Text2Model: Text-based Model Induction for Zero-shot Image Classification

Anonymous ACL submission

Abstract

We address the challenge of building task-agnostic classifiers using only text descriptions, demonstrating a unified approach to image classification, 3D point cloud classification, and action recognition from scenes. Unlike approaches that learn a fixed representation of the output classes, we *generate at inference time a model* tailored to a query classification task. To generate task-based zero-shot classifiers, we train a hypernetwork that receives class descriptions and outputs a multi-class model. The hypernetwork is designed to be equivariant with respect to the set of descriptions and the classification layer, thus obeying the symmetries of the problem and improving generalization. Our approach generates non-linear classifiers and can handle rich textual descriptions. We evaluate this approach in a series of zero-shot classification tasks, for image, point-cloud, and action recognition, using a range of text descriptions: From single words to rich descriptions. Our results demonstrate strong improvements over previous approaches, showing that zero-shot learning can be applied with little training data. Furthermore, we conduct an analysis with foundational vision and language models, demonstrating that they struggle to generalize when describing what attributes the class lacks.

1 Introduction

We explore the challenge of zero-shot image classification by leveraging text descriptions. This approach pushes the boundaries of conventional classification methods by demanding that models categorize images into specific classes based solely on written descriptions, without having previously encountered these classes during training.[1] In various domains, numerous attempts have been made to achieve zero-shot classification capacity (§2). Unfortunately, as we now explain, existing studies are limited in two major ways: (1) Query-dependence; and (2) Richness of Language description.

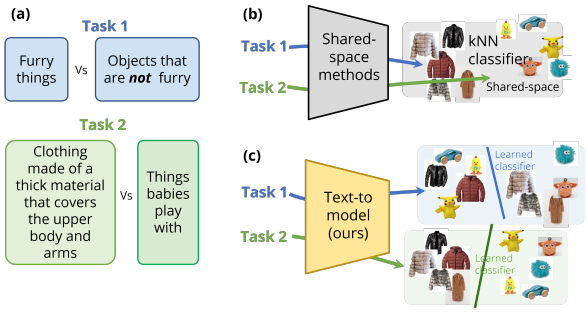


Figure 1: The text-to-model (T2M) setup. (a) Classification tasks are described in rich language. (b) Traditional zero-shot methods produce static representations, shared for all tasks. (c) T2M generates *task-specific representations and classifiers*. This allows T2M to extract task-specific discriminative features.

First, *Query-dependence*. To illustrate the issue, consider a popular family of zero-shot learning (ZSL) approaches, which maps text (like class labels) and images to a shared space (Globerson et al., 2004; Zhang and Saligrama, 2015; Zhang et al., 2017a; Sung et al., 2018; Pahde et al., 2021). To classify a new image from an unseen class, one finds the closest class label in the shared space. The problem with this family of shared-space approaches is that the learned representation (and the kNN classifier that it induces) remains "frozen" after training, and is not tuned to the classification task given at inference time. For instance, *furry toys* would be mapped to the same shared representation regardless of whether they are to be distinguished from other *toys*, or from other *furry things* (see Figure 1). The same limitation also hinders another family of ZSL approaches, which synthesize samples from unseen classes at inference time using conditional generative models, and use these samples with kNN classification (Elhoseiny and Elfeki, 2019; Jha et al., 2021). Some approaches address the query-dependence limitation by assum-

[1] We note that our definitions of "zero shot" or "zero shot learning" are slightly different than the ones used in the context of text-only language models.

ing that test descriptions are known during training (Han et al., 2021; Schonfeld et al., 2019), or by (costly) training a classifier or generator at inference time (Xian et al., 2018; Schonfeld et al., 2019). Instead, here we learn a model that produces task-dependent classifiers and representations without test-time training.

The second limitation is *language richness*. Natural language can be used to describe classes in complex ways. Most notably, people use negative terms, like "dogs without fur", to distinguish class members from other items. Previous work could only handle limited richness of language descriptions. For instance, it cannot represent adequately textual descriptions with negative terms (Akata et al., 2015; Xie et al., 2021b,a; Elhoseiny and Elfeki, 2019; Jha et al., 2021). In this paper, we wish to handle the inherent linguistic richness of natural language.

An alternative approach to address zero shot image recognition tasks involves leveraging large generative vision and language models (e.g., GPT-vision). These foundational models, trained on extensive datasets, exhibit high performance in zero and few-shot scenarios. However, these models are associated with certain limitations: (1) They entail significant computational expenses in both training and inference. (2) Their training is specific to particular domains (e.g., vision and language) and may not extend seamlessly to other modalities (e.g., 3D data and language). (3) Remarkably, even state-of-the-art foundational models encounter challenges when confronted with tasks involving uncommon descriptions, as demonstrated in §5.3.

Here, we describe a novel deep network architecture and a learning workflow that addresses these two aspects: (1) generating a discriminative model tuned to requested classes at query time and (2) supporting rich language and negative terms.

To achieve these properties, we propose an approach based on hypernetworks (HNs) (Ha et al., 2016). An HN is a deep network that emits the weights of another deep network (see Figure 2 for an illustration). Here, the HN receives a set of class descriptions and emits a multi-class model that can classify images according to these classes. Interestingly, this text-image ZSL setup has an important symmetric structure. In essence, if the order of input descriptions is permuted, one would expect the same classifiers to be produced, reflecting the same permutation applied to the outputs. This property is called *equivariance*, and it can be leveraged to design better architectures (Finzi et al., 2020; Cohen et al., 2019; Kondor and Trivedi, 2018; Finzi et al., 2021). Taking invariance and equivariance into account has been shown to provide significant benefits for learning in spaces with symmetries like sets (Zaheer et al., 2017; Maron et al., 2020) graphs (Herzig et al., 2018; Wu et al., 2020) and deep weight spaces (Navon et al., 2023). In general, however, HNs are not always permutation equivariant. We design invariant and equivariant layers and describe an HN architecture that respects the symmetries of the problem, and term it T2M-HN: *a text-to-model hypernetwork*.

We put the versatility of T2M-HN to the test across an array of zero-shot classification tasks, spanning diverse data types including images, 3D point clouds, and 3D skeletal data for action recognition. Our framework exhibits a remarkable ability to incorporate various forms of class descriptions including long and short texts, as well as class names. Notably, T2M-HN surpasses the performance of previous state-of-the-art methods in all of these setups.

