# Bootstrapped Unseen Prototypes for Few-Shot Inspired Generative Zero-Shot Learning

Md Shakil Ahamed Shohag
Electrical and Computer Engineering
University of Windsor
Windsor, Ontario, Canada
Email: shakilshohag@gmail.com

Q. M. Jonathan Wu
Electrical and Computer Engineering
University of Windsor
Windsor, Ontario, Canada
Email: jwu@uwindsor.ca

Ashab Uddin
Electrical and Computer Engineering
University of Windsor
Windsor, Ontario, Canada
Email: uddin81@uwindsor.ca

Ning Zhang
Electrical and Computer Engineering
University of Windsor
Windsor, Ontario, Canada
Email: ning.zhang@uwindsor.ca

***Abstract*—Generative zero-shot learning (ZSL) synthesizes visual features for unseen classes from semantic descriptors, then trains a fully supervised classifier. Although effective, most methods depend on large volumes of synthetic data and heavyweight generators which dilutes the original ZSL premise. We propose BUP-FSIGenZ framework that approaches the generative ZSL by taking few-shot learning (FSL) as an inspiration instead of conventional supervised formulation. Consequently, the method focuses on generating only a handful of bootstrapped prototypes per unseen class. Instead of modeling full distributions with adversarial or variational generators, we expose the variability of seen classes using statistical resampling and estimate the same for unseen classes by knowledge transfer to unseen domain. Concretely, we bootstrap the seen data to obtain multiple class prototypes that capture stable yet diverse modes; and then estimate unseen bootstrapped prototypes through knowledge transfer from seen to unseen domain. This way we generate few prototypes for each unseen class and use them as unseen synthetic training data. For classification, we introduce a classifier trained jointly with binary cross-entropy and KL-divergence objectives on visual–semantic contrast. This unified design drastically reduces compute and sample count, and attains competitive ZSL performance on SUN, AWA2, and CUB with significantly fewer synthetic features than conventional generative baselines.**

***Index Terms*—Zero-shot learning, generalized zero-shot learning, knowledge transfer, feature synthesis**

## I. Introduction

Zero-shot learning (ZSL) is a machine learning paradigm that aims to recognize objects from unseen (target) classes that do not appear during training by transferring knowledge from seen (source) classes through semantic side information such as attributes or word vectors [1], [2]. This ability to generalize beyond labeled training classes makes ZSL particularly appealing in large-scale recognition scenarios, where collecting data for every possible category is impractical due to the scale and annotation cost of modern datasets [3]. Depending on how the test data are constructed, ZSL is commonly evaluated under two settings: *(i)* conventional ZSL (CZSL), in which test instances are drawn exclusively from unseen classes, and *(ii)* generalized ZSL (GZSL), a more realistic and challenging setting where test instances may belong to either seen or unseen classes [4], [5].

A long line of methods learns a compatibility function or a shared embedding space that aligns visual features with semantic representations, enabling classification by nearest-neighbor search or distance-based matching in the semantic space [2]. Although effective, embedding-based models are typically trained on seen data and hence develop a bias toward seen classes, which often leads to misclassifying unseen examples as seen in the GZSL setting [6]. This "seen-class bias" is a central obstacle to robust GZSL performance.

Generative approaches attempt to mitigate this bias by synthesizing visual features for unseen classes, effectively converting ZSL into a fully supervised problem [7]. Popular choices include Generative Adversarial Networks (GANs) [8] and Variational Autoencoders (VAEs) [9], which condition on class-level semantics to produce pseudo-examples for unseen classes. While successful, these methods often rely on large quantities of synthetic features and frequently select samples based on perceived "realness" rather than downstream utility [10]. The attendant computational burden and the difficulty of curating informative synthetic sets can limit scalability and interpretability.

Inspired by few-shot learning (FSL), we explore an alternative perspective: instead of flooding the learner with vast synthetic datasets, furnish it with a small number of prototype-like synthetic examples per unseen class. From this viewpoint, the goal is not to model the full data distribution for each class but to estimate a compact and discriminative set of representative points that capture the principal modes of variation relevant for recognition. Such a strategy preserves the benefits of generative ZSL (reducing seen-class bias by supplying unseen evidence) while markedly lowering computational cost and simplifying training.

In this paper we instantiate this idea with a lightweight and interpretable mechanism by utilizing multiple "means" (prototypes) for each seen class by bootstrapping the training data

and then estimate bootstrapped means for the unseen classes through seen-to-unseen knowledge transfer. These predicted means act as a few synthetic training examples—compact surrogates for the unseen distribution that can be fed to a carefully designed classifier. By design, this approach avoids heavy generative modeling and yields prototypes that are easy to inspect and reason about. An overview of the proposed method is presented in Fig. 1, and the complete procedure is outlined in Algorithm 1.

We evaluate our approach under both CZSL and GZSL on the standard SUN, AWA2, and CUB benchmarks. Despite generating only a handful of synthetic points per unseen class, our method attains competitive performance with state-of-the-art generative ZSL approaches. The main contributions are as follows:

- We approach generative ZSL through the lens of FSL, replacing large-scale synthetic datasets with a compact set of bootstrapped prototype-like synthetic examples per unseen class.
- We propose a unified ZSL framework that estimates multiple bootstrapped-like unseen means per class and utilize them as unseen training data to train a classifier specially designed to handle limited unseen data.
- Extensive experiments on SUN, AWA2, and CUB show that our method attains competitive ZSL results while using far fewer synthetic features than prior generative methods.

## II. Related Works

Embedding methods preserve the original training set of seen classes and aim to learn a projection or compatibility function that enables generalization to unseen categories. These approaches differ mainly in the space where classification occurs—visual, semantic, or a shared latent embedding space [5]. Visual embedding methods project semantic attributes into the visual domain, either by generating visual prototypes directly from semantic vectors [11] or by exploiting semantic relationships among classes to construct classifiers capable of extending to unseen categories [12], [13]. Semantic embedding methods instead map visual features into the semantic space using learned projection functions [14]. Latent embedding approaches jointly embed both visual and semantic representations into a shared latent space to bridge the gap between seen and unseen classes and facilitate transfer [15]. Overall, embedding-based ZSL methods focus on aligning visual and semantic modalities through appropriate projection functions, enabling recognition of unseen classes without synthesizing new visual data.

