Evolutionary Generalized Zero-Shot Learning

Dubing Chen[1] , **Chenyi Jiang**[1] and **Haofeng Zhang**[1*]
[1]School of Artificial Intelligence, Nanjing University of Science and Technology
{db.chen, JiangChenyi, zhanghf}@njust.edu.cn

Abstract

Attribute-based Zero-Shot Learning (ZSL) has revolutionized the ability of models to recognize new classes not seen during training. However, with the advancement of large-scale models, the expectations have risen. Beyond merely achieving zero-shot generalization, there is a growing demand for universal models that can continually evolve in expert domains using unlabeled data. To address this, we introduce a scaled-down instantiation of this challenge: Evolutionary Generalized Zero-Shot Learning (EGZSL). This setting allows a low-performing zero-shot model to adapt to the test data stream and evolve online. We elaborate on three challenges of this special task, *i.e.*, catastrophic forgetting, initial prediction bias, and evolutionary data class bias. Moreover, we propose targeted solutions for each challenge, resulting in a generic method capable of continuous evolution from a given initial IGZSL model. Experiments on three popular GZSL benchmark datasets demonstrate that our model can learn from the test data stream while other baselines fail. The codes are available at https://github.com/cdb342/EGZSL.

1 Introduction

In the era of large-scale models, it is critical that systems learn autonomously from data without human supervision, generalize to new concepts, and minimize data-induced biases [Radford *et al.*, 2021; Ferrara, 2023; Burns *et al.*, 2023]. Traditional attribute-based zero-shot learning (ZSL) [Lampert *et al.*, 2009; Farhadi *et al.*, 2009] has instantiated these challenges on a smaller scale by utilizing attributes as intermediaries that enable models to recognize novel categories. However, conventional ZSL paradigms primarily engage in training static models, which struggle to correct prediction biases from unseen concepts and adapt to varying dynamic demands. Consequently, we ponder how a model trained on limited data can dynamically, cost-effectively, and efficiently self-evolve when exposed to data on novel concepts during deployment, making better decisions on unfamiliar concepts while maintaining its core capabilities.

In this paper, we build on the foundational principles of attribute-based ZSL and introduce a new setting: Evolutionary Generalized Zero-Shot Learning (EGZSL). We envisage a ZSL model that, after its initial training, can continually learn from a stream of unlabeled data. This model is designed to autonomously identify and adapt to unseen concepts, thereby evolving in conjunction with its underlying knowledge.

Distinct from existing ZSL settings (*i.e.*, inductive ZSL (IGZSL) [Chao *et al.*, 2016; Xian *et al.*, 2017] and transductive ZSL (TGZSL) [Kodirov *et al.*, 2015; Paul *et al.*, 2019; Wan *et al.*, 2019; Narayan *et al.*, 2020]), EGZSL allows for unsupervised online enhancement during deployment, enabling the model to perpetually evolve through exposure to an unlabeled test data stream. This makes EGZSL *(i) mitigate the domain shift problem [Fu* et al.*, 2014] by exposing the model to previously unseen class samples;* and *(ii) suitable for real-world deployment.* Fig. 1 briefly depicts the training and testing process of the proposed setting. At time 0, a base model is trained with the same settings as in IGZSL. At each subsequent time t, the model from time $t-1$ is first tested on the current batch, followed by unsupervised evolution. Unlike continual ZSL [Chaudhry *et al.*, 2019; Gautam *et al.*, 2020], we do not assume a fixed ratio of seen-unseen classes in each batch. Instead, data in each batch is randomly sampled from a mixture of both seen and unseen test sets. The model can only access the current data stream without having access to the base training data or the test data at other time stamps.

EGZSL meets three main challenges. First, the model is prone to catastrophic forgetting [McCloskey and Cohen, 1989; French, 1999] when training on streaming data. Second, due to the lack of unseen class samples in the base training phase, the prediction bias of the model is easily and consistently amplified when trained on the unlabeled data stream. Third, the model is vulnerable to potential data class imbalance problems. We then propose specific approaches to address these challenges. The overall framework is based on pseudo-label learning [Lee and others, 2013; Xie *et al.*, 2020], which is a common self-training approach for limited-supervised learning. We avoid forgetting by maintaining a global model, updated as the exponential moving average of the per-stage model. Historical information of the

*Corresponding author.

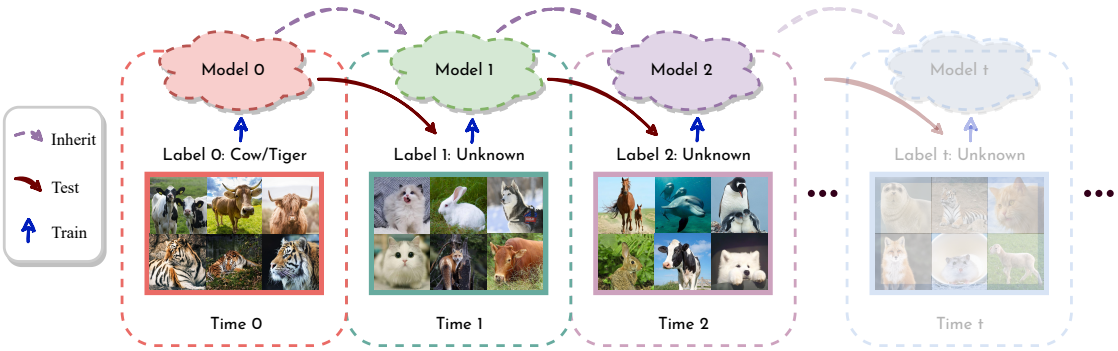


Figure 1: Illustration of the proposed EGZSL setting, featuring training with labeled seen class samples at time 0, followed by iterative predictive re-training on randomly divided data from the mixture of seen and unseen test sets in subsequent time steps (ratio of classes in a small batch is undefined). **Train:** train the current model at each time step with only the data indicated by the arrows. **Test:** predict the current data with the model obtained in the last time step. **Inherit:** train the current model based on the model of the last time step.

current model can be preserved by distilling it from the global model. We specify updateable class-related parameters for the class imbalance problem based on the data classes that occurred each time step. This avoids the error accumulation that causes predictions to deviate from certain classes. Moreover, we prevent confirmation bias by filtering noise labels. To avoid the effect of the initial prediction bias problem, we set a class-independent filtering threshold for each class. Finally, we propose evaluation criteria for this novel task, along with four baselines. The effectiveness of our proposed approach is demonstrated on three public ZSL benchmarks, on which the performance of our approach surpasses the base IGZSL method, while other baselines fail. Our contributions are summarized as follows:

- We establish a practical yet challenging evolutionary generalized zero-shot learning task, that is better suited for real-world applications than existing ZSL settings.
- We analyze the main challenges of EGZSL and propose a targeted approach to address each of them.
- We determine the evaluation criteria for EGZSL and conduct extensive experiments on three public ZSL datasets. The proposed method consistently improves over baselines. The effectiveness of our method is demonstrated by a series of explanatory experiments.