Our paper offers four key contributions: (1) It identifies limitations of existing ZSL methods that rely on fixed representations and distance-based classifiers for text and image data. It proposes task-dependent representations as an alternative; (2) It introduces the Text-to-Model (T2M) approach for generating deep classification models from textual descriptions; (3) It investigates equivariance and invariance properties of T2M models and designs T2M-HN, an architecture based on HNs that adheres to the symmetries of the learning problem; and (4) It shows T2M-HN's success in a range of zero-shot tasks, including image and point-cloud classification and action recognition, using diverse text descriptions, surpassing current leading methods in all tasks.

2 Related work

In this section we cover previous approaches to leverage textual description to classify images of unseen classes.

Zero-shot learning (ZSL). The core challenge in ZSL lies in recognizing images of unseen classes based on their semantic associations with seen classes. This association is learned using human-annotated attributes (Li et al., 2019; Song et al., 2018; Morgado and Vasconcelos, 2017; Annadani and Biswas, 2018). Another source of information

Dataset name and type	Sample data	Description type	Example description
AwA (Lampert et al., 2009) Animal images		Class name	(1) *Moose* (2) *Elephant*
		Long	(1) *"An animal of the deer family with humped shoulders, long legs, and a large head with antlers."*, (2) *"A plant-eating mammal with a long trunk, large ears, and thick, grey skin."*
		Negative	(1) *"An animal without stripes and not gray"*, (2) *"An animal without fur and without horns"*
		Attribute	(1) *"Animals with fur"* (2) *"Animals with long trunk"*

Table 1: An illustration depicting the diverse tasks within the AwA dataset is provided. Appendix A contains illustrations for the remaining datasets.

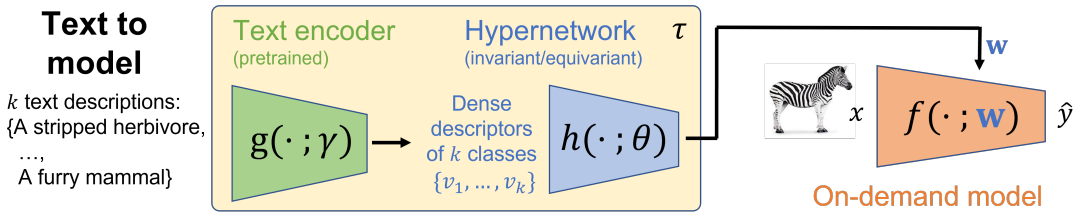


Figure 2: The text-to-model learning problem and our architecture. Our model (yellow box) receives a set of class descriptions as input and outputs weights w for a downstream on-demand model (orange). The model has two main blocks: A pretrained text encoder and a hypernetwork that obeys certain invariance and equivariance symmetries. The hypernetwork receives a set of dense descriptors to produce weights for the on-demand model.

for learning 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 and 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 and Hays, 2012; Wah et al., 2011). These can then be aggregated into class-level descriptions.

Shared space ZSL. 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; Atzmon and Chechik, 2018; Sung et al., 2018; Zhang and Saligrama, 2015; Atzmon and Chechik, 2019; Atzmon et al., 2020; Samuel et al., 2021; 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 further 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.

Generation-based ZSL. Another line of ZSL studies uses generative models like GANs to generate representations of samples from unseen classes (Elhoseiny and Elfeki, 2019; Jha et al., 2021). 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 methods assuming train-time access to information about the test classes are beyond our scope.[2] Yet, works

[2]While these algorithms could in principle be re-trained

assuming only test-time access to test-class information form some of our baselines (Elhoseiny and Elfeki, 2019; Jha et al., 2021).

Hypernetworks (HNs, Ha et al. (2016)) were applied to many computer vision and NLP problems, including ZSL (Yin et al., 2022), federated learning (Amosy et al., 2024), domain adaptation (Volk et al., 2022), language modeling (Suarez, 2017), machine translation (Platanios et al., 2018) and many more. Here we use HNs for text-based ZSL. The work by Lei Ba et al. (2015) also predicts model weights from textual descriptions, but differs in two key ways. (1) They learn a constant representation of each class; our method uses the context of all the classes in a task to predict data representation. (2) They predict weights of a linear architecture; our T2M-HN applies to deeper ones.

Large vision-language models (LVLM) CLIP (Radford et al., 2021), BLIP2 (Li et al., 2023) and GPT4Vision show remarkable zero-shot capabilities for vision-and-language tasks. A key difference between those approaches and this paper is that CLIP and BLIP2 (the training approach of GPT4Vision remains undisclosed) were trained on *massive multimodal data*. In contrast, our approach leverages the semantic compositionality of *language models*, without requiring paired image-text data. Furthermore, such large models are costly in both training and inference. They demand substantial resources, time and specialized knowledge that is not accessible to most of the research community. We successfully applied T2M-HN in domains lacking large multimodal data, such as 3D point cloud object recognition and skeleton sequence action recognition. The drawback is that the T2M-HN representation might react to language differences that don't matter for visual tasks.

3 Problem formulation

Our objective is to learn a mapping τ from a set of k natural language descriptions into the space of a k-class image classifier. 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 $S^k = \{s_1, \dots, s_k\}$ 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.

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

Let $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ 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, $\mathcal{Y} = \{1, \dots, k\}$. We can explicitly write the loss in terms of ϕ as follows. $l(y_i, \hat{y}_i) = l(y_i, f(x_i; W)) = l\left(y_i, f(x_i; \tau_\phi(S^k))\right)$. See also Figure 2 and note that $\tau = h \circ g$. The goal of T2M is to minimize $\phi^* = \arg\min_\phi \mathbb{E}_{S^k \sim \mathcal{P}^k} \mathbb{E}_{(x,y)\sim\mathcal{P}} \left[l\left(y, f(x; \tau_\phi(S^k))\right)\right]$. The training objective becomes

$$\phi^* = \arg\min_\phi \sum_j \sum_i l\left(y_i, f(x_i; \tau_\phi(S^{k_j}))\right), \tag{1}$$

where the sum over j means summing over all descriptions from all sets in the training set.

4 Our approach

We first describe our approach, based on HNs. We then discuss the symmetries of the problem, and an architecture that can leverage these symmetries.

We propose to address the T2M problem, using an HNs. An HN is a model that outputs the weights of another model (Ha et al., 2016). In our case, it receives a set of textual descriptions of classes to be recognized, and outputs the weights of a classifier that can discriminate them. Figure 2 illustrates our architecture. It has two components. First, a text encoder g takes natural language descriptions and transforms them into dense descriptors; and second, an HN h 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).