Generative methods synthesize visual features for unseen classes using seen class images and semantic information, effectively converting ZSL into a supervised learning problem. This approache leverages adversarial frameworks such as GANs conditioned on class semantics. For example, f-CLSWGAN [7] generates discriminative visual features by aligning synthesized samples with semantic attributes in a joint embedding space. Extensions such as cycle-CLSWGAN [16] impose semantic cycle consistency to better match generated and true feature distributions, while f-VAEGAN [17] merges the strengths of VAEs and GANs into a unified generative framework. Methods like Dual-VAEGAN [18], FREE [19], and CMC-GAN [20] further refine feature synthesis through enhanced semantic alignment, cross-dataset generalization, or hallucination strategies. ZeroGen [21] employs large language models to synthesize features directly from textual prompts, reducing reliance on heavy generative architectures. More recent methods integrate embedding models with generative mechanisms to improve robustness across seen and unseen classes [22]–[24]. For instance, CE-GZSL [22] combines contrastive embedding with class- and instance-level supervision, while RE-GZSL [24] extrapolates unseen features using semantic relations, contrastive losses, and feature mixing to produce more realistic and discriminative samples.

Non-adversarial generative methods offer an alternative to GAN-based feature synthesis by relying on sampling, interpolation, or statistical modeling rather than heavy adversarial training. These approaches generally provide better training stability, lower computational cost, and fewer issues such as mode collapse and high computational costs [25], [26]. For example, TDCSS [27] disentangles attribute-relevant and irrelevant components and controls pseudo-sample diversity to improve feature transferability. GG [28] models each class with a simple Gaussian distribution, generating features using estimated statistical parameters without any deep generator. Attribute-based models such as Composer [29] and ABS-Net [30] synthesize unseen examples by recombining or perturbing attribute patterns, while interpolation-based methods like BPL [31] and AGZSL [32] transfer class-specific variations from seen to unseen categories through projection and semantic enrichment techniques.

These non-adversarial strategies demonstrate that effective unseen feature synthesis does not require complex generative architectures; instead, lightweight statistical or compositional models can achieve strong performance with significantly lower computational overhead. Our method is conceptually aligned with this line of work, as it avoids adversarial generation and employs a simple, interpretable mechanism for producing unseen class prototypes.

## III. Approach

### A. Problem Settings

In ZSL framework, the label space is partitioned into two disjoint subsets: the seen classes $\mathcal{Y}^s$ with $K$ categories and the unseen classes $\mathcal{Y}^u$ with $L$ categories, where $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. The indices $\{1, \ldots, K\}$ correspond to seen classes, while $\{K+1, \ldots, K+L\}$ denote unseen ones. The training dataset comprises $N^s$ labeled samples from the seen classes, represented as $D = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathcal{X},\ y_i \in \mathcal{Y}^s\}_{i=1}^{N^s}$, where $\mathcal{X}$ denotes the visual feature space. No labeled instances from unseen categories are accessible during training. Each class $c \in \mathcal{Y}^s \cup \mathcal{Y}^u$ is associated with a semantic descriptor $\mathbf{a}_c$, and the complete attribute set is denoted by $\mathbf{A_p} = \{\mathbf{a}_c\}_{c=1}^{K+L}$, which encodes transferable semantic information between seen

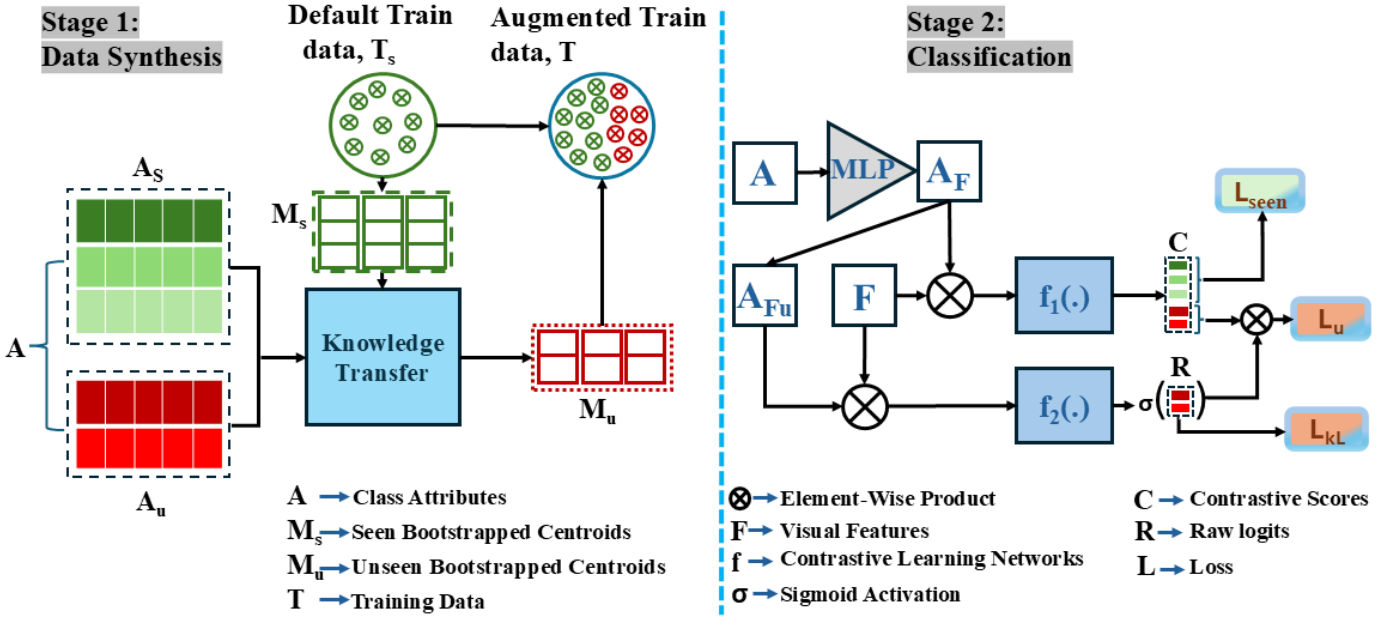


Fig. 1: Overview of the BUP-FSIGenZ framework. The model comprises two main phases: feature generation (on left of the dotted line) and visual-semantic contrastive learning (right). Green-toned elements denote seen classes, while red-toned elements denote unseen classes.

and unseen classes. The objective of CZSL is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}^u$ that predicts unseen categories only, while in the GZSL setting, the classifier must infer over both seen and unseen categories, i.e., $f : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$.

