On the Transferability of Visual Features in Generalized Zero-Shot Learning

Paola Cascante-Bonilla† Leonid Karlinsky† James Seale Smith§ Yanjun Qi² Vicente Ordonez†
†Rice University ‡MIT-IBM Watson AI Lab §Georgia Institute of Technology §Georgia Institute of Technology
{pc5l, vicenteor}@rice.com, leonidka@ibm.com, jamessealesmith@gatech.edu, yq2h@virginia.edu

Abstract

Generalized Zero-Shot Learning (GZSL) aims to train a classifier that can generalize to unseen classes, using a set of attributes as auxiliary information, and the visual features extracted from a pre-trained convolutional neural network. While recent GZSL methods have explored various techniques to leverage the capacity of these features, there has been an extensive growth of representation learning techniques that remain under-explored. In this work, we investigate the utility of different GZSL methods when using different feature extractors, and examine how these models’ pre-training objectives, datasets, and architecture design affect their feature representation ability. Our results indicate that 1) methods using generative components for GZSL provide more advantages when using recent feature extractors; 2) feature extractors pre-trained using self-supervised learning objectives combined with cross-entropy and knowledge distillation provide better feature representations, increasing up to 15% performance when used with recent GZSL techniques; 3) specific feature extractors pre-trained with larger datasets do not necessarily boost the performance of GZSL methods. In addition, we investigate how GZSL methods fare against CLIP, a more recent multi-modal pre-trained model with strong zero-shot performance. We found that GZSL tasks still benefit from generative-based GZSL methods along with CLIP’s internet-scale pre-training to achieve state-of-the-art performance in fine-grained datasets. We release a modular framework for analyzing representation learning issues in GZSL here: https://github.com/uvavision/TV-GZSL.

1. Introduction

Deep learning models have achieved remarkable accuracy in many computer vision classification tasks when labeled data is available, and the data distribution is consistent during training and test time [19, 35, 43]. It is now possible to train image classifiers that can distinguish with high accuracy thousands of image categories [37]. However, in order to enable a model to recognize novel categories, it is still necessary to collect a dataset with representative human-labeled examples. For this reason, literature has proposed zero-shot learning to help a model recognize novel unseen categories without needing any images of the new category. Zero-shot learning relies on auxiliary information such as textual descriptions or category attributes [14, 15, 28]. The general idea is to leverage the auxiliary information to transfer visual knowledge from images in seen categories to a set of images from unseen categories. Once this function is learned, it can be used to categorize unseen samples into novel classes.

Early work on zero-shot learning focused on obtaining a good accuracy just on a set of unseen categories at inference time. In a more challenging scenario known as Generalized Zero-shot Learning (GZSL), both seen and unseen categories are considered at test time [6, 32]. This setup is more realistic and has been adopted in all the recent works in zero-shot learning; therefore, we focus exclusively on this setting.

First-generation methods are coined as embedding-based techniques, which focus on learning a function to align the
seen images and unseen attributes and further measure the similarity between the mapped and predicted representations of the data samples in the embedding space [3, 36, 51, 52]. Further GZSL methods have shown significant improvement over this early work by teaching a model to generate the visual features of the unseen classes based on the visual features of the seen classes and the semantic representations of both seen and unseen categories. More recent works have instead explored rich image feature representations and their ability to provide enough information for a mapping function to generalize to new classes [9, 46]. These methods propose to learn discriminative representations from image features through disentanglement over feature groups by factorizing the useful dimensions to avoid bias towards the seen categories when trying to learn an attribute-visual alignment.

While disentanglement-based methods have significantly improved over prior work, the representation capabilities of image features have mostly been tested under a ResNet101 model [22] pre-trained on Imagenet [29, 50]. In this work, we propose to explore different architectures trained with different objectives to be used as feature extractors. Given the wide availability of pre-trained model parameters [2, 10, 17], we perform a large-scale analysis to assess the impact of modern visual features backbones, our results are summarized in Figure 1. In our analysis, we consider visual features extracted from both uni-modal and multi-modal architectures pre-trained on standard settings, i.e., models pre-trained with ImageNet-1K; to models pretrained with even larger amounts of data, i.e., models pre-trained with ImageNet-21K and models pre-trained with 400 million Image/Text pairs.

Our results show that ResNet101 and similar feature extractors may not provide enough information, given the nature of the backbone architecture limitations and the training objective. Moreover, when examining feature extractors pre-trained with larger datasets, one would assume that the features extracted would contain more similarities to the data present in the GSZL test splits, causing all methods to perform better due to data leakage. However, we found that using features extracted from large-scale pre-trained uni-modal networks does not significantly impact the zero-shot performance. Our experiments also reveal that with feature representations extracted from Visual Transformer [12] based architectures, no semantic disentanglement module is necessary to achieve state-of-the-art performance, and generative-based methods are superior in all benchmarks by a large margin (≈ 23%) when using features extracted from Transformer based architectures [12] trained with a contrastive objective. Furthermore, given the capabilities of new models [23, 33, 41] to generalize to new tasks, we investigate how CLIP [33] performs against GZSL methods and reveal that generative-based GZSL techniques are still necessary to achieve state-of-the-art results on fine-grained datasets.

Our primary contributions can be summarized as follows:

- A large-scale study of different methods and features extracted from a diverse set of architectures and training approaches, applied to state-of-the-art methods from different GZSL families (i.e., embedding based [3, 16, 36], generative based [20, 30, 38], semantic disentanglement based [8, 9, 46]).
- A library containing the revisited GSZL methods we chose to explore, allowing a unified codebase for reproducibility and further analysis based on the findings shown in this work. Texts generated to finetune the models based on the attribute vectors provided in each dataset, along with the weights and the extracted features we use in our experiments.
- Updates and key insights which we hope will reshape the Generalized Zero-Shot Learning research track in favor of leveraging richer feature representations.

We expect our work to motivate further feature representation explorations to the GZSL task in a more realistic, practical, and challenging scenario, given the recent advances with large pre-trained models. All the resources used, including GPU information and computing infrastructure are detailed in the Appendix, along with hyperparameter selections and data splits we use for all our experiments.

2. Related Work

The goal of GZSL is to classify images from both seen and unseen categories by transferring knowledge from seen to unseen classes using a set of attributes as auxiliary information. Since every attribute vector entry represents a class description, the assumption is that the classes with similar descriptions contain a similar attribute vector in the semantic space. Therefore, the general idea is to learn a function that allows this mapping between modalities. The general idea is to align visual features from seen classes with their corresponding attribute vectors. Current methods can be broadly categorized into:

- Embedding-based methods, which aim to learn a mapping function or a projection between visual features and their attributes or descriptions [3, 16, 36]. This mapping function is used to project image features into a semantic space so that it is possible to classify seen and unseen classes by estimating how close these features are to a class embedding vector [7, 27, 34, 40].

- Generative-based methods, which synthesize an unlimited number of visual features using the auxiliary information from the unseen classes, and compensate for the imbalance classification problem that poses the GZSL task [20, 30, 38]. A generative model is a probabilistic model that is representative of the conditional probability of the observable input $X$, given a target $y$ [18, 24]. Recent advances in generative modeling have gained a significant amount of attention.
In the GZSL setting, generative models are leveraged to learn to generate visual features or images for the unseen classes [25, 42]. This is achieved using samples from the seen classes and semantic representations of both seen and unseen classes. Generative-based methods convert the GZSL problem into a supervised learning problem by generating samples for both seen and unseen classes.

Semantic disentanglement-based methods, which aim to factorize the useful dimensions of a given visual feature to learn the attribute-visual alignment. The assumption is that the image features extracted from a ResNet101 that was pretrained on ImageNet, broadly used in all traditional GSZL datasets [31, 49, 50], are not ideal for the zero-shot learning task [8, 9, 46]. Since these features are not correlated with respect to the specific attributes that describe the image parts/composition, not all the dimensions of these features are semantically related to the given attributes; thus, it is necessary to factorize or disentangle the useful dimensions to avoid bias when trying to learn the attribute-visual alignment. In this context, entanglement refers to the property of not having independence among attributes of one representation. In an entangled representation, all the factors of variation are mixed, and there is no explicit separation that represents the important characteristics in the images [4]. On the other hand, given an image dataset of birds such as the CUB dataset [49], a disentangled representation may consist of separate dimensions for wing color, breast color, bill shape, tail pattern, crown color, wing pattern, etc [13].

