Appendix
Unlocking the Full Potential of Small Data with Diverse Supervision

Ziqi Pang*[1] Zhiyuan Hu*[2] Pavel Tokmakov[3]
Yu-Xiong Wang[4] Martial Hebert[5]
[1]TuSimple [2]UCSD [3]Toyota Research Institute [4]UIUC [5]CMU
ziqi.pang@tusimple.ai, z8hu@ucsd.edu, pavel.tokmakov@tri.global,
yxw@illinois.edu, hebert@cs.cmu.edu

This appendix provides additional experimental results and details. We summarize and highlight the relationship and difference with related work in Section A. We include additional implementation and experimental details in Section B. We provide additional experimental results and analysis about the "Small Data" scenarios in Section C, including combining multiple sources of supervision and comparing the Scarce-Class and Scarce-Image settings. We include extensive evaluation on all the supervision sources in Section D. We show the cross-domain generalization of the learned representation on other datasets in Section E. We demonstrate the consistent usefulness of incorporating diverse supervision into other few-shot learning methods in Section F. We include the results of pre-trained models based on self-supervision loss in Section G. We provide the baseline of using object crops for classification in Section H. We present experiments using advanced multi-task learning methods in Section I. Finally for completeness, we provide the corresponding numbers for the experimental results shown in the form of figures in Section J.

A. Summary of Connection and Difference with Related Work

Our work is broadly related to the general investigation of learning with varying amount of data and annotation. While the detailed discussion on our proposed settings of Scarce-Class and Scarce-Image and existing work has been already covered in Section 2 of the main paper, here we *further summarize and highlight* their connection and difference in Table A. As shown in Table A, our work is unique and addresses significant limitations in existing work.

B. Additional Implementation Details

B.1. Benchmark Construction

In this section, we include additional details for constructing our LRDS benchmark, especially the algorithms for box enlargement and random jittering (Algorithm 1), which were briefly discussed in Section 3.2 of the main paper.

First, we enlarge the bounding box size by a context ratio γ, which is 2.7, computed from the average ratio between the tight bounding boxes and full images on ImageNet [2]. Then in function `RatioAssign`, we assign the enlargement in height h and width w by γ_h and γ_w, respectively. Note that γ_h and γ_w are generated randomly. They are larger than 1 and $\gamma_h \gamma_w = \gamma$.

Then in function `FindJitterRange`, we compute the range of movement for jittering the bounding box center, including the maximum range on the y-axis and x-axis $y_{min}, y_{max}, x_{min}, x_{max}$. The jittered box should still be inside the image and contain the original tight bounding box.

Finally, we randomly sample the jittering movement m_x, m_y from the computed range and apply it to the bounding box center, thus finishing the box enlargement and jittering.

Algorithm 1: Enlarging and Jittering Boxes

```
Input: Image Size: $(H, W)$, Tight BBox Size: $(h, w)$,
    BBox Center $(y, x)$, Context Ratio $\gamma$
Output: BBox for LRDS, including: Size $(\widetilde{h}, \widetilde{w})$,
    Center $(\widetilde{y}, \widetilde{x})$
$\gamma_h, \gamma_w \leftarrow$ RatioAssign$(H, W, h, w, y, x, \gamma)$
$\widetilde{h} \leftarrow h\gamma_h, \widetilde{w} \leftarrow w\gamma_w$
$y_{min}, y_{max}, x_{min}, x_{max} \leftarrow$
    FindJitterRange$(H, W, h, w, \widetilde{h}, \widetilde{w}, y, x)$
$m_x \leftarrow$ UniformSampling$(x_{min}, x_{max})$
$m_y \leftarrow$ UniformSampling$(y_{min}, y_{max})$
$\widetilde{y} \leftarrow y + m_y, \widetilde{x} \leftarrow x + m_x$
Return $\widetilde{h}, \widetilde{w}, \widetilde{y}, \widetilde{x}$
```

B.2. Model Training

In this section, we provide the implementation details for the experiments in Section 5 of the main paper. Un-

Related Settings	Included Tasks	Data			Annotations	
		Amount for Feature Learning	Data Distribution for Feature Learning	Test on Novel	Amount	Types
Scarce-Class and Scarce-Image	Single target task Multiple supervisory tasks	Small	Imbalanced	✓	Full	Multiple
Few-shot Learning	Single target task	Large	\	✓	Full	Single
Multi-task Learning	Multiple target tasks	Large	\	✗	Full	Multiple
Long-tail Learning	Single target task	Large	Imbalanced	✗	Full	Single
Weakly-supervised Learning	Single target task	Large	\	✗	Partial	Single
Unsupervised Learning	Single target task	Large	\	✗	None	None

