

# Large-Scale Zero-Shot Image Classification from Rich and Diverse Textual Descriptions

**Sebastian Bujwid**

KTH Royal Institute of Technology  
Stockholm, Sweden  
bujwid@kth.se

**Josephine Sullivan**

KTH Royal Institute of Technology  
Stockholm, Sweden  
sullivan@kth.se

## Abstract

We study the impact of using rich and diverse textual descriptions of classes for zero-shot learning (ZSL) on ImageNet. We create a new dataset *ImageNet-Wiki* that matches each ImageNet class to its corresponding Wikipedia article. We show that merely employing these Wikipedia articles as class descriptions yields much higher ZSL performance than prior works. Even a simple model using this type of auxiliary data outperforms state-of-the-art models that rely on standard features of word embedding encodings of class names. These results highlight the usefulness and importance of textual descriptions for ZSL, as well as the relative importance of auxiliary data type compared to algorithmic progress. Our experimental results also show that standard zero-shot learning approaches generalize poorly across categories of classes.

## 1 Introduction

Zero-shot learning (ZSL) relates information from different modalities (e.g., text or attributes with images), and the hope is that a sparsity of information or training data in one modality can be compensated by the other. This is important when the cost of creation or collection of training data greatly differs between the different modalities. Natural language descriptions and existing text repositories can provide this rich and practically accessible information about visual concepts and classes when no accompanying labeled images are available.

Recent works on zero-shot image classification (Xian et al., 2019; Schonfeld et al., 2019; Xian et al., 2018) show the field’s substantial progress on many of the standard ZSL benchmarks. However, most of those benchmarks mostly cover either a very small or narrow set of classes (e.g., only bird species), where human-made class attributes are often used as auxiliary data. Unfortunately, on

ImageNet, where such attributes are not available, the performance is still very low. Therefore, in this work, rather than focus on algorithmic development, we instead study how the auxiliary data type impacts performance. We evaluate the benefits of using textual descriptions of the ImageNet classes. We collect text from Wikipedia articles describing each class in ImageNet, illustrated in Figure 1. Throughout the paper, we refer to the dataset of ImageNet classes and their corresponding Wikipedia articles, as well as the extracted text, as *ImageNet-Wiki*.<sup>1</sup>

The use of textual descriptions for ZSL has been studied before on smaller datasets (Reed et al., 2016; Elhoseiny et al., 2016, 2017; Zhu et al., 2018). However, this is the first work to use such textual descriptions on a large dataset, with a broad set of classes. The major differences between commonly used datasets are highlighted in Table 1. The large-scale setup enables us to study the main challenges of a more realistic and practical zero-shot image classification scenario and study the generalization of models to novel groups of classes (e.g., animal species in general), not only individual classes (e.g., specific animal species).

Our experimental results lead us to two significant observations. First, models using the textual descriptions perform much better than those using class name features. Remarkably, a simple ZSL model based on DeVISE (Frome et al., 2013) and trained on text from *ImageNet-Wiki* clearly outperforms more recent, state-of-the-art models that use word embedding features of class names as the auxiliary data. For the CADA-VAE model (Schonfeld et al., 2019), *ImageNet-Wiki* leads to almost a 12 pp. improvement (38.63% vs. 50.50%) in top-5 accuracy on ImageNet *mp500* test split, which is far higher than achieved in any prior work. These

<sup>1</sup>The dataset, code and trained models available at: <https://bujwid.eu/p/zsl-imagenet-wiki>



Figure 1: **ImageNet classes and their corresponding auxiliary data: Wikipedia articles or class names.** Above illustrates the difference between class names, the standard source of auxiliary data used on ImageNet, and the textual descriptions we collected. The short, selected article fragments shown contain information that is visually discriminative for the classes. In ZSL, for seen classes, we have access to both auxiliary data and images. However, for unseen classes only auxiliary data is available for training, with no images.

Table 1: **High-level overview of the most popular existing zero-shot learning image datasets.** Numbers with (\*) correspond only to the training set only. Provided by: (†) - Reed et al. (2016); (‡) - Elhoseiny et al. (2017).

Dataset	# images	# classes	Used auxiliary data			Semantic classes
			Other	Wiki articles	Attrib.	
aP/aY	15k	32			✓	person, animals, objects
AwA1	30k	50			✓	animals
AwA2	37k	50			✓	animals
LAD	78k	230			✓	animals, fruits, objects, other
SUN	14k	717			✓	scenes
CUB	11k	200	short image capts. (†)	✓(‡)	✓	birds
NABirds	48k	1,011		✓(‡)	✓	birds
ImageNet-Wiki	1.3M*	1,000*	class names	✓(this work)		animals, plants, objects, other

results strongly suggest that algorithmic development on ZSL models is not sufficient for further substantial progress. It is necessary to also consider the quality and type of auxiliary data used. Fortunately, creating or collecting already available textual information about classes is practical, viable, and much less labor-intensive than collecting and annotating images with labels or attributes. Our second main observation is that, regardless of the type of auxiliary data used, ZSL models generalize poorly from one large class category to another. For example, excluding all animal classes from the training set leads to large performance drops on animal species in the test set. Therefore, the field’s algorithmic progress measured on small datasets with few categories (Table 1) might not be well aligned with the end goal of using ZSL to scale image classifiers to diverse sets of unseen classes

from many diverse categories. Though textual descriptions are already available for smaller datasets (CUB, NABirds) we believe *ImageNet-Wiki* will facilitate further research in this area.

The textual descriptions for ZSL lead to significant performance improvements, but they also have several theoretical and practical advantages over other types of auxiliary data, as we detail in Table 2. Moreover, we believe a large-scale zero-shot learning setup with rich and diverse textual descriptions can be a useful for studying general multimodal machine learning, specifically on the interaction between language and vision. This is because the task is challenging and requires effective modeling of the interaction between modalities for better-than-random accuracy (unlike say VQA where relying on language priors may be possible (Zhang et al., 2016; Goyal et al., 2017)). Additionally, *ImageNet-*

Table 2: Advantages and disadvantages of different auxiliary data types for zero-shot image classification.

Auxiliary data type	Advantages / Disadvantages
Attributes	<ul style="list-style-type: none"> <li>+ <i>Little noise</i>: attributes selected by experts</li> <li>+ <i>Potentially highly discriminative</i>: depending on the selected features, might be highly informative</li> <li>– <i>Difficult to scale</i>: as the number of classes increases, the more attributes are required to discriminate</li> <li>– <i>Simple relations</i>: attributes may be conditional and difficult to quantify (e.g. zebra has brown and white stripes at birth)</li> <li>– <i>Expensive annotation</i>: requires domain knowledge to define discriminative attributes</li> <li>– <i>Low flexibility</i>: difficult to aggregate multiple independent datasets as attributes not standardized</li> </ul>
Word embedding of class names	<ul style="list-style-type: none"> <li>+ <i>Very simple approach</i>: little or no labor required to create data</li> <li>– <i>Sensitive to linguistic issues</i>: names may partially overlap or not reflect well semantic similarity</li> <li>– <i>Little information</i>: the name gives little discriminability of classes</li> </ul>
Text descriptions	<ul style="list-style-type: none"> <li>+ <i>Rich in information</i>: likely to be discriminative</li> <li>+ <i>Easy to collect</i>: many existing resources</li> <li>+ <i>Easy aggregation</i>: allows conceptually easy aggregation of text from multiple sources</li> <li>– <i>High noise</i>: contains non-relevant information</li> </ul>

*Wiki* covers a broad range of many classes, with Wikipedia articles that are long, complex, diverse, and written by many authors. We expect that this experimental setting will lead to more progress as methods become more tailored to relating text language and vision.