2 Related Work

Zero-Shot Learning (ZSL) [Lampert *et al.*, 2009; Lampert *et al.*, 2013; Xian *et al.*, 2017] aims at recognizing unseen classes when given only seen class samples. Early approaches [Akata *et al.*, 2013; Elhoseiny *et al.*, 2013; Frome *et al.*, 2013]. typically embedded images and semantic descriptors (*e.g.*, attributes, word vectors) to the same space, then conduct a nearest neighbor search. However, these methods were sensitive to the domain shift problem [Fu *et al.*, 2014]. They performed especially poorly in the **Generalized Zero-Shot Learning (GZSL)** setting [Chao *et al.*, 2016; Xian *et al.*, 2017], which requires classifying both seen and unseen classes in the test phase. In the follow-up research, [Xian *et al.*, 2018; Xian *et al.*, 2019; Shen *et al.*, 2020; Han *et al.*, 2021; Chen *et al.*, 2023] employed conditional generative models [Kingma and Welling, 2013; Arjovsky *et al.*, 2017; Dinh *et al.*, 2014] to generate pseudo-unseen class samples, thereby transferring GZSL into a supervised task. [Atzmon and Chechik, 2019; Chou *et al.*, 2021] distinguished seen or unseen classes with out-of-distribution detectors [Fang *et al.*, 2022], then classified in the corresponding subset of classes. [Xu *et al.*, 2020; Jiang *et al.*, 2024] emphasized learning a deep embedding model.

Transductive Zero-Shot Learning (TZSL) [Kodirov *et al.*, 2015; Wan *et al.*, 2019; Narayan *et al.*, 2020] assumed the unlabeled unseen test data is available during training. As a distinction, the earlier setting is called inductive ZSL. Existing methods [Fu *et al.*, 2014; Bo *et al.*, 2021] typically relied on pseudo-labeling strategies. [Xian *et al.*, 2019; Narayan *et al.*, 2020] also employed generative models. TZSL is a variant of semi-supervised learning [Grandvalet and Bengio, 2004] on ZSL. It mitigates the domain shift problem and yields better recognition performance. However, for ZSL, the accessibility of unseen class samples in training is a too strong hypothesis, leading to limited application scenarios.

Continual Zero-Shot Learning (CZSL) [Chaudhry *et al.*, 2019; Gautam *et al.*, 2020; Skorokhodov and Elhoseiny, 2021; Yi and Elhoseiny, 2021] extended traditional ZSL into a class-incremental paradigm [Rebuffi *et al.*, 2017]. A-GEM [Chaudhry *et al.*, 2019] marked the inception of lifelong learning within the ZSL framework, exploring the efficacy of continuous learning methods and introducing a pragmatic evaluation protocol where each example is encountered only once. [Wei *et al.*, 2020] refined this setup, proposing Lifelong Zero-Shot Learning, which sequentially learns from all seen classes across multiple datasets and evaluates on unseen data with the learned model. [Skorokhodov and Elhoseiny, 2021] extended it to unlimited label searching space and let the model recognize unseen classes sequentially. Many subsequent methods in CZSL have built upon this framework, with a focus on enhancing performance [Ghosh, 2021] or adapting to diverse applications. [Yi and Elhoseiny, 2021] extended the paradigm to various domains such as painting and sketching, introducing domain-aware continual zero-shot learning. In contrast, our EGZSL begins with a model trained on seen class samples, which are no longer accessible during

the evolutionary process. We target to recognize forthcoming data streams comprising both seen and unseen classes, iteratively refining the model's recognition capability over time.

Test Time Adaptation (TTA) [Sun *et al.*, 2020; Liu *et al.*, 2021; Wang *et al.*, 2021; Wang *et al.*, 2022] enables the model to better adapt the test domain by constructing self-supervised learning (SSL) tasks on test data. This concept has been further developed into an online framework for continual refinement. [Sun *et al.*, 2020] employed a rotation prediction task to update the model in test time, which also served as an auxiliary task in training. [Liu *et al.*, 2021] assessed TTA performance across various distribution shifts, and proposed to adopt contrastive learning as the SSL task. [Wang *et al.*, 2021] removed the auxiliary task during training and employed the minimum entropy strategy for optimization during testing. Existing TTA research has focused chiefly on the distribution shift task, *i.e.*, domain adaptation [Wang *et al.*, 2021; Wang *et al.*, 2022]. We adapt TTA's strategy of unsupervised self-training during the testing phase to extend traditional ZSL, providing a more realistic setting than TGZSL and mitigating the domain shift problem in IGZSL. Due to the extreme class imbalance problem in the base training phase, the proposed EGZSL faces specific challenges.

3 EGZSL: Settings and Challenges

In this section, we formulate the EGZSL setting, analyze its key challenges, and compare it to other limited-supervision or incremental learning tasks. Fig. 2 illustrates the differences between EGZSL and other similar settings.

3.1 Problem Formulation

EGZSL aims to evolve continually from a data stream. Let $\mathcal{Y}^s$ and $\mathcal{Y}^u$ denote two disjoint class label sets ($\mathcal{Y} = \mathcal{Y}^s \cup \mathcal{Y}^u$). $\mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ and $\mathcal{A} \subseteq \mathbb{R}^{d_\mathbf{a}}$ are feature space and attribute space, respectively, and d_x and d_a are dimensions of these two spaces. ZSL conventionally entails acquiring the associative relationship between visual features and semantic attributes to facilitate the transfer of knowledge to previously unseen classes. the goal of traditional GZSL is to learn such a classifier, *i.e.*, $f_{gzsl} : \mathcal{X} \rightarrow \mathcal{Y}^s \cup \mathcal{Y}^u$ given the training set $\mathcal{D}^{tr} = \{\mathbf{x}, y | \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}^s\}$ and the global semantic set $\mathcal{A}$.

