ON THE CAPABILITY OF CNNS TO GENERALIZE TO UNSEEN CATEGORY-VIEWPOINT COMBINATIONS

Anonymous authors
Paper under double-blind review

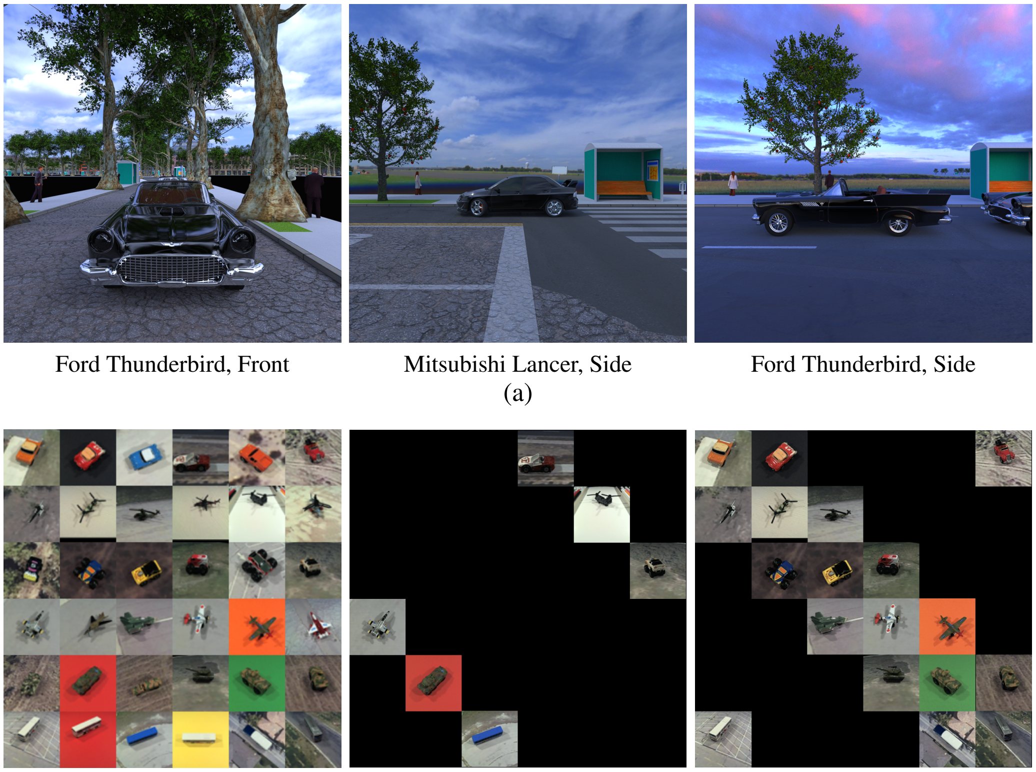
ABSTRACT

Object recognition and viewpoint estimation lie at the heart of visual understanding. Recent works suggest that convolutional neural networks (CNNs) fail to generalize to category-viewpoint combinations not seen during training. However, it is unclear when and how such generalization may be possible. Does the number of combinations seen during training impact generalization? What architectures better enable generalization in the multi-task setting of simultaneous category and viewpoint classification? Furthermore, what are the underlying mechanisms that drive the network's generalization? In this paper, we answer these questions by analyzing state-of-the-art CNNs trained to classify both object category and 3D viewpoint, with quantitative control over the number of category-viewpoint combinations seen during training. We also investigate the emergence of two types of specialized neurons that can explain generalization to unseen combinations—neurons selective to category and invariant to viewpoint, and vice versa. We perform experiments on MNIST extended with position or scale, the iLab dataset with vehicles at different viewpoints, and a challenging new dataset for car model recognition and viewpoint estimation that we introduce in this paper - the *Biased-Cars* dataset. Our results demonstrate that as the number of combinations seen during training increase, networks generalize better to unseen category-viewpoint combinations, facilitated by an increase in the selectivity and invariance of individual neurons. We find that learning category and viewpoint in separate networks compared to a shared one leads to an increase in selectivity and invariance, as separate networks are not forced to preserve information about both category and viewpoint. This enables separate networks to significantly outperform shared ones at classifying unseen category-viewpoint combinations.

1 INTRODUCTION

The combination of object recognition and viewpoint estimation is essential for effective visual understanding. In recent years, convolutional neural networks (CNNs) have offered state-of-the-art solutions for both these fundamental tasks (He et al., 2016; Szegedy et al., 2016; Huang et al., 2017; Su et al., 2015; Massa et al., 2016; Elhoseiny et al., 2016; Mahendran et al., 2018; Afifi et al., 2018). However, recent works suggest that CNNs have a hard time generalizing to combinations of object categories and viewpoints not seen during training. For object recognition, works have shown CNNs struggling to generalize across spatial transformations like 2D rotation and translation (Engstrom et al., 2019; Azulay & Weiss, 2019; Srivastava et al., 2019), and non-canonical 3D views (Alcorn et al., 2019; Barbu et al., 2019). For viewpoint estimation, previous works suggest learning category specific models (Massa et al., 2016; Tulsiani & Malik, 2015) or feed class predictions as input to the model (Xiang et al., 2017; Manhardt et al., 2020), as generalizing to novel categories is a challenging task.

It remains unclear *when* and *how* CNNs may generalize across category-viewpoint combinations not seen during training, *ie.,* extrapolate. Fig. 1a presents a motivating example: would a network trained with examples of a Ford Thunderbird seen only from the front, and a Mitsubishi Lancer seen only from the side generalize to predict car model (category) and viewpoint for a Thunderbird shown from the side? If so, what underlying mechanisms enable a network to do so?



Ford Thunderbird, Front

Mitsubishi Lancer, Side

Ford Thunderbird, Side

(a)

iLab-2M dataset

(b)

Unseen Combinations (held-out)

(c)

50% *Seen* Combinations

(d)

Figure 1: *Category-Viewpoint datasets.* (a) Our new *Biased-Cars* dataset: Can a network shown only the Ford Thunderbird from front and the Mitsubishi Lancer from side generalize to classify the category and viewpoint for a Thunderbird seen from the side? (b) iLab-2M dataset (Borji et al., 2016): Each cell represents a unique category-viewpoint combination (categories vary between rows, viewpoint between columns) with multiple object instances per category and backgrounds. (c) Held-out test set of category-viewpoint combinations. Same held-out test set is used to evaluate networks trained with different percentage of *seen* combinations. (d) Biased training set with 50% of category-viewpoint combinations. Number of categories and viewpoints selected is always equal.

In order to understand the capability of CNNs to generalize to unseen category-viewpoint combinations, we investigate the impact of two key factors—data diversity and architectural choices. We train state-of-the-art CNNs on an increasing number of object category-viewpoint combinations and evaluate them on a held out test set of combinations never seen during training. Further, we analyze what architectures better enable generalization in the multi-task setting of simultaneous category and viewpoint classification. Finally, to understand the underlying mechanisms driving the generalization of the network, we investigate the emergence of selective and invariant representations which have been shown to facilitate generalization (Giles & Maxwell, 1987; Riesenhuber & Poggio, 1998; Goodfellow et al., 2009; Achille & Soatto, 2018).

In a series of experiments, we demonstrate that the percentage of category-viewpoint combinations seen during training (training data diversity) matters significantly. As CNNs are trained with more combinations, there is an increase in their accuracy on unseen combinations. Note that in the usual supervised learning setting train and test samples are selected randomly, while here we enforce a difference in test and train distributions, thereby testing extrapolation. Our results also reveal that learning category and viewpoint in separate networks helps CNNs generalize substantially better compared to learning them together. We show that this generalization behaviour is driven by an increase in the selectivity and invariance of the neurons, which is even more pronounced when category and viewpoint classification are learned in separate networks compared to a shared one.

These results are consistent across multiple state-of-the-art CNNs in MNIST (LeCun et al., 1998) extended with position and scale, the iLab-2M dataset (Borji et al., 2016) and in a new challenging dataset of car model recognition and viewpoint estimation - the *Biased-Cars* dataset, which we introduce in this paper. Our dataset consists of 15K photorealistic rendered images of several car models at different positions, scales and viewpoints, and under various illumination, background, clutter and occlusion conditions. With this, we hope to provide a first milestone at understanding the underlying mechanisms which enable CNNs to generalize to unseen category-viewpoint combinations.

2 Datasets for simultaneous category-viewpoint classification

Most existing datasets with category and viewpoint labels (Xiang et al., 2014; Caesar et al., 2019) present two major challenges - *(i)* lack of control over the distribution of categories and viewpoints, or *(ii)* small size. Thus, we chose the following datasets:

MNIST-Position and MNIST-Scale. These are variants of the MNIST dataset (LeCun et al., 1998), created by adding viewpoint in the form of position or scale to MNIST. MNIST-Position was created by placing MNIST images into one of nine possible locations in an empty 3-by-3 grid. For MNIST-Scale we resized images to one of nine possible sizes followed by zero-padding. Images of the digit 9 were left out in both these datasets ensuring nine category and nine viewpoint classes (total of 81 category-viewpoint combinations). Sample images are available in the supplement.

iLab-2M dataset. iLab-2M (Borji et al., 2016) is a large scale (two million images), natural image dataset with 3D variations in viewpoint and multiple object instances for each category (Fig.1b). The dataset was created by placing toy objects on a turntable and photographing them from six different azimuth viewpoints, each at five different zenith angles (total 30). From the original dataset, we chose a subset of six object categories - Bus, Car, Helicopter, Monster Truck, Plane, and Tank. In Fig. 1b, each row represents images from one category, and each column images from one azimuth angle. All networks are trained to predict one of six category and viewpoint (azimuth) labels each.

