Interpretable Compositional Representations for Robust Few-Shot Generalization

Samarth Mishra*, Pengkai Zhu*, and Venkatesh Saligrama, *Fellow, IEEE*

Abstract—We propose Recognition as Part Composition (RPC), an image encoding approach inspired by human cognition. It is based on the cognitive theory that humans recognize complex objects by components, and that they build a small compact vocabulary of concepts to represent each instance with. RPC encodes images by first decomposing them into salient parts, and then encoding each part as a mixture of a small number of prototypes, each representing a certain concept. We find that this type of learning inspired by human cognition can overcome hurdles faced by deep convolutional networks in low-shot generalization tasks, like zero-shot learning, few-shot learning and unsupervised domain adaptation. Furthermore, we find a classifier using an RPC image encoder is fairly robust to adversarial attacks, that deep neural networks are known to be prone to. Given that our image encoding principle is based on human cognition, one would expect the encodings to be interpretable by humans, which we find to be the case via crowd-sourcing experiments. Finally, we propose an application of these interpretable encodings in the form of generating synthetic attribute annotations for evaluating zero-shot learning methods on new datasets.

Index Terms—Explainable AI, Few-Shot Learning, Zero-Shot Learning, Domain Adaptation, Adversarial Machine Learning, Human Cognition, Compositional Learning, Computer Vision

1 Introduction

Deep convolutional networks (DCNs) although effective at image classification, need large amounts of annotated images to learn from [1]. They encounter major hurdles when learning from a few labeled examples [2], [3]. Moreover, the image features that these networks learn are often quite fickle and do not adapt to changes in the image domain and are also susceptible to adversarial image perturbations [4], imperceptibly small to the human eye.

Humans, even children, on the other hand, do not face these challenges and can effectively learn concepts having only seen a few examples of it [5], [6], [7]. A well accepted theory behind this capability of humans is the ability to "recognize by components" [8]. In other words humans can learn to recognize concepts as a composition of simpler pieces. Lake *et al*. [9] attempted at leveraging this in a Bayesian Program Learning framework, to build a system for effective one-shot generalization at tasks like recognizing categories of hand-written characters, and found low-shot generalization comparable to that of humans and much better than state of the art DCNs for image recognition.

Decomposing an image into parts is a recognition approach suggested by this theory. Additionally, we draw inspiration from quantum cognition theory, that suggests each instance of an object typically is learnt as a superposition of multiple concepts [10]. An instance can be represented as a weighted sum of these concepts with coefficients representing collapse probabilities. Under this framework, each part recognized in an image can be represented as a convex combination of a vocabulary of few concepts or prototypes.

Inspired from this theory, we propose Recognition as Part Composition (RPC), a method that learns image representations by first decomposing it into a few semantically representative parts, and then learning an encoding of each part in terms of a small number of prototypes. For discovering different parts of an image, we use a multi-attention convolutional neural network (MACNN) (similar to the model used by [11]), and unlike the approach of Lake *et al*. [9], which used additional annotations in the form of pen strokes for character recognition, our model automatically recognizes representative parts of an image. Only annotations we need to use for our model are class-labels for images.

To illustrate the encodings generated by RPC, we show an example in Fig 1. Our model learns to recognize key parts of a bird like its head (part-0), breast (part-1) *etc*. Additionally it learns certain prototypical types for each part, which we represent via 4 images of parts closest to that type in Fig. 1. Our model produces a distribution over part-types representing the likelihood that the given part comes from a certain part-type, for each part recognized in the image. The RPC encoding is a collection of these distributions for each part recognized by our model. This image encoding has surprisingly good low-shot generalization properties as we shall see in our evaluations in few-shot learning, zero-shot learning and domain adaptive image recognition tasks. This indicates that concepts learnt by our method using a training set of images, generalize well to an unseen set. We also find that these representations are interpretable by humans and can be robust to adversarial image perturbations.

Explainability is favored in models that are deployed for making predictions in the real world, so human oper-

- *: Equal contribution.
- *Samarth is with the Department of Computer Science at Boston University.*
- *Pengkai is with AWS AI.*
- *Venkatesh is with the Department of Electrical and Computer Engineering at Boston University.*
  Primary contact: zpk@bu.edu

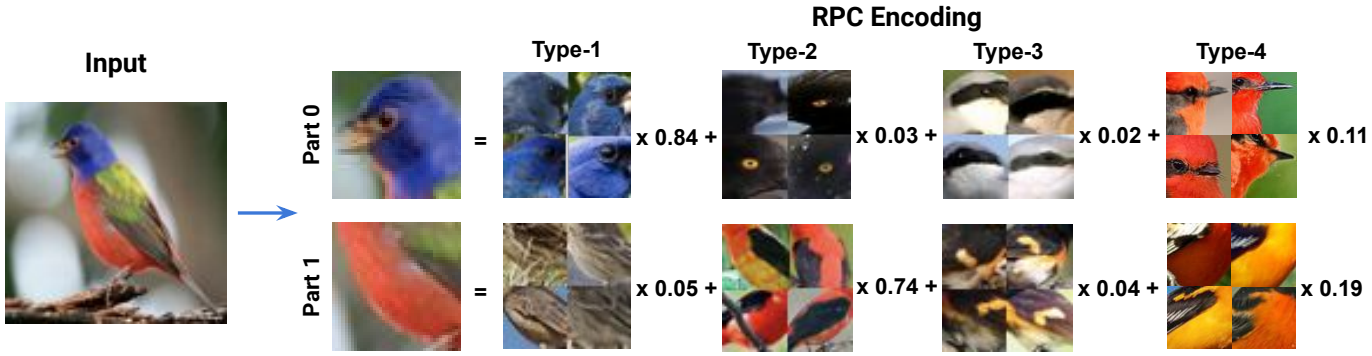


Fig. 1: Recognition as Part Composition (RPC) learns to recognize representative parts in an image and represents each part instance using a mixture of prototypes, each representing a concept/part-type. The coefficients of the mixture are the likelihoods that that specific instance is of the corresponding part-type.

ators can trust the model's decision and possibly diagnose problems when they arise. DCNs, however, with their high dimensional image representations and highly non-linear reasoning end up becoming "black-box" models, making it difficult to understand the reasoning behind their decisions. Conventional wisdom suggests that representations learned end-to-end for a task achieve better performance than semantic representations designed to be interpretable. Thus existing efforts mostly focus on explaining end-to-end networks post-hoc, by grounding their decision in image pixels via an attention map or active patches [12], [13], [14]. In contrast, we learn explainable representations in an end-to-end manner, and our method enhances rather than limits the system's ability to efficiently learn low-shot concepts and transfer to novel visual domains and small-data problems.

This paper builds on our preliminary work [15], and extends it substantially in multiple directions, exploring new concepts and applications to novel scenarios. We summarize these contributions below:

- *Compositionality of Concepts.* We present Recognition as Part Composition (RPC), a general image encoder that is inspired by human cognition and produces image encodings by decomposing an image into parts and then representing each part in compact vocabulary of a few concepts.
- *Low-Shot Generalization.* We show that the RPC encoding is useful in low-shot generalization tasks like few-shot learning, zero-shot learning and visual domain adaptation, and that a simple model using an RPC encoder performs favorably compared to state of the art methods in those tasks.
- *Robustness to Adversarial Attacks.* A classifier with an RPC encoder can be robust to *adversarial* attacks, than a standard DCN. This is because the RPC encoder embeds inputs into a space of discrete concepts, and thus for an attack to be successful, the adversary must modify the input significantly to modify a concept.
- *Interpretability.* We also show that RPC encodings are also interpretable by humans by crowd-sourcing questions regarding our model encodings.
- *Synthesizing Datasets.* Finally, given that RPC encodings are human interpretable we propose another possible application of our model as a synthetic attribute generator for evaluating zero-shot learning methods on new datasets.

We note while mentioning our contributions that the goal of this paper is an exposition of certain nice properties our RPC encodings have. While on different low-shot generalization tasks, we will see that a model using these image representations performs competitively with recent approaches specifically developed for those tasks, the goal is not to show RPC achieves performance better than those task-specific approaches. Rather, our framework should be viewed as exposing the benefits of human-like learning for limited-shot learning problems.

2 Related Work

In the introduction, we discussed how compositionality and grounding objects in a vocabulary of a few simple concepts are key ingredients in human intelligence. Lake *et al*. [9] followed this via a somewhat restrictive model of Bayesian Program Learning. In contrast, our RPC model, while leveraging recent advances in deep neural networks, induces desirable properties that are evident in our understanding of human intelligence. In the sequel, we discuss prior work in three different domains. Note that with the range of research that has gone into each of these problems, this section provides only a flavor of related research and readers are suggested to read review articles on the topics for a more thorough and wider coverage of them (*e.g*. [16], [17]).

Explainability. As mentioned in the introduction a large part of recent explainability research is driven by post-hoc analysis of deep networks. This is because these models although highly non-linear with high dimensional feature spaces, have been the strongest performers on recognition tasks. Many approaches search for prediction explanations in the form of saliency maps. Among them, a common approach is to use gradient magnitudes for different class activations [12], [14], [18]. While this assumes access to the model and hence its gradients, some other approaches like [13] aim to get a saliency map by generating random masks for input images and using the network prediction to determine the closeness of the random mask to the actual saliency map. Another set of approaches tries to assign semantic concepts to hidden neurons in a neural network [19], [20]. Our approach is different from these post-hoc

interpretability analyses in that our image encodings are a composition of a small number of concepts that are interpretable by humans, as we shall see in our experiments. We find that the restrictions that a small number of parts and concepts can create, do not encumber our model's low-shot generalization capabilities, but rather enhance it.

Adversarial attacks and robustness. Szegedy *et al*. [4] first discovered that deep neural networks are vulnerable to adversarial examples, which are only slightly different to correctly classified examples from the original data distribution. The examples on the Imagenet dataset [21] were often so close to the original examples that the two were indistinguishable to humans. Subsequent research led to the development of different kinds of adversarial attacks on images, both in the white-box scenario [22], [23], [24], [25] where adversaries can access the trained model and the black-box scenario [26], [27], where they cannot. Various defense strategies against these attacks have also been developed [28], [29], [30]. A simple white box attack that was found to reliably generate adversarial examples for a wide range of models was the fast gradient sign method (FGSM) [22], and they proposed training of the deep neural networks with such adversarial examples as one possible strategy for defending against them. We evaluated our RPC model using adversarial examples from such an FGSM attack and found it to have much better robustness to them as compared to a deep convolutional network trained on the same task. Note that no adversarial examples were used in the training of the two models.