In the initial phase (time 0), EGZSL endeavors to learn a foundational model f_0 using a base set $\mathcal{D}^b = \{\mathbf{x}_i, y_i, \mathbf{a}_{y_i} | \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}^s, \mathbf{a}_{y_i} \in \mathcal{A}\}_{i=1}^{N_b}$, where N_b represents the volume of data in the base set. The base model f_0 inherently possesses the capability to distinguish between both seen and unseen classes, denoted as $f_0 : \mathcal{X} \rightarrow \mathcal{Y}$. Subsequently, in time $1, ..., t, ..., T$, the base model undergoes testing and evolution on the test data streams $\mathcal{D}_1^{te}, ..., \mathcal{D}_t^{te}, ..., \mathcal{D}_T^{te}$, resulting in $f_1, ..., f_t, ..., f_T$, where $\mathcal{D}_t^{te} = \{\mathbf{x}_j, y_j | \mathbf{x}_j \in \mathcal{X}, y_j \in \mathcal{Y}\}_{j=1}^{N_t}$. Here, N_t denotes the data volume at time t. Notably, $\mathcal{D}_t^{te}$ is tested with f_{t-1}, and f_{t-1} is subsequently retrained with the unlabeled data in $\mathcal{D}_t^{te}$. The objective for f_t is to exhibit improved performance compared to f_{t-1} in classifying data with labels in $\mathcal{Y}$. Training in the labeled base set and the unlabeled sequential test set is referred to as base learning and evolutionary learning, respectively. Ultimately, EGZSL performance is assessed based on the test results across all test subsets.

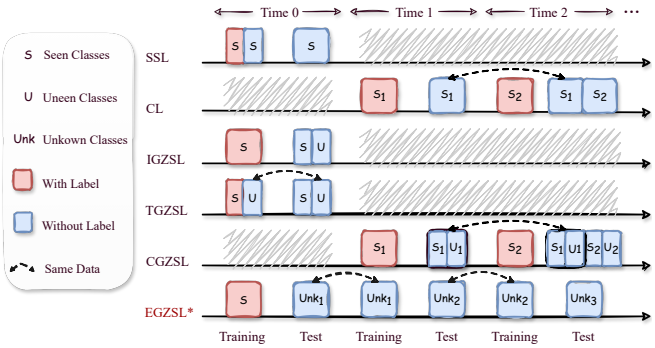


Figure 2: Comparasion of **EGZSL** with other similar settings in chronological progression. **TTA:** Test-Time Adaptation; **CL:** Continual Learning; **IGZSL:** Inductive Generalized Zero-Shot Learning; **TGZSL:** Transductive Generalized Zero-Shot Learning; **CGZSL:** Continual Generalized Zero-Shot Learning. In TTA, seen represents the source domain, and unseen is the target domain. In other settings, the labeled classes that appear in the training set are denoted as seen, and vice versa are unseen. An unknown class means that it can be any class (in seen or unseen classes).

3.2 Challenges Analysis

Catastrophic forgetting problem. When training on the one-time given data, it is able to repeatedly utilize the data to achieve the global optimum. On the contrary, there is a contradiction between falling into a local optimum and underutilization of data when dealing with incremental data streams. Since only a small amount of data can be accessed at a time, overfitting on this batch of data will lead to catastrophic forgetting [McCloskey and Cohen, 1989; French, 1999] the previously learned knowledge. Conversely, the batch of data cannot be fully utilized, resulting in low training efficiency. Finding a solution that balances efficient data utilization with preventing forgetting is imperative.

Initial bias problem. Unlabeled training on evolving data is heavily influenced by the accuracy of pseudo-labels predicted by the base model. However, when the base model is trained on imbalanced data classes, its prediction will be biased towards specific classes. This problem exists in EGZSL since the base set lacks unseen class samples. The prediction imbalance problem will cause error accumulation when continuing training with unbalanced pseudo-labels.

Sensitivity to data class bias. Since the EGZSL setting assumes arbitrary class distributions for the evolutionary learning phase, the model is at risk of being exposed to class-biased batch data. Training on biased data can lead to bias in the model predictions for pseudo-labeling the next phase, which in turn increases the model bias, ultimately causing a progressive impact on the sequential data.

3.3 EGZSL *v.s.* Similar Settings

As shown in Fig. 2, we compare EGZSL to existing settings. In detail, **IGZSL** depicts a static model that is trained once on seen classes, which does not adapt over time. **TGZSL** is similar to IGZSL but integrates (unlabeled) unseen class

data during training. **TTA** adapts at test time, but the model trains on a fixed set of classes and it does not continue to learn or adapt beyond time. **CL** progressively trains the model on different subsets of classes, testing it on both new and previously learned classes, challenging the model to remember old knowledge. **CGZSL** extends CL by including unseen classes in tests, pushing the model to adapt to new information constantly. Moreover, the proposed **EGZSL** raises the stakes, requiring the model to identify new classes, learn from ongoing unlabeled data flows, make sense of data without labels, and work without setting class limits at each time step. To put it in another perspective, EGZSL can be regarded as a strict version of TGZSL. TGZSL assumes that all labeled seen class data and unlabeled unseen class test data are given one-off, along with a fixed test set and known seen-unseen class splitting in the test set. In experiments, we consider IGZSL and TGZSL as upper and lower bounds for EGZSL to evaluate its performance since there is no existing method for EGZSL.

4 Method

EGZSL extends the IGZSL setting in test time. Any existing IGZSL models can be employed in our setting without retraining on the base data. Since base learning has been well studied, we focus mainly on the evolutionary learning phase. This section describes our method and explains each component that promotes evolutionary learning.

At each time step t, we first predict the pseudo-label of current data. To prevent catastrophic forgetting, we maintain a momentum model to preserve global data information and distill it to the current model. In addition, we select learnable class parameters to prevent class imbalance learning and filter unreliable data to avoid error accumulation. A specific training process is described in Algorithm 1.