4.1 Symmetries of the T2M problem

Interestingly, the T2M setup imposes certain invariance and equivariance properties. Design an architecture that takes them into account can improve generalization. We now discuss these properties and then derive an architecture that captures them.

Equivariance properties of the classifier layer. As an illustrative example, consider a downstream

when new classes are presented at test-time (e.g. in a continual learning (Ring, 1995) setup), this would result in costly and inefficient inference mechanism, and possibly also in catastrophic forgetting (McCloskey and Cohen, 1989). We hence do not include them in our experiments.

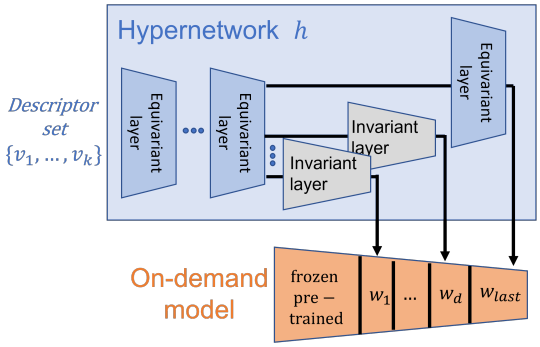


Figure 3: (a) The T2M-HN architecture for equivariant-invariant HN. The input is processed by equivariant layers, followed by a prediction head for each layer of the target on-demand classifier f. The prediction head for W_{last} is equivariant. Heads for earlier layers of f, $w_1, ...w_k$ are invariant. Refer to Appendix Figure 7 for schematics of the invariant and equivariant layers.

multi-class classifier f_1, that is designed to distinguish *cats* from *dogs*, and another classifier f_2, designed to distinguish *dogs* from *cats*. Intuitively, at the optimum, the two classifiers should be identical except for a switch of two weight vectors at the last layer (w_1 in f_1 equal to w_2 in f_2). This has an important implication for the hypernetwork. Any permutation applied to its input class descriptions should be reflected in a parallel ordering of the weight vectors that it produces. Appendix D.1 provides a formal definition of this property.

Invariance properties of intermediate layers. Considering now the layers of the downstream classifier before the last (classifier) layer. In Appendix D.1, we prove that using an equivariant transformation for the last layer and an invariant transformation for earlier layers is sufficient to ensure that the downstream classifier is equivariant to permutation over the descriptions.

4.2 Invariant and equivariant Architectures.

Given the equivariance property discussed above, we wish to design a deep architecture that adheres to those symmetries, because that improves generalization. To ensure that certain elements remain invariant permutation, they should be processed with a shared set of parameters (Wood and Shawe-Taylor, 1996; Ravanbakhsh et al., 2017). In our case, we need to share the parameters that process input descriptions, so the model is equivariant to permutations of those inputs.

Figure 3 gives the high-level structure of the equivariant architecture of T2M-HN. Schematics of equivariant layers and invariant layers are detailed in Appendix D.1. In the Appendix F.1 we present experiments demonstrating that using an equivariant architecture consistently improves generalization (Figure 9).

5 Experiments

The T2M setup is about producing a 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 three image datasets, one 3D point cloud dataset, and one action recognition dataset. We consider various forms of text description, including single-word class labels, few-word class names, 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. Due to space constraints, we provide a concise description of our experimental settings here. Further details can be found in Appendix B.

Baselines: We compare our T2M-HN with five 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 and Elfeki, 2019) trains conditional GANs with a loss designed to 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; (4) GRaWD (Jha et al., 2021) trains a conditional GAN with a loss that helps to reach regions in space that are hard to classify as seen classes; and (5) ZSML (Verma et al., 2020) combines meta-learning with a WGAN, to generate samples from unseen classes, and use them to train a classifier at test time. When relevant, we also computed the performance obtained when using CLIP, BLIP2, and GPT4Vision. Note that those models were trained using massively large datasets, so it is reasonable to assume they have seen all classes studied here. This is

	AWA by class name			ModelNet40 by class name		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
DeViSE	78.1 ± 1.0	58.9 ± 1.4	67.2 ± 1.9	83.6 ± 2.7	58.6 ± 3.4	68.9 ± 3
DEM	83.1 ± 1.6	75.1 ± 1.2	78.9 ± 2.0	86.7 ± 2.4	57.3 ± 3.3	69.0 ± 2.8
CIZSL	97.0 ± 0.1	74.7 ± 3.2	84.20 ± 2.0	97.6 ± 0.6	50.1 ± 3.6	66.3 ± 3.3
GRaWD	96.9 ± 0.1	81.6 ± 1.9	88.6 ± 1.1	97.8 ± 0.5	52.8 ± 3.3	68.3 ± 2.8
ZSML	96.1 ± 1.0	80.4 ± 2.4	87.5 ± 1.5	90.2 ± 1.5	68.6 ± 4.5	77.8 ± 3.0
T2M-HN (ours)	$\mathbf{98.9 \pm 0.1}$	$\mathbf{87.3 \pm 0.2}$	$\mathbf{92.7 \pm 0.1}$	$\mathbf{97.9 \pm 0.1}$	$\mathbf{75.1 \pm 0.9}$	$\mathbf{85.0 \pm 0.4}$
CLIP	98.9 ± 0.2	NA	NA	NA	NA	NA
BLIP2	99.6 ± 0.1	NA	NA	NA	NA	NA
GPT4Vision	100 ± 0.0	NA	NA	NA	NA	NA

Table 2: **Classification by single-word class names**. Accuracy on seen and unseen classes for AWA and ModelNet-40. Values are averages and SEM across all class pairs. LVLM have encountered all unseen classes, and cannot be applied to point clouds, hence marked as NA.

hence not zero-shot classification, and the results can be viewed as a "skyline" value that zero-shot approaches should aim at.

Datasets: We experiment with three image datasets: **(1) Animals with attributes** (AWA) (Lampert et al., 2009); **(2) SUN** (Patterson and Hays, 2012); and **(3) CUB** (Wah et al., 2011); a 3D point-clouds dataset: **(4) ModelNet40** (Wu et al., 2015); and an action recognition dataset: **(5) BABEL 120**(Punnakkal et al., 2021), containing sequences of body skeletons.

Experimental protocol: We split the data in two dimensions: Classes and samples. For standardized comparisons the splitting classes 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, and used to evaluate the model on the seen classes. We stress that "Seen" in our tables means *novel images* from *seen classes*.