**Summary of the contributions** We demonstrate that simply using better auxiliary data in the form of Wikipedia description of classes outperforms all prior works on zero-shot learning on the *mp500* test split of ImageNet. Additionally, we show ZSL models generalize very poorly to novel super-categories of classes. Finally, we introduce *ImageNet-Wiki*, which provides text from Wikipedia articles of ImageNet classes, which we hope will facilitate the research on ZSL and the interaction between language and vision.

## 2 Related works

**Zero-shot learning** The problem of zero-shot learning (ZSL) was initially defined and introduced by Larochelle et al. (2008). A significant fraction of the recent works on ZSL uses attributes as auxiliary data, and small or medium-sized datasets (Xian et al., 2018; Romera-Paredes and Torr, 2015; Changpinyo et al., 2016; Schonfeld et al., 2019). The datasets generally used contain either animal species (CUB, AWA1, AWA2), flower species (Oxford Flower Dataset), or a broader set of classes (SUN, aPY). Some of the works experiment on ImageNet, but due to the lack of attribute annotations, they use word2vec embeddings of class names instead (Xian et al., 2018; Changpinyo et al., 2016; Schonfeld et al., 2019). Creating discriminative attribute annotations becomes more challenging as the number of classes grows, and the differences between classes become more nuanced. We would like the *ImageNet-Wiki* dataset to stimulate future research on large-scale ZSL as it provides a semantically rich source of auxiliary data, shown by our experiments, for ImageNet.

**Zero-shot learning from text** The portion of ZSL research that is more closely related to our work uses text as auxiliary data. Reed et al. (2016) collected a dataset with multiple single-sentence image-level descriptions for two datasets: CUB-2011 containing bird species and Oxford Flowers dataset with flower species. The general idea of utilizing Wikipedia articles for ZSL has already been studied by Elhoseiny et al. (2016, 2017) on datasets with bird species (CUB-2011, NABirds). The usefulness of such data was more recently additionally supported by Zhu et al. (2018) and Chen et al. (2020). These works were, however, limited to relatively small datasets with only closely related classes compared to ImageNet, which is a much larger dataset and covers a more diverse set of classes.

## 3 ImageNet–Wikipedia correspondences

ImageNet classes correspond to a subset of synsets representing nouns in WordNet 3.0. Each class is defined by its wnid (WordNet ID), class phrases (which we also refer to as *class names* in this paper), and a gloss (brief description), for example (n02129165; “lion, king of beasts, Panthera leo”; *large gregarious predatory feline of Africa and India having a tawny coat with a shaggy mane in*

*the male*). The synsets have a hierarchical structure, such that our example class is a descendant of classes like “big cat” and “animal”.

We create an automatic matching of ImageNet classes to their corresponding Wikipedia pages.<sup>2</sup> The matching is based on the similarity between the synset *words* and Wikipedia titles, as well as their ancestor categories. Unfortunately, such matching occasionally produces false-positives, as it is a difficult problem. One reason for this could be that classes with similar names might represent very different concepts – e.g. (n02389026; “sorrel”; *a horse of a brownish orange to light brown color*) with the herb called “sorrel”. To ensure high-quality correspondences in *ImageNet-Wiki*, we first compared all automatic correspondences from both Niemann and Gurevych (2011) and Matuschek and Gurevych (2013), then also manually verified all the matches, and modified them if necessary. The quality of the automatic matches often depends more, for example, on how irrelevant articles are filtered out than on the exact matching approach taken. We refer the interested reader to the source code we provide for the full details.

#### 4 Encoding textual descriptions for zero-shot learning

The text for each class is encoded into a single feature vector. Having features of fixed dimensionality for each class allows us to use standard ZSL approaches. As text encoders we use either ALBERT (Lan et al., 2019), which is an NLP model based on Transformer (Vaswani et al., 2017), or word embeddings models: GloVe (Pennington et al., 2014) or word2vec (Mikolov et al., 2013).

For ALBERT we consider two official pre-trained models: ALBERT-base and ALBERT-xxlarge, which have hidden layer sizes of 768 and 4096 respectively. ALBERT (Lan et al., 2019), similarly to its predecessor BERT (Devlin et al., 2019), uses learnable positional encoding, which has fixed size. Therefore in order to encode pages that can be longer than the maximal length, we split the text into partially overlapping sequences of 256 tokens with an overlap of 50 tokens. We encode each of the sequences with ALBERT, and average the hidden states from the last layer over each token. Such representations of all the sequences are then

<sup>2</sup>We use enwiki-20200120 dump from <https://dumps.wikimedia.org/backup-index.html> (Accessed: 22 Jan 2020).

also averaged to get the final feature vector. We compared alternative ways to extract features in Appendix E.

For word embedding models, either GloVe (Pennington et al., 2014) or word2vec (Mikolov et al., 2013), we use official pre-trained weights. To encode the text we simply embed each token and then average the embedding features over the whole sequence. Finally, if a class has multiple corresponding articles, we average the representations obtained from each of them.

## 5 Experiments

To compare the quality of different types of auxiliary data, we run experiments on two existing ZSL models. We choose to run all experiments on the standard ZSL task, instead of generalized ZSL (GZSL). That means we predict only unseen classes and not the union between seen and unseen classes. Although GZSL is the more practical long-term scenario, the standard ZSL setup better isolates the usefulness and discriminativeness of auxiliary data from the model’s level of overfitting to the seen classes. As we introduce no new methods and focus on the importance of auxiliary data instead, the standard ZSL setup is more informative in our case.

### 5.1 Zero-shot learning models

We run experiments with a state-of-the-art CADA-VAE (Schonfeld et al., 2019) model. Although it was initially proposed for GZSL, we found that it also works well for the standard ZSL setup. To compare the relative importance of the quality of the model vs. the quality of information contained in auxiliary data, the second model we use is a simple approach based on linear projections. We refer to it as *Simple ZSL*.

**CADA-VAE (Schonfeld et al., 2019)** The model is based on two Variational Auto-Encoders (VAEs) (Diederik et al., 2014) - one for image features and another for auxiliary data. It aligns the latent spaces of the two VAEs, such that the latent features of an image encoded with the image feature VAE and its class features encoded with auxiliary data VAE should be close to each other. After training the VAEs, a linear classifier is trained from the latent space vectors, obtained from the text description of the unseen classes, to the class labels of



these unseen classes.<sup>3</sup> At test time the latent space features obtained from the image VAE are passed to the classifier. We use our own implementation of the model which yields similar results to those reported in the original paper. The details can be found in Appendix F, along with a discussion on the small differences between our and the original implementation.