4.1 Training with Pseudo Labels

Suppose a base model has been trained on the base set. We employ a pseudo-labeling strategy [Lee and others, 2013; Xie *et al.*, 2020] to enable continual improvement from the unlabeled data, which is a typical technique in semi-supervised learning [Lee and others, 2013] and domain adaptation [Wang *et al.*, 2021]. In time t, the pseudo label $\hat{y}_{\mathbf{x}}$ of a datum $\mathbf{x}$ is predicted with the highest compatibility with the model of the immediately preceding stage:

$$\hat{y}_{\mathbf{x}} = f_{t-1}(\mathbf{x}) = \underset{y}{\operatorname{argmax}}\, F_{t-1}(\mathbf{x}, y; \mathbf{W}_{t-1}). \tag{1}$$

Here, F_{t-1} measures the compatibility score between $\mathbf{x}$ and any class y, with $\mathbf{W}_{t-1}$ denoting its parameter. Based on the pseudo labels, we employ cross-entropy in label space $\mathcal{Y}$ to further optimize $\mathbf{W}_{t-1}$, *i.e.*,

$$\begin{aligned} \ell_{ce}(\mathbf{x}) &= -\log p_{t-1}(\mathbf{x}, \hat{y}_{\mathbf{x}}; \mathcal{Y}), \\ p_{t-1}(\mathbf{x}, y; \mathcal{Y}) &= \frac{\exp(F_{t-1}(\mathbf{x}, y; \mathbf{W}_{t-1}))}{\sum_{c\in\mathcal{Y}} \exp(F_{t-1}(\mathbf{x}, c; \mathbf{W}_{t-1}))}. \end{aligned} \tag{2}$$

4.2 Maintenance of Global Information

A challenge of EGZSL is the unavailability of all evolutionary data at one time. Directly updating the model with gradient descent at a certain time step can lead to catastrophic forgetting [McCloskey and Cohen, 1989; French, 1999], which is a typical difficulty in sequential learning. We resort to the momentum updated model as the surrogate of historical information, similar to MoCo [He *et al.*, 2020; Chen *et al.*, 2020]. Formally, the parameters of the momentum model undergo updates as exponential moving averages (EMA) based on the parameters of the model from the previous time step. Denoting F_{ema} as the momentum model with parameters $\mathbf{W}_{ema}$, at each time step t, $\mathbf{W}_{ema}$ is updated as

$$\mathbf{W}_{ema} = m_1 \cdot \mathbf{W}_{ema} + (1 - m_1) \cdot \mathbf{W}_t, \tag{3}$$

where $m_1 \in [0, 1)$ is a smoothing factor. We update $\mathbf{W}_{ema}$ with the gradient detached. $\mathbf{W}_{ema}$ is considered to retain the global information from past time steps, exhibiting smoother changes than $\mathbf{W}_t$. We distill [Hinton *et al.*, 2015] this information into the current model with Kullback-Leibler (KL) divergence:

$$\ell_{kl}(\mathbf{x}) = \sum_{y\in\mathcal{Y}} p_t(\mathbf{x}, y; \mathcal{Y}) \log \frac{p_t(\mathbf{x}, y; \mathcal{Y})}{p_{ema}(\mathbf{x}, y; \mathcal{Y})}. \tag{4}$$

Here $p_t(\mathbf{x}, y; \mathcal{Y})$ and $p_{ema}(\mathbf{x}, y; \mathcal{Y})$ denote the probability distribution over the variable y. Note that at each time step t, $\mathbf{W}_{ema}$ is updated after $\mathbf{W}_t$. With the distillation loss, the model can learn from the current data stream while avoiding catastrophic forgetting of previous information. This creates conditions for balancing the data utilization efficiency.

4.3 Class Selection for Stable Training

As discussed in Sec. 3.2, the potential imbalance of data classes in the evolutionary stage can lead to unbalanced predictions by the model, resulting in error accumulation. This problem becomes more pronounced when the number of samples available at each time step is small. It is prone to missing samples in certain classes, and cross-entropy based on pseudo-hard labels may produce sharper constraints that cause model predictions to abruptly deviate from these classes. To address this, we propose selecting specific class parameters to update at each time step. Consider typical GZSL classifiers are implemented with a linear model, *i.e.*, $\mathbf{W}$ is a matrix with $|\mathcal{Y}|$ rows and $d_{\mathbf{x}}$ columns:

$$F(\mathbf{x}, y; \mathbf{W}) := \mathbf{W}_y \cdot \mathbf{x}. \tag{5}$$

To ensure smoother updates of the weights for each class, at time step t, we choose to update only the classes present in the pseudo labels, *i.e.*,

$$\mathcal{Y}_t^{sel} = \operatorname{unique}(\{\hat{y}_{\mathbf{x}}\}_{\mathbf{x}\in\mathcal{D}_t^{te}}). \tag{6}$$

Here $\operatorname{unique}(\cdot)$ denotes the function that returns the unique elements of the label set, which can be achieved by directly calling the *PyTorch* function. Consequently, the cross-entropy loss in Eq. (2) is substituted with

$$\ell_{ce}^{sel}(\mathbf{x}) = -\log p_{t-1}(\mathbf{x}, \hat{y}_{\mathbf{x}}; \mathcal{Y}_t^{sel}). \tag{7}$$

Note that Eq. (4) is still computed with the full label set $\mathcal{Y}$.

4.4 Data Selection for Effective Training

Since the unlabeled data is trained using the pseudo-labeling strategy, noisy pseudo-labels will introduce confirmation bias. Hence, at each time step, we employ the model from the previous stage to select samples with low uncertainty. The uncertainty reflects the confidence level of the model prediction, typically measured by entropy or max softmax prediction [Mukhoti *et al.*, 2021]. We use the latter to select samples with more reliable pseudo labels. Intuitively, we can establish a constant threshold to filter the samples where softmax prediction fails to surpass this value, *i.e.*,

$$\mathtt{M}(\mathbf{x}) = \mathbb{1}(\max_{y} p_{t-1}(\mathbf{x}, y; \mathcal{Y}) > \tau), \tag{8}$$

where $\mathbb{1}$ denotes the indicator function, and $\tau \in (0, 1]$ is the predefined threshold. $\mathtt{M}(\cdot)$ enables the selection of the samples with high confidence. However, when softmax prediction values are unbalanced across classes, employing a fixed threshold results in unbalanced data selection. For instance, the base model is trained only on the seen classes, exhibiting higher confidence in samples from seen classes and lower confidence in those from unseen classes. Using the fixed threshold approach could lead to filtering out too many unseen class samples. This selection imbalance in sequential data learning can give rise to the Matthew Effect.

