TEXT2MODEL: MODEL INDUCTION FOR ZERO-SHOT GENERALIZATION USING TASK DESCRIPTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of generating a training-free task-dependent visual classifier from text descriptions without visual samples. This *Text-to-Model* (T2M) problem is closely related to zero-shot learning, but unlike previous work, a T2M model infers a model tailored to a task, taking into account all classes in the task. We analyze the symmetries of T2M, and characterize the equivariance and invariance properties of corresponding models. In light of these properties we design an architecture based on hypernetworks that given a set of new class descriptions predicts the weights for an object recognition model which classifies images from those zero-shot classes. We demonstrate the benefits of our approach compared to zero-shot learning from text descriptions in image and point-cloud classification using various types of text descriptions: From single words to rich text descriptions.

1 INTRODUCTION

The dominant paradigm for obtaining predictive models in machine learning is inductive training, often using massive labeled datasets. In contrast, people employ other techniques to obtain predictive models. Specifically, they create task-specific discriminative models based on language instructions, such as “separate soft toys from hard ones” or ”collect the furry toy animals” (Markman, 1990). This contrast between machine and human learning is striking, but until now, teaching machines to obtain task-specific discriminative models from natural language descriptions has been limited.

Language-based classification has been studied for the closely related, yet different, task of zero-shot learning from text or attributes (ZSL) (Frome et al., 2013; Lampert et al., 2013). To illustrate the difference, consider a popular family of approaches to ZSL which maps text and images to a shared space (Globerson et al., 2004; Zhang & Saligrama, 2015; Akata et al., 2015; Zhang et al., 2017a; Sung et al., 2018; Xie et al., 2021b; Pahde et al., 2021; Xie et al., 2021a; Radford et al., 2021). Then, images of an unseen concept can be categorized by finding the class whose descriptor is closest to the image in the shared space. The issue is that in this family of approaches the learned representation (and the kNN classifier that it induces) are fixed after training, and are not tuned to a classification task given at inference time. For instance, furry toys would be mapped to the same representation regardless if they are to be distinguished from other toys, or from other furry things. Instead, we wish to produce classifiers and representations that depend on the task provided at inference time. That same limitation is also present with approaches to ZSL that train conditional generative models to synthesize samples from unseen classes at inference time and then use them for kNN classification Elhoseiny & Elfeki (2019); Jha et al. (2021). Other approaches try to address this issue by assuming that test descriptions are known during training, or by training a classifier or generator over generated images at inference time, which is clearly undesirable Xian et al. (2018); Schonfeld et al. (2019).

Here, we describe a novel deep network architecture and a learning workflow to predict a task-dependent, training-free zero-shot model from the language description of the task. This approach has three main benefits. First, it generates a model that is task-dependent and discriminative. This means that parameters of the produced model takes into account the context of other classes in the task. Second, the generated model is not limited to a distance-based classification and can be a more general deep model. Finally, the model is generated as an inference step, without costly training on generated data. To achieve all this, we propose an approach based on hypernetworks (HNs, Ha

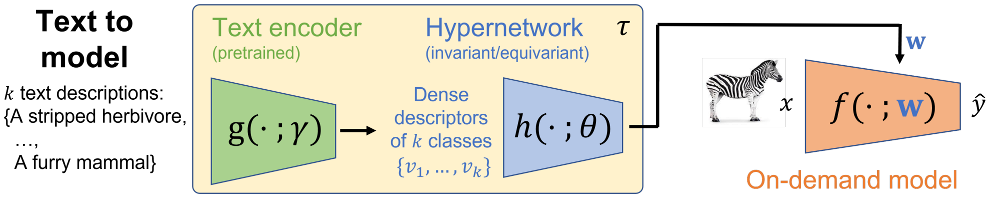

Figure 1: The text-to-model learning problem and our architecture. Our model (yellow box) receives a set of class descriptions as input and outputs weights w for a downstream on-demand model (orange). The model has two main blocks: A pretrained text encoder and a hypernetwork that obeys certain invariance and equivariance symmetries. The hypernetwork receives a set of dense descriptors to produce weights for the on-demand model.

et al., 2016). An HN is a deep network that emits the weights of another deep network (see Figure 1 for an illustration). For the current T2M setup, the HN receives a set of class descriptions and emits a multiclass model that can classify images according to these classes.

Formulating the ZSL problem as a problem of predicting a model from a set of textual descriptions reveals the symmetries of the problem. Such symmetries can be leveraged to design better architectures by making the model invariant or equivariant to group operations over its input-output. This has been shown to provide significant benefits for learning over sets (Zaheer et al., 2017) (Maron et al., 2020) and graphs Herzig et al. (2018); Wu et al. (2020). For the T2M problem studied here, we show that any function that maps a set of descriptions to a multiclass classifier should be equivariant with respect to the output layer of the classifier, and invariant with respect to the earlier layers. Given this characterization, we design an HN architecture that obeys these symmetries, which we call T2M-HN for *text-to-model hypernetwork*. Our architecture is based on the results of Maron et al. (2020), which characterize universal group-equivariant layers.

We test T2M-HN in several classification scenarios. First, we demonstrate its effectiveness in zero-shot classification using data from AwA and SUN. We test two forms of text description: Short descriptions based on class names, and long descriptions generated by a massive language model (Brown et al., 2020). We evaluate our method on ModelNet40 dataset for the task of 3D point-cloud classification, using new class names as text descriptors. Finally, to demonstrate the flexibility of our framework, we perform one class classification with respect to class sets that are described by their attributes. In all these scenarios T2M-HN, outperforms strong methods from previous work.

Recently, large-scale vision-language models like CLIP (Radford et al., 2021) were shown to exhibit fantastic zero-shot capabilities. It is important to highlight a key difference between the CLIP approach and the approach discussed here. Since CLIP was trained on massive multimodal data, it generalizes to new multimodal combinations. In contrast, our approach leverages the semantic compositionality of language models, without requiring paired image-text data. As a result, T2M-HN can be applied directly to domains where there are no massive multimodal data. Indeed, we demonstrate its applicability to 3D point clouds. The downside is that its representation may be sensitive to language and semantic distinctions irrelevant to the visual modality.