**Simple ZSL model** We also create a simple zero-shot learning model inspired by DeVISE (Frome et al., 2013). Unlike the original work, we do not fine-tune the visual model and use two separate linear transformations for image features and auxiliary data to map the different modalities to a joint embedding space. The transformations are trained using a hinge loss which is defined: Assume there are  $K$  classes and  $\mathbf{x}$  is the visual feature representation of a sample with class label  $1 \leq y \leq K$  then the loss is

$$\sum_{\substack{i=1 \\ i \neq y}}^K \max(0, m - \phi_x(\mathbf{x})^T \phi_t(\mathbf{t}_y) + \phi_x(\mathbf{x})^T \phi_t(\mathbf{t}_i)) \quad (1)$$

where  $\mathbf{t}_i$  is the feature vector for the auxiliary data of class  $i$ ,  $m \geq 0$  is a scalar corresponding to the margin,  $\phi_x(\mathbf{x}) = W_x \mathbf{x} + \mathbf{b}_x$  and  $\phi_t(\mathbf{t}_i) = W_t \mathbf{t}_i + \mathbf{b}_t$ . Then  $W_x$ ,  $W_t$ ,  $\mathbf{b}_x$  and  $\mathbf{b}_t$  are the trainable parameters in the model. At test time, to classify samples, we use the nearest neighbors between projected visual features and text labels.

## 5.2 Experimental setup

Our textual descriptions are assessed by comparing them to using word2vec embeddings (Mikolov et al., 2013) of class names as the auxiliary data provided by Changpinyo et al. (2016). The latter is a standard feature used in zero-shot learning on ImageNet (Xian et al., 2018, 2019; Schonfeld et al., 2019). These word2vec representations were also learned on Wikipedia data. However, this is different from encoding explicit textual descriptions of classes.

**Data** We use standard splits between ImageNet classes proposed by Xian et al. (2018). The *train* and *val* splits consist of 750 and 250 classes, which contain images from ILSVRC-2012 (Russakovsky

et al., 2015). Even though the original setup consists of multiple sets of separate test classes, we only use one of them: *mp500* - the 500 most populated classes. We leave studying other splits for future work since it requires providing the corresponding Wikipedia articles for the additional classes in those splits.

**Features** To represent images we use the 2048-dimensional features provided by Xian et al. (2018), which come from the last layer of a ResNet-101 before pooling. The model was pre-trained on ILSVRC 2012 part of ImageNet, which corresponds to the *trainval* ZSL split we use. Word2vec vector features of ImageNet class names (referred to as word2vec\*) are from Changpinyo et al. (2016) have 500-dimensions and were trained on Wikipedia data. These are standard features used on ImageNet and were used by all the models we compare against. For encoding the article text with word2vec, we instead use standard pre-trained 300-dimensional vectors provided by Mikolov et al. (2013). GloVe features are also 300-dimensional and are from Pennington et al. (2014). In all the experiments, we keep both image and auxiliary features fixed.

**Choice of the hyper-parameters** The hyperparameter values of the models are selected independently for each model and type of auxiliary data feature. The models are trained on the set of *train* classes and their performance evaluated on the *val* classes. We use both random and manual searches over possible settings, and the same number of runs was made on each model variant. More details are in Appendix F. The *val* set consists of a subset of classes that were used to train the image feature extractor, which violates ZSL assumptions. Although that is likely to lead to over-optimistic numbers, *val* performance is used only for the hyperparameter selection.

**Evaluation** We evaluate the models using the mean per class top-1 and top-5 accuracy which are standard ZSL metrics. All the hyperparameter values used for evaluation on *mp500* set were solely determined based on *val* performance of the models. For the final evaluation, we train the models on *train+val* with 1000 classes (750 *train* + 250 *val*) and use separate 500 classes from *mp500* for measuring performance. Since *ImageNet-Wiki* contains auxiliary features for 489 out of 500 classes from *mp500*, unless stated differently, for a fair com-

<sup>3</sup>This describes the ZSL setting. Originally for GZSL, the latent features from the image samples are also used.

parison, we compute the accuracies of the models using Wikipedia articles assuming 0 accuracy on the remaining classes.

### 5.3 Results

**Comparison of different auxiliary data encodings** First, in Table 3, we compare ways to encode the auxiliary data on the *val* set only. We observe that using the whole Wiki articles works better than using just the abstracts (first paragraphs of the articles before any section starts). Also, word embedding encoding of the text appears to work better than the more complex ALBERT model (Lan et al., 2019) with GloVe (Pennington et al., 2014) being the best feature extractor. For completeness of the comparison, we also try to encode class names with ALBERT and GloVe, although ALBERT was designed to work with longer sequences, therefore it can perform poorly with simple class names.

Table 3: **Validation set comparison of different textual inputs and encodings for zero-shot learning on ImageNet.** 750 *train* classes were used as seen and 250 *val* classes as unseen. Hyperparameters for each features type were tuned only on Wiki article data. All ALBERT encodings are of type ALBERT (xxlarge). word2vec\* refers to the features provided by Xian et al. (2018) typically used for ZSL, which are different from the standard pretrained word2vec model.

Auxiliary data	Features	Result on val	
		top-1 (%)	top-5 (%)
Wiki article	GloVe	<b>40.24</b>	<b>77.96</b>
Wiki article	word2vec	36.14	72.80
Wiki article	ALBERT	31.29	64.92
Wiki abstract	ALBERT	22.71	52.84
Class names	ALBERT	14.19	30.53
Gloss	ALBERT	19.67	48.09
Class names & gloss	ALBERT	21.38	49.75
Class names	GloVe	29.47	59.26
Gloss	GloVe	22.41	54.91
Class names & gloss	GloVe	33.26	65.47
Class names*	word2vec*	27.87	61.91

**Evaluating models and feature types on the test set** In Table 4, we compare zero-shot learning accuracies of various models on the *mp500* test split. Our experiments compare two different types of auxiliary information, class names encoded with word2vec, which is a standard approach on Im-

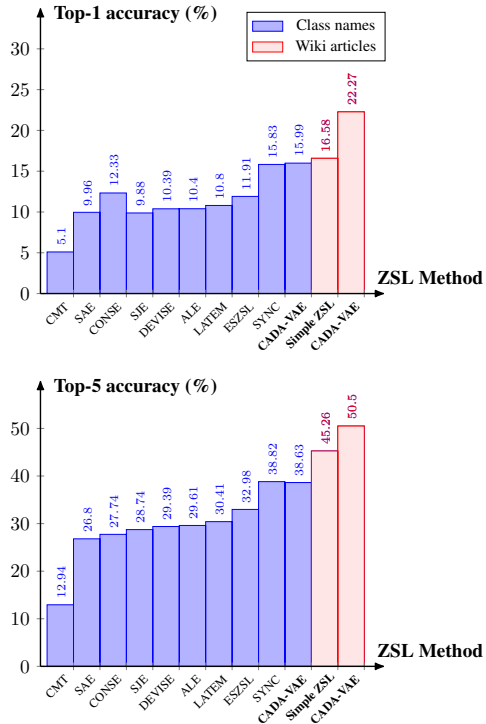


Figure 2: Reported results for zero-shot learning on the ImageNet dataset and the mp500 test split. Color indicates the auxiliary data type used. Methods in bold-face correspond to the implementations in our paper, the other numbers were reported by Xian et al. (2018). We chose the best performing variants of our models from Table 4.

