Neural Networks 175 (2024) 106278

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Generalizability and robustness evaluation of attribute-based zero-shot learning

Luca Rossi [a,*], Maria Chiara Fiorentino [a], Adriano Mancini [a], Marina Paolanti [b], Riccardo Rosati [a], Primo Zingaretti [a]

[a] *Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131, Ancona, Italy*

[b] *Dipartimento di Scienze politiche, della Comunicazione e delle Relazioni Internazionali, Università di Macerata, 62100, Macerata, Italy*

ARTICLE INFO

Dataset link: https://drive.google.com/drive/folders/16Xk1eFSWjQTtuQivTogMmvL3P6F_084u, https://github.com/luca-rossi/grabzsl

Keywords:
Zero-shot learning
Evaluation metrics
Generalizability
Robustness

ABSTRACT

In the field of deep learning, large quantities of data are typically required to effectively train models. This challenge has given rise to techniques like zero-shot learning (ZSL), which trains models on a set of "seen" classes and evaluates them on a set of "unseen" classes. Although ZSL has shown considerable potential, particularly with the employment of generative methods, its generalizability to real-world scenarios remains uncertain.

The hypothesis of this work is that the performance of ZSL models is systematically influenced by the chosen "splits"; in particular, the statistical properties of the classes and attributes used in training. In this paper, we test this hypothesis by introducing the concepts of generalizability and robustness in attribute-based ZSL and carry out a variety of experiments to stress-test ZSL models against different splits. Our aim is to lay the groundwork for future research on ZSL models' generalizability, robustness, and practical applications.

We evaluate the accuracy of state-of-the-art models on benchmark datasets and identify consistent trends in generalizability and robustness. We analyze how these properties vary based on the dataset type, differentiating between coarse- and fine-grained datasets, and our findings indicate significant room for improvement in both generalizability and robustness. Furthermore, our results demonstrate the effectiveness of dimensionality reduction techniques in improving the performance of state-of-the-art models in fine-grained datasets.

1. Introduction

Deep learning (DL) has become a widely used tool in image recognition and computer vision tasks thanks to its ability to extract patterns from data and generate effective decision-making rules. However, the necessity for large datasets with corresponding manual annotations has hampered its adoption in situations where data acquisition is laborious.

To address this challenge, several techniques have been developed to leverage knowledge from readily accessible data. These include semi-supervised learning (Reddy, Viswanath, & Reddy, 2018), transfer learning (Tan et al., 2018), self-taught learning (Wang, Nie, & Huang, 2013), and zero-shot learning (ZSL) (Wang, Yao, Kwok, & Ni, 2020). ZSL is specifically devised to train a model to classify objects from *unseen classes* (the target domain) by transferring knowledge from *seen classes* (the source domain) (Changpinyo, Chao, Gong, & Sha, 2020; Wang, Zheng, Yu, & Miao, 2019), using semantic connections in a defined space (Pourpanah et al., 2023).

Under a conventional ZSL setting (CZSL), the test set solely contains samples from unseen classes. This is an unrealistic scenario since seen classes are frequently present in the model's deployment environment. To rectify this, the generalized zero-shot learning (GZSL) setting has been introduced, where models are evaluated on both seen and unseen classes (Rahman, Khan, & Porikli, 2018; Wang & Breckon, 2023; Ye, Hu, & Zhan, 2021). While GZSL methods show promising results (Liu & Ozay, 2023), we are far from optimal performance due to unresolved challenges such as domain shift, seen class bias, cross-domain transfer, hubness, and semantic loss (Pourpanah et al., 2023). In recent years, generative methods have been employed to mitigate some of these issues, especially domain shift and bias. Such methods (Xian, Lorenz, Schiele, & Akata, 2018) generally achieve better accuracy by generating images or visual features for unseen classes and, for this reason, they have replaced previous embedding methods as the preferred approach to ZSL (Sun, Gu, & Sun, 2021).

* Corresponding author.
E-mail addresses: l.rossi@pm.univpm.it (L. Rossi), m.c.fiorentino@pm.univpm.it (M.C. Fiorentino), a.mancini@univpm.it (A. Mancini), marina.paolanti@unimc.it (M. Paolanti), r.rosati@pm.univpm.it (R. Rosati), p.zingaretti@univpm.it (P. Zingaretti).

https://doi.org/10.1016/j.neunet.2024.106278
Received 6 September 2023; Received in revised form 15 February 2024; Accepted 26 March 2024
Available online 28 March 2024

However, there is an overlooked challenge for ZSL methods that has not yet garnered sufficient attention from the research community. In order to allow for a direct comparison among different methods, Xian, Lampert, Schiele, and Akata (2018) propose a set of standard conditions (i.e., the classes used for training and the structure of the semantic space) for the evaluation of GZSL models, that, from now on, we will refer to as the *benchmark split*.

Most studies focus on this split (see Section 2), but neglect the generalization to different splits, which is crucial for real-world applications. This raises concerns about whether a higher performance of a novel GZSL model can be interpreted as an improvement or if the model is just overfitting the benchmark split.

This research is thus motivated by the need to quantify how ZSL performance generalizes across different splits and assess the real-world applicability of the evaluated models. Any semantically-defined split introduces a loss of information (semantic loss), and the hypothesis behind this work is that this semantic loss varies in a non-negligible way with each split, as some splits have lower entropy than others. The goal is to quantify the extent to which this effect varies as a function of the split.

This paper has two key contributions. First, we propose a theoretical and practical framework to define the concepts of generalizability and robustness of ZSL models, with a particular focus on the concept of split. Second, we define novel metrics to perform this kind of evaluation; in particular, we propose four splitting methods among classes and attributes, with the goal of stress-testing the models.

We conduct a series of experiments to evaluate these properties using both coarse- and fine-grained benchmark datasets, demonstrating a significant margin for improvement in generalizability and robustness. Our results demonstrate how these splitting methods are responsible for wide changes in the performance of the models; often negative, but sometimes positive, as we show that dimensionality reduction can be effective in improving ZSL performance in fine-grained datasets.

In this work, we focus on generative methods for attribute-based inductive ZSL (from now on, ABZSL or simply ZSL). Generative models are used to synthesize unseen classes, attribute-based means that a list of engineered (rather than learned) attributes defines the classes, and inductive means that only the seen classes are used during training. We will consider both CZSL and GZSL in our experiments.

The rest of this paper is organized as follows. In Section 2 we discuss the ZSL taxonomy and related works. In Section 3 we describe our methodology, introducing the concepts of generalizability and robustness, and proposing the splitting methods used for robustness evaluation. In Section 4 we present and discuss the results of our experiments on the tested models with the proposed splits on benchmark datasets. Finally, in Section 5 we draw conclusions on the generalizability and robustness of ZSL models, and suggest some directions for future research.

2. Related works

During the last decade, several methods have been developed to tackle the problem of image classification in ZSL settings (Akata, Harchaoui, & Schmid, 2015; Frome et al., 2013; Fu, Hospedales, Xiang, & Gong, 2015; Jayaraman & Grauman, 2014; Rohrbach, Ebert, & Schiele, 2013; Romera-Paredes & Torr, 2015; Xian, Lorenz, Schiele, & Akata, 2018; Ye & Guo, 2017). These methods are typically classified according to four criteria: (1) the training methodology (e.g. embedding-based, generative), (2) the type of semantic space (e.g. attribute-based, corpora-based), (3) the data available in the training set (e.g. inductive, transductive), and (4) the data available in the test set (e.g. conventional ZSL, generalized ZSL). Our work focuses on generative methods for attribute-based inductive CZSL and GZSL:

- *Generative methods* are models, such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), which synthesize images of unseen classes conditioned on their semantic description, so that a standard classifier can be trained on this synthetic dataset;
- *Attribute-based* means that the semantic space is characterized by a list of attributes designed to describe each class. This engineered semantic space is easier to create than a learned representation (e.g. from text corpora), but it carries the biases of the human annotators, thereby potentially compromising generalizability;
- *Inductive* means that there are no images from unseen classes in the training phase. This is a more challenging setting than transductive ZSL, which includes unlabeled images from unseen classes in the training phase;
- *Conventional ZSL (CZSL)* indicates that only the unseen classes are considered in the test phase. *Generalized ZSL (GZSL)* indicates that the test set consists of images from both seen and unseen classes. GZSL is more appropriate for real-world scenarios, but models tend to be biased towards recognizing seen classes.

Traditional "embedding-based" ZSL methods (Han, Fu, Chen, & Yang, 2021; Liu et al., 2023; Pourpanah et al., 2023; Van Gansbeke, Vandenhende, Georgoulis, Proesmans, & Van Gool, 2020; Xu, Xian, Wang, Schiele, & Akata, 2022; Yun, Wang, Hou, & Gao, 2022) learn to project visual and semantic features from seen classes into a common space called embedding space. This learned space is then used to recognize novel classes. To overcome the aforementioned ZSL challenges, especially domain shift and bias, most of the recent methods employ generative models instead.

A range of generative methods adopt GANs to synthesize unseen class features, which are then used in a fully supervised setting to train a standard classifier. Xian, Lorenz, Schiele, and Akata (2018) propose f-CLSWGAN, a model consisting of a conditional Wasserstein GAN (WGAN) (Arjovsky, Chintala, & Bottou, 2017) paired with a classification loss, able to generate discriminative features for unseen classes. Felix, Reid, and Carneiro (2018) replace the seen category classifier with a decoder using a cycle-consistency loss (Zhu, Park, Isola, & Efros, 2017). Schönfeld, Ebrahimi, Sinha, Darrell, and Akata (2019) introduce cross and distribution alignment losses for aligning the visual features and corresponding embeddings in a shared latent space using two Variational Autoencoders (VAEs) (Kingma & Welling, 2014), proposing the CADA-VAE approach.

Xian, Sharma, Schiele, and Akata (2019) introduce an F-VAEGAN framework that combines a VAE decoder and a GAN generator for feature synthesis with a cycle-consistency loss between generated and original visual features. Other GAN-based ZSL classification methods (Felix et al., 2018; Huang, Wang, Yu, & Wang, 2019; Mandal et al., 2019; Zhang & Peng, 2018) use auxiliary modules to enforce cycle-consistency on the embeddings during training. Yu and Lee (2019) use generated unseen classes as training data points to update model parameters step by step. Narayan, Gupta, Khan, Snoek, and Shao (2020) propose TF-VAEGAN by extending f-VAEGAN with a feedback loop that iteratively improves the quality of synthesized features. To solve visual-semantic domain gap and seen–unseen bias, a method named FREE is proposed (Chen et al., 2021). FREE exploits a feature refinement module consisting of a semantic/visual mapping coupled with a generative model with the aim of refining visual features of both seen and unseen classes. In the study conducted by Zhao, Shen, Wang, and Zhang (2023), a generator is trained to augment category semantics and generate visual features. This process enhances the alignment between the generated visual features and the distribution of real features, resulting in improved performance. To address the challenge of redundancy in synthetic features, Gowda (2023) introduces SPOT, a novel reinforcement learning-based approach. This method employs a transformer-based selector trained through proximal policy optimization to enhance the selection of synthetic features, thereby improving classification accuracy. Yang, Lee, Lin, and Wang (2023) introduce the Cross-Model Consistency GAN (CMC-GAN), which is a generative model that enables data hallucination for unseen classes through semantics-guided intra-category knowledge transfer across image categories. By incorporating appropriate semantics and ensuring ample visual diversity, CMC-GAN facilitates the generation of data that resembles unseen classes. The seen and unseen bias problem is tackled also by Yue, Wang, Sun, Hua, and Zhang (2021), who propose a generative causal model to produce faithful counterfactuals, which allows using a consistency rule for balanced seen/unseen classification. Kong et al. (2022), propose a method to enhance intra-class compactness while maintaining inter-class separability of both seen and unseen classes in the visual feature and embedding spaces. Su, Li, Chen, Zhu, and Lu (2022) propose to generate fictitious classes to separate seen and unseen samples for GZSL, leveraging both visual and semantic modalities to distinguish seen and unseen categories. The framework proposed by Han, Fu, Chen, and Yang (2022) combines an embedding model with a feature generation model, introducing a semantic contrastive embedding that consists of both attribute-level and class-level embeddings.

Xian, Lampert, Schiele, and Akata (2018) propose a *benchmark split* that defines the seen/unseen class split to be used for evaluation, as well as the attributes that describe the semantic space. The works described here focus predominantly on performing evaluations on this specific split, which is a straightforward way to compare different models. However, this raises concerns regarding the generalizability and robustness of such models, as the performance on the benchmark split may not necessarily translate to similar performance across different splits. This limitation highlights a crucial gap in the current research landscape, emphasizing the need for models that not only excel in standard benchmark settings but also demonstrate robust generalization across varied data distributions (Ge et al., 2023). To the best of our knowledge, no other works have tackled this problem at the time of this writing. Therefore, there is a compelling case for the necessity of our work, which is aimed at defining a novel evaluation framework and conducting a series of experiments to assess the generalizability and robustness of ZSL models over different splits.

3. Methodology

In this section, a full description of the proposed methodology is provided. First, we formalize the ABZSL problem in Section 3.1. Then we define the key concepts of our work, such as generalizability and robustness, in Section 3.2. Finally, in Section 3.3, we illustrate the splitting methods proposed in this work to evaluate the robustness.

3.1. Formalization of the attribute-based ZSL problem

Let $k \in \mathbb{N}$ be the number of attributes, $d \in \mathbb{N}$ the number of features (also referred to as images or samples), $n \in \mathbb{N}$ the number of classes, with n_S as the number of seen classes ($n_S < n$) and $n_U = n - n_S$ the number of unseen classes.

Let $A = \mathbb{R}^k$ be the semantic space (also referred to as the attribute space), $X = \mathbb{R}^d$ the feature space, and $Y = Y_1 \ldots Y_n$ the set of all classes (seen and unseen). Let $Y^S \subset Y$ be the subset of seen classes and $Y^U = Y \setminus Y^S$ the subset of unseen classes. Similarly, $X^S \subset X$ will refer to the samples belonging to seen classes and $X^U = X \setminus X^S$ to the samples belonging to unseen classes. The semantic space A provides extra information and acts as a bridge between seen and unseen classes.

We define the signature of a class $a : Y \rightarrow A$ as the function that maps each class to the attribute vector in the semantic space that uniquely identifies it. For convenience, we will use the notation a_y instead of $a(y)$. We define an attribute $a^i : Y \rightarrow \{0, 1\}$ as a function indicating whether such attribute is present (with value 1) or absent (with value 0) in that class or, alternatively, the ith element of the attribute vector returned by a. This is the definition of a binary attribute; our experiments use continuous attributes with values included in $[0, 1]$ indicating their frequency in a class, but we will treat them as binary attributes for convenience. With this notation, we refer to the ith element of the attribute vector of the class y as a^i_y.

We also define the set of labeled samples of seen classes as in Eq. (1) and the set of labeled samples of unseen classes as in Eq. (2).

$$S = \{(x, y, a_y) | x \in X^S, y \in Y^S, a_y \in A^S\} \tag{1}$$

$$U = \{(x, y, a_y) | x \in X^U, y \in Y^U, a_y \in A^U\} \tag{2}$$

In CZSL, we define $f : X \rightarrow Y^U$ as an unknown function that maps features to classes and $\tilde{f} : X \rightarrow Y^U$ as a classifier that approximates f. In GZSL, We define $f : X \rightarrow Y$ as an unknown function that maps features to classes and $\tilde{f} : X \rightarrow Y$ as a classifier that approximates f. In general, our goal is to train the classifier f to minimize the error $(f - \tilde{f})^2$.

The training set consists of all the pairs of features and labels only belonging to the seen classes, defined as $T = \{(x, f(x)) : x \in X^{*S}\}$ where $X^{*S} \subset X^S$ is the set of available samples. This is the inductive setting; in the transductive setting we also include unlabeled unseen samples in the training set, but for our purposes, we are only interested in the inductive setting.

In CZSL, models are commonly evaluated using the average per-class-top-1 accuracy as in Eq. (3), where N is the number of classes:

$$acc_Y = \frac{1}{N} \sum_{c=1}^{n} \frac{\#\text{ correct predictions in c}}{\#\text{ samples in c}} \tag{3}$$

This encourages high performance on both sparsely and densely populated classes. In GZSL, the accuracy is evaluated on both seen (acc_{Y^S}) and unseen (acc_{Y^U}) classes. Since the accuracy needs to be good on both, the harmonic mean is used as the preferred performance metric, defined in Eq. (4):

$$acc_H = \frac{2 * acc_{Y^S} * acc_{Y^U}}{acc_{Y^S} + acc_{Y^U}} \tag{4}$$

While embedding-based methods estimate $\tilde{f}$ by exploiting the mapping between A and Y, generative methods train a generator on T to create a synthetic dataset $\tilde{X}^U$, this dataset will be used to train a standard classifier on a complete training set $T = \{(x, f(x)) : x \in X^{*S} \cup \tilde{X}^{*U}\}$ where $\tilde{X}^{*U} \subset \tilde{X}^U$.

3.2. Proposed framework

Current works in ZSL attempt to build classifiers that improve the state-of-the-art accuracy on the benchmark split proposed by Xian, Lampert, Schiele, and Akata (2018). We, however, hypothesize that current attribute-based ZSL models lack complete generalizability and robustness across various splits, with implications not only for theoretical ZSL results but also for the applicability of these models in real-world scenarios.

The goal is to assess whether the performance of current ZSL models can be effectively transferred to real-world conditions, or if these models are merely overfitting to the benchmark split. Our proposed evaluation framework introduces an additional step in the evaluation pipeline to analyze the generalizability and robustness of ZSL models (Fig. 1). However, before diving into these topics, we will introduce some fundamental concepts such as splits, the upper bound, and the performance gap.

3.2.1. Splits and splitting methods

We evaluate ZSL models on different *splits*, selected according to the criteria and *splitting methods* outlined in Section 3.3. We distinguish between *class splits* and *attribute splits*.

Given a set of classes Y, we define a class split σ_Y as a partition of Y into two disjoint subsets Y^S and Y^U such that $Y^S \cup Y^U = Y$ and $Y^S \cap Y^U = \emptyset$. We define an attribute split σ_A simply as a set of attributes,

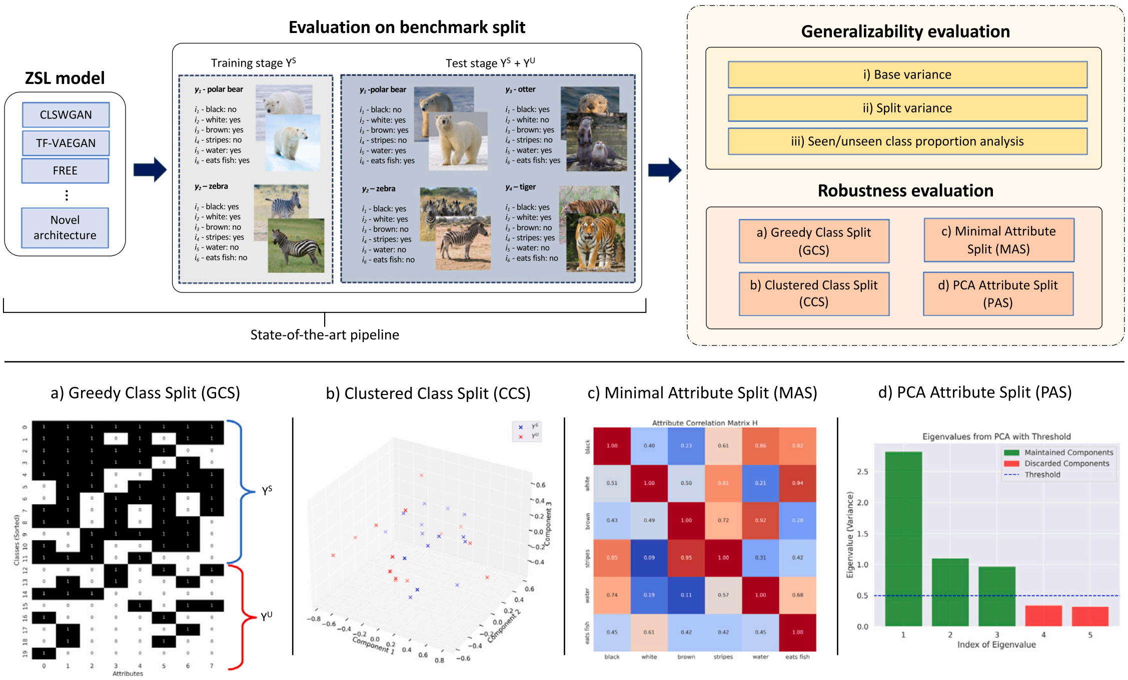

Fig. 1. The proposed framework to evaluate the generalizability and robustness of ZSL models, with a particular focus on the concept of the split. The yellow block symbolizes our novel contribution: while previous works only evaluate models on the benchmark split, we include an additional step of generalizability and robustness evaluation. Notably, this framework is independent of the specific ZSL model or dataset, and it allows for the integration of additional splits for evaluation. For each proposed split related to robustness evaluation, a visual representation is provided at the bottom of the figure.

defined as a mapping from Y to A. We define a split σ as a pair of a class split σ_Y and an attribute split σ_A.