Workflow: When training the whole architecture, we split the train seen classes. 80% of the classes were used for training the backbone. Then, we froze the weights of the backbone and use the remaining 20% to train the HN. This way, the HN learns to generalize to new classes. Finally, we evaluate the entire architecture on the evaluation split of the seen classes, and on the unseen classes. At test time, the model receives k class descriptions and predicts a model to classify images drawn from the corresponding k classes. Unless otherwise specified, we experiment with the value of $k = 2$.

5.1 ZSL using class names: Images and 3D point clouds

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

Results: Table 2 shows our model reaches the highest accuracy in both experimental setups and datasets.

5.2 ZSL using text descriptions: Images and sequences of 3D skeletons

Next, we evaluate T2M-HN when using richer 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". **(2) For BABEL 120** we use the action names provided by the original dataset. Many of the actions have multi-word, descriptive names such as *"take of bag"*. **(3) For AwA**, we use synthetic class descriptions generated by a GPT model. See detailed examples in the Appendix G. We will publish the full set of descriptions for reproducibility. **(4) For CUB**, we use the descriptions of each image in a given class as a possible description of the class.

In the CUB dataset, bird species from the same taxonomic family are harder to distinguish from each other than random pairs of species (Vedantam et al., 2017). We used the Datazone dataset of bird species (BirdLife, 2022) and annotated each species with its corresponding taxonomic family. Based on this information, we defined pairs of bird species from two different families as *easy* and pairs from the same family as *hard*.

Results: Table 3 presents the classification accuracy obtained using class descriptions, for the AWA, SUN, and BABLE datasets. T2M-HN outperforms all baselines. Figure 4 shows the results for the CUB dataset with easy and hard tasks. To better understand the results, consider an important distinction between our approach and previous *shared-representation* approaches. These approaches aim to learn class representations that would generalize to new classification tasks. In contrast, our ap-

	SUN by short description			BABEL by short descriptions			AWA by GPT descriptions		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
CLIP	99.1 ± 0.4	NA	NA	NA	NA	NA	93.7 ± 0.2	NA	NA
BLIP2	98.9 ± 0.1	NA	NA	NA	NA	NA	92.1 ± 0.4	NA	NA
GPT4Vision	99.8 ± 0.2	NA	NA	NA	NA	NA	97.5 ± 0.5	NA	NA
DeViSE	52.0 ± 1.4	58.9 ± 1.1	55.2 ± 0.9	65.9 ± 4.4	51.1 ± 2.0	57.6 ± 2.8	91.8 ± 1.6	70.0 ± 3.7	79.4 ± 2.2
DEM	83.2 ± 1.1	83.2 ± 1.4	83.2 ± 0.9	56.6 ± 2.4	50.2 ± 1.1	53.2 ± 1.5	93.9 ± 1.2	73.0 ± 3.3	82.1 ± 1.8
CIZSL	94.0 ± 0.1	80.3 ± 0.6	86.6 ± 0.3	82.7 ± 2.1	62.5 ± 1.3	71.2 ± 1.2	96.6 ± 0.1	80.7 ± 2.2	87.9 ± 1.3
GRaWD	95.5 ± 0.1	84.7 ± 0.5	89.8 ± 0.3	83.7 ± 1.8	62.2 ± 1.1	71.3 ± 1.0	96.8 ± 0.1	81.1 ± 0.2	88.3 ± 1.2
ZSML	$\mathbf{96.9 \pm 0.1}$	85.5 ± 0.4	90.8 ± 0.2	52.6 ± 1.3	51.2 ± 0.9	51.9 ± 1.1	97.4 ± 0.5	72.3 ± 2.7	82.9 ± 1.8
T2M-HN (ours)	95.8 ± 0.1	$\mathbf{88.4 \pm 0.1}$	$\mathbf{92.0 \pm 0.1}$	$\mathbf{95.3 \pm 0.1}$	$\mathbf{77.6 \pm 0.1}$	$\mathbf{85.5 \pm 0.1}$	$\mathbf{98.7 \pm 0.1}$	$\mathbf{83.3 \pm 0.1}$	$\mathbf{90.3 \pm 0.1}$

Table 3: **Classification using short and rich class descriptions**. Values are the mean ($\pm$ s.e.m) accuracy averaged over 100 random class pairs (for SUN and BABEL 120) and all class pairs (for AwA). LVLM have encountered all unseen classes, and cannot be applied to 3D skeletons, hence marked as NA.

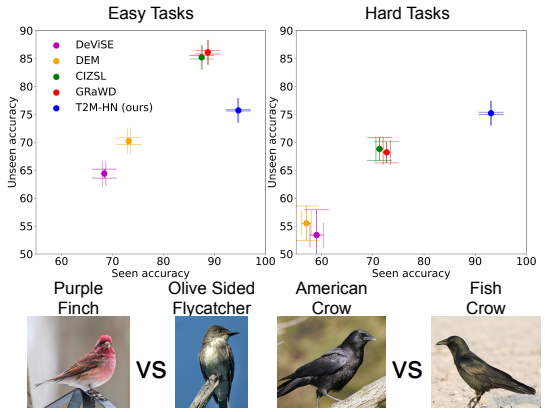


Figure 4: **Classifying easy and hard pairs of bird species from the CUB dataset**. Easy tasks involve binary classification of bird pairs from different taxonomy families. Hard tasks classify bird pairs within the same taxonomy family. Mean accuracy is shown for images from both seen (x-axis) and unseen (y-axis) classes, averaged across all pairs.

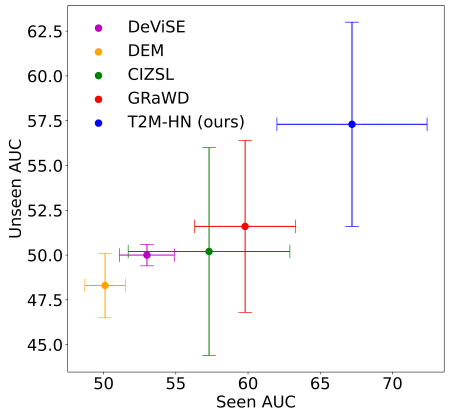


Figure 5: AUC of seen and unseen classes, in a one class task that crosses species boundaries: "*Animals that have horns*". Shown are averages over 53 attributes.

proach aims to build task-specific representations and classifiers. For easy tasks, task-dependent representation may not be important because the input contains a sufficient signal for accurate classification. In contrast, in hard tasks, a model would benefit from task-dependent representation to focus on the few existing discriminative features of the input examples. Indeed, as demonstrated in Figure 4, in the easy tasks, although our model is superior on the seen classes, it is outperformed by the GAN-based baselines on unseen classes. In contrast, for the hard tasks, where task-specific class representation is more valuable, our model is superior on both seen and unseen classes.