ageNet, as well as using Wikipedia articles that we extract, encoded with two variants of ALBERT and two variants of word embedding encodings. The results show that models using the textual descriptions consistently achieve higher accuracies. The increased performance is even more prominent on the simple ZSL model. When using textual descriptions, such a simple model outperforms much more complex and generally better CADA-VAE trained with word2vec features of class names. This demonstrates not only the high quality of information the Wikipedia articles convey about the classes but also that the information can be effectively extracted even by simple models. In Figure 2, we additionally compare the results we achieve with those reported by Xian et al. (2018) of different zero-shot learning methods. The general setup we used is the same and uses the same image features, data splits and word2vec vectors of the class names. The comparison demonstrates that the models using Wikipedia articles outperform all the prior methods, highlighting the relative impor-

Table 4: Zero-shot learning results of the models we evaluate on ImageNet dataset. word2vec\* uses features from Changpinyo et al. (2016) and is different from word2vec which are standard pre-trained vectors. We report the mean and standard deviation values from 5 training runs with different random seeds.

Model	Auxiliary data	Features	Result on mp500	
			top-1 (%)	top-5 (%)
CADA-VAE (Schonfeld et al., 2019)	class names	word2vec*	15.99 ± 0.25	38.63 ± 0.43
CADA-VAE (Schonfeld et al., 2019)	Wiki articles	ALBERT (base)	17.04 ± 0.13	40.23 ± 0.27
CADA-VAE (Schonfeld et al., 2019)	Wiki articles	ALBERT (xxlarge)	18.84 ± 0.21	43.94 ± 0.24
CADA-VAE (Schonfeld et al., 2019)	Wiki articles	word2vec	20.34 ± 0.31	47.50 ± 0.01
CADA-VAE (Schonfeld et al., 2019)	Wiki articles	GloVe	<b>22.27 ± 0.20</b>	<b>50.50 ± 0.33</b>
Simple ZSL	class names	word2vec*	9.08 ± 0.18	27.31 ± 0.23
Simple ZSL	Wiki articles	ALBERT (base)	13.91 ± 0.23	37.31 ± 0.18
Simple ZSL	Wiki articles	ALBERT (xxlarge)	<b>16.60 ± 0.08</b>	42.39 ± 0.30
Simple ZSL	Wiki articles	word2vec	14.84 ± 0.24	42.93 ± 0.20
Simple ZSL	Wiki articles	GloVe	<b>16.58 ± 0.19</b>	<b>45.26 ± 0.24</b>

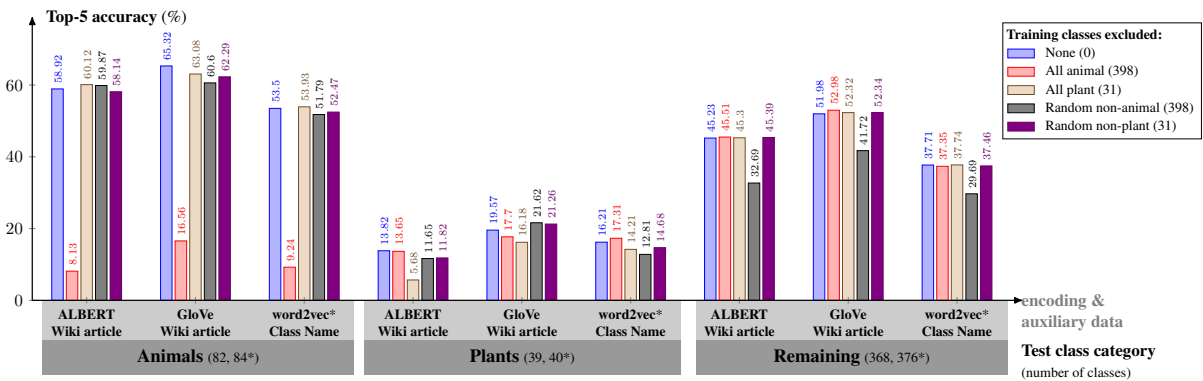


Figure 3: **Within and across category generalization of ZSL. Top-5 per class accuracy on CADA-VAE.** We show the effect of excluding a category of classes from the training set on ZSL performance for all mp500 test set categories. Specifically we compare excluding all *animal* classes vs. excluding the same number of random *non-animal* classes. We repeat the same process for the *plant* class. Different models trained on different auxiliary data are compared. The performance numbers are from a single run each. The performance on the unseen *animal* classes drops dramatically when the animal classes are removed from the training set, the red bars in the first grouping of plots. The same trend, though not as pronounced, can be seen for the *remaining* classes, the gray bars in the last grouping of plots. The number of test classes is slightly higher for models using class name data (marked with \*) since Wiki articles are missing for some classes.

tance of the auxiliary data type and algorithmic development.

### Generalization outside the seen groups of classes

The hierarchical structure of WordNet entities, used as ImageNet classes, allows us to study different groups of classes separately. In Figure 3, we split the set of both train and test classes into three disjoint partitions: *animals*, *plants*, and *remaining* and compare the test performance on all subgroups when some of them are excluded from the training set. First of all, the performance on *plants* is generally much lower than the much more populated *animals* group. Moreover, we see that all the models have poor accuracy on *animals* when

that group is excluded from the training set. These results show the models’ inability to generalize to unseen classes that are semantically far from seen classes. Therefore, they also suggest that studying ZSL on ImageNet might be more practical than datasets with a less diverse set of classes.

### Analyzing the impact of class similarity

Even though ImageNet classes represent distinct concepts, since WordNet and Wikipedia were created independently, naturally, the granularity of the concepts they define can be different. As a consequence, some of the classes can be mapped to multiple articles. For example, class (n02965783, “car mirror”) was assigned two more detailed articles:

Table 5: **Performance on subsets of mp500 that excludes classes that are similar to those in the training set.** We take two CADA-VAE models (single runs), trained to predict all unseen classes and present their mean per-class accuracies of different subsets of the *mp500* classes. Test classes are excluded based on their overlap with the train+val classes in either: Wikipedia pages correspondences or WordNet phrases (defining the classes). Each row corresponds to a different subset of excluded classes. Examples of excluded classes are: **Row 3:** (n03792972, “mountain tent”) is removed as train+val contains (n02769963, “backpacking tent”, “pack tent”) and all are mapped to “Tent”. **Row 4:** (n04543772, “wagon wheel”) is removed as it maps to [“Wagon”, “Wheel”], and class (n02974003, “car wheel”) in train+val maps to “Wheel”. **Row 5:** (n03222516, “doorbell, bell, buzzer”) is removed as there is (n03017168, “chime, bell, gong”) in train+val. The numbers shown in bold display the results with the biggest drop in performance.