Low shot generalization. We use this term to refer to a collection of problems with some constraints surrounding availability of labeled data from the target distribution. Note that most of these problems do assume an abundance of some form of training data, but the scarcity arises because the model is tested on its predictions made on data from a different target distribution than the one it was trained on. Information about this target distribution is unavailable or scarcely available to the model during training. We discuss 3 such problems in recognition tasks:

(1) *Unsupervised Domain Adaptation (UDA).* This problem focuses on a task where at inference time, images come from a target distribution or domain that has the same set of classes, but has visually dissimilar images to the ones in the training or source distribution. We specifically focus on the scenario where the model has access to unlabeled images from the target distribution. Ben-David *et al*. [31] provided an upper bound on a classifier's target error consisting of its source error and a divergence between the two distributions. Using this result many approaches were derived that attempt to solve UDA by aligning the two distributions in feature space [32], [33], [34], [35], [36]. Some other approaches like [37], [38] leverage the transductivity of the problem to use pseudo-labeling for unlabeled target domain images. Still others attempt to train generative models to translate source images to target images allowing to train a classifier with these generated target labeled images [39], [40], [41], [42]. In our evaluation of RPC, we utilize a pseudo-labeling approach that is a simpler version of [37].

(2) *Few-Shot Learning (FSL).* This problem tests a model's ability to learn classification using a few labeled examples per class. Typically the model is allowed to be trained on a label abundant training dataset, and at inference time, it "adapts" to the classification problem defined by the few labeled examples. Some common approaches here have attempted to learn a feature space, with a class-defined neighborhood in a distance metric (say ℓ_2). This allows for a non-parametric "adaptation" at inference time, using simply a nearest neighbors classifier in the feature space [2], [43], [44]. Meta-learning or learning-to-learn (or more aptly for this problem, learning-to-adapt) is an approach also quite quite popular [3], [45], [46]. Finally, generative models like GANs [47] have also found use in this problem, with approaches augmenting the few-labeled examples with generated ones for classifier learning [48], [49], [50]. For our RPC model, we do nearest neighbor classification at inference time similar to [43], but we train the model for classification on the training set using a 2 layer MLP classifier on the RPC encodings.

(3) *Zero-Shot Learning (GZSL).* This problem, at inference time, requires a model to make class predictions on images where certain classes (simply termed "unseen classes") have no examples in the training set. The classifier accesses some semantic vector representation of classes, to relate an example of an unseen class to its semantic vector. Some common approaches include [51], [52], [53] which learn feature embeddings that directly map the visual domain to the semantic domain and infer classifiers for unseen classes. Some other approaches like [44] learn a distance metric between the images and the semantic vectors of the class they belong to. Similar to both the above problems, some approaches have also attempted to synthesize unseen class images in the new environment from the given semantic attributes [54], [55], [56], [57]. We trained the RPC model to learn a simple distance metric between the image feature space and the semantic vectors for this problem.

Recognition using object parts has been an ingredient in multiple recognition approaches over the years. Ullman *et al*. [58], showed that information maximization with respect to classes of images resulted in visual features that were of intermediate complexity (neither too high, nor too low), indicating these were optimally beneficial for classification. These would result in selecting components like eyes, mouth, *etc*. in facial images and tyres, bumper, windows *etc*. in images of cars, further motivating recognition by components. Object geometries were paid particular attention to in Deformable Part Models [59], [60] which learned class models using both part features and their geometries, and assigned probability scores for object presence based on both how well the features matched as well as how much the geometry was deformed. Originally the method used Histogram of Gradients (HoG) features, but subequently, DPMs were also developed with deep CNNs [61], [62]. In fine-grained recognition multi-attention has been used in prior works for recognizing different object parts in images [11]. Attention based approaches [63], [64], [65] have also been found to be able to discover different semantically meaningful parts like objects and background components in images of synthetically generated 3D scenes. RPC uses attention maps corresponding to different object parts as well, along with specific priors that incorporate properties which make the RPC encoding interpretable (described in sections 3 and 4).

3 Recognition as Part Composition

Our approach learns an image classifier by learning an intermediate feature representation that we refer to simply as the RPC encoding. We describe the notation used in our exposition: $x \in \mathcal{X} \subset \mathbb{R}^D$ denotes an image and $y \in \mathcal{Y}$ denotes a class label for the image. We will use $p(x, y)$ to denote the joint distribution over image, label pairs.

Inductive Bias. Our proposed approach injects an inductive bias, to factorize a task posterior $p_t(y|x)$ into a task agnostic mapping function, $\pi : \mathcal{X} \rightarrow \Pi$ or the RPC encoder, and a task-specific posterior acting on $\pi(x)$ for $x \in \mathcal{X}$. The mapping π involves learnable parameters as we shall see in Sections 3.1 and 3.2. Thus [1]

$$p_t(y|x) = p_t(y|\pi(x)) \tag{1}$$

The above structural decompostion is useful only if the latent variable $\pi(x)$ which is a task independent embedding, does the "heavy-lifting" allowing to distill predictive information from the training distribution to the target task distribution. Note that the decomposition itself is not special and it is not clear whether we can learn maps that allow for such information transfer. For instance, if π is the identity map, no information is distilled from the training task distribution for the target tasks. We however, draw on insights from human cognition to propose our RPC encoder, which, as we shall see from our experiments, does allow for effective knowledge transfer from training tasks.

An overview of the model is shown in Fig 2. Given an image $x \in \mathcal{X}$, the model first identifies M parts in the image and extracts features z for each part using the "Part feature extractor" (Sec. 3.1). It then uses the Part-type likehood encoder (Sec. 3.2), which learns certain prototypical representations for the different part types, to generate an encoding for each part as a convex combination of these prototypes. These features denoted using π is the RPC encoding of the image. Finally, depending on the task, a different task specific predictor V (Sec. 5) is chosen, which outputs a class prediction $\widehat{y} = V(\pi(x))$ for the image x, and the model is trained end-to-end.

3.1 Part Feature Extractor

Inspired by [11], we use a multi-attention convolutional neural network (MACNN) to map input images into a finite set of part feature vectors, $z_m \in \mathbb{R}^C$. It contains a module E which, for input image x generates a convolutional feature representation $E(x) \in \mathbb{R}^{W \times H \times C}$ with width W, height H and C channels. A second module G, then uses this representation to produce a set of weights for combining different channels in $E(x)$ to get an attention map per part. Specifically $G(E(x)) \in \mathbb{R}^{M \times C}$, where M is the number of different parts in the image (a set hyperparameter), and the attention map $A_m \in \mathbb{R}^{W \times H}$ for the m^{th} part is computed as

$$A_m(x) = \text{normalize}\left(\text{sigmoid}\left(\sum_{c \in [C]} G_{m,c} \times E_c(x)\right)\right) \tag{2}$$

where the sum is over the channel dimension, the notation $G_{m,c}$ drops the dependence on $E(x)$ for conciseness and the operation "normalize" divides all elements by a constant to make the elements of A_m sum to 1. The m^{th} part feature $z_m \in \mathbb{R}^C$ is then calculated as:

$$z_{m,c} = \sum_{w,h} [A_m(x) \odot E_c(x)]_{(w,h)}, \quad \forall c \in [C] \tag{3}$$

where $\odot$ is element-wise multiplication and the sum is over the width and height dimensions for each individual channel. We parameterized $E(\cdot)$ using a ResNet-34 backbone (till the $conv5$ block), and $G(\cdot)$ using a fully-connected layer.

As we see from Eq. 3, z_m can be decomposed into an element-wise product of the attention maps ($A_m(x)$) and the extracted convolutional features ($E(x)$), followed by a sum over the width and height dimensions. For the attention maps to correspond to different parts of an image, we would like them to be compact within a given part and diversely spread over different parts. To encourage this behavior, we use the following two criteria:

- **Compactness.** This criterion encourages the attention map for each part to be concentrated around a peak value and is defined as:

$$L_{com}(A_m) = \sum_{w,h} A_m^{w,h} [\|w - w^*\|^2 + \|h - h^*\|^2] \tag{4}$$

  where $A_m^{w,h}$ is the amplitude of A_m at coordinate (w, h), and (w^*, h^*) is the coordinate of the peak value of A_m. Note that the loss penalizes high values of A_m that are away from its peak value.
- **Diversity.** This criterion encourages the attention maps of the different parts to have peak values at different locations. It is defined as :

$$L_{div}(A_m) = \sum_{w,h} A_m^{w,h} [\max_{n:n \neq m} A_n^{w,h} - \zeta] \tag{5}$$

  where ζ is a small margin to ensure training robustness. Note that this loss is high when there are portions of the image where two different attention maps have high values.

Combining the two losses, we get

$$\ell_{part}(x) = \sum_m (L_{com}(A_m(x)) + \lambda_1 L_{div}(A_m(x))) \tag{6}$$

where λ_1 is a hyperparameter balancing the two criteria.

3.2 Part-type Likelihood Encoder

We assume an underlying mixture of gaussians distribution which z_m comes from:

$$z_m \sim \sum_{k \in [K]} \pi_{k,m} \mathcal{N}(D_{k,m}, \gamma^2 I),$$

where $\pi_{k,m}$ represents the likelihood that the sample belongs to the Gaussian component k with mean $D_{k,m}$ in part m. This means $D_{k,m}$ are prototypical part-types, a convex

1. Note that although we make these statements for the true task distribution, we can only impose the bias in the predicted posterior. RPC's generalization power leads us to believe that the true posterior for our evaluation tasks may follow a similar factorization.

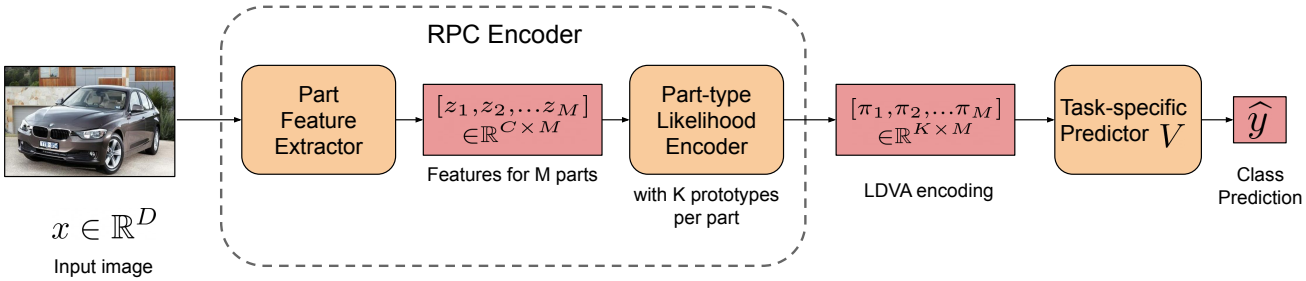


Fig. 2: **Recognition as Part Composition (RPC).** For an input image x, the part feature extractor decomposes x into M parts and extracts associated features z_m. The Part-type Likelihood encoder then encodes each part feature as a low-dimensional encoding π_m by projecting the features onto a dictionary of part prototypes automatically discovered by the RPC model. π_m is then used as input to train task-specific predictor models for GZSL, FSL and UDA. The model is trained end-to-end for each task

combination of which using the weights $\pi_{k,m}$ gives the mean of the Gaussian mixture for z_m. We also refer to these prototypical part-types $D_{k,m}$ simply as prototypes for part m. The above entails an autoencoder implementation, where we can use a projection matrix $P_m \in \mathbb{R}^{K \times C}$ to transform $z_m \in \mathbb{R}^C$ to the lower dimensional $\pi_m \in \mathbb{R}^K$ (Note that we use a $K \ll C$):

$$\pi_m(x) = \phi(P_m z_m(x)) \tag{7}$$

where $\phi(\cdot)$ is a softmax function.