To summarize, this paper has three main contributions: (1) We describe a new learning setup: Text-to-model (T2M), producing a standalone model given textual descriptions of the target classes; (2) We characterize the equivariance and invariance properties of T2M models, and describe an architecture, T2M-HN, based on HNs, that obeys the symmetries; and (3) we demonstrate the empirical benefits of T2M-HN in a series of zero-shot tasks, datasets and text descriptions. These include zero-shot image and point-cloud classification, generated using class names, long descriptions and negative descriptions. T2M-HN outperforms SoTa zero-shot approaches in these tasks.

2 RELATED WORK

Zero-shot learning (ZSL). The core challenge in ZSL lies in recognizing unseen classes based on their semantic associations with seen classes. This association can be learned use human-annotated attributes Li et al. (2019); Song et al. (2018); Morgado & Vasconcelos (2017); Annadani & Biswas

(2018). Another source of information for learning the semantic associations is to use textual descriptions. Three main sources were used in the literature to obtain text descriptions of classes: (1) Using class names as descriptions (Zhang et al., 2017a; Frome et al., 2013; Changpinyo et al., 2017; Cheraghian et al., 2022); (2) using encyclopedia articles that describe the class (Lei Ba et al., 2015; Elhoseiny et al., 2017; Qin et al., 2020; Bujwid & Sullivan, 2021; Paz-Argaman et al., 2020; Zhu et al., 2018); and (3) providing per-image descriptions manually annotated by domain experts (Reed et al., 2016; Patterson & Hays, 2012; Wah et al., 2011). These can then be aggregated into class-level descriptions.

Embedding-based zero-shot learning. One popular approach to ZSL is to learn a joint visual-semantic representation, using either attributes or natural text descriptions. Some studies project visual features onto the textual space (Frome et al., 2013; Lampert et al., 2013; Xie et al., 2021b), others learn a mapping from a textual to a visual space (Zhang et al., 2017a; Pahde et al., 2021), and some project both images and texts into a new shared space (Akata et al., 2015; Sung et al., 2018; Zhang & Saligrama, 2015; Xie et al., 2021a; Radford et al., 2021). Once both image and text can be encoded in the same space, classifying an image from a new class can be achieved without furtehr training by first encoding the image and then selecting the nearest class in the shared space. In comparison, instead of nearest-neighbour based classification, our approach is learned in a discriminative way, which may result in richer and potentially stronger models.

Generation-based zero-shot learning. Another line of ZSL studies uses generative models like GANs to generate representations of samples from unseen classes, Such generative approaches have been applied in two settings. Some studies assume they have access to test-class descriptions (attributes or text) during model training. Hence, they can train a classifier over test-class images, generated by leveraging the test-class descriptions (Liu et al., 2018; Schonfeld et al., 2019; Han et al., 2021). Other studies assume access to test-class descriptions only at test time. Hence, they map the test-class descriptions to the shared space of training classes and apply a nearest-neighbor inference mechanism. In this work, we assume that any information about test classes is only available at test time. As a result, ZSL approaches that assume training-time access to information about the test classes are beyond our scope.[1] However, works that assume only test-time access to test-class information form some of our baselines (Elhoseiny & Elfeki, 2019; Jha et al., 2021).

ZSL with large models CLIP (Radford et al., 2021) is based on a contrastive approach for learning image and text representations, and was trained on a large corpus of 400 million image-text pairs by maximizing the similarity of correct text-image pair embeddings. CLIP is highly effective for vision and language zero-shot tasks, as demonstrated, e.g., by its ImageNet classification accuracy of 75% when no supervision from ImageNet is available Radford et al. (2021). With that said, CLIP was probably exposed to images from all ImageNet classes during its training, so is not really zero-shot classifier. In contrast, our approach can be applied directly to domains where there are no massive paired multimodal data. We demonstrate this using experiments with a 3D point cloud dataset.

3 METHOD

We first formalize the text-to-model (T2M) learning problem and then present our approach.

3.1 PROBLEM FORMULATION

We describe the problem of text-to-model in the context of multiclass classification. It can be naturally extended to regression and ranking problems. Here, our objective is to learn a mapping τ from a set of k natural language descriptions into the space of k-class classifiers. Here, we address the case where the architecture of the downstream classifier is fixed and given in advance, but this assumption can be relaxed (as in Litany et al., 2022).

Formally, let $S^k = \{s_1, \ldots, s_k\}$ be a set of k class descriptions drawn from a distribution $\mathcal{P}_k$, where s_j is a text description of the jth class. The distribution $\mathcal{P}_k$ can be characterized by a two-

---

[1] While these algorithms could in principle be re-trained when new classes are presented at test-time (e.g. in a continual learning (Ring et al., 1994) setup), this would result in costly and inefficient inference mechanism, and possibly also in catastrophic forgetting (McCloskey & Cohen, 1989) of previous class information. We hence do not include them in our experiments.

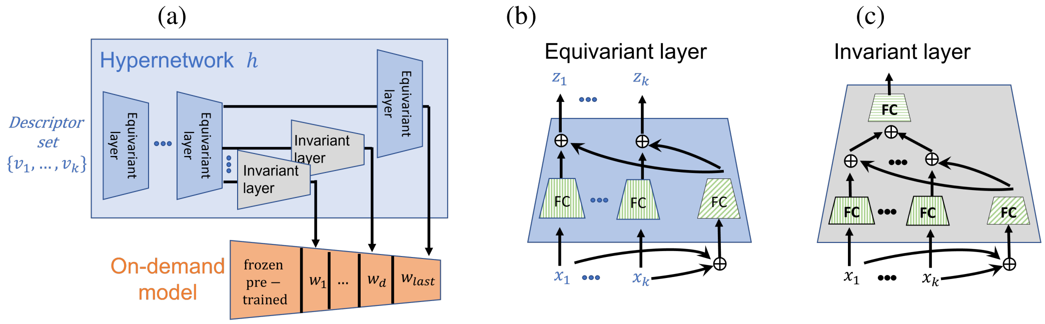

Figure 2: (a) The T2M-HN architecture for equivariant-invariant hypernetwork. The input is processed by equivariant layers, followed by a prediction head for each layer of the target on-demand classifier f. The prediction head for W_{last} is equivariant. Heads for earlier layers of f, $w_1, \ldots w_k$ are invariant. (b) An architecture for the equivariant layer. Every input is processed by a fully connected (FC) layer in a Siamese manner (shared weights). Inputs are also summed and processed by a second FC layer, whose output is added back to each output. (c) An architecture for an invariant layer, following a similar structure to b.

stage process: First, a set of k classes is drawn from a large set of classes. Then, a text description is drawn for each class.