Table A. Summary of the commonalities and differences between our proposed settings of Scarce-Class and Scarce-Image in LRDS and existing work on learning with varying amounts of data and annotation. '\' means that the setting poses no requirements on whether the training data should be balanced or imbalanced.

der the backbone of ResNet-18 [6], we experimented with varied implementations of the model and different hyper-parameter settings. In the process of data loading, we resize all the short edges of the images to the length of 800 following the protocol in the ADE20K dataset [14]. As for the model, we modify the down-sampling rate of ResNet-18, with the first three Residual Blocks yielding a down-sampling rate of 2, which together down-samples the image by 8, compared to 32 in original ResNet. During the training of the model, we use a batch size of 8, optimizer of SGD with learning rate 0.1, and cosine scheduler [9]. The whole training process of the baseline model takes 6 epochs to run, roughly 3 hours on a 4-GPU machine.

As for the combination weights for different types of supervision, we adjust the weights so that the total loss of each supervision branch has the same scale as the classification branch. The detailed values of the weight hyper-parameters are summarized in Table B.

Supervision Type	Weight
Attribute	25.0
Hierarchy	1.0
Scene	0.2
Part	25.0
Bounding Box	5.0
Segmentation	0.5
Rotation	10.0
Patch Location	1.0

Table B. Weights for different types of supervision during training.

B.3. Training Few-shot Learning Methods

In this section, we provide the details for experimenting the few-shot learning methods in Section 5.1 of the main paper. During the feature representation learning stage, we train the models following Section B.2. Then during the stage of few-shot learning on novel classes, we append and train an additional linear layer on top of the learned features.

For Cosine Classifier [1], we simply replace the linear classifier in our baseline model with a cosine classifier. For Prototypical Network [11], we use a full 193-way, 5-shot (or 1-shot) support set to calculate the mean of each category and then perform 193-way classification on a query set. We reproduce Relational Network [12] in a setting similar to Prototypical Network. For Proto-MAML [13], we first add one linear layer after the fixed feature extractor as an additional encoder. We use the 100-way novel-val set to estimate initialization parameters for the encoder. Then we evaluate model performance on the 193-way novel-test set. On both novel-val set and novel-test set, we use a prototypical network initialization as is described in [13].

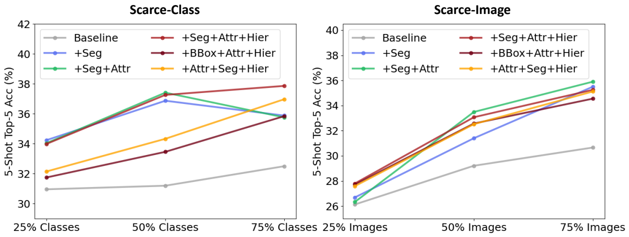

Figure A. Combining multiple sources of supervision consistently improves the performance under Scarce-Class and Scarce-Image settings.

C. Additional Analysis on "Small Data" Scenarios

This section provides additional experimental evaluation and analysis in the "Small Data" regimes.

C.1. Combining Multiple Sources of Supervision

In Section 5.4 of the main paper, we showed that combining multiple sources of supervision leads to improvements in full data settings. In this section, we further demonstrate the *consistent effectiveness* of diverse supervision in the very challenging "Small Data" regimes. As summarized in Figure A, incorporating supervision improves the performance of the baseline model by large margins, even with very few classes or images.

In addition, we have the following observations from

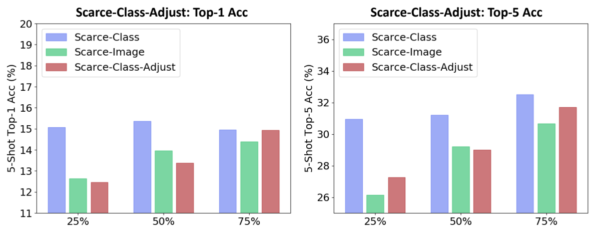

Figure B. Performance comparison of the baseline model under Scarce-Class, Scarce-Class-Adjust, and Scarce-Image settings. The new setting "Scarce-Class-Adjust" is constructed through randomly down-sampling the instances of each category in the original Scarce-Class setting, so that the total number of training samples is the same as that of the Scarce-Image setting. With the same amount of training data, the lack of classes has a similar effect as the lack of images on the performance drop of the baseline.