Class splits refer to the seen/unseen class partitioning and indicate the classes chosen for training (seen classes) and those used for evaluation (unseen classes). On the other hand, attribute splits pertain to a specific configuration of the semantic space and indicate the set of semantic descriptors, or *attributes*, which together uniquely identify a class. A point in the semantic space is a vector that assigns a value to each of those attributes. We use the term *splits* to refer to both class splits and attribute splits, i.e. a split refers to a specific class partition (seen/unseen) coupled with a distinct set of attributes.

3.2.2. Upper bound and performance gap

The introduction of the following concepts, although not strictly required for understanding this work, allows shaping the problem in an intuitive, practical, and theoretical way. Our subsequent definitions are applicable only for models trained "under reasonable conditions" (u.r.c.), i.e. ignoring extreme scenarios that may invalidate our conclusions but lack practical relevance (e.g., identical images in the training and test set, a test set with a single image, etc.).

We characterize the Upper Bound (UB) as the highest possible accuracy that any ZSL model could theoretically achieve on a given dataset. This is determined by a classifier with a similar architecture trained on all classes, both seen and unseen. Our definition is qualitative as a precise definition of UB is not straightforward, and its quantification is unnecessary for our discussion (refer to the limitations subsection for more details).

The intuition is that ZSL models, due to limited data during training, cannot achieve an accuracy as high as a model trained with all classes' data u.r.c., in general.

We define the Performance Gap (PG) relative to a split and a ZSL model as the difference in accuracy between that model and its corresponding UB. Let M be the set of all ZSL models, D be the set of all datasets defined u.r.c., and Σ the set of all splits defined u.r.c. Let $d \in D$ be a dataset and $\sigma \in \Sigma$ a split. Let $m \in M$ be a ZSL model and $acc : M \times \Sigma \times D \to [0, 1]$ the accuracy of model m on dataset d for the split σ. Let $m^* \in M$ an *equivalent* non-ZSL architecture of m (we leave out the formal definition of equivalent architecture for brevity) and $\sigma^* \in \Sigma$ a *null split* (a split where the unseen classes set is empty). We define PG, averaged across different training runs of the models, as in Eq. (5):

$$\mathrm{PG}(m, \sigma, d) = \mathbb{E}\left[\mathrm{UB} - acc(m, \sigma, d)\right] \tag{5}$$

where $\mathrm{UB} = acc(m^*, \sigma^*, d)$. From this point forward, we will leave out the parameters m, σ, d for simplicity, unless necessary to understand a given context.

3.2.3. Qualitative interpretations of the performance gap

By decomposing PG, we can gain a better understanding of generalizability and robustness. We can decompose PG according to the source of the error, as in Eq. (6):

$$\mathrm{PG} = \mathrm{ML}(m) + \mathrm{SL}(m, \sigma, d) \tag{6}$$

ML denotes the *Model Loss*, which is the error that arises from the model's inherent inability to accurately fit the data distribution, and is the component that does not depend on the chosen split. SL is the *Semantic Loss*, or the error that originates from the semantic space's inability to adequately represent the unseen classes. Here, we use the term *loss* to refer to the accuracy metric, not the function to be minimized during training.

ML can be improved with methodological improvements. For example, generative ZSL models typically demonstrate lower ML than embedding-based models, as their performance tends to be superior when evaluated on the same splits. Most aforementioned ZSL works only compare and improve ML. Improvements in SL can be achieved by modifying the split or selecting a more descriptive set of attributes. While it is not always possible to select the split in practical applications, we can make these changes when evaluating methodological improvements on benchmark datasets to determine whether their performance is consistent under varying conditions or if they are merely overfitting to the benchmark split.

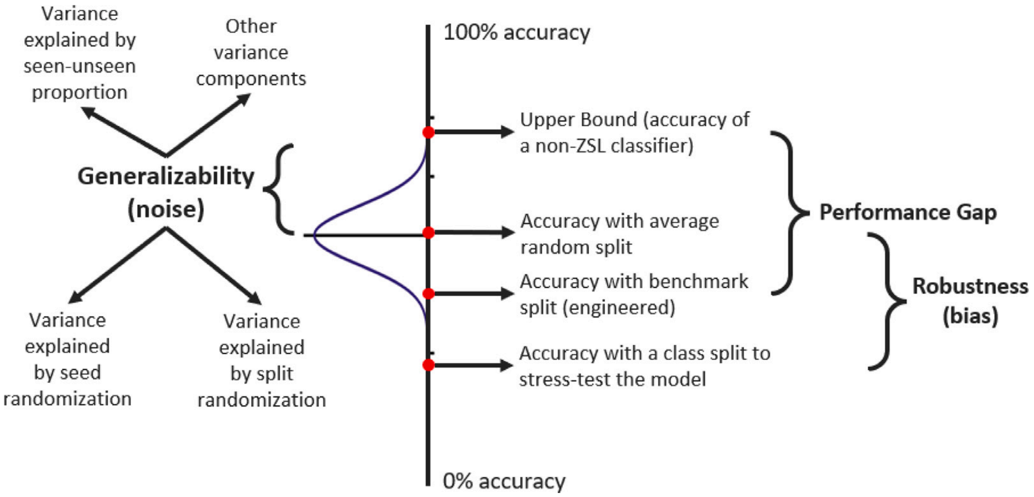
Variance explained by seen-unseen proportion
Other variance components
Generalizability (noise)
Variance explained by seed randomization
Variance explained by split randomization
100% accuracy
Upper Bound (accuracy of a non-ZSL classifier)
Accuracy with average random split
Accuracy with benchmark split (engineered)
Accuracy with a class split to stress-test the model
0% accuracy
Performance Gap
Robustness (bias)

Fig. 2. A qualitative, high-level representation of the Performance Gap.

We can further decompose SL into two additional components, as in Eq. (7):

$$\mathrm{SL} = \mathrm{AL} + \mathrm{DL} \tag{7}$$

AL, or *Attribute Loss*, is the loss of information resulting from an incomplete definition of the attributes. Since attributes are a semantic representation of a class, they are by definition an incomplete description, i.e., there is no one-to-one mapping from the feature space to the semantic space. For example, we can define the class *zebra* with the semantic description *horse with stripes*, but if we do not define the attribute *stripes*, no ZSL method will be able to learn to recognize a *zebra*, no matter how good it is (high ML). AL can be improved by changing the definition of the attributes, which involves some human annotation work but does not require additional data.

DL, or *Data Loss*, is the information loss stemming from insufficient training data for the defined attributes. For example, we can define a *zebra* as a *horse with stripes* and define the attribute *stripes* to minimize AL, but if we have no *stripes* among the seen classes, no method will be able to learn to classify a *zebra*, no matter how good it is. Improving DL can be achieved by supplementing more data and introducing novel seen classes with the required attributes.

Our proposed splitting methods alter the seen/unseen class partition and reduce the number of attributes, but we do not define new attributes. Therefore, systematic accuracy changes among different splits can be considered as changes in DL.

We can consider a second interpretation of PG as an error metric, by dividing it into the bias and noise components:

$$\mathrm{PG}^2 = \mathrm{Bias}^2 + \mathrm{Noise}^2 \tag{8}$$

As in the previous case, quantifying these components is neither feasible nor necessary, but this perspective provides a valuable intuitive framework for understanding generalizability and robustness. We can minimize bias (average error across multiple runs) by making the model more robust, and we can minimize noise (error variance across multiple runs) by making the model more generalizable. Fig. 2 qualitatively shows PG and how the accuracy varies with the split.

3.2.4. Generalizability and robustness

We define generalizability as the ability of a ZSL model to perform consistently well on seen and unseen classes, regardless of the specific split used. Generalizability analysis is straightforward, as it consists of calculating the variance of the accuracy over different splits, with a lower accuracy variance indicating higher generalizability. However, splits are not the only metric for evaluating generalizability. We will first evaluate the base variance, trivially, by training the models with different initialization seeds on a fixed split, the benchmark split, to assess how much of the variance is intrinsic to the model and how much is given by the split. We will also evaluate the model over different proportions of seen/unseen classes to test the effect of additional seen classes on the accuracy of the models. We are particularly interested in assessing how generalizability differs with different models and datasets. In our experiments, we use the unbiased estimator since the sample size is low and calculate the standard deviation, but we will keep using the term variance for brevity.

To summarize, we can define generalizability as a function of some variable X, e.g. the seed, the split, or the proportion of seen/unseen classes, as in Eq. (9) (a lower value corresponds to a more generalizable model):

$$\mathrm{Gen}(m, X) = \mathrm{Var}_X\left[acc(m, \sigma)\right] \tag{9}$$

We define robustness as the ability of a ZSL model to perform consistently well when stress-tested with particular splits that present significant variations in some statistical properties. We propose some criteria, or *splitting methods*, to produce these splits in Section 3.3. The goal is to assess the sensitivity of ZSL models to these properties, by testing performance changes without altering the architecture, the data, or the training hyperparameters. A robust ZSL model should maintain good performance relative to its average accuracy and not be sensitive to these splits. Evaluating generalizability, particularly the split variance, is a necessary step for evaluating robustness. Since robustness evaluation involves experimenting with different splits, we need to separate the effect of randomness from the effect of the particular splitting method used.

To summarize, we can define robustness as the difference between the mean accuracy over all possible splits and the minimum accuracy over a subset of carefully selected splits $\Sigma' \subset \Sigma$ as in Eq. (10) (a lower value corresponds to a more robust model):

$$\mathrm{Rob}(m, \Sigma') = \mathbb{E}\left[acc(m, \sigma)\right] - \min_{\sigma' \in \Sigma'} \left[acc(m, \sigma')\right] \tag{10}$$

We will define our subset Σ' in Section 3.3. These Eq. (9) and (10) respectively map to the noise and bias components defined in Eq. (8).

Sometimes, we will use the term generalizability to refer to both the generalizability and robustness problems. Unless otherwise specified, the meaning of the term generalizability should be either obvious from the context or irrelevant.

3.2.5. Assumptions and limitations

In the definitions above, we have made some simplifying assumptions that do not always apply, but are sufficient for our purposes and can be addressed by future research.

The first simplification is the definition of UB. It is not granted that a non-ZSL model with a similar architecture to a ZSL model trained u.r.c. is a valid UB. This is because the ZSL model employs the semantic space for additional data, which might boost accuracy beyond the defined UB. This particularly holds true for fine-grained datasets with a significant number of classes and attributes. For instance, CUB, containing about 200 classes with approximately 60 elements each, would result in subpar performance for a standard classifier when compared to a ZSL method leveraging additional semantic information.

To account for this limitation, we should consider an equivalent non-ZSL architecture that would also include the extra information in its training data, not just the samples from seen and unseen classes. This can be practically non-trivial to implement, but for our purposes, we are only interested in the implications of UB, not in evaluating it exactly, and we can just consider UB as a function of the semantic space as well. However, it is reasonable to state that for coarse-grained datasets like AWA, the accuracy in a non-ZSL setting generally surpasses the accuracy in a ZSL setting, since training a model u.r.c. directly through the feature space is easier than doing it indirectly through the semantic space.