Biased-Cars dataset. Building on other multi-view car datasets for viewpoint estimation (Krause et al., 2013; Ozuysal et al., 2009), we introduce a challenging new dataset for simultaneous object category and viewpoint classification - the *Biased-Cars* dataset. Our dataset features photo-realistic outdoor scene data with fine control over scene clutter (trees, street furniture, and pedestrians), car colors, object occlusions, diverse backgrounds (building/road textures) and lighting conditions (sky maps). *Biased-Cars* consists of 15K images of five different car models seen from viewpoints varying between 0-90 degrees of azimuth, and 0-50 degrees of zenith across multiple scales. Our dataset offers two main advantages: *(a)* complete control over the joint distribution of categories, viewpoints, and other scene parameters, and *(b)* unlike most existing synthetic city datasets (Qiu & Yuille, 2016; Caesar et al., 2019; Dosovitskiy et al., 2017) we use physically based rendering for greater photo-realism, which has been shown to help networks transfer to natural image data significantly better (Zhang et al., 2017; Halder et al., 2019). Sample images are shown in Fig. 1a. As in (Xiang et al., 2014; Divon & Tal, 2018), we choose to focus on azimuth prediction. The azimuth is divided into five bins of 18 degrees each, thus ensuring five category (car models) and five viewpoint classes (azimuth bins), for a total of 25 different category-viewpoint combinations.

For all datasets networks are trained to classify both category and viewpoint simultaneously without pretraining, and number of classes for each task is kept equal to ensure equal treatment. More details can be found in the supplement. As shown in Section 5, these datasets are challenging benchmarks for testing generalization, with a huge scope for improvement for state-of-the-art CNNs.

3 Factors affecting generalization behaviour

Below we present the two factors we study for their impact on generalization to *unseen* category-viewpoint combinations - *(i)* data diversity, and *(ii)* architectural choices.

3.1 Generating train/test splits with desired data diversity

Every dataset described in Section 2 can be visualized as a square category-viewpoint *combinations grid* as shown for the iLab dataset in Fig. 1b. Each row represents images from one category, and each column a viewpoint, *ie.,* each cell represents images from one category-viewpoint combination.

Constructing the test split. For every dataset, we select and hold out a set of combinations from the *combinations grid* as shown in Fig. 1c. We refer to these as the *unseen* combinations. Images from these combinations are never shown to any network during training. Instead, these images are used

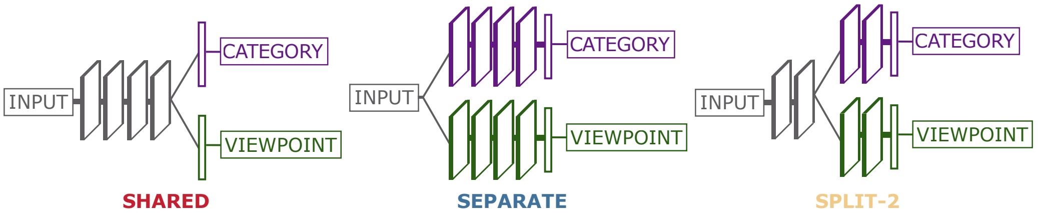


Figure 2: *Architectures for Category Recognition and Viewpoint Estimation. Shared*, *Separate* and *Split-2* architectures for ResNet-18. In the *Shared* architecture, all layers until the last convolutional block are shared between tasks, followed by task specific fully connected branches. In the *Separate* architecture, each task is trained in a separate network with no layer sharing. *Split-2* presents a middle ground. These architectures are designed similarly for backbones other than ResNet-18.

as an *unseen* test split, which allows us to evaluate how well networks generalize to combinations never seen during training. For a fair representation of each category and viewpoint, we ensure that every category and viewpoint class occurs exactly once in the *unseen* combinations, *ie.,* one cell each per row and column is selected. One such *unseen* test split is constructed for each dataset.

Constructing multiple train splits. Remaining cells in the *combinations grid* are used to construct multiple training splits with an increasing percentage of category-viewpoint combinations. For each training split, we first sample a set of combinations as shown in Fig. 1d, which we call the *seen* combinations. Then, we build the training data-split by sampling images from these *seen* combinations. We ensure that every category and viewpoint occurs equally in the *seen* combinations, *ie.,* equal numbers of cells per each row and column. Fig. 1d shows the 50% *seen* training split for the iLab dataset. To ensure that we evaluate the effect of data diversity and not that of data amount, number of images is kept constant across train splits as the percentage of *seen* combinations is increased. Thus, number of images per *seen* combination decreases as percentage of *seen* combinations is increased.

3.2 Architectural choices

One central question addressed in this paper is the impact of architectural choices on the capability to generalize to *unseen* category-viewpoint combinations. While many separate models have been proposed for object recognition and viewpoint estimation (Ghodrati et al., 2014; Tulsiani et al., 2015), recent years have seen a growing a trend of multi-task learning inspired architectures which suggest that recognition models can benefit from an understanding of object viewpoint, and vice versa (Penedones et al., 2012; Zhao & Itti, 2017; Massa et al., 2016; Su et al., 2015; Li et al., 2018). These architectures often learn a shared representation for both tasks, followed by task specific branches (Su et al., 2015; Zhao & Itti, 2017; Grabner et al., 2018).

Here we investigate the impact of learning shared representations on the network's capability to generalize to unseen category-viewpoint combinations *ie.,* to extrapolate in the multi-task setting of simultaneous category and viewpoint classification. For this, we defined two types of backbone agnostic architectures—the *Shared* and the *Separate* architectures. Fig. 2 depicts these architectures for a ResNet-18 backbone (He et al., 2016). In the *Shared* case, all convolutional blocks are shared between tasks, followed by task-specific fully connected layers, while there are no layers shared between tasks in the *Separate* architecture. We also investigated 3 additional *Split* architectures which represent a gradual transition from *Separate* to *Shared* ResNet-18: the *Split-1*, *Split-2*, and *Split-3* architectures. These were constructed by branching ResNet-18 after 1, 2, and 3 convolutional blocks as shown in Fig. 2. Note that splitting at a layer leads to doubling of the number of neurons in that layer. In our experiments we show that this increase in width does not provide an advantage.

4 Generalization through selectivity and invariance

Selectivity and invariance of neurons have long been hypothesized to facilitate generalization in both biological and artificial neural networks (Bricolo et al., 1997; Riesenhuber & Poggio, 1998; Goodfellow et al., 2009; Achille & Soatto, 2018; Poggio & Anselmi, 2016; Olshausen et al., 1993; Quiroga et al., 2005; Rust & DiCarlo, 2010). Neurons are commonly interpreted as image feature detectors, such that the neuron's activity is high only when certain features are present in the image (Zeiler & Fergus, 2014; Simonyan et al., 2014; Zhou et al., 2015; Bau et al., 2017; Oquab et al., 2015). We refer to this property as *selectivity* to an image feature. Selectivity alone, however, is not sufficient to generalize to *unseen* category-viewpoint combinations. For example, a neuron may be selective to features relevant to a category, but only so for a subset of all the viewpoints. Generalization is facilitated by selective neurons that are also *invariant* to nuisance features. For instance, in Fig. 1a, neurons that are selective to the Ford Thunderbird and invariant to viewpoint would have very similar activity for the Ford Thunderbird in *seen* and *unseen* viewpoints, thus enabling generalization to category recognition. Similarly, generalization to viewpoint estimation can be enabled by neurons selective to viewpoint and invariant to category.

Here, we present our implementation for quantifying the amount of *selectivity* and *invariance* of an individual neuron. Let N be the number of categories or viewpoints in the dataset. We represent the activations for a neuron across all category-viewpoint combinations as an $N \times N$ *activations grid*, as shown in Fig. 5a. Each cell in this *activations grid* represents the average activation of a neuron for images from one category-viewpoint combination, with rows and columns representing average activations for all images from a single category (*e.g.,* Ford Thunderbird) and a viewpoint (*e.g.,* front), respectively. These activations are normalized to lie between 0 and 1 (see supplement). For neuron k, we define a^k_{ij} as the entry in the *activations grid* for column (category) i and row (viewpoint) j. Below we introduce the evaluation of a neuron's *selectivity score* with respect to category and *invariance score* with respect to viewpoint. Viewpoint selectivity score and category invariance score can be derived analogously.

Selectivity score. We first identify the category that the neuron is activated for the most on average, *ie.,* the category which has the maximum sum across the column in Fig. 5a. We call this category the neuron's *preferred category*, and denote it as $i^{\star k}$, such that $i^{\star k} = \arg\max_i \sum_j a^k_{ij}$. The selectivity score compares the average activity for the *preferred category* (denoted as $\hat{a}^k$) with the average activity of the remaining categories ($\bar{a}^k$). Let S^k_c be the selectivity score with respect to category, which we define as is usual in the literature (*e.g.,* Morcos et al. (2018); Zhou et al. (2018)) with the following expression:

$$S^k_c = \frac{\hat{a}^k - \bar{a}^k}{\hat{a}^k + \bar{a}^k}, \quad \text{where } \hat{a}^k = \frac{1}{N}\sum_j a^k_{i^{\star k}j}, \;\; \bar{a}^k = \frac{\sum_{i \neq i^{\star k}} \sum_j a^k_{ij}}{N(N-1)}. \tag{1}$$

Observe that S^k_c is a value between 0 and 1, and higher values of S^k_c indicate that the neuron is more active for the *preferred category* as compared to the rest. Selectivity with respect to viewpoint, denoted as S^k_v, can be derived analogously by swapping indices (i, j).

Invariance score. A neuron's invariance to viewpoint captures the range of its average activity for the *preferred category* as the viewpoint (nuisance parameter) is changed. Let I^k_v be the invariance score with respect to viewpoint which we define as the difference between the highest and lowest activity across all viewpoints for the *preferred category*, *ie.,*

$$I^k_v = 1 - \left(\max_j \; a^k_{i^{\star k}j} - \min_j \; a^k_{i^{\star k}j}\right), \tag{2}$$

where the range is subtracted from 1 to have the invariance score equal to 1 when there is maximal invariance. Invariance with respect to category, denoted I^k_c, can be derived analogously.