3. Generalized Zero-Shot Learning

In the GZSL setting, we define $S$ as the set of seen classes within $N_s$ categories and $Y_s$ as their corresponding labels. We also define $U$ as the set of unseen classes with $N_u$ categories and $Y_u$ as their corresponding labels. Here, $S$ and $U$ have no intersection. Each set consists of image-features $x$, class labels $y$ available during training and class-embeddings $c(y)$. Thus, $S = \{x_s^i, y_s^i\}_{i=1}^{N_s}$ and $U = \{x_u^i, y_u^i\}_{i=1}^{N_u}$. Additionally, we have access to a set of semantic descriptions of the seen and unseen classes $A = \{a_i\}_{i=1}^{N_s+N_u}$, which are typically class-embeddings vectors of hand-annotated attributes. The image features are typically extracted from a feature backbone (i.e., ResNet101 pretrained on ImageNet-1K). When training, we use $A$ along with the visual features and labels from the seen set (i.e., $\{x_s^i, y_s^i\}_{i=1}^{N_s}$) and only the labels from the unseen set (i.e., $\{y_u^i\}_{i=1}^{N_u}$). Finally, the test set contains image-features from both $S$ and $U$, and their corresponding labels. This section presents and describes all the methods we include in this study grouped by their corresponding family of methods. These can be characterized as: embedding-based, generative-based and disentanglement-based. We also present all the datasets we use in our study. Additional training and computational details are included in the Appendix.

3.1. Embedding-based Methods

DeViSE: A Deep Visual-Semantic Embedding Model [16] (2013) propose to learn a linear mapping between the image and the semantic space and introduce the use of a ranking loss instead of an $L_2$ loss. This avoids making the vectors become closer to one another without taking into account the incorrect labels that are closer to the target image.

An embarrassingly simple approach to zero-shot learning (ESZSL) [36] (2015) use a linear model to create relationships between features, attributes and classes. This linear model has two layers, the first layer helps in defining the relationship between features and attributes, and the second layer deals with modeling the relationship between attributes and classes where the prescribed attribute signatures are fixed. This method uses a square loss to learn the bilinear compatibility between attributes and their corresponding classes.

Label-Embedding for Image Classification present an Attribute Label Embedding approach (ALE) [3] (2016), they propose to embed each class in the space of attribute vectors, and further propose to learn a bilinear compatibility function between the image and a label embedding using a ranking objective function.

3.2. Generative-based Methods

Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders (CADA-VAE) [38] (2019) use two variational autoencoders (VAEs [24]) to align the visual and semantic features by learning a shared latent space between both modalities. Then, these VAEs are used to synthesize a large number of seen and unseen features, which are then used to train a classifier.

Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification (tfVAEGAN) [30] (2020) propose to use a feedback module in a model that combines a variational autoencoder (VAE [24]) and a generative adversarial network (GAN [18]) to modulate the latent representation of the generator. They propose to enforce semantic consistency by introducing a feedback loop from the semantic embedding decoder. They propose to use both synthesized features and latent embeddings during classification.

More recently, hybrid methods such as Contrastive Embedding for Generalized Zero-Shot Learning (CE) [20] (2021) have emerged. This method proposes to integrate the generation model with the embedding model to map the real and the synthetic samples produced by the generation model into an embedding space. To do this, they leverage a contrastive loss that learns to discriminate between one positive sample and a large number of negative samples from different semantic descriptor classes. They claim that the original visual feature space is suboptimal for GZSL classification since it lacks discriminative information, which they aim to learn using the contrastive objective.
3.3. Semantic Disentanglement-based Methods

FREE: Feature Refinement for Generalized Zero-Shot Learning [8] (2021) use a feature refinement module to jointly map the semantic and visual modalities, and refine the visual features of seen and unseen class samples – dropping unnecessary information from the visual features. They use the class label supervision and a semantic cycle-consistency constraint to guide the proposed module to learn relevant feature representations that are also semantically relevant with respect to their corresponding classes.

Semantics Disentangling for Generalized Zero-Shot Learning (SDGZSL) [9] (2021) introduce a total correlation penalty that is applied to the visual features of unseen classes that were generated by a conditional VAE [24]. This work uses a Relation Network [44] to correlate the factorized estimated features – the semantic-consistent and the semantic-unrelated latent features.

3.4. GZSL Datasets

We use three datasets. Caltech-UCSD Birds-200-2011 (CUB) [49], a fine-grained dataset with 11,788 images from 200 different types of birds annotated corresponding to 150 seen and 50 unseen classes, with 312 attributes. SUN Attribute (SUN) [31], a fine-grained dataset with 14,340 images from 717 types of scenes corresponding to 645 seen and 72 unseen classes, annotated with 102 attributes. Animals with Attributes2 (AWA2) [50], a dataset with 37,322 images from 50 animal classes corresponding to 40 seen and 10 unseen classes, annotated with 85 attributes. We report the seen and unseen accuracy, and their harmonic mean.

4. Pre-Trained Image Feature Extractors

We begin our analysis by extracting the corresponding image features for all our samples using a diverse set of popular models pre-trained on Imagenet-1k and Imagenet-21k – whose weights are publicly available. Prior work [50] mention that using GoogLeNet features is not as effective as using Resnet101 features; thus, we want to further investigate: does feature extractor size matters for GZSL? This work aims to evaluate if pre-training on a bigger set impacts the outcomes, and if similarly robust features extracted from different model architectures – other than Resnet101 – makes a significant difference. For the backbones that were pre-trained using Imagenet-1k, we make sure that we use the same splits proposed in [50] so that pre-trained features do not violate the zero-shot principle. We also want to measure the impact of using visual features extracted from networks that were pre-trained with data present in the test set of our settings. Thus, we investigate how the selected GZSL methods leverage the information extracted from larger backbones pre-trained with bigger and more diverse datasets (i.e., Imagenet-21k).

4.1. Unimodal Feature Backbones

Overall, the pool of networks in this study have been trained using two learning objectives: Supervised Learning (SL), which aims to learn a function that maps an input to a known output, and is typically trained using a cross-entropy loss (e.g., Resnet101 [22]) or distillation (e.g., DeiT [47]). On the other hand, Self-Supervised Learning (SSL), aims to learn a function that maps an input to a unknown output; SSL can be accomplished using a contrastive loss such as InfoNCE (e.g., MoCo [21]) or adding distillation along with a similarity metric measured with a cross-entropy loss applied over the features of two different random transformations of an input image (e.g., DINO [5]). We perform experiments on three broad types of unimodal architectures trained on images only (Imagenet-1k or Imagenet-21k): Convolutional Neural Networks (CNN) [26] trained using the images as a whole under SL or SSL; Vision Transformers (ViT) [12] which takes an image, transforms it in a sequence of image patches and is trained using either SL or SSL; and Multi-Layer Perceptron Mixers (MLP) [45] which also exploit image patches and are trained with supervision.

We chose the most performant models and fine-tuned them using only the training samples from the seen classes. Since the visual features are extracted from a diverse set of architectures trained in the wild, we focus on the inductive setting, where there is no access to the unlabeled visual data from the unseen classes. In this way, we mitigate any bias reinforced by additional training of the visual representations. This practice has been followed to achieve better results in prior literature (which only uses Resnet101 features); however, there are no available reports for all the methods using these features; thus, we run all methods and report our findings in Section 5.

4.2. Multimodal Feature Backbones

In addition, we study CLIP [33], a multi-modal model with a visual encoder and a textual encoder trained with 400 million image/text pairs in a contrastive way. Traditionally, all GZSL methods disregard the attribute values as they are given for granted, and no additional analysis is performed. We instead take a closer look at these semantic features and their corresponding attributes (e.g., color, shape, type of animal, type of place, etc) and use them to fine-tune several pre-trained CLIP models.

We perform three experiments with CLIP: (A) We directly evaluate the model using the images and class names without any further pre-training or post-processing by directly looking at the ranking the model yields for each seen and unseen samples, (B) We fine-tune CLIP using the class names, the attribute values and a combination of class names + attribute values, and evaluate its performance, (C) We extract the visual features from the CNN and ViT based visual backbones available in their public repository, and use the features ex-
5. Results using Unimodal Feature Extractors

We evaluate all GZSL methods trained with feature extractors pre-trained using ImageNet-1k. In Figure 2, we show the Harmonic Mean performance of different methods when using a specific model to extract the features of image samples. Surprisingly, CADA-VAE [38] and tVAEGAN [30] consistently outperform all methods. While current disentanglement methods show significant improvements when using features from transformer-based architectures, they are outperformed by the generative-based methods. We can also observe that DINO [5] provides better feature representations for all methods across all datasets.

