A. Illustrative Schematic of MetaQDA

To illustrate the mechanism of MetaQDA, we compare it schematically to conventional linear classifier used in many studies [2, 8, 5], and vanilla QDA in Figure 1. In the figure, the colored circles indicate 3-way-5-shot support datasets, and the "x" data points with are the query set of the corresponding color. The dashed line is the decision boundary of different classifiers. Figure 1(a) shows Nearest Centre Classifier (NCC) [8, 5], where the stars represents the mean of the support set class distributions, and these induce linear decision boundaries. Figure 1(b) depicts the Quadratic Discriminant Analysis (QDA) classifier, where the dashed ellipses represents the class covariance models, estimated from the support set. These induce a non-linear decision boundary. Figure 1(c) illustrates our MetaQDA, where the meta-training process learns a shared NIW prior (the shadow ellipse) from many few-shot training tasks. Then MetaQDA uses conjugacy to update the class covariances (solid line) using the support set and prior, and so induces a better non-linear decision boundary.

This illustrates how the MetaQDA setup allows us to exploit the benefit of a non-linear classifier, without the associated overfitting risk that would normally undermine such an attempt (as illustrated by the poor results of vanilla MetaQDA in Tab 7, 8 of the main manuscript).

B. Additional Experimental Setting Details: Standard Few-shot Learning

Parameters for training the Conv-4 extractor Following [13], we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.01 for both *mini*ImageNet and CIFAR-FS, and 0.001 for *tiered*ImageNet. At epochs 70 and 100 we reduce the learning rate by a factor of 0.1. Weight decay is set as 0.0001 through out training.

Parameters for training the ResNet-18 extractor Following [13], we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.001 for *tiered*ImageNet. At epochs 70 and 100 we reduce the learning rate by a factor of 0.1. Weight decay is set as 0.0001 throughout training. Batch size is 256 images.

Parameters for training the WRN-28-10 extractor Following [6], as for 1-shot classification on *mini*ImageNet, we use stochastic gradient descent (SGD) with a multi-step learning rate schedule, momentum of 0.9, and the initial learning rate is set to 0.001. For 5-shot classification on *mini*ImageNetand 1-shot classification on *tiered*ImageNet, we use ADAM optimiser. For CIFAR-FS, we use the pre-trained WRN backbone of S2M2.

C. Additional Experimental Setting Details: Few-Shot Class Incremental Learning

Training setup We follow the experimental setup of [10]. Specifically, we use the same 60 base classes to pre-train an initial ResNet-18 backbone using mini-batch size as 128 and use stochastic gradient descent (SGD) with the initial learning rate of 0.1, decreasing the learning rate to 0.01/0.001 after 30/40 epochs, respectively.

Meta-Training: The MetaQDA prior is then trained using Algorithm 1 (main manuscript) by generating episodes from the 60 base class set, using the feature extractor trained as above.

Meta-Testing: Due to our Bayesian class-conditional modeling, meta-testing decomposes over classes. Class-incremental learning is thus trivially realized by running MetaQDA's update step for each new category, and adding the final mean and covariance to the set used by the final QDA classifier. We apply MetaQDA both for the many-shot base classes, and 5-shot incrementally added classes.

The results in Tab 6 of the main manuscript are averages generated by independently repeating both meta-train and meta-test (8 incremental sessions each) phases 10 times.

D. Full Meta-Dataset Results

Implementation Details We use the same backbone as SUR [3] and URT [5], and take the trained fused features by URT [5]. We use ADAM optimizer and cosine learning rate scheduler, and the initial learning rate is set to 0.0003, beta is set as 0.9 and 0.999. Weight decay is set as 0.0001 throughout training. The number of training episodes is 10000.

Results Following [11], few-shot tasks are sampled with varing number of classes N, varying number of shots K and class imbalance. Table 1 reports performance in accuracy over over 600 sampled meta-test tasks. Because most of the results have very similar confidence interval, we omit this part to make the table more readable. The results of other SotA algorithms are taken from URT [5] and SCNAP[1]. From the results we can see that MetaQDA performs well in both seen domains (left) and out-of-distribution unseen (right) domains. It achieves highest performance in 8 of 13 domains within the meta-dataset benchmark.