Specialization score. Generalization to category recognition may be facilitated by neurons selective to category and invariant to viewpoint. Similarly, viewpoint selective and category invariant neurons can help generalize well to viewpoint estimation. This reveals a tension when category and viewpoint are learned together, as a neuron which is selective to category, cannot be invariant to category. The same is true for viewpoint. One way to resolve this contradiction is the emergence of two sets of *specialized* neurons - category selective and viewpoint invariant, and vice versa. This hypothesis is well-aligned with the findings in (Yang et al., 2019), which showed the emergence of groups of neurons contributing exclusively to single tasks. Thus, in the context of category recognition and viewpoint estimation, we hypothesize that neurons become selective to either category or viewpoint at later layers as the relevant image features for these tasks are disjoint (the category of an object cannot predict its pose, and vice-versa).

To classify neuron k as a category or viewpoint neuron, we compare its selectivity for both category and viewpoint (S^k_c and S^k_v). If S^k_c is greater than S^k_v, then neuron k is a category neuron, otherwise,

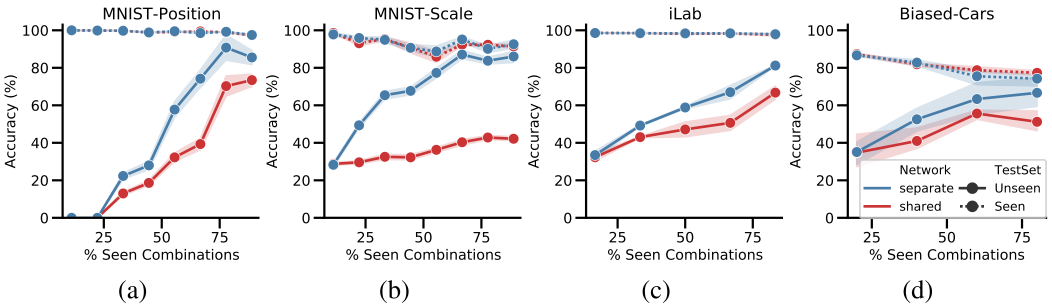


Figure 3: *Generalization performance for Shared and Separate ResNet-18 as seen combinations are increased for all datasets.* The geometric mean between category recognition accuracy and viewpoint estimation accuracy is reported along with confidence intervals (a) MNIST-Position dataset. (b) MNIST-Scale dataset. (c) iLab dataset. (d) Biased-Cars dataset.

it is a viewpoint neuron. Since generalization capability relies on both invariance and selectivity, we introduce a new metric for a neuron, the *specialization score* denoted as Γ^k, which is the geometric mean of its selectivity and invariance scores, *ie.,*

$$\Gamma^k = \begin{cases} \sqrt{S_c^k I_v^k} & \text{if } S_c^k > S_v^k \quad \text{(category neuron)} \\ \sqrt{S_v^k I_c^k} & \text{if } S_c^k \leq S_v^k \quad \text{(viewpoint neuron)} \end{cases} . \tag{3}$$

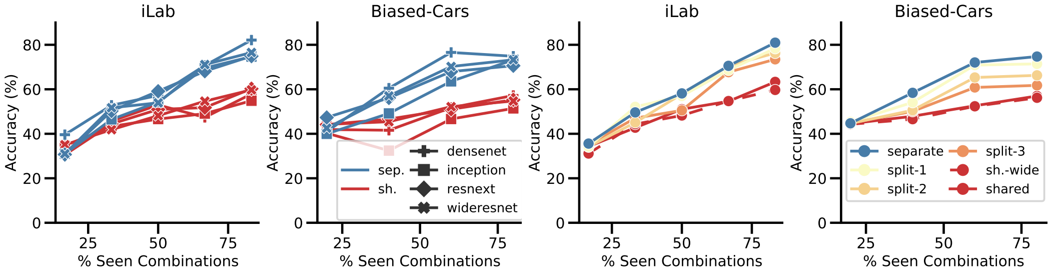
In Section 6 we present results that show that the *specialization score* is highly indicative of a network's capability to generalize to *unseen* combinations.

5 When do CNNs generalize to unseen combinations?

Below we summarize our findings from evaluating *Separate* and *Shared* architectures when tested on images from *seen* (different from train images) and *unseen* category-viewpoint combinations. See supplement for experimental details.

CNNs generalize better to *unseen* combinations as they see more combinations. Fig. 3 presents the geometric mean of category and viewpoint classification accuracy for *Separate* and *Shared* architectures with the ResNet-18 backbone, for all datasets. These experiments were repeated three times, and here we present the mean performance with confidence intervals. As Fig. 3 shows, both architectures show a significant improvement in their performance on images from *unseen* combinations, as the *seen* combinations are increased. Thus, an increase in data diversity in the form of *seen* combinations enables both these architectures to generalize better. It is to be noted that state-of-the-art CNNs do not theoretically guarantee viewpoint invariance (Poggio & Anselmi, 2016). But this result provides reassurance that CNNs can become robust to unseen category-viewpoint combinations as long as they are shown enough diversity during training. However, the accuracy for both category and viewpoint classification may not always be increasing consistently (see supplement), even though their geometric mean (shown above) is always increasing. We attribute this to the randomness in the selection of *seen* and *unseen* combinations. Also, note that as seen combinations increase, the task becomes more challenging as images with each category and pose become more diverse, leading to some drop in accuracy on seen combinations. This suggests an inherent trade-off between getting better on seen combinations and extrapolating to unseen combinations, which is impacted by training data diversity. This result adds to a growing body of works investigating the trade-offs inherent to multi-task learning (Standley et al., 2019; Shin et al., 2018).

***Separate* architectures generalize significantly better than *Shared* ones.** A striking finding that emerged from our analysis is that while both architectures perform well on new images from *seen* combinations, for images from *unseen* combinations *Separate* architectures outperform *Shared* ones by a very large margin. For the ResNet-18 backbone, this result can be seen consistently across all 4 datasets as shown in Fig. 3. Results for each individual task have been shown in the supplement. Note that previous works have shown that the *Shared* architecture is superior as it can share features among tasks. These works test on same combinations as seen during training, and when we do so we also observe that *Shared* architectures perform same or better than *Separate* (Fig.3 dashed lines).



(a) (b) (c) (d)

Figure 4: *Generalization performance for different architectures and backbones as seen combinations are increased for all iLab and Biased-City datasets.* The geometric mean between category recognition accuracy and viewpoint recognition accuracy is reported for *unseen* combinations as percentage of *seen* combinations is increased. (a) and (b) Accuracy of *separate* and *shared* for backbones other than ResNet-18, for iLab and Biased-Cars datasets, respectively. (c) and (d) Accuracy of ResNet-18 *Separate*, *Shared* and different *Split* architectures made at different blocks of the network, for iLab and Biased-Cars datasets, respectively.

We extended our analysis to *Separate* and *Shared* architectures with different backbones (ResNeXt (Xie et al., 2017), WideResNet (Zagoruyko & Komodakis, 2016), Inception v3 (Szegedy et al., 2016) and the DenseNet (Huang et al., 2017)), as shown in Fig. 4a and b. As can be seen, *Separate* architectures outperform *Shared* ones by a large margin for all backbones, which confirms that this result is not backbone specific. Investigating further, we experiment with *Split* architectures, and as can be seen in Fig. 4c and d, there is a consistent, gradual dip in the performance as we move from the *Separate* to the *Shared* architectures. Thus, generalization to *unseen* category-viewpoint combinations is best achieved by learning both tasks separately, with a consistent decrease in generalization as more parameter sharing is enforced.

To make sure that *Separate* architectures do not perform better due to the added number of neurons, we made the *Shared-Wide* architecture by doubling the neurons in each layer of the *Shared* ResNet-18 network. As Fig. 4c and d show, this architecture performs very similarly to the *Shared* one. This is in accordance with previous results that show that modern CNNs may improve in performance as the width is increased but to a limited extent (Nakkiran et al., 2019; Casper et al., 2019). See additional results in supplement along with results demonstrating that these findings are robust for a different number of training images, and also for category recognition for viewpoints not included in the training set and viewpoint estimation for new car models.

6 How do CNNs generalize to unseen combinations?

We now analyze the role of specialized (*ie.,* selective and invariant) neurons in driving generalization to *unseen* category-viewpoint combinations presented in Section 5.

Specialization score correlates with generalization to *unseen* category-viewpoint. We first investigate the emergence of category and viewpoint neurons in the final convolutional layer of the networks. Fig. 5b and c show the percentage of neurons of each type in *Shared* and *Separate* architectures as *seen* combinations are increased. As can be seen, all neurons in the category and viewpoint branches of the *Separate* architecture become specialized to category and viewpoint respectively. But in the *Shared* case, as the network is expected to simultaneously learn both tasks, both kinds of neurons emerge at a ratio of about 50%. We found that this ratio depends on the relative weight of loss terms for the two tasks. When using a different weight from the optimal in terms of maximum geometric mean accuracy, the 50% ratio of specialized neuron becomes unbalanced.

In Fig. 6 we present the median of specialization scores across neurons, *ie.,* the median of Γ^k, in the final convolutional layer for *Shared*, *Split*, and *Separate* architectures across multiple backbones in *Biased-Cars* dataset. These are presented separately for the category and viewpoint neurons. We show that as *seen* combinations increase, there is a steady increase in the specialization score for both category and viewpoint neurons, suggesting specialization. These trends mirror the generalization trends for networks which suggests that specialization facilitates generalization to *unseen* combina-

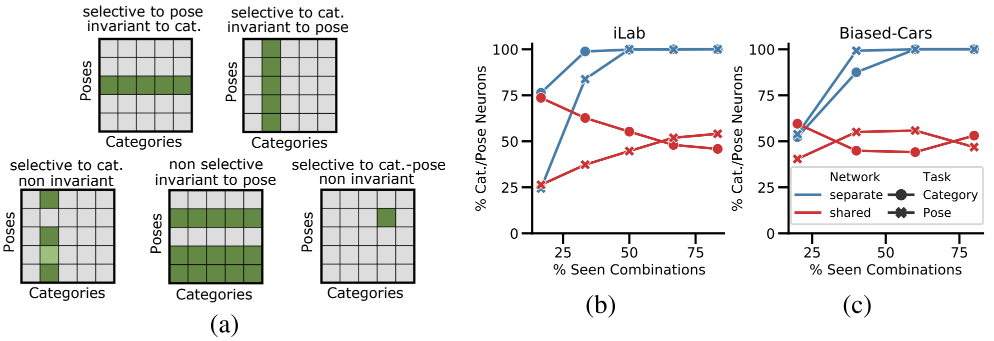


Figure 5: *Specialization to category recognition and viewpoint estimation.* (a) Prototypical *activation grids* for different types of selective and invariant neurons. (b) and (c) Percentage of neurons after ResNet-18 block-4 that are specialized to category and viewpoint, for iLab and Biased-Cars datasets, respectively. ResNet-18 *Separate* and *Shared* networks are evaluated; for *Separate*, only the task-relevant neurons for each branch are displayed.