We adopt an adaptive threshold for each class to address this problem. Recognizing that the imbalance in selection arises from variations in softmax prediction distributions across classes, we leverage class statistics to establish class-independent thresholds. Specifically, we incorporate a curriculum learning strategy [Bengio *et al.*, 2009] to consider the learning progress of each class. Given the limited number of samples available at each time step, we aggregate statistics across all historical time steps to compute class confidence statistics. These statistics are momentum updated as follows:

$$\begin{aligned} \delta_{ema}(y) &= m_2 \cdot \delta_{ema}(y) + (1 - m_2) \cdot \delta_{t-1}(y), \\ \delta_{t-1}(y) &= \frac{1}{N_t^y} \sum_{\mathbf{x} \in \mathcal{D}_t^{te}} \mathbb{1}(y = \hat{y}_{\mathbf{x}}) p_{t-1}(\mathbf{x}, \hat{y}_{\mathbf{x}}; \mathcal{Y}), \end{aligned} \tag{9}$$

where $m_2 \in [0, 1)$ denotes a momentum coefficient, and $N_t^y = \sum_{\mathbf{x} \in \mathcal{D}_t^{te}} \mathbb{1}(y = \hat{y}_{\mathbf{x}})$ ($\hat{y}_{\mathbf{x}}$ is defined in Eq. (1)). $\delta_{t-1}(y)$ represents the averaged softmax prediction for class y, predicted by the immediately preceding stage model. δ_{ema} serves as the surrogate of the class learning status and is updated before data selection at time t. It is then utilized to adjust the fixed threshold τ. The scaled data mask is

$$\mathtt{M}_{scl}(\mathbf{x}) = \mathbb{1}((\mathbf{x}, f_{t-1}(\mathbf{x})) > \delta_{t-1}(f_{t-1}(\mathbf{x})) \cdot \tau). \tag{10}$$

$\mathtt{M}_{scl}(\cdot)$ is subsequently employed as a weighting factor for the loss of each datum to facilitate data selection, *i.e.*,

$$\mathcal{L}_{ce}^{sel} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_t^{te}} \mathtt{M}_{scl}(\mathbf{x}) \cdot \ell_{ce}^{sel}(\mathbf{x}), \mathcal{L}_{kl} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_t^{te}} \mathtt{M}_{scl}(\mathbf{x}) \cdot \ell_{kl}(\mathbf{x}). \tag{11}$$

Notably, the class and data selection processes only incur negligible extra computation.

4.5 Overall Objectives

Overall, the total objective loss function at each time t is

$$\mathcal{L}_{all} = \mathcal{L}_{ce}^{sel} + \lambda \mathcal{L}_{kl}, \tag{12}$$

Algorithm 1 The Proposed EGZSL Method

Input: Subset of data $\{\mathbf{x}_i\}_{i=1}^{N_t}$; Model f_{t-1} with parameters $\mathbf{W}_{t-1}$; Momentum model f_{ema} with parameters $\mathbf{W}_{ema}$; Class momentum confidence δ_{ema}; Hyper-parameters m_1, m_2, τ, λ.

1: Predict pseudo-label: $\hat{y}_{\mathbf{x}} = f_{t-1}(\mathbf{x})$.
2: Select training classes by Eq. (6).
3: Update class momentum confidence δ_{ema} by Eq. (9).
4: Calculate data mask by Eq. (10).
5: Update $\mathbf{W}_{t-1}$ by cross-entropy loss and KL divergence loss in Eq. (12).
6: Update weights $\mathbf{W}_{ema}$ of momentum model by Eq. (3).

Output: Prediction $\{f_{t-1}(\mathbf{x}) | \mathbf{x} \in \mathcal{D}_t^{te}\}$; Updated model f_t; Updated momentum model f_{ema}; Updated class momentum confidence δ_{ema}.

where λ is a hyper-parameter for balancing loss $\mathcal{L}_{ce}^{sel}$ and $\mathcal{L}_{kl}$. Algorithm 1 describes the concrete training process in one evolutionary learning step.

5 Experiments

In this section, we propose a protocol for evaluating EGZSL methods and compare the performance of our method to potential upper and lower boundaries. We also report on further experiments that shed light on the working mechanisms of our method by isolating the effects of individual components.

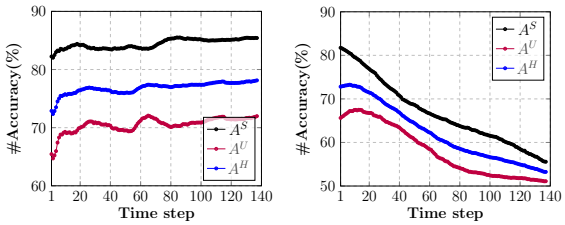
5.1 Benchmark Protocol

Evaluation Procedure. As there is no established benchmark protocol for assessing EGZSL performance, we propose the following evaluation procedure: for a given ZSL dataset, the original training set serves as the base set, while the test set is partitioned into various batches in a fixed random order. Each method is initially trained on the base set, followed by prediction and online updating on the divided test data stream. Predictions for the current time step are made by the model from the previous step. We present two variations of the test data splitting: dividing the test set into batches of 10 or 100 samples. To compare with traditional GZSL methods, we employ the same metrics [Xian *et al.*, 2017] for the EGZSL task, computed as the harmonic mean (H) of the average per-class top-1 accuracies in the seen (A^s) and unseen (A^u) classes. EGZSL performance is evaluated by aggregating predictions across all test batches. Each benchmark is repeated five times with different random data orders, and averages along with standard deviations of the results are reported.