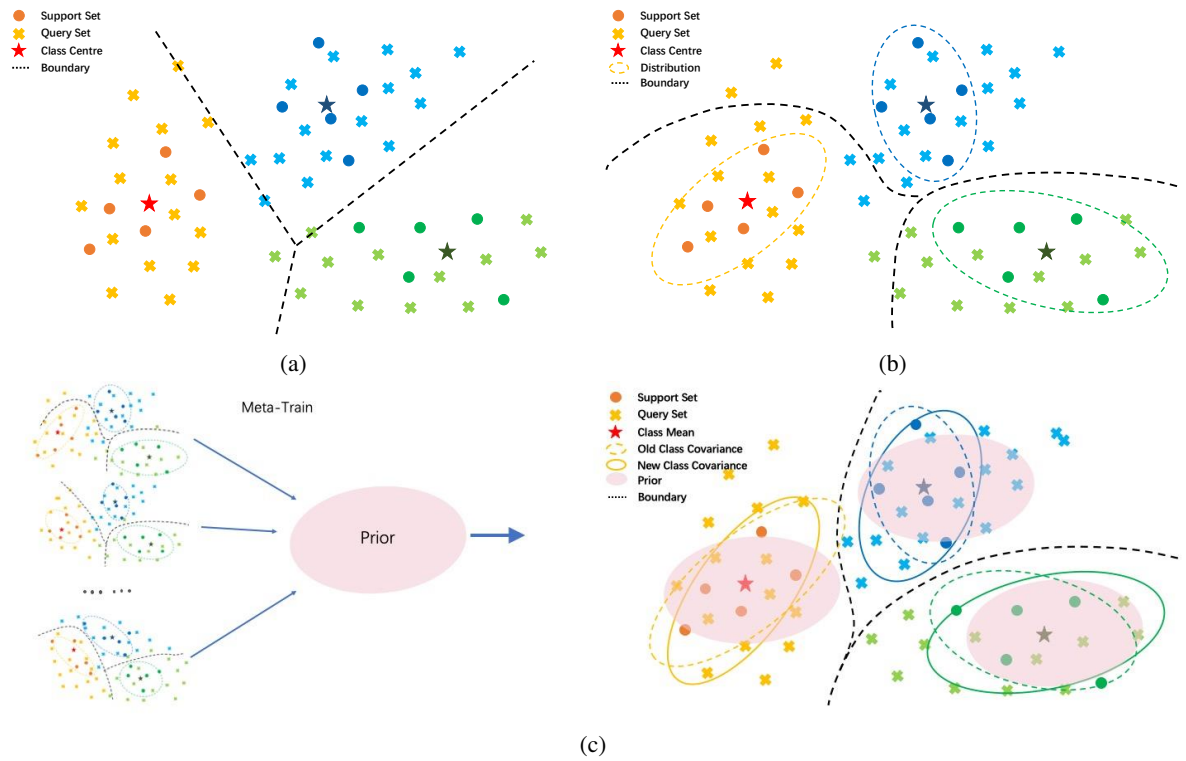

Figure 1: **Illustrative Schematic of MetaQDA.** (a) NCC classifier uses the class mean to induce linear decision boundaries. (b) QDA uses both the support class mean and covariance to induce a curved decision boundary, but easily overfits in a few-shot regime due. (c) MetaQDA meta-learns the QDA parameter prior to provide stable estimation of a non-linear decision boundary without overfitting.

Model	ImageNet	Omniglot	Aircraft	Birds	DTD	Quickdraw	Fungi	Flower	Signs	Mscoco	MNIST	CIFAR10	CIFAR100
MAML [4]	32.4	71.9	52.8	47.2	56.7	50.5	21.0	70.9	34.2	24.1	NA	NA	NA
RELATIONNET [9]	30.9	86.6	69.7	54.1	56.6	61.8	32.6	76.1	37.5	27.4	NA	NA	NA
MATCHINGNET [12]	36.1	78.3	69.2	56.4	61.8	60.8	33.7	81.9	55.6	28.8	NA	NA	NA
FINETUNE [14]	43.1	71.1	72.0	59.8	69.1	47.1	38.2	85.3	66.7	35.2	NA	NA	NA
PROTONET [8]	44.5	79.6	71.1	67.0	65.2	64.9	40.3	86.9	46.5	39.9	74.3	66.4	54.7
CNAP [7]	51.3	88.0	76.8	71.4	62.5	71.9	46.0	89.2	60.1	42.3	88.6	60.0	48.1
SCNAP [1]	**58.6**	91.7	82.4	74.9	67.8	77.7	46.9	**90.7**	73.5	46.2	93.9	74.3	60.5
SUR [3]	56.3	93.1	85.4	71.4	71.5	81.3	63.1	82.8	70.4	**52.4**	94.3	66.8	56.6
URT [5]	55.7	94.9	85.8	**76.3**	71.8	82.5	63.5	88.2	69.4	52.2	**94.8**	67.3	56.9
METAQDA	56.5	**96.3**	**86.5**	75.1	**73.4**	**82.6**	**63.7**	87.4	**73.8**	49.8	94.3	**68.2**	**57.8**

Table 1: **Full details of testing performance on the extended meta-dataset benchmark.** Left is the in-domain (seen) dataset performance, where MetaQDA ranks first 5 times in 8 domains. Right is the out-of-domain (unseen) dataset performance, where MetaQDA ranks first 3 times in 5 domains. Overall, MetaQDA has state-of-the-art performance.

References

[1] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *CVPR*, 2020. 1, 2

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

[3] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Selecting relevant features from a multi-domain representation for few-shot classification. In *ECCV*, 2020. 1, 2

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 2

[5] Lu Liu, William Hamilton, Guodong Long, Jing Jiang, and

Hugo Larochelle. A universal representation transformer layer for few-shot image classification. In *ICLR*, 2021. 1, 2

[6] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *WACV*, 2020. 1

[7] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *