We thank our reviewers for their insightful comments, and address their remarks below.

R1: Why SimCLR Episodes help supervision collapse? For example, in the ImageNet training set, all airplanes are a single category. Therefore, for ProtoNets, the loss is minimized if all aircraft have identical features, which makes it difficult to tell different models apart when evaluating on Aircraft. SimCLR Episodes prevent this collapse. SimCLR Episodes do improve nearest neighbor results: proportion of test images with 1 or more correct matches improves from 34.1% to 48.8%, and for those with 2 or more neighbors from the same train set class decreases from 55.3% to 43.3%.

R1: Shared heads. Distances between similar images should be small. Shared heads achieve this as they yield similar representations (both keys and values) for similar support and query images, even with little training.

R1: Few-shot. We followed the standard procedure for Meta-Dataset, as we were particularly interested in the kind of transfer challenges it poses. However, we agree that k-shot is also interesting and will include it.

R2: Differences with SimCLR. The original SimCLR is proposed as a pre-training method, while SimCLR Episodes reformulate SimCLR to work in an *episodic training setting* with instance discrimination over the support set images. We will improve the "SimCLR Baseline" name.

R2: Baselines. We tried self-supervised pre-training (rotations and SimCLR) and found like other works [3] that it doesn't improve over supervised pre-training. [2] has comparable numbers only on 6 datasets: 2 datasets are missing and 2 ignore the fine-grained split; for the remainder, we have better results on all but DTD. We will include these results.

	ImNet	Omni	Acraft	Bird	DTD	QDraw	Fungi	Flower	Sign	COCO
ProtoNet+R34+224+N.SGD **2x**	50.50	51.33	51.22	73.89	66.16	46.04	40.58	84.76	49.54	39.98
Chen et al. [1] orig	59.20	69.10	54.10	77.30	76.00	57.30	45.40	89.60	66.20	55.70
Chen et al. [1]+R34+224	64.41	61.22	56.09	80.69	78.86	54.89	47.29	90.00	68.90	58.60
Tian et al. LR [4]	60.14	64.92	63.12	77.69	78.59	62.48	47.12	91.60	77.51	57.00
Tian et al. LR-distill [4]	61.58	64.31	62.32	79.47	79.28	60.83	48.53	91.00	76.33	59.28
CTX	61.53	73.34	72.32	80.83	72.25	55.61	49.54	93.16	66.02	51.22
CTX+SimCLR Eps	61.54	81.88	81.53	80.30	75.64	59.91	49.76	94.19	77.00	53.48
CTX+SimCLR Eps+Aug	61.42	82.75	81.84	77.98	74.77	64.80	49.95	95.28	84.80	57.87

R3: Baselines. We agree that [1] and [4] are interesting (although they added Meta-Dataset results on April 1 and June 17, respectively, not "3-4 months earlier than the [June 5] NeurIPS deadline"). They engineered two key aspects of the training which we did not explore: (1) During training, categories are sampled *uniformly* from ImageNet (rather than sampling episodes which contain related categories); we incorporate this by sampling 50% of episodes where the category distribution is also sampled uniformly from ImageNet. (2) They use training moving averaged batch norm statistics for test episodes, rather than using batch norm statistics from the test-time support set. The table shows updated results incorporating these ideas; we outperform or match (within the confidence interval) the cited works on all datasets but DTD (which has no spatial structure for CTX to exploit). Note the especially large gap for Omni & Acraft.

R3: Apples & oranges. We are also surprised that larger networks and higher resolution make little difference. We reproduced the result by running the open-source code for the latest SotA method of Chen et al. [1] with ResNet-34 and 224×224 images and found only small improvements (1-3%) over the original ResNet18-128px on most datasets (see table), while harming datasets with a larger domain gap (Omniglot & QuickDraw). The "suspicious" loss of performance from data augmentation is also reported in Table 1 of the original SimCLR paper.

R3: Auto-Augment. CTX+Aug experiments used the exact augmentation settings described in prior work (BOHB). BOHB used only the Meta-Dataset's ILSVRC validation classes for tuning the augmentation, *not* all 1k classes.

R3: Two-stage training. We follow Meta-Datasets' standard training procedure with classification-based pre-training.

R3: Reproducibility. We will release the full code to enable reproducibility.

R3: Longer training schedule. As suggested, we re-ran ProtoNets doubling the number of episodes and doubling the episodes between LR decays ("2x" in table above). We saw no improvement.

R3: Related work. Thanks for the suggestions. Work on self-supervised (SSL) for few-shot was omitted due to an editing error. This prior work uses auxiliary losses and networks, whereas ours directly integrates SSL into episodic training. We will improve our discussion on SSL, including contrastive methods, and add the suggested few-shot papers.

R4: Failure cases. We agree that the example of green and red triangles would be a challenge for our algorithm. However, such cases are rare in Meta-Datataset, since the dataset is mostly about distinguishing between fine-grained object categories. Still, we agree this point is worth discussing. We will include this along with more qualitative examples of failures.

R4: Naming. We agree that the transformer naming convention is less-than-ideal, but computer vision now has a tradition of using 'transformer' to refer to only the attention mechanism: e.g., Fang et al. "Scene memory transformer for embodied agents in long-horizon tasks," Girdhar et al. "Video action transformer network." Using this name will help others working on similar models find our work.

[1] Y. Chen et al. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[2] G. S. Dhillon et al. A baseline for few-shot image classification. *Proc. ICLR*, 2020.

[3] J.-C. Su, S. Maji, and B. Hariharan. When does self-supervision improve few-shot learning? In *Proc. ECCV*, 2020.

[4] Y. Tian et al. Rethinking few-shot image classification: a good embedding is all you need? In *Proc. ECCV*, 2020.