We then evaluate the impact of using feature extractors with the same architecture type but trained with different learning objectives. In Figure 3, we show the Harmonic Mean performance of different GZSL methods when using a Resnet model architecture pre-trained on ImageNet-1k as the image feature extractor. Surprisingly, the features extracted from DINO [5] increase the Harmonic Mean performance up to 15% in both fine-grained datasets (i.e., CUB and SUN datasets). More surprisingly, the feature vectors extracted from Moco perform worse than traditional supervised learning models trained with a cross-entropy objective function. Moco’s training objective is formulated by the InfoNCE loss, which encourages the model to maximize the Mutual Information (MI) between N random samples containing one positive sample, and minimize the Mutual Information between the anchor sample and N – 1 negative samples. On the other hand, in DINO, a teacher and a student model are trained by feeding two different random transformations of an input image to each network; the objective is to maximize the similarity between both outputs, which is encouraged and measured with a cross-entropy loss. Thus, the generalization capabilities shown with DINO support prior observations when using its image features in classification tasks with respect to other self-supervised techniques [5].
from bigger architectures seem consistently better.

We show more detailed results on CUB, SUN and AWA2 using GZSL methods grouped in their corresponding families in the following subsections. Please refer to the Appendix to check the full list of numerical results for all unimodal backbones and GZSL methods.

### 5.1. Results of Embedding-based Methods

The ViT$_L$-based features pre-trained on ImageNet-21k seem to be the best for all the methods using ALE. However, for the AWA2 dataset, all methods perform better using the features extracted from a network pre-trained using ImageNet-1k. For CUB and SUN datasets, the performance gap against the features extracted from a network trained using ImageNet-1k and ImageNet-21k is not significant for all methods. More detailed results are available in Tables 6, 7 and 8 from Section D in the Appendix.

### 5.2. Results of Generative-based Methods

The most performant visual features are extracted from a ViT$_L$-based model to extract the features of image samples from all datasets. [ViT-L - ImageNet-1k] and [ViT-L - ImageNet-21k] indicates the dataset the model was pretrained with. Best viewed in color.

### 5.3. Results of Disentanglement-based Methods

The most performant visual features are extracted from a ViT$_L$-based pre-trained on ImageNet-1k using the SDGZSL method. Here, the features extracted from architectures pre-trained using ImageNet-21k perform worse than the ones pre-trained using ImageNet-1k, except for the CUB dataset. More detailed results are available in Tables 9, 10 and 11 from Section D in the Appendix.

### 6. Results using CLIP as a Multimodal Feature Extractor

**Direct evaluation of CLIP** using the images and class names without any further pre-training or post-processing: we use different template captions to generate the textual seen classes, using the CADA-VAE method. More interestingly, the features from a ViT$_L$-based pretrained on ImageNet-1k seem competitive with the features from a ViT$_L$-based pretrained on ImageNet-21k for the CE and tVAEGAN methods respectively. More detailed results are available in Tables 12, 13 and 14 from Section D in the Appendix.

### Table 1. Results of using publicly available pre-trained CLIP [33] models with different backbones to evaluate three standard GSZL datasets.

| Backbone | CUB | SUN | AWA2 |
|----------|-----|-----|------|
|          | Seen | Novel | Harm. | Seen | Novel | Harm. | Seen | Novel | Harm. |
| RN50     | 45.90 | 45.44 | 45.67 | 44.61 | 48.96 | 46.68 | 87.86 | 82.48 | 85.08 |
| RN101    | 48.86 | 49.44 | 49.15 | 45.16 | 49.24 | 47.11 | 88.66 | 84.79 | 86.67 |
| RN50x4   | 51.89 | 55.27 | 53.53 | 48.53 | 50.56 | 49.52 | 92.09 | 86.52 | 89.22 |
| RN50x16  | 56.82 | 55.19 | 55.99 | 49.07 | 54.44 | 51.62 | 94.36 | 89.11 | 91.65 |
| RN50x64  | 63.81 | 57.19 | 60.32 | 55.28 | 51.59 | 53.37 | 95.11 | 90.11 | 92.54 |
| ViTB32   | 51.35 | 49.65 | 50.49 | 48.26 | 50.97 | 49.58 | 90.99 | 85.69 | 88.26 |
| ViTB32†  | 61.46 | 58.09 | 59.73 | 50.66 | 53.75 | 52.16 | 87.84 | 82.12 | 84.88 |
| ViTB16   | 55.76 | 56.23 | 55.99 | 50.97 | 56.11 | 53.42 | 93.68 | 87.19 | 90.32 |
| ViTL14†  | 62.62 | 63.19 | 62.90 | 55.97 | 58.06 | 56.99 | 95.80 | 89.39 | 92.48 |
|          | 64.45 | 62.69 | 63.56 | 57.79 | 62.64 | 60.12 | 96.06 | 89.91 | 92.88 |

† indicate we used a set of captions similar to the proposed by OpenAI to test on Imagenet [11]. † indicates the model used was trained on the Laion400M [39] dataset.
Figure 5. Fine-tuning CLIP with seen samples. Results of fine-tuning a CLIP model using the images and different text descriptions, including the class name, the attribute text, and combining both text captions. We show accuracy variations after 110k training iterations for CUB [49], SUN [31] and AWA2 [50], where BS and BS indicate the base performance of seen and unseen classes without fine-tuning. CS refers to the seen classes, and CV refers to the unseen (novel) classes after fine-tuning.

descriptions (e.g., "An image of a [class name"]”) and chose to evaluate the best-performing model with the template captions proposed by OpenAI to test on ImageNet [1]. We also evaluate a CLIP model trained with the publicly available LAION-400M dataset [39] in the publicly available visual transformer backbone (i.e., ViTB32). We show results in Table 1. Overall, we expected CLIP to perform well in all the selected datasets, even with CUB, whose class names are particularly specific; similar results on this dataset have been reported in concurrent work [48]. Moreover, our results show that generative-based models work on par and even outperform CLIP when using the Resnet101 fine-tuned features, indicating that there might be room for improvement.

Evaluation of CLIP performance after fine-tuning using the class names, the attribute values and a combination of class names & attribute values: Figure 5 shows the accuracy variations after fine-tuning CLIP for 110k iterations using different types of text prompts, using only the seen training set for each dataset. Interestingly, we observed that fine-tune enhances both the seen and unseen accuracy.

| Data Set | CUB | SUN | AWA2 |
|----------|-----|-----|------|
| Backbone | R50 | ViT | ViT |
| R5064   | 82.09 | 77.67 | 80.39 |
| ViT14   | 80.39 | 75.91 | 80.73 |
| ViT14   | 82.82 | 81.96 | 80.40 |
| ViT14   | 82.96 | 75.05 | 80.38 |

Table 2. Results of Generative and Disentanglement Based Methods for the CUB, SUN and AWA2 datasets using different features extracted from different size and architecture of the visual head from diverse CLIP models (i.e., Resnet50 (R50) and Vision Transformer (ViT)). The three bottom rows per section correspond to fine-tuned features using sentences with: † the class names, ‡ the attributes, and § both class names and attributes. The bold numbers correspond to the highest scores per column, the underline numbers correspond to the to the highest scores using features not fine-tuned, and the shaded rows correspond to the most performant image feature per method over all.
Evaluation of the GZSL methods using the visual features extracted from the visual encoder of CLIP to train all GZSL methods: We show in Table 2 the effect of using CLIP features and which fine-tuned model performs the best. Unsurprisingly, using both class name and class attributes outperforms other backbones. Similarly to uni-modal backbones, the generative-based models (e.g., CADA-VAE and tVAEGAN) outperform other methods on all datasets. More surprisingly, both methods also outperform the CLIP results in CUB and SUN, but CLIP alone outperforms all methods in the AWA2 dataset. We run experiments using all eight visual encoders available from CLIP model, but show the most relevant results in Table 2. Please refer to the Appendix to check the full list of results.