Classes excluded	# classes left	ALBERT-xxlarge Wiki articles		GloVe Wiki articles		word2vec class names	
		top-1 (%)	top-5 (%)	top-1 (%)	top-5 (%)	top-1 (%)	top-5 (%)
None	500	18.73	44.03	22.26	50.49	16.38	39.07
No matching articles	489	19.16	45.02	22.76	51.63	16.28	39.00
Same set of articles matched (*)	459	16.82	42.63	21.08	49.82	16.18	39.04
(Any) article overlap	439	<b>15.07</b>	<b>40.64</b>	<b>19.87</b>	<b>48.38</b>	15.68	38.60
Class names overlap (†)	463	18.24	44.50	21.99	51.31	<b>15.53</b>	<b>38.05</b>
(*) $\cup$ (†)	438	16.39	42.42	20.79	49.76	15.70	38.35

“Rear-view mirror”, “Wing mirror”. In fact, WordNet synsets used as ImageNet classes consist of one or more (closely related) concepts that are considered equivalent. On the other hand, two different classes can be assigned the same set of articles. Potential overlap of sources of auxiliary information (Wiki pages or class phrases) can be present due to a high degree of similarity between corresponding classes, or compound classes being defined on other classes (e.g. *wagon wheel* being a *wheel* of a *wagon*). This phenomenon is associated not only with the particular auxiliary data we propose. As we observe in Table 5, models using both sources of data have lower performance on subsets of test classes which exclude those that are similar to the training classes under various criteria defined for those types of auxiliary data. As shown in the last row, the model using Wikipedia text still outperforms the one using class phrases on the subset of test classes that exclude those that are similar to training classes in either Wikipedia pages or class phrases. However, as demonstrated in our results in Figure 3, certain degree of similarity between seen and unseen classes is necessary for reasonable performance.

**Qualitative evaluation of prediction** To provide further insight into models’ performance, we analyze confusion matrices with predictions of different models. Due to the immense size, the full

confusion matrices are included only in the supplementary material. Figure 4 shows only a selected subset of the confusion matrices, only with classes that were often confused. The mispredictions are mainly within relatively semantically similar classes – different types of furniture in this case. Generally, the pattern for models using Wiki articles seems more diagonal, with more values close to 0. This suggests that Wiki articles are richer in discriminative information. More figures are available in Appendix C.

**Text length vs. performance** Finally, we also analyse the impact of the length of textual descriptions of classes and performance on them. Longer descriptions tend to lead to higher accuracy. The trend is not strong, however. The details are discussed in Appendix B.

## 6 Limitations & future work

The performance of our approach is likely to depend on some general quality of Wikipedia articles. Although we have not evaluated this dependence directly in our work, at least the article’s length indeed correlates with the corresponding class’s accuracy (Appendix B). Therefore, the adaptability of the general approach to some particular domains not well-covered in Wikipedia can also be limited. Additionally, we assume that the classes used are possible to match with Wikipedia articles. With



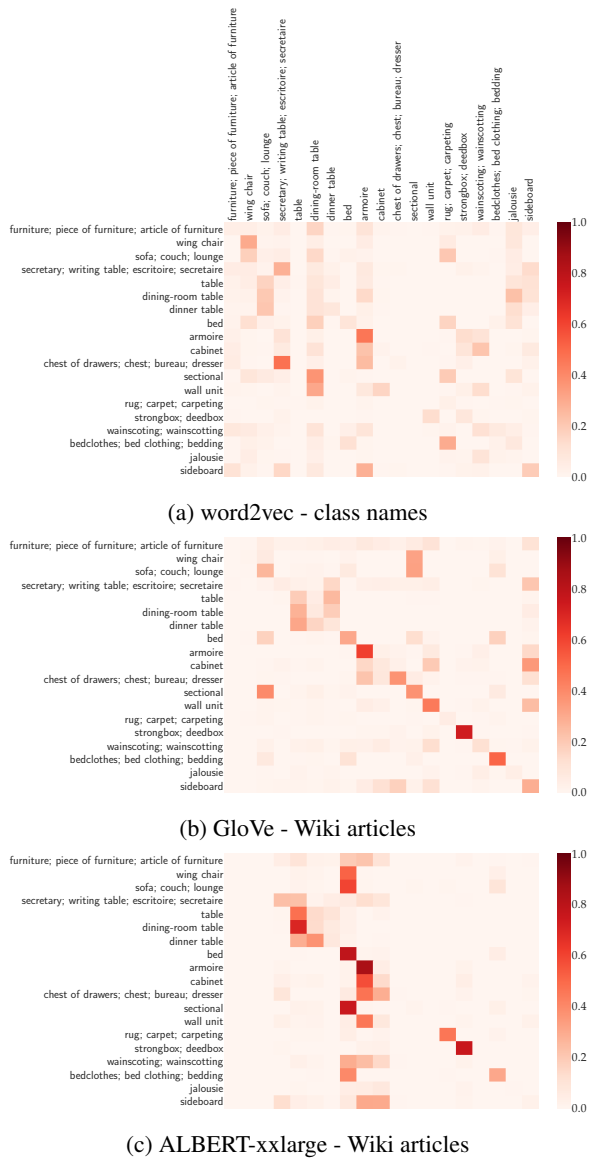


Figure 4: **A selected subset of confusion matrices on mp500 test set.** All the models are CADA-VAE (Schonfeld et al., 2019) trained on different types of auxiliary data. The rows correspond to true labels, columns to predicted classes - the matrices are row-normalized. The subset of classes was selected by choosing one class and including others that were the most frequently mutually confused by any of the models.

very few exceptions, this is true for the ImageNet classes we consider. However, it is possible to find cases where this would not be possible due to the different granularities of the classes and Wikipedia pages.

The ZSL methods we use can utilize the information from our textual descriptions effectively, however, they only rely on features from pretrained, off-the-shelf models. Given that the textual de-

scriptions are complex and contain only a sparse amount of visually relevant information, future research is needed to find methods more effective at fully utilizing our textual descriptions. Additionally, there is a potential of using our textual descriptions for different purposes, such as pretraining of image recognition models, similarly to the recent works by Desai and Johnson (2020) and Radford et al. (2021). Finally, the issue of generalization to groups of classes remains an open challenge.

## 7 Conclusions

We demonstrate that simply employing Wikipedia articles as auxiliary data for ZSL on ImageNet yields a state-of-the-art performance, even on a relatively weak model. Additionally, we show that standard ZSL models poorly generalize across categories of classes. Finally, we share the *ImageNet-Wiki* dataset with Wikipedia articles describing corresponding ImageNet classes to facilitate further the research on ZSL, as well as on the interaction between language and vision more generally.

## Acknowledgments

This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

## References

Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. 2016. Synthesized classifiers for zero-shot learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhi Chen, Jingjing Li, Yadan Luo, Zi Huang, and Yang Yang. 2020. Canzsl: Cycle-consistent adversarial networks for zero-shot learning from natural language. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 874–883.