Datasets. We evaluate EGZSL methods on three public ZSL benchmarks: 1) *Animals with Attributes 2 (AWA2)* [Lampert *et al.*, 2013] contains 50 animal species and 85 attribute annotations, accounting for 37,322 samples. 2) *Attribute Pascal and Yahoo (APY)* [Farhadi *et al.*, 2009] includes 32 classes of 15,339 samples and 64 attributes. 3) *Caltech-UCSD Birds-200-2011 (CUB)* [Wah *et al.*, 2011] consists of 11,788 samples of 200 bird species, annotated by 312 attributes. We split the data into seen and unseen classes according to the common GZSL benchmark procedure in [Xian *et al.*, 2017]. We

	Method	AWA2			CUB			APY		
		A^u	A^s	H	A^u	A^s	H	A^u	A^s	H
$\mathcal{T}$	COND [Li *et al.*, 2019]	80.2	90.0	84.8	57.0	68.7	62.3	51.8	87.6	65.1
	TF-VAEGAN [Narayan *et al.*, 2020]	87.3	89.6	88.4	69.9	72.1	71.0	-	-	-
	STHS [Bo *et al.*, 2021]	94.9	92.3	93.6	77.4	74.5	75.9	-	-	-
$\mathcal{I}$	COND [Li *et al.*, 2019]	56.4	81.4	66.7	47.4	47.6	47.5	26.5	74.0	39.0
	FREE [Chen *et al.*, 2021]	60.4	75.4	67.1	55.7	59.9	57.7	-	-	-
	Chou et al. [Chou *et al.*, 2021]	65.1	78.9	71.3	41.4	49.7	45.2	35.1	65.5	45.7
	GCM-CF [Yue *et al.*, 2021]	60.4	75.1	67.0	61.0	59.7	60.3	37.1	56.8	44.9
	ZLA [Chen *et al.*, 2022]	65.4	82.2	72.8	50.9	58.4	54.4	38.4	60.3	46.9
$\mathcal{E}$	COND+ERM@10	$51.9_{\pm 0.3}$	$75.5_{\pm 0.2}$	$61.5_{\pm 0.3}$	$41.1_{\pm 0.4}$	$45.4_{\pm 0.5}$	$43.1_{\pm 0.3}$	$26.3_{\pm 0.4}$	$45.8_{\pm 0.8}$	$33.4_{\pm 0.5}$
	COND+ERM@100	$51.2_{\pm 0.3}$	$74.7_{\pm 0.4}$	$60.1_{\pm 0.1}$	$36.3_{\pm 0.3}$	$39.6_{\pm 0.7}$	$37.9_{\pm 0.4}$	$25.4_{\pm 1.1}$	$43.0_{\pm 0.6}$	$32.0_{\pm 0.8}$
	COND+**ours@10**	$57.1_{\pm 0.5}$	$\mathbf{82.1}_{\pm 0.1}$	$67.4_{\pm 0.3}$	$\mathbf{45.2}_{\pm 0.1}$	$54.6_{\pm 0.1}$	$49.4_{\pm 0.1}$	$32.1_{\pm 0.7}$	$\mathbf{60.1}_{\pm 0.2}$	$41.9_{\pm 0.6}$
	COND+**ours@100**	$\mathbf{59.2}_{\pm 1.1}$	$80.7_{\pm 0.4}$	$\mathbf{68.3}_{\pm 0.8}$	$45.0_{\pm 0.2}$	$\mathbf{55.2}_{\pm 0.3}$	$\mathbf{49.6}_{\pm 0.1}$	$\mathbf{35.2}_{\pm 0.2}$	$58.0_{\pm 0.6}$	$\mathbf{43.8}_{\pm 0.2}$
	ZLA+ERM@10	$54.2_{\pm 6.5}$	$60.4_{\pm 3.1}$	$56.9_{\pm 4.2}$	$45.2_{\pm 4.3}$	$42.7_{\pm 4.0}$	$43.9_{\pm 4.0}$	$8.6_{\pm 0.2}$	$0.4_{\pm 0.3}$	$0.9_{\pm 0.4}$
	ZLA+ERM@100	$55.0_{\pm 3.1}$	$64.6_{\pm 2.5}$	$59.4_{\pm 2.7}$	$\mathbf{52.0}_{\pm 0.7}$	$51.2_{\pm 0.7}$	$51.6_{\pm 0.6}$	$11.5_{\pm 1.4}$	$5.1_{\pm 0.9}$	$6.8_{\pm 0.5}$
	ZLA+**ours@10**	$65.4_{\pm 0.6}$	$\mathbf{85.8}_{\pm 0.5}$	$74.2_{\pm 0.2}$	$51.0_{\pm 0.3}$	$\mathbf{58.9}_{\pm 0.3}$	$\mathbf{54.6}_{\pm 0.1}$	$39.1_{\pm 1.1}$	$\mathbf{60.1}_{\pm 1.1}$	$47.3_{\pm 0.8}$
	ZLA+**ours@100**	$\mathbf{73.3}_{\pm 1.0}$	$81.3_{\pm 0.8}$	$\mathbf{77.0}_{\pm 0.2}$	$51.7_{\pm 0.6}$	$57.9_{\pm 0.3}$	$54.6_{\pm 0.3}$	$\mathbf{40.0}_{\pm 1.0}$	$58.6_{\pm 0.7}$	$\mathbf{47.5}_{\pm 0.8}$

Table 1: Performance comparison between the proposed baselines and sota IGZSL and TGZSL methods. $\mathcal{T}$, $\mathcal{I}$, and $\mathcal{E}$ denote methods in the TGZSL, IGZSL, and EGZSL settings, respectively. @10 and @100 indicate the amount of data accessed in a single evolutionary time step. A^u and A^s represent per-class accuracy scores (%) in seen and unseen test sets, and H is their harmonic mean. The best results are bolded.



(a) Ours (b) ERM

Figure 3: **(a), (b)** A comparison of evolution curves between our approach and ERM (on AWA2, with 100 samples per time step). Our method displays a rise in accuracy over time, while ERM experiences a decline in accuracy.

follow [Xian *et al.*, 2017] to adapt the 2048-dimensional visual representation (instead of the original images) extracted from the pre-trained ResNet101 [He *et al.*, 2016].

5.2 Implementation Details

Model. As the base learning phase setup remains unchanged from IGZSL, we simply borrow the off-the-shelf IGZSL model for conducting EGZSL experiments. Our EGZSL approach is developed using linear classifiers (Eq. (5)) that were trained by COND [Li *et al.*, 2019] and ZLA [Chen *et al.*, 2022]. The base models are acquired with their official codes. Please refer to the original papers for details on the base phase training procedure.

Optimization. We employ the Adam optimizer [Kingma and Ba, 2015] with a learning rate of $5e$-5 for the main experiments. We set the (mini) batch size equal to the total number of data in each evolutionary stage. Each stage of data is optimized for one epoch only.