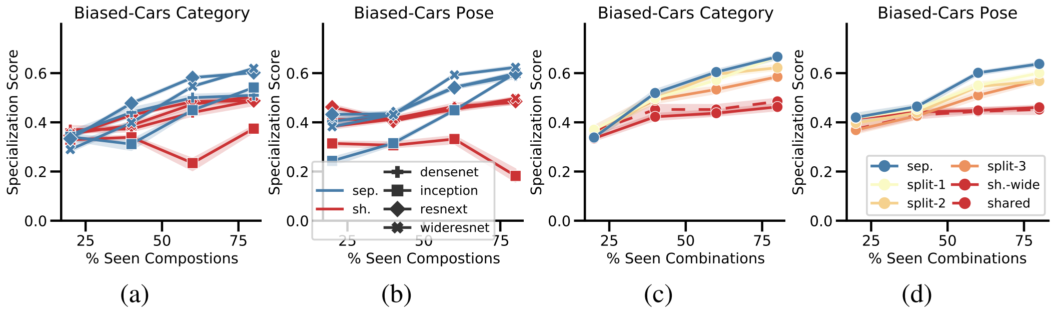


Figure 6: *Neuron specialization (selectivity to category and invariance to viewpoint, and vice versa) in the Biased-Cars dataset.* (a) and (b) Median of the specialization score among neurons (Γ^k) in network architectures, other than ResNet-18, *separate* and *shared*, for category and viewpoint classification tasks, respectively. Confidence intervals displayed in low opacity. (c) and (d) Median of the specialization score among neurons in ResNet-18 *Separate* and *Shared* with splits made at different blocks of the network, for category and viewpoint classification tasks, respectively.

tions. In the supplement, we present these results for the other datasets, which support the same conclusions. Also, we show that as expected, the specialization builds up across layers (Goodfellow et al., 2009; Poggio & Anselmi, 2016).

***Separate* networks facilitate the emergence of specialized neurons.** Fig. 6 reveals that *Separate* architectures facilitate specialization, while the *Shared* architecture makes it harder for the neurons to specialize (lower specialization scores). This might be because the *Shared* architecture tries to split into two specialized parts, but this specialization is much stronger in the *Separate* architecture due to already having separate branches. This capability to specialize could explain why the *Separate* architecture generalizes better to *unseen* category-viewpoint combinations.

7 Conclusions

Here we demonstrate that CNNs generalize better to unseen category-viewpoint combinations as the training data diversity grows. We also show that networks trained separately for category recognition and viewpoint estimation surpass by a large margin a shared network trained on both tasks when tested on *unseen* combinations. We attribute this to the branches in the *Separate* architecture not being forced to preserve information about both tasks, which facilitates an increase in the selectivity and invariance. These results are consistent across five CNN backbones and four datasets, one of them introduced in this paper as a controlled yet photo-realistic benchmark for CNN generalization. As a first step towards understanding generalization to *unseen* combinations, our work makes certain assumptions (summarized in the supplement), which present interesting directions for future work. These include understanding how generalization is impacted by—a larger number of tasks, multiple objects in the image, non-rigid objects, non-uniform ways of holding-out the test set, among others. Finally, we are intrigued to explore what other factors can help learn selective and invariant neural representations which can generalize better and lead the way towards robust, trustable CNNs.

REFERENCES

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

Ahmed J Afifi, Olaf Hellwich, and Toufique Ahmed Soomro. Simultaneous object classification and viewpoint estimation using deep multi-task convolutional neural network. In *VISIGRAPP (5: VISAPP)*, pp. 177–184, 2018.

Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4845–4854, 2019.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, pp. 9448–9458, 2019.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6541–6549, 2017.

Ali Borji, Saeed Izadi, and Laurent Itti. iLab-20M: A large-scale controlled object dataset to investigate deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2221–2230, 2016. The dataset is available at `https://bmobear.github.io/projects/viva/`.

Emanuela Bricolo, Tomaso Poggio, and Nikos K Logothetis. 3D object recognition: A model of view-tuned neurons. In *Advances in Neural Information Processing Systems*, pp. 41–47, 1997.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Stephen Casper, Xavier Boix, Vanessa D'Amario, Ling Guo, Kasper Vinken, and Gabriel Kreiman. Removable and/or repeated units emerge in overparametrized deep neural networks. *arXiv preprint arXiv:1912.04783*, 2019.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Taco S Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. *arXiv preprint arXiv:1902.04615*, 2019.

Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, Accessed 2020. URL `http://www.blender.org`.

Gilad Divon and Ayellet Tal. Viewpoint estimation—insights & model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 252–268, 2018.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the Annual Conference on Robot Learning (CoRL)*, pp. 1–16, 2017.

Mohamed Elhoseiny, Tarek El-Gaaly, Amr Bakry, and Ahmed Elgammal. A comparative analysis and study of multiview cnn models for joint object categorization and pose estimation. In *International Conference on Machine learning*, pp. 888–897, 2016.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1802–1811, 2019.

Amir Ghodrati, Marco Pedersoli, and Tinne Tuytelaars. Is 2d information enough for viewpoint estimation? *Proceedings BMVC 2014*, pp. 1–12, 2014.

C Lee Giles and Tom Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, 26(23):4972–4978, 1987.

Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, pp. 646–654, 2009.

Alexander Grabner, Peter M Roth, and Vincent Lepetit. 3d pose estimation and 3d model retrieval for objects in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3022–3031, 2018.

Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 10203–10212, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. The dataset is available at `http://yann.lecun.com/exdb/mnist/`.

Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 254–269, 2018.

Siddharth Mahendran, Haider Ali, and René Vidal. Convolutional networks for object category and 3d pose estimation from 2d images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 0–0, 2018.

Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. Cps++: Improving class-level 6d pose and shape estimation from monocular images with self-supervised learning, 2020.

Francisco Massa, Renaud Marlet, and Mathieu Aubry. Crafting a multi-task cnn for viewpoint estimation. *arXiv preprint arXiv:1609.03894*, 2016.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Pascal Mueller, Simon Haegler, Andreas Ulmer, Matthias Schubiger, Stefan Müller Arisona, and Basil Weber. *Esri CityEngine - a 3D city modeling software for urban design, visual effects, and VR/AR*. Esri R&D Center Zurich, Accessed 2020. URL `http://www.esri.com/cityengine`.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

Bruno A Olshausen, Charles H Anderson, and David C Van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719, 1993.

Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 685–694, 2015.

Mustafa Ozuysal, Vincent Lepetit, and Pascal Fua. Pose estimation for category specific multiview object localization. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 778–785. IEEE, 2009.

Hugo Penedones, Ronan Collobert, Francois Fleuret, and David Grangier. Improving object classification using pose information. Technical report, Idiap, 2012.

Tomaso Poggio and Fabio Anselmi. *Visual cortex and deep networks: learning invariant representations*. MIT Press, 2016.

Weichao Qiu and Alan Yuille. UnrealCV: Connecting computer vision to Unreal Engine. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 909–916, 2016.

R Quian Quiroga, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107, 2005.

Maximilian Riesenhuber and Tomaso Poggio. Just one view: Invariances in inferotemporal cell tuning. In *Advances in Neural Information Processing Systems*, pp. 215–221, 1998.

Nicole C Rust and James J DiCarlo. Selectivity and tolerance (“invariance”) both increase as visual information propagates from cortical area V4 to IT. *Journal of Neuroscience*, 30(39):12978–12995, 2010.

Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3061–3069, 2018.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*, 2014.

Sanjana Srivastava, Guy Ben-Yosef, and Xavier Boix. Minimal images in deep neural networks: Fragile object recognition in natural images. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv preprint arXiv:1905.07553*, 2019.

Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2686–2694, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1528, 2011.

Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1510–1519, 2015.

Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Pose induction for novel object categories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 64–72, 2015.

Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3D object detection in the wild. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 75–82. IEEE, 2014.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1500, 2017.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12, 2016.

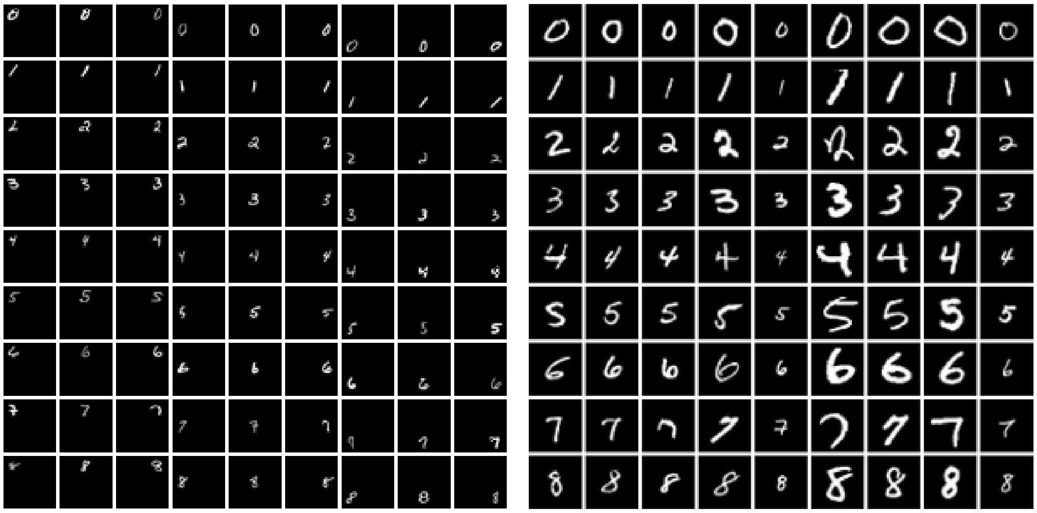
Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 818–833, 2014.

Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5287–5295, 2017.

Jiaping Zhao and Laurent Itti. Improved deep learning of object category using pose information. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 550–559, 2017.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018.



(a) MNIST-Position (b) MNIST-Scale

Figure A.1: *Combinations grids for MNIST-Position and MNIST-Scale.* Each row represents images from a category and each column from a viewpoint. (a) MNIST-Position was created by adding viewpoint in the form of position to images. For this, MNIST images were placed into one of nine positions in an empty three-by-three grid with equal probability. (b) MNIST-Scale was created by resizing images from MNIST to one of nine possible sizes, and then zero-padding.

SUPPLEMENT

A ADDITIONAL DETAILS ON DATASETS (SECTION 2)

A.1 SAMPLES FROM MNIST-POSITION AND MNIST-SCALE DATASETS

Fig. A.1 presents one representative example for each category-viewpoint combination through the *combinations grid* for the MNIST-Position and MNIST-Scale datasets.

A.2 RENDERING PIPELINE FOR *Biased-Cars* DATASET

To generate photo-realistic data with systematic, controlled biases we implemented our computer graphics pipeline which offered us fine grained control over scene attributes including but not limited to - backgrounds, textures, lighting and geometry. Below we present the details of our rendering pipeline, along with some sample images.

Pipeline Details: We used Esri CityEngine (Mueller et al., Accessed 2020) to model the city layout and geometry, to which we add 3D assets - car models, pedestrians, trees, street furniture like bus stops, textures for buildings, roads and car paints. Blender Python API (Community, Accessed 2020) is used to modify the 3D city file. This includes placing vehicles and other assets at user defined locations, modifying their material properties including vehicle paint, adding specified textures to roads, buildings and pedestrians, and defining camera attributes (lens, field of view, motion blur etc) and camera locations. For randomization, a distribution over each parameters was defined. For instance, a discrete uniform distribution over possible car color paints. Similarly, we defined distributions over object positions in the city, camera viewpoint and distance, among other factors.

Sample images are shown in Fig. A.2 below, rendered at 1024×1024 pixels. As network input was 224×224, training images were rendered at 256×256 and then resized to 224×224 (as side length of the form 2^k lead to computational gains in physically based rendering). Physically based rendering accurately models the flow of light in the scene resulting in highly photo-realistic images. As can be seen, our pipeline reproduces lighting artefacts like color bleeding and specular highlights very gracefully. As shown, images include cars seen from different distances and viewpoints, under different lighting conditions, scene clutter and even occlusions.

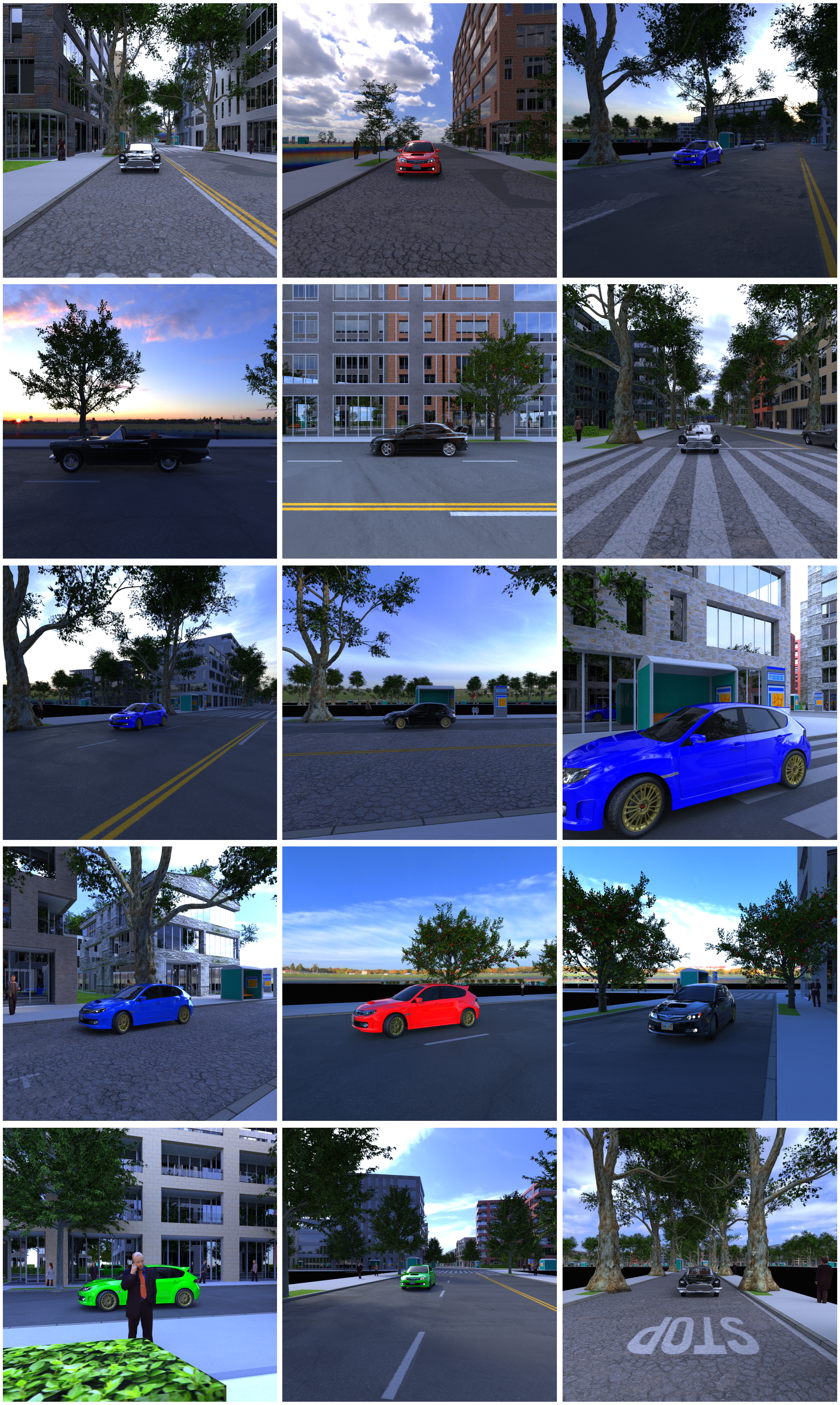


Figure A.2: *Sample images from the Biased-Cars dataset.*

B Experimental Details and Hyper-Parameters (Section 3)

Each of our four datasets contains both category and viewpoint labels for all images. We define the location and the scale as the viewpoint for MNIST-Position and MNIST-Scale datasets respectively. For both iLab and *Biased-Cars* dataset, the viewpoint refers to the azimuth viewpoint. Networks are trained to classify both category and viewpoint labels simultaneously, and all models are trained from scratch, without any pre-training to ensure controlled testing. This ensures that any existing biases in common pre-training datasets like ImageNet (Torralba & Efros, 2011) do not impact our results.

Number of Images: The number of training images is kept fixed for every dataset, and was decided by training networks on these datasets while gradually increasing size, till the performance on *unseen* combinations saturated. For the *Biased-Cars* dataset, performance plateaud at 3,400 train, 445 validation, and 800 *unseen* test images. For iLab, we used 70,000 train, 8,000 validation images, and 8,000 *unseen* test images. As the iLab dataset is a natural image dataset, it required much more images to saturate. For MNIST, 54,000 train, 8,000 validation and 8,000 test images were used.

Hyper-parameters: We used the Adam (Kingma & Ba, 2014) optimizer with 0.001 as learning rate, and ReLU activations. For the *Biased-Cars* datasets, all models were trained for 200 epochs, while we trained for 50 epochs for the iLab dataset. MNIST-Position and MNIST-Scale were trained for 5 epochs. These stopping criterion were picked to ensure convergence on generalization to *unseen* combinations. All experiments were repeated multiple times and confidence intervals are shown in the plots in the main paper. iLab and *Biased-Cars* experiments were repeated 3 times each, and MNIST experiments were repeated 10 times. Loss for training *Shared* architectures was simply the sum of CrossEntropy Loss for both category and viewpoint classification. We compared how different weighted sums perform, and found this to be performing best as measured by the geometric mean of category and viewpoint classification.

C Selectivity and Invariance (Section 4)

In the paper we defined the selectivity score of a neuron with respect to category and its invariance score with respect to viewpoint. Following the same notation as the paper: a^k_{ij} denotes the *activations grid* for neuron k, where each row represents one category and each column represents a viewpoint.

C.1 Normalization of *activations grid*

For every neuron, we first normalize its activations for every image by dividing them by its maximum activation across all images. This ensures that that the activation for every image lies between 0 and 1 for all neurons. The entries of the *activations grid* for a neuron are then computed by averaging these normalized activation for images belonging to each category-viewpoint combination.

The *activations grid* is then normalized to be between 0 and 1. To do so, we subtract the minimum of the *activations grid* and then divide it by the maximum.