Karan Desai and Justin Johnson. 2020. Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

- P Kingma Diederik, Max Welling, et al. 2014. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, volume 1.
- Mohamed Elhoseiny, Ahmed Elgammal, and Babak Saleh. 2016. Write a classifier: Predicting visual classifiers from unstructured text. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2539–2553.
- Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. 2017. Link the head to the” beak”: Zero shot learning from noisy text description at part precision. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 2*, pages 646–651.
- Michael Matuschek and Iryna Gurevych. 2013. Dijkstra-wsa: A graph-based approach to word sense alignment. *Transactions of the Association for Computational Linguistics*, 1:151–164.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Elisabeth Niemann and Iryna Gurevych. 2011. The people’s web meets linguistic knowledge: automatic sense alignment of wikipedia and wordnet. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 205–214. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. [Learning transferable visual models from natural language supervision](#).
- Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58.
- Bernardino Romera-Paredes and Philip Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.
- Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 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.
- Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10275–10284.
- Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. 2018. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1004–1013.

## A Broader Impact

We anticipate that the dataset we share will make research on multimodal machine learning more accessible, especially to research communities with fewer resources available. More speculatively, we hope that in the future, work on incorporating information from multiple modalities could potentially contribute towards progress on the robustness and flexibility of machine learning models.

We would also like to point out the potential risk of the models reinforcing biases present in sources of text, in particular, in the data we use. The authors of Wikipedia generally represent a very non-diverse social group.<sup>4</sup> Moreover, the content of the database has been shown to contain biases, including the selection of topics it covers.<sup>5</sup> This challenge could potentially be mitigated to some degree with

<sup>4</sup>[https://en.wikipedia.org/wiki/Gender\\_bias\\_on\\_Wikipedia](https://en.wikipedia.org/wiki/Gender_bias_on_Wikipedia) (Accessed: 16 Nov 2020).

<sup>5</sup>[https://en.wikipedia.org/wiki/Wikipedia#Coverage\\_of\\_topics\\_and\\_systemic\\_bias](https://en.wikipedia.org/wiki/Wikipedia#Coverage_of_topics_and_systemic_bias) (Accessed: 16 Nov 2020).

more research progress on interpretable machine learning and bias reduction methods.

## B The effect of text length on accuracy

In Figure 5, we show the relation between the top-5 class accuracy and the length of the textual description of that class. In general, the classes with longer corresponding text tend to yield better accuracy, although the general trend is not strong. However, the impact of the text length appears to be different among different class groups. On animal classes, the relation is negligible. However, on plant classes the relation is much stronger. This observation is especially important since the plant classes generally compose a relatively small fraction of seen classes, and the model has relatively low performance on this group (see Figure 3). This result indicates that using longer textual descriptions can partially mitigate the effect of a group of classes not being well represented in the seen classes.

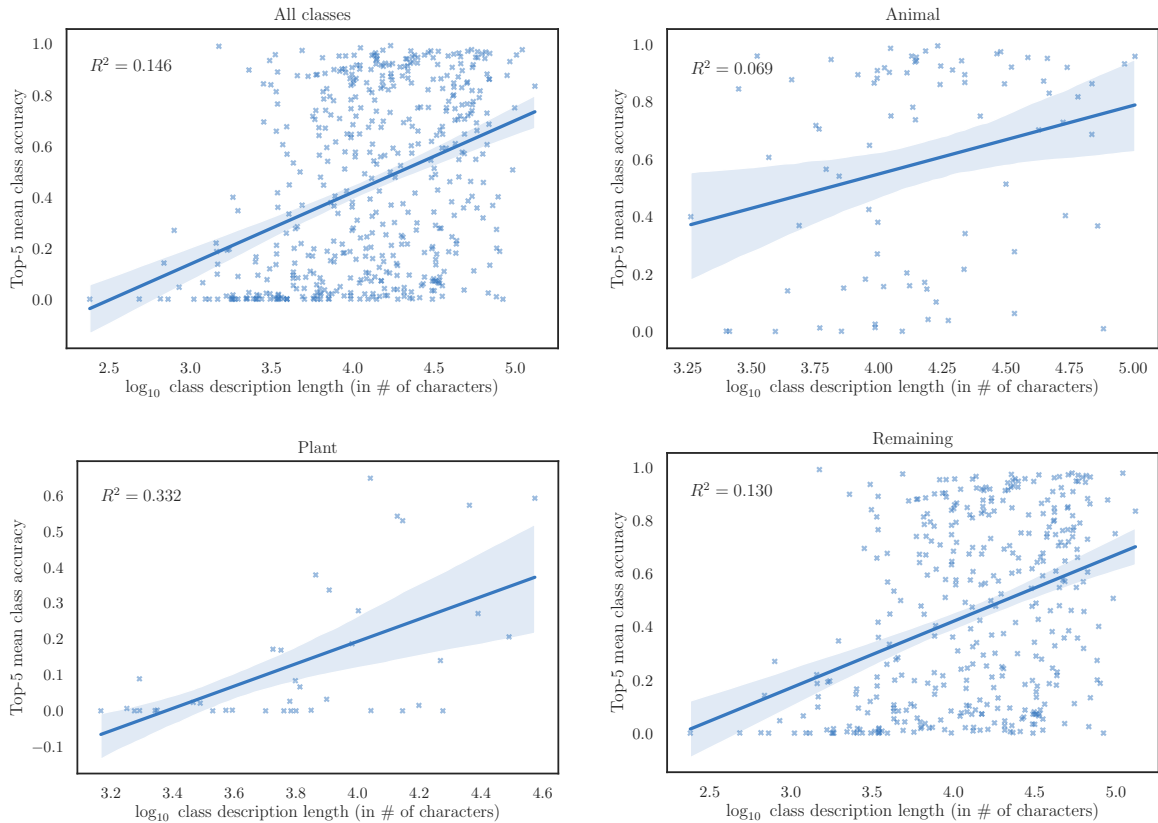


Figure 5: **The relation between the top-5 accuracy on a given class and the length of its Wikipedia pages, separately for different groups of classes from *mp500* test set.** The model used is CADA-VAE trained on the Wikipedia articles using ALBERT-xxlarge features. The lengths are expressed in the  $\log_{10}$  of the number of characters. In case a class has more than one page, we sum the lengths of all pages. Note different ranges of axes.