### *B. Attribute Rescoring*

We utilize the model-specific attribute scoring strategy introduced in [33] to adjust the class-level attributes as follows:

$$\mathbf{A} = (\mathbf{A_p} + \mathbf{A_q})W_a, \tag{1}$$

where $W_a$ is a scalar coefficient, $\mathbf{A_p}$ represents the original attribute matrix, and $\mathbf{A_q}$ comprises the elements of $\mathbf{A_p}$ that surpass a predefined threshold $t^h$, formulated as:

$$\mathbf{A_q} = \mathbf{A_p} \odot (\mathbf{A_p} > t_h), \tag{2}$$

with $\odot$ denoting the element-wise multiplication operator, and $(\mathbf{A_p} > t_h)$ yielding 1 when the condition is satisfied and 0 otherwise.

### *C. Unseen Data Synthesis*

We can estimate virtual class centers for unseen classes by transferring the semantic embedding manifold into the visual feature space [34], as follows:

$$A^u = \mathcal{R}_f(A^s) \Rightarrow M^u = \mathcal{R}_f(M^s), \tag{3}$$

where $A^s$ and $A^u$ are the attribute descriptions of seen and unseen classes, respectively; $M^s$ and $M^u$ correspond to their feature means; and $\mathcal{R}_s$ represents the relation function that maps seen classes to unseen ones.

In this formulation, the attribute embeddings $A^u$ for unseen classes are inferred by applying the relation function $\mathcal{R}_s$ to the attribute descriptions $A^s$ of seen classes. Likewise, the virtual class centers $M^u$ of unseen classes are computed by transferring the seen-class cluster centers $M^s$ through $\mathcal{R}_s$. The relation function $\mathcal{R}_s(\cdot)$ is learned using a Ridge-regularized linear reconstruction model, as:

$$\min_{\alpha} \|a_c^u - A^s\alpha\|_2^2 + \lambda\|\alpha\|_2^2, \tag{4}$$

where $\alpha = [\alpha_1, \ldots, \alpha_K]^T$ is the reconstruction coefficient vector, and $\lambda$ controls the degree of regularization.

Finally, the estimated cluster centers of the unseen classes are obtained as:

$$\mu_l^u = M^s\alpha, \tag{5}$$

where $\mu_l^u$ represents the estimated cluster center of the $l^{th}$ unseen class.

Estimating a single cluster center for each unseen class is relatively straightforward; however, the challenge arises when multiple centers must be generated per class. To address this, we adopt a bootstrapped mean estimation strategy for unseen classes. Specifically, we first compute $N_P^{SF}$ bootstrapped means for each seen class and group them into $N_P^{SF}$ batches, where each batch $M^{s_b}$ contains exactly one mean for every seen class. Using these batches, we then synthesize $N_P^{SF}$ synthetic features for each unseen class as follows:

$$\mu_l^{u_b} = M^{s_b}\alpha, \quad \text{for } b = 1, 2, \ldots, N_P^{SF} \tag{6}$$

where $\mu_l^{u_b}$ represents the estimated bootstrapped prototypes of unseen classes in $b^{th}$ batch .

**Algorithm 1** BUP-FSIGenZ

**Input**: Images $I$, Number of Unseen features per class $N_P^{SF}$, class attributes $\mathbf{A_p}$, initialization $\Theta$
**Output**: ZSL classification

*# Extracting Image Features*
**for** each image $i$ **do**
  Extract ViT features: $F_i^{S+U} \leftarrow \text{ViT}(I_i)$
**end for**
Rescore class attributes: $\mathbf{A} \leftarrow \mathbf{A_p}$
# Synthesizing Unseen Features
Compute the relation, $\alpha \leftarrow \mathcal{R}_f$ for a given $\lambda$
Obtain $N_P^{SF}$ batches of bootstrapped means for the seen classes
Estimate bootstrapped prototypes for unseen classes: $\mu_b^u \leftarrow M_b^s \alpha$, for $b = 1, 2, ..., N_P^{SF}$
Utilize $\mu_b^u$ as synthetic unseen features
Obtain augmented train data, $F$ by combining synthetic unseen data and real train data
*# Train the Classifier*
**for** each iteration $t$ **do**
  Encode class semantics with MLP: $\mathcal{E}_j \leftarrow \text{MLP}(\mathbf{a_j})$
  Compute instance to all class scores: $c_{ij} \leftarrow f_1(\mathcal{Z}_{ij})$ with $\mathcal{Z}_{ij} \leftarrow \mathcal{F}_i \otimes \mathcal{E}_j$
  Compute instance to unseen class scores: $c_{ij_u} \leftarrow f_2(\mathcal{Z}_{ij_u})$ with $\mathcal{Z}_{ij_u} \leftarrow \sigma(R_{ij_u}) \leftarrow \mathcal{F}_i \otimes \mathcal{E}_{j_u}$
  Optimize training objective, $\mathcal{L}$
**end for**
*# Perform Inference*
**for** a given test image $i$ **do**
  Predict CZSL class: $y_i^{CZSL} \leftarrow \arg\max_j \{c_{ij}\}_{j=K+1}^{K+L}$
  Predict GZSL class: $y_i^{GZSL} \leftarrow \arg\max_j \{c_{ij}\}_{j=1}^{K+L}$
**end for**

Substituting these diverse $M^{s_b}$ values into equation (5) results in $N_P^{SF}$ number of cluster centers per unseen class. These centers are visual prototypes, each capturing a different bootstrap statistics, act as unseen train data. In essence, the estimation of bootstrap statistics offers an efficient and principled way to simulate the natural heterogeneity within unseen classes—an idea that aligns with our few-shot-inspired approach.

### *D. Classifier Design*

Let $N^u$ is the number of bootstrapped unseen class means. These means are included as unseen class features with the real seen features and form a augmented training set. We define a one-hot class indicator vector $m_{ij}$ based on the ground-truth class label $y_i$ of the $i^{\text{th}}$ instance as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } y_i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The joint representation of an instance–class pair is then formulated as

$$\mathcal{Z}_{ij} = \mathcal{F}(\mathbf{x}_i) \otimes \mathcal{E}(\mathbf{a}_j), \tag{8}$$

where $\mathcal{F}(\mathbf{x}_i)$ denotes the visual feature of the $i^{\text{th}}$ instance, $\mathcal{E}(\mathbf{a}_j)$ represents the semantic embedding of class $j$, and $\otimes$ indicates element-wise multiplication. To quantify the compatibility between an instance $i$ and a class $j$, we define a contrastive score $c_{ij}$ as

$$c_{ij} = f_1(\mathcal{Z}_{ij}), \tag{9}$$

where $f_1(\cdot)$ denotes a contrastive learning function that measures the alignment between the fused visual–semantic representation $\mathcal{Z}_{ij}$ and the target class embedding.