The inverse of this transform $D_m^\top \in \mathbb{R}^{K \times C}$ is a matrix containing the prototypes $D_{k,m}$ along its rows and approximately satisfying:

$$z_m(x) \approx D_m^\top \pi_m(x) \tag{8}$$

Treating $P_m, D_m \in \mathbb{R}^{K \times C}$ as parameters of an autoencoder, the training objective to learn these low dimensional encodings π_m for part feature z_m is as follows:

$$\ell_{ae}(x) = \sum_m (\|z_m(x) - D_m^\top \phi(P_m z_m(x))\|^2 + \lambda_2 \|P_m\|^2 + \lambda_3 \|D_m\|^2). \tag{9}$$

where λ_2 and λ_3 are the weights for the regularizers.

4 Explainability and Explainable Models

As mentioned in the introduction explainable models are preferred in many practical scenarios. Before moving on to specific low shot generalization tasks, we elaborate how explainability is a consequence of the RPC encoder and the way it is trained, and even before doing that, we try to define explainability in models in the form of certain properties they should have.

Definition 1 (Explainable Model). An Explainable Model is one that has the following three properties:

1) *Compact Vocabulary.* Its prediction can be associated with a small number of finite discrete concepts. While, finiteness leaves the actual number of such concepts arbitrary, what we require is motivated by humans posed with choosing among multiple choices. In particular, it is reasonable to assume that for each part we have about half dozen prototypes, and each of these prototypes exhibit about 4-6 variants, and finally there are about 4-6 total number of parts.

2) *Consistency.* It assigns importance values to each of these concepts such that its final output is directly related to these values; the concepts and importance values together constitute an *explanation*.

3) *Meaningfulness:* The concepts must be semantically meaningful to or recognizable by a human either immediately or after seeing a few examples.

As per the above properties, our RPC model has a compact vocabulary because it breaks up an image into parts such that each part is representable by only a small number of prototypes/concepts. There is consistency, since the model's outputs are a simple function (using the task specific predictor; see Sec 5) of the RPC encodings which are simply a set of importance values representing the likelihood that each part comes from a certain concept/prototype. The only component remaining is meaningfulness, which we demonstrate by crowd-sourcing answers to certain questions about example encodings that our model produces for different images (See Sec. 7). Using ℓ_{task} as a proxy for some loss function related to a classification task (assuming additional possibly parametric functions acting on the RPC encodings to generate a class prediction), when the RPC model gets trained with the loss function $\ell_{part} + \ell_{ae} + \ell_{task}$, it attains the properties mentioned above, making it explainable.

5 RPC for Low-Shot Generalization

Using the RPC encoder, a classifier can be defined as a simple multi-layer perceptron (or even a linear classifier) that operates on the RPC encodings π. The encodings π_m from all parts are concatenated before feeding into the classifier. We specifically used a 2-layer MLP for simply learning a source/training dataset classifier, which we use for evaluating the encodings' interpretability and its adversarial robustness (sections 7 & 6.4 respectively). Additionally, we evaluated RPC on 3 different low shot generalization tasks, each with its constraints and hence requiring a separate task-specific predictor that we describe next.

5.1 Unsupervised Domain Adaptation

This problem involves tackling image classification in a target domain, where the joint distribution of the image-

label pairs is different from the training data(*i.e.* $p_S(x,y) \neq p_T(x,y)$, where p_S is the distribution of the source domain set of labeled training images available to the learner). The problem specifies that p_S and p_T share the same set of labels. An example where this problem arises is when a learner is expected to recognize real images of an object, having only access to labeled hand-sketched depictions of the same.

Concretely, the learner gets access to a labeled set of images $\mathcal{D}_S = \{(x_i^s, y_i^s) \mid i \in [n_s]\}$, and an unlabeled set of images $\mathcal{D}_T = \{x_i^t \mid i \in [n_t]\}$, and the goal of the learner is to predict class labels for images in the set $\mathcal{D}_T$.

The task-specific predictor in this case, is a simple 2 layer MLP classifier on top of the RPC encodings, $V : \mathbb{R}^{M \cdot K} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$, where $|\mathcal{Y}|$ is the number of classes in the set of possible labels $\mathcal{Y}$. We first train V along with the rest of the model using only the source domain images to minimize the objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_S}[\ell_{part}(x) + \ell_{ae}(x) + CE(V(\pi(x)), o(y))], \quad (10)$$

where $CE(\cdot,\cdot)$ is cross-entropy and $o(y)$, is a one-hot representation of class label y. Similar to prior work [37], [38], we then pseudo-label the target domain images $x \in \mathcal{D}_T$ with $\tilde{y} = \arg\max_y V(\pi(x))_y$, where $V(\pi(x))_y$ is the y-th element in the $V(\pi(x))$ vector and create a set $\widetilde{\mathcal{D}}_T = \{(x, \tilde{y} | x \in \mathcal{D}_T)\}$. We then re-train the entire model to minimize the objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_S\cup\widetilde{\mathcal{D}}_T}[\ell_{part}(x) + \ell_{ae}(x) + CE(V(\pi(x)), o(y))], \quad (11)$$

5.2 Few Shot Learning

Few shot learning (FSL) tests a classifier's ability in learning to classify from a few labeled examples. Evaluation typically involves a "base training set" $\mathcal{D}_S$ of image-label pairs, that a learner can use for training. At inference time, the learner is provided with "episodes" with a small number α of classes and few labeled images (or support images), β in number, per class. A certain number of query images are also provided, and the learner is expected to classify them, into one of the α classes, using the support images to possibly adapt itself. This is called an α-way β-shot learning task. The classes in $\mathcal{D}_S$ are typically disjoint from classes used in episodes for evaluation.

We first train the model with a 2-layer MLP classifier as V, on the following loss function

$$\mathbb{E}_{(x,y)\in\mathcal{D}_S}\ell_{part}(x) + \ell_{ae}(x) + CE(V(\pi(x)), o(y)) \quad (12)$$

where CE and o are the same as used in Eq. 10.

At inference time, we then use the model as a non-parametric nearest neighbors classifier. Let an evaluation episode be notated as follows : $\mathcal{D}_{sup} = \{(x,y) \mid y \in [\alpha]\}$ is the support set of image label pairs, with exactly β images per class and $\mathcal{D}_q = \{x_i \mid i \in [q]\}$ is the query set with only images and no labels. For an $x \in \mathcal{D}_q$, the model predicts the label

$$\widehat{y} = \underset{y\in[\alpha]}{\arg\min} ||\pi(x) - \overline{\pi}_y||_2 \quad (13)$$

where $\overline{\pi}_y$ is the average RPC encoding of the class y images in $\mathcal{D}_k$, *i.e.*

$$\overline{\pi}_y = \frac{\sum_{(x',y')\in\mathcal{D}_{sup}} \mathbb{1}[y = y']\pi(x')}{\sum_{(x',y')\in\mathcal{D}_{sup}} \mathbb{1}[y = y']}$$

5.3 Generalized Zero-shot Learning

Zero-shot learning (ZSL) involves a recognition problem where the class labels for training images have no overlap with the ground truth labels of images seen at inference time. What accompanies this problem is a semantic vector for each class label, that carries meaning for a learner to leverage when it tries to classify images of classes it has not encountered during training. We note here that we will also use the term semantic attributes of a class to refer to this semantic vector, since quite often this vector comes from a labeling of attributes in a dataset.

Specifically, the problem involves a training set of image-label pairs such that $\mathcal{D}_S$ for $(x,y) \in \mathcal{D}_S, y \in \mathcal{Y}^s \subset \mathcal{Y}$. The set $\mathcal{Y}^s$ is often termed "seen classes", and the set $\mathcal{Y}^u := \mathcal{Y} \setminus \mathcal{Y}^s$, "unseen classes". As mentioned above, each class label $y \in \mathcal{Y}$ has a corresponding semantic vector σ_y. Additionally we notate with Σ^s, Σ^u and Σ, the set of semantic vectors corresponding to $\mathcal{Y}^s, \mathcal{Y}^u$ and $\mathcal{Y}$. Zero-shot learning refers to the problem where at inference time, images that the model makes predictions on, have classes in $\mathcal{Y}^u$ and the model makes use of the set Σ^u to make this prediction. A modification of this, called Generalized Zero-shot Learning (GZSL), at inference time, asks the model to predict class labels for images that could be from any class in the set $\mathcal{Y}$, using the set of semantic vectors Σ. ZSL is a simpler problem compared to GZSL since in the former, the learner at inference time, has the knowledge that the correct answer is in the set $\mathcal{Y}^u$ which is strictly smaller than $\mathcal{Y}$ in cardinality. We use GZSL for evaluation. The task predictor V is a 2-layer neural network, and the objective function minimized by the model is

$$\mathbb{E}_{(x,y)\in\mathcal{D}_S}[\ell_{part}(x) + \ell_{ae}(x) + \ell_{GZSL}(x,y)] \quad (14)$$

where

$$\ell_{GZSL}(x,y) = \sum_{\substack{y'\in\mathcal{Y} \\ y'\neq y}} \max(\eta + (\sigma_{y'} - \sigma_y)^\top V(\pi(x)), 0) \quad (15)$$

is a hinge loss treating class semantic vectors as the weights of a maximum margin classifier on $V(\pi(x))$.

At inference time, the class prediction for an image x is made as $\widehat{y} = \arg\max_{y\in\mathcal{Y}} \sigma_y^\top V(\pi(x))$.

5.4 Implementation Details

Recall from Sec. 3.1, that the feature extractor module $E(\cdot)$ is parametrized by a Resnet-34 (up to the $conv5$ block) and $G(\cdot)$ is a fully connected layer. The number of parts M and the number of prototypes K in each part are hyper-parameters. In our experiments, M is set to 4 and K is set to 16, if not specified. Also, the softmax function ϕ in Eq 7 has a temperature 100 (selected using the validation set accuracy of few-shot classification on *mini*Imagenet; the sweep showed that accuracy did not change a lot for lower

temperatures down to 10, but degraded quickly below a temperature of 1). ζ in Eq. 5 is empirically set to a value of 0.02 that achieves robust training. We set λ_2, λ_3 to $1e-3$ in all the experiments. Model optimization for all problems is done in an alternating manner where in step (A) we optimize the parameters of $G(\cdot)$ on the only the objective ℓ_{part}, and in step (B) we freeze these weights and optimize the rest of the model parameters on the entire (task-specific) objective function.

For FSL and for DA, an input image size of 224 $\times$ 224 pixels is used, and λ_1 in Eq.(6) is set to 2. For GZSL, our model takes input image of size 448 $\times$ 448 and λ_1 is set to 5. The task-specific predictor $V(\cdot)$ for both GZSL and DA is implemented by a two FC-layer neural network with ReLU activation, the number of neurons in the hidden layer is set to 32. Note that the details of training for each task are mentioned in Sec 6.

6 Experiments

6.1 Domain Adaptation

Datasets. We evaluated the RPC model in unsupervised domain adaptation task between three digits datasets: MNIST [66], USPS and SVHN [67]. Each dataset contains 10 classes of digit numbers (0-9). MNIST and USPS are handwritten digits while SVHN is obtained from house number in google street view images.