Another assumption is that the PG decomposes into independent components. However, there are likely some non-linear relationships among those components. Similarly, there could be some non-linear relationships between the base variance and the split variance defined in the context of generalizability, given that different splits could have different base variances. In general, there is no need to quantify these relationships: our definitions are meant to be taken qualitatively rather than quantitatively, and we focus on quantifying evaluations of models trained on different splits instead. We consider the potential non-linear relationships negligible and irrelevant for our purposes, although further research might investigate them in detail.

Finally, we defined UB and PG in terms of accuracy as a single variable. In reality, accuracy alone is not sufficient to evaluate ZSL models, since we want to separate the accuracy on seen classes from the accuracy on unseen classes, as we do in Section 4. Similarly, Fig. 2 is an oversimplification, but it is useful to understand how error metrics could have some components that depend on the defined splits.

3.3. Proposed splitting methods

To thoroughly evaluate and test the robustness of ZSL models, we have developed our subset Σ' consisting of four specific splitting methods: Greedy Class Split (GCS), Clustered Class Split (CCS), Minimal Attribute Split (MAS), and PCA Attribute Split (PAS). Each of these methods is defined as a function of binary attributes for the sake of simplicity. However, it is important to notice that in our actual experiments, we use continuous attributes. Throughout this discussion, we will use the terms “splitting methods” and “splits” interchangeably to refer to these approaches. This terminology will help simplify our explanation and ensure clarity in our discussion of these methods.

The attribute splitting methods, MAS and PAS, are designed to reformulate the semantic space, denoted as A. In contrast, the class splitting methods, GCS and CCS, focus on reorganizing Y^S (the set of seen classes) and Y^U (the set of unseen classes). This reorganization is achieved by sorting the classes according to specific criteria and then selecting the first n_S classes to be included in Y^S. In Eq. (11), we define Y^S as a function of an ordered set $\tilde{Y}$, which includes all classes:

$$Y^S = \{y_i \in \tilde{Y} \mid i \leq n_S\} \tag{11}$$

With MAS and PAS, the new semantic space is defined as $\tilde{A} = \mathbb{R}^{\tilde{k}}$, and it only includes a subset of the attributes. We select the first $\tilde{k}$ attributes from an ordered set of attribute indices $K = (i_1, i_2, \dots, i_k)$. Attribute splits like MAS and PAS are inherently parametric, as we need to specify at least the number $\tilde{k}$ of attributes we want to retain.

It is important to note that class splits are employed exclusively for stress-testing models and cannot be applied “in production” to improve the performance of a ZSL model. This would require having access to samples from the unseen classes, which are unavailable by definition, otherwise it would not be a ZSL problem. Thus, if a class split indicates improvement over the baseline, it cannot be regarded as an advancement over the state of the art. In contrast, the attribute splits defined here originate from the available attributes without any extra information, meaning they can be used “in production”. As we will show in Section 4.3, our attribute splits do improve baseline results, which can be seen as an advancement over the state of the art.

We will now define the ordered set $\tilde{Y}$ derived from Y for GCS and CCS, and the new semantic space A for MAS and PAS. In Section 4, we will compare each of the resulting splits (except PAS) with its inverse by considering both ascending and descending orders of $\tilde{Y}$ or K. It is important to note that this list of splits is not meant to be exhaustive and future research could define additional methods.

3.3.1. Greedy Class Split (GCS)

The Greedy Class Split (GCS) is designed to maximize the semantic information retained in the set of seen classes Y^S. This approach specifically aims to prevent scenarios where, for example, a *zebra* is defined as a *horse with stripes*, but we have no *stripes* in the training samples.

In the binary definition of the semantic space, the value 1 indicates the presence of an attribute in an image, while the value 0 indicates its absence. Since ones are more informative than zeros, we maximize the entropy in Y^S by maximizing the norm of the signature vectors in Y^S, which is equivalent to maximizing the number of ones. For each class, we sum the values of its signature vector and we sort the classes by these sums in descending order. The resulting ordered set of classes is defined in Eq. (12):

$$\tilde{Y}_{\text{GCS}} = \text{sort}_y \left(Y, \sum_i a^i(y) \right) \tag{12}$$

3.3.2. Clustered Class Split (CCS)

The Clustered Class Split (CCS) defines Y^S and Y^U as two distinct clusters with the intent of minimizing intra-cluster distance while maximizing inter-cluster distance.

We first define the Class Semantic Distance matrix $D = (d_{i,j}) \in \mathbb{R}^{n \times n}$ where $i, j \leq n$. Then we define each element $d_{i,j} = l_2(y_i, y_j)$ as the Euclidean distance between class y_i and class y_j, where $l_2 : Y \times Y \to \mathbb{R}$ is the distance between the two class signatures (attribute vectors).

The intuition is that by, having similar classes in Y^S, the model could learn to better represent attributes as there are samples of multiple classes. In the inverse setting, Y^S contains dissimilar classes, and the intuition is that the chances of overfitting are reduced. Similar to the GCS, the clusters are defined by sorting the classes by the sum of their row (or column) values. The first n_S classes are those with the lowest distances overall, meaning that they form a cluster in the semantic space. Those classes will be the seen classes. The other n_U are far from this cluster in the semantic space, so they will form another cluster (although it is not a proper cluster since those classes are probably far away from each other as well), and they will be the unseen classes. The resulting ordered set of classes is defined in Eq. (13):

$$\tilde{Y}_{\text{CCS}} = \text{sort}_y \left(Y, \sum_i d(y, i) \right) \tag{13}$$

3.3.3. Minimal Attribute Split (MAS)

The Minimal Attribute Split (MAS) transforms the semantic space into a more compact, lower-dimensional form by filtering out attributes that are highly correlated, meaning those that frequently occur together. This process prioritizes attributes that provide unique and informative insights, thus improving the efficiency and effectiveness of the subsequent generative and classification steps by focusing on the most distinct and informative features.

We first define the Attribute Correlation matrix $H = (k_{i,j}) \in \mathbb{R}^{n\times n}$ where $i, j \leq n$. Then we define each element $s_{i,j} = |O^Y_{i,j}|/|O^Y_i|$ as the ratio of co-occurrences of attributes a^i and a^j in all the classes $y \in Y$, normalized to the occurrences of the attribute a^i, where $O^Y_i = \{y \in Y : a^i(y) = 1\}$ is the set of classes with the attribute i, and $O^Y_{i,j} = \{y \in Y : a^i(y) = 1, a^j(y) = 1\}$ is the set of classes with both attributes i and j. Due to normalization, this matrix is asymmetric.

The intuition is that highly correlated attributes provide less information, and we want to force generative models to synthesize samples conditioned on highly informative attributes. In the inverse setting, we only keep highly correlated attributes.

The resulting ordered set of indices, K_{MAS} is defined in Eq. (14):

$$K_{\mathrm{MAS}} = \mathrm{sort}_i\left(H, \sum_j s(i,j)\right) \tag{14}$$

3.3.4. PCA Attribute Split (PAS)

The PCA Attribute Split (PAS) applies the Principal Component Analysis (PCA) algorithm to the attributes, reducing the dimensionality by deriving a new set of attributes, or principal components, which capture the most significant information in the original data. For this split, we do not define an inverse.

To perform PCA, the attribute matrix A_Y containing the attribute vectors of all classes is first normalized. Next, the covariance matrix A_{COV} is computed, followed by its eigendecomposition. The resulting eigenvalues and eigenvectors correspond to the amount of variance and the principal components, respectively. Unlike MAS, where we just remove some attributes, here we derive new attributes from the existing ones (the principal components). After sorting the eigenvalues in descending order, the top $\tilde{k}$ eigenvectors associated with the highest eigenvalues are selected to form the new attribute matrix $A_{\tilde{k}}$ serving as the semantic space for the PAS split.

The resulting ordered set of attribute indices, K_{PAS} is defined in Eq. (15), where $\mathbf{v}$ denotes the eigenvectors and λ denotes the eigenvalues:

$$K_{\mathrm{PAS}} = \mathrm{sort}_i\left(\mathbf{v}_{A_{\mathrm{COV}}}, \lambda_{A_{\mathrm{COV}}}\right) \tag{15}$$

In the next section, we will present the results of our experiments using these proposed splitting methods and discuss their impact on zero-shot learning performance. We will analyze the potential benefits and limitations of each split and provide insights into how they affect the model's ability to generalize to unseen classes.

4. Results and discussion

The following section provides a comprehensive analysis and discussion of a series of experiments designed to evaluate the generalizability and robustness of the following ABZSL models: CLSWGAN (Xian, Lorenz, Schiele, & Akata, 2018), TF-VAEGAN (Narayan et al., 2020), and FREE (Chen et al., 2021).

The organization of this section is as follows: Section 4.1 details the datasets and evaluation metrics employed, Section 4.2 illustrates the generalizability experiments, and Section 4.3 illustrates the robustness experiments. Moreover, we demonstrate that, in some scenarios, our proposed attribute splits deliver better performance on fine-grained datasets compared to base models. This suggests that having a large number of attributes could be counterproductive, and employing dimensionality reduction techniques may be beneficial.

4.1. Technical details and experiments setup

In our evaluations, we focused on four benchmark datasets commonly referenced in the ABZSL literature: AWA2 (henceforth referred to as AWA) (Xian, Lampert, Schiele, & Akata, 2018), CUB (Welinder et al., 2010), FLO (Nilsback & Zisserman, 2008), and SUN (Patterson & Hays, 2012). Each of these datasets offers distinct characteristics that are critical for a comprehensive understanding of our results. AWA is a coarse-grained dataset with a limited number of classes and attributes but a large number of samples per class. Conversely, CUB, SUN, and FLO are considered fine-grained datasets, but each exhibits unique characteristics that will be significant for an accurate interpretation of our results. SUN has the highest number of classes with a comparatively low number of attributes, while FLO, on the other hand, has the highest number of attributes with a relatively low number of classes. CUB represents a middle ground in terms of class and attribute numbers. By evaluating different splits across these varied datasets, we aim to shed light on how granularity, in terms of class and attribute counts, affects model performance. The following are the specific details of each dataset:

- AWA: 85 attributes, 40 seen classes, 10 unseen classes, and 30,475 instances.
- CUB: 312 attributes, 150 seen classes, 50 unseen classes, and 11,788 instances.
- FLO: 1024 attributes, 82 seen classes, 20 unseen classes, and 8,189 instances.
- SUN: 102 attributes, 645 seen classes, 72 unseen classes, and 14,340 instances.

