.6	**66.7**
FT-TR	`SBAR-T` [31]	74.0	-	67.5	88.9	67.2	73.7	70.3	-	-	-	**58.8**	41.5	48.6	79.7	**91.0**	85.0
	`UE-finetune` [37]	72.1	-	58.3	79.7	74.9	71.5	73.2	-	-	-	33.6	54.8	41.7	**93.1**	66.2	77.4
	`f-VAEGAN` [43]	82.6	95.4	72.6	89.3	73.8	81.4	77.3	91.0	97.4	94.1	54.2	41.8	47.2	86.3	88.7	87.5
	`Ours: TF-VAEGAN`	**85.1**	**96.0**	**73.8**	**93.0**	**78.4**	**83.5**	**80.9**	**96.1**	**97.6**	**96.8**	44.3	**66.9**	**53.3**	89.2	90.0	**89.6**

based methods for both ZSL and GZSL in fine-tuned inductive (FT-IN) and fine-tuned transductive (FT-TR) settings. For FT-IN ZSL, `f-VAEGAN` obtains classification scores of 72.9%, 70.4%, 65.6%, and 70.3% on CUB, FLO, SUN and AWA, respectively. Our `TF-VAEGAN` achieves consistent improvement over `f-VAEGAN` on *all* datasets, achieving classification scores of 74.3%, 74.7%, 66.7%, and 73.4% on CUB, FLO, SUN and AWA, respectively. Our approach also improves over `f-VAEGAN` for the FT-TR ZSL setting. In the case of FT-IN GZSL, our `TF-VAEGAN` achieves gains (in terms of **H**) of 1.8%, 4.3%, 3.2%, and 1.5% on CUB, FLO, SUN and AWA, respectively over `f-VAEGAN`. A similar trend is also observed for the FT-TR GZSL setting. In summary, our `TF-VAEGAN` achieves promising results for various settings and backbone feature combinations.

4.2 Ablation Study

Baseline comparison: We first compare our proposed `TF-VAEGAN` with the baseline `f-VAEGAN` [43] on CUB for (generalized) zero-shot recognition in both inductive and transductive settings. The results are reported in Tab. 2 in terms of average *top-1* classification accuracy for ZSL and harmonic mean of the classification accuracies of seen and unseen classes for GZSL. For the baseline, we present the results based on our re-implementation. In addition to our final `TF-VAEGAN`, we report results of our feedback module alone (denoted

Table 2: **Baseline performance comparison** on CUB [40]. In both inductive and transductive settings, our Feedback and T-feature provide consistent improvements over the baseline for both ZSL and GZSL. Further, our final TF-VAEGAN framework, integrating both Feedback and T-feature, achieves further gains over the baseline in both inductive and transductive settings, for ZSL and GZSL.

	INDUCTIVE				TRANSDUCTIVE			
	Baseline	Feedback	T-feature	TF-VAEGAN	Baseline	Feedback	T-feature	TF-VAEGAN
ZSL	61.2	62.8	64.0	**64.9**	70.6	71.7	73.5	**74.7**
GZSL	53.5	54.8	56.9	**58.1**	63.7	66.8	69.2	**71.0**

as Feedback in Tab. 2) without feature transformation utilized at classification stage. Moreover, the performance of discriminative feature transformation alone (denoted as T-feature), without utilizing the feedback is also presented. For the inductive setting, Baseline obtains a classification performance of 61.2% and 53.5% for ZSL and GZSL. Both our contributions, Feedback and T-feature, consistently improve the performance over the baseline. The best results are obtained by our TF-VAEGAN, with gains of 3.7% and 4.6% over the baseline, for ZSL and GZSL. Similar to the inductive (IN) setting, our proposed TF-VAEGAN also achieves favourable performance in transductive (TR) setting. Fig. 4 shows a comparison between baseline and our TF-VAEGAN methods, using t-SNE visualizations [24] of test instances from four example fine-grained classes of CUB. While the baseline struggles to correctly classify these fine-grained class instances due to inter-class confusion, our TF-VAEGAN improves inter-class grouping leading to a favorable classification performance.