Figure A. First, we observe a significant performance drop when we replace the segmentation supervision with bounding box supervision. This is due to the fact that segmentation provides a stronger learning signal by exactly delineating the objects from the background, in contrast to approximately localizing them with bounding boxes. This further confirms that investing in more expensive types of annotations is valuable, especially when the number of training instances is limited.

Second, we can see that when the number of training samples is decreased, the improvements from additional supervision sources decrease. This demonstrates the limitations of not only our approach, but also modern representation learning techniques in general, since they struggle in extremely low data regimes. That said, using additional sources of supervision leads to significant improvements at moderate data scarcity levels.

Finally, we find that the performance of the "+Seg" and "+Seg+Attr" models drops in the Scarce-Class evaluation when the number of available classes increases from 50% to 75%. A potential explanation for this lies in the long-tail distribution of LRDS. Removing the tail classes from the training set makes the category-level instance distribution more balanced, thus simplifying the optimization.

C.2. Comparison Between Scarce-Image and Scarce-Class Settings

In Section 5.5 and Table 4 of the main paper, we showed that the lack of images (Scarce-Image) has a larger effect than the lack of classes (Scarce-Class) on the performance drop of the baseline. This is attributed to the fact that removing images reduces the actual number of training instances by a large margin, compared to removing the least frequent classes. To further illustrate this observation, we construct a new setting "Scarce-Class-Adjust," through randomly down-sampling the instances of each category in the original Scarce-Class setting, so that the total number of training samples is the same as that of the Scarce-Image setting. As shown in Figure B, the baseline's performance is comparable in the settings of Scarce-Class-Adjust and Scarce-Image.

D. Additional Exploration of Individual Supervision Sources

We demonstrated the effectiveness of five representative types of supervision in Section 5.2 of the main paper. In this section, we provide *extensive* evaluation on all the supervision sources, including semantic supervision in Section D.1, localization supervision in Section D.2, and self-supervision in Section D.3.

D.1. Semantic Supervision

In Section 5.2 of the main paper, we discussed the types of supervision that leverage semantic information: "Attributes," "Class Hierarchy," and "Scene Labels." Here we further investigate another type of semantic supervision: "Object Parts."

Object parts. Similar to the attributes, we use a multi-label classification loss for the object parts as shown in Figure 4 of the main paper. The ground-truth of the object parts comes from the part annotation in ADE20K [14]. The results in row 6 of Table C indicate that part labels also result in improved generalization performance, though the improvement is not as significant as that of the other types of semantic supervision.

D.2. Localization Supervision

In Section 5.2 of the main paper, we discussed the types of supervision that leverage location information: "Segmentation" and "Bounding Boxes." Here we further investigate how to use the background segmentation information, namely "Stuff Segmentation."

Stuff segmentation. For stuff segmentation, we follow FCN [8] and append a convolutional layer after the feature map, predicting a binary label for whether a pixel is object or stuff. Note that the pixels not belonging to the training set are still marked as unknown. The results in row 10 of Table C show a small decrease in performance with respect to the baseline. We hypothesize that this is because the stuff supervision forces the representation to focus on the background features, which the object classifier then latches onto.

We further combine the foreground and stuff labels together, giving a weight of 0.1 to the background classes. This combined supervision results in a performance improvement (row 11 in Table C), outperforming even the

Row Number	Type of supervision	Model	Base-val	Novel-test set 1-shot Top-1	Novel-test set 1-shot Top-5	Novel-test set 5-shot Top-1	Novel-test set 5-shot Top-5
1		Baseline	44.13	7.00	16.36	17.10	34.46
2	Semantic supervision	+Attribute	45.38	7.56	17.00	19.66	37.62
3		+Hierarchy Embedding	44.57	7.48	17.34	19.12	36.93
4		+Hierarchy Classifier	46.43	7.83	17.97	19.57	37.19
5		+Scene	45.03	7.37	18.08	18.26	36.45
6		+Part	45.68	7.39	17.23	19.39	37.2
7	Localization supervision	+Bounding Box	45.97	7.14	17.16	19.64	37.40
8		+Segmentation Region	45.68	7.68	17.35	18.95	37.46
9		+Segmentation FCN	45.82	7.84	17.53	20.02	38.26
10		+Stuff	43.86	6.58	15.04	15.63	32.65
11		+(Object+Background)	46.03	7.09	16.92	20.37	38.55
12	Self-supervision	+Rotation	44.31	6.16	15.16	17.57	35.04
13		+Patch Location	44.43	7.28	16.64	18.49	35.53

Table C. Comparison of different supervision sources on the base-validation set and novel-test set of LRDS. The models are trained with full data. All types of supervision are effective by themselves (except stuff supervision which needs to be combined together with foreground supervision). And annotated semantic and localization supervision outperforms self-supervision.

variant with foreground segmentation only. This result demonstrates that stuff supervision can still be helpful, but only when combined with foreground supervision.