C.2 Selectivity and Invariance with respect to viewpoint

In the paper, we used $i^{\star k}$, S^k_c, I^k_v to denote the *preferred category*, selectivity score with respect to category and invariance score with respect to viewpoint respectively. We also presented these equations to compute these quantities:

$$i^{\star k} = \arg\max_i \sum_j a^k_{ij}. \tag{C.1}$$

$$S^k_c = \frac{\hat{a}^k - \bar{a}^k}{\hat{a}^k + \bar{a}^k}, \quad \text{where } \hat{a}^k = \frac{1}{N}\sum_j a^k_{i^{\star k}j}, \;\; \bar{a}^k = \frac{\sum_{i \neq i^{\star k}} \sum_j a^k_{ij}}{N(N-1)}. \tag{C.2}$$

$$I^k_v = 1 - \left(\max_j \;\; a^k_{i^{\star k}j} - \min_j \;\; a^k_{i^{\star k}j}\right) \tag{C.3}$$

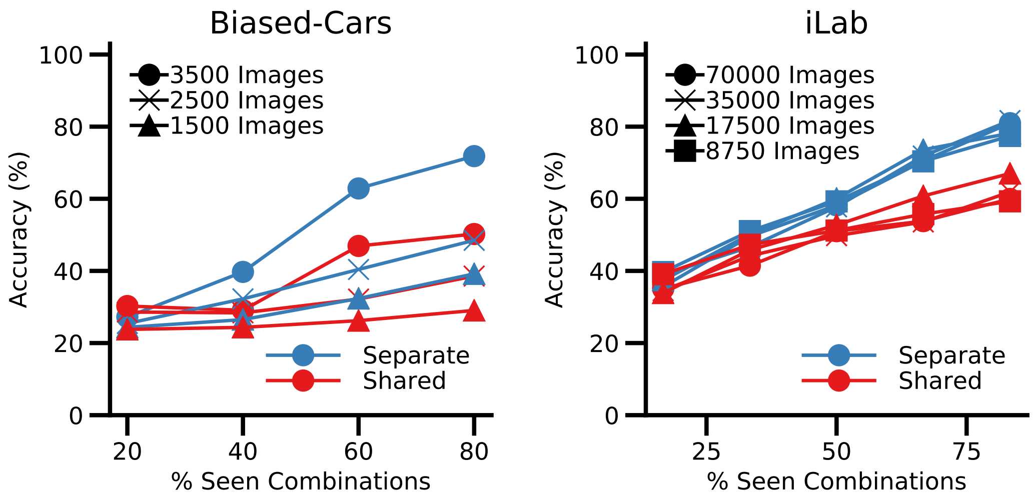


Figure D.3: *Generalization to unseen combinations as number of training images is varied.* For both iLab and Biased-Cars dataset, *Separate* architecture outperforms the *Shared* architecture trained with the same number of images.

We now present how to compute the selectivity with respect to viewpoint, and invariance with respect to category, denoted as S_v^k and I_c^k respectively. These can be obtained by first finding the *preferred viewpoint*, denoted as $j^{\star k}$, and proceeding as in the above equations:

$$j^{\star k} = \arg\max_j \sum_i a_{ij}^k. \tag{C.4}$$

$$S_v^k = \frac{\hat{a}^k - \bar{a}^k}{\hat{a}^k + \bar{a}^k}, \quad \text{where } \hat{a}^k = \frac{1}{N}\sum_i a_{ij^{\star k}}^k, \;\; \bar{a}^k = \frac{\sum_{j \neq j^{\star k}} \sum_i a_{ij}^k}{N(N-1)}. \tag{C.5}$$

$$I_c^k = 1 - \left(\max_i \;\; a_{ij^{\star k}}^k - \min_i \;\; a_{ij^{\star k}}^k \right) \tag{C.6}$$

Observe that like S_c^k, S_v^k is a value between 0 and 1, and higher value indicates that the neuron is more active for the *preferred viewpoint* as compared to the rest of the viewpoints. I_c^k too is a value between 0 and 1, with higher values indicating higher invariance to the category for images containing the *preferred viewpoint*.

D Additional Results for Section 5

Below we present additional results that re-inforce our findings presented in Section 5 of the main paper.

D.1 Number of Training Examples

To ensure that our findings are not a function of the amount of training data, we present the results for different number of images for the *Biased-Cars* and the iLab dataset in Fig. D.3. As can be seen in both these datasets, across a different number of images the *Separate* architecture substantially outperforms the *Shared* one at generalizing to *unseen* category-viewpoint combinations.

D.2 Number of Neurons in Shared *vs.* Separate Networks

To control for the number of neurons in *Shared* and *Separate* architectures, we present additional results with the *Biased-Cars* dataset in Fig. D.4. In the paper, we presented the *Shared-Wide* ar-

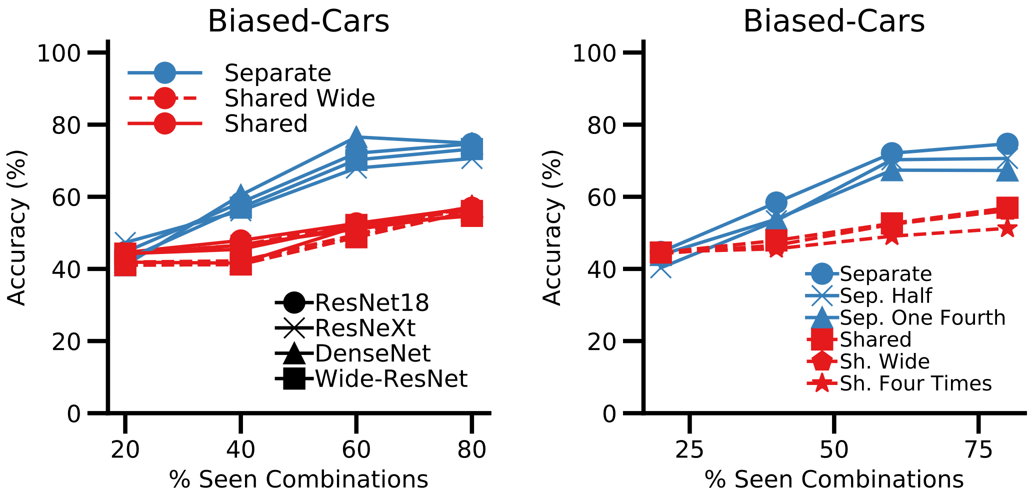


Figure D.4: *Generalization to unseen combinations as number of neurons per layer are varied for the ResNet-18 backbone. Separate* architectures substantially outperform *Shared* architectures across a range of widths, *ie.,* number of neurons per layer. The *Separate* architecture contains double the parameters as the *Shared* architecture, as there is no weight sharing in the *Separate* case. Variants of these architectures are created by increasing or decreasing the neurons in each layer by a factor of 2 at a time. Even at one-eighth the number of neurons, the *Separate One Fourth* architecture generalizes much better to unseen combinations as compared to the *Shared Four Times* architecture.

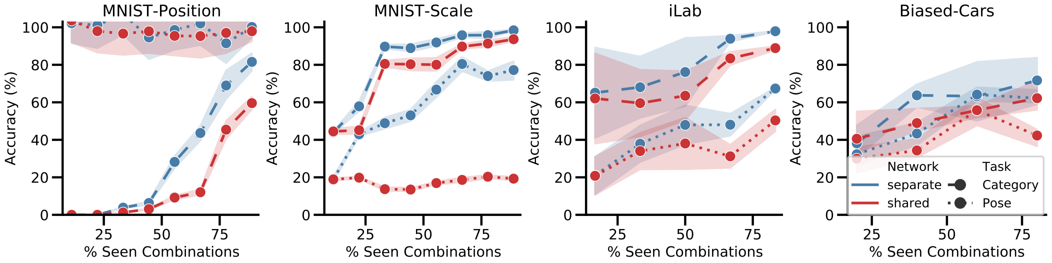
chitecture for the ResNet-18 backbone, which is the *Shared* architecture with double the number of neurons per layer, *ie.,* double the width. Here we go one step further and test a number of similar scenarios with the ResNet-18 backbone. The *Separate Half* and *Separete One Fourth* architectures are made by reducing the number of neurons in every layer to one half, and one fourth of the original number respectively. It is to be noted, that the *Separate* architectures has double the number of neurons as the *Shared* architecture, as there is no weight sharing between branches in the *Separate* case. Thus, the *Separate Half* architecture has the same number of neurons as the *Shared* architecture, and the *Separate* architecture has the same number as the *Shared-Wide* architecture. In a similar vein, the *Shared Four Times* was created by multiplying the neurons in each layer of the *Shared* architecture four times. Thus, the *Shared Four Times* has double the number of neurons as compared to the *Shared Wide* architecture, and 4 times the *Shared* architecture.

As can be seen in Fig. D.4, even at one-eighth number of neurons, the *Separate One Fourth* architecture substantially outperforms the *Shared Four Times* architecture at generalizing to *unseen* category-viewpoint combinations. This confirms that our findings are not a function of the number of neurons in the *Shared* and *Separate* architectures.

D.3 Separate performance of category and viewpoint classification

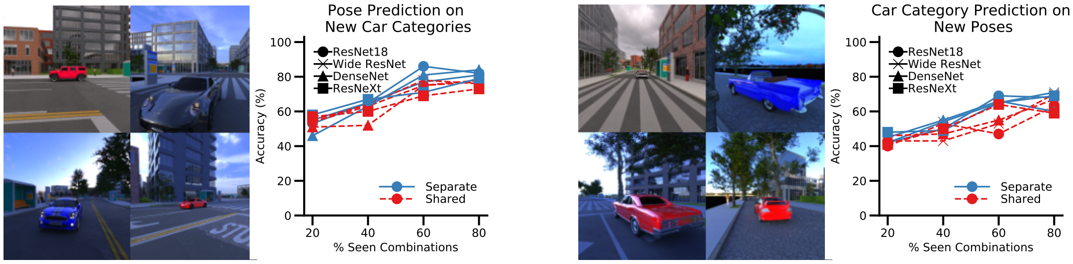
In Fig. D.5, we show that accuracy for category and viewpoint classification in *unseen* category-viewpoint combinations. The results show that *Separate* also obtains better accuracy than *Shared* for each individual task accuracy. Note that depending on the dataset, category or viewpoint classification have different degrees of difficulty and not always category recognition is more difficult than viewpoint estimation.