7. Fine-tuning Feature Extractors

We observe that in the generative and disentanglement-based methods, there seems to be a trade-off in the multi-modal latent space, where some features from the seen set are distilled to the projected features of the unseen set. Typically, these methods augment the training set and convert the problem into a classification task, thus, the final seen accuracy is penalized by the classifier. For this reason, we also investigate how much these models penalize the final accuracy of the seen sets by measuring the difference of a classifier trained only using the seen set versus the best-seen accuracy achieved among all GZSL generative and disentanglement-based methods. The results are shown in Figure 6. We observe that while fine-tuning a Resnet101 model increment the seen accuracy of both the classifier and the GZSL method, it does not increase the seen accuracy of recent models such as vision transformers. We also observe that the seen accuracy of the GZSL method does not improve substantially when using ViT features, and fine-tuning on AWA2 hurts the best seen accuracy on the GZSL result while not improving the classifier seen accuracy.

8. Conclusion

In this paper, we provide strong empirical evidence that indicates that:

- Using Transformer based architectures provides superior feature representation capabilities while not violating the zero-shot principle of being pre-trained on unseen classes.
- The feature representations extracted from unimodal architectures that were pre-trained on larger datasets (e.g., ImageNet-21k) do not necessarily boost GZSL performance.
- Using Convolutional based architectures pre-trained without labels, using contrastive learning and self-distillation, provides better feature representations for GZSL than models trained using supervised learning, with known labels and cross-entropy loss alone.
- Fine-tuning does not significantly impact the performance on Transformer based unimodal backbones but may boost the performance on multimodal backbones.
- Multimodal architectures trained on internet-scale large data (CLIP) still benefit from generative based GZSL methods to achieve state-of-the-art performance in CUB and SUN, which are fine-grained datasets. This may indicate that feature representations from CLIP are more suitable for GZSL when there is less inter-class correlation among data samples.
- Fine-tuning a CLIP model using prompts including the class names and attributes from the seen categories also boosts the ranking performance of the unseen classes.

In summary, our work provides an update on GZSL methods in the era of large-scale multi-modal pre-training, and re-evaluates in this context the progress that has been made so far in this area. We release a well-documented codebase that both replicates our findings and provides a modular framework for further feature representation explorations to the GZSL task with recent large pre-trained models.

Acknowledgements This material is based upon work sup-
References

[1] Clip prompt engineering for imagenet. https://github.com/openai/CLIP/blob/main/notebooks/Prompt_Engineering_for_ImageNet.ipynb. Accessed: 2022-06-01. 7

[2] Pytorch image models (timm). https://github.com/rwightman/pytorch-image-models. Accessed: 2022-06-01. 2

[3] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38:1425–1438, 2016. 2, 3, 11, 12

[4] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35:1798–1828, 2013. 3

[5] Matthi1e1 Caron, Hugo Touvron, Ishan Misra, Hervé’1egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 9630–9640, 2021. 4, 5

[6] 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. ArXiv, abs/1605.04253, 2016. 1

[7] Long Chen, Hanwang Zhang, Jun Xiao, W. Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1043–1052, 2018. 2

[8] Shiming Chen, Wenjie Wang, Beihao Xia, Qimiu Peng, Xinge You, Feng Zheng, and Ling Shao. Free: Feature refinement for generalized zero-shot learning. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 122–131, 2021. 2, 3, 4, 11, 12

[9] Zhi Chen, Yadan Luo, Ruihong Qiu, Sen Wang, Zi-Yu Huang, Jingjing Li, and Zheng Zhang. Semantics disentangling for generalized zero-shot learning. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 8692–8700, 2021. 2, 3, 4, 11, 12

[10] Francesco Croce, Maksym Andriushchenko, Vikash Sainath, Edoardo DeBenedetti, Nicolas Flammarion, Ming Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. ArXiv, abs/2010.09670, 2021. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 6, 15

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, abs/2010.11929, 2021. 2, 4, 8

[13] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In ICLR, 2018. 3

[14] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In Proceedings of the IEEE International Conference on Computer Vision, pages 2584–2591, 2013. 1

[15] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 1778–1785, 2009. 1

[16] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In NIPS, 2013. 2, 3, 11, 12

[17] Paul Gavrikov and Janis Keuper. Cnn filter db: An empirical investigation of trained convolutional filters. ArXiv, abs/2203.15331, 2022. 2

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, 2014. 2, 3

[19] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. Intelligent Systems, IEEE, 24:8–12, 05 2009. 1

[20] Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2371–2381, 2021. 2, 3, 11, 12

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9726–9735, 2020. 4, 5

[22] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–777, 2016. 2, 4, 8

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In ICML, 2021. 2

[24] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. CoRR, abs/1312.6114, 2014. 2, 3, 4

[25] Xia Kong, Zuodong Gao, Xiaofan Li, Ming Hong, Jun Liu, Chengjie Wang, Yuan Xie, and Yanyun Qu. 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 (CVPR), pages 9306–9315, June 2022. 3

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60:84 – 90, 2012. 4

[27] Gukyeong Kwon and Ghassan AlRegib. A gating model for bias calibration in generalized zero-shot learning. IEEE Transactions on Image Processing, pages 1–1, 2022. 2
[28] Christoph Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

[29] Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Generalized zero-shot learning with deep calibration network. In *NeurIPS*, 2018.

[30] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees G. M. Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. *ArXiv*, abs/2003.07833, 2020.

[31] Genevieve Paterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 2751–2758, 2012.

[32] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, and Xizhao Wang. A review of generalized zero-shot learning methods. *ArXiv*, abs/2011.08641, 2020.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[34] Shafin Rahman, Salman Hameed Khan, and Fatih Murat Porikli. A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing*, 27:5652–5667, 2018.

[35] Shaqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[36] Bernardino Romera-Paredes and Philip H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.

[37] Olga Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Zhiheng Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[38] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8239–8247, 2019.

[39] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarchzyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021.

[40] Yuming Shen, Jie Qin, and Lei Huang. Invertible zero-shot recognition flows. *ArXiv*, abs/2007.04873, 2020.

[41] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15617–15629, 2022.

[42] Hongzu Su, Jingjing Li, Zhi Chen, Lei Zhu, and Ke Lu. Distinguishing unseen from seen for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7885–7894, June 2022.

[43] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *CoRR*, abs/1707.02968, 2017.

[44] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[45] Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021.

[46] Bin Tong, Chao Wang, Martin Klinkigt, Yoshiyuki Kobayashi, and Yuuichi Nonaka. Hierarchical disentanglement of discriminative latent features for zero-shot learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11459–11468, 2019.

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[48] Felix Vogel, Nina Shvetsova, Leonid Karlinsky, and Hilde Kuehne. Vl-taboo: An analysis of attribute-based zero-shot capabilities of vision-language models. *ArXiv*, abs/2209.06103, 2022.

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[50] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2251–2265, 2019.

[51] Lei Zhang, Peng Wang, Lingqiao Liu, Chunhua Shen, Wei Wei, Yanning Zhang, and Anton van den Hengel. Towards effective deep embedding for zero-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30:2843–2852, 2020.

[52] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3010–3019, 2017.
A. Appendix

In Section B we show implementation details with hyperparameter selections for all GZSL methods, and detail the computing infrastructure we use while conducting all of our experiments. Then we show in fine-tuning information for the multi-modal backbone in Section C. We also show extended results for all of our experiments using all features from different backbones and methods in Section D. Lastly, we discuss about some ethical considerations and the importance of Generalized Zero-Shot Learning in Section E.

B. Implementation Details

We follow the original code and recommended hyperparameters from the existing implementations provided by their corresponding authors for all the GZSL methods in this study. In Table 3 we detail all values for all methods.

B.1. Computing Infrastructure

We performed all the GZSL methods experiments using features extracted from unimodal backbones in 3 servers with 4 NVIDIA TITAN RTX GPUs each. We performed all the GZSL methods experiments using features extracted from multimodal backbones in a single server with 8 NVIDIA A40 GPUs. Feature extraction and fine-tuning were done in a single server with 8 NVIDIA A40 GPUs. All experiments were run on a single GPU.

C. Prompt Engineering for CLIP Fine-Tuning

Prompt for all classes adding class name in the sentence: 'This is a photo of a {}', shown in Table 4.

We also finetune taking into account the attribute labels and scores per dataset, shown in Table 5.

D. Full experimental results for our large scale analysis

In this section we show all the results obtained using a large variety of visual backbones from different architecture types. We showcase the performance of all methods, grouped by their corresponding GZSL families and datasets as follows:

- Embedding-based methods DEVISE [16], ESZSL [36] and ALE [3]:
  - CUB results in Table 6.
  - SUN results in table 7.
  - AWA2 results in table 8.
- Generative-based methods TF-VAEGAN [30], CADA-VAE [38], CE [20]:
  - CUB results in Table 9.
  - SUN results in table 10.
  - AWA2 results in table 11.
- Semantic disentanglement-based methods SDGZSL [9], FREE [8]:
  - CUB results in Table 12.
  - SUN results in table 13.
  - AWA2 results in table 14.