Setup. We follow the same protocol as in [33], where the three adaptation scenarios used for evaluation are ($\mathcal{D}_S \rightarrow \mathcal{D}_T$): MNIST→USPS, USPS→MNIST, and SVHN→MNIST. In the experiments, two variants of our model are evaluated: (1) **RPC(source π)**: During training, the model is purely learned from source data, which corresponds to the model minimizing the objective in Eq. 10. This model does not utilize any information from the unlabeled target data in the training. (2) **RPC(joint π)**: This model learns the visual encoder from the joint dataset $\mathcal{D}_S \cup \widetilde{\mathcal{D}}_T$ by minimizing the objective in Eq. 11. Recall that $\widetilde{\mathcal{D}}_T$ contains images from the target domain with pseudo-labels produced by the "source π" model.

Training Details. RPC (source π) is trained on the source domain dataset, as described above. The learning rate for step (A) and step (B) is 1e-6 and 1e-5. The training epochs are set to be 40, 20, and 40 on MNIST, USPS, and SVHN, respectively. For joint π, we first initialize with weights with those of the source-π model. Next, the model is trained on the joint dataset $\mathcal{D}_S \cup \widetilde{\mathcal{D}}_T$ for 10 epochs. The learning rate for step (B) is modified to 1e-6.

Results. Target classification accuracies for different scenarios are reported in *Table 1*. Our RPC approach fares surprisingly well, especially when we use target pseudo-labels to train, *i.e.*, RPC(Joint π). It is worth noting that Mean-Teacher uses a data augmentation technique which models the distortion in target data. Evidently, this technique for the specific dataset is powerful enough that the reported accuracies are higher than those reported for a fully supervised model on target data. In contrast, our method learns a static universal representation for both source and target domain, which does not require the prior knowledge on the domain distortion. The data augmentation is complementary to our model and it can be expected that our model can also benefit from the increased training data.

Methods	M → U	U → M	S → M
Source Only*	75.2	57.1	60.1
Gradient reversal* [32]	77.1	73.0	73.9
Domain confusion* [68]	79.1	66.5	68.1
CoGAN* [69]	91.2	89.1	-
ADDA* [33]	89.4	90.1	76.0
DTN [40]	-	-	84.4
UNIT [70]	96.0	93.6	90.5
CyCADA [39]	95.6	96.5	90.4
MSTN [71]	92.9	-	91.7
ADR [35]	91.3	91.5	94.1
MCD [36]	94.2	94.1	96.2
CDAN [34]	95.6	**98.0**	89.2
Mean-Teacher [72]	**98.3**	**99.5**	**99.3**
SHOT [73]	97.9	**98.0**	**98.9**
RPC (source π)	94.8	96.1	82.4
RPC (joint π)	**98.8**	96.8	95.2

TABLE 1: Domain adaptation classification results. M = MNIST, U = USPS, S = SVHN. The highest accuracy is in **red** color and the second is in **blue** (better viewed in color). Self-ensembling, unlike other methods, leverages data-augmentation and reports accuracy numbers that are evidently higher than those obtained in the fully supervised case for $U \rightarrow M$, $S \rightarrow M$. *Numbers reported in [33].

The results demonstrate the benefits of proposed RPC representation. Specifically, in the same domain, the distance or dissimilarity between RPC encodings for different classes are large enough to learn a good classifier. Meanwhile, the representations of the same class from different domains are much more similar than the high-dimensional features of a DCN (this can be seen by comparing target accuracies of "Source only" and the "RPC(source π)" models), resulting in a similar distribution of features across the two domains. The classifier trained on the source domain is thus able to be used on target domain data. This also means that RPC is more tolerant to visual distortions. In Fig. 3(a-b), we see that the encodings learnt by RPC(joint π) model are quite similar for images of the same class across the two domains.

Source vs. Joint π. Comparing accuracies in Table 1, we see that RPC(joint π) has better performance, showing that cross-entropy loss using pseudo-labels on the target domain in Eq. 11 helps. This model benefits more over RPC(source π) when the domain shift is severe (*e.g.* S → M). This is also clear from a visual representation of the encodings in Figure 3(c), where we see a larger difference in encodings across the two domains using RPC(source π) than RPC(joint π).

6.2 Few-Shot Learning

Datasets. We first evaluate the few shot learning performance of the proposed model on three benchmark datasets: Omniglot [9], *mini*ImageNet [2] and CUB [74]. Omniglot consists of 1623 characters from 50 alphabets. Each character (class) contains 20 handwritten images from people. *mini*ImageNet is a subset of ImageNet [75] which contains 60,000 images from 100 categories. The CUB dataset contains 200 classes each corresponding to a different species of bird and a total of 11,788 images.

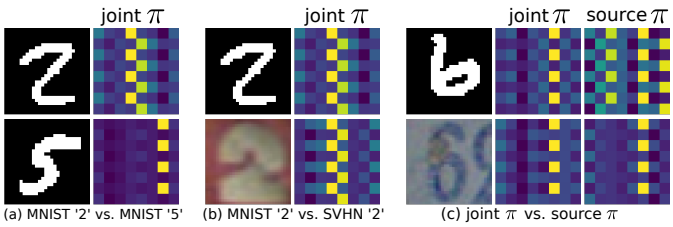


Fig. 3: Proposed RPC encoding π on digit datasets. The 64-dimensional π vector is reshaped to a 8×8 matrix for better visualization. For all three examples (a-c), π is trained for SVHN→MNIST experiment.

Setup. We follow the same protocol as [44] for the first two benchmarks. For Omniglot, the dataset is augmented with new classes through 90°, 180° and 270°rotations of existing characters. 1200 original classes plus rotations are selected as training set and the remaining 423 classes with rotations are test set. For *mini*ImageNet, the dataset is split into 64 training, 16 validation and 20 testing classes. For the CUB dataset, we followed [76] and used 100 classes for training, 50 for validation and 50 for testing. The model is only trained on the training set and the validation set is used for development.

We evaluate the 5-way accuracy on *mini*ImageNet and 5-way plus 20-way accuracy on Omniglot. 1-shot and 5-shot learning performance is evaluated in each setting. For α-*way* β-*shot* learning, in each test episode, α classes will be randomly selected from the test set, then k samples will be drawn from these classes as support examples, and 15 examples will be drawn from the rest images to construct the test set. We run 1000 and 600 test episodes on Omniglot and *mini*ImageNet, respectively, to compute the average classification accuracy.

Training Details. Our model is trained for 80, 10 and 100 epochs on Omniglot, *mini*ImageNet and CUB, respectively. The learning rate for step (A) is set to 1e-6, and the learning rate of step (B) is 1e-4 for Omniglot. The learning rate is set to 1e-4 for *mini*ImageNet and 5e-4 for CUB. On *mini*ImageNet and CUB, we use $K = 256$ prototypes for few-shot classification. For few-shot classification on *mini*ImageNet, we used a Resnet-18 feature extractor as opposed to the Resnet-34 used for other experiments, and the weights for the feature extractor $E(\cdot)$ are pre-trained on an appropriate Imagenet subset following [77], while they are randomly initialized for Omniglot and CUB.

Results. Few shot learning results for the different benchmarks are reported in *Tables* 2, 3 and 4. Again, our RPC approach, which simply trained a classifier on base classes and used the learned encoder, along with nearest neighbors classification, is competitive on the benchmark and does not lag by much compared to the most recent state of the art methods on this task. We interpret this as a result of the low intra class and the high inter class divergences that are inherently exhibited by the encodings generated by our RPC trained classifier, allowing for the use of nearest neighbors for few-shot inference. The high dimensional feature spaces of DCNs do not exhibit these nice properties, and need specific training strategies to function well in this problem.

Methods	**5-way Acc.**		**20-way Acc.**	
	1-shot	*5-shot*	*1-shot*	*5-shot*
Matching Net [2]	98.1	98.9	93.8	98.5
MAML [45]	98.7	**99.9**	95.8	98.9
Prototypical Net [43]	98.8	99.7	96.0	98.9
Relation Net [44]	99.6	**99.8**	97.6	99.1
GCR [78]	**99.7**	**99.9**	**99.6**	99.3
TapNet [79]	-	-	98.1	**99.5**
DCN6-E [80]	**99.9**	**99.9**	**99.1**	**99.6**
RPC	98.9	**99.8**	96.5	99.3

TABLE 2: Few-shot classification accuracy on Omniglot. In each column, **red**=highest and **blue**=2^{nd} highest accuracy

Methods	*1-shot*	*5-shot*
Baseline++ [76]	51.87	75.68
ProtoNet [43]	54.16	73.68
MetaOptNet [81]	64.09	80.00
New-Meta [82]	63.17	79.26
FEAT [83]	66.78	82.05
DeepEMD [84]	65.91	82.41
DPGN [85]	67.77	84.60
RENet [86]	67.60	82.58
ZN [87]	67.35	83.04
MeTAL [88]	66.61	81.43
BML [89]	67.04	83.63
ECSIER [90]	67.28	84.78
ECKPN [91]	**70.48**	**85.42**
Simple-CNAPS [77]	**77.40**	**90.30**
RPC	63.92	84.57

TABLE 3: Few-shot accuracy in % on *mini*ImageNet. In each column, **red**=highest and **blue**=2^{nd} highest accuracy

Methods	*1-shot*	*5-shot*
Baseline++ [76]	68.00	84.50
ProtoNet [43]	72.94	87.86
SimpleShot [92]	68.90	84.01
DN4† [93]	70.47	84.43
COMET [94]	72.20	87.60
MetaOptNet* [81]	75.15	87.09
DeepEMD [84]	75.65	88.69
AFHN [95]	70.53	83.95
BSNet [96]	69.61	83.24
MTL* [97]	73.31	82.29
VFD* [98]	79.12	**91.48**
FOT [99]	72.56	87.22
FRN [100]	**83.16**	**92.59**
RENet [86]	**79.49**	91.11
RPC	75.01	89.61

TABLE 4: Few-shot accuracy in % on CUB. If not specified, the results reported by the original paper. *: reported in [98]. †: results are obtained using the authors' implementation. In each column, **red**=highest and **blue**=2^{nd} highest accuracy

6.3 Generalized Zero-Shot Learning

Datasets. The performance of our model for GZSL is evaluated on three commonly used benchmark datasets: *Caltech-UCSD Birds-200-2011* (CUB) [74], *Animals with Attributes 2*