5.3 Main Results

Baselines. We establish our proposed method on COND [Li *et al.*, 2019] and ZLA [Chen *et al.*, 2022], as shown in Tab. 1. For comparison, we also report the results of a basic empirical risk minimization (ERM) approach with pseudo-labels. Additionally, we evaluate the performance of our approach against the IGZSL and TGZSL methods, which establish the potential high and low limits of the EGZSL performance.

Results. Tab. 1 presents our main experimental results, according to which we have the following findings:

Our approach improves upon the IGZSL baseline, whereas the straightforward ERM approach fails. For unsupervised data stream learning, relying on the basic method without addressing issues such as catastrophic forgetting, prediction bias, and data bias, results in unreliable gradients, especially when the initial model exhibits significant bias (as in the case of APY). In contrast, the superior results of our method demonstrate that it effectively handles these challenges.

The process of evolutionary learning provides greater benefits to coarse-grained datasets, such as AWA2 and APY, compared to fine-grained datasets like CUB. This is attributed to the prevalence of visual bias [Fu *et al.*, 2014] resulting from missing unseen classes during base training, which is more pronounced in coarse-grained datasets. Consequently, mitigating this bias leads to a more substantial performance improvement. Additionally, the limited amount of evolutionary data per class (23 in CUB *v.s.*275 in AWA2) also contribute to the modest improvement observed in CUB.

The performance improvement is slightly larger when more data is provided per time step. Accessing more data at once lessens the probability of being overwhelmed by erroneous pseudo-label samples and presents a consistent gradient. In a real-world deployment, the batch size of a single evolutionary step can be selected according to specific requirements and resource usage.

Even though our approach outperforms the IGZSL baseline, there remains a significant gap between its results and those obtained by methods in the TGZSL setting. This is primarily owing to three reasons: **first**, TGZSL allows for repeatedly training on all of the test data; **second**, TGZSL offers a more relaxed constraint by providing prior knowl-

	Baseline	AWA2			APY		
		A^u	A^s	H	A^u	A^s	H
(i)	W/O Momentum Model	57.2	80.2	66.7	17.8	45.5	25.6
(ii)	W/O Class Selection	**66.4**	72.9	69.5	**39.1**	51.3	44.4
(iii)	W/O Data Selection	62.3	86.7	72.5	38.7	57.3	46.2
(iv)	Adaptive→fixed thre.	57.4	**86.2**	68.9	27.6	50.4	35.7
	Full model	65.9	85.5	**74.4**	38.6	**61.5**	**47.4**

Table 2: Ablation study results of the proposed method on AWA2 and APY datasets (with 10 samples per time step).

edge regarding whether the test data belong to seen or unseen classes; and **third**, greater attention has been paid to the TGZSL area, while there is still potential for further advancement in EGZSL. Regardless, the EGZSL setting is better suited for practical applications and holds greater potential for real-world deployment.

5.4 Evolution Curves

To evaluate model performance across different time steps, we plot its evolution curve as the evolutionary task progresses. Given that the test data differs among time steps, we compile the evolution curve using all test data. This is legitimate within the TGZSL setting and only applies to explanatory experiments. As shown in Fig. 3, our method demonstrates a consistent improvement in performance over time, whereas basic ERM leads to continuous forgetting of initial knowledge. This experiment is conducted based on ZLA.

5.5 Ablation Analysis

We validated the effectiveness of our motivation and design through the following baselines, and the results are presented in Tab. 2. These results are obtained by utilizing ZLA as the base model and maintaining a consistent random seed throughout the testing process.

(i) W/O Momentum Model. We first investigate the effect of the momentum model, which aids in preserving historical information. As shown in Tab. 2, omitting this component leads to performance degradation on both datasets. The decline in performance is particularly pronounced when the initial accuracy is low (as observed in APY). In the absence of historical information, noisy pseudo-labels dominate the training process. This demonstrates the importance of suppressing catastrophic forgetting in training with streaming data.

(ii) W/O Class Selection. We also evaluate the importance of the class selection module. As previously discussed, this module helps mitigate the adverse effects of potentially imbalanced data classes. There is a noticeable decrease in performance when ablating it.

(iii) W/O Data Selection. This baseline removes the operations defined by Eq. (9), (10), and (11). The removal of data selection implies that all samples are involved in the training. The decline in performance aligns with our expectation that filtering out low-confidence samples is beneficial.

(iv) Adaptive→fixed thre. To demonstrate the effectiveness of the adaptive threshold strategy in data selection, we conduct an experiment with a fixed threshold. This baseline is also described in Sec. 4.4 that replaces Eq. (10) with Eq. (9). We set τ to 0.8 for the best results of this baseline, but the performance is even worse than without data selection. This validates our analysis that a fixed threshold comes with the risk of data class imbalance.

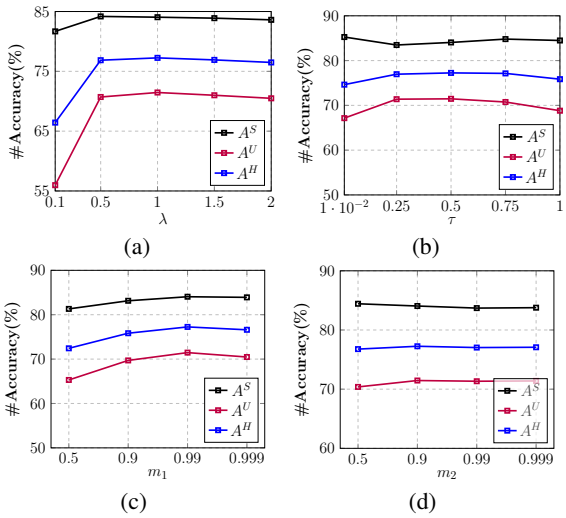


Figure 4: Hyperparameters *w.r.t.* EGZSL performance on AWA2. **(a)** Effects of loss balancing coefficient λ (Eq. 12). **(b)** Effects of thresholds τ in Eq. (10). **(c), (d)** Effects of the smoothing factors m_1 and m_2 in Eq. (3) and (9).

