S1. Removing Novel Classes from ImageNet

We propose a realistic setting for evaluating the few-shot object detection methods, where novel classes are completely removed from the classification dataset used for training a model to initialize the backbone network in the detector. This can guarantee that the object concept of novel classes will not be encoded in the pretrained model before training the few-shot detector. Because the novel class data is so rare in the real world that pretraining a classifier on it is not realistic.

ImageNet [2] is widely used for pretraining the classification model. It has 1000 classes organized according to the WordNet hierarchy. Each class has over 1000 images for training. We systematically and hierarchically remove novel classes by finding each synset and its corresponding full hyponym (synset of the whole sub-tree starting from that synset) using the ImageNet API. So each novel class may contain multiple ImageNet classes.

For the novel classes in the VOC dataset [3], their corresponding WordNet IDs to be removed are as follows.

- aeroplane: n02690373, n02692877, n04552348
- bird: n01514668, n01514859, n01518878, n01530575, n01531178, n01532829, n01534433, n01537544, n01558993, n01560419, n01580077, n01582220, n01582884, n01601694, n01608432, n01614925, n01616318, n01622779, n01795545, n01796340, n01797886, n01798484, n01806143, n01806567, n01807496, n01817953, n01818515, n01819313, n01820546, n01824575, n01828970, n01829413, n01833805, n01843065, n01847000, n01847000, n01855032, n01855672, n01860187, n02002556, n02002724, n02006656, n02007558, n02009229, n02009912, n02014460, n02012849, n02013706, n02017213, n02018207, n02018795, n02025239, n02027492, n02028035, n02033041, n02037110, n02051845, n02056570, n02058221
- boat: n02687172, n02951358, n03095699, n03344393, n03447447, n03662601, n03673027, n03873416, n03947888, n04147183, n04273569, n04347754, n04606251, n04612504
- bottle: n02823428, n03062245, n03937543, n03983396, n04552216, n04557648, n04560804, n04579145, n04591713
- bus: n03769881, n04065272, n04146614, n04487081
- cat: n02123045, n02123159, n02123394, n02123597, n02124075, n02125311, n02127052
- cow: n02403003, n02408429, n02410509
- horse: n02389026, n02391049
- motorbike: n03785016, n03791053
- sheep: n02412080, n02415577, n02417914, n02422106, n02422699, n02423022
- sofa: n04344873

S2. Visualization of Relation Reasoning

Figure S1 visualizes the correlation maps between the semantic embeddings of novel and base classes before and after the relation reasoning, as well as the difference between the two maps. Nearly all the correlations are increased slightly, indicating better knowledge propagation between the two groups of classes. Additionally, it is interesting to see that some novel classes get more correlated than others, e.g. “sofa” with “bottle” and “sofa” with “table”, probably because “sofa” can often be seen together...
with “bottle” and “table” in the living room but the original semantic embeddings cannot capture these relationships.

## S3. Using Other Word Embeddings

In the semantic space projection, we represent the semantic space using word embeddings from the Word2Vec [8]. We could simply set the $W_c$ to be random vectors. Additionally, there are other language models for obtaining vector representations for words, such as the GloVe [10]. The GloVe is trained with aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. We also explored using word embedding with different dimensions from the GloVe in the semantic space projection step and compared with the results by the Word2Vec. Performance on the VOC Novel Set 1 is reported in Table S1. The Word2Vec can provide better representations than the GloVe of both 300 dimensions and 200 dimensions. The performance of random embeddings is significantly worse than the meaningful Word2Vec and GloVe, which again verifies the importance of semantic information for shot-stable FSOD.

## S4. Reduced Dimension in Relation Reasoning

In the relation reasoning module, the dimension of word embeddings is reduced by linear layers before computing the attention map, which saves computational time. We empirically test different dimensions and select the one with the best performance, i.e. when the dimension is 32. But other choices are just slightly worse. Table S2 reports the results on VOC dataset under different dimensions. All the experiments are following the same setting as in the main paper. The only exception is that we use ResNet-50 network.
reduce the computational cost of tuning hyperparameters.

**S5. Finetuning More Parameters**

Similar to TFA [13], we have a finetuning stage to make the detector generalized to novel classes. For the classification subnet, we finetune the parameters in the relation reasoning module and the projection matrix while all the parameters in previous layers are frozen. Some may argue that the improvement of our SRR-FSD over the baseline is due to more parameters finetuned in the relation reasoning module compared to the Faster R-CNN [11] baseline. But we show that finetuning more parameters does not necessarily lead to better results in Table S3. We take the TFA model which is essentially a Faster R-CNN finetuned with only the last layer trainable and gradually unfreeze the previous layers. It turns out more parameters involved in finetuning do not change the results substantially and that too many parameters will lead to severe overfitting.

**S6. Complete Results on VOC**

In Table S4, we present the complete results on the VOC [3] dataset as in FSRW [5] and Meta R-CNN [16]. We also include the very recent MPSR [15] for comparison. MPSR develops an auxiliary branch to generate multi-scale positive samples as object pyramids and to refine the prediction at various scales. Note that MPSR improves its baseline by a considerable margin but its research direction is orthogonal and complimentary to ours because it is still exclusively dependent on visual information. Therefore, our approach combining visual information and semantic relation reasoning can achieve superior performance at extremely low shot (e.g., 1, 2) conditions.

**S7. Interpretation of the Dynamic Relation Graph**

In the relation reasoning module, we propose to learn a dynamic relation graph driven by the data, which is conceptually different from the predefined fixed knowledge graphs used in [14, 1, 9]. We implement the dynamic graph with the self-attention architecture [12]. Although it is in the form of a feedforward network, it can also be interpreted as a computation related to the knowledge graph. If we denote the transformations in the linear layers \( f, g, h, l \) as \( T_f, T_g, T_h, T_l \) respectively, we can formulate the relation reasoning in Eq. (S1)

\[
W'_e = \delta(W_e T_T T_g^T W_e^T)W_e T_T T_l + W_e
\]

where \( W'_e \) is the matrix of augmented word embeddings after the relation reasoning which will be used as the weights to compute classification scores and \( \delta \) is the softmax function operated on the last dimension of the input matrix. The item \( \delta(W_e T_T T_g^T W_e^T) \) can be interpreted as a \( N \times N \) dynamic knowledge graph in which the learnable parameters.
are $T_f$ and $T_g$. And it is involved in the computation of the classification scores via the graph convolution operation [6], which connects the $N$ word embeddings in $W_e$ to allow knowledge propagation among them. The item $T_i$ can be viewed as a learnable transformation applied to each embedding independently.

References