Let τ be a T2M model parameterized by a set of parameters ϕ. It takes the text descriptors and produces a set of parameters W of a k-class classification model $f(\cdot; W)$. Therefore, we have $\tau_\phi : \{s_1, \ldots s_k\} \rightarrow \mathbb{R}^d$, where d is the dimension of W, that is, the number of parameters of the classification model $f(\cdot; W)$, and we denote $W = \tau_\phi(S^k)$.

Let $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ be a loss function, and let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ be a labeled dataset from a distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$. For k-class classification, $\mathcal{Y} = \{1, \ldots, k\}$. We can explicitly write the loss in terms of ϕ as follows. $l\left(y_i, \hat{y_i}\right) = l\left(y_i, f(x_i; W)\right) = l\left(y_i, f(x_i; \tau_\phi(S^k))\right)$. See also Figure 1 and note that $\tau = h \circ g$.

The goal of T2M is to minimize

$$\phi^* = \arg\min_\phi \mathbb{E}_{S^k \sim \mathcal{P}^k} \mathbb{E}_{(x,y) \sim \mathcal{P}} \left[l\left(y, f(x; \tau_\phi(S^k))\right) \right] . \tag{1}$$

The training objective becomes $\phi^* = \arg\min_\phi \sum_j \sum_i l\left(y_i, f(x_i; \tau_\phi(S^{k_j}))\right)$,where the sum over j means summing over all sets of descriptions in the training set.

3.2 ARCHITECTURES AND SYMMETRIES OF TEXT-TO-MODEL HYPERNETWORKS

We propose a model with an architecture consisting of two components (see Figure 1). A text encoder g that takes natural language descriptions and transforms them into dense descriptors, and a hypernetwork h that takes these dense descriptors and emits weights for a downstream classifier. In this paper, we do not impose any special properties on the text encoder g. It can be any model trained using language data (no need for multi-modal data). Interestingly, the text-to-model setup imposes certain invariance and equivariance properties that the HN should obey. We now discuss these properties and then derive the proper architecture that captures these properties (See Figure 2).

EQUIVARIANCE PROPERTIES OF THE CLASSIFIER LAYER.

Consider a downstream multiclass deep classifier. Its last (classification) layer has a weight vector $w_i \in \mathbb{R}^m$ for every output class $W_{last} = \{w_1, \ldots, w_k\}$ (See Figure 2a). Importantly, the HN receives k class descriptors and outputs their corresponding weights

$$W_{last} = \{w_1, \ldots, w_k\} = R_{last}(\tau_\theta\left(\{s_1, \ldots, s_k\}\right)), \tag{2}$$

where R_{last} is a function that takes the output of τ and resizes the last $k * m$ elements to the matrix W_{last}. The HN must therefore be equivariant to permutations over its input. If the input descriptions are permuted by a permutation $\mathcal{P}$ the columns of the last layer weight should be permuted accordingly:

$$\mathcal{P}(f(x;\tau_\phi(S^k)) = f(x;\tau_\phi(\mathcal{P}(S^k)). \tag{3}$$

INVARIANCE PROPERTIES OF INTERMEDIATE LAYERS

Now consider the layer of the downstream classifier that is just before the last layer. A similar argument holds for earlier (lower) intermediate layers. We want the HN to be equivariant to permutations over its inputs by design. We now show that using an equivariant transformation for the last layer and an invariant transformation for the penultimate layer is sufficient to ensure that this requirement is met.

Theorem 3.1. *Let f be a two-layer neural network $f(x) = W^{last}\sigma(W^{pen}x)$, whose weights are predicted by τ $[W^{last}, W^{pen}] = \tau(S^k)$. If $\tau(S^k)$ is equivariant to a permutation $\mathcal{P}$ with respect to W^{last}, and invariant to $\mathcal{P}$ with respect to W^{pen}, then $f(x)$ is equivariant to $\mathcal{P}$ with respect to the input of $\tau(S^k)$.*

See a formal proof in the Supplemental Section B.

3.3 INVARIANT AND EQUIVARIANT ARCHITECTURES

The equivariance property requires that we share parameters such that the same parameters process those elements that we want to be invariant to any permutation (Wood & Shawe-Taylor, 1996; Ravanbakhsh et al., 2017). In our case, since we aim to be equivariant to possible permutations of input descriptions, we must use the same parameters to process all inputs.

Figure 2(b) shows the architecture of our equivariant layers. All inputs are fed into the same fully connected layer (vertical stripes). To take into account the context of each input, we sum all the inputs to obtain a context vector. We fed the context vector to a different fully connected layer (diagonal stripes) and add it to each one of the processed inputs. The same architecture holds for the invariant layer (Figure 2(c)), but with additional summation over all equivariant outputs and another different fully connected layer (horizontal stripes).

Our HN uses several equivariant layers to process the input descriptions. We then used one prediction head for each layer of the output model. The last layer should be equivariant, so we use an equivariant prediction head. For the hidden layers, we use invariant layers (See Figure 2(a)).

4 EXPERIMENTS

The T2M setup is about producing a standalone model that can be applied to data from new classes. Accordingly, the model trains on data from a set of training classes, alongside their text descriptions. Then, it is tested on data from new classes, given the text descriptions of these classes.

We evaluate T2M-HN in zero-shot classification, using two image datasets, AwA and SUN, and one 3D point cloud dataset, ModelNet40. We consider several forms of text description, including single-word class labels (for AwA and ModelNet 40), few-word class names (for SUN) and longer descriptions that could also include negative properties (i.e. properties that the images in the class do not have). Finally, we study one-class classification based on text attributes.