5.3 Descriptions with negative terms

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", or animals that "do not fly". To create such negative descriptions for the AwA data, we used the list of attributes provided for each class in AwA. For each class, we randomly sampled 4 attributes that do not apply to that class.

Results: Table 4 shows our findings for two scenarios: purely negative descriptions (left side) and balanced positive and negative descriptions (right side), maintaining equal training and testing ratios for both scenarios.

T2M-HN outperforms all baselines by significant gaps. Presumably, the best baseline, GRaWD, which generates image features from the textual descriptions, fails to generate proper images given negative attributes. Interestingly, LVLM performance significantly drops in these scenarios, likely because these models were trained on image captions that seldom include negative descriptions.

AWA data	Negative descriptions			Negative & positive descriptions		
	Seen	Unseen	Harmonic	Seen	Unseen	Harmonic
CLIP	19.9±2.2	NA	NA	56.8±2.9	NA	NA
BLIP2	73.9±0.6	NA	NA	50.1±0.7	NA	NA
GPT4Vision	27.6±0.9	NA	NA	54.1±0.9	NA	NA
DeViSE	57.3±4.9	54.5±5.2	55.9±5.0	79.5±3.6	61.5±4.5	69.4±4.0
DEM	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	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	54.9±0.8	56.0±3.2	55.3±1.6	95.0±0.2	73.9±2.0	83.2±1.5
T2M-HN(ours)	**90.0±0.2**	**77.1±0.3**	**83.0±0.2**	**96.6±0.2**	**82.9±0.2**	**89.2±0.1**

Table 4: **Classification using negative descriptions**. Mean accuracy for images from seen and unseen AwA classes, averaged over all class pairs. LVLMs, trained on extensive datasets, likely encountered all unseen classes, hence marked as NA.

5.4 Identifying complex classes membership

Typically, zero-shot classification involves distinguishing "natural categories" (Rosch, 1973) like *"cats"* and *"dogs"*. However, We may want to generate classifiers that follow more complex class boundaries, aggregating over multiple natural classes. For instance, *"animals with horns"* combine several classes from a rhino to a deer.

To test T2M-HN in this scenario, we created a set of one-class classification tasks designed to recognize images based on properties that cut through class boundaries. To make the evaluation systematic, we used attributes from AwA, and eliminate non-visual attributes. Details of the protocol are given in Appendix I. We report the average Area Under the Recall-Precision Curve over seen classes and unseen classes.

Results: Figure 5 shows that T2M-HN captures the complex semantic distinctions of our task better than baselines. We attribute this to its ability to draw new classifiers for each new description.

5.5 T2M-HN classifiers are task-specific

Leading text-based ZSL methods map class descriptions or images to a shared representation, but that mapping is constant for all classification tasks. Our T2M-HN is designed to use information about the classes of each specific classification task.

We use GradCam (Selvaraju et al., 2017) and examine what image areas are used in different classification tasks. Figure 6 explores two such examples. The upper three panels show the image regions that are used for classifying the image as a *Dolphin*. 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.

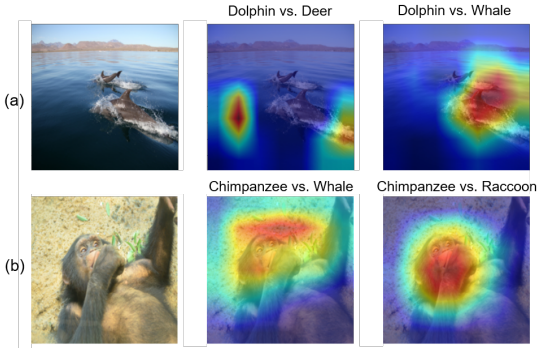


Figure 6: Class context affects the predicted classifier. **Top left:** An image of a dolphin. **Top middle:** grad-cam heat map when classifying the dolphin image using a model trained for *dolphin* vs *deer*: The model is strongly affected by the background ocean water, presumably because the negative class lives on land. **Top right:** Recognition using a model for *dolphin* vs. *killer whale*: the model attends to the dolphin, since background would be similar for both classes. **Bottom:** A similar effect for a chimpanzee.

6 Conclusion

We introduced the T2M learning algorithm, a novel approach that generates an image recognition classifier "on demand" using only class descriptions provided at test time. T2M allows for task-dependent class representations rather than fixed ones. We analyzed the group symmetries a T2M model must adhere to and introduced T2M-HN, a model based on HNs that obeys these symmetries. Through extensive experiments across various classification scenarios—including images, 3D point clouds, and action recognition—we explored the adaptability of the model to descriptions of differing complexities, from single and few-word class names, through long text descriptions, all the way to "negative" and attribute descriptions. Our results clearly demonstrate the potential of the T2M modeling approach.

7 Limitations

Non-Visual descriptions Our objective is to classify images belonging to previously unseen classes by leveraging textual descriptions. Nevertheless, it is noteworthy that textual descriptions may occasionally encompass non-visual attributes, that may mislead the model to look for irrelevant features. Although we tested our approach in similar challenges like negative descriptions (§5.3) and complex classes membership (§5.4), this is a potential challenge to the efficacy of our approach.

Massive Datasets The primary rationale behind employing the zero-shot learning paradigm stems from the inherent difficulty in amassing comprehensive datasets encompassing all classes expected to be encountered during inference. Nevertheless, contemporary big tech industries have been able to compile exceedingly large datasets, which, with rare exceptions, span the entirety of conceivable classes, thus potentially diminishing the significance of the zero-shot setup in practical applications. Although such datasets are presently not publicly accessible, the evolving trend toward their potential availability may impact the pertinence of the zero-shot learning framework in the foreseeable future.

References

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

Ohad Amosy, Gal Eyal, and Gal Chechik. 2024. Late to the party? on-demand unlabeled personalized federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2184–2193.

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

Yuval Atzmon and Gal Chechik. 2018. Probabilistic and-or attribute grouping for zero-shot learning. *arXiv preprint arXiv:1806.02664*.

Yuval Atzmon and Gal Chechik. 2019. Adaptive confidence smoothing for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11671–11680.

Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. 2020. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 33:1462–1473.

BirdLife. 2022. Data retrieved from http://datazone.birdlife.org/species/taxonomy.