For our experiments, we train the models with 30 epochs for AWA, 56 for CUB, 80 for FLO, and 40 for SUN, to be consistent with the number of epochs used by the three works we draw upon. For GZSL evaluation, we consider the top-1 accuracy for both seen (S) and unseen (U) classes and calculate the harmonic mean (H) of the two. Although S is typically higher than U, an improvement in the latter is usually more desirable, provided it does not compromise the accuracy of S. For each experiment, we present the results of the epoch with the highest H. Before discussing the experiments, we provide the baseline results of our implementation of the models, evaluated on the benchmark split, in Table 1.

We train the models on features of size 2048 extracted from the ImageNet-1K (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009) pre-trained ResNet-101 (He, Zhang, Ren, & Sun, 2015). All the modules (generator, discriminator, etc.) are implemented as two-layer fully-connected networks with 4096 hidden units, while the classifiers are implemented as single-layer networks. LeakyReLU activation is used for all the hidden layers. The networks are trained using the Adam optimizer with varying learning rates depending on the dataset and model. For further technical details and more information about the learning rates and other hyperparameters, refer to the original works (Chen et al., 2021; Narayan et al., 2020; Xian, Lorenz, Schiele, & Akata, 2018) or our implementation at https://github.com/luca-rossi/grabzsl.

For an accurate comparison between splits, we utilize the results obtained from our implementation of the models as a baseline, instead of referring to the results mentioned in the original papers. This is to account for potential slight differences arising from differing hyperparameters and epochs.

In general, S accuracy is higher than U, because the models are biased towards recognizing seen classes. The best overall GZSL performance is achieved on the FLO dataset, followed by AWA, CUB, and SUN.

4.2. Generalizability evaluation

As previously stated in Section 3.3, we evaluate generalizability by measuring the base variance and the split variance. Base variance is calculated by training the models on 5 random initialization seeds while keeping the (benchmark) split fixed. Changing the initialization seed affects the generative training as well as the downstream classifier, meaning that the synthetic dataset generated by the former to train the latter will be different each time. On the other hand, split variance is calculated by training the models on 5 random class splits, introducing

Table 1
Baseline accuracy of the CLSWGAN (C), TF-VAEGAN (T), and FREE (F) models.

Model	CZSL				GZSL											
	AWA	CUB	FLO	SUN	AWA			CUB			FLO			SUN		
	T1	T1	T1	T1	S	U	H	S	U	H	S	U	H	S	U	H
C base	68.2	57.0	65.5	58.9	68.0	54.1	60.2	58.3	44.1	50.2	81.8	53.6	64.8	35.7	43.5	39.3
T base	63.4	61.2	66.8	65.4	80.0	49.7	61.3	59.2	51.9	55.3	81.9	58.8	68.5	38.1	45.8	41.6
F base	63.3	61.2	62.2	51.7	65.3	56.4	60.5	59.3	49.4	53.9	78.7	57.5	66.5	32.6	35.8	34.1

Table 2
Base and split variance. These values show the normalized standard deviation (multiplied by 100 for clarity) for our experiments, obtained on samples of 6 training runs (including the base one).

Model	CZSL				GZSL											
	AWA	CUB	FLO	SUN	AWA			CUB			FLO			SUN		
	T1	T1	T1	T1	S	U	H	S	U	H	S	U	H	S	U	H
C base	0.632	0.248	0.861	0.498	1.506	0.965	0.362	1.935	1.019	0.228	2.529	1.891	0.668	0.876	1.396	0.179
C split	6.195	1.917	2.013	2.253	2.874	5.317	3.937	2.941	2.601	0.985	2.761	2.486	2.077	0.792	1.835	0.588
T base	0.657	0.264	0.500	3.576	0.520	1.141	0.762	1.765	1.354	0.098	2.245	1.172	0.187	1.566	3.742	2.195
T split	5.039	2.217	4.773	4.397	5.591	6.558	3.342	1.176	1.545	1.258	1.899	5.430	3.904	1.373	4.062	2.082
F base	2.097	0.248	0.725	0.914	4.905	2.856	0.216	1.873	1.286	0.190	1.070	0.911	0.397	1.376	1.990	0.147
F split	3.734	2.367	4.139	2.121	4.397	3.695	2.561	1.868	1.925	1.103	1.989	4.733	3.488	2.396	3.054	0.452

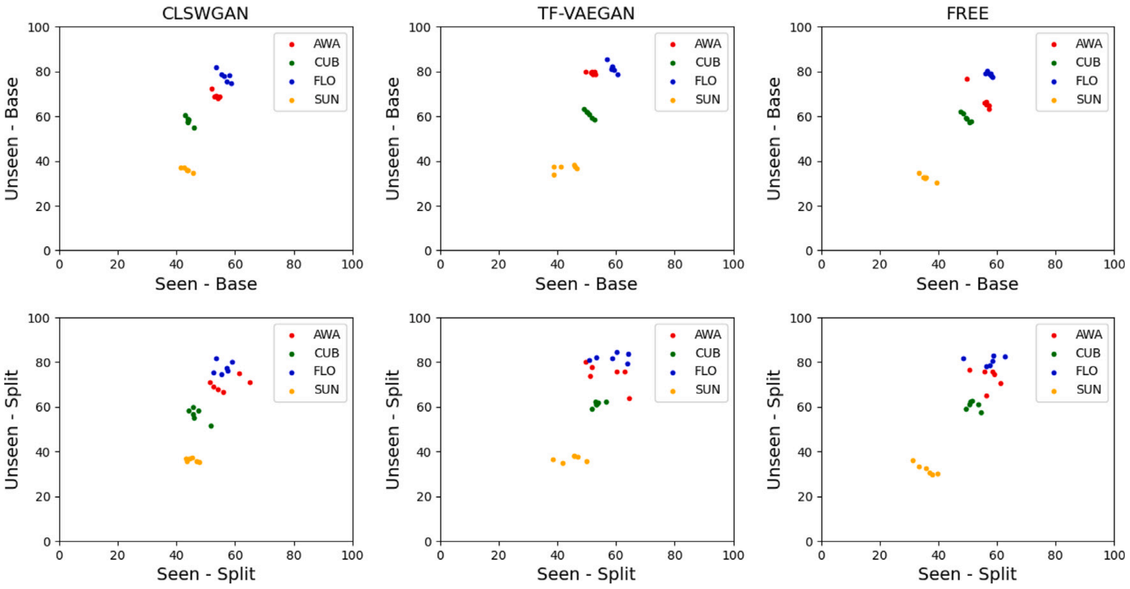

Fig. 3. Scatter plots of our experiments for base variance (top) and split variance (bottom). We only show the GZSL experiments. The x-axis shows the accuracy on seen classes, while the y-axis shows the accuracy on unseen classes.

an additional source of noise. Furthermore, we examine how accuracy varies with the seen/unseen class proportion. Generalizability analysis is not only important in its own right, but it is also a prerequisite for evaluating robustness, as the split variance alone could significantly account for differences in accuracy emerged from our proposed splits. If the split variance is too large, the effects of robustness may be difficult to discern.

Table 2 and Fig. 3 show the base variance and the split variance. The calculation of the base variance is performed in order to understand the intrinsic variance of a model with a fixed split; in this case, the benchmark split. Despite the low sample size could make these results noisy, we can still observe some trends among models and datasets:

- Typically, the base variance is lower than the split variance due to the additional noise introduced by the split. Nevertheless, the base variance is not negligible, and its impact depends on the specific dataset.
- Both in CZSL and GZSL, AWA emerges as the noisiest dataset, with the highest median base and split variance. SUN, FLO, and CUB follow in that order. A particular outlier in our experiments causes the base variance for TF-VAEGAN with SUN to be especially high.
- There are no obvious consistent results among models, so we can assume that different models do not have a significant effect on variance, and the differences depicted in the table can be mostly attributed to noise.
- The harmonic mean tends to be much lower than both the seen and unseen accuracy. This suggests that, when accuracy decreases on unseen (or seen) classes, the accuracy increases on seen (or unseen) classes, indicating an amplified (or reduced) bias towards seen classes.
- From Fig. 3 we can observe that random class splits tend to outperform the baseline results in Table 1. We can explain this phenomenon in terms of entropy and robustness. The methods defined in Section 3.3 produce splits with specific properties, and thus have low entropy. Random splits, by definition, have high entropy. As we will illustrate in Section 4.3, our tested models tend to perform worse on these class splits, as expected. We can observe a trend whereby the accuracy tends to increase as the

Table 3
Seen/unseen classes proportion experiments (fixed number of unseen classes: 5 for AWA, 25 for CUB, 10 for FLO, 36 for SUN).

Model	CZSL				GZSL											
	AWA	CUB	FLO	SUN	AWA			CUB			FLO			SUN		
	T1	T1	T1	T1	S	U	H	S	U	H	S	U	H	S	U	H
C prop 1	76.9	62.1	65.4	70.3	72.5	57.7	64.3	61.1	53.0	56.8	80.6	54.0	64.7	37.5	54.0	44.3
C prop 2	72.2	71.9	77.4	76.3	65.9	60.4	63.0	65.2	56.1	60.3	83.3	67.3	74.4	38.2	57.1	45.7
C prop 3	88.9	71.3	83.7	78.1	81.0	70.5	75.4	58.1	58.2	58.2	82.1	70.5	75.9	38.9	55.3	45.7
C prop 4	97.6	69.6	65.0	75.7	64.3	75.2	69.3	56.9	58.4	57.7	84.1	56.0	67.2	38.0	55.4	45.1
T prop 1	85.5	66.7	80.4	72.5	81.0	57.8	67.4	66.9	55.7	60.8	87.8	67.3	76.2	38.7	53.9	45.0
T prop 2	69.3	78.1	74.3	78.9	78.7	48.4	59.9	67.1	66.3	66.7	86.0	68.9	76.5	42.5	53.6	47.4
T prop 3	81.1	78.8	80.7	81.0	84.7	61.5	71.3	62.1	68.5	65.1	81.4	74.8	78.0	41.0	57.8	48.0
T prop 4	96.6	73.2	64.2	76.7	66.2	77.1	71.3	66.5	57.7	61.8	86.1	56.0	67.9	38.8	52.5	44.6
F prop 1	84.8	66.0	80.4	60.1	84.8	56.3	67.7	65.9	55.3	60.1	86.0	74.6	79.9	35.2	37.4	36.3
F prop 2	67.8	77.4	78.1	68.5	74.0	52.5	61.4	67.6	64.9	66.2	86.4	71.6	78.3	38.3	44.3	41.1
F prop 3	78.5	77.4	77.8	72.9	79.9	71.3	75.3	64.0	62.6	63.3	82.9	67.2	74.2	36.9	43.2	39.8
F prop 4	96.8	71.4	58.4	69.6	76.3	71.8	74.0	61.4	60.9	61.1	81.1	55.9	66.1	33.7	45.8	38.8