Baselines We compare our T2M-HN with four text-based zero-shot approaches for image recognition: (1) DEVISE (Frome et al., 2013) projects images to a pre-trained language model space by adding a projection head to a pre-trained visual classification model; (2) Deep Embedding Model (DEM) (Zhang et al., 2017b) uses the visual space as the shared embedding space; (3) CIZSL (Elhoseiny & Elfeki, 2019) trains conditional GANs with a modified loss designed to make them generate samples from unseen classes without synthesizing unrealistic images. At inference time, the GAN is conditioned on test descriptions, generates synthetic image representations and test images are classified using kNN w.r.t. to the synthetic images.; and (4) GRaWD (Jha et al., 2021): Similar in spirit to CIZSL, GRaWD trains a conditional GAN with a loss that is based on a random walk which reaches regions in space that are hard to classify as seen classes.

When relevant, we also computed the performance obtained when using the CLIP encoder followed by k-NN classification in the CLIP space (Radford et al., 2021). Since CLIP was trained on 400

	AWA by class name			ModelNet40 by class name		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
DeViSE (Frome et al., 2013)	78.1 ± 1.0	58.9 ± 1.4	67.2 ± 1.9	78.9 ± 1.3	63.9 ± 2.8	70.6 ± 3.1
DEM (Zhang et al., 2017b)	83.1 ± 1.6	75.1 ± 1.2	78.9 ± 2.0	81.3 ± 1.6	59.1 ± 3.7	68.4 ± 4.0
CIZSL (Elhoseiny & Elfeki, 2019)	97.0 ± 0.1	74.7 ± 3.2	84.20 ± 2.0	97.6 ± 0.1	72.9 ± 3.6	83.4 ± 2.4
GRaWD (Jha et al., 2021)	96.9 ± 0.1	81.6 ± 1.9	88.6 ± 1.1	97.5 ± 0.1	73.5 ± 3.2	83.9 ± 2.1
T2M-HN (ours)	$\mathbf{98.9 \pm 0.1}$	$\mathbf{87.3 \pm 0.2}$	$\mathbf{92.7 \pm 0.1}$	$\mathbf{98.5 \pm 0.1}$	$\mathbf{74.5 \pm 0.4}$	$\mathbf{84.8 \pm 0.3}$
CLIP (Radford et al., 2021)	98.9 ± 0.2	NA	NA	NA	NA	NA

Table 1: **Classification by single-word class names**. Mean classification accuracy on seen and unseen classes for AWA and ModelNet-40. Values are averages and SEM over all class pairs.

million pairs of images and their captions, it is likely to assume it has seen all classes studied here. It is therefore not a zero-shot classifier and the results can be viewed as a "skyline" value that zero-shot approaches should aim at.

Datasets: We experiment with two image datasets and a 3D pointcloud dataset: **(1) Animals with attributes** (AWA, Lampert et al., 2009), consisting of 30,475 images of 50 animal classes, divided into 40 seen training classes and ten unseen test classes; (2) SUN (Patterson & Hays, 2012), consisting of 131,072 images of 397 environmental scenes and places; and **(3) ModelNet40** (Wu et al., 2015), a 3D point cloud dataset, consisting of 12,311 CAD models from 40 categories. Our class splits are given in the Supplemental Section D for reproduciblity. Full data splits will be provided with our code.

Implementation and architecture We encode single-word class names using Glove (Pennington et al., 2014) and longer descriptions using SBERT (Reimers & Gurevych, 2019). For images, the visual target model had a backbone based on a frozen ResNet-18 (He et al., 2016), pretrained on ImageNet with one or two fully connected layers, predicted by the HN. For 3D point-cloud data, the backbone was PointNet (Qi et al., 2017), again with one or two predicted fully-connected layers.

Experimental protocol: We split the data in two dimensions: Classes and samples. Classes are split into *seen classes* used for training and *unseen classes* used in evaluation. For each seen class we split out a set of evaluation images that are not presented during training, but used to evaluate the model on the seen classes. Specifically, for this set in ModelNet40 we use the original test split in each class. Similarly, for AwA and SUN we randomly selected 10% of images for "seen" evaluation. In summary, "Seen" in our tables means novel images from seen classes.

Workflow: When training the whole architecture, we split the train (seen) classes in a 80:20 manner. 80% of the classes were used for training the backbone (ResNet for images and PointNet for point clouds). Then, we froze the weights of the backbone and used all the training classes to train the HN. This way, the HN learns to generalize to classes the backbone did not see. Finally, we evaluate the entire architecture on the evaluation split of the seen classes, and on the test (unseen) classes.

At test time, the model receives k descriptions of unseen classes and predicts a standalone model to classify images drawn from the corresponding k classes. Unless otherwise specified, we experiment with the value of $k = 2$.

4.1 ZERO-SHOT USING CLASS NAMES AS DESCRIPTIONS

In the following experiment, we evaluate T2M-HN under two tasks: Zero-shot image classification and zero-shot 3D point cloud classification. We use single-word class names for both tasks as the textual class descriptions.

Results: Table 1 shows the average classification accuracy of all participating models. Our model reaches the highest results in both setups and datasets, outperforming the strongest baseline by 5.7% and 1% on the unseen classes in AwA and ModelNet40, respectively. These results demonstrate our model's effectiveness in learning to generate powerful classifiers for unseen (and seen) classes.

	SUN by short description			AWA by GPT descriptions		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
DeViSE (Frome et al., 2013)	52.0 ± 1.4	58.9 ± 1.1	55.2 ± 0.9	68.0 ± 1.2	55.7 ± 0.7	61.2 ± 0.6
DEM (Zhang et al., 2017b)	83.2 ± 1.1	83.2 ± 1.4	83.2 ± 0.9	78.2 ± 1.7	69.1 ± 1.6	73.4 ± 2.3
CIZSL (Elhoseiny & Elfeki, 2019)	94.0 ± 0.1	80.3 ± 0.6	86.6 ± 0.3	96.6 ± 0.1	80.7 ± 2.2	87.9 ± 1.3
GRaWD (Jha et al., 2021)	95.5 ± 0.1	84.7 ± 0.5	89.8 ± 0.3	96.8 ± 0.1	81.1 ± 0.2	88.3 ± 1.2
T2M-HN (ours)	$\mathbf{95.8 \pm 0.1}$	$\mathbf{88.4 \pm 0.1}$	$\mathbf{92.0 \pm 0.1}$	$\mathbf{98.7 \pm 0.1}$	$\mathbf{83.3 \pm 0.1}$	$\mathbf{90.3 \pm 0.1}$
CLIP (Radford et al., 2021)	99.1 ± 0.4	NA	NA	93.7 ± 0.2	NA	NA

Table 2: **Classification by class descriptions**. Mean classification accuracy and SEM on images from seen and unseen classes in SUN and AwA. Averages are over 100 random class pairs (for SUN) and all class pairs (for AwA).