Methods	CUB			AWA2			aPY		
	U	S	H	U	S	H	U	S	H
SJE [101]	23.5	59.2	33.6	8.0	73.9	14.4	3.7	55.7	6.9
SAE [102]	7.8	54.0	13.6	1.1	82.2	2.2	0.4	**80.9**	0.9
SSE [103]	8.5	46.9	14.4	8.1	82.5	14.8	0.2	78.9	0.4
ALE [104]	23.7	62.8	34.4	14.0	81.8	23.9	4.6	73.7	8.7
SYNC [105]	11.5	70.9	19.8	10.0	90.5	18.0	7.4	66.3	13.3
PSRZSL [106]	24.6	54.3	33.9	20.7	73.8	32.3	13.5	51.4	21.4
SP-AEN [107]	34.7	70.6	46.6	23.3	90.9	37.1	13.7	63.4	22.6
CE-GZSL [108]	63.9	66.8	65.3	63.1	78.6	70.0	-	-	-
GEM-ZSL [109]	**64.8**	**77.1**	**70.4**	64.8	77.5	70.6	-	-	-
Generative ZSL									
GDAN [110]	39.3	66.7	49.5	32.1	67.5	43.5	**30.4**	75.0	**43.4**
CADA-VAE [111]	51.6	53.5	52.4	55.8	75.0	63.9	-	-	-
LisGAN [112]	46.5	57.9	51.6	-	-	-	34.3	68.2	45.7
f-CLSWGAN [113]	43.7	57.7	49.7	-	-	-	-	-	-
SE-GZSL [55]	41.5	53.3	46.7	**58.3**	68.1	62.8	-	-	-
DA-GZSL [114]	47.9	56.9	51.8	-	-	-	-	-	-
Trans-ZSL									
DIPL [115]	41.7	44.8	43.2	-	-	-	-	-	-
TEDE [116]	**54.0**	62.9	58.1	**68.4**	**93.2**	**78.9**	29.8	**79.4**	43.3
STHS [117]	**77.4**	74.5	**75.9**	**94.9**	**92.3**	**93.6**	-	-	-
RPC	33.4	**87.5**	48.4	41.6	91.3	57.2	24.5	72.0	36.6
RPC + CS	59.2	74.6	66.0	54.6	87.7	67.3	**41.1**	68.0	**51.2**

TABLE 5: GZSL results on CUB, AWA2 and aPY. U = unseen classes, S = seen classes, H = harmonic mean. The accuracy is class-average Top-1 in %. In each column, **red**=highest and **blue**=2^{nd} highest accuracy

(AWA2) [118] and Attribute Pascal and Yahoo (aPY) [119]. CUB is a fine-grained dataset consisting of 11,788 images from 200 different types of birds. 312-dimensional semantic attributes are annotated for each category. AWA2 has 37,222 images from 50 different animals and 85-dim class-level semantic attributes. aPY contains 20 Pascal classes and 12 Yahoo classes. It has 15,339 images in total and 64-dimensional semantic attributes are provided. We did not choose another popular GZSL benchmark dataset, SUN [120], for the reason that the scene images in SUN are not typical objects that can be decomposed into our part-prototype hierarchy.

Setup. It has been shown [118] that the conventional ZSL setting is overly optimistic because it leverages absence of seen classes at test-time and there is consensus that methods should focus on the generalized ZSL setting. We thus, use GZSL for our evaluations. Following the protocol in [118], we evaluated the average-class Top-1 accuracy on unseen classes (U), seen classes (S) and the harmonic mean (H) of S and U.

It has been observed that in GZSL, a classifier trained using seen class images often predicts output class probabilities that are higher for seen classes than unseen [121], which results in poor performance. Calibrated Stacking(CS) is proposed in [121] to balance the performance between seen and unseen classes by calibrating the scores of seen classes. Hence, in addition to our original model, we also evaluated with CS (denoted as RPC+CS in Table 5). The parameters for CS were chosen using cross validation.

Training Details. Our models are trained for 120, 100 and 110 epochs on CUB, AWA2 and aPY, respectively. The learning rate for step (A) is set to 1e-6 and that for step (B) is set to 1e-5.

Methods for Comparison. We list here, GZSL methods that we compared RPC with in Table 5. Comparisons are not all apples-to-apples since some of these approaches use different assumptions: (1) Methods in the top section, learn a compatibility function between the visual and semantic representations: SJE [101], ALE [104], SAE [102], SSE [103], SYNC [105], PSRZSL [106], SP-AEN [107], CE-GZSL [108], and GEM-ZSL [109]. **Our method also uses this strategy.** (2) Generative model based methods (*Generative-ZSL*) synthesize unseen examples or features using generative models like GAN and VAE thus requiring unseen class semantics at training time: GDAN [110], CADA-VAE [111], 3ME [122], SE-GZSL [55], LisGAN [112], f-CLSWGAN [113], and DA-GZSL [114]. (3) Transductive ZSL methods (*Trans-ZSL*) methods work in a transductive setting which allows the model access to unlabeled images from unseen classes during training: DIPL [115], TEDE [116], and STHS [117].

Results. Results for GZSL are in Table 5. Without calibrated stacking, comparing the harmonic mean (H) accuracy, we see RPC outperforms all other compatibility function based methods (the first section of the table). After the scores are calibrated, our model (RPC+CS) obtains 66.0%, 67.3% and 51.2% for the harmonic mean, respectively, which outperforms other approaches compared to, except TEDE on AWA2.

It is worth noting that, the *Generative ZSL* and *Trans-ZSL* methods always obtain higher accuracy than compatibility function methods, except for our models. This is because the generative and trans-ZSL methods have access to additional information of unseen classes during training. However, this assumption is too optimistic in a real world ZSL scenario since it is unlikely to have full knowledge of all unseen categories at training time. In contrast, our models can be applied in the scenario where novel classes may only appear at test time. Still, by only leveraging seen classes knowledge, our RPC model obtains competitive and sometimes even better performance than generative and trans-ZSL methods.

The success of our model can be attributed primarily to

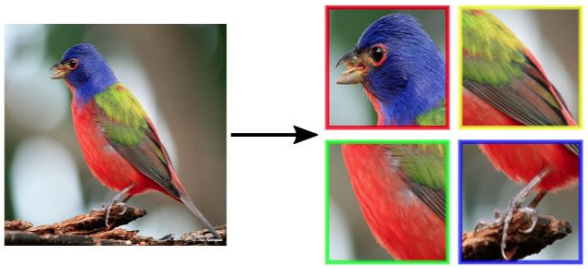


Fig. 4: Example parts recognized by our model in the image of a "Painted Bunting" from the CUB dataset. These parts closely correspond to common semantic attributes labeled in the dataset like "crown color", "feet shape" etc.

the proposed RPC encodings, which close the gap between the images and their semantic attributes. For example, in Figure 4, we visualize the part attentions discovered by our model and several semantic attributes for the class *'Painted_Bunting'* in CUB dataset. Our model learns the part areas around "head", "wing", "body", and "feet", which correspond to most semantic attribute annotations in the dataset (e.g. crown color:blue, wing color:green, etc.). Using the RPC encodings, our visual attributes mirror the representation of semantic vectors, thus mitigating the large gap between the semantic attributes and high-dimensional visual features learned by DCNs.

6.4 Robustness to Adversarial Attack

As mentioned in Sec. 2, Szegedy *et al*. [4], first discovered the susceptibility of deep neural networks to such attacks. In this section we show that our RPC encodings are inherently more robust to adversarial attacks than a deep convolutional classifier trained for the same task. We demonstrate this by choosing a simple FGSM attack [22] (as described next) on models trained for the classification task. No techniques were used in the compared models to specifically train them to be adversarially robust. Note that a range of different attacks and defences have been developed since [4], but a simple FGSM attack, serves the purpose of demonstrating that our RPC model learns encodings which are less susceptible to small adversarial image distortions.

Datasets. We evaluated on two fine-grained classification datasets, CUB [74] and Stanford Car [123]. The Car dataset contains 16,185 images from 196 different car models. The accuracies on the official test splits are reported.

Adversarial Attack. We use Fast Gradient Sign Attack (FGSA) [22] as the attacker. FGSA is a white-box attacker that has full knowledge and access to the model. Albeit simple, it is a powerful adversarial attack that affects a wide range of classification models. For an image x, FGSA generates adversarial perturbation η by calculating the gradient over the input x w.r.t. the cost $J(\theta, x, y)$ used to train the model:

$$\eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \tag{16}$$

where θ is the parameters of the model, and ϵ controls the intensity of the perturbations. The perturbed image $x + \eta$ often looks visually similar to x to the human eye, but different models misclassify $x + \eta$, even when their prediction on x is correct (see Fig 5c).

In our experiments $J(\cdot)$ is implemented by cross-entropy loss for all the models. We evaluate the test accuracy under different distortion level ϵ.

Models Compared. We compare the robustness of RPC with two baseline models:

- BS-1 is an adapted ResNet-34 model that has a two-layer MLP as the classifier. It does not have the multi-attention module G and the part-type likelihood encoder compared to our model. BS-1 is trained using a standard cross-entropy loss.
- BS-2 has the same MACNN architecture as our model, but the part-type likelihood encoder is absent. The part features z_m for different parts are concatenated and input to the classifier (a 2-layer MLP). BS-2 is trained using the loss ℓ_{part} (Eq. 6) and the cross-entropy loss, using the same alternating optimization strategy as our model.

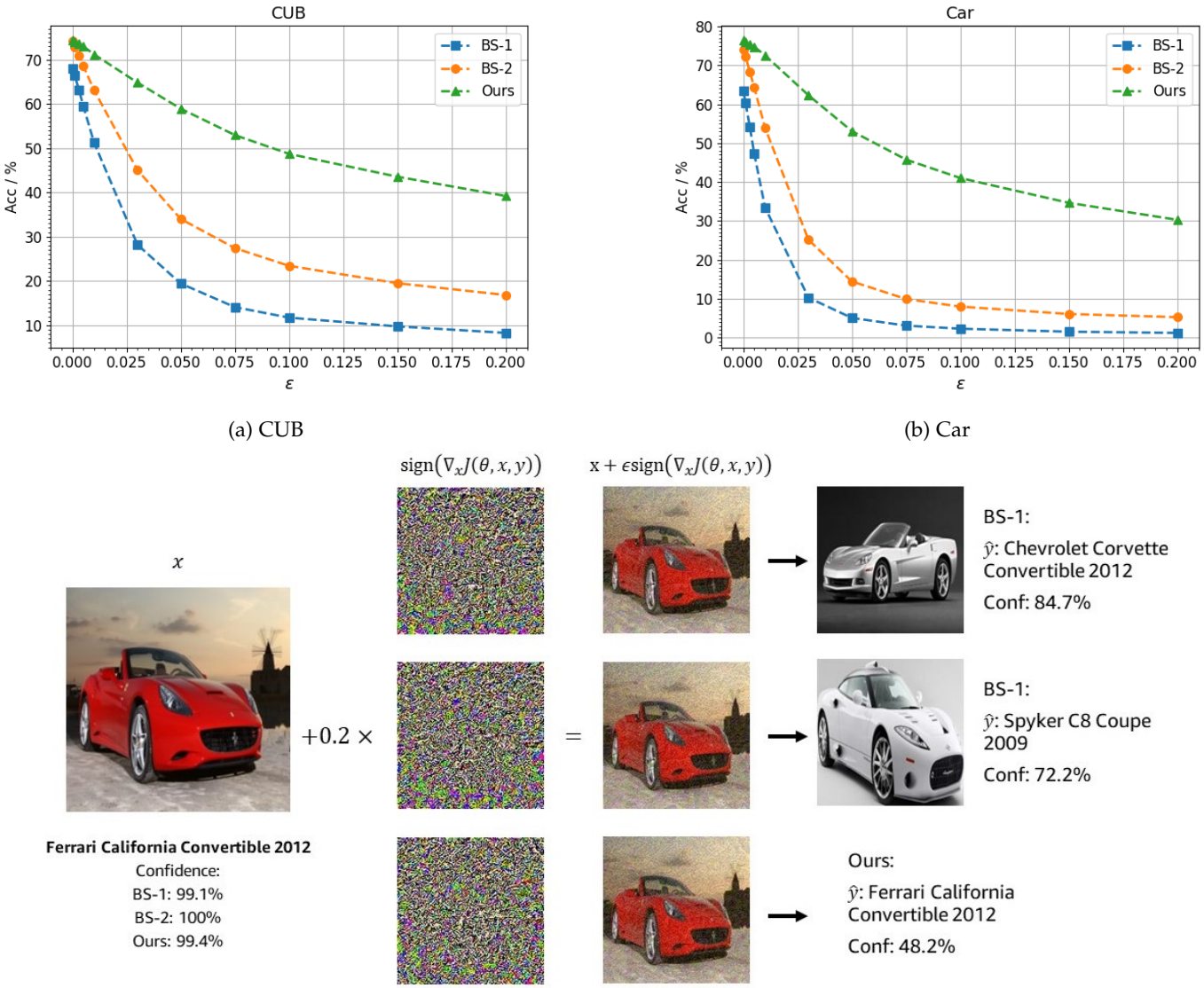
Training Details. BS-1 is trained using a learning rate 5e-5 for 100 epochs on CUB and 110 epochs on Car. The learning rate is decayed by a factor of 0.5 at epochs [60, 80] for CUB and epochs [70, 90] for Car. BS-2 is trained with learning rate 2e-4 on CUB and 4e-4 on Car. It is trained for 60 epochs and the learning rate is decayed by 0.5 every 20 steps. Our RPC model initialized its parameters of the part feature extractor from the trained BS-2 weights. It then is only trained for 5 more epochs with a learning rate 2e-5.