D.3. Self-supervision

Figure 4 in the main paper demonstrated that self-supervision can be naturally incorporated in our framework. In this section, we discuss in detail the effect of two widely used types of self-supervision, including “Rotation” and “Relative Patch Location.”

Rotation. We follow the pretext task in [4] by rotating the input image $\{0°, 90°, 180°, 270°\}$, and training an additional classifier to predict the angle of rotation. This method leads to improvement on the learned representation (row 12 of Table C), but it is much smaller than that comes from annotated supervision.

Relative patch location. Following [3], we divide the input image into a 3×3 grid of crops. Then the center crop and another randomly picked crop are passed though a model to predict their relative locations. This self-supervision also effectively regularizes representation learning, as shown in row 13 of Table C. Similar to the rotation supervision, the improvement still cannot match the other types of annotated supervision.

E. Cross-Domain Generalization

In addition to the main experiments on LRDS, we further investigate the generalization of the learned representation on ImageNet [2]. Specifically, we use the feature representations trained on the base set of LRDS with and without additional supervision sources, and learn a linear classifier on top of them for the few-shot split of ImageNet defined in [5]. From the results in Table D, we observe that using additional sources of supervision also leads to improvements in a different domain, indicating a more generalizable representation.

Model	Top-1	Top-5
Baseline	7.22	19.79
+Attribute	8.06	21.40
+Bounding Box	7.49	21.04
+Class Hierarchy	7.59	20.78
+Segmentation	8.27	21.96
+Seg+Attr	8.49	22.86
+Seg+Attr+Hie	8.96	23.17

Table D. Investigation of cross-domain generalization of the feature representation trained on LRDS for few-shot classification on ImageNet. Leveraging additional sources of supervision leads to more generalizable representation and thus improves the performance on ImageNet as well.

F. Effect on Other Few-shot Learning Methods

We mainly focused on adding supervision on top of the linear classifier baseline in Section 5. Here in Figure C, we show the *consistent usefulness* of incorporating diverse supervision into other few-shot learning methods, such as Prototypical Network [11].

G. Pre-training with Self-supervision

In this section, we provide the performance for the models pre-trained with a self-supervised pretext task: rotation

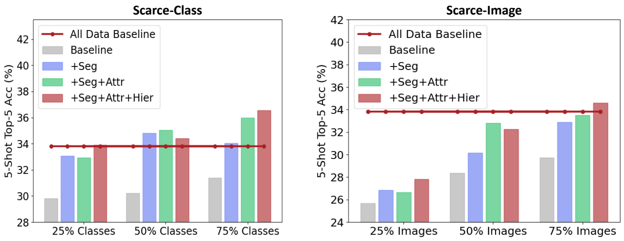

Figure C. Incorporating additional supervision is consistently effective, which also improves the performance of the Prototypical Network baseline.

Model	Top-1	Top-5
PT-Baseline	17.41	36.05
+Segmentation	17.59	36.64
+Seg+Attr	17.82	35.93
Baseline	17.10	34.46
+Segmentation	20.02	38.26
+Seg+Attr	20.96	39.41

Table E. Effect of self-supervision pre-training. PT-baseline denotes the model initialized with a set of self-supervision pre-trained parameters. Baseline denotes the randomly initialized model.

loss [4]. We first train the model with the rotation branch only, and then add the class labels for fine-tuning. To study the effect of additional supervision on this model, we further incorporate several supervision sources to the model. As demonstrated in Table E, the model pre-trained with the rotation self-supervision performs better than the baseline of training with class labels only, which is the result of better initialization. However, when adding more supervision "Seg" and "Attr," while the performance of this variant still improves, the accuracy is lower than that of the model initialized with classification pre-training. This is a counter-intuitive observation, which demonstrates that self-supervised objectives are not always beneficial for representation learning.