Using these definitions, a classifier can be trained to learn visual–semantic alignment through the following contrastive loss function, computed over the unified dataset that includes both real and synthesized samples:

$$\mathcal{L}_{\mathcal{N}} = -\sum_{i=1}^{N^s+N^u} \sum_{j=1}^{K+L} m_{ij} \log(c_{ij}) + (1 - m_{ij}) \log(1 - c_{ij}), \tag{10}$$

This formulation does not explicitly address the issue of class imbalance—particularly among unseen categories—since the few-shot-inspired ZSL setting often yields a limited number of synthetic samples per unseen class.

*Training Strategy:* To mitigate this we decompose the above training objective into the following:

$$\mathcal{L} = \mathcal{L}\text{seen} + \beta \mathcal{L}\text{unseen}, \tag{11}$$

where $\beta$ is a hyperparameter that controls the contribution of the unseen-class loss.

*Computing $\mathcal{L}$seen:* For each training instance, whether originating from real data (seen classes) or synthesized samples (unseen classes), we partition the class indicator $m_{ij}$ and the contrastive score $c_{ij}$ into their respective seen and unseen components, denoted by superscripts $s$ and $u$. The first objective, $\mathcal{L}$seen, is a binary cross-entropy (BCE) loss applied to the model's predictions over seen classes, defined as:

$$\mathcal{L}\text{seen} = -\sum_{i=1}^{N^s+N^u} \sum_{j=1}^{K} m_{ij}^s \log(c_{ij}^s) + (1 - m_{ij}^s) \log(1 - c_{ij}^s), \tag{12}$$

*Computing $\mathcal{L}$unseen:* Now, to compute the loss using the model's prediction over unseen classes, we need to consider that the unseen class features are limited set of estimated bootstrapped class means and may not fully reflect the complexity of the real data. To mitigate this, we compute a unseen loss term as:

$$\mathcal{L}\text{unseen} = \mathcal{L}_{\mathcal{U}} + \eta \mathcal{L}_{\text{KL}}, \tag{13}$$

To compute $\mathcal{L}$unseen, we first employ another branch of contrastive learning that measures scores between an instance and unseen classes. To do that, we take the instance to unseen class joint representation from equation (8)

$$\mathcal{Z}_{ij_u} = \mathcal{F}(\mathbf{x}_i) \otimes \mathcal{E}(\mathbf{a}_{j_u}), \tag{14}$$

and compute

$$R_{ij_u} = f_2(\mathcal{Z}_{ij_u}), \tag{15}$$

TABLE I: Performance of different methods on CZSL (T1) and GZSL (H). The best and second-best results are shown in bold and underlined, respectively, while red and blue indicate the best and second-best results among generative approaches.

| | Method | SUN | | | | AwA2 | | | | CUB | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | T1 | U | S | H | T1 | U | S | H | T1 | U | S | H |
| **Embedding-based** | TCN [35] | 61.5 | 31.2 | 37.3 | 34.0 | 71.2 | 61.2 | 65.8 | 63.4 | 59.5 | 52.6 | 52.0 | 52.3 |
| | DAZLE [36] | - | 52.3 | 24.3 | 33.2 | - | 60.3 | 75.7 | 67.1 | 65.9 | 56.7 | 59.6 | 58.1 |
| | ViT-ZSL [37] | - | 44.5 | 55.3 | <u>49.3</u> | - | 51.9 | 90.0 | 65.8 | - | 67.3 | 75.2 | **71.0** |
| | MSDN [38] | 65.8 | 52.2 | 34.2 | 41.3 | 70.1 | 62.0 | 74.5 | 67.7 | 76.1 | 68.7 | 67.5 | 68.1 |
| | SCILM [39] | 62.4 | 24.8 | 32.6 | 28.2 | 71.2 | 48.9 | 77.8 | 60.1 | 52.3 | 24.5 | 54.9 | 33.8 |
| | DUET [6] | 64.4 | 45.7 | 45.8 | 45.8 | 69.9 | 63.7 | 84.7 | 72.7 | 72.3 | 62.9 | 72.8 | 67.5 |
| | BGSNet [40] | 63.9 | 45.2 | 34.3 | 39.0 | 69.1 | 61.0 | 81.8 | 69.9 | 73.3 | 60.9 | 73.6 | 66.7 |
| | PRZSL [41] | 64.2 | 53.6 | 37.7 | 44.4 | 73.6 | 65.8 | 77.8 | 71.3 | 77.1 | 68.8 | 63.7 | 66.2 |
| | ZS-VAT [42] | 62.6 | 45.6 | 33.8 | 38.8 | 72.2 | 59.9 | 80.8 | 68.8 | 75.2 | 67.5 | 68.1 | 67.8 |
| **Generative-based** | f-CLSWGAN [7] | 60.8 | 42.6 | 36.6 | 39.4 | - | - | - | - | 57.3 | 43.7 | 57.7 | 49.7 |
| | f-VAEGAN-D2 [17] | 64.7 | 45.1 | 38.0 | 41.3 | 71.1 | 57.6 | 70.6 | 63.5 | 61.0 | 48.4 | 60.1 | 53.6 |
| | OCD-CVAE [43] | 63.5 | 44.8 | 42.9 | 43.8 | 71.3 | 59.5 | 73.4 | 65.7 | 60.3 | 44.8 | 59.9 | 51.3 |
| | TF-VAEGAN [44] | 66.0 | 45.6 | 40.7 | 43.0 | 72.2 | 59.8 | 75.1 | 66.6 | 64.9 | 52.8 | 64.7 | 58.1 |
| | HSVA [45] | 63.8 | 48.6 | 39.0 | 43.3 | - | 56.7 | 79.8 | 66.3 | 62.8 | 52.7 | 58.3 | 55.3 |
| | TGMZ [46] | - | - | - | - | - | 64.1 | 77.3 | 70.1 | - | 60.3 | 56.8 | 58.5 |
| | GCM-CF [47] | - | 47.9 | 37.8 | 42.2 | - | 60.4 | 75.1 | 67.0 | - | 61.0 | 59.7 | 60.3 |
| | CE-GZSL [22] | 63.3 | 48.8 | 38.6 | 43.1 | 70.4 | 63.1 | 78.64 | 70.0 | 77.5 | 63.9 | 66.8 | 65.3 |
| | FREE [19] | - | 47.4 | 37.2 | 41.7 | - | 60.4 | 75.4 | 67.1 | - | 55.7 | 59.9 | 57.7 |
| | AGZSL [32] | 63.3 | 29.9 | 40.2 | 34.3 | 73.8 | 65.1 | 78.9 | 71.3 | 57.2 | 41.4 | 49.7 | 45.2 |
| | SE-GZSL [48] | - | 45.8 | 40.7 | 43.1 | - | 59.9 | 80.7 | 68.8 | - | 53.1 | 60.3 | 56.4 |
| | ICCE [23] | - | - | - | - | 72.7 | 65.3 | 82.3 | 72.8 | 78.4 | 67.3 | 65.5 | 66.4 |
| | TDCSS [27] | - | - | - | - | - | 59.2 | 74.9 | 66.1 | - | 44.2 | 62.8 | 51.9 |
| | LCR-GAN [49] | - | 57.6 | 43.8 | **49.8** | - | - | - | - | - | 53.6 | 67.5 | 59.7 |
| | DFCA-GZSL [50] | 62.6 | 48.9 | 38.8 | 43.3 | 74.7 | 66.5 | 81.5 | 73.3 | <u>80.0</u> | 70.9 | 63.1 | 66.8 |
| | RE-GZSL [24] | - | - | - | - | 73.1 | 67.7 | 81.1 | 73.8 | 78.9 | 72.3 | 62.4 | 67.0 |
| | AREES [51] | 64.3 | 51.3 | 35.9 | 42.2 | 73.6 | 57.9 | 77.0 | 66.1 | 65.7 | 53.6 | 56.9 | 55.2 |
| | JFGOPL [52] | - | 48.8 | 38.0 | 42.7 | - | 62.6 | 74.2 | 67.9 | - | 56.4 | 62.7 | 59.4 |
| | DENet [53] | - | 52.3 | 40.8 | 45.8 | - | 62.6 | 84.8 | 72.0 | - | 65.0 | 71.9 | 68.3 |
| | DPCN [54] | 63.8 | 48.1 | 39.4 | 43.3 | 70.6 | 65.4 | 78.6 | 71.4 | **80.1** | 72.7 | 65.7 | 69.0 |
| | Zheng et al. [55] | - | - | - | - | - | 63.3 | 74.0 | 68.2 | - | 71.0 | 65.7 | 68.3 |
| | FSIGenZ [33] | <u>67.8</u> | 42.5 | 49.9 | 45.9 | <u>75.0</u> | 67.6 | 82.3 | <u>74.2</u> | 73.0 | 65.9 | 72.7 | 69.1 |
| | BUP-FSIGenZ (Ours) | **67.9** | 42.6 | 50.1 | 46.0 | **77.4** | 68.3 | 84.2 | **75.4** | 73.1 | 64.6 | 75.0 | <u>69.4</u> |