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2018. SMASH: one-shot model architecture search through hypernetworks. In *ICLR (Poster)*. OpenReview.net.

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. 2020. Language models are few-shot learners. In *NeurIPS 2020*.

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

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

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

Ali Cheraghian, Shafin Rahman, and Lars Petersson. 2019. Zero-shot learning of 3d point cloud objects. In *16th International Conference on Machine Vision Applications, MVA 2019, Tokyo, Japan, May 27-31, 2019*, pages 1–6. IEEE.

Taco S Cohen, Mario Geiger, and Maurice Weiler. 2019. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32.

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

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 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5640–5649.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. 2020. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. 2021. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pages 3318–3328. PMLR.

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. *Advances in neural information processing systems*, 26.

Tomer Galanti and Lior Wolf. 2020. On the modularity of hypernetworks. *Advances in Neural Information Processing Systems*.

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

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

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

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

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

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

Risi Kondor and Shubhendu Trivedi. 2018. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR.

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

Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

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

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

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

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

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

Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. 2021. Generative zero-shot learning for semantic segmentation of 3d point clouds. In *International Conference on 3D Vision, 3DV 2021, London, United Kingdom, December 1-3, 2021*, pages 992–1002. IEEE.

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

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. 2023. Equivariant architectures for learning in deep weight spaces. *CoRR*, abs/2301.12780.

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

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

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

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.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. *arXiv preprint arXiv:1808.08493*.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. 2021. BABEL: bodies, action and behavior with english labels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 722–731. Computer Vision Foundation / IEEE.

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

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

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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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

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.

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

Mark B. Ring. 1995. *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas at Austin, TX, USA.

Eleanor H Rosch. 1973. Natural categories. *Cognitive psychology*, 4(3):328–350.

Dvir Samuel, Yuval Atzmon, and Gal Chechik. 2021. From generalized zero-shot learning to long-tail with class descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 286–295.

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/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255.

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

Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. 2019. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12026–12035. Computer Vision Foundation / IEEE.

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

Joseph Suarez. 2017. Language modeling with recurrent highway hypernetworks. *Advances in neural information processing systems*, 30.

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

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–260.

Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. 2020. Meta-learning for generalized zero-shot learning. In *Proceedings of the AAAI*.

Tomer Volk, Eyal Ben-David, Ohad Amosy, Gal Chechik, and Roi Reichart. 2022. Example-based hypernetworks for out-of-distribution generalization. *arXiv preprint arXiv:2203.14276*.

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

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

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

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

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

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

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

Li Yin, Juan M Perez-Rua, and Kevin J Liang. 2022. Sylph: A hypernetwork framework for incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9035–9045.

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

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

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

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

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 Overview of evaluation datasets and tasks.

We evaluate the versatility of T2M-HN across a spectrum of zero-shot classification tasks, encompassing different data types such as images, 3D point clouds, and 3D skeletal data for action recognition (see Table 5). Our framework demonstrates a remarkable capability to assimilate diverse forms of class descriptions, including both long and short texts, as well as class names. Importantly, T2M-HN outperforms previous state-of-the-art methods in all of these experimental setups.

B Implementation and architecture

Implementation and architecture: We encode single-word class names from the AwA dataset using Glove (Pennington et al., 2014) and longer descriptions, as well as class names, from ModelNet-40 using SBERT (Reimers and 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. For action recognition data, we follow (Punnakkal et al., 2021) and use 2 stream-AGCN (Shi et al., 2019), with one or two predicted fully-connected layers as well.

For CLIP, we use the CLIP encoder followed by k-NN classifier in the CLIP space (Radford et al., 2021). For BLIP we use LoRA to tune the model to the classification task using the train split. For GPT4Vision we use the prompt to demonstrate the task, followed by the classification task from the test split. Since we have a limited number of calls to those models we sampled classes and descriptions from the test split. We increased the sample size until the SEM was small enough to claim statistical significance.

Experimental protocol: We split the data in two dimensions: Classes and samples. For standardized comparisons the splitting classes into *seen classes* used for training and *unseen classes* used in evaluation, follows the split used by (Xian et al., 2018) for AWA, the split of (Cheraghian et al., 2022) for Modelnet40 and the standard split of (Wah et al., 2011) for CUB. Since there is no official split for SUN and BABEL, we share our random split in Section H. As in other ZSL protocols, for each seen class we split out a set of evaluation images that are not presented during training, and used to evaluate the model on the seen classes. For AwA, CUB, SUN and BABEL 120 we randomly selected 10% of images for “seen" evaluation. For ModelNet40 we use the test split in (Wu et al., 2015). We stress that "Seen" in our tables means *novel images* from *seen classes*.

Training cost: Our training was completed in $\sim$ 30 minutes on a single 2080Ti GPU. This is faster than baselines: DEM and DEVISE require twice as long for training (1 hr), while ZSML, GRAWD and CIZSL took x4 the time (2 hrs). This result agrees with previous literature on HNs, e.g. (Brock et al., 2018; Galanti and Wolf, 2020).

C Hyperparameter optimization

We tune hyperparameters using 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 $\in \{0.00001, 0.0001, 0.001, 0.1\}$, and 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 $\{1, 2, 3, 5, 10\}$.

We tried several sizes for the HN architecture with one hidden layer, $\{30, 50, 120, 300\}$. We also describe results with two layers in the ablation section at the supplemental Sec. F.1.

Recall that we split the data across two dimensions: classes and samples. When training the backbone model, we held 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.

D Equivariant and invariant layers

As an illustrative example, consider a downstream multi-class classifier f_1, that is designed to distinguish *cats* from *dogs*, and another classifier f_2, designed to distinguish *dogs* from *cats*. Intuitively, at the optimum, the two classifiers should be identical except for a switch of two weight vectors at the last layer (w_1 in f_1 equal to w_2 in f_2). This has an important implication for the hypernetwork. Any

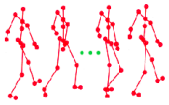
Dataset name and type	Sample data	Description type	Example description
ModelNet-40 (Wu et al., 2015) 3D Point Clouds CAD models		Class name	(1) *Airplane* (2) *Chair*
AwA (Lampert et al., 2009) Animal images		Class name	(1) *Moose* (2) *Elephant*
		Long	(1) *"An animal of the deer family with humped shoulders, long legs, and a large head with antlers."*, (2) *"A plant-eating mammal with a long trunk, large ears, and thick, grey skin."*
		Negative	(1) *"An animal without stripes and not gray"*, (2) *"An animal without fur and without horns"*
		Attribute	(1) *"Animals with fur"* (2) *"Animals with long trunk"*
SUN (Patterson and Hays, 2012) Images of scenes and places		Short	(1) *"Desert vegetation"*, (2) *"Lecture room"*
CUB (Wah et al., 2011) Images of bird species		Long	(1) *"This bird is red with an orange beak and black eyes and eyebrow."*, (2) *"a small yellow bird with a black chest and tail."*
BABEL 120 (Punnakkal et al., 2021) Sequences of 3D skeletal data		Short	(1) *"Take off bag"*, (2) *"Type on a keyboard"*

Table 5: Overview of evaluation datasets and tasks.

permutation applied to its input class descriptions should be reflected in a parallel ordering of the weight vectors that it produces. We now show how to design a hypernetwork that obeys this property.