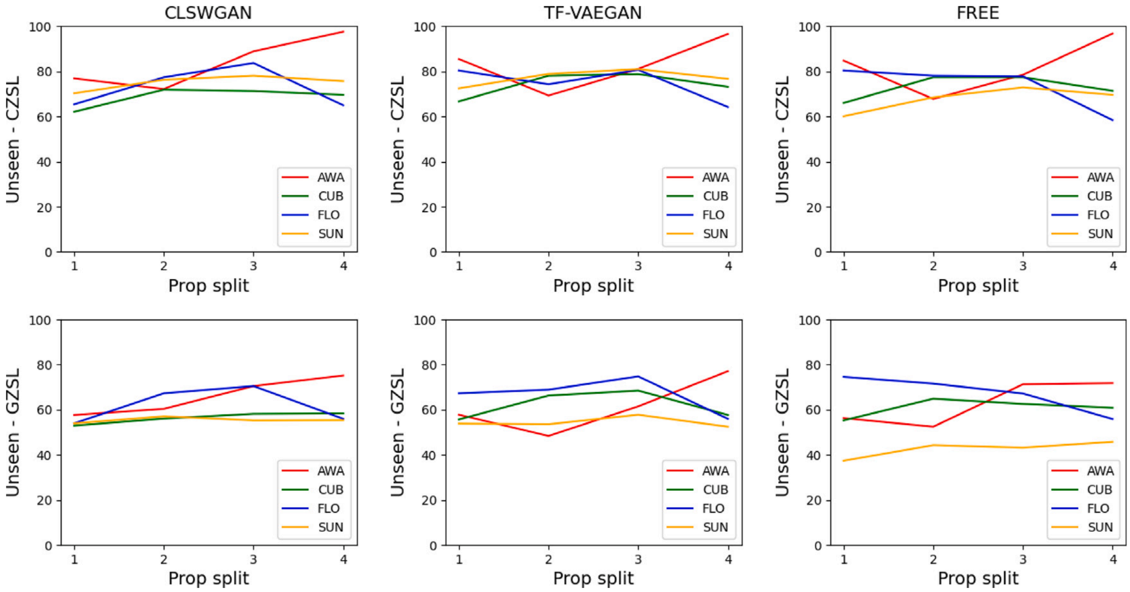

Fig. 4. Plots of our experiments with different seen/unseen class proportions. The x-axis shows the split as defined earlier (e.g. 1 for AWA is 30-5), while the y-axis shows the CZSL accuracy (top) and the GZSL unseen accuracy (bottom).

splits approach maximum entropy. If the benchmark split has been engineered rather than defined randomly, it will have lower entropy and the tested models will be less robust, with slightly degraded performance.

We define the proportion splits as follows (each pair is the number of seen and unseen classes):

- Prop 1: AWA 30/5, CUB 100/25, FLO 62/10, SUN 573/36.
- Prop 2: AWA 35/5, CUB 125/25, FLO 72/10, SUN 609/36.
- Prop 3: AWA 40/5, CUB 150/25, FLO 82/10, SUN 645/36.
- Prop 4: AWA 45/5, CUB 175/25, FLO 92/10, SUN 681/36.

Smaller sets of seen classes are always subsets of larger ones. For instance, the seen classes of the 35/5 AWA split include all the seen classes of the 30/5 AWA split, plus five other randomly selected classes. The number of unseen classes is kept constant to avoid artificially increasing unseen accuracy by reducing the class choices for the model. For a given dataset, the set of unseen classes is always the same.

The experiments in Table 3 and Fig. 4 show the relationship between accuracy and the number of seen classes. Here are some observations:

- On average, both CZSL and GZSL unseen accuracy increase with the number of seen classes, up to a point. This could be due to an increase in the seen class bias as the ability to generalize improves with the number of seen classes. Initially, the ability to generalize increases more rapidly than the seen class bias, but over time, diminishing returns set in, and the seen class bias starts to dominate. The only exception seems to be AWA, where the accuracy keeps increasing, likely because the number of classes is lower.
- Some datasets have larger improvements than others. AWA, for instance, shows the most significant improvements, while SUN largely remains stable. This is consistent with the fact that AWA has the fewest classes, while SUN has the most. FLO displays the most substantial average decrease in accuracy.
- Some noise can be observed in the GZSL seen accuracy (potentially explained by split variance), but overall no particular trend is apparent. We might have expected GZSL seen accuracy to decrease as the number of seen classes grows, considering there are more classes to choose from, but this is not the case.
- For the AWA dataset, the CZSL accuracy gain is particularly pronounced, achieving 97.6% accuracy with 45 seen classes. Again, this is likely because AWA has a limited number of classes and attributes, resulting in fewer diminishing returns as the number of seen classes increases.
- There are some exceptions to these trends. For example, the AWA dataset is particularly noisy, as seen from the earlier experiments

with base and split variance. FLO with the FREE model shows an inverse trend, where accuracy deteriorates as the number of seen classes increases, likely because its baseline accuracy is already higher than the other datasets.

These experiments suggest that an increase in seen classes can lead to improved model accuracy, and these improvements can sometimes be substantial. However, this effect plateaus as increasing the number of seen classes also intensifies the seen class bias.

The generalizability experiments presented here reveal that noise and the type of dataset often play a substantial role in determining the accuracy of ZSL models. The variance in results can sometimes eclipse methodological advancements, indicating the need for more stringent evaluation criteria. We found that coarse-grained datasets like AWA are more prone to noise, while fine-grained datasets tend to be more stable. Furthermore, we illustrated how increasing the number of seen classes can improve the generalizability of ZSL models to unseen classes, but this effect is quickly overtaken by the seen class bias.

4.3. Robustness evaluation

The subsequent experiments aim to evaluate the models using the proposed splits, as discussed in Section 3.3. GCS and CCS are non-parametric splits, while MAS and PAS are parametric, as they require the definition of the number of attributes $\tilde{k}$. Considering the non-negligible noise component present in our experiments, as demonstrated in Section 4.2, we deem the results significant only if they deviate significantly from the baseline or display consistency across various datasets and models.

To reduce verbosity, we will use the notation $acc_{\text{[Type][Model][Dataset]}}$, where:

- Type can be S (seen GZSL), U (unseen GZSL), or Z (CZSL).
- Model can be C (CLSWGAN), T (TFVAEGAN), or F (FREE).
- Dataset can be A (AWA), C (CUB), F (FLO), or S (SUN).

For instance, acc_{UCA} can be interpreted as "the accuracy on unseen classes for the model CLSWGAN on the dataset AWA". An asterisk in a given position signifies "any", for instance, $acc_{\text{U*A}}$ can be interpreted as "the accuracy on unseen classes for any model on the dataset AWA". We use the suffix *inv* to denote the inverse of a split, for example, GCS_{inv} means inverse GCS.

For each of the parametric splits, we carry out tests with two distinct values for $\tilde{k}$:

- MAS^1: AWA 40, CUB 150, FLO 500, SUN 50
- MAS^2: AWA 20, CUB 75, FLO 250, SUN 25
- PAS^1: AWA 40, CUB 150, FLO 100, SUN 50
- PAS^2: AWA 20, CUB 75, FLO 50, SUN 25

Below, we outline some general observations. Since comments on unseen accuracy for GZSL often mirror those for CZSL, we omit the latter for brevity, except when discrepancies between the two arise:

- The results presented in Table 4 indicate some level of variability when the experiments are conducted on the proposed splits, suggesting that the models may not be perfectly robust.
- The $acc_{\text{U**}}$ variance generally surpasses the $acc_{\text{S**}}$ variance considerably. This is expected, as classifying seen classes does not differ significantly from a typical classification task in a non-ZSL setting. This further suggests that the lack of robustness is specific to the ZSL setting (inferring different unseen classes), and cannot merely be attributed to the complexity of the dataset as in a typical classification task.
- $acc_{\text{S**}}$ accuracy tends to improve with most splits, either due to an increased seen class bias or because the split itself improves generalization. A few instances exist where $acc_{\text{S**}}$ accuracy decreases by more than 10%, specifically acc_{STA} with MAS and acc_{SCF} with GCS, which could be attributed to noise.
- Overall, a trend seems to emerge whereby $acc_{\text{U**}}$ tends to be lower for the proposed class splits compared to the benchmark split. This implies that the models are not particularly robust to these types of splits. In contrast, attribute splits occasionally improve baseline accuracy.
- Regarding $acc_{\text{U**}}$, most improvements over the baseline can be attributed to noise, for instance, acc_{UCF} with CCS_{inv}. Consequently, we are more interested in consistent improvements (such as the PAS splits consistently displaying improvements across most datasets and models) and large ones (e.g., acc_{UFS} with PAS^1 shows a 7.7% improvement over the baseline).
- In numerous cases, both a split (e.g., CCS) and its inverse (e.g., CCS_{inv}) underperform the baseline. This indicates that the correlation between the class splits and robustness is non-linear, and we speculate that it follows an "inverse U" pattern. Both extremes (the split and its inverse) have low entropy and thus low accuracy, while randomly generated splits tend to be in the middle with the highest entropy and accuracy. The benchmark split, which is engineered rather than completely random, probably retains some biases that lower its entropy, and thus is somewhere in between. Future research could empirically validate this.
- Regarding CZSL results, PAS^1 improves the baseline on all datasets with the FREE model. Regarding GZSL results, a few Pareto improvements over the baseline are noticeable, specifically $acc_{\text{*TF}}$ with PAS^1 and PAS^2, $acc_{\text{*FF}}$ with CCS_{inv}, GCS, and PAS^2, and $acc_{\text{*FS}}$ with PAS^1.
- In general, there are no significant differences between models. This implies that robustness might be an intrinsic property of the datasets and our results are generalizable across models, at least across the ones we examined in this work. One possible exception occurs with the SUN dataset, where the PAS split improves the baseline with FREE but not with CLSWGAN and TF-VAEGAN. A likely explanation is that dimensionality reduction aids faster convergence, and FREE is a slower model that had not yet converged on the baseline when we stopped training.
- Conversely, significant differences between datasets exist. From a qualitative perspective, the fine-grained datasets are more robust than the coarse-grained AWA, which exhibits higher variance among different splits. Datasets such as CUB and SUN have substantially more classes than AWA, implying that less semantic information is lost when altering the split, resulting in increased robustness. If information about a specific attribute is spread among numerous classes, we are less likely to face a situation where not enough images include that attribute (contributing to the DL component we described in Section 3.2). The dataset FLO, which possesses a large number of attributes, shows consistent accuracy improvements with various splits, particularly attribute splits.