Finally, in Table 15, we show full results of generative and disentanglement based methods for the CUB, SUN and AWA2 datasets when using different features extracted from different size and architectures of the visual encoder from all available OpenAI CLIP [33] models.

E. Ethical Considerations

Machine learning models still require collecting large amounts of annotated data. In the case of fine-grained recognition, these annotations often require specialized human knowledge. Zero-shot learning offers a way for bypassing the need to collect extensive amounts of data for training models for new classes of objects. We show that large-scale pre-trained models along with Generalized Zero-Shot Learning methods can obtain results that are competitive with the specialized knowledge from experts on classes that a trained model has never seen. We hope that the key insights and analysis provided in this paper will be useful in expanding and leveraging zero-shot research along with current progress in multi-modal learning. Allowing for the creation of models that do not depend on large amounts of data could be useful for practitioners without access to large scale resources or in domains where data is scarce such as the medical domain.

1https://github.com/openai/CLIP
| Parameter | DEVISE | ESZSL | ALE |
|-----------|--------|-------|-----|
| norm      | L2     | STD   | L2  |
| α         | 3      | 3     | 3   |
| lr        | 0.0001 | 0.3   | 0.1 |
| syn num   | 100    | 2048  | 300 |
| nepoch    | 400    | 120   | 401 |
| nhF       | 2048   | 2048  | 2048|
| nz        | 1024   | 1024  | 1024|
| ndh       | 4096   | 4096  | 4096|
| attSize   | 312    | 312   | 312 |
| outzSize  | 512    | 512   | 512 |
| center weight | 0.5  | 0.5   | 0.5 |
| incenter weight | 0.8 | 0.8   | 0.8 |
| manualSeed | 3483   | 4115  | 9182|
| Table 3. Hyper-parameter selection details for all methods. |
| CLASS NAME | ALT. NAME | DESCRIPTION |
|------------|----------|-------------|
| CUB [49]   |          |             |
| Black footed Albatross | Laysan Albatross | Sooty Albatross | Groove billed Ani | Crested Auklet |
| Least Auklet | Parakeet Auklet | Rhinoceros Auklet | Brewer Blackbird | Red winged Blackbird |
| Rusty Blackbird | Yellow headed Blackbird | Bobolink | Indigo Bunting | Yellow breasted Chat |
| Painted Bunting | Cardinal | Spotted Caddie | Gray Catbird | Purple Finch |
| Eastern Towhee | Chuck will Widow | Brandt Cormorant | Red faced Cormorant | Pelecanus Cororant |
| Bronzed Cowbird | Shiny Cowbird | Brown Creeper | American Crow | Fish Crow |
| Black billed Cuckoo | Mangrove Cuckoo | Yellow billed Cuckoo | Gray crowned Rosy Finch | Olive sided Flycatcher |
| Northern Flicker | Acadian Flycatcher | Great Crested Flycatcher | Least Flycatcher | Northern Fulmar |
| Scissor tailed Flycatcher | Vermilion Flycatcher | Yellow bellied Flycatcher | Female Hummingbird | Green Violetear |
| Gadwall | American Goldfinch | European Goldfinch | Boat tailed Grackle | Eared Grebe |
| Horned Grebe | Pied billed Grebe | Western Grebe | Blue Grosbeak | Evening Grosbeak |
| Pine Grosbeak | Rose breasted Grosbeak | Pigeon Guilmot | California Gull | Glaucous winged Gull |
| Heermann Gull | Herring Gull | Ivory Gull | Ring billed Gull | Slaty backed Gull |
| Western Gull | Anna Hummingbird | Ruby throated Hummingbird | Rufous Hummingbird | Green Violetear |
| Long tailed Jaeger | Pomaire Jaeger | Blue Jay | Florida Jay | Green Jay |
| Dark eyed Junco | Tropical Kingbird | Gray Kingbird | Belted Kingfisher | Green Kingfisher |
| Pied Kingfisher | Ringed Kingfisher | White breasted Kingfisher | Red legged Kiwiwake | Horned Lark |
| Pacific Loon | Mallard | Western Meadowlark | Hooded Merganser | Red breasted Merganser |
| Mockingbird | Nighthawk | Clark Nutcracker | White breasted Nuthatch | Baltimore Oriole |
| Hooded Oriole | Orchard Oriole | Scott Oriole | Ovenbird | Brown Pelican |
| White Pelican | Western Wood Pewee | Sayornis | American Pipit | Whip poor Will |
| Horned Puffin | Common Raven | White necked Raven | American Redstart | Geococcyx |
| Loggerhead Shrike | Great Grey Shrike | Baird Sparrow | Black throated Sparrow | Brewer Sparrow |
| Chipping Sparrow | Clay colored Sparrow | House Sparrow | Field Sparrow | Fox Sparrow |
| Grasshopper Sparrow | Harris Sparrow | Henslow Sparrow | Le Conte Sparrow | Lincoln Sparrow |
| Nelson Sharp tailed Sparrow | Savannah Sparrow | Seaside Sparrow | Song Sparrow | Tree Sparrow |
| Vesper Sparrow | White crowned Sparrow | White throated Sparrow | Cape Glossy Starling | Bank Swallow |
| Barn Swallow | Cliff Swallow | Tree Swallow | Scarlet Tanager | Summer Tanager |
| Artic Tern | Black Tern | Caspian Tern | Common Tern | ... |

|-|-
|SUN [13]|-
|abbey| access road|
|airport airport | airport entrance |
|alley| amphitheater |
|apartment building outdoor | apse indoor |
|aqueduct| arch |
|arena hockey | arena performance |
|art school | art studio |
|atrium home | atrium public |
|auto mechanics indoor | auto racing paddock |
|badminton court indoor | badminton court indoor |
|balcony exterior | balcony interior |
|bank indoor | bank outdoor |
|baptistery outdoor | bar |
|basement| basilica |
|bathers box | bathe cage indoor |
|bazaar outdoor | beach |
|bedroom| beer garden |
|betting shop | bicycle racks |
|bistro outdoor | bleachers outdoor |
|boat outdoor | bookstore |
|bow window outdoor | bowling alley |
|brickyard outdoor | bridge |
|bullring | burial chamber |
|bus station outdoor | butchers shop |
|cafe| call center |
|canal urban | candy store |
|car interior frontseat | caravansary |
|carport outdoor | carrousel |
|catacomb | cathedral indoor |
|cemetery | chalet |
|cheese factory | chemical plant |

|AWA2 [10]|-
|antelope| grizzly bear |
|persian cat| horse |
|skunk| mole |
|moose | spider monkey |
|ox | fox |
|hamster | squirrel |
|giraffe | wolf |
|otter | buffalo |
|bobcat | pig |
|collie | walrus |
|killer whale | beaver |
|german shepherd | blue whale |
|tiger | hippopotamus |
|humpback whale | elephant |
|sheep | seal |
|rhinoceros | rabbit |
|chihuahua | giant panda |
|zebra | deer |
|lion | mouse |
|raccoon | cow |

Table 4. Some class names per dataset.