Results. We sweep ϵ (in Eq 16) from 0 to 0.2 and evaluated the test accuracy for the three models. The results are plotted in Fig. 5(a-b). We see that performs decays at a much lower rate with attacks of increasing intensity for our RPC model as compared to the baselines. Specifically, when $\epsilon = 0.2$, on CUB, BS-1 has less than 10% accuracy and BS-2 is less than 20%, while our model can still achieve 40%. On the Car dataset, BS-2 only obtain 5% and BS-1 even drops to 1%, but our RPC model can maintain an impressive 30% accuracy. In Fig. 5c we illustrate an example that recognized by our model is misclassified by the two baseline models with the same distortion intensity. Notice that BS-2 is still very vulnerable to the adversarial attack, even though it has the same multi-attention mechanism in RPC. Because RPC learns to represents each image part in the vocabulary of a small number of prototypes, it is less sensitive to the perturbations in the input image compared to other high-dimensional visual features.

6.5 Ablating Training Objectives

For training our RPC encoder, we used two penalties L_{com} and L_{div} that correspond to the compactness and diversity priors on part locations as mentioned in section 3. Here, we demonstrate the effect that each of these losses has on the model performance and the encodings themselves.

Additionally, the features $z_m(x)$ are encoded into part-type likelihood scores of a gaussian mixture model constituting the RPC code $\pi(x)$ (see Eq. 7). The autoencoder loss from Eq. 9 was used for training the parameters in this mapping. We also demonstrate the effect of removing L_{ae} and using the part features $z_m(x)$ (instead of $\pi_m(x)$) for the classifier.



(a) CUB

(b) Car

(c) Example of perturbed images from Car dataset. The original ("Ferrari California Convertible 2012") were recognized by all three models with >99% confidence. The distortion η (Eq. 16) to the input is imperceivable to human eys. The two baseline models recognize the perturbed input as incorrect classes "Chevrolet Corvette Convertible 2012" and "Spyker C8 Coupe 2009", while our model predict the correct class label. RPC achieves this by virtue of encoding images in a manner similar to how humans would.

Fig. 5: Robustness to adversarial attacks. (a-b): Test accuracy under various level of FGSA perturbations on CUB (a) and Car (b). (c): Example perturbed images and predictions for baselines and our models. ϵ: the perturbation intensity.

In Table 6 we can see that the three major components of RPC L_{ae}, L_{com} and L_{div} play a positive role when it comes to performance in low shot generalization tasks. In Fig. 6 we see that the two losses on part attentions have intended effects and that when we remove L_{com}, the part attention maps are more dispersed, than concentrated at the part locations. When we remove L_{div}, the part locations detected by the model can end up on top of each other since each part would seek just the most salient feature disregarding where other parts lie.

Robustness. In passing we also observe from Sec. 6.4 that our scheme with the auto-encoder, diversity and compactness also plays a significant role in improving robustness relative to the baseline schemes that do not have the auto-encoder in particular. The auto-encoder, together with compactness and diversity ensure, that the RPC encoder allows for discretization of concepts, and thus for an attack to be successful, the adversary must modify the input image significantly to modify a concept.

Sensitivity to softmax temperature. In Table 7, we report the 5-way 5-shot accuracy of a classifier using the RPC encodings with different softmax temperatures used for ϕ (Eq. 7 of the revised paper). Smaller temperatures make the distribution output by the softmax function more peaked. Recall that after validation, we set the temperature for all experiments to 100. We find that the accuracy is quite stable till temperature=10. However, at low temperatures like 1 and below, where the model is forced to recognize object

L_{com}	L_{div}	L_{ae}	FSL CUB *1-shot*	FSL CUB *5-shot*	DA S→M	GZSL CUB Harmonic mean
☑	☐	☑	74.72	88.83	94.2	43.7
☐	☑	☑	73.01	87.74	94.5	39.6
☑	☑	☐	71.84	85.63	94.4	48.1
☑	☑	☑	75.01	89.61	95.2	48.4

TABLE 6: Performance metrics on different benchmarks (5-way few shot classification on CUB, domain adaptive classification on transfer from SVHN → MNIST, and the harmonic mean accuracy on seen and unseen classes in the case of GZSL on the CUB dataset) when different losses in RPC training are ablated.

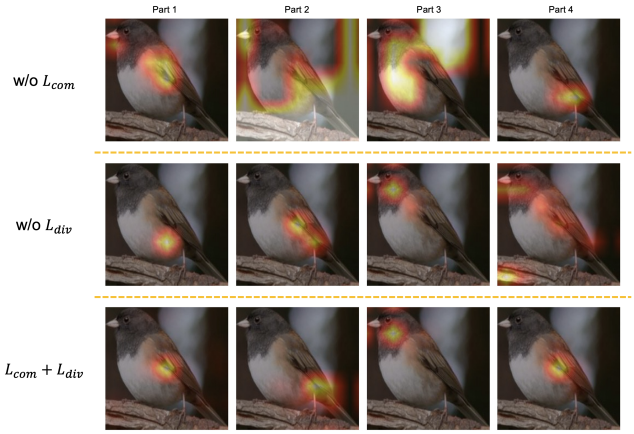


Fig. 6: Attention maps of models learned with different objectives from CUB. Row 1: L_{com} is not used; Row 2: L_{div} is not used; Row 3: both L_{com} and L_{div} are used.

Softmax Temp.	*mini*Imagenet	CUB
1000	82.41	88.62
100	84.57	89.61
10	84.31	88.81
1	27.47	57.38

TABLE 7: 5-way 5-shot accuracy in % on CUB and *mini*Imagenet with different temperatures of softmax.

parts as being predominantly of one part-type right from the start of training, we see that the final classification performance degrades.

7 Human Evaluation

In this section, we present results of crowd-sourced experiments conducted using Amazon Mechanical Turk (MTurk), that indicate that our RPC encodings are interpretable and agreeable with human perception. We designed three questions to gauge this agreement along different aspects : 1) *Discriminability of parts*: Are the parts discriminative enough for humans to recognize the class? 2) *Prototype recognition*: Can humans recognize the prototypes for parts of a certain image? 3) *Part-type likelihood prediction*: Do humans agree with the part-type likelihood scores output by the model?

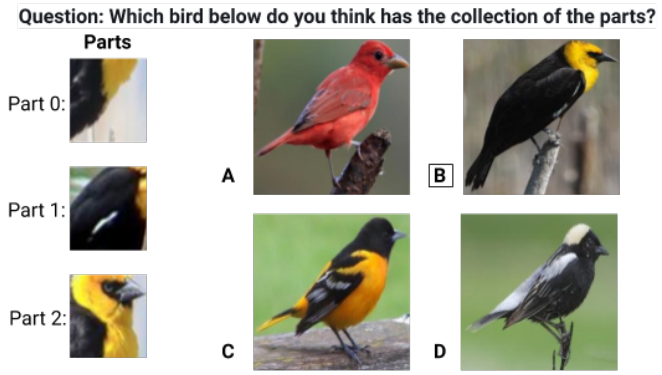


Fig. 7: **Q1.** Exemplar question for discriminability. The workers are asked to select the class they think has the parts provided on the left side. If none of the given classes applies, they should select "None of above". The ground truth class is B in this example which is bounded by a box (Best viewed in color).

No.	**Class**
0	Yellow_headed_Blackbird*
1	Bobolink
2	Indigo_Bunting
3	Painted_Bunting*
4	Vermilion_Flycatcher
5	American_Goldfinch
6	Baltimore_Oriole
7	Tree_Swallow
8	Summer_Tanager
9	Prothonotary_Warbler

TABLE 8: Selected classes of CUB for crowd-sourcing. *: the selected unseen classes.

For the purpose of this section, we select a subset containing 10 classes of the CUB dataset. The classes are listed in Table 8. Each class has around 60 images. To simulate the FSL and ZSL scenarios where the novel classes have no samples during training, we held out two as unseen classes, and train our model on the remaining 8. The model has 3 parts and the number of prototypes in each part is set to 5. After the model is trained, the 3 parts located by our model are cropped from the original images for all classes centered around the peak values of the attention maps. In each part, we select one example that represents each prototype through nearest neighbor search in the RPC features of the seen class images.

We use these prototypes and examples to create the assignments in MTurk. Each assignment is answered by 5 different turkers. The questions and results are detailed below:

Discriminability of parts. This question provides turkers with a collection of 3 parts identified by our model and asks them to select the class which those parts belong to. An exemplar question is shown in Fig. 7. The turker can choose one class out of four options or "None of the above". The intent of this question is to ascertain whether humans find the parts discriminative enough to distinguish between different categories, and demonstrates that our model learns to recognize salient parts of an image. For this question we used 502 examples in total from all the 10 classes. We use the

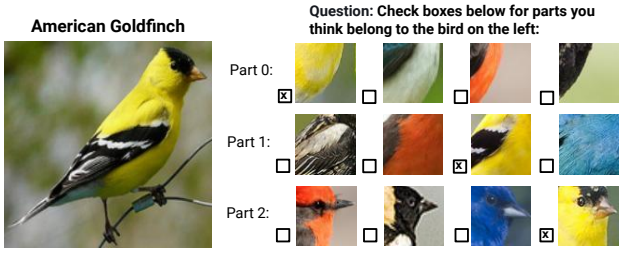


Fig. 8: **Q2.** Exemplar question for prototype recognition. Turkers are asked to pick the prototypes they think belong to the given image on the left. The ground truth is marked with an 'x' in its box (best viewed in color).