where $f_2(\cdot)$ is the contrastive learning function that outputs raw contrastive logits, $R_{ij_u}$ that pass through sigmoid activations denoted by $\sigma(\cdot)$:

$$c_{ij_u} = \sigma(R_{ij_u}). \tag{16}$$

Then, we compute

$$\mathcal{L}_{\mathcal{U}} = -\sum_{i=1}^{N^s+N^u}\sum_{j=K+1}^{L} m_{ij}^u \log(c_{ij}^u c_{ij_u}) + (1-m_{ij}^u)\log(1-c_{ij}^u c_{ij_u}), \tag{17}$$

Now, we compute class-to-class semantic similarities by solving the following optimization problem:

$$s_p = \arg\min_{s_p} \left\| \mathbf{a}_p - \sum_{q=1}^{K+L} \mathbf{a}_q s_{pq} \right\|_2^2 + \phi\|s_p\|_2 \tag{18}$$

where $\mathbf{a}_p$ represents the semantic embedding of class $p$, and $s_{pq}$ is the $q^{\text{th}}$ element of the similarity vector $s_p$, representing the semantic proximity between class $p$ and class $q$. The regularization parameter $\phi$ prevents trivial solutions by discouraging any single similarity score, particularly self-similarity, from dominating. Once the similarity vector is obtained, we apply temperature-scaled normalization to its unseen portion:

$$\tilde{s}_{pq}^u = \frac{\exp\left(\frac{s_{pq}^u}{\tau}\right)}{\sum_{q'=K+1}^{K+L}\exp\left(\frac{s_{pq'}^u}{\tau}\right)}, \tag{19}$$

$R_{ij_u}$ is also normalized via a temperature-scaled softmax:

$$\tilde{r}_{ij_u} = \frac{\exp\left(\frac{r_{ij_u}}{\tau}\right)}{\sum_{j'_u=K+1}^{K+L}\exp\left(\frac{r_{ij'_u}}{\tau}\right)}, \tag{20}$$

We compute the temperature-scaled knowledge distillation loss by minimizing the Kullback–Leibler (KL) divergence as follows:

$$\mathcal{L}_{\text{KL}} = \tau^2 D_{\text{KL}}\left(\tilde{R} \,\|\, \tilde{S}^u\right), \tag{21}$$

### *E. Zero-Shot Recognition*

Zero-shot recognition is performed using $f_1$ network by evaluating contrastive scores between the visual feature of an input image and the semantic embeddings of all candidate

classes. In the CZSL setting, predictions are restricted to unseen classes, and each image is assigned to the unseen class with the highest contrastive score. In contrast, the GZSL setting considers both seen and unseen classes, and the predicted label corresponds to the class with the maximum contrastive score across all categories.

$$\mathcal{P}_{czsl}(x_i) = \max_j \{c_{ij}\}_{j=K+1}^{K+L}. \tag{22}$$

$$\mathcal{P}_{gzsl}(x_i) = \max_j \{c_{ij}\}_{j=1}^{K+L}. \tag{23}$$

## IV. Experimental studies

This section presents the experimental setup, including the datasets, evaluation protocols, and implementation details. It then reports the experimental results and ablation studies.

### A. Experimental Setup

***Datasets.*** We evaluate our method on three widely used ZSL benchmarks: SUN [56], AwA2 [1], and CUB [57]. AwA2 is a medium-scale, coarse-grained dataset containing 37,322 images from 50 animal categories described by 85 attributes. CUB is a fine-grained bird classification dataset with 11,788 images spanning 200 species, annotated with 312 attributes. SUN is another fine-grained benchmark, consisting of 14,340 images across 717 scene classes, each represented by 102 attributes.

***Evaluation Protocols.*** We assess performance under both CZSL and GZSL settings. For CZSL, we report the average per-class Top-1 accuracy (T1) on unseen classes. For GZSL, we compute Top-1 accuracies on seen (S) and unseen (U) classes, and summarize overall performance using the harmonic mean, $H = 2 \times \frac{S \times U}{S+U}$, which reflects the balance between seen and unseen class recognition.