13
Table 5. Prompt and some class attributes labels per dataset.
Table 6. Results of Embedding Based Methods for the CUB [49] dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k) [11]. These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The ViT$_{huge}$ features pretrained on ImageNet-21k are the best for all the methods using ALE.
| Dataset | Arch Type | Backbone | DEVISE | ESZSL | ALE |
|---------|-----------|----------|--------|-------|-----|
|         |           |          | Seen   | Novel | Harm.| Seen   | Novel | Harm. | Seen   | Novel | Harm. |
| I-1k    | CNN       | RN101    | 32.75  | 18.54 | 23.68| 28.41  | 13.75 | 18.53 | 37.13  | 23.68 | 28.92 |
|         |           | RN101+FT | 34.03  | 19.93 | 25.14| 33.18  | 13.82 | 19.51 | 37.95  | 22.43 | 28.19 |
|         |           | RN50     | 29.84  | 17.43 | 22.01| 25.08  | 14.38 | 18.27 | 33.91  | 23.54 | 27.79 |
|         |           | RN152    | 30.39  | 17.29 | 22.04| 26.63  | 15.90 | 19.91 | 35.70  | 23.40 | 28.27 |
|         |           | GoogleNet| 18.02  | 10.83 | 13.53| 17.52  | 9.31  | 12.15 | 24.88  | 15.76 | 19.30 |
|         |           | VGG16    | 28.91  | 13.06 | 17.99| 25.85  | 9.93  | 14.35 | 31.78  | 20.21 | 24.71 |
|         |           | Alexnet  | 19.61  | 9.31  | 12.62| 19.03  | 6.88  | 10.10 | 23.60  | 13.82 | 17.43 |
|         |           | Shufflenet| 0.23  | 0.00  | 0.00 | 0.74   | 1.74  | 1.03  | 29.53  | 17.78 | 22.20 |
|         |           | Inceptionv3| 31.01 | 14.03 | 19.32| 25.08  | 11.25 | 15.53 | 32.29  | 18.47 | 23.50 |
|         |           | Inceptionv3_3adv| 27.91 | 15.00 | 19.51 | 24.69 | 12.08 | 16.23 | 33.88 | 21.39 | 26.22 |
|         | MLP       | MLP-Mixer| 6.32   | 3.33  | 4.36 | 6.40   | 2.36  | 3.45  | 8.29   | 3.96  | 5.36  |
|         | ViT       | ViTlarge | 52.33  | 24.51 | 33.39| 44.84  | 18.82 | 26.51 | 59.77  | 34.10 | 43.42 |
|         |           | DeiTbase | 37.17  | 17.22 | 23.54| 28.49  | 12.57 | 17.44 | 38.37  | 21.53 | 27.58 |
|         |           | ViT16-DINO†| 35.19 | 20.97 | 26.28| 32.48  | 16.11 | 21.54 | 40.89  | 28.40 | 33.52 |
| I-21k   | MLP       | MLP-MixerL16| 24.57 | 11.18 | 15.37| 20.74  | 9.24  | 12.78 | 29.03  | 13.61 | 18.53 |
|         | ViT       | ViTbase  | 45.19  | 25.07 | 32.25| 40.43  | 19.51 | 26.32 | 52.87  | 31.81 | 39.72 |
|         |           | ViTlarge | 49.15  | 25.56 | 33.63| 43.29  | 19.86 | 27.23 | 55.23  | 33.26 | 41.52 |
|         |           | ViThuge  | 38.22  | 19.24 | 25.59| 31.71  | 18.13 | 23.06 | 51.55  | 30.07 | 37.98 |

Table 7. Results of Embedding Based Methods for the SUN dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. Surprisingly, Shufflenet and RN50-MOCO features seem to be not suited for this dataset, and using ViTlarge pretrained on ImageNet-1k features with ALE beat all methods, including its counterpart pretrained on ImageNet-21k by a reasonable margin (1.9%). Moreover, using the ViTlarge pretrained on ImageNet-21k features beats all other methods when using DEVISE and ESZSL.
| Dataset Pret. on | Arch Type | Backbone | DEVISE Seen | DEVISE Novel | DEVISE Harm. | ESZSL Seen | ESZSL Novel | ESZSL Harm. | ALE Seen | ALE Novel | ALE Harm. |
|----------------|-----------|----------|-------------|-------------|-------------|------------|-------------|-------------|----------|-----------|----------|
| I-1k           | CNN       | RN101    | 71.78       | 17.30       | 27.88       | 88.84      | 4.04        | 7.72        | 77.59    | 12.15     | 21.01    |
|                |           | RN101+FT | 87.34       | 18.83       | 30.99       | 93.07      | 6.12        | 11.49       | 92.64    | 8.25      | 15.16    |
|                |           | RN50     | 86.02       | 19.49       | 31.78       | 89.05      | 4.75        | 9.02        | 84.37    | 10.48     | 18.65    |
|                |           | RN152    | 88.30       | **21.18**   | **34.17**   | 91.31      | 5.94        | 11.15       | 85.46    | 12.91     | 22.43    |
|                |           | GoogleNet| 68.56       | 17.40       | 27.76       | 80.11      | 4.26        | 8.10        | 82.82    | 5.13      | 9.66     |
|                |           | VGG16    | 78.85       | 16.21       | 26.89       | 90.06      | 3.11        | 6.00        | 77.34    | 11.52     | 20.06    |
|                |           | AlexNet  | 72.91       | 12.17       | 20.86       | 79.09      | 2.77        | 5.36        | 79.17    | 6.11      | 11.34    |
|                |           | Shufflenet| 74.74      | 20.04       | 31.61       | 51.66      | 3.69        | 6.89        | 80.75    | 8.84      | 15.93    |
|                |           | InceptionV3 | 74.54   | 8.08        | 14.58       | 91.49      | 4.43        | 8.46        | 78.24    | 10.32     | 18.23    |
|                |           | InceptionV3 Adv | 89.33 | 12.14       | 21.38       | 91.69      | 3.53        | 6.79        | 82.21    | 8.06      | 14.69    |
|                | MLP       | RN50-MOCO† | 78.23      | 11.07       | 19.39       | 57.62      | 3.73        | 7.01        | 81.51    | 4.60      | 8.71     |
|                | MLP       | RN50-DINO† | 78.97      | 18.11       | 29.46       | 81.25      | 8.08        | 14.71       | 82.78    | 7.62      | 13.96    |
|                | MLP       | MLP-Mixer | 21.06      | 10.87       | 14.34       | 38.51      | 2.67        | 4.99        | 94.29    | 12.78     | 22.52    |
|                | ViT       | ViT Large | 83.78      | 18.00       | 29.63       | 97.07      | **18.62**   | **31.25**   | 92.28    | 13.75     | 23.93    |
|                |           | DeiT base | **91.41**  | 9.51        | 17.22       | 94.08      | 2.33        | 4.54        | 87.37    | **14.67** | **25.12** |
|                |           | ViTB16-DINO† | 71.83 | 19.41       | 30.56       | 92.34      | 5.08        | 9.63        | 79.73    | 6.55      | 12.10    |
| I-21k          | MLP       | MLP-Mixer1,16 | 82.92 | 10.37       | 18.43       | 85.48      | 1.47        | 2.90        | 86.24    | 1.88      | 3.69     |
|                | ViT       | ViT base  | 82.35      | 14.05       | 24.00       | 96.18      | 7.77        | 14.38       | **95.62** | 11.07     | 19.85    |
|                |           | ViT Large | 86.49      | 19.53       | 31.87       | 96.47      | 10.43       | 18.83       | 95.03    | 12.76     | 22.49    |
|                |           | ViT huge  | 80.00      | 12.37       | 21.43       | 89.57      | 2.55        | 4.95        | 81.40    | 6.72      | 12.41    |

Table 8. Results of Embedding Based Methods for the AWA2 dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. Surprisingly, the most performant visual features are extracted from a RN152 pretrained on ImageNet-1k, using the DEVISE method.
### Table 9. Results of Generative Based Methods for the CUB dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The most performant visual features are extracted from a ViT\textsubscript{huge} pretrained on ImageNet-21k and fine-tuned with the seen classes, using the tfVAEGAN method.