Fig. 9: **Q3.** Exemplar question for part-type likelihood prediction. Turkers are asked to choose a score for the similarity between the given part and the prototype. 1 means least and 5 means most similar (best viewed in color).

majority response of the 5 workers as the final answer for each image and consider the image to be "correct" if the final answer is the same as the ground truth class. The accuracy we obtained for all examples is 94.2%, which indicates a high probability that humans can distinguish the bird using only the parts discovered by our model. Note that our model was only trained on the 8 seen classes, but it still produced discriminative parts for the novel classes, demonstrating that the parts learned are generalizable.

Prototype recognition. This question provides workers with an image, which we shall refer to as the query image, from one of the seen classes and a set of prototypical examples for each part. The workers are asked to choose the prototype in each part that they think belongs to the bird in the given image, as shown in Fig. 8. The question is intended to determine if humans can learn to recognize images using the prototypes learned by our model. Good accuracy would demonstrate that the compact vocabulary of prototypes/concepts that our model learns, are meaningful to humans, and help them recognize whether a certain instance comes from a given learned concept. We used 406 examples from seen classes for this question and in each part we also use the majority vote of the 5 answers for computing accuracy. For each prototype, we find the closest set of 5 examples as a representative set and the set of classes that these 5 images belong to is called the class-set corresponding to that prototype. Each question picks 4 prototypes randomly, and 1 out of the 5 representative examples is displayed on the question to represent that prototype. A response is considered "correct" if the class of the query image lies in the class-set of the prototype selected. The accuracy for the three parts are 92.96%, 95.73%, and 96.98%, respectively, validating that humans can learn to recognize images in terms of concepts represented by the prototypes learned by our model. Note that a random guess would result in a 25% accuracy in expectation. Also, note that the intention of this question is to evaluate the quality of our prototypes/concepts that form the vocabulary for encoding part instances in. It does not evaluate the agreement between RPC model's prediction of whether an image corresponds to a certain concept and a human's prediction. This is evaluated in the next question.

Part-type likelihood prediction. In this question, turkers are provided with a part example and a list of all prototypes for the same part. Turkers are asked to rate the probability on a scale of 1 to 5, 1 being least possible, that the given example belongs to a particular prototype, as shown in Fig. 9. We used 502 images from all classes per part for this experiment. For each of the 3 parts, in order to obtain the class probability score, we average the scores provided by turkers over all images belonging to a certain category. As a comparison, we also average the RPC encoding from our model for each category. The class probability scores of the workers, our model, and their absolute differences, are visualized in Fig. 10a, Fig. 10b and Fig. 10c for the three parts, respectively. For better visualization, the probability scores are linearly scaled to lie in the range $[0, 1]$. The results confirm that human annotators largely agree with the probability that our model assigns for a part in an image belonging to a given part-type represented by the prototypes. This holds true for scores from classes 0 and 3 as well, which are not seen during training. Thus the model's perception of visual similarity agrees with those of humans and this perception generalizes to unseen images.

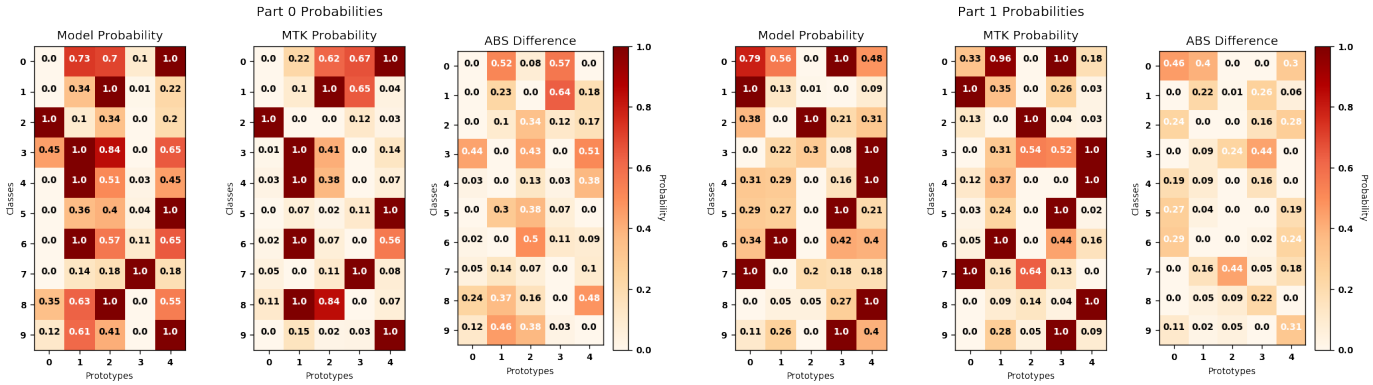
8 Synthetic Attribute Generation

Learning at large-scale poses two fundamental challenges. First, acquiring ground-truth annotations for training instances is expensive. Second, as we scale the number of object classes, we observe few instances for many (rare) object classes. ZSL proposes to overcome these challenges by leveraging semantic descriptions or attributes for the different classes.

In practice, advancements in ZSL methods is fundamentally limited by the unavailability of "good" zero-shot datasets. Semantic attributes that are visually well-aligned, such as in the CUB dataset, can be obtained through crowdsourcing but doing so is often very expensive. Popular but less effective are datasets leveraging language corpora to derive encodings, such as word-embeddings, for different object class categories. Such word-embeddings although semantically meaningful, are visually misaligned, and may not represent a visually meaningful description of the classes.

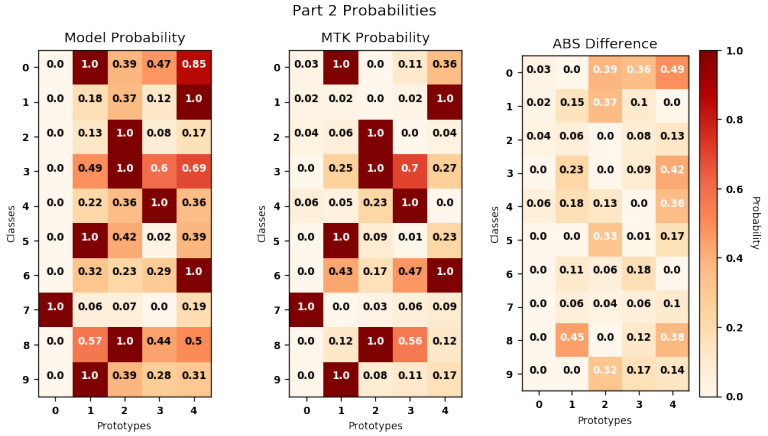
Since our framework outputs image encodings that are interpretable by humans, we propose to leverage them as a proxy for human annotations to provide synthetic semantic attributes. Zero-shot learning methods can be evaluated using these attributes when human annotated attributes are unavailable.

To generate synthetic semantic attributes, we train our RPC model for classification on all "seen" classes of the dataset. In this case, for our RPC model, we use $M = 3$



(a) Part 0

(b) Part 1



(c) Part 2

Fig. 10: The class probability scores from our model (left) and the Mturk workers (middle), and their absolute difference (right). The RPC model's predictions of the likelihood that part instances come from specific concepts/prototypes agree with the predictions of humans

Methods	CUB			CUB-syn			Cars-syn		
	U	S	H	U	S	H	U	S	H
CADA-VAE [111]	**51.6**	53.5	**52.4**	**64.0**	54.7	**59.0**	**65.3**	**77.7**	**70.9**
fCLSWGAN [113]	**43.7**	57.7	**49.7**	**56.0**	**59.5**	**57.7**	**46.1**	**57.2**	**51.1**
RN [44]	38.1	**61.1**	47.0	54.2	**60.5**	57.2	30.6	49.0	37.7
GDAN [110]	39.3	**66.7**	49.5	20.5	33.2	25.4	8.1	38.4	13.4

TABLE 9: GZSL results on CUB, CUB-syn and Cars-syn. CUB uses original class semantic vectors and CUB-syn and Car-syn use synthetically generate class semantic vectors using the RPC model. U = unseen classes, S = seen classes, H = harmonic mean. The accuracy is class-average Top-1 in %. In each column, **red**=highest and **blue**=2^{nd} highest accuracy

parts and $K = 64$ prototypes per part. Once trained, we generate RPC encodings of images from both seen and unseen classes, and use the average RPC encoding for each class, as its semantic vector. Note that this is similar to how the semantic class attributes were generated for the CUB dataset, the only difference is that attributes annotations per image were collected from humans for CUB. Note that we increased the number of prototypes used, compared to our previous experiments, to have a semantic vector that is similar to the size of the semantic vector for CUB (Our semantic vector size $= 64 \times 3 = 192$ compared to the original 312-dimensional semantic vectors in CUB).

Using the approach above, we generated semantic vectors for the CUB dataset [74], which already has human annotated semantic vectors and the Cars dataset [123], which is not a ZSL dataset and does not have any semantic vectors associated with class labels. We hence propose a novel GZSL split for the Cars dataset. Classes are split into 131 seen and 65 unseen. For each seen class, we randomly sample $3/4^{th}$ of its images as training data, and use the rest for testing (test-seen). All images of unseen classes belong to test-unseen. The proposed split has 8,100 training images, 2,637 test-seen images and 5,448 test-unseen images. This split, along with the synthetically generated semantic vectors, will be made publicly accessible.

We report the result of 4 zero-shot learning methods (citations in table), which we evaluated using implementations from the authors' publicly available code. Results of the evaluation are reported in Table 9. The columns titled CUB-syn and Cars-syn are the respective datasets with the synthetically generated semantic vectors. For this evaluation, we used the original hyperparameters of the specific GZSL

methods for their evaluation on the CUB dataset. Note that we used the same hyperparameters for both CUB-syn and Cars-syn.

First, comparing the harmonic mean accuracies (H) for CUB and CUB-Syn in Table 9, we see that all methods perform similarly as they did on the original semantic attributes on CUB, with CADA-VAE performing the best. All methods improve in absolute accuracy, possibly since our attributes better mirror visual information than the semantic attributes that are collected via a noisy crowd-sourcing procedure. This improvement is with the exception of GDAN, which decreases in performance, possibly due to its sensitivity to hyperparameters, corresponding to the specific semantic vectors. On the Cars-syn GZSL dataset, using our synthetically generated class semantic vectors, we see that CADA-VAE and fCLSWGAN remain the two best performing methods, but it seems RN and GDAN do not do well because of their sensitivity to specific hyperparameters used.