***Implementation Details.*** We extract 786-dimensional image features using the ViT-Base backbone [58] pre-trained on ImageNet-1k, and utilize the class attributes provided in [1]. For attribute rescoring, the threshold–weight pairs $(t_h, W_a)$ are set to (0.005, 0.8), (0.005, 0.4), and (0.3, 0.7) for the SUN, AwA2, and CUB datasets, respectively. Our classifier has a two-layer fully connected network that projects class semantic vectors into the feature space, with 1024 units in the first layer and 786 units in the second. It further employs two contrastive learning networks, $f_1$ and $f_2$, each consisting of a fully connected layer with a 1024-dimensional hidden representation. The network $f_1$ outputs a single-dimensional sigmoid score, whereas $f_2$ outputs raw logits. The MLP classifier uses ReLU and Leaky ReLU activations, while $f_1$ adopts ReLU followed by a sigmoid activation, and $f_2$ applies ReLU only in its hidden layer. The hyperparameters $\beta$ and $\eta$ are selected empirically, with $\beta = 0.2$ for all datasets, and $\eta = 0.4$ for SUN and CUB and $0.5$ for AwA2. The temperature $\tau$ is set to 0.6, 0.8 and 0.6 for SUN, AwA2, and CUB, respectively.

### B. Comparison With State-of-the-Art Methods

Table I provides a detailed comparison of state-of-the-art ZSL approaches on the SUN, AwA2, and CUB benchmarks. BUP-FSIGenZ attains the best CZSL performance on SUN (67.9%) and AwA2 (77.4%), and achieves competitive accuracy on CUB (73.1%). Under the GZSL setting, it obtains harmonic mean (H) scores of 75.4% on AwA2, 69.4% on CUB, and 46.0% on SUN—ranking first, second, and third on these datasets, respectively. No other method in the comparison demonstrates this level of consistent strength across all benchmarks; most alternatives excel only on particular datasets. Moreover, several methods that surpass BUP-FSIGenZ on isolated GZSL results—such as LCR-GAN (49.8%) and ViT-ZSL (49.3%) on SUN, or ViT-ZSL (71.0%) on CUB—do not report CZSL scores, limiting their overall evaluability. In contrast, BUP-FSIGenZ delivers robust performance across both CZSL and GZSL settings, making it the most consistently effective method in Table I.

Compared with other generative approaches, BUP-FSIGenZ attains the highest CZSL Top-1 accuracy on SUN (67.9%) and AwA2 (77.4%), as well as the best GZSL harmonic mean on AwA2 (75.4%) and CUB (69.4%). On SUN, it also achieves the second-highest harmonic mean (46.0%), trailing only LCR-GAN (49.8%). Notably, although LCR-GAN achieves a higher harmonic mean on SUN, its CZSL Top-1 accuracy is lower than that of BUP-FSIGenZ, suggesting weaker performance on the core ZSL task. A similar pattern appears on CUB, where DPCN reports the best CZSL accuracy (80.1%), yet BUP-FSIGenZ slightly surpasses it in harmonic mean (69.4% vs. 69.0%). In addition to its strong predictive performance, BUP-FSIGenZ is also highly efficient in feature synthesis. It generates only 15, 90, and 10 synthetic features per class for SUN, AwA2, and CUB, amounting to 1080, 900, and 500 total synthetic features—orders of magnitude fewer than those generated by competing models (see Table II). For example, LCR-GAN and DPCN synthesize up to 43,200 and 15,000 unseen features on SUN and CUB, respectively. Despite using dramatically fewer synthetic samples, BUP-FSIGenZ matches or outperforms these methods across ZSL metrics. This efficiency reflects a different design philosophy: rather than reframing ZSL as a large-scale supervised learning problem, BUP-FSIGenZ adopts a more FSL-oriented perspective, relying on a small number of features that effectively capture the underlying structure of unseen classes.

### C. Ablation Study and Sensitivity Analysis

**Ablation Study.** To better understand the contribution of each component in BUP-FSIGenZ, we conduct ablation studies by removing the attribute rescoring mechanism, the unseen-class loss $\mathcal{L}_{\text{unseen}}$, and the KL-alignment loss $\mathcal{L}_{\text{KL}}$. The results on SUN, AWA2, and CUB are reported in Table III. Removing attribute rescoring leads to consistent drops in both T1 and H across all datasets e.g., SUN: T1 decreases from 67.9% to 66.0% and H decreases from 46.0% to 41.4%; CUB: T1 from 73.1 to 72.1 and H from 69.4% to 66.7%. Excluding $\mathcal{L}_{\text{unseen}}$ results in the most severe degradation, especially on

TABLE II: Statistics of unseen data across different methods after incorporating synthetic features. Here, $N^{USF}$ denotes the number of synthetic unseen features, $N^{URF}$ represents the number of real unseen features, and $N_P^{SF}$ indicates the number of synthetic features generated per class. The notation (U/S) specifies whether synthesis is performed only for unseen classes or for both seen and unseen classes.

| Method | SUN | | | | | AwA2 | | | | | CUB | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | T1 | H | $N^{USF}$ | $N^{URF}$ | $N_P^{SF}$ | T1 | H | $N^{USF}$ | $N^{URF}$ | $N_P^{SF}$ | T1 | H | $N^{USF}$ | $N^{URF}$ | $N_P^{SF}$ |
| HSVA [45] | 63.8 | 43.3 | 28800/14400 | 1440 | 400/200 (U/S) | - | 66.3 | 4000/2000 | 7913 | 400/200 (U/S) | 62.8 | 55.3 | 4000/2000 | 2967 | 400/200 (U/S) |
| CE-GZSL [22] | 63.3 | 43.1 | 7200 | 1440 | 100 (U) | 70.4 | 70.0 | 24000 | 7913 | 2400 (U) | 77.5 | 65.3 | 15000 | 2967 | 300 (U) |
| FREE [19] | - | 41.7 | 21600 | 1440 | 300 (U) | - | 67.1 | 46000 | 7913 | 4600 (U) | - | 57.7 | 35000 | 2967 | 700 (U) |
| ICCE [23] | - | - | - | - | - | 72.7 | 72.8 | 50000 | 7913 | 5000 (U) | 78.4 | 66.4 | 20000 | 2967 | 400 (U) |
| LCR-GAN [49] | - | 49.8 | 43200 | 1440 | 600 (U) | - | - | - | - | - | - | 59.7 | 20000 | 2967 | 400 (U) |
| RE-GZSL [24] | - | - | - | - | - | 73.1 | 73.8 | 50000 | 7913 | 5000 (U) | 78.9 | 67.0 | 20000 | 2967 | 400 (U) |
| DENet [53] | - | 45.8 | 7200 | 1440 | 100 (U) | - | 72.0 | 35000 | 7913 | 3500 (U) | - | 68.3 | 10000 | 2967 | 200 (U) |
| DPCN [54] | 63.8 | 43.3 | 5760 | 1440 | 80 (U) | 70.6 | 71.4 | 25000 | 7913 | 2500 (U) | 80.1 | 69.0 | 15000 | 2967 | 300 (U) |
| Zheng et al. [55] | - | - | - | - | - | - | 68.2 | 24000 | 7913 | 2400 (U) | - | 68.3 | 35000 | 2967 | 700 (U) |
| **BUP-FSIGenZ** (Ours) | 67.9 | 46.0 | 1080 | 1440 | 15 (U) | 77.4 | 75.4 | 900 | 7913 | 90 (U) | 73.1 | 69.4 | 500 | 2967 | 10 (U) |