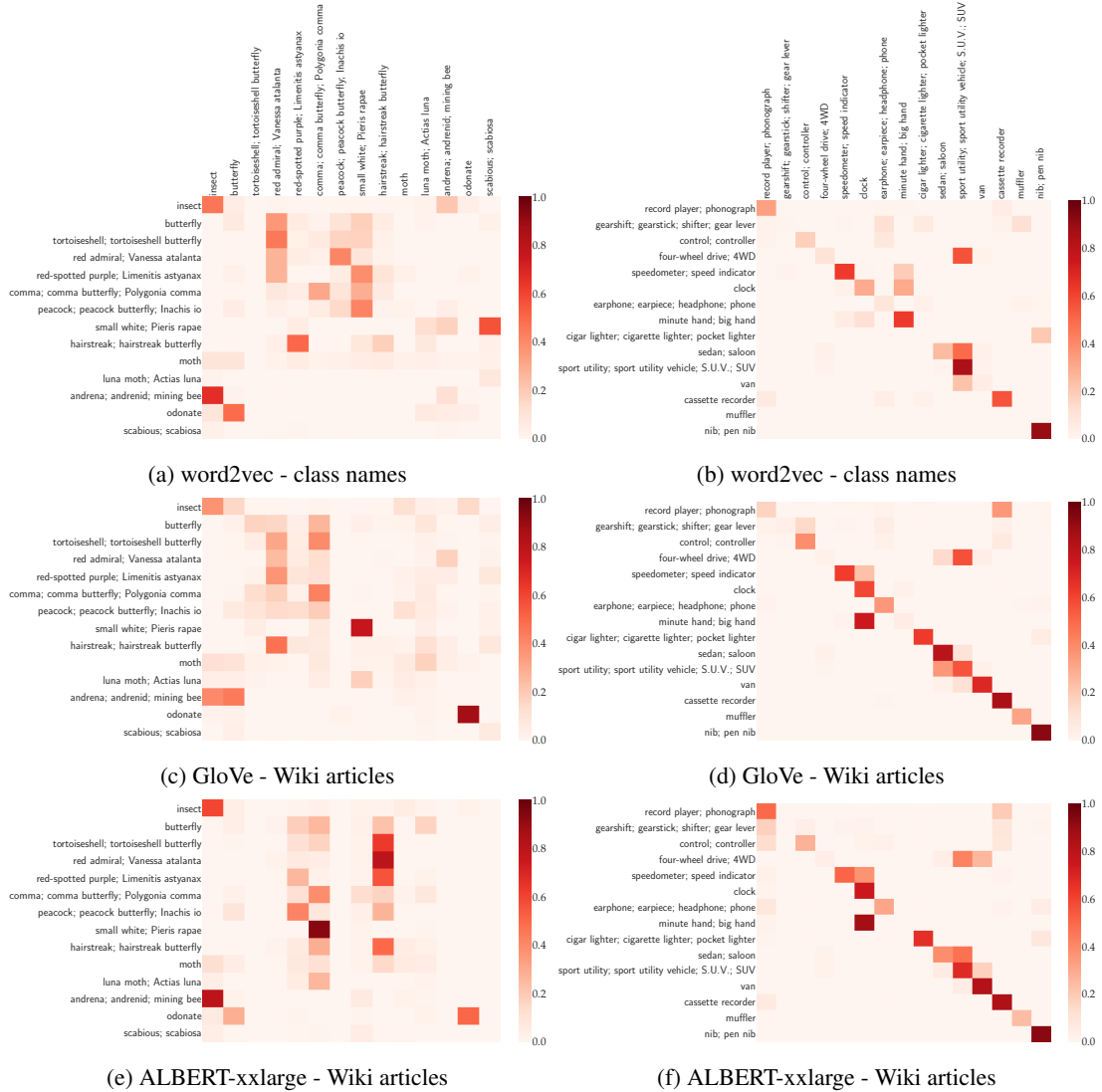


Figure 6: **Subsets of confusion matrices on mp500 test set.** All the models are CADA-VAE (Schonfeld et al., 2019) trained on different types of auxiliary data. The rows correspond to true labels, columns to predicted classes - the matrices are row-normalized. The subset of classes was selected by choosing one class and including others that were the most frequently mutually confused by any of the models. The general performance differs between different subsets of classes - the pattern is less diagonal on the left than on the right column.

## C Additional confusion matrices

In Figure 6, we show confusion matrices for more subsets of classes. On one of the subsets (left), the predictions are generally more noisy. However, the mispredictions generally fall within similar classes (different species of insects on the left side). Additionally, we include the full confusion matrices in the supplementary material as standalone files.

## D Additional across-category generalization results

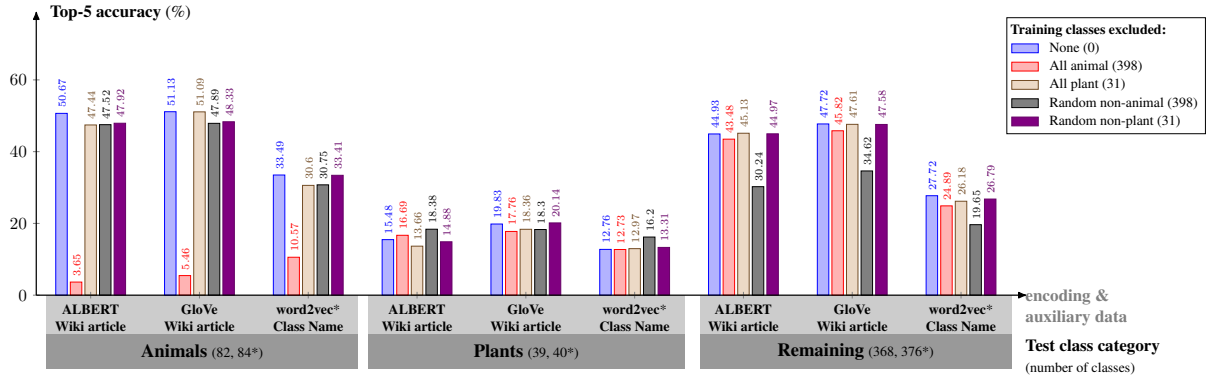
For completeness, in Figure 7, we also show the additional results of models trained on datasets that excluded different groups of classes. The

main trend is, however, similar to the observed in Figure 3. We observe that models perform very poorly on animal classes if the training set excluded animals. Additionally, the performance on plant classes is generally lower - likely due to the smaller set of classes in the training set.

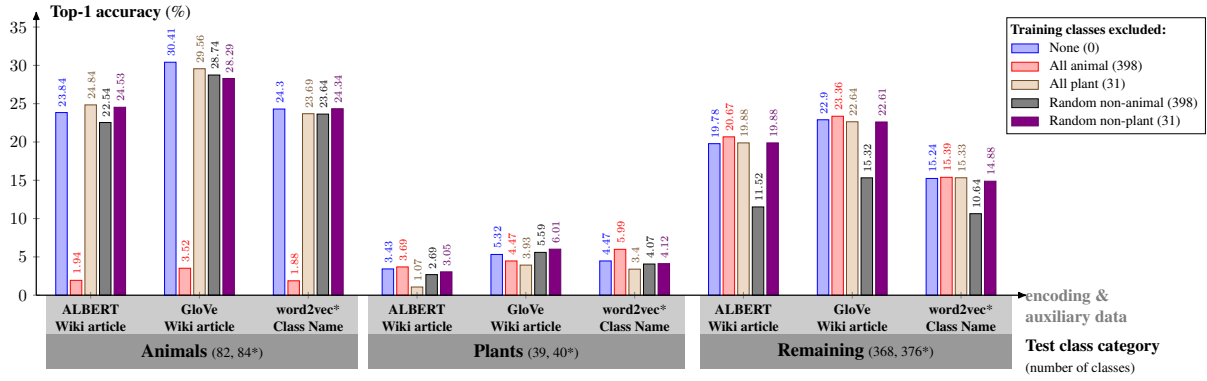
## E Different ALBERT encodings

Table 6 compares different ways to encode the textual descriptions with ALBERT (Lan et al., 2019), such as how to extract features from individual pages and aggregate the page features for classes with more than one corresponding Wikipedia article. ALBERT (Lan et al., 2019) has a special

