A Appendix

A.1 Experimental Details

Encoder Architecture A Conv-4 network architecture is used as an encoder in all experiments. Our implementation follows the widely used architecture proposed by [34]. The architecture consists of 4 stacked network modules. Each module contains a 3x3 convolutional layer with 64 filters, batch normalization, a Relu activation and a 2x2 max pooling layer.

Open-World Few-Shot Lifelong. Meta-training tasks are sampled from the MiniImageNet training classes. Each task samples 40 classes for inclusion in the support set. For each support set class, the number of datapoints included are uniformly sampled from the range $[1, 10]$. Incremental data is sampled uniformly from the remaining 24 classes. The number of support set incremental datapoints is treated as a hyperparameter, with 100 datapoints sampled in this work. Meta-test tasks uniformly sample between $[5, 10]$ support classes from the MiniImageNet test split. Each support class samples 10 datapoints for inclusion in the query set, in addition to 20 datapoints sampled from the remaining MiniImageNet test classes. At test time, the NCM baseline computes class means using the task support set and the class means are updated after each observed label. Each model is trained for 60 epochs (each consisting of 1000 tasks) and is evaluated across 300 tasks.

Open-World Few-Shot Incremental. We create a training and test set for the incremental setting by further partitioning MiniImageNet. The MiniImageNet train data is segmented into base-class train and test data. The training set is comprised of only the base-class train data and training tasks are sampled in the same manner as the lifelong setting. The test set is constructed by augmenting the MiniImageNet test class data with the segmented base-class test data. Test tasks sample 10 datapoints for each of the 64 base-classes in the test set. Additionally, 20 incremental datapoints are sampled uniformly from the MiniImageNet test split classes. It is important to note that at test time we perform a 64+I-way classification where I is the number of observed incremental classes. Prior to evaluation, the NCM baseline initializes and caches the class means using all of the training data. Again, each model is trained for 60 epochs and evaluated over 300 tasks.