The Functional Correspondence Problem

Zihang Lai* Senthil Purushwalkam* Abhinav Gupta
Carnegie Mellon University

A. Appendix

A.1. Annotation Interface

Figure 2 shows the annotation interface we used in the Amazon Mechanical Turk system. The image for labelling is shown to the left, together with the specific action we are considering. In the middle, we show 5 examples of labelled images. To the right, we show specific instructions and definitions of each point that is being labelled. Both image examples and point definition are conditioned on the given action. Three extra constraints are put on the labelled points:

1. The worker must add all keypoint annotations and use each label only once
2. The worker must annotates all points inside the given "Image Area".
3. The worker must add annotation for the difficulty.

The labelling process took around 5 days. We then check for errors in the annotations and relabel as described in the main text.

A.2. Annotation Difficulties

Figure 1 shows the level of difficulties provided by the annotators. Note annotators could be different for different categories, and the difficulty values may not be consistent across all annotators (different annotators may feel different difficulty for labelling the same image). Here, 0 means *Easy*, 0.5 means *Medium* and 1 means *Hard*. The values shown are computed from an average over all objects in the class. As one could expect, screwdriver is the easiest object category because there are very little ambiguities in defintions of each point and the shape variations are small. Two most difficult object classes are baskets and dustpan. The potential reason could be their large shape variance. For baskets, there are woven basket, shopping basket, basket with lids, without lids, and many others. Similarly, dustpan could have different handle length and orientation. Some so called lobby dustpans could have another structure that functions as a lid. As comparison, the easier object classes such as bottle, cup and tablefork have relatively little shape variance.

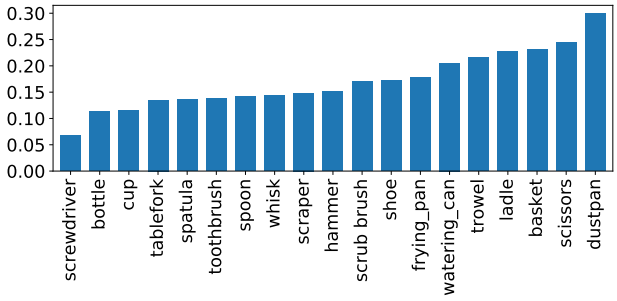

Figure 1: Level of difficulties for each object category.

A.3. Implementation Details

In this section, we provide additional hyperparameter details to facilitate easy reproduction of our results.

As explained in Sec 4.1 of the main text, our proposed model is inspired from the task-driven modular networks proposed in [1]. We design the modular network as 4 layers with 6,6,6,1 modules in each layer respectively. Here we present the hyperparameters of the convolution layers:
1st layer: 128 filters, kernel size=7, stride=1, padding=3
2nd layer: 128 filters, kernel size=3, stride=1, padding=1
3rd layer: 128 filters, kernel size=3, stride=1, padding=1
4th layer: 128 filters, kernel size=1, stride=1, padding=0

We train the modules using SGD with learning rate 0.01, momentum 0.9 and weight decay 0.00001 with a batch size of 256. The gating network takes as input a 100-dimensional embedding based on the task under consideration. The 10 embeddings for the 10 tasks in the FunKPoint dataset are randomly initialize and learned during the optimization process. The gating network consists of a 2-layer fully-connected neural network with a hidden embedding size of 100.

B. Dataset Statistics

As explained in the main text, each object could be associated with multiple actions. This leads to varying number of keypoint annotations based on object category. In Figure 3, we present these statistics:

* Authors contributed equally

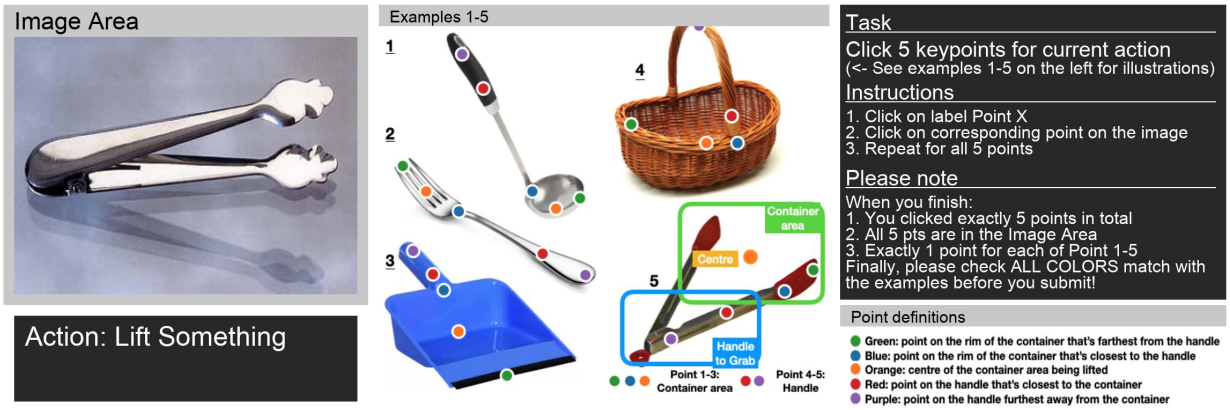

Figure 2: Annotation interface: The workers are asked the label the image to the left with instructions and examples given.

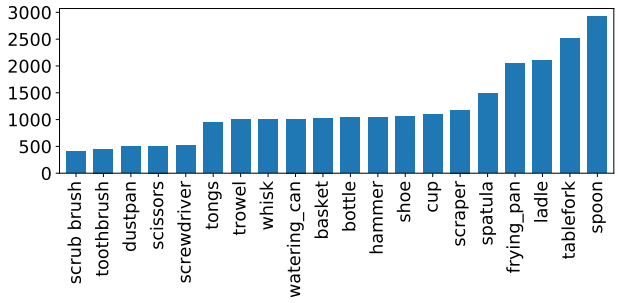

Figure 3: Number of keypoint annotations for each object category.

References

[1] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3593–3602, 2019. 1