| Dataset | Arch Pretrained on | Arch Type | Backbone | tfVAEGAN Seen | Novel | Harm. | CADA-VAE Seen | Novel | Harm. | CE Seen | Novel | Harm. |
|---------|-------------------|-----------|----------|---------------|-------|-------|---------------|-------|-------|---------|-------|-------|
|         |                   |           |          |               |       |       |               |       |       |         |       |       |
| I-1k    |                   | CNN       | RN101    | 57.08         | 42.88 | 48.97 | 58.27         | 49.71 | 53.65 | 60.09   | 49.05 | 54.01 |
|         |                   |           | RN101+FT | 72.44         | 53.66 | 61.65 | 76.45         | 57.53 | 65.65 | 76.71   | 48.81 | 59.66 |
|         |                   |           | RN50     | 49.29         | 42.35 | 45.55 | 45.88         | 38.95 | 42.13 | 42.79   | 35.46 | 38.78 |
|         |                   |           | RN152    | 50.23         | 44.48 | 47.18 | 47.58         | 41.64 | 44.41 | 45.77   | 35.52 | 40.00 |
|         |                   |           | GoogLeNet| 38.54         | 33.34 | 35.76 | 33.84         | 30.20 | 31.92 | 33.72   | 26.05 | 29.39 |
|         |                   |           | VGG16    | 36.67         | 38.46 | 37.54 | 37.12         | 35.38 | 36.23 | 35.00   | 37.84 | 36.36 |
|         |                   |           | AlexNet  | 21.48         | 32.52 | 25.87 | 22.36         | 24.31 | 23.29 | 23.73   | 28.62 | 25.95 |
|         |                   |           | ShuffleNet| 51.62        | 43.83 | 47.41 | 43.14         | 38.56 | 40.72 | 48.95   | 37.69 | 42.59 |
|         |                   |           | InceptionV3| 54.01       | 50.41 | 52.15 | 50.32         | 39.87 | 44.49 | 54.80   | 38.45 | 45.19 |
|         |                   |           | InceptionV3\textsubscript{adv} | 62.81 | 41.34 | 49.86 | 50.28         | 37.90 | 43.22 | 54.83   | 35.41 | 43.03 |
|         |                   | MLP       | RN50     | 41.88         | 29.13 | 34.36 | 27.40         | 22.39 | 24.64 | 34.01   | 24.09 | 28.21 |
|         |                   |           | RN50-MOCO† | 48.11     | 53.84 | 58.53 | 55.05         | 47.59 | 51.05 | 62.45   | 45.13 | 52.39 |
|         |                   | MLP-Mixer | ViT\textsubscript{large} | 80.34 | 54.34 | 64.83 | 61.23         | 53.31 | 56.99 | 70.22   | 43.07 | 53.39 |
|         |                   |           | DeiT\textsubscript{base} | 73.29 | 49.44 | 59.05 | 60.39         | 50.05 | 54.74 | 55.68   | 38.68 | 45.65 |
|         |                   |           | ViT\textsubscript{B16-DINO}† | 76.82 | 57.94 | 66.06 | 71.95         | 55.37 | 62.58 | 61.29   | 45.47 | 52.21 |
|         |                   | MLP       | ViT\textsubscript{huge} | 30.91 | 28.77 | 29.80 | 28.68         | 25.19 | 26.82 | 17.46   | 20.76 | 18.96 |
|         |                   | MLP-Mixer\textsubscript{16} | ViT\textsubscript{base} | 74.16 | 71.13 | 72.61 | 74.46         | 60.77 | 66.93 | 61.01   | 51.25 | 55.71 |
|         |                   |           | ViT\textsubscript{large} | 76.95 | 61.56 | 68.40 | 72.54         | 58.94 | 65.04 | 67.16   | 46.94 | 55.26 |
|         |                   |           | ViT\textsubscript{huge} | 75.15 | 62.76 | 68.80 | 70.53         | 60.50 | 65.13 | 49.37   | 43.76 | 46.40 |
|         |                   |           | ViT\textsubscript{huge}+FT | 78.32 | 76.26 | 77.27 | 77.99         | 74.46 | 76.18 | 70.87   | 44.66 | 54.79 |
### Generative Based GZSL Methods

**SUN Dataset**

| Dataset | Arch Type | Backbone | tVAEGAN | CADA-VAE | CE |
|---------|-----------|----------|---------|----------|----|
|         |           |          | Seen    | Novel    | Harm. | Seen   | Novel   | Harm.   | Seen    | Novel   | Harm.   |
| I-1k    | CNN       | RN101    | 38.95   | 45.62    | 42.03  | 34.15  | 48.96   | 40.23   | 51.24   | **55.83** | **53.44** |
|         |           | RN101+FT | 35.08   | 38.06    | 36.51  | 39.84  | 50.60   | 44.97   | 29.07   | 38.75   | 33.22   |
|         |           | RN50     | 34.31   | 45.07    | 39.15  | 34.07  | 41.53   | 37.43   | 23.95   | 42.36   | 30.60   |
|         |           | RN152    | 35.35   | 45.97    | 39.97  | 37.05  | 40.00   | 38.47   | 26.20   | 43.33   | 32.66   |
|         |           | GoogleNet| 27.17   | 38.26    | 31.78  | 24.96  | 36.32   | 29.59   | 20.43   | 38.96   | 26.80   |
|         |           | VGG16    | 25.04   | 28.61    | 26.71  | 31.32  | 37.85   | 34.27   | 24.11   | 36.81   | 29.13   |
|         |           | Alexnet  | 25.27   | 34.72    | 29.25  | 16.51  | 23.06   | 19.24   | 13.84   | 30.90   | 19.12   |
|         |           | Shufflenet| 31.59   | 42.71    | 36.32  | 30.62  | 37.64   | 33.77   | 24.92   | 37.50   | 29.94   |
|         |           | Inceptionv3| 30.00  | 32.15    | 31.04  | 32.64  | 38.26   | 35.23   | 26.09   | 37.43   | 30.74   |
|         |           | Inceptionv3 adv | 33.37 | 45.42    | 38.47  | 32.02  | 40.83   | 35.89   | 27.09   | 39.44   | 32.12   |
|         | CNN       | RN50-MOCO† | 37.44  | 42.99    | 40.02  | 35.08  | 38.13   | 36.54   | 30.62   | 44.24   | 36.19   |
|         |           | RN50-DINO† | 42.60  | 46.67    | 44.54  | 41.16  | 46.39   | 43.62   | 29.38   | 55.56   | 38.43   |
| I-21k   | MLP       | MLP-Mixer | 24.96  | 32.22    | 28.13  | 6.82   | 12.78   | 8.89    | 8.64    | 8.33     | 8.49     |
|         | ViT       | ViTlarge | 55.12  | **64.93** | **59.62** | **52.64** | **61.32** | **56.65** | 50.70   | 52.78   | 51.72   |
|         |           | DeiTbase | 34.34  | 44.72    | 38.85  | 37.44  | 44.51   | 40.67   | 36.36   | 37.57   | 36.95   |
|         |           | ViTB16-DINO† | 42.64 | 52.71    | 47.14  | 42.52  | 51.18   | 46.45   | 40.39   | 45.56   | 42.82   |
|         | MLP       | MLP-MixerL16 | 24.22 | 43.03    | 28.30  | 24.77  | 29.65   | 26.99   | 23.84   | 20.00   | 21.75   |
|         | ViT       | ViTbase | 54.19  | 58.75    | 56.38  | 52.02  | 60.14   | 55.78   | **51.53** | 50.74   | 51.13   |
|         |           | ViTlarge | **57.13** | 61.32    | 59.15  | 52.83  | 60.69   | 56.49   | 50.12   | 50.69   | 50.40   |
|         |           | ViThuge  | 43.37  | 53.61    | 47.95  | 47.64  | 51.18   | 49.34   | 49.79   | 16.05   | 24.27   |
|         |           | ViThuge+FT | 44.26 | 54.24    | 48.75  | 44.99  | 55.35   | 49.64   | 6.86    | 53.61   | 12.16   |

Table 10. Results of Generative Based Methods for the SUN dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. Surprisingly, the CE method does not seem to get any significant advantage from any of the ViT features, and overall, the most performant visual features are extracted from a ViTlarge pretrained on ImageNet-1k using the tVAEGAN method.
### Generative Based GZSL Methods
**AWA2 Dataset**