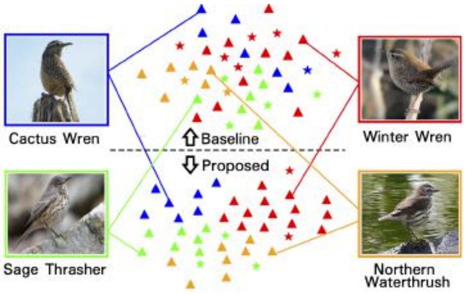
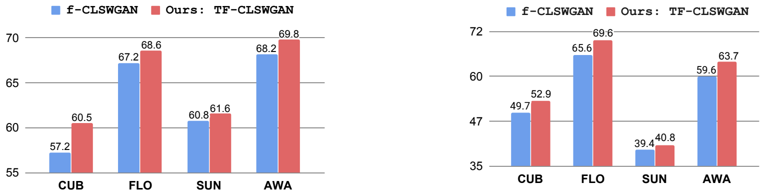


Fig. 4: **t-SNE visualization** of test instances of four fine-grained classes in CUB [40] dataset. Both Cactus Wren and Winter Wren belong to the same family Troglodytidae. Further, Cactus Wren is visually similar to Sage Thrasher and Northern Waterthrush. Top: the baseline method struggles to correctly classify instances of these categories (denoted by ★ with respective class color) due to inter-class confusion. Bottom: our approach improves the inter-class grouping and decreases misclassifications, leading to favourable performance.

Generalization capabilities: Here, we base our approach on a VAE-GAN architecture [43]. However, our proposed contributions (a semantic embedding decoder at all stages of the ZSL pipeline and the feedback module) are generic and can also be utilized in other GAN-based ZSL frameworks. To this end, we perform an experiment by integrating our contributions in the f-CLSWGAN [42] architecture. Fig. 5 shows the comparison between the baseline f-CLSWGAN and our TF-CLSWGAN for ZSL and GZSL tasks, on all four datasets. Our TF-CLSWGAN



(a) ZSL (b) GZSL

Fig. 5: **Generalization capabilities.** (a) ZSL and (b) GZSL performance comparison to validate the generalization capabilities of our contributions. Instead of a VAE-GAN architecture, we integrate our proposed contributions in the `f-CLSWGAN` framework. Our `TF-CLSWGAN` outperforms the vanilla `f-CLSWGAN` on all datasets. Best viewed in zoom.

outperforms the vanilla `f-CLSWGAN` in all cases for both ZSL and GZSL tasks.
Feature visualization: To qualitatively assess the feature synthesis stage, we train an upconvolutional network to invert the feature instances back to the image space by following a similar strategy as in [7,43]. Corresponding implementation details are provided in the supplementary. The model is trained on all real feature-image pairs of the 102 classes of FLO [29]. The comparison between `Baseline` and our `Feedback` synthesized features on four example flowers is shown in Fig. 6. For each flower class, a ground-truth (GT) image along with three images inverted from its GT feature, `Baseline` and `Feedback` synthesized features, respectively are shown. Generally, inverting the `Feedback` synthesized feature yields an image that is semantically closer to the GT image than inverting the `Baseline` synthesized feature. This suggests that our `Feedback` improves the feature synthesis stage over the `Baseline`, where no feedback is present. Additional quantitative and qualitative results are given in the supplementary.

5 (Generalized) Zero-Shot Action Recognition

Finally, we validate our `TF-VAEGAN` for action recognition in videos under ZSL and GZSL. Here, we use the I3D features [5], as in the GAN-based zero-shot action classification method `CEWGAN` [25]. While using improved video features is likely to improve the performance of a zero-shot action recognition framework, our goal is to show that our `TF-VAEGAN` generalizes to action classification and improves the performance using the same underlying video features. As in [25], we extract spatio-temporally pooled 4096-d I3D features from pre-trained RGB and Flow I3D networks and concatenate them to obtain 8192-d video features. Further, an out-of-distribution classifier is utilized at the classification stage, as in [25]. For HMDB51, a skip-gram model [27] is used to generate semantic embeddings of size 300, using action class names as input. For UCF101, we use semantic embeddings of size 115, provided with the dataset.
Tab. 3 shows state-of-the-art comparison on HMDB51 [20] and UCF101 [38]. For a fair comparison, we use the same splits, embeddings and evaluation pro-

Table 3: **State-of-the-art ZSL and GZSL comparison for action recognition**. Our TF-VAEGAN performs favorably against all existing methods, on both datasets.