4.2 ZERO-SHOT USING TEXT DESCRIPTIONS

Next, we evaluate our model's performance when using more complicated text descriptions: **(1) For SUN**, we use short class descriptions provided by the original dataset. Specifically, SUN includes many multi-word class names like "parking garage indoor" or "control tower outdoor". These may not be coded well by a single-word embedding model such as Glove, so we treated them as word sequences and encoded them with SBERT. **(2) For AwA**, we use synthetic class descriptions generated by a GPT model. Specifically, we used GPT3 (Brown et al., 2020) to generate five different descriptions for each class of AwA. During training and evaluation, we randomly choose one description for each class in the batch, from its corresponding 5 class descriptions. See detailed examples in the Supplementary Section C. We will publish the full set of descriptions for reproducibility.

Results: Table 2 presents the classification accuracy obtained using class descriptions, for T2M-HN and the ZSL-baselines. Across the unseen classes, T2M-HN exceeds all baselines by at least 3.7% for SUN and 2.2% for AwA. This suggests that our T2M-HN model is useful across diverse scenarios, consisting of various different textual and visual inputs.

4.3 DESCRIPTIONS WITH NEGATIVES

To this point, we have assumed that the descriptions correspond to properties of the class. However, descriptions could also state which properties the class does **not** have. For example, one may want to classify animals that "do not live in the water", "have no tail", or animals that "do not fly".

To create such negative descriptions for the AwA data, we considered each class and randomly sampled four attributes that do not exist in that class. For example, an elephant may be described as an "Animal that does not: fly, small, furry or white". Table 3 presents our results. We tested two scenarios: Considering only negative descriptions (left side of the table) and including equal portions of positive and negative descriptions (right side of the table). For both scenarios, we keep the same ratio of positive and negative descriptions for training and testing.

Results: Table 3 shows that T2M-HN achieves an average of 77.1% and 82.9% in zero-shot classification (for fully-negative and half-negative setups, respectively), outperforming all baselines by significant gaps. We hypothesize that our most substantial baseline, which generates image features from the textual descriptions, fails at generating an image when the available information is what the image does not contain. Additionally, CLIP performance degrades dramatically in these setups, from 93.7% accuracy to 19.9% and 56.8% (for both setups, respectively). The reason is probably that the CLIP training set consisted of image captions, which rarely contain negative descriptions.

4.4 BEYOND INDIVIDUAL CLASSES

The above experiments focused on recognizing novel but predefined classes. One may be interested in semantic distinctions that are more general than single class. For instance, one may want to recognize *"animals with fur"* or "indoor scenes". We now show that our T2M-HN framework is flexible enough to handle such distinctions.

To evaluate richer semantic descriptions, we created a set of one-class classification tasks designed to recognize images based on properties which cut through class boundary. To create these tasks,

	AwA by negative descriptions			AwA by negative and positive descriptions		
AWA data	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
DEM (Zhang et al., 2017b)	81.7 ± 1.2	73.7 ± 1.6	77.5 ± 1.0	78.2 ± 1.7	69.1 ± 1.6	73.4 ± 1.2
CIZSL (Elhoseiny & Elfeki, 2019)	58.3 ± 0.8	56.6 ± 3.4	57.5 ± 1.8	93.9 ± 0.2	71.6 ± 2.3	81.2 ± 1.5
GRaWD (Jha et al., 2021)	54.9 ± 0.8	56.0 ± 3.2	55.3 ± 1.6	95.0 ± 0.2	73.9 ± 2	83.2 ± 1.5
T2M-HN (ours)	$\mathbf{90.0 \pm 0.2}$	$\mathbf{77.1 \pm 0.3}$	$\mathbf{83.0 \pm 0.2}$	$\mathbf{96.6 \pm 0.2}$	$\mathbf{82.9 \pm 0.2}$	$\mathbf{89.2 \pm 0.1}$
CLIP	19.9 ± 2.2	NA	NA	56.8 ± 2.9	NA	NA

Table 3: **Classification with negative descriptions**. Mean classification accuracy on images from seen and unseen AwA classes. Values are averages over all class pairs. CLIP has seen all classes and therefore is marked as NA for unseen classes.

	AwA Super Sets		
	Seen	Unseen	Harmonic
DeViSE (Frome et al., 2013)	53.0 ± 1.9	50 ± 0.6	51.5 ± 0.9
DEM (Zhang et al., 2017b)	50.1 ± 1.4	48.3 ± 1.8	49.2 ± 1.6
CIZSL (Elhoseiny & Elfeki, 2019)	57.3 ± 5.6	50.2 ± 5.8	55 ± 4
GRaWD (Jha et al., 2021)	59.8 ± 3.5	51.6 ± 4.8	55.3 ± 3.1
T2M-HN (ours)	$\mathbf{67.2 \pm 5.2}$	$\mathbf{57.3 \pm 5.7}$	$\mathbf{61.9 \pm 5.4}$

Table 4: Average Area under the Recall-Precision curve per attribute. For each attribute, we evaluate the AUC of the classifier and report AUC and SEM across all the attributes

we used attributes from the AwA dataset. We eliminated non-visual attributes (like habitats and diet), leaving a total of 53 attributes, which we split to 30 train attributes, 10 validation and 13 test attributes. When constructing a set of test images, we also take into account the data split over classes (training classes and test classes). These two splitting operations induce the following sets of images: (1) *Training images* from training attributes and training classes, used to train the hypernetwork; (2) *Validation images* from the training classes, with the validation attributes used to tune hyperparameters; (3) *Test images from seen classes*, new images of test attributes, whose class was seen during training (but not the specific images); and (4) *Test images from unseen classes* new images of test attributes, whose class was not seen during training. We report the average Area under the Recall-Precision curve over seen (group (3)) and unseen classes (group (4)).