H. Baseline of Object Crops

In addition to the Faster R-CNN [10] based architecture in the main paper, we experiment with the most basic form of object classification, where the models operate on individual object crops instead of the whole feature map. Specifically, we crop each object out, resize each individual crop to 224 × 224, and then apply a CNN with global average pooling on it. From the numbers in Table F, this baseline achieves better performance than the Faster R-CNN [10] based architecture. This is because many objects are in small scales, and the operations of cropping and resizing enlarge them greatly.

Model	Top-1	Top-5
Baseline	21.13	42.51
+Attr	22.64	44.25
+Attr+Hie	23.43	45.34

Table F. Object crop baseline model. Since the input object size is fixed, the performance is higher than the Faster R-CNN based model.

Model	Top-1	Top-5
Baseline	17.10	34.46
+Seg+Attr [7]	18.56	37.02
+Seg+Attr+Hie [7]	18.16	36.92
+Seg+Attr (Ours)	20.96	39.41
+Seg+Attr+Hie (Ours)	21.18	39.99

Table G. Applying advanced MTL method in our model. Since we evaluate our model only by classification performance, using advanced MTL method does not lead to further improvement.

I. Advanced Multi-task Learning Methods

To explore more ways of combining multiple sources of supervision, we experiment with a more complicated multi-task learning (MTL) method from [7]. However, adding this new term does not improve the final classification performance, as shown in Table G. This is due to the difference between the objectives of our problem and MTL: we are interested in improving the classification performance on novel categories, whereas MTL is concerned with the joint improvement on all the tasks.

J. Quantitative Results for Experiments

For those experimental results shown in the form of figures in the main paper and this appendix, here for completeness we provide their corresponding numbers in Tables H, I, J, K, L, M, N, and O.

Type of supervision	Portion	Top-1	Top-5
None(Baseline)	\	17.1	34.46
Attribute	25%	18.68	36.84
Attribute	50%	18.64	37.29
Attribute	75%	18.86	36.97
Attribute	100%	18.51	37.12
Hierarchy	25%	17.97	35.95
Hierarchy	50%	18.69	37.05
Hierarchy	75%	19.31	37.76
Hierarchy	100%	18.95	38.18
BoundingBox	25%	18.64	37.51
BoundingBox	50%	18.33	37.27
BoundingBox	75%	18.46	37.04
BoundingBox	100%	18.26	37.29

Table H. Performance of varied amount of supervision. Table for Figure 5 in the main paper.

Model	Portion of Classes	Top-1	Top-5
Baseline-Full Classes	100%	17.10	34.46
Baseline	25%	15.08	30.96
Baseline	50%	15.36	31.20
Baseline	75%	14.96	32.50
+Seg+Attr+Hier	25%	17.02	33.99
+Seg+Attr+Hier	50%	18.26	37.26
+Seg+Attr+Hier	75%	19.30	37.86
+BBox+Attr+Scene	25%	16.11	31.90
+BBox+Attr+Scene	50%	17.37	34.71
+BBox+Attr+Scene	75%	18.67	35.98
+Attr+Hier+Scene	25%	15.73	31.32
+Attr+Hier+Scene	50%	16.99	33.31
+Attr+Hier+Scene	75%	18.21	35.52

Table I. The effect of supervision combinations under the Scarce-Class setting. Table for Figure 6 in the main paper, top row.

Model	Portion of Images	Top-1	Top-5
Baseline-Full Images	100%	17.10	34.46
Baseline	25%	12.64	26.13
Baseline	50%	13.96	29.21
Baseline	75%	14.39	30.66
+Seg+Attr+Hier	25%	13.05	27.80
+Seg+Attr+Hier	50%	16.52	33.07
+Seg+Attr+Hier	75%	18.03	35.25
+BBox+Attr+Scene	25%	12.95	27.88
+BBox+Attr+Scene	50%	16.50	32.66
+BBox+Attr+Scene	75%	18.01	34.75
+Attr+Hier+Scene	25%	12.65	27.16
+Attr+Hier+Scene	50%	15.50	31.93
+Attr+Hier+Scene	75%	17.56	34.08

Table J. The effect of supervision combinations under the Scarce-Image setting. Table for Figure 6 in the main paper, bottom row.