		GGM [28]	CLSWGAN [42]	CEWGAN [25]	Obj2Act [15]	ObjEmb [26]	f-VAEGAN [43]	TF-VAEGAN
HMDB51	ZSL	20.7	29.1	30.2	24.5	-	31.1	**33.0**
	GZSL	20.1	32.7	36.1	-	-	35.6	**37.6**
UCF101	ZSL	20.3	37.5	38.3	38.9	40.4	38.2	**41.0**
	GZSL	17.5	44.4	49.4	-	-	47.2	**50.9**

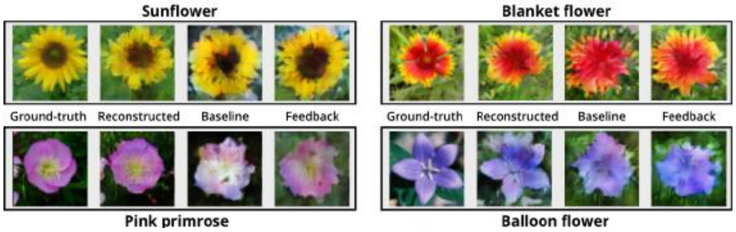


Fig. 6: **Qualitative comparison** between inverted images of Baseline synthesized features and our Feedback synthesized features on four example classes of FLO [29]. The ground-truth image and the reconstructed inversion of its real feature are also shown for each example. Our Feedback improves petal shapes (*Sunflower*), shape of bud and petals (*Blanket flower*), color (*Pink primrose*), black lining on petals (*Balloon flower*) and achieves promising improvements over Baseline. Best viewed in zoom.

tocols as in [25]. On HMDB51, f-VAEGAN obtains classification scores of 31.1% and 35.6% for ZSL and GZSL. The work of [50] provides classification results of 24.4% and 17.5% for HMDB51 and UCF101, respectively for ZSL. Note that [50] also reports results using cross-dataset training on large-scale ActivityNet [4]. On HMDB51, CEWGAN [25] obtains 30.2% and 36.1% for ZSL and GZSL. Our TF-VAEGAN achieves 33.0% and 37.6% for ZSL and GZSL. Similarly, our approach performs favourably compared to existing methods on UCF101. Hence, our TF-VAEGAN generalizes to action recognition and achieves promising results.

6 Conclusion

We propose an approach that utilizes the semantic embedding decoder (SED) at all stages (training, feature synthesis and classification) of a VAE-GAN based ZSL framework. Since SED performs inverse transformations in relation to the generator, its deployment at all stages enables exploiting complementary information with respect to feature instances. To effectively utilize SED during both training and feature synthesis, we introduce a feedback module that transforms the latent embeddings of the SED and modulates the latent representations of the generator. We further introduce a discriminative feature transformation, during the classification stage, which utilizes the latent embeddings of SED along with respective features. Experiments on six datasets clearly suggest that our approach achieves favorable performance, compared to existing methods.

References

1. Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 2015. 1, 3
2. Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017. 2
3. Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 3, 10
4. Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 14
5. Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 13
6. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 10
7. Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NeurIPS*, 2016. 13
8. Rafael Felix, B. G. Vijay Kumar, Ian Reid, and Gustavo Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, 2018. 1, 2, 3, 5, 7, 10, 11
9. Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 3
10. Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *TPAMI*, 2015. 3
11. Ian Goodfellow, Jean PougetAbadie, Mehdi Mirza, Bing Xu, and David Warde-Farley. Generative adversarial nets. In *NeurIPS*, 2014. 1, 3, 4
12. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 10
13. He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *CVPR*, 2019. 1, 2, 3
14. Minyoung Huh, Shao-Hua Sun, and Ning Zhang. Feedback adversarial learning: Spatial feedback for improving generative adversarial networks. In *CVPR*, 2019. 3
15. Mihir Jain, Jan C van Gemert, Thomas Mensink, and Cees GM Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015. 4, 14
16. Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *NeurIPS*, 2014. 3
17. Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *ICCV*, 2019. 11
18. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 2, 3, 4
19. Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015. 4
20. H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 3, 13
21. Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 2013. 3
22. Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, 2019. 1, 3, 11

23. Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *CVPR*, 2019. 3
24. Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 12
25. Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *CVPR*, 2019. 2, 3, 4, 13, 14
26. Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017. 4, 14
27. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 2, 13
28. Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, Arulkumar S, Piyush Rai, and Anurag Mittal. A generative approach to zero-shot and few-shot action recognition. In *WACV*, 2018. 4, 14
29. Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 2, 9, 13, 14, 19, 20, 21, 23
30. Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 2, 10
31. Akanksha Paul, Narayanan C Krishnan, and Prateek Munjal. Semantically aligned bias reducing zero shot learning. In *CVPR*, 2019. 11
32. Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016. 10
33. Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *NeurIPS*, 2013. 3
34. Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 1, 3
35. Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019. 3
36. Firas Shama, Roey Mechrez, Alon Shoshan, and Lihi Zelnik-Manor. Adversarial feedback loop. In *ICCV*, 2019. 3, 8
37. Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *CVPR*, 2018. 11
38. Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 3, 13
39. Vinay Kumar Verma and Piyush Rai. A simple exponential family framework for zero-shot learning. In *ECML*, 2017. 11
40. Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. *Technical Report CNS-TR-2010-001, Caltech*, 2010. 2, 9, 12, 21, 22
41. Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018. 1, 2, 10, 11
42. Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 1, 2, 3, 4, 11, 12, 14
43. Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 1, 2, 3, 4, 5, 10, 11, 12, 13, 14, 21
44. Xun Xu, Timothy Hospedales, and Shaogang Gong. Transductive zero-shot action recognition by word-vector embedding. *IJCV*, 2017. 4

45. Meng Ye and Yuhong Guo. Zero-shot classification with discriminative semantic representation learning. In *CVPR*, 2017. 1, 3, 11
46. Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *CVPR*, 2017. 3
47. Chenrui Zhang and Yuxin Peng. Visual data synthesis via gan for zero-shot video classification. In *IJCAI*, 2018. 3, 4, 5, 7
48. Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*, 2019. 2
49. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 3
50. Yi Zhu, Yang Long, Yu Guan, Shawn Newsam, and Ling Shao. Towards universal representation for unseen action recognition. In *CVPR*, 2018. 14

A Quantitative Results

In this section, we present the ablation studies with respect to the feedback design choices and the choice of latent embeddings.

Feedback design choices: Here, we explore the effect of changing the input to the feedback module F and its associated training strategy on CUB. Originally, the input to F is taken from discriminator D and the training of F is performed in a two-stage strategy. This setup is denoted by `TwoStage+D` and obtains classification performance of 61.4% and 53.3% for ZSL and GZSL. Instead, in our approach, the input to F is taken from SED *Dec*. This setup is denoted by `TwoStage+Dec` and achieves performance of 62.0% and 53.8% for ZSL and GZSL. Further, we utilize an alternate training strategy combined with `TwoStage+Dec` to facilitate the generator training, thereby improving feature synthesis. This setup, denoted by `Our Feedback`, achieves improved performance of 62.8% and 54.8% for ZSL and GZSL. These results show that (i) `TwoStage+Dec` provides improved performance over original `TwoStage+D` and (ii) the best results are obtained by `Our Feedback`, demonstrating the impact of our modifications for improved zero-shot recognition.