Results: Table 4 shows that T2M-HN captures well complex semantic distinctions. We attribute this to its ability to draw new classifiers for each new textual description. This ability is acquired during training, drawn from the classification task objective, which propagates to the HN weights.

4.5 T2M-HN CLASSIFIERS DEPEND ON TASK CONTEXT

Current leading text-based ZSL methods map class descriptions or images to a shared representation, but that mapping does not take into account the classification task, namely, the mapping does not depend on the other "on-demand" classes that the model should distinguish. Our T2M-HN is designed to use that information, because it generates a classifier for a given set of classes.

To demonstrate this effect, we use GradCam (Selvaraju et al., 2017) and examine what image areas are used in different classification tasks. Figure 3 explores two such examples. The left three panels show the image regions that are used for classifying the image as a *Dolphin* when the negative class is *Dear*. When classifying dolphin vs. deer, the model gives most of its weight to the background (ocean water and waves), which is reasonable since an image of a deer probably will not contain those elements in the background. However, when classifying dolphin vs. killer whale, the model gives most of its weight to the dolphin itself, since the background of a dolphin image may be similar to the background of a whale image.

4.6 ABLATION

To evaluate the effect that equivariance has on HN performance, we compared HN with and without equivariant design. We repeat the experiment for an on-demand models with one or two fully connected layers. Table 5 gives the mean accuracy of the following variants: **(1) T2M-HN 1-layer**

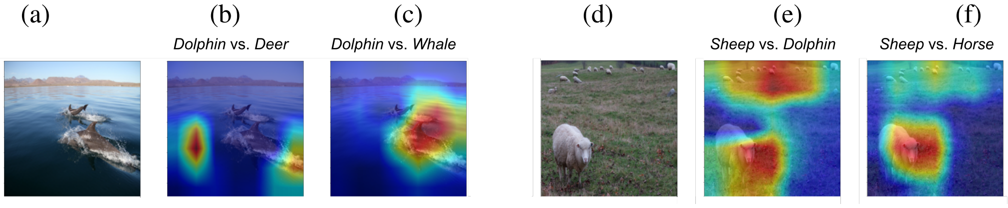

Figure 3: Class context affects the classifier learned. Each triplet of images compares GradCAM heat maps for the same image, but with two different models. **(a-c) Dolphin:** panel (b) is for classifying dolphin vs deer and panel (c) is for classifying dolphin vs killer whale. In b, the model is strongly affected by background ocean water, presumably because the negative class that lives on land . However, when classifying dolphin vs. killer whale (c), the model attends to the dolphin itself, since the background would be similar for both classes. **(d-f) Sheep** A similar effect demonstrated for sheep against Dolphin and Horse.

An equivariant HN that predicts one equivariant FC layer for the on-demand model. **(2) 1-layer w.o. EV** An FC HN that predict one fully connected layer to the on-demand model. **(3) T2M-HN 2-layers** An equivariant HN that predicts two FC layers for the on-demand model: The first is invariant and the second is equivariant. **(4) 2-layer w.o. EV** An FC HN that predicts two FC layers for the on-demand model.

	AWA by class name			AWA by GPT descriptions			ModelNet40 class names		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
T2M-HN 1-layer	**98.9 ± .1**	**87.3 ± .2**	**92.7 ± .1**	**98.7 ± .1**	**83.3 ± .1**	**9.3 ± .1**	**98.5 ± .1**	74.5 ± .4	84.8 ± .3
1-layer w.o. EV	98.6 ± .1	85.9 ± .2	91.8 ± .1	97.0 ± .1	78.2 ± .1	86.6 ± .1	98.1 ± .1	7.1 ± 1.0	81.8 ± .7
T2M-HN 2-layers	98.0 ± .1	87.2 ± .1	92.3 ± .1	96.4 ± .1	76.7 ± .1	85.4 ± .1	95.5 ± .1	**77.7 ± .7**	**85.7 ± .4**
2-layers w.o. EV	98.6 ± .1	75.7 ± .1	85.5 ± .1	74.0 ± .3	57.4 ± .1	64.6 ± .2	97.9 ± .1	74.9 ± .7	84.9 ± .5

Table 5: **Ablation study**. Mean classification accuracy on seen and unseen classes and their harmonic mean for AWA and ModelNet40 datasets. Values are averages over all classes pairs. T2M-HN is the proposed method designed with equivariant and invariance properties. We evaluate two variants of T2M-HN, one that produces a single FC layer to the on-demand model, and a second variant that produces two FC layers. To demonstrate the importance of the equivariant design, we evaluate an HN that produce 1 and 2 layers without equivariant design.

For AwA, the single equivariant layer performs better than one invariant and one equivariant. We believe this is because ResNet is trained with only one fully connected layer, so the ResNet features are linearly separable. For ModelNet40, we used PointNet (which ends with 3 fully connected layers) as a features extractor and in that case, the Inv+EV architecture generalizes better to unseen classes. Since we use the accuracy over the seen classes to chose the model architecture, in the paper we report the score of the EV hypernetwork.

5 CONCLUSION

We presented T2M, a learning setup in which a discriminative model is obtained ”on demand”, given only class descriptions at test-time. We analyzed the group symmetries that a T2M model should obey, and characterized the proper invariance and equivariance properties that ensure these symmetries. We then proposed T2M-HN, a deep architecture based on hypernetworks, which addresses the T2M setup and obeys the required symmetries. Next, we evaluated our T2M-HN approach in a series of recognition tasks, considering image and 3D point cloud data. We experiment with descriptions at varying complexity: From single-word class names, through few-word class names and long text descriptions, all the way to “negative” descriptions. Recently, multi-modal models like CLIP has shown fantastic compositional capabilities, achieved through training on massive paired data. While our T2M approach leverages compositionality in language space without massive paired data, it still succeeds to generalize to new classes and new distinctions better than existing ZSL methods.