Furthermore, we have found that for MNIST-Position, the pooling operation at the end of ResNet-18 is critical to obtain good generalization accuracy to *unseen* category-viewpoint combinations. We evaluated ResNet-18 without the pooling operation and the category recognition accuracy of *unseen* category-viewpoint combinations dropped to baseline. Pooling facilitates an increase of position invariance and it does not harm the viewpoint estimation accuracy (as shown by Azulay & Weiss (2019), pooling does not remove the position information).



(a) (b) (c) (d)

Figure D.5: *Generalization performance for Shared and Separate ResNet-18 as seen combinations are increased for all datasets.* The category recognition accuracy and viewpoint estimation accuracy are reported along with confidence intervals (a) MNIST-Position dataset. (b) MNIST-Scale dataset. (c) iLab dataset. (d) Biased-Cars dataset.



(a) (b) (c) (d)

Figure D.6: *Controlling category and pose separately.* (a) Images of 4 new car categories, (b) Pose prediction accuracy for the 4 new car categories, (c) Images of new poses, (d) Car category prediction accuracy for the new poses.

D.4 Separate controls for category and pose prediction

We also evaluated our trained CNNs on classifying the pose of 4 new car categories (Fig. D.6a). Analogously, we also evaluated category prediction in new poses (side-to-back of car as in Fig. D.6c, instead of the front-to-side shown in training). As shown in the Fig. D.6b and d, these results confirm that our conclusions also apply to new car categories and new poses: generalization increases with more data diversity and *Separate* architecture.

D.5 Different Training Protocols

Going beyond the training protocols for Shared and Separate architecture, here we present results on two additional training protocols which have been reported in multi-task learning literature. Our networks contain three components: (i) shared layers, (ii) category branch and (iii) viewpoint branch. Here we start by training on one task first, say Category recognition. We then train the other task, *ie.,* Viewpoint estimation starting from these features learned from the first task. We call this the *Category first* protocol. The *Viewpoint first* protocol is defined analogously by starting with viewpoint prediction first, and then training for category recognition. Results for these are provided in Fig. D.7.

As can be seen, our findings are consistent with these new protocols as well. The *Separate* architecture outperforms the *Shared* architectures independent of the training protocol. Furthermore, all architectures get better with unseen combinations as seen combinations are increased.

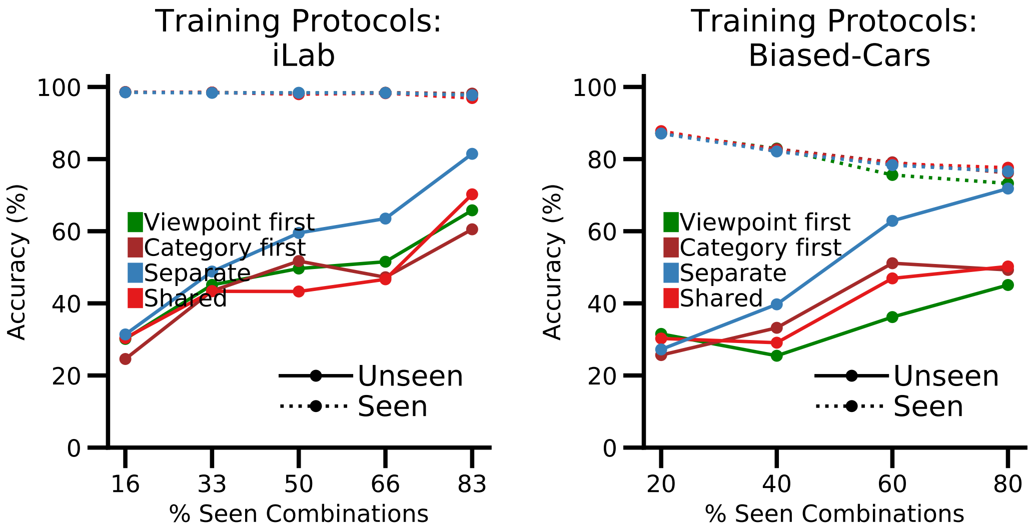


Figure D.7: *Generalization performance for additional training protocols beside Separate and Shared protocols for the ResNet-18 backbone.* The geometric mean between category recognition accuracy and viewpoint recognition accuracy is reported for unseen combinations as percentage of seen combinations is increased. For these, we start by training the shared network on only one task first *ie., Viewpoint first* or *Category first*. We then train the second task starting from these learned features from the first task. We present their comparison with our *Shared* and *Separate* training protocols presented in the paper in Section 5. (a) Accuracy for the iLab dataset, (b) Accuracy for the Biased-Cars dataset.

D.6 Results on additional datasets: UIUC 3D and MNIST-Rotation

Going beyond the four datasets presented in the paper, we replicate our analysis on two additional datasets as a confirmatory experiment: (1) the UIUC 3D Dataset, and (2) the MNIST-Rotation dataset. As can be seen from Fig.D.8 (a) and (b), our findings are consistent across these additional datasets as well - *Separate* outperforms the *Shared* independent of the training protocol, and all architectures get better at unseen combinations as seen combinations are increased.

Small size of UIUC 3D dataset: It is important to note that the small size of the UIUC 3D dataset makes it difficult to adapt it for training with biased seen combinations. We picked 8 of the total 10 object categories (to ensure symmetry as explained in Sec.2), which amounts to 5,400 images in total across 64 category-viewpoint combinations. Thus, there are only 1700 training images for the 24 seen combinations case, which is kept constant as seen combinations are increased. In contrast the other natural image dataset used in this paper, the iLab dataset contains 70,000 train images for 6 categories and viewpoints each. Due to this the generalization performance is slightly low, however the findings are still consistent as listed above. As an additional control, we also tried using all available 4500 images for the 87.5% seen case (*ie.,* all images other than the unseen test set) - generalization numbers were still low overall, but trends were preserved.

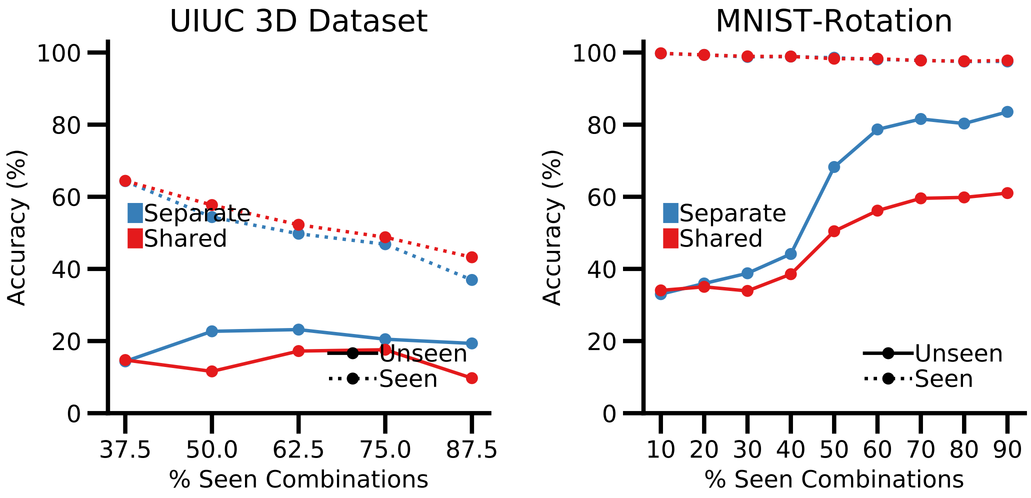


Figure D.8: *Generalization performance for additional datasets for ResNet-18 backbone.* The geometric mean between category recognition accuracy and viewpoint recognition accuracy is reported for unseen combinations as percentage of seen combinations is increased. (a) Accuracy for the UIUC 3D dataset, (b) Accuracy for the MNIST-Rotation dataset. Due to the small size of the UIUC dataset there is poor generalization - there are only 1700 train set for 37.5% seen combinations (which is kept constant as percentages of seen combinations increases). This leads to lesser generalization, but our findings still hold true - (1) Increasing seen combinations improves performance on unseen data, and (2) Separate architectures outperform shared on on unseen data.

D.7 Results on group equivariant architectures

Group and gauge equivariant CNNs have recently emerged as an alternative to standard CNNs which theoretically offer better viewpoint invariance. While these architectures (Cohen et al., 2018; 2019) are yet to be adapted to more complex datasets like ImageNet, they have shown great results on simpler image datasets like MNIST-Rotation. Here, we present results with two such architectures in Fig. D.9. As can be seen, our findings also extend to these architectures - *Separate* outperforms the *Shared* independent of the training protocol, and all architectures get better at unseen combinations as seen combinations are increased. This suggests our findings extend beyond standard CNNs. We believe that a detailed comparison between GCNNs and standard CNNs with respect to generalization to unseen combinations would be an interesting starting point for future work.

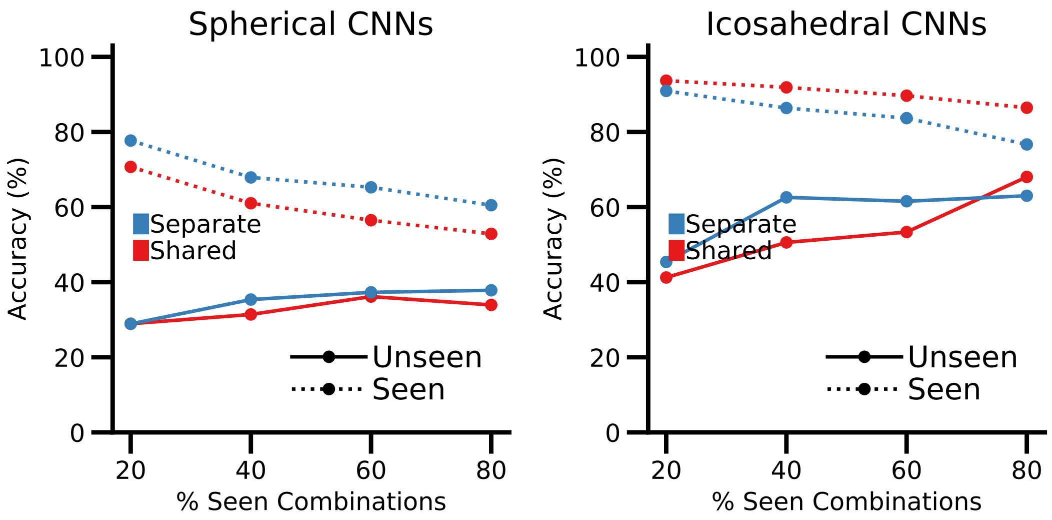


Figure D.9: *Generalization performance for different group equivariant architectures as seen combinations are increased for MNIST-Position dataset.*The geometric mean between category recognition accuracy and viewpoint recognition accuracy is reported for unseen combinations as percentage of seen combinations is increased. (a) Accuracy of separate and shared architectures using a Spherical CNN as backbone, (b) Accuracy using an Icosahedral CNN as backbone.