TABLE III: Ablation studies for different components of BUP-FSIGenZ on the SUN, AWA2, and CUB datasets. The red and blue highlights are the best and second-best results, respectively.

| Method | SUN | | | | AwA2 | | | | CUB | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | T1 | U | S | H | T1 | U | S | H | T1 | U | S | H |
| BUP-FSIGenZ w/o attribute rescoring | 66.0 | 36.7 | 47.5 | 41.4 | 76.3 | 65.5 | 84.9 | 73.9 | 72.1 | 60.2 | 74.7 | 66.7 |
| BUP-FSIGenZ w/o $\mathcal{L}$unseen | 65.8 | 20.2 | 51.0 | 28.9 | 67.8 | 8.6 | 88.0 | 15.7 | 69.4 | 32.5 | 84.1 | 46.9 |
| BUP-FSIGenZ w/o $\mathcal{L}_{\text{KL}}$ | 66.1 | 30.1 | 50.8 | 37.8 | 77.5 | 67.9 | 83.4 | 74.8 | 72.2 | 64.2 | 69.6 | 66.8 |
| BUP-FSIGenZ (full) | 67.9 | 42.6 | 50.1 | 46.0 | 77.4 | 68.3 | 84.2 | 75.4 | 73.1 | 64.6 | 75.0 | 69.4 |

AWA2 where unseen accuracy falls from 68.3% to 8.6%. Eliminating the KL-alignment loss $\mathcal{L}_{\text{KL}}$ also harms performance (e.g., CUB: T1 drops from 73.1% to 72.2% and H drops from 69.4% to 66.8%), demonstrating that distributional regularization improves semantic–visual alignment. Overall, the full BUP-FSIGenZ model achieves the strongest results on nearly all metrics, confirming that each component contributes complementary benefits.

**Impact of Synthetic Features.** Across all datasets, the number of synthesized unseen-class features consistently affects both T1 and H (see Fig. 2). On SUN, T1 peaks at 15 instances and then stabilizes, while H follows a similar upward trend before gradually declining. For AwA2, both metrics increase steadily as more synthetic features are generated, reaching their highest values at 90 instances before slightly dropping. On CUB, T1 and H achieve their best performance at 10 instances, with mild decreases afterward. Overall, these results indicate that balanced feature synthesis improves both T1 and H, though the optimal number of synthesized samples is dataset-dependent.

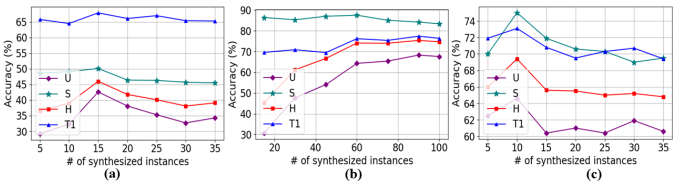


Fig. 2: Results with varying synthetic instances for unseen classes of the (a) SUN, (b) AwA2, and (c) CUB datasets.

## V. Conclusion

In this paper, we present BUP-FSIGenZ, a generative zero-shot learning framework that synthesizes a compact set of diverse prototypes per unseen class, in contrast to conventional approaches that rely on large volumes of synthetic data. It transfers the statistical bootstrapping of seen class features to unseen domain utilizing class semantics to synthesize unseen prototypes. To address the inherent class imbalance arising from the small number of generated prototypes, we employ a carefully designed contrastive classifier that enhances discrimination between seen and unseen classes. Overall, BUP-FSIGenZ supports that few-shot inspired generative modeling can serve as a powerful alternative to traditional large-scale generation by achieving competitive performance on SUN, AwA2, and CUB.

## References


[1] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, 2018.

[2] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 453–465, 2013.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Ieee, 2009, pp. 248–255.

[4] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 52–68.

[5] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim, X.-Z. Wang, and Q. J. Wu, "A review of generalized zero-shot learning methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4051–4070, 2022.

[6] Z. Chen, Y. Huang, J. Chen, Y. Geng, W. Zhang, Y. Fang, J. Z. Pan, and H. Chen, "Duet: Cross-modal semantic grounding for contrastive zero-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 405–413.

[7] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5542–5551.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[10] S. N. Gowda, "Synthetic sample selection for generalized zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 58–67.

[11] Y. Annadani and S. Biswas, "Preserving semantic relations for zero-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7603–7612.

[12] Y. Geng, J. Chen, Z. Ye, Z. Yuan, W. Zhang, and H. Chen, "Explainable zero-shot learning via attentive graph convolutional network and knowledge graphs," *Semant. Web*, vol. 12, no. 5, pp. 741–765, 2021.

[13] Y. Liu, X. Gao, Q. Gao, J. Han, and L. Shao, "Label-activating framework for zero-shot learning," *Neural Netw.*, vol. 121, pp. 1–9, 2020.

[14] F. Zhang and G. Shi, "Co-representation network for generalized zero-shot learning," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 7434–7443.

[15] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 4166–4174.

[16] R. Felix, I. Reid, G. Carneiro *et al.*, "Multi-modal cycle-consistent generalized zero-shot learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 21–37.

[17] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 275–10 284.

[18] Y. Luo, X. Wang, and F. Pourpanah, "Dual vaegan: A generative model for generalized zero-shot learning," *Appl. Soft Comput.*, vol. 107, p. 107352, 2021.

[19] S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao, "Free: Feature refinement for generalized zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 122–131.

[20] F.-E. Yang, Y.-H. Lee, C.-C. Lin, and Y.-C. F. Wang, "Semantics-guided intra-category knowledge transfer for generalized zero-shot learning," *Int. J. Comput. Vis.*, vol. 131, no. 6, pp. 1331–1345, 2023.