Choice of latent embeddings for `T-feature`: Here, we evaluate the impact of concatenating different embeddings from SED to the baseline features. We compare our proposed concatenation (`T-feature`) of baseline features with latent embeddings h of SED with both the original baseline features (`OrigFeat`) and the baseline features concatenated with the reconstructed attributes (`ConcatFeat`). On CUB, `OrigFeat` achieves 61.2% and 53.5% on ZSL and GZSL tasks, respectively. `ConcatFeat` achieves gains of 1.6% and 2.0% over `OrigFeat`. In case of `ConcatFeat`, the reconstructed attributes have single feature representations per-class with inter-class separability but no intra-class diversity. Different to reconstructed attributes, the latent embeddings h possess both intra-class diversity (multiple feature instances per class) and inter-class separability. Our `T-feature` exploits these properties of latent embeddings with improved results over both `OrigFeat` and `ConcatFeat`. Compared to `OrigFeat`, `T-feature` obtains gains of 2.8% and 3.4% on ZSL and GZSL tasks, respectively.

B Qualitative Analysis

B.1 Feature Visualization Comparison

Here, we present the implementation details and additional qualitative results for the visualization of synthesized features discussed in Sec. 4.2 of the paper.

Implementation details: The image generator, which inverts the feature instances to images of size 64x64, consists of a fully-connected (FC) layer followed by five upconvolutional blocks. Each upconvolutional block contains an Upsampling layer, a 3x3 convolution, BatchNorm and ReLu non-linearity. An ℓ_1 loss between the ground truth and inverted images, along with a perceptual loss (ℓ_2 loss between the corresponding feature vectors at conv5 of a pre-trained ResNet-101) and an adversarial loss are employed to construct good quality images. The discriminator, required for adversarial training, takes image and feature embedding as inputs. The input image is processed through four downsampling blocks to obtain an image embedding, while the feature embedding is passed through an FC layer and spatially replicated to match the spatial dimensions of the obtained image embedding. The resulting two embeddings are concatenated and passed through convolutional and *sigmoid* layers for predicting whether the input image is real or fake. The model is trained on all the real feature-image pairs of the 102 classes of FLO [29].

Visualization: The comparison between `Baseline` and our `Feedback` synthesized features on eight example flowers is shown in Fig. 7. For each flower class, a ground-truth (GT) image along with three images inverted from its GT feature, `Baseline` and `Feedback` synthesized features, respectively are shown. Generally, inverting the `Feedback` synthesized feature yields an image that is semantically closer to the GT image than inverting the `Baseline` synthesized feature. Inverting the feature instances from our `Feedback` improves the color of bud and shape of petals (*Californian poppy*, *Globe flower* and *Osteospermum*), structure of the flower (*Hippeastrum*), in comparison to the `Baseline` synthesized features. A considerable improvement for our `Feedback` over the `Baseline` is visible in these flowers (*Californian poppy*, *Globe flower*, *Hippeastrum* and *Osteospermum*). However, there are a few challenging cases (*e.g.*, *Globe thistle*, *Windflower*, *Sweet william*, *Moon orchid*), where a semantic gap still exists between the inversion of real features (denoted as `Reconstructed`) and inversion of `Feedback` synthesized features, even though there is a marginal improvement for our `Feedback` over the `Baseline`. These qualitative observations suggest that our `Feedback` improves the feature synthesis stage over the `Baseline`, where no feedback is present, resulting in improved zero-shot classification.