9 Conclusion

We proposed Recognition as Part Composition, an approach for image recognition inspired by human cognition. Our approach first decomposes an image into salient parts, and then learns to represent each part instance as a mixture of a few concepts. We found that this approach, imparted big benefits to classifiers in low-shot generalization tasks like zero-shot learning, few-shot learning and unsupervised domain adaptation. We also found that using these encodings also makes a classifier more robust to adversarial attacks, which impercetibly change an input image to induce a classifier error. Via crowd-sourcing, we also demonstrated that the encodings agree with human perception and that humans can recognize images using the parts learnt by our model and recognize the part instances in the vocabulary that RPC learns. Given the fact our encodings are human-interpretable, we proposed an application of them, to generate synthetic attributes for evaluating zero-shot learning methods on new datasets, before collecting human annotated class semantics for them. We demonstrated this on the Stanford Cars dataset as a proof of concept.

Acknowledgements

This research was supported by the Army Research Office Grant W911NF2110246, the National Science Foundation grants CCF-2007350 and CCF-1955981, and the Hariri Data Science Faculty Fellowship Grants. The authors would like to thank Ruizhao Zhu for helpful discussions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[3] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] B. Benelli, "Em markman, categorization and naming in children. problems of induction. cambridge, ma: The mit press, 1989. pp. i+ 250." *Journal of Child Language*, vol. 18, no. 3, pp. 717–720, 1991.

[6] J. Feldman, "The structure of perceptual categories," *Journal of mathematical psychology*, vol. 41, no. 2, pp. 145–170, 1997.

[7] F. Xu and J. B. Tenenbaum, "Word learning as bayesian inference." *Psychological review*, vol. 114, no. 2, p. 245, 2007.

[8] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987.

[9] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[10] D. Aerts, J. Broekaert, L. Gabora, and S. Sozzo, "Generalizing prototype theory: A formal quantum framework," *Frontiers in Psychology*, vol. 7, p. 418, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2016.00418

[11] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Int. Conf. on Computer Vision*, vol. 6, 2017.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[13] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *British Machine Vision Conference (BMVC)*, 2018. [Online]. Available: http://bmvc2018.org/contents/papers/1064.pdf

[14] S. A. Bargal, A. Zunino, D. Kim, J. Zhang, V. Murino, and S. Sclaroff, "Excitation backprop for rnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1440–1449.

[15] P. Zhu, H. Wang, and V. Saligrama, "Learning classifiers for target domain with limited or no labels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7643–7653.

[16] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[17] G. Vilone and L. Longo, "Explainable artificial intelligence: a systematic review," *arXiv preprint arXiv:2006.00093*, 2020.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[19] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6541–6549.

[20] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba, "Understanding the role of individual units in a deep neural network," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 071–30 078, 2020.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[23] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.

[24] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[25] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[26] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[27] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of*

the 10th ACM workshop on artificial intelligence and security, 2017, pp. 15–26.

[28] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[29] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[30] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*, 2018.

[31] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[32] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[33] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 4.

[34] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 1647–1657.

[35] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Adversarial dropout regularization," *arXiv preprint arXiv:1711.01575*, 2017.

[36] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3723–3732.

[37] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," *arXiv preprint arXiv:1702.08400*, 2017.

[38] A. Chadha and Y. Andreopoulos, "Improving adversarial discriminative domain adaptation," *arXiv preprint arXiv:1809.03625*, 2018.

[39] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.

[40] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

[41] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2242–2251.

[42] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 7.

[43] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[44] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[45] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[46] T. Munkhdalai and H. Yu, "Meta networks," *arXiv preprint arXiv:1703.00837*, 2017.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[48] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[49] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," *arXiv preprint arXiv:1801.05401*, vol. 8, 2018.

[50] A. Mehrotra and A. Dukkipati, "Generative adversarial residual pairwise networks for one shot learning," *arXiv preprint arXiv:1703.08033*, 2017.

[51] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov *et al.*, "Devise: A deep visual-semantic embedding model," in *Advances in neural information processing systems*, 2013, pp. 2121–2129.

[52] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[53] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[54] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai, "Generalized zero-shot learning via synthesized examples," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[56] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[57] H. Jiang, R. Wang, S. Shan, and X. Chen, "Learning class prototypes via structure alignment for zero-shot recognition," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[58] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature neuroscience*, vol. 5, no. 7, pp. 682–687, 2002.

[59] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[60] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *2010 IEEE Computer society conference on computer vision and pattern recognition*. Ieee, 2010, pp. 2241–2248.

[61] P.-A. Savalle, S. Tsogkas, G. Papandreou, and I. Kokkinos, "Deformable part models with cnn features," in *European Conference on Computer Vision, Parts and Attributes Workshop*, 2014.

[62] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 437–446.

[63] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," *arXiv preprint arXiv:2006.15055*, 2020.

[64] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner, "Multi-object representation learning with iterative variational inference," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2424–2433.

[65] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner, "Monet: Unsupervised scene decomposition and representation," *arXiv preprint arXiv:1901.11390*, 2019.

[66] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[67] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2, 2011, p. 5.

[68] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[69] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[70] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[71] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2018, pp. 5419–5428.

[72] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," *arXiv preprint arXiv:1706.05208*, 2017.

[73] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6028–6039.

[74] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[75] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 2014.

[76] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[77] P. Bateni, R. Goyal, V. Masrani, F. Wood, and L. Sigal, "Improved few-shot visual classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 493–14 502.

[78] A. Li, T. Luo, T. Xiang, W. Huang, and L. Wang, "Few-shot learning with global class representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9715–9724.

[79] S. W. Yoon, J. Seo, and J. Moon, "Tapnet: Neural network augmented with task-adaptive projection for few-shot learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7115–7123.

[80] J. Liu, F. Chao, L. Yang, C.-M. Lin, C. Shang, and Q. Shen, "Decoder choice network for metalearning," *IEEE Transactions on Cybernetics*, 2021.

[81] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

[82] Y. Chen, X. Wang, Z. Liu, H. Xu, and T. Darrell, "A new meta-baseline for few-shot learning," *arXiv preprint arXiv:2003.04390*, 2020.

[83] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8808–8817.

[84] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 203–12 213.

[85] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "Dpgn: Distribution propagation graph network for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 390–13 399.

[86] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8822–8833.

[87] N. Fei, Y. Gao, Z. Lu, and T. Xiang, "Z-score normalization, hubness, and few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 142–151.

[88] S. Baik, J. Choi, H. Kim, D. Cho, J. Min, and K. M. Lee, "Meta-learning with task-adaptive loss function for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9465–9474.

[89] Z. Zhou, X. Qiu, J. Xie, J. Wu, and C. Zhang, "Binocular mutual learning for improving few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8402–8411.

[90] M. N. Rizve, S. Khan, F. S. Khan, and M. Shah, "Exploring complementary strengths of invariant and equivariant representations for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 836–10 846.

[91] C. Chen, X. Yang, C. Xu, X. Huang, and Z. Ma, "Eckpn: Explicit class knowledge propagation network for transductive few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6596–6605.

[92] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten, "Simpleshot: Revisiting nearest-neighbor classification for few-shot learning," *arXiv preprint arXiv:1911.04623*, 2019.

[93] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7260–7268.

[94] K. Cao, M. Brbic, and J. Leskovec, "Concept learners for few-shot learning," *arXiv preprint arXiv:2007.07375*, 2020.

[95] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 470–13 479.

[96] X. Li, J. Wu, Z. Sun, Z. Ma, J. Cao, and J.-H. Xue, "Bsnet: Bi-similarity network for few-shot fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1318–1331, 2020.

[97] Q. S. Y. Liu, T. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[98] J. Xu, H. Le, M. Huang, S. Athar, and D. Samaras, "Variational feature disentangling for fine-grained few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8812–8821.

[99] C. Wang, S. Song, Q. Yang, X. Li, and G. Huang, "Fine-grained few shot learning with foreground object transformation," *Neurocomputing*, vol. 466, pp. 16–26, 2021.

[100] D. Wertheimer, L. Tang, and B. Hariharan, "Few-shot classification with feature map reconstruction networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8012–8021.

[101] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2927–2936.

[102] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," *arXiv preprint arXiv:1704.08345*, 2017.

[103] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4166–4174.

[104] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1425–1438, 2016.

[105] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5327–5336.

[106] Y. Annadani and S. Biswas, "Preserving semantic relations for zero-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[107] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang, "Zero-shot visual recognition using semantics-preserving adversarial embedding network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2018.

[108] Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2371–2381.

[109] Y. Liu, L. Zhou, X. Bai, Y. Huang, L. Gu, J. Zhou, and T. Harada, "Goal-oriented gaze estimation for zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3794–3803.

[110] H. Huang, C. Wang, P. S. Yu, and C.-D. Wang, "Generative dual adversarial network for generalized zero-shot learning," *arXiv preprint arXiv:1811.04857*, 2018.

[111] E. Schönfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," *arXiv preprint arXiv:1812.01784*, 2018.

[112] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7402–7411.

[113] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5542–5551.

[114] Y. Atzmon and G. Chechik, "Domain-aware generalized zero-shot learning," *arXiv preprint arXiv:1812.09903*, 2018.

[115] A. Zhao, M. Ding, J. Guan, Z. Lu, T. Xiang, and J.-R. Wen, "Domain-invariant projection learning for zero-shot recognition," in *Advances in Neural Information Processing Systems 31*,

S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 1019–1030.

[116] L. Zhang, P. Wang, L. Liu, C. Shen, W. Wei, Y. Zhang, and A. V. D. Hengel, "Towards effective deep embedding for zero-shot learning," 2018.

[117] L. Bo, Q. Dong, and Z. Hu, "Hardness sampling for self-training based transductive zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 499–16 508.

[118] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

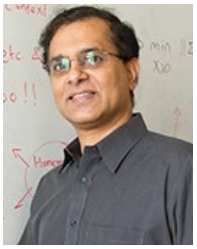
[119] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1778–1785.

[120] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3485–3492.

[121] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *ECCV*, 2016.

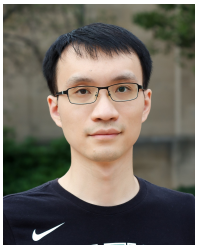
[122] R. Felix, M. Sasdelli, I. Reid, and G. Carneiro, "Multi-modal ensemble classification for generalized zero shot learning," *arXiv preprint arXiv:1901.04623*, 2019.

[123] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.



Venkatesh Saligrama is a faculty member in the Department of Electrical and Computer Engineering, the Department of Computer Science (by courtesy), and a founding member of the Faculty of Computing and Data Sciences at Boston University. His research interests are broadly in the area of Artificial Intelligence, and his recent work has focused on machine learning with resource-constraints. He is an IEEE Fellow and recipient of several awards including Distinguished Lecturer for IEEE Signal Processing Society, the US Presidential Early Career Award (PECASE), ONR Young Investigator Award, and the NSF Career Award. More information about his work is available at http://sites.bu.edu/data

Samarth Mishra is a Ph.D. student in the Image and Video Computing group at Boston University, co-supervised by Profs. Venkatesh Saligrama and Kate Saenko. His research interests lie in computer vision and machine learning, and specifically in problems dealing with a scarcity of labeled data. More information about his work can be found at https://samarth4149.github.io



Pengkai Zhu is an applied scientist in AWS AI. He received his Ph.D from Boston University, where he was supervised by Prof. Venkatesh Saligrama. His research interests include computer vision and machine learning, specially low-shot recognition and visual document understanding.