# Few-Shot Object Detection with Foundation Models (Supplementary Material)

Guangxing Han
Columbia University
guangxinghan@gmail.com

Ser-Nam Lim
University of Central Florida
sernam@ucf.edu

## 1. Implementation Details for Model Training

We have three steps for model training, and provide the detailed training procedures next.

In the first step, we train the Deformable DETR with frozen DINOv2 as our proposal generator. Supposing that we have a ground truth set of objects $y$, and a set of N predictions $\hat{y} = \{\hat{y}\}_{i=1}^{N} = \{(\hat{p}_i, \hat{b}_i)\}_{i=1}^{N}$. Following original DETR [1], we enlarge $y = \{(c_i, b_i)\}_{i=1}^{N}$ to be a set of size N padded with $\varnothing$ (no object), and calculate the optimal bipartite matching between the two sets of objects.

$$\hat{\sigma} = \arg\min_{\sigma \in \Omega_N} \sum_{i=1}^{N} \mathcal{L}_{match}(y_i, \hat{y}_{\sigma(i)})$$

The matching loss $\mathcal{L}_{match}$ is the sum of class prediction and bounding box location loss. Then, after finding the optimal matching, the Hungarian loss $\mathcal{L}_{Hungarian}$ is used to calculate the final loss.

$$\mathcal{L}_{Hungarian} = \sum_{i=1}^{N} \left[ -\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{c_i \neq \varnothing} \mathcal{L}_{box}(b_i, b_{\hat{\sigma}(i)}) \right]$$

We train the model 50 epochs on the MSCOCO dataset with base classes only, using AdamW optimizer with an initial learning rate of 1e-4, divided by 10 after 40 epochs. For training on PASCAL VOC dataset, we use a smaller training epoch of 40 and the learning rate decreases at the 11 epoch.

In the second step, we train the LLM with few-shot proposal classification. We carefully design language instructions to prompt the LLM to classify each of the proposal. The predictions after LLM is $\overline{y} = \{(\overline{p}_i, \hat{b}_i)\}_{i=1}^{N}$. We reuse the bipartite matching $\hat{\sigma}$ in the first step to supervise the training of LLM. We use the next-token prediction objective calculated over the ground-truth.

$$\mathcal{L}_{LLM} = \sum_{i=1}^{N} -\log \overline{p}_{\hat{\sigma}(i)}(c_i)$$

The total loss $\mathcal{L}$ is the summation of LLM prediction loss $\mathcal{L}_{LLM}$ and Hungarian loss $\mathcal{L}_{Hungarian}$.

$$\mathcal{L} = \mathcal{L}_{LLM} + \mathcal{L}_{Hungarian}$$

We train the model with 3 epochs, using cosine scheduler with a peak learning rate of 2e-5 and Adam optimizer.

In the third step, we fine-tune our model on novel classes. Specifically, we first fine-tune the proposal generator using down-sampled base classes and novel classes with the same number of training samples, and using Hungarian loss $\mathcal{L}_{Hungarian}$. We train the model with 5,000 iterations, and use AdamW optimizer with an initial learning rate of 1e-4, divided by 10 after 4,000 iterations. Then, we fine-tune the LLM using base classes and up-sampled novel classes, and using $\mathcal{L}$ for training. We train the model with 1 epoch, using cosine scheduler with a peak learning rate of 2e-5 and Adam optimizer.

## 2. More Visualizations and Failure Cases

We provide more visualizations in Figure A1. Failure cases include missing small objects or occluded objects, and misclassification for confusing categories.

## References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

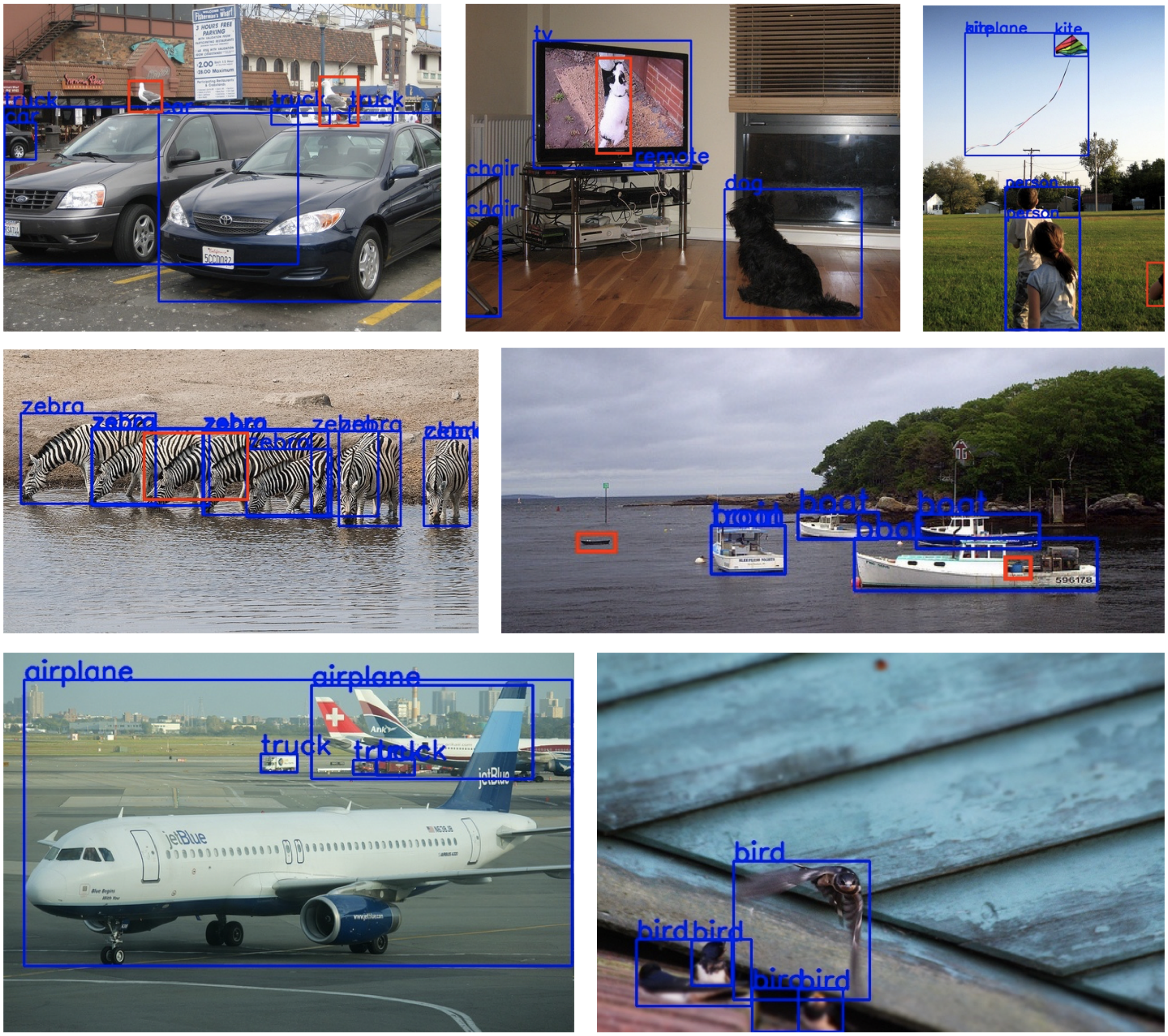

Figure A1. Detection Visualization and Failure Case Analysis. Blue means our detection results and red means false negatives. We use our 30-shot fine-tuned G-FSOD model for visualization.