The following are some split-specific observations:

- CCS always worsens $acc_{\text{U**}}$. Sometimes, it increases seen accuracy (e.g. acc_{SCC}), suggesting that this split tends to increase the seen class bias. CCS_{inv} decreases $acc_{\text{U*A}}$ and $acc_{\text{U*S}}$, but leaves $acc_{\text{U*C}}$ and $acc_{\text{U*F}}$ unaffected (there is even a slight increase for acc_{UFC} and acc_{UFF}). This is consistent with all three models, suggesting a dataset trend. It also increases $acc_{\text{S**}}$ in multiple cases, pointing towards an increase in seen class bias. A correlation appears to exist between the number of attributes and the effect of CCS_{inv}. Fewer attributes yield a worse effect on accuracy (CUB and FLO have more attributes than AWA and SUN).
- Generally, both GCS and GCS_{inv} negatively affect $acc_{\text{U**}}$, with $acc_{\text{S**}}$ fluctuating. Both acc_{UFF} and acc_{SFF} see an increase, likely attributed to noise.

Table 4
Robustness results. The top results refer to CLSWGAN, the middle ones to TF-VAEGAN, and the bottom ones to FREE. The best epoch has been chosen for the results (with the highest harmonic mean). The results that outperform the baseline are in bold, while those that underperform it by over 10% are underlined.

Model	CZSL				GZSL											
	AWA	CUB	FLO	SUN	AWA			CUB			FLO			SUN		
	T1	T1	T1	T1	S	U	H	S	U	H	S	U	H	S	U	H
C base	68.2	57.0	65.5	58.9	68.0	54.1	60.2	58.3	44.1	50.2	81.8	53.6	64.8	35.7	43.5	39.3
C CCS	50.8	33.8	48.3	36.3	67.1	44.2	53.3	**62.1**	30.4	40.8	78.7	41.6	54.4	34.2	30.8	32.4
C CCS$_{inv}$	42.2	52.0	64.3	36.2	65.7	39.5	49.4	58.1	41.5	48.4	80.9	**54.4**	**65.1**	**39.0**	34.0	36.3
C GCS	53.8	43.8	59.1	43.5	**69.6**	47.0	56.1	56.2	39.3	46.3	71.7	51.9	60.2	34.2	38.4	36.2
C GCS$_{inv}$	56.0	44.0	40.3	47.0	**69.0**	53.6	**60.3**	53.6	38.6	44.9	78.4	39.5	52.5	**40.1**	36.9	38.4
C MAS^{1}	64.9	49.3	**68.1**	49.9	67.7	53.1	59.5	53.4	38.6	44.8	78.2	**61.7**	**68.9**	31.0	36.6	33.6
C MAS$^{1}_{inv}$	56.6	55.6	65.3	49.8	62.7	43.5	51.3	52.9	**44.6**	48.4	79.8	**57.7**	**66.9**	32.6	34.8	33.7
C MAS^{2}	66.5	43.3	**66.5**	37.4	**69.6**	47.9	56.8	55.5	32.6	41.1	78.7	**60.9**	**68.7**	30.3	27.5	28.8
C MAS$^{2}_{inv}$	51.6	49.0	**66.6**	36.9	63.9	36.3	46.3	56.1	36.9	44.5	79.2	**59.5**	**68.0**	30.9	26.3	28.4
C PAS^{1}	65.5	55.9	64.1	58.5	**68.8**	52.0	59.2	56.7	**45.5**	**50.5**	81.1	**56.7**	**66.7**	34.9	39.5	37.1
C PAS^{2}	**68.3**	54.7	62.3	56.4	63.3	53.5	58.0	**58.6**	42.1	49.0	80.4	54.1	64.7	33.2	38.5	35.7
T base	63.4	61.2	66.8	65.4	80.0	49.7	61.3	59.2	51.9	55.3	81.9	58.8	68.5	38.1	45.8	41.6
T CCS	45.2	37.7	42.8	39.4	**83.8**	34.4	48.8	**66.5**	34.4	45.4	**85.7**	37.5	52.2	37.8	31.3	34.3
T CCS$_{inv}$	43.3	60.3	64.3	40.2	71.1	36.1	47.9	**64.4**	50.3	**56.4**	**86.1**	55.9	67.8	**43.0**	35.4	38.8
T GCS	52.5	52.4	62.9	46.7	78.8	41.8	54.6	**63.6**	47.2	54.2	81.3	53.9	64.8	37.1	38.7	37.9
T GCS$_{inv}$	55.8	51.9	47.7	51.0	73.6	49.6	59.3	**62.4**	46.1	53.0	78.9	46.3	58.4	37.8	43.0	40.3
T MAS^{1}	60.2	54.5	66.4	58.5	69.2	43.6	53.5	**60.9**	43.1	50.5	**84.4**	57.7	**68.6**	33.8	42.4	37.6
T MAS$^{1}_{inv}$	55.8	58.8	**67.3**	56.7	72.1	37.4	49.2	**61.8**	48.2	54.2	**84.6**	58.1	68.9	35.3	38.2	36.7
T MAS^{2}	**71.0**	50.8	64.4	44.5	68.7	**55.3**	61.3	54.1	41.1	46.8	**83.5**	55.7	66.8	32.4	29.1	30.7
T MAS$^{2}_{inv}$	**63.8**	55.3	64.7	46.9	80.0	47.3	59.5	58.4	43.5	49.8	80.4	57.2	66.9	32.7	29.9	31.3
T PAS^{1}	57.9	**62.4**	64.6	63.5	72.2	43.5	54.3	**59.5**	51.9	55.4	**84.1**	**59.5**	**69.7**	37.3	42.9	39.9
T PAS^{2}	**66.4**	**61.8**	66.0	61.3	70.4	48.9	57.7	**61.0**	50.0	55.0	**83.0**	58.8	**68.8**	37.3	38.3	37.8
F base	63.3	61.2	62.2	51.7	65.3	56.4	60.5	59.3	49.4	53.9	78.7	57.5	66.5	32.6	35.8	34.1
F CCS	46.0	37.4	41.1	28.8	**77.2**	41.0	53.5	**65.5**	33.9	44.7	**82.9**	38.8	52.8	**34.4**	21.3	26.3
F CCS$_{inv}$	43.0	58.6	**65.7**	29.7	**69.1**	39.0	49.9	59.1	**52.2**	**55.4**	**85.7**	**62.8**	**72.5**	**37.5**	24.4	29.6
F GCS	51.9	50.2	**62.4**	32.7	**77.4**	41.7	54.2	**60.8**	46.0	52.4	**81.4**	**58.3**	**67.9**	30.4	25.6	27.8
F GCS$_{inv}$	56.0	51.0	46.3	41.7	**67.8**	53.4	59.7	**61.7**	44.0	51.4	**79.8**	44.3	57.0	**33.9**	34.0	34.0
F MAS^{1}	58.6	52.9	61.5	45.5	**66.7**	54.2	59.8	55.7	43.7	49.0	77.5	57.2	65.8	30.7	30.9	30.8
F MAS$^{1}_{inv}$	55.1	58.2	60.9	43.5	**68.0**	43.8	53.3	**59.9**	46.5	52.3	75.9	56.7	64.9	28.2	28.8	28.5
F MAS^{2}	54.3	48.4	**63.5**	34.4	63.8	45.3	53.0	53.2	38.5	44.7	74.1	**58.0**	65.1	28.8	22.4	25.2
F MAS$^{2}_{inv}$	51.5	53.6	60.2	32.8	**66.3**	35.8	46.5	58.2	41.6	48.5	73.0	54.5	62.4	30.3	20.1	24.2
F PAS^{1}	**66.9**	**61.7**	**63.6**	**59.7**	**79.1**	52.2	**62.9**	**61.2**	48.7	**54.3**	**80.7**	56.8	**66.7**	32.6	**43.5**	**37.2**
F PAS^{2}	64.5	59.5	**64.9**	**56.6**	**68.6**	55.3	**61.2**	58.2	48.8	53.1	78.7	**58.1**	**66.8**	31.9	**39.6**	**35.3**

- The performance of both MAS and MAS$_{inv}$ is characterized by a degree of unpredictability, as they can either improve or diminish accuracy depending on the context. For instance, MAS and MAS$_{inv}$ notably improve acc_{UCF}, but leave acc_{UTF} and acc_{UFF} unaffected. There is also a noticeable improvement in acc_{ZTA} with MAS^{2}. These inconsistent results can be attributed to the propensity of MAS and MAS$_{inv}$ to generate a semantic space with a higher susceptibility to noise, suggesting that experiments performed in this kind of semantic space may not be statistically significant.
- However, an alternative explanation for the good performance of MAS and MAS$_{inv}$ on acc_{UCF} may be dimensionality reduction. The observed improvements remain consistent in both the split and its inverse, suggesting that they may not be directly attributable to the specific methodology that produces the MAS split, but rather to the mere reduction in the number of attributes. Given that FLO possesses the highest number of attributes (1024), it is plausible that models struggle with large attribute sets, and thus, derive benefits from dimensionality reduction.
- Like MAS, PAS also performs dimensionality reduction within the attribute space. Unlike MAS, however, PAS does not appear to adversely affect $acc_{\mathrm{U**}}$. On the contrary, it consistently elevates accuracy beyond the baseline for the FLO and SUN datasets across all models.

The robustness experiments presented here are generally consistent across different models but not across datasets, where performance varies based on the granularity of the datasets themselves. Class splits often decrease unseen accuracy, suggesting considerable room for improvement in model robustness as the accuracy drop often exceeds 10%. Conversely, attribute splits such as MAS and PAS can increase the unseen accuracy of the base model if the initial attribute count is high (as in FLO). Since dimensionality reduction methods can be applied without accessing information from unseen classes, we can consider it as an effective low-hanging fruit for improving the accuracy of ZSL models used in real-world applications. This can be achieved either by reducing the number of attributes so that the remaining ones retain semantic meaning, like with MAS, or by transforming the attributes to achieve higher improvements at the cost of losing semantic meaning, like with PAS. The trends observed here hold across all three models, suggesting a generalizable pattern rather than a noise-induced phenomenon. A summary of the effects of these splitting methods is presented in Table 5.

5. Conclusions and future work

This study introduced the concepts of generalizability and robustness for ABZSL. We conducted various experiments on four datasets to test the impact of granularity: one coarse-grained, AWA, and three fine-grained, CUB, FLO, and SUN. Similarly, we used three ABZSL models to ensure consistency in our results: CLSWGAN, TF-VAEGAN, and FREE.

Our generalizability experiments revealed some non-negligible performance variability in ABZSL models across different splits, indicating that they may not be as generalizable as previously thought. This effect was particularly pronounced in the coarse-grained AWA, where greater information loss and fewer classes and attributes negatively impacted generalizability. Furthermore, we showed that increasing the number of seen classes generally has a positive effect on accuracy, until diminishing returns set in and this effect gets dominated by the seen class bias.

We proposed four splitting methods to stress test the robustness of the models, and we observed that they can significantly impact

Table 5
Summary of the effects of the proposed splitting methods.