E Additional Results for Section 6

E.1 Specialization for other datasets

In the main paper we have presented specialization scores for the iLab and *Biased-Cars* dataset. Here we also provide these for the MNIST-Position and MNIST-Scale datasets. As can be seen, our findings are consistent across these datasets as well. Fig E.10a and b show that neurons in the final convolutional layer specialize to become either category or viewpoint neurons as more category-viewpoint combinations are shown. Category and viewpoint branches of the *Separate* architecture become completely specialized to category and viewpoint respectively. In the *Shared* architecture, both kinds of neurons emerge in roughly equal numbers. Fig E.10c and d show that as the number of *seen* combinations are increased, there is a steady increase in the specialization score for both MNIST-Position and MNIST-Scale.

In Fig. E.11, we show that the selectivity score results are also consistent in iLab for different backbones and split architectures.

E.2 Invariance and Selectivity Scores

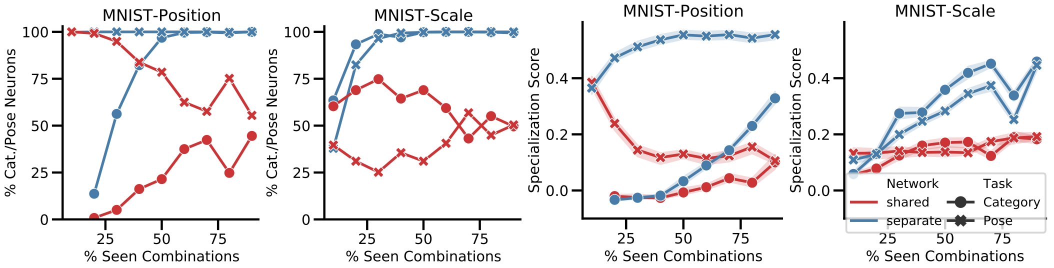
In Fig. E.12 and E.13, we show the invariance and selectivity scores separately for the *Biased-Cars* dataset. In both cases, the trends follow what we observed for the specialization score, though the differences are much more pronounced in terms of invariance rather than selectivity.

E.3 Specialization Score per Layer

In Fig. E.14, we show the specialization score in each layer. We can see that it builds up across layers, and this is more pronounced for *Separate* architectures than for *Shared*.

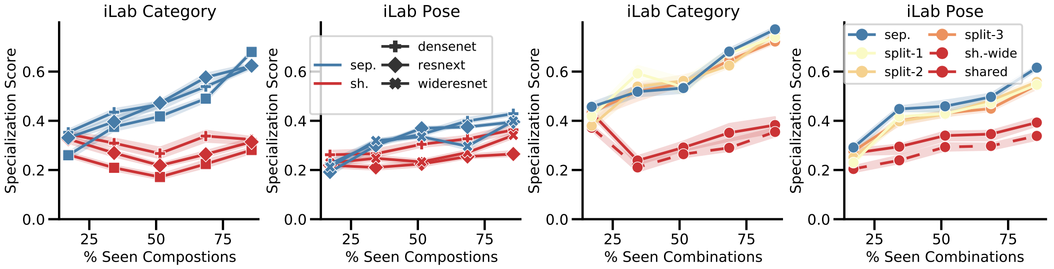
E.4 Limitations

In this paper, we have considered selectivity and invariance of individual neurons as a model for understanding generalization to *unseen* combinations. This model is limited in several ways as it only considers the properties of individual neurons, and assumes that selectivity to one single category



(a) (b) (c) (d)

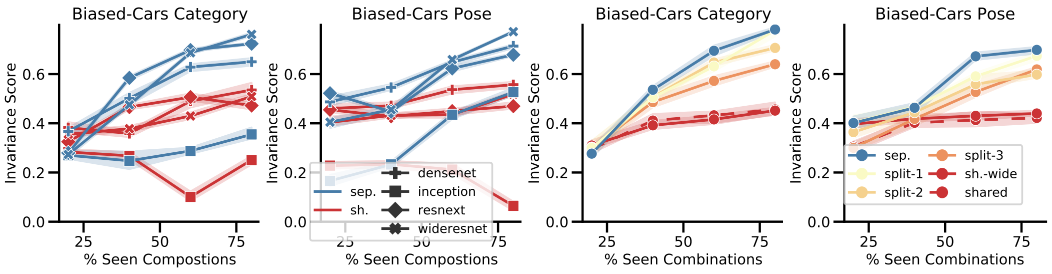
Figure E.10: *Neuron specialization in MNIST-Position and MNIST-Scale datasets.* (a) and (b) Percentage of neurons in the final convolutional layer of ResNet-18 that are specialized to category and viewpoint, for MNIST-Position and MNIST-Scale datasets, respectively. (c) and (d) Median of the specialization scores of neurons in the final convolutional layer of ResNet-18 *Separate* and *Shared* architectures, for category and viewpoint classification tasks, respectively.



(a) (b) (c) (d)

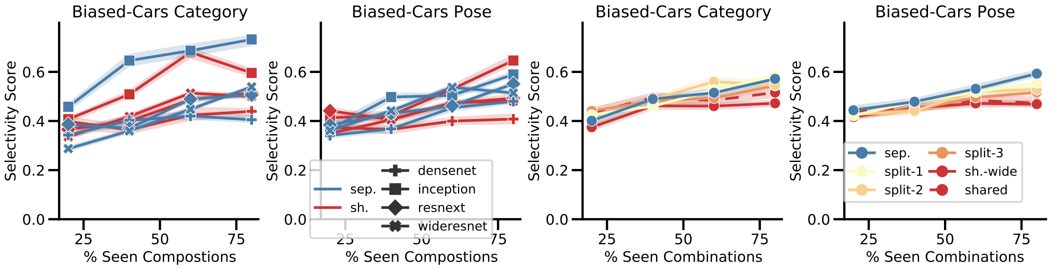
Figure E.11: *Neuron specialization (selectivity to category and invariance to viewpoint, and vice versa) in the iLab dataset.* (a) and (b) Median of the specialization score among neurons (Γ^k) in network architectures, other than ResNet-18, *separate* and *shared*, for category and viewpoint classification tasks, respectively. Confidence intervals displayed in low opacity. (c) and (d) Median of the specialization score among neurons in ResNet-18 *Separate* and *Shared* with splits made at different blocks of the network, for category and viewpoint classification tasks, respectively. Similar results for the *Biased-Cars* dataset are provided in the main paper.

(or viewpoint) is needed alongside invariance to viewpoint (or category) to achieve generalization. There could be other ways to achieve generalization not taken into account by the model. Also, the evidence presented here is correlational and based on the average neural activity for a set of images. Nonetheless, the model has been shown to be useful to explain in simple and intuitive terms why the *Separate* architecture outperforms the *Shared* one, and how these generalize as more category-viewpoint combinations are seen.



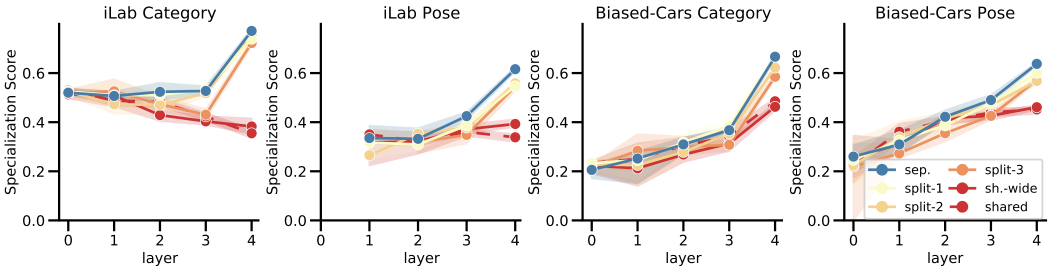
(a) (b) (c) (d)

Figure E.12: *Invariance scores in the Biased-Cars dataset.* (a) and (b) Median of the invariance score among neurons in network architectures, other than ResNet-18, *separate* and *shared*, for category and viewpoint recognition tasks, respectively. Confidence intervals displayed in low opacity. (c) and (d) Median of the invariance score among neurons in ResNet-18 *Separate* and *Shared* with splits made at different blocks of the network, for category and viewpoint recognition tasks, respectively.



(a) (b) (c) (d)

Figure E.13: *Selectivity scores in the Biased-Cars dataset.* (a) and (b) Median of the selectivity score among neurons in network architectures, other than ResNet-18, *separate* and *shared*, for category and viewpoint recognition tasks, respectively. Confidence intervals displayed in low opacity. (c) and (d) Median of the selectivity score among neurons in ResNet-18 *Separate* and *Shared* with splits made at different blocks of the network, for category and viewpoint recognition tasks, respectively.



(a) (b) (c) (d)

Figure E.14: *Specialization Score Per Layer for* 30 seen *category-viewpoint Combinations for iLab, and* 20 seen *category-viewpoint Combinations for the Biased-Cars dataset.* (a) and (b) Median of the specialization score among neurons in ResNet-18 *Separate* and *Shared* with splits made at different blocks of the network, for category and viewpoint classification tasks, respectively. (c) and (d) Same as (a) and (b) for *Biased-Cars* dataset.