5.6 Hyperparameters

We study the influence of the loss weighting coefficient λ, the threshold τ, and the momentum coefficients m_1 and m_2, which are reported in Fig. 4. Although performance under different hyperparameter settings varies, our method is overall stable. The results are more sensitive to λ and m_1 as these two parameters are related to catastrophic forgetting. In contrast, τ and m_2, two variables related to data selection, have a slightly smaller fluctuation in performance. We set λ at 1, τ at 0.5, m_1 at 0.99, and m_2 at 0.9 for the best results. More experiments can be found in the supplemental.

6 Conclusion

In this paper, we introduce a novel and more realistic GZSL setting: Evolutionary GZSL. This setting aims to address the domain shift problem inherent in IGZSL, while maintaining greater deployability than TGZSL. EGZSL starts from the traditional learned GZSL models and gradually boosts itself by simultaneously recognizing and learning from the unlabeled test data. To evaluate the proposed EGZSL, we devise a new protocol involving random division of datasets into episodic training and testing with multiple time steps. Furthermore, we propose a method to tackle this task and present baseline results on three benchmark datasets. The results demonstrate the feasibility and superiority of our approach compared to several traditional methods.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62371235 and 62072246, and in part by the Key Research and Development Plan of Jiangsu Province (Industry Foresight and Key Core Technology Project) under Grant BE2023008-2.

References

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

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

[Atzmon and Chechik, 2019] Yuval Atzmon and Gal Chechik. Adaptive confidence smoothing for generalized zero-shot learning. In *CVPR*, pages 11671–11680, 2019.

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

[Bo *et al.*, 2021] Liu Bo, Qiulei Dong, and Zhanyi Hu. Hardness sampling for self-training based transductive zero-shot learning. In *CVPR*, pages 16499–16508, 2021.

[Burns *et al.*, 2023] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

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

[Chaudhry *et al.*, 2019] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[Chen *et al.*, 2020] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[Chen *et al.*, 2021] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. In *ICCV*, 2021.

[Chen *et al.*, 2022] Dubing Chen, Yuming Shen, Haofeng Zhang, and Philip H.S. Torr. Zero-shot logit adjustment. In *IJCAI*, pages 813–819, 2022.

[Chen *et al.*, 2023] Dubing Chen, Yuming Shen, Haofeng Zhang, and Philip HS Torr. Deconstructed generation-based zero-shot model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 295–303, 2023.

[Chou *et al.*, 2021] Yu-Ying Chou, Hsuan-Tien Lin, and Tyng-Luh Liu. Adaptive and generative zero-shot learning. In *ICLR*, 2021.

[Dinh *et al.*, 2014] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014.

[Elhoseiny *et al.*, 2013] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *ICCV*, pages 2584–2591, 2013.

[Fang *et al.*, 2022] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? In *NeurIPS*, 2022.

[Farhadi *et al.*, 2009] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, pages 1778–1785, 2009.

[Ferrara, 2023] Emilio Ferrara. Should chatgpt be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*, 2023.

[French, 1999] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[Frome *et al.*, 2013] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, page 2121–2129, 2013.

[Fu *et al.*, 2014] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, pages 584–599. Springer, 2014.

[Gautam *et al.*, 2020] Chandan Gautam, Sethupathy Parameswaran, Ashish Mishra, and Suresh Sundaram. Generalized continual zero-shot learning. *arXiv preprint arXiv:2011.08508*, 2020.

[Ghosh, 2021] Subhankar Ghosh. Dynamic vaes with generative replay for continual zero-shot learning. *arXiv preprint arXiv:2104.12468*, 2021.

[Grandvalet and Bengio, 2004] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, volume 17, 2004.

[Han *et al.*, 2021] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. In *CVPR*, pages 2371–2381, 2021.

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

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS*, 2015.

[Jiang *et al.*, 2024] Chenyi Jiang, Yuming Shen, Dubing Chen, Haofeng Zhang, Ling Shao, and Philip HS Torr. Estimation of near-instance-level attribute bottleneck for zero-shot learning. *International Journal of Computer Vision*, pages 1–27, 2024.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2013.

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

[Lampert *et al.*, 2009] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.

[Lampert *et al.*, 2013] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, pages 453–465, 2013.

[Lee and others, 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML work shop*, page 896, 2013.

[Li *et al.*, 2019] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*, pages 3583–3592, 2019.

[Liu *et al.*, 2021] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, volume 34, pages 21808–21820, 2021.

[McCloskey and Cohen, 1989] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[Mukhoti *et al.*, 2021] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty: A simple baseline. *arXiv e-prints*, pages arXiv–2102, 2021.

[Narayan *et al.*, 2020] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, pages 479–495, 2020.

[Paul *et al.*, 2019] Akanksha Paul, Narayanan C Krishnan, and Prateek Munjal. Semantically aligned bias reducing zero shot learning. In *CVPR*, pages 7056–7065, 2019.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017.

[Shen *et al.*, 2020] Yuming Shen, Jie Qin, Lei Huang, Li Liu, Fan Zhu, and Ling Shao. Invertible zero-shot recognition flows. In *ECCV*, pages 614–631, 2020.

[Skorokhodov and Elhoseiny, 2021] Ivan Skorokhodov and Mohamed Elhoseiny. Class normalization for (continual)? generalized zero-shot learning. In *ICLR*, 2021.

[Sun *et al.*, 2020] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pages 9229–9248. PMLR, 2020.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, california institute of technology, 2011.

[Wan *et al.*, 2019] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. In *NeurIPS*, volume 32, 2019.

[Wang *et al.*, 2021] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[Wang *et al.*, 2022] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7201–7211, 2022.

[Wei *et al.*, 2020] Kun Wei, Cheng Deng, Xu Yang, et al. Lifelong zero-shot learning. In *IJCAI*, pages 551–557, 2020.

[Xian *et al.*, 2017] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *CVPR*, pages 4582–4591, 2017.

[Xian *et al.*, 2018] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, pages 5542–5551, 2018.

[Xian *et al.*, 2019] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-gan-d2: A feature generating framework for any-shot learning. In *CVPR*, pages 10275–10284, 2019.

[Xie *et al.*, 2020] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, volume 33, pages 6256–6268, 2020.

[Xu *et al.*, 2020] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. In *NeurIPS*, pages 21969–21980, 2020.

[Yi and Elhoseiny, 2021] Kai Yi and Mohamed Elhoseiny. Domain-aware continual zero-shot learning. *arXiv preprint arXiv:2112.12989*, 2021.

[Yue *et al.*, 2021] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, pages 15404–15414, 2021.