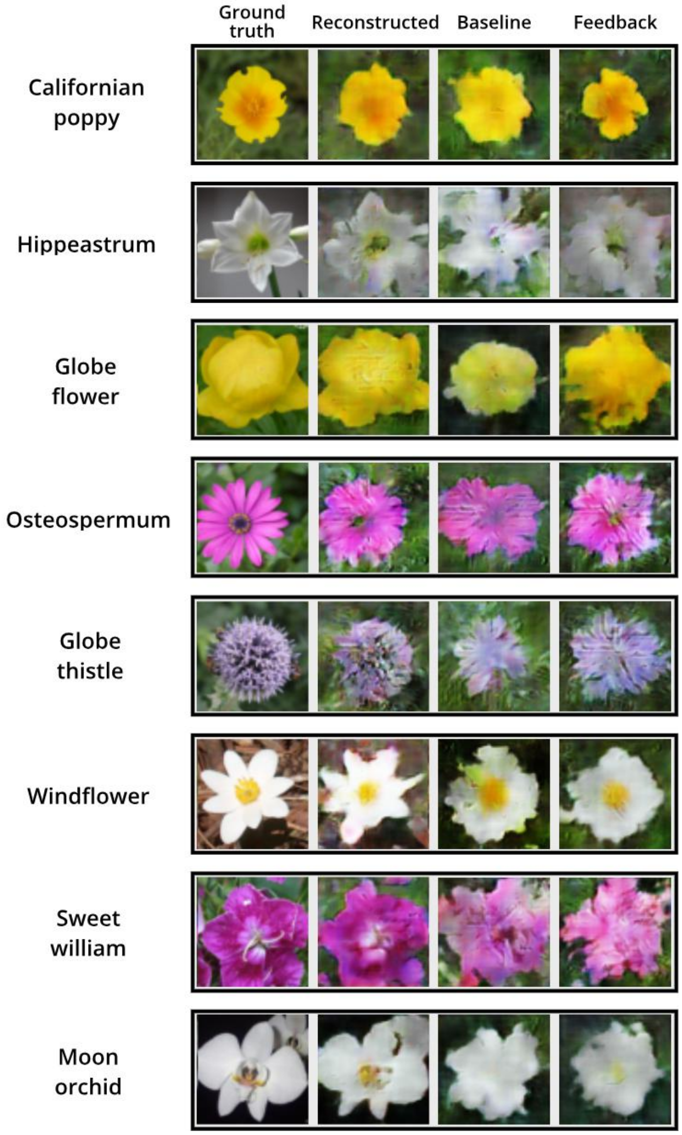


Fig. 7: **Qualitative comparison** between inverted images of `Baseline` synthesized features and our `Feedback` synthesized features on eight example classes of FLO [29]. The ground-truth image (GT) and the reconstructed inversion (`Reconstructed`) of its real feature are also shown for each example. Inverting the feature instances from our `Feedback` improves the color of bud and shape of petals (*Californian poppy*, *Globe flower* and *Osteospermum*), structure of the flower (*Hippeastrum*), in comparison to the `Baseline` synthesized features. Semantic gap still exists between the inversion of real features (denoted as `Reconstructed`) and inversion of `Feedback` synthesized features for a few challenging cases (*e.g.*, *Globe thistle*, *Windflower*, *Sweet william*, *Moon orchid*), even though there is some improvement for our `Feedback` over the `Baseline`. These observations suggest that our `Feedback` improves the quality of synthesized features over the `Baseline`, where no feedback is present. Best viewed in color and zoom.

B.2 Classification Performance Comparison

Here, we qualitatively illustrate the performance of our `TF-VAEGAN` framework, in comparison to the baseline `f-VAEGAN` [43] method, on two fine-grained object recognition datasets: CUB and FLO. Fig. 8 and 9 present the comparison on CUB and FLO, respectively. For each dataset, images from five most confusing categories (with respect to the baseline `f-VAEGAN`) are shown. The comparison is illustrated for five image instances in each category. The ground truth instances are shown in the top row for each category, followed by the classification results of the baseline and proposed frameworks in second and third rows, respectively. Correctly classified images are marked with a green border, while the incorrectly classified images are marked with a red border. For the misclassifications, the name of the incorrectly predicted class is denoted below the instance for the respective methods.

CUB: The qualitative comparison between the baseline and the proposed approaches for the CUB [40] dataset is shown in Fig. 8. Five categories of birds that are most confusing for the baseline approach are presented. The categories are *Prairie warbler*, *Great crested flycatcher*, *Grovve billed ani*, *Herring gull* and *California gull*. Generally, for all these categories, the baseline `f-VAEGAN` approach confuses with similar looking bird categories in the dataset. Our `TF-VAEGAN` reduces this confusion between similar looking classes and improves the classification performance. In Fig. 8, we observe that the baseline approach confuses *Prairie warbler* class with other similar looking *warbler* categories such as *Blue winged warbler*, *Magnolia warbler* and *Orange crowned warbler*. This confusion is reduced in the predictions of our `TF-VAEGAN`. Similarly, the confusion present, in the baseline method, between the *Great crested flycatcher* and other *flycatcher* categories is reduced for the proposed method. As a result, the overall classification performance improves for the proposed method over the baseline.