6 ETHICS STATEMENT

This paper proposes a new technique to generate models on-demand at inference time from text descriptions. As with other learned predictive models, the approach may be susceptible to biases found in the training data, and produce classifiers that are by themselves biased in unexpected ways.

7 REPRODUCIBILITY STATEMENT

Our work makes the following effort to ensure reproducibility: (1) We will release our code, and data splits. (2) We provide details on hyperparameter choices for both our training and evaluation setups in the Supplemental Section A. (3) Supplemental Section C provides examples of long text descriptions generated using GPT3, and the full set of descriptions will be published with the code. (4) Supplemental section D provides the class splits we used for SUN and ModelNet40.

REFERENCES

Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2927–2936, 2015.

Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7603–7612, 2018.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *NeurIPS 2020*, 2020.

Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. *arXiv preprint arXiv:2103.09669*, 2021.

Soravit Changpinyo, Wei-Lun Chao, and Fei Sha. Predicting visual exemplars of unseen classes for zero-shot learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 3476–3485, 2017.

Ali Cheraghian, Shafin Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3d point cloud objects and beyond. *International Journal of Computer Vision*, pp. 1–21, 2022.

Mohamed Elhoseiny and Mohamed Elfeki. Creativity inspired zero-shot learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 5783–5792. IEEE, 2019.

Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5640–5649, 2017.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Advances in neural information processing systems*, 17, 2004.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Zongyan Han, Zhenyong Fu, Shuo Chen, and Jian Yang. Contrastive embedding for generalized zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 2371–2381. Computer Vision Foundation / IEEE, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. *Advances in Neural Information Processing Systems*, 31, 2018.

Divyansh Jha, Kai Yi, Ivan Skorokhodov, and Mohamed Elhoseiny. Imaginative walks: Generative random walk deviation loss for improved unseen learning representation. *arXiv preprint arXiv:2104.09757*, abs/2104.09757, 2021.

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 951–958. IEEE, 2009.

Christoph H 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.

Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE international conference on computer vision*, pp. 4247–4255, 2015.

Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3583–3592, 2019.

Or Litany, Haggai Maron, David Acuna, Jan Kautz, Gal Chechik, and Sanja Fidler. Federated learning with heterogeneous architectures using graph hypernetworks. *arXiv preprint arXiv:2201.08459*, 2022.

Shichen Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Generalized zero-shot learning with deep calibration network. *Advances in neural information processing systems*, 31, 2018.

Ellen M Markman. Constraints children place on word meanings. *Cognitive science*, 14(1):57–77, 1990.

Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pp. 6734–6744. PMLR, 2020.

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

Pedro Morgado and Nuno Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6060–6069, 2017.

Frederik Pahde, Mihai Puscas, Tassilo Klein, and Moin Nabi. Multimodal prototypical networks for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2644–2653, January 2021.

Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758. IEEE, 2012.

Tzuf Paz-Argaman, Yuval Atzmon, Gal Chechik, and Reut Tsarfaty. Zest: Zero-shot learning from text descriptions using textual similarity and visual summarization. *arXiv preprint arXiv:2010.03276*, 2020.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Pengda Qin, Xin Wang, Wenhu Chen, Chunyun Zhang, Weiran Xu, and William Yang Wang. Generative adversarial zero-shot relational learning for knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8673–8680, 2020.

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

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *International conference on machine learning*, pp. 2892–2901. PMLR, 2017.

Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 49–58, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Mark Bishop Ring et al. Continual learning in reinforcement environments. 1994.

Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8247–8255, 2019.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1024–1033, 2018.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

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

Jeffrey Wood and John Shawe-Taylor. Representation theory and invariant neural networks. *Discrete applied mathematics*, 69(1-2):33–60, 1996.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5542–5551, 2018.

Cheng Xie, Hongxin Xiang, Ting Zeng, Yun Yang, Beibei Yu, and Qing Liu. Cross knowledge-based generative zero-shot learning approach with taxonomy regularization. *Neural Networks*, 139:168–178, 2021a.

Guo-Sen Xie, Xu-Yao Zhang, Yazhou Yao, Zheng Zhang, Fang Zhao, and Ling Shao. Vman: A virtual mainstay alignment network for transductive zero-shot learning. *IEEE Transactions on Image Processing*, 30:4316–4329, 2021b.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2021–2030, 2017a.

Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2021–2030, 2017b.

Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pp. 4166–4174, 2015.

Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1004–1013, 2018.

Supplemental information

A HYPERPARAMETER OPTIMIZATION

We tune the hyperparameters a held-out set described below.

For the HN optimizer, we tuned the learning rate $\in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, momentum $\in \{0.1, 0.3, 0.9\}$, weight decay $\in \{0.00001, 0.0001, 0.001, 0.1\}$, number of HN training epochs $\in \{50, 70, 100\}$.

For the on-demand target model we fixed the optimizer to have a learning rate of 0.01, momentum of 0.9 and weight decay of 0.01. We tuned the batch size $\in \{16, 32, 64, 128\}$ and the number of training epochs $\{1, 2, 3, 5, 10\}$.

For the HN architecture with only one hidden layer, we tried several sizes for that layer $\{30, 50, 120, 300\}$. We also describe results with two layers in the ablation section 4.6.

Recall that we split the data across two dimensions: classes and samples. When training the backbone model, we hold out 20% of training (seen) classes for training the HN on classes the backbone does not see. From those classes, we held out images to serve as a validation set. We used those images of seen classes to evaluate the architecture performance and chose the hyperparameters based on that estimation.