[21] J. Ye, J. Gao, Q. Li, H. Xu, J. Feng, Z. Wu, T. Yu, and L. Kong, "Zerogen: Efficient zero-shot learning via dataset generation," *arXiv preprint arXiv:2202.07922*, 2022.

[22] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2371–2381.

[23] X. Kong, Z. Gao, X. Li, M. Hong, J. Liu, C. Wang, Y. Xie, and Y. Qu, "En-compactness: Self-distillation embedding & contrastive generation for generalized zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9306–9315.

[24] Y. Wu, X. Kong, Y. Xie, and Y. Qu, "Re-gzsl: Relation extrapolation for generalized zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, 2024.

[25] W. Guo, J. Wang, and S. Wang, "Deep multimodal representation learning: A survey," *Ieee Access*, vol. 7, pp. 63 373–63 394, 2019.

[26] A. Jahanian, L. Chai, and P. Isola, "On the" steerability" of generative adversarial networks," *arXiv preprint arXiv:1907.07171*, 2019.

[27] Y. Feng, X. Huang, P. Yang, J. Yu, and J. Sang, "Non-generative generalized zero-shot learning via task-correlated disentanglement and controllable samples synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9346–9355.

[28] J. Cavazza, V. Murino, and A. Del Bue, "No adversaries to zero-shot learning: Distilling an ensemble of gaussian feature generators," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[29] D. Huynh and E. Elhamifar, "Compositional zero-shot learning via fine-grained dense feature composition," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 19 849–19 860.

[30] J. Lu, J. Li, Z. Yan, F. Mei, and C. Zhang, "Attribute-based synthetic network (abs-net): Learning more from pseudo feature representations," *Pattern Recognit.*, vol. 80, pp. 129–142, 2018.

[31] J. Guan, Z. Lu, T. Xiang, A. Li, A. Zhao, and J.-R. Wen, "Zero and few shot learning with semantic feature synthesis and competitive learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2510–2523, 2020.

[32] Y.-Y. Chou, H.-T. Lin, and T.-L. Liu, "Adaptive and generative zero-shot learning," in *Int. Conf. Learn. Represent*, 2021.

[33] M. S. A. Shohag, Q. Wu, and F. Pourpanah, "Few-shot inspired generative zero-shot learning," *arXiv preprint arXiv:2507.01026*, 2025.

[34] B. Zhao, B. Wu, T. Wu, and Y. Wang, "Zero-shot learning posed as a missing data problem," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 2616–2622.

[35] H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9765–9774.

[36] D. Huynh and E. Elhamifar, "Fine-grained generalized zero-shot learning via dense attribute-based attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4483–4493.

[37] F. Alamri and A. Dutta, "Multi-head self-attention via vision transformer for zero-shot learning," *arXiv preprint arXiv:2108.00045*, 2021.

[38] S. Chen, Z. Hong, G.-S. Xie, W. Yang, Q. Peng, K. Wang, J. Zhao, and X. You, "Msdn: Mutually semantic distillation network for zero-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7612–7621.

[39] Z. Ji, X. Yu, Y. Yu, Y. Pang, and Z. Zhang, "Semantic-guided class-imbalance learning model for zero-shot image classification," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6543–6554, 2022.

[40] Y. Li, Z. Liu, X. Chang, J. McAuley, and L. Yao, "Diversity-boosted generalization-specialization balancing for zero-shot learning," *IEEE Trans. Multimedia*, vol. 25, pp. 8372–8382, 2023.

[41] Y. Yi, G. Zeng, B. Ren, L. T. Yang, B. Chai, and Y. Li, "Prototype rectification for zero-shot learning," *Pattern Recognit.*, vol. 156, p. 110750, 2024.

[42] Z. Han, Z. Fu, S. Chen, L. Hui, G. Li, J. Yang, and C. W. Chen, "Zs-vat: Learning unbiased attribute knowledge for zero-shot recognition through visual attribute transformer," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 7025–7036, 2025.

[43] R. Keshari, R. Singh, and M. Vatsa, "Generalized zero-shot learning via over-complete distribution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13 300–13 308.

[44] S. Narayan, A. Gupta, F. S. Khan, C. G. Snoek, and L. Shao, "Latent embedding feedback and discriminative features for zero-shot classification," in *Proc. Eur. Conf. Comput. Vis*. Springer, 2020, pp. 479–495.

[45] S. Chen, G. Xie, Y. Liu, Q. Peng, B. Sun, H. Li, X. You, and L. Shao, "Hsva: Hierarchical semantic-visual adaptation for zero-shot learning," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 16 622–16 634.

[46] Z. Liu, Y. Li, L. Yao, X. Wang, and G. Long, "Task aligned generative meta-learning for zero-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 10, 2021, pp. 8723–8731.

[47] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang, "Counterfactual zero-shot and open-set visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 404–15 414.

[48] J. Kim, K. Shim, and B. Shim, "Semantic feature extraction for generalized zero-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, 2022, pp. 1166–1173.

[49] Y. Ye, T. Pan, T. Luo, J. Li, and H. T. Shen, "Learning mlatent representations for generalized zero-shot learning," *IEEE Trans. Multimedia*, vol. 25, pp. 2252–2265, 2023.

[50] H. Su, J. Li, K. Lu, L. Zhu, and H. T. Shen, "Dual-aligned feature confusion alleviation for generalized zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3774–3785, 2023.

[51] Y. Liu, Y. Dang, X. Gao, J. Han, and L. Shao, "Zero-shot learning with attentive region embedding and enhanced semantics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 4220–4231, 2024.

[52] X. Li, M. Fang, and Z. Zhai, "Joint feature generation and open-set prototype learning for generalized zero-shot open-set classification," *Pattern Recognit.*, vol. 147, p. 110133, 2024.

[53] J. Ge, H. Xie, P. Li, L. Xie, S. Min, and Y. Zhang, "Towards discriminative feature generation for generalized zero-shot learning," *IEEE Trans. Multimedia*, 2024.

[54] H. Jiang, Z. Li, Y. Hu, B. Yin, J. Yang, A. van den Hengel, M.-H. Yang, and Y. Qi, "Dual prototype contrastive network for generalized zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 2, pp. 1111–1122, 2025.

[55] B. Zheng, Z. Li, and J. Li, "Class-wise and instance-wise contrastive learning for zero-shot learning based on vaegan," *Expert Syst. Appl.*, p. 126671, 2025.

[56] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 2751–2758.

[57] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.