D.1 Equivariance properties of the classifier layer.

Consider a downstream multiclass deep classifier whose last (classification) layer has a weight vector $w_i \in \mathbb{R}^m$ for the output class i. The weight matrix of the last layer is $W_{last} = \{w_1, \ldots, w_k\}$ (See Figure 7a).

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-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 $W = \tau_\phi(S^k)$.

The HN receives k class descriptors and outputs their corresponding weights

$$\begin{aligned} W_{last} &= \{w_1, \ldots, w_k\} \\ &= R_{last}(\tau_\theta(\{s_1, \ldots, s_k\})), \end{aligned} \tag{2}$$

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

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

This is the equivariant property, and the HN must obey it.

D.2 Invariance properties of intermediate layers

Considering now the layer of the downstream classifier before the last layer (w_d in Figure 7a). A similar argument holds for earlier (lower) intermediate layers. 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 the downstream classifier is equivariant to permutation over the descriptions.

Theorem D.1. *Let f be a two-layer neural network $f(x) = W^{last}\sigma(W^{pen}x)$, whose weights are predicted from descriptors $S^k = \{s_1, \dots, s_k\}$ such that $[W^{last}, W^{pen}] = \tau(S^k)$. 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 m the number of rows of W^{last} and $z^{pen} = \sigma(W^{pen}x)$. We have

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

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

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

□

D.3 Invariant and equivariant Architectures

Figure 7(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 invariant layer has a similar architecture (Figure 7(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 use 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 7(a)).

E Multi-class classification

To demonstrate the flexibility of our approach to deal with multiple classes, we evaluated T2M-HN in 3-way classification tasks. In each task, the on-demand model classifies the image into one out of three classes. For example, such a task could be to classify whether an image is a dog, a cat, or an elephant. We use the same workflow as described in Section 5, with $k = 3$. Results are in Table 6. T2M-HN outperforms all baselines by a large margin.

	AwA triplets by class name		
	Seen	Unseen	Harmonic
DeViSE	95.1 ± 0.7	55.6 ± 3.6	70.2 ± 1.2
DEM	94.6 ± 0.7	64.3 ± 3.0	76.6 ± 1.1
CIZSL	97.0 ± 0.4	62.0 ± 2.9	75.6 ± 2.1
GRaWD	96.4 ± 0.5	68.5 ± 3.0	80.0 ± 2.0
T2M-HN (ours)	$\mathbf{98.1 \pm 0.1}$	$\mathbf{75.3 \pm 0.1}$	$\mathbf{85.2 \pm 0.1}$

Table 6: Classification by class descriptions. Mean classification accuracy and SEM on images from seen and unseen classes. Averages are over 100 random class triplets

F 3D point cloud multiclass classification

While T2M-HN is designed to excel in binary classification, it can be easily applied to multiclass problems. For comparison with previous models we evaluate its performance in multi-class settings, where T2M-HN predicts a model that classifies all seen and unseen classes, instead of two specific classes. Table 7 shows the results of this experiment. We report the result when classifying new samples from the seen classes (30-classes classification) and from the unseen classes (10-classes classification). T2M-HN achieves SOTA results in this setup as well. It leverages the text generalization of the HN model to distinguish between unseen classes.

We further computed the top-k accuracy achieved by running T2M-HN for the unseen classes. Figure 8 plots the accuracy as a function of k. T2M-HN provides superior accuracy for all tested values of k. To calculate the top-k perfor-

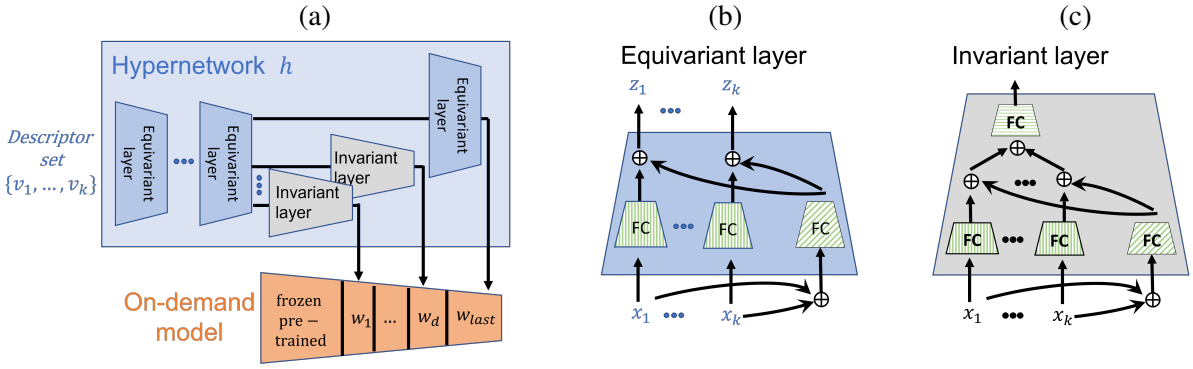


Figure 7: (a) The T2M-HN architecture for equivariant-invariant hypernetwork. The input is processed by equivariant layers, followed by a prediction head for each layer of the target on-demand classifier f. The prediction head for W_{last} is equivariant. Heads for earlier layers of f, $w_1, ...w_k$ 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.

	ModelNet40 by class name		
	Seen	Unseen	Harmonic
DeViSE	47.2	14.5	22.2
DEM	46.8	7.0	12.3
CIZSL	75.6	6.0	11.0
GRaWD	75.2	10.9	19.0
T2M-HN (ours)	**76.3**	**18.9**	**30.3**

Table 7: **3D point-cloud object recognition using single-word class names**. Multiclass accuracy on seen and unseen classes for ModelNet-40. The seen accuracy is between 30 classes, and the unseen accuracy is between 10 classes.