B EV-PROOF

Theorem B.1. *Let f be a two-layer neural network $f(x) = W^{last}\sigma(W^{pen}x)$, whose weights are predicted by τ $[W^{last}, W^{pen}] = \tau(S^k)$. If $\tau(S^k)$ is equivariant to a permutation $\mathcal{P}$ with respect to W^{last}, and invariant to $\mathcal{P}$ with respect to W_{pen}, then $f(x)$ is equivariant to $\mathcal{P}$ with respect to the input of $\tau(S^k)$.*

Proof. From the equivariance of $f(x)$ to a permutation P over the input S^k, we have $\mathcal{P}(f(x_i;\tau_\phi(S^k)) = f(x_i;\tau_\phi(\mathcal{P}(S^k))$. Denote by l the number of rows of W^{last} and $z^{pen} = \sigma(W^{pen}x)$. We have

$$\begin{aligned}\mathcal{P}(f(x;\tau_\phi(S^k)) = \mathcal{P}(W^{last}\sigma(W^{pen}x)) = \mathcal{P}(W^{last}z^{pen})) = \mathcal{P}(\begin{bmatrix} W_1^{last}z^{pen} \\ \cdot \\ \cdot \\ W_l^{last}z^{pen} \end{bmatrix}) \\ = \begin{bmatrix} W_{\mathcal{P}(1)}^{last}z^{pen} \\ \cdot \\ \cdot \\ W_{\mathcal{P}(l)}^{last}z^{pen} \end{bmatrix} = \mathcal{P}(W^{last})z^{pen}.\end{aligned} \tag{4}$$

If $\tau(S^k)$ is equivariant to $\mathcal{P}$ with respect to W^{last}, and invariant to $\mathcal{P}$ with respect to W_{pen}, then $\tau(\mathcal{P}(S^k)) = [\mathcal{P}(W^{last}), W^{pen}]$, so

$$\mathcal{P}(f(x;\tau_\phi(S^k)) = \mathcal{P}(W^{last})z^{pen} = f(x;\tau_\phi(\mathcal{P}(S^k)). \tag{5}$$

□

C AWA GPT-3 DESCRIPTIONS

We use GPT3 (Brown et al., 2020) to generate 5 synthetic descriptions for each class of AwA. We use the API provided by OpenAI to ask ”text-davinci-002” engine with temperature of 0, max tokens of 512 and the following prompt:

”””Suggest 5 definitions for an animal.

Animal: moose
Definitions:

1. A large, dark-colored deer with enormous antlers, native to North America and Europe.
2. An animal of the deer family with humped shoulders, long legs, and a large head with antlers.
3. A large, awkward-looking mammal with a long face and humped shoulders.
4. A very large deer with antlers that can spread six feet or more from tip to tip.
5. The largest member of the deer family, with males weighing up to 1,800 pounds and having antlers that can spread up to six feet from tip to tip.

Animal: spider monkey
Definitions:

1. A type of monkey that has long legs and arms and a long tail.
2. A monkey that is found in the rainforests of Central and South America.
3. A monkey that is known for its acrobatic abilities.
4. A monkey that is considered to be one of the most intelligent primates.
5. A monkey that is endangered in many parts of its range.

Animal: rhinoceros
Definitions:

1. A large, thick-skinned mammal with one or two horns on its snout, native to Africa and southern Asia.
2. An animal that is hunted for its horn, which is used in traditional Chinese medicine.
3. A large, herbivorous mammal with a single horn on its nose, found in Africa and southern Asia.
4. A mammal of the family Rhinocerotidae, having thick, grey or brown skin and one or two horns on the snout.
5. A very large, plant-eating mammal with one or two horns on its nose, found in Africa and southern Asia.

Animal:
Definitions:”””.format(animal)

We provide here typical reply and will publish the full set of descriptions for reproducibility.

Elephant:

1. The largest land animal in the world, with males weighing up to six tons.
2. A plant-eating mammal with a long trunk, large ears, and thick, grey skin.
3. A mammal of the family Elephantidae, having a long trunk, large ears, and thick, grey skin.
4. An intelligent animal that is known for its memory and its ability to use its trunk for a variety of tasks.
5. An endangered species that is hunted for its ivory tusks.

D DATASETS SPLITS

SUN unseen classes: ’volcano’, ’poolroom establishment’, ’veterinarians office’, ’reception’, ’field wild’, ’diner indoor’, ’garbage dump’, ’server room’, ’vineyard’, ’jewelry shop’, ’drugstore’, ’herb garden’, ’lock chamber’, ’temple east asia’, ’marsh’, ’cottage garden’, ’cathedral outdoor’, ’dentists office’, ’pharmacy’, ’hangar indoor’, ’volleyball court indoor’, ’lift bridge’, ’synagogue outdoor’, ’boathouse’, ’ice shelf’, ’boxing ring’, ’rope bridge’, ’electrical substation’, ’auditorium’, ’chalet’, ’booth indoor’, ’wine cellar barrel storage’, ’greenhouse outdoor’, ’badminton court indoor’, ’thrift-shop’, ’cemetery’, ’rainforest’, ’courtyard’, ’underwater coral reef’, ’formal garden’, ’ice skating rink outdoor’, ’palace’, ’movie theater indoor’, ’dinette home’, ’sandbar’, ’ball pit’, ’amphitheater’

All remaining classes are seen classes.

ModelNet40 seen classes: 'airplane', 'bowl', 'desk', 'keyboard', 'person', 'sofa', 'tv stand', 'bathtub', 'car', 'door', 'lamp', 'piano', 'stairs', 'vase', 'bed', 'chair', 'dresser', 'laptop', 'plant', 'stool', 'wardrobe', 'bench', 'cone', 'flower pot', 'mantel', 'radio', 'table', 'xbox', 'bookshelf', 'cup', 'glass box', 'monitor', 'range hood'.

ModelNet40 unseen classes: 'tent', 'bottle', 'curtain', 'guitar', 'night stand', 'sink', 'toilet'.

AwA train attributes: 'orange', 'red', 'longneck', 'horns', 'tusks', 'flys', 'desert', 'cave', 'jungle', 'water', 'bush', 'lean', 'forest', 'gray', 'strainteeth', 'stripes', 'mountains', 'arctic', 'paws', 'hooves', 'pads', 'small', 'furry', 'ground', 'patches', 'white', 'fields', 'bipedal', 'toughskin', 'plains'.

AwA validation attributes: 'buckteeth', 'chewteeth', 'yellow', 'hairless', 'bulbous', 'big', 'flippers', 'tree', 'walks', 'coastal'.

AwA test attributes: 'quadrapedal', 'black', 'blue', 'ocean', 'longleg', 'spots', 'hands', 'claws', 'muscle', 'meatteeth', 'tail', 'brown', 'swims'.