FLO: Fig. 9 shows the qualitative comparison for five categories of flowers from the Oxford Flowers [29] dataset that are most confusing for the baseline method. The categories are *Dafodil*, *Pink primrose*, *Siam tulip*, *King Protea* and *Common dandelion*. For all these categories, the proposed `TF-VAEGAN` reduces the confusion present between the similar looking classes in the baseline `f-VAEGAN` approach and improves the classification performance. In general, we observe that the instances are misclassified to other similar looking categories in the dataset. *E.g.*, instances of *Common dandelion* are commonly misclassified as either *Colt's foot* or *Yellow iris*. All three categories have yellow flowers and share similar appearance. We observe that the baseline makes confused predictions with respect to these classes. However, the confusion is less in the predictions of the proposed `TF-VAEGAN`. This leads to a favourable improvement in the zero-shot classification performance for the proposed approach. Similar observations can also be made in the case of other categories. The baseline `f-VAEGAN` generally confuses *Dafodil* with *Globe flower* and *Yellow iris* due to the yellow colour, while *Pink primrose* is mostly confused with *Petunia* and *Monkshood* due to the pinkish petals in the flowers. The misclassifications are reduced when using the proposed `TF-VAEGAN` for classification, resulting in an improved performance.

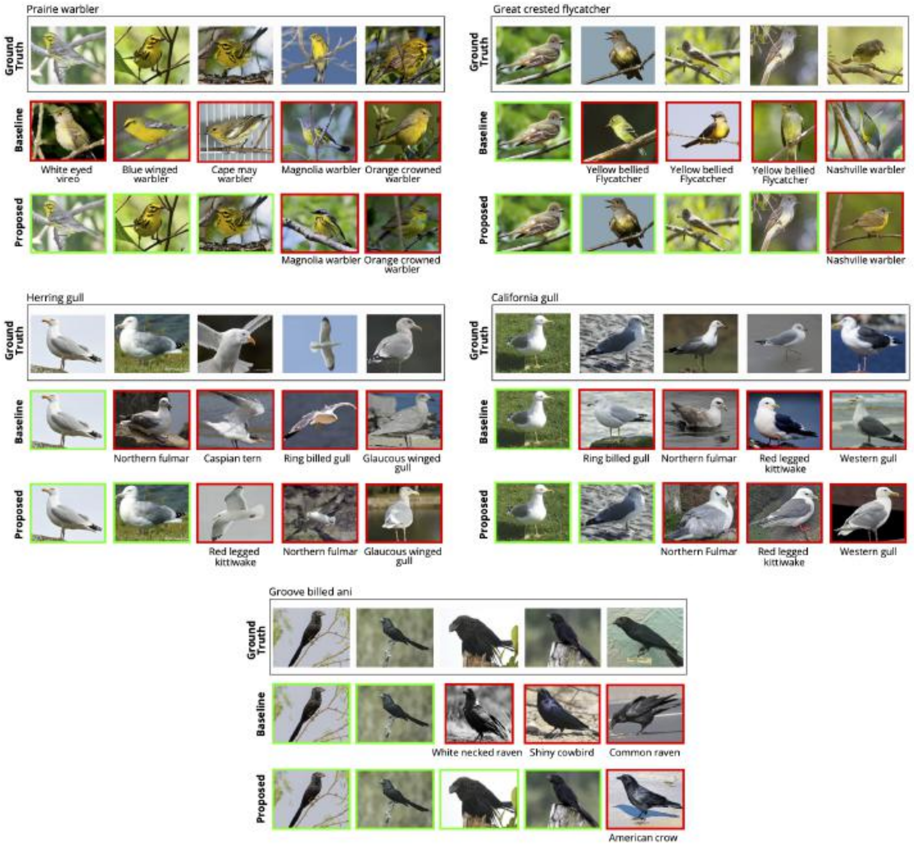


Fig. 8: Qualitative comparison between the baseline and our proposed approach on the CUB [40] dataset. The comparison is based on the most confusing categories as per the baseline performance. For each category, while the top row denotes different variations of ground truth class instances, the second and third rows show the classification predictions by the baseline and