Split	Effect
GCS	Generally worsens unseen accuracy, with fluctuating effects on seen accuracy, indicating occasional increases in seen class bias. Both GCS and its inverse have this effect, suggesting that both extremes hamper the model's ability to generalize to unseen classes. Some combinations of models and datasets illustrate larger decreases in accuracy than others. Rare increases in unseen accuracy are likely due to noise.
CCS	Similarly to GCS, generally worsens unseen accuracy, with fluctuating effects on seen accuracy, indicating occasional increases in seen class bias. Both CCS and its inverse have this effect, suggesting that both extremes hamper the model's ability to generalize to unseen classes. Some combinations of models and datasets illustrate larger decreases in accuracy than others. Rare increases in unseen accuracy are likely due to noise.
MAS	Yields unpredictable effects on accuracy, potentially improving or diminishing it depending on the model and dataset. This effect is observed in both MAS and its inverse, suggesting that this split may create a semantic space prone to noise. Improvements in accuracy are likely a consequence of dimensionality reduction, rather than the specific nature of the MAS split itself. This effect is more pronounced in fine-grained datasets like FLO, suggesting that dimensionality reduction is only effective when the original semantic space is particularly large.
PAS	Consistently improves accuracy beyond the baseline for the FLO and SUN datasets, suggesting that models can generalize more effectively to unseen classes with a PCA-generated semantic space, and this effect is particularly noticeable when the original semantic space is particularly large. Although accuracy improvements with PAS are larger and more consistent than those we obtain with MAS, it is worth noting that the semantic space generated with PAS has no semantic meaning, which could be undesirable in some ZSL applications.

accuracy. In particular, two of these methods, GCS and CCS, produce class splits, and they negatively impact the performance of the model, thereby revealing opportunities for improving robustness in future research. On the other hand, the other two methods, MAS and PAS, produce attribute splits, and sometimes they improve the accuracy of the base models, indicating that dimensionality reduction in the semantic space can improve SOTA results, particularly for fine-grained datasets like FLO. Therefore, dimensionality reduction could be an effective low-hanging fruit for improving the accuracy of ZSL models used in real-world applications.

Overall, our findings underline the potential for improving the generalizability and robustness of ABZSL models. This work has the potential to open new avenues of research in ABZSL generalizability evaluation: more experiments are needed to better understand the factors that contribute to the variability in performance observed in our work, and to develop more robust ABZSL models that can effectively transfer knowledge across a wide range of training conditions. Moreover, future research could define new splitting methods, conduct a more comprehensive evaluation of parametric splitting methods (e.g. MAS), and assess robustness across different seen/unseen class proportions. The relationship between split entropy and accuracy should be further investigated: random splits slightly but consistently outperform the benchmark split, potentially because the engineered benchmark split has lower entropy and thus higher information loss. It could also be worth exploring more sophisticated dimensionality reduction techniques that retain the advantages of both MAS and PAS: consistently improving accuracy like the latter, but retaining semantic meaning like the former. Finally, we encourage the development of unique approaches to extend this analysis to other categories of ZSL not considered in this study, for instance, ZSL based on learned attributes rather than engineered ones.

CRediT authorship contribution statement

Luca Rossi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Maria Chiara Fiorentino:** Resources, Visualization, Writing – review & editing. **Adriano Mancini:** Supervision. **Marina Paolanti:** Supervision. **Riccardo Rosati:** Resources, Visualization, Writing – review & editing. **Primo Zingaretti:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The datasets analyzed during the current study are available at https://drive.google.com/drive/folders/16Xk1eFSWjQTtuQivTogMmvL3P6F_084u, last access 2023-07-18. The code to reproduce our results is available at https://github.com/luca-rossi/grabzsl.

References

Akata, Z., Harchaoui, Z., & Schmid, C. (2015). Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*, 1425–1438. http://dx.doi.org/10.1109/TPAMI.2015.2487986.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223). PMLR.

Changpinyo, S., Chao, W.-L., Gong, B., & Sha, F. (2020). Classifier and exemplar synthesis for zero-shot learning. *International Journal of Computer Vision*, *128*, 166–201. http://dx.doi.org/10.1007/s11263-019-01193-1.

Chen, S., Wang, W., Xia, B., Peng, Q., You, X., Zheng, F., et al. (2021). Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 122–131).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). URL https://api.semanticscholar.org/CorpusID:57246310.

Felix, R., Reid, I., & Carneiro, G. (2018). Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European conference on computer vision* (pp. 21–37).

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., et al. (2013). DeViSE: A deep visual-semantic embedding model. In *Advances in neural information processing systems*: *vol. 26*.

Fu, Y., Hospedales, T. M., Xiang, T., & Gong, S. (2015). Transductive multi-view zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(11), 2332–2345. http://dx.doi.org/10.1109/TPAMI.2015.2408354.

Ge, Y., Ren, J., Gallagher, A., Wang, Y., Yang, M.-H., Adam, H., et al. (2023). Improving zero-shot generalization and robustness of multi-modal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11093–11101).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems*: *vol. 27*.

Gowda, S. N. (2023). Synthetic sample selection for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 58–67).

Han, Z., Fu, Z., Chen, S., & Yang, J. (2021). Contrastive embedding for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2371–2381).

Han, Z., Fu, Z., Chen, S., & Yang, J. (2022). Semantic contrastive embedding for generalized zero-shot learning. *International Journal of Computer Vision*, *130*(11), 2606–2622. http://dx.doi.org/10.1007/s11263-022-01656-y.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778). URL https://api.semanticscholar.org/CorpusID:206594692.

Huang, H., Wang, C., Yu, P. S., & Wang, C.-D. (2019). Generative dual adversarial network for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 801–810).

Jayaraman, D., & Grauman, K. (2014). Zero-shot recognition with unreliable attributes. In *Advances in neural information processing systems*: *vol. 27*.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. CoRR. arXiv:1312.6114.

Kong, X., Gao, Z., Li, X., Hong, M., Liu, J., Wang, C., et al. (2022). En-compactness: Self-distillation embedding & contrastive generation for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9306–9315).

Liu, M., Li, F., Zhang, C., Wei, Y., Bai, H., & Zhao, Y. (2023). Progressive semantic-visual mutual adaption for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15337–15346).

Liu, S., & Ozay, M. (2023). Task guided representation learning using compositional models for zero-shot domain adaptation. *Neural Networks*.

Mandal, D., Narayan, S., Dwivedi, S. K., Gupta, V., Ahmed, S., Khan, F. S., et al. (2019). Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9985–9993).

Narayan, S., Gupta, A., Khan, F. S., Snoek, C. G., & Shao, L. (2020). Latent embedding feedback and discriminative features for zero-shot classification. In *European conference on computer vision* (pp. 479–495). Springer.

Nilsback, M.-E., & Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing* (pp. 722–729).

Patterson, G., & Hays, J. (2012). SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 2751–2758).

Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., et al. (2023). A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(4), 4051–4070. http://dx.doi.org/10.1109/TPAMI.2022.3191696.

Rahman, S., Khan, S., & Porikli, F. (2018). A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing*, *27*(11), 5652–5667. http://dx.doi.org/10.1109/TIP.2018.2861573.

Reddy, Y., Viswanath, P., & Reddy, B. E. (2018). Semi-supervised learning: A brief review. *International Journal of Engineering & Technology*, *7*(1.8), 81.

Rohrbach, M., Ebert, S., & Schiele, B. (2013). Transfer learning in a transductive setting. In *Advances in neural information processing systems*: *vol. 26*.

Romera-Paredes, B., & Torr, P. (2015). An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning* (pp. 2152–2161). PMLR.

Schönfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., & Akata, Z. (2019). Generalized zero- and few-shot learning via aligned variational autoencoders. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 8239–8247).

Su, H., Li, J., Chen, Z., Zhu, L., & Lu, K. (2022). Distinguishing unseen from seen for generalized zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7885–7894).

Sun, X., Gu, J., & Sun, H. (2021). Research progress of zero-shot learning. *Applied Intelligence*, *51*(6), 3600–3614. http://dx.doi.org/10.1007/s10489-020-02075-7.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In *Artificial neural networks and machine learning–ICANN 2018: 27th international conference on artificial neural networks* (pp. 270–279).

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., & Van Gool, L. (2020). Scan: Learning to classify images without labels. In *European conference on computer vision* (pp. 268–285). Springer.

Wang, Q., & Breckon, T. P. (2023). Generalized zero-shot domain adaptation via coupled conditional variational autoencoders. *Neural Networks*, *163*, 40–52.

Wang, H., Nie, F., & Huang, H. (2013). Robust and discriminative self-taught learning. In *International conference on machine learning* (pp. 298–306). PMLR.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, *53*(3), 1–34. http://dx.doi.org/10.1145/3386252.

Wang, W., Zheng, V. W., Yu, H., & Miao, C. (2019). A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*(2), 1–37. http://dx.doi.org/10.1145/3293318.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S. J., et al. (2010). Caltech-UCSD birds 200.

Xian, Y., Lampert, C. H., Schiele, B., & Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(9), 2251–2265. http://dx.doi.org/10.1109/TPAMI.2018.2857768.

Xian, Y., Lorenz, T., Schiele, B., & Akata, Z. (2018). Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5542–5551).

Xian, Y., Sharma, S., Schiele, B., & Akata, Z. (2019). f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10275–10284).

Xu, W., Xian, Y., Wang, J., Schiele, B., & Akata, Z. (2022). VGSE: Visually-grounded semantic embeddings for zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9316–9325).

Yang, F.-E., Lee, Y.-H., Lin, C.-C., & Wang, Y.-C. F. (2023). Semantics-guided intra-category knowledge transfer for generalized zero-shot learning. *International Journal of Computer Vision*, *131*(6), 1331–1345. http://dx.doi.org/10.1007/s11263-023-01767-0.

Ye, M., & Guo, Y. (2017). Zero-shot classification with discriminative semantic representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7140–7148).

Ye, H.-J., Hu, H., & Zhan, D.-C. (2021). Learning adaptive classifiers synthesis for generalized few-shot learning. *International Journal of Computer Vision*, *129*, 1930–1953. http://dx.doi.org/10.1007/s11263-020-01381-4.

Yu, H., & Lee, B. (2019). Zero-shot learning via simultaneous generating and learning. In *Advances in neural information processing systems*: *vol. 32*.

Yue, Z., Wang, T., Sun, Q., Hua, X.-S., & Zhang, H. (2021). Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15404–15414).

Yun, Y., Wang, S., Hou, M., & Gao, Q. (2022). Attributes learning network for generalized zero-shot learning. *Neural Networks*, *150*, 112–118.

Zhang, C., & Peng, Y. (2018). Visual data synthesis via GAN for zero-shot video classification. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 1128–1134).

Zhao, X., Shen, Y., Wang, S., & Zhang, H. (2023). Generating diverse augmented attributes for generalized zero shot learning. *Pattern Recognition Letters*, *166*, 126–133. http://dx.doi.org/10.1016/j.patrec.2023.01.005.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*.