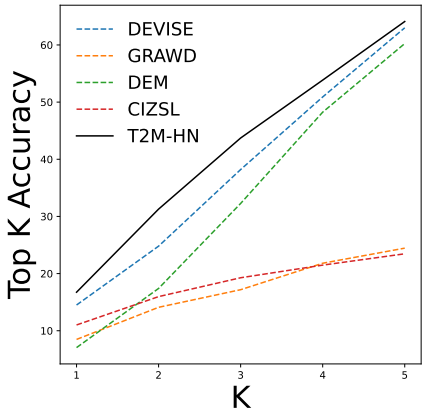


Figure 8: Accuracy at k for experiments with 3D point cloud from ModelNet-40. The solid line is our T2M model, dashed lines are for the baseline models.

	AwA Super Sets		
	Seen	Unseen	Harmonic
DeViSE	53.0 ± 1.9	50 ± 0.6	51.5 ± 0.9
DEM	50.1 ± 1.4	48.3 ± 1.8	49.2 ± 1.6
CIZSL	57.3 ± 5.6	50.2 ± 5.8	55.0 ± 4.0
GRaWD	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 8: **Classification using attributes**. Values denote the Area under the Recall-Precision curve averaged over the 13 test attributes $\pm$ s.e.m. over these attributes. The seen results are new images from the seen classes, while the unseen results are images from unseen classes. Both are evaluated when classifying only the test attributes. The full protocol is in I.

mance of the GAN-based models, after generating the images, we checked if any of K closest neighbors of an image is of the correct class.

F.1 The Impact of Equivariance Design on HNs

To evaluate the effect of the equivariance property on our HN-based model performance, we compared variants with and without the equivariance design. We repeat the experiment for an on-demand model with one or two fully connected layers. Figure 9 shows the mean accuracy of the following variants: **(1) T2M-HN 1-layer** An equivariant HN that predicts one equivariant FC layer; **(2) 1-layer w.o. EV** A FC HN that predicts one fully connected layer; **(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; and **(4) 2-layer w.o. EV** A FC HN that predicts two FC layers.

In all cases, the equivariant HN performs better than the simple fully connected. For AwA, T2M-HN 1-layer performs better than T2M-HN 2-layers. We believe this is because ResNet backbone separates the images to be linearly separable. For BABEL, we used 2s-AGCN as a features extractor and in that case, T2M-HN 2-layer generalizes better to unseen classes.

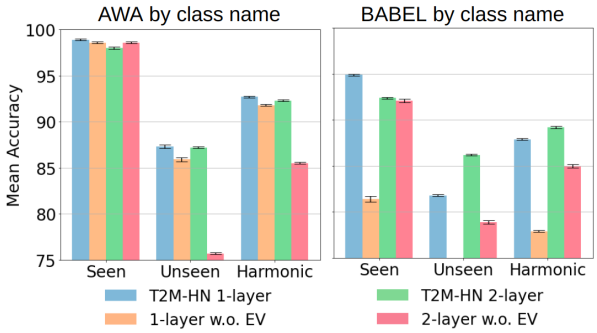


Figure 9: Ablation study. Mean classification accuracy (averaged across class pairs) on seen and unseen classes and their harmonic mean for the AWA and BABEL datasets.

G AwA GPT-3 descriptions

We use GPT3 (Brown et al., 2020) to generate 5 synthetic 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. We use the API provided by OpenAI to ask "text-davinci-002" engine with a temperature of 0, max tokens of 512, and the prompt: "Suggest 5 definitions for an animal. Animal: {animal_name}. Definitions:"

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.

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.

H Data 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', 'thriftshop', 'cemetery', 'rainforest', 'courtyard', 'underwater coral reef', 'formal garden', 'ice skating rink outdoor', 'palace', 'movie theater indoor', 'dinette home', 'sandbar', 'ball pit', 'amphitheater'

SUN seen classes: All remaining classes.

ModelNet40: We follow (Cheraghian et al., 2022, 2019; Michele et al., 2021) and use the 10 classes included in ModelNet-10 as unseen classes, and the other 30 as seen.

BABEL unseen classes: 'a pose', 'action with ball', 'adjust', 'catch', 'clean something', 'communicate (vocalise)', 'crawl', 'get injured', 'hand movements', 'hop', 'limp', 'mix', 'play sport', 'press something', 'rolling movement', 'shuffle', 'side to side movement', 'sneak', 'spread', 'support', 'swing body part', 'trip', 'upper body movements', 'wait'

BABEL seen classes: All remaining classes.

CUB unseen classes: 'Acadian Flycatcher', 'American Crow', 'American Three Toed Woodpecker', 'Baltimore Oriole', 'Bank Swallow', 'Belted Kingfisher', 'Black Billed Cuckoo', 'Black Footed Albatross', 'Black Throated Sparrow', 'Boat Tailed Grackle', 'Bohemian Waxwing', 'Brandt Cormorant', 'Brewer Blackbird', 'Cape May Warbler', 'Cedar Waxwing', 'Chestnut Sided Warbler', 'Field Sparrow', 'Golden Winged Warbler', 'Grasshopper Sparrow', 'Gray Crowned Rosy Finch', 'Great Crested Flycatcher', 'Great Grey Shrike', 'Groove Billed Ani', 'Hooded Oriole', 'Horned Grebe', 'Indigo Bunting', 'Least Auklet', 'Least Tern', 'Marsh Wren', 'Mockingbird', 'Northern Flicker', 'Northern Waterthrush', 'Pacific Loon', 'Pied Billed Grebe', 'Pomarine Jaeger', 'Purple Finch', 'Red Legged Kittiwake', 'Rhinoceros Auklet', 'Sayornis', 'Scott Oriole', 'Tree Sparrow', 'Tree Swallow', 'Western Grebe', 'Western Gull', 'Western Wood Pewee', 'White Breasted Kingfisher', 'White Eyed Vireo', 'White Pelican', 'Wilson Warbler', 'Yellow Bellied Flycatcher', 'Yellow Billed Cuckoo'

CUB seen classes: All remaining classes.

I Attributes used for one-class classification

As mentioned in section 5.4, we use some of the attributes from the AwA dataset to define one-class classification tasks. First, we removed non-visual attributes. Then, we randomly split the remaining 53 attributes into 30 train, 10 validation, and 13 test attributes. We split both the images and the attributes, constructing 4 groups of images and attributes: (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)). The results are shown in Figure 5 and in Table 8. The attributes split is as follows:

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'.