proposed approaches, respectively. The green and red boxes denote correct and incorrect classification predictions, respectively. The class names under each red box show the corresponding incorrectly predicted label. In general, we observe that the instances are misclassified to other similar looking categories in the dataset. For instance, *Prairie warbler* is confused with *Blue winged warbler*, while *Groove billed ani* is confused commonly with *Common raven*. For all these categories, the proposed `TF-VAEGAN` reduces the confusion among similar looking classes in the baseline `f-VAEGAN` and improves the classification performance over the baseline. See associated text for additional details. Best viewed in color and zoom.

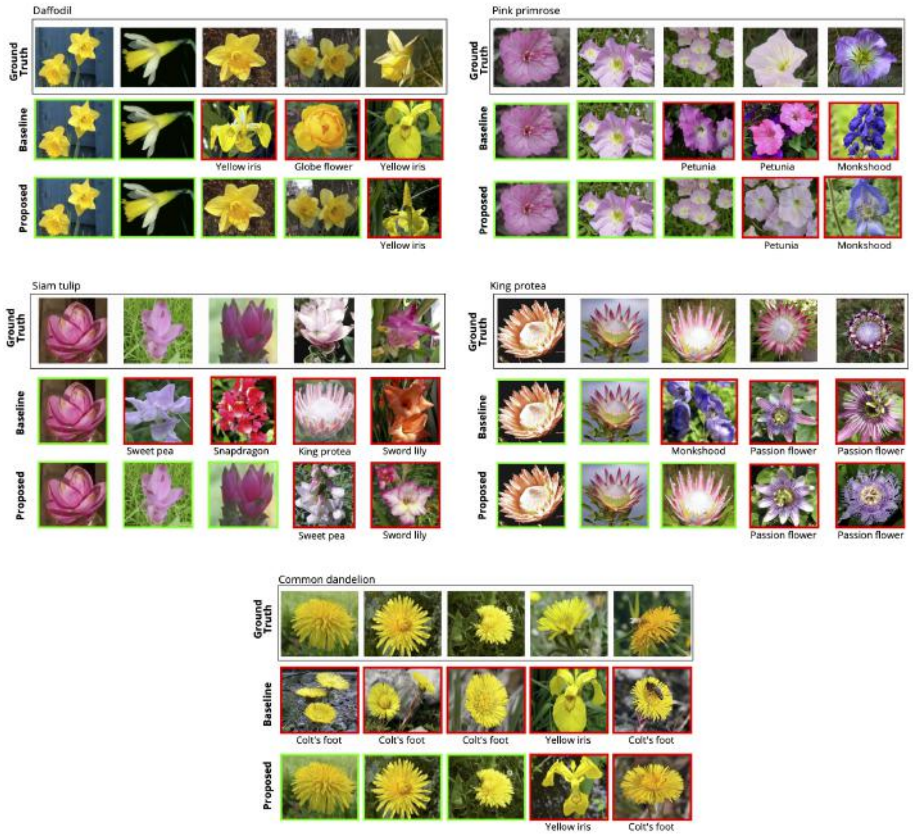


Fig. 9: Qualitative comparison between the baseline and our proposed approach on the Oxford Flowers [29] dataset. The comparison is based on the most confusing categories as per the baseline performance. For each category, while the top row denotes different variations of ground truth class instances, the second and third rows show the classification predictions by the baseline and proposed approaches, respectively. The green and red boxes denote correct and incorrect classification predictions, respectively. The class names under each red box show the corresponding incorrectly predicted label. In general, we observe that the instances are misclassified to other similar looking categories in the dataset. For instance, *Common dandelion* is confused with *Colt's foot*, while *Pink primrose* is confused with *Petunia*. For all these categories, the proposed **TF-VAEGAN** reduces the confusion among similar looking classes in the baseline **f-VAEGAN** and improves the classification performance over the baseline. See associated text for additional details. Best viewed in color and zoom.