Model	Portion of Classes	Top-1	Top-5
Baseline	25%	15.08	30.96
Baseline	50%	15.36	31.20
Baseline	75%	14.96	32.50
+Seg	25%	17.31	34.24
+Seg	50%	18.76	36.87
+Seg	75%	18.11	35.88
+Seg+Attr	25%	17.27	34.04
+Seg+Attr	50%	19.38	37.41
+Seg+Attr	75%	19.00	35.76
+Seg+Attr+Hier	25%	17.02	33.99
+Seg+Attr+Hier	50%	18.26	37.26
+Seg+Attr+Hier	75%	19.30	37.86
+BBox+Attr+Hier	25%	16.06	31.75
+BBox+Attr+Hier	50%	17.21	33.46
+BBox+Attr+Hier	75%	18.77	35.84
+Attr+Seg+Hier	25%	16.59	32.15
+Attr+Seg+Hier	50%	17.92	34.33
+Attr+Seg+Hier	75%	18.52	36.97

Table K. The effect of diverse supervision under the Scarce-Class setting. Table for Figure A, left column.

Model	Portion of Images	Top-1	Top-5
Baseline	25%	12.64	26.13
Baseline	50%	13.96	29.21
Baseline	75%	14.39	30.66
+Seg	25%	12.54	26.69
+Seg	50%	14.81	31.40
+Seg	75%	17.07	35.50
+Seg+Attr	25%	12.28	26.33
+Seg+Attr	50%	16.57	33.49
+Seg+Attr	75%	17.94	35.90
+Seg+Attr+Hier	25%	13.05	27.80
+Seg+Attr+Hier	50%	16.52	33.07
+Seg+Attr+Hier	75%	18.03	35.25
+BBox+Attr+Hier	25%	12.84	27.71
+BBox+Attr+Hier	50%	15.81	32.57
+BBox+Attr+Hier	75%	17.42	34.55
+Attr+Seg+Hier	25%	12.76	27.59
+Attr+Seg+Hier	50%	16.11	32.52
+Attr+Seg+Hier	75%	16.96	35.12

Table L. The effect of diverse supervision under the Scarce-Image setting. Table for Figure A, right column.

Setting	Portion	Top-1	Top-5
Scarce-Class	25%	15.08	30.96
Scarce-Image	25%	12.64	26.13
Scarce-Class-Adjust	25%	12.47	27.26
Scarce-Class	50%	15.36	31.20
Scarce-Image	50%	13.96	29.21
Scarce-Class-Adjust	50%	13.37	29.00
Scarce-Class	75%	14.96	32.50
Scarce-Image	75%	14.39	30.66
Scarce-Class-Adjust	75%	14.93	31.70

Table M. Performance comparison of the Scarce-Class, Scarce-Image and Scarce-Class-Adjust settings. Table for Figure B.

Model	Portion of Classes	Top-1	Top-5
Baseline-Full Classes	100%	17.41	33.81
Baseline	25%	14.51	29.81
Baseline	50%	14.63	30.21
Baseline	75%	14.81	31.38
+Seg	25%	16.08	33.05
+Seg	50%	17.57	34.81
+Seg	75%	16.95	34.04
+Seg+Attr	25%	16.42	32.94
+Seg+Attr	50%	18.36	35.03
+Seg+Attr	75%	18.54	35.98
+Seg+Attr+Hier	25%	16.59	33.91
+Seg+Attr+Hier	50%	17.07	34.41
+Seg+Attr+Hier	75%	18.65	36.57

Table N. Performance of Prototypical Network with diverse supervision. Table for Figure C, left column.

References

[1] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 2

Model	Portion of Images	Top-1	Top-5
Baseline-Full Images	100%	17.41	33.81
Baseline	25%	11.43	25.69
Baseline	50%	13.10	28.38
Baseline	75%	13.77	29.71
+Seg	25%	12.42	26.85
+Seg	50%	14.33	30.16
+Seg	75%	16.44	32.90
+Seg+Attr	25%	12.62	26.66
+Seg+Attr	50%	15.93	32.82
+Seg+Attr	75%	16.79	33.50
+Seg+Attr+Hier	25%	13.24	27.80
+Seg+Attr+Hier	50%	15.92	32.27
+Seg+Attr+Hier	75%	17.66	34.59

Table O. Performance of Prototypical Network with diverse supervision. Table for Figure C, right column.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 4

[3] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICLR*, 2015. 4

[4] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 4, 5

[5] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *CVPR*, 2017. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[7] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 5

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[9] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[10] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 5

[11] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2, 4

[12] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2

[13] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *ICLR*, 2020. 2

[14] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 2, 3