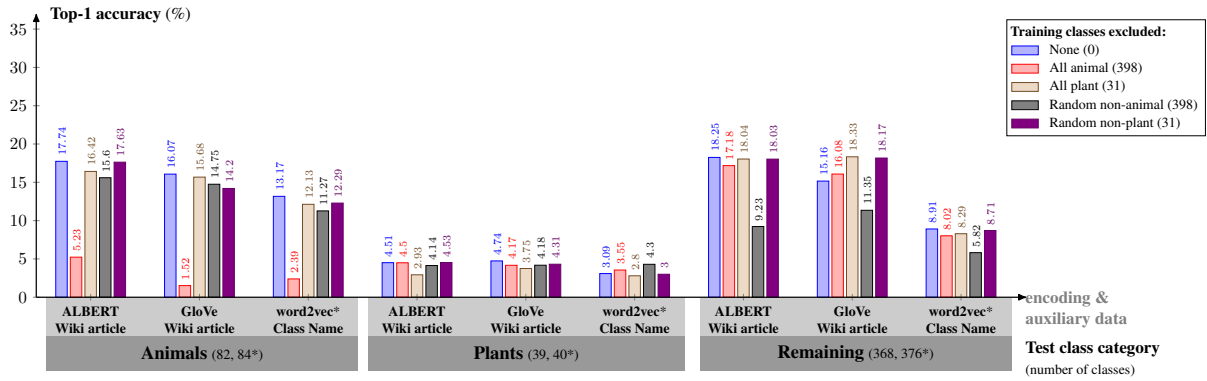




(a) Simple ZSL, Top-5 accuracy



(b) CADA-VAE, Top-1 accuracy



(c) Simple ZSL, Top-1 accuracy

Figure 7: **Within and across category generalization of ZSL.** We show the effect of excluding a category of classes from the training set on ZSL performance for all mp500 test set categories. Specifically we compare excluding all *animal* classes vs. excluding the same number of random *non-animal* classes. We repeat the same process for the *plant* class. Different models trained on different auxiliary data are compared. The performance numbers are from a single run each. The performance on the unseen *animal* classes drops dramatically when the animal classes are removed from the training set, the red bars in the first grouping of plots. The same trend, though not as pronounced, can be seen for the *remaining* classes, the gray bars in the last grouping of plots. The number of test classes is slightly higher for models using class name data (marked with \*) since Wiki articles are missing for some classes.

output used for classification (“[CLS]”), which we additionally evaluate for extracting features. However, averaging features over each tokens seem to

work much better. Additionally, averaging features among multiple pages works best. All other our experiments use averaging both the page represen-

Table 6: **Validation set comparison of different textual inputs and encodings for zero-shot learning on ImageNet.** The reported metrics are mean per class top-1 and top-5 accuracies of the validation classes. 750 *train* classes were used as seen and 250 *val* classes as unseen. All other hyperparameters of Wikipedia models are the same and were chosen based on the performance of the model from the first row. The model using class names had hyperparameters tuned independently. [CLS] represents the special output in ALBERT, typically used for classification tasks.

Auxiliary data	Features	Page rep.	Page aggreg.	Result	
				top-1 (%)	top-5 (%)
Wiki articles	ALBERT (xxlarge)	mean tokens	mean	<b>31.29</b>	<b>64.92</b>
Wiki articles	ALBERT (xxlarge)	mean tokens	sum	29.77	62.44
Wiki articles	ALBERT (xxlarge)	[CLS] output	mean	3.24	12.45
Class names	word2vec	N/A	N/A	27.87	61.91

tations and for aggregating page features (as in row 1, Table 6).

## F Model details

### F.1 Simple ZSL

We individually tune model hyperparameters for three different types of auxiliary data: class names with word2vec features, Wikipedia articles with either ALBERT-base or ALBERT-xxlarge features. We use Adam optimizer and a random search to generate 40 different hyperparameter settings and choose the best setting for each model variant. The ranges of values used for different hyperparameters were as following: batch size  $\in \{32, 128, 256, 512, 1024\}$ , target projection dimension defined by the shapes of  $W_t$  and  $W_x \in \{32, 128, 256, 512, 1024\}$ , margin  $m \in (0.1, 100.)$  or with equal probability  $m = 1, \beta_1$  (for Adam)  $\in \{0.5, 0.9\}$  learning rate  $\in (0.00003, 0.01)$ . The values are sampled uniformly in the log space of the ranges.

The best configurations of hyperparameter values for each model variant are attached in the supplementary material together with the model weights.

### F.2 CADA-VAE

**Implementation details** We use our own implementation of CADA-VAE [35] and verify that it performs similarly to the original. However, we there are two modification that we do. The official implementation of CADA-VAE<sup>6</sup> appears to

incorrectly implement reparameterization trick of Variation Auto-Encoders [11]. It misses the factor of 0.5 when transforming  $\log(\sigma^2)$  into  $\sigma$ . Instead, we use  $\sigma = \exp(0.5 \cdot \log(\sigma^2))$ . Also, unlike the original authors, instead of using a sum to aggregate loss function over samples in a batch, we use a mean. However, this difference should not be important as it can be compensated for by different learning rate or scaling factors of the loss function.

**Hyperparameters** Like for Simple ZSL, the best values of hyperparameters are chosen individually for each of the three different auxiliary data sources. However, for CADA-VAE we use a combination of manual and random search over hyperparameter values.

The random search consisted of 42 runs on each of the model variant using the following ranges: batch size  $\in \{32, 128, 256, 512, 1024\}$ , VAE latent space dimension  $\in \{32, 128, 256, 512, 1024\}$ , VAE image features encoder  $\in \{[1560, 1560], [2048, 1024], [1560], [1024, 512]\}$ , VAE image features decoder  $\in \{[1660], [1024, 2048], [1560], [512]\}$ , VAE auxiliary features encoder  $\in \{[1560, 1560], [2048, 1024], [1560], [1024, 512]\}$ , VAE auxiliary features decoder  $\in \{[1660], [1024, 2048], [1560], [2048], [512]\}$ ,  $\beta$  factor  $\in (0.1, 30)$  or with probability 0.3 fixed  $\beta = 1$ , cross reconstruction loss factor  $\in (0.25, 50.)$ , distributional alignment loss factor  $\in (0.25, 100.)$ . Same as for Simple ZSL, the values were sampled uniformly in the log space of the ranges. The loss function factors were used for linearly increasing loss function coefficients.

<sup>6</sup><https://github.com/edgarschnfld/CADA-VAE-PyTorch>

The schedules for the coefficients were the same as in the original work, that is:  $\beta$  (for scaling VAE KL-divergence loss) was increased from epoch 0 to 93 (or fixed  $\beta = 1$ , cross reconstruction loss from epoch 21 to 75, and distribution alignment from epoch 6 to 22. The rest of the hyperparameters were constant for the random search. We used AMSGrad optimizer, learning rate of VAEs of 0.00015. For the linear classifier we use: learning rate of 0.001, Adam optimizer, batch size of 32 and 200 sampled latent space vectors from each class. Additionally, we tried 58 evaluation runs on the variant using word2vec features, and only a subset of those settings on the models using ALBERT-base (21 runs), or ALBERT-xxlarge (19 runs) features.

The best configurations of hyperparameter values for each model variant are attached in the supplementary material together with the model weights.