| Dataset Pret. | Arch Type | Backbone | tVAEGAN | CADA-VAE | CE |
|---------------|-----------|----------|---------|----------|----|
| Pret. on      |           |          | Seen    | Novel    | Harm. | Seen    | Novel    | Harm.    | Seen    | Novel    | Harm.    |
| I-1k          | CNN       | RN101    | 75.48   | 59.56    | 66.58  | 75.95   | 54.76    | 63.87    | 69.26   | 56.03    | 61.95    |
|               |           | RN101+FT | 84.81   | 58.44    | 69.20  | 77.74   | 59.95    | 69.14    | 83.43   | 47.66    | 60.66    |
|               |           | RN50     | 79.91   | 55.83    | 65.73  | 74.56   | 59.48    | 67.80    | 71.72   | 46.65    | 56.53    |
|               |           | RN152    | 84.48   | 60.01    | 70.17  | **88.88** | 57.69    | 70.48    | 76.74   | 40.25    | 52.8     |
|               |           | GoogleNet| 69.08   | 55.17    | 61.35  | 71.51   | 53.22    | 62.22    | 73.36   | 47.63    | 57.76    |
|               |           | Alexnet  | 64.31   | 41.11    | 50.16  | 61.98   | 40.80    | 49.60    | 57.13   | 39.39    | 46.63    |
|               |           | Shufflenet| 69.53  | 55.38    | 61.66  | 68.07   | 54.09    | 61.84    | 71.87   | 50.11    | 59.05    |
|               |           | Inceptionv3| 84.00  | 58.45    | 68.94  | 78.97   | 61.77    | 70.86    | 74.69   | 52.09    | 61.38    |
|               |           | Inceptionv3_adv| 85.48 | 59.21    | 69.96  | 82.15   | 53.43    | 65.23    | 66.96   | 53.73    | 59.62    |
|               | MLP       | MLP-Mixer | 28.14   | 25.27    | 26.63  | 14.01   | 41.56    | 27.90    | 22.93   | 20.68    | 21.75    |
|               | MLP       | MLP-MixerL16 | 72.63  | 42.47    | 53.60  | 70.15   | 51.01    | 60.12    | 61.51   | 39.01    | 47.74    |
|               | ViT       | ViTbase  | 54.19   | 58.75    | 56.38  | 84.48   | 67.42    | 76.67    | 77.80   | 49.54    | 60.53    |
|               | ViT       | ViTlarge | **91.05** | **63.58** | **74.87** | **88.69** | **70.75** | **80.40** | **78.32** | 59.21    | 67.44    |
|               | ViT       | ViTlarge  | 88.85   | 60.79    | 72.19  | 85.68   | 60.95    | 72.25    | 75.57   | 53.24    | 62.47    |
|               | ViT       | ViTlarge+FT | 68.23  | 61.63    | 64.76  | 80.69   | 60.76    | 69.32    | 75.23   | 60.10    | 66.82    |

Table 11. Results of Generative Based Methods for the AWA2 dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The most performant visual features are extracted from a ViTlarge pretrained on ImageNet-21k and fine-tuned with the seen classes, using the CADA-VAE method. More interestingly, the features from a ViTlarge pretrained on ImageNet-1k seem competitive with the features from a ViTlarge pretrained on ImageNet-21k for the CE and tVAEGAN methods respectively.
| Dataset Pret. on | Arch Type | Backbone  | SDGZSL | FREE |
|-----------------|-----------|-----------|--------|------|
| I-1k            | CNN       | RN101     | 56.41  | 58.30 |
|                 |           | RN101+FT  | 75.00  | 75.30 |
|                 |           | RN50      | 48.48  | 48.40 |
|                 |           | RN152     | 48.22  | 50.70 |
|                 |           | GoogleNet | 32.72  | 33.78 |
|                 |           | VGG16     | 37.20  | 30.02 |
|                 |           | Alexnet   | 27.12  | 15.47 |
|                 |           | Shufflenet| 47.66  | 47.65 |
|                 |           | Inceptionv3| 56.70 | 61.88 |
|                 |           | Inceptionv3\_adv | 51.38 | 60.82 |
|                 |           | RN50-MOCO | 36.67 | 42.53 |
|                 |           | RN50-DINO † | 59.27 | 66.62 |
| I-21k           | MLP       | MLP-Mixer | 17.66  | 17.59 |
|                 | ViT       | ViT\_large | 74.17  | 69.53 |
|                 |           | DeiT\_base | 62.91  | 61.80 |
|                 |           | ViT\_B16-DINO † | 68.72 | 73.70 |
|                 | MLP       | MLP-Mixer\_16 | 33.77  | 31.58 |
|                 | ViT       | ViT\_base | 78.01  | 67.50 |
|                 |           | ViT\_large | 79.55  | 73.07 |
|                 |           | ViT\_huge | 75.05  | 66.17 |
|                 |           | ViT\_huge+FT | 79.68 | 80.57 |

Table 12. Results of Disentanglement Based Methods for the CUB dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The most performant visual features are extracted from a ViT\_huge pretrained on ImageNet-21k and fine-tuned with the seen classes, using the SDGZSL method.
## Disentanglement Based GZSL Methods

### SUN Dataset

| Dataset Pret. on | Arch Type | Backbone       | SDGZSL    | FREE     |
|-----------------|-----------|----------------|-----------|----------|
|                 |           |                | Seen      | Novel    | Harm.    |
|                 | CNN       | I-1k           |           |          |          |
|                 |           | RN101          | 36.67     | 44.44    | 40.18    |
|                 |           | RN101+FT       | 38.57     | 49.44    | 43.33    |
|                 |           | RN50           | 33.33     | 41.87    | 37.12    |
|                 |           | RN152          | 34.77     | 44.44    | 39.01    |
|                 |           | GoogleNet      | 26.24     | 31.94    | 28.81    |
|                 |           | VGG16          | 30.81     | 33.82    | 32.25    |
|                 |           | Alexnet        | 22.64     | 25.76    | 24.10    |
|                 |           | Shufflenet     | 31.55     | 36.39    | 33.80    |
|                 |           | Inceptionv3    | 31.05     | 43.26    | 36.15    |
|                 |           | Inceptionv3\_adv | 31.43     | 44.37    | 36.80    |
|                 |           | RN50-MOCO\dagger | 36.16     | 37.85    | 36.99    |
|                 |           | RN50-DINO\dagger | 40.54     | 49.24    | 44.47    |
|                 | MLP       | MLP-Mixer      | 6.94      | 8.40     | 7.60     |
|                 |           | ViT\_large     | 51.59     | 63.75    | **57.03**|
|                 |           | DeiT\_base     | 31.94     | 45.69    | 37.60    |
|                 |           | ViTB16-DINO\dagger | 40.04     | 51.53    | 45.06    |
|                 | MLP       | MLP-Mixer\_16  | 19.92     | 32.99    | 24.84    |
|                 | VI T      | ViT\_base      | **51.63** | 55.56    | 53.52    |
|                 |           | ViT\_large     | 32.13     | **68.96**| 43.84    |
|                 |           | ViT\_huge      | 30.43     | 62.01    | 40.82    |
|                 |           | ViT\_huge+FT   | 39.53     | 47.99    | 43.35    |
|                 | I-21k     | MLP-Mixer\_16  | 19.92     | 32.99    | 24.84    |
|                 |           | ViT\_base      | **51.63** | 55.56    | 53.52    |
|                 |           | ViT\_large     | 32.13     | **68.96**| 43.84    |
|                 |           | ViT\_huge      | 30.43     | 62.01    | 40.82    |
|                 |           | ViT\_huge+FT   | 39.53     | 47.99    | 43.35    |

Table 13. Results of Disentanglement Based Methods for the SUN dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-Ik (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The most performant visual features are extracted from a ViT\_large pretrained on ImageNet-Ik using the SDGZSL method.
### Disentanglement Based GZSL Methods

**AWA2 Dataset**

| Dataset Pret. on | Arch Type | Backbone | SDGZSL | FREE |
|-----------------|-----------|----------|--------|------|
|                 | CNN       |          |        |      |
| I-1k            |           |          |        |      |
|                 | MLP       |          |        |      |
|                 | MLP-Mixer |          |        |      |
|                 | ViT       |          |        |      |
|                 | ViTbase   |          |        |      |
|                 | ViTlarge  |          |        |      |
|                 | ViTbase+FT|          |        |      |
|                 | ViThuge   |          |        |      |
|                 | ViThuge+FT|          |        |      |
|                 | MLP       |          |        |      |
| I-21k           |           |          |        |      |
|                 | MLP       |          |        |      |

*Table 14. Results of Disentanglement Based Methods for the AWA2 dataset using different features extracted from a diverse set of architecture types pretrained on ImageNet-1k (I-1k) and ImageNet-21k (I-21k). These backbones were trained via: supervised and self-supervised (†) learning. The bold numbers correspond to the highest scores per column, and the shaded rows correspond to the most performant image feature per method. +FT indicates the features were fine-tuned with the seen classes from the training set. The most performant visual features are extracted from a ViT<sub>large</sub> pretrained on ImageNet-1k using the SDGZSL method.*
Table 15. Results of Generative and Disentanglement Based Methods for the CUB, SUN and AWA2 datasets using different features extracted from different size and architecture of the visual head from diverse CLIP models (i.e., ResNet50 (R50) and Vision Transformer (ViT)). The three bottom rows per section correspond to fine-tuned features using sentences with: † the class names, ‡ the attributes, and § both class names and attributes. The bold numbers correspond to the highest scores per column, the underline numbers correspond to the highest scores using features not fine-tuned, and the shaded rows correspond to the most performant image feature per method over all. Surprisingly, the most performant method for all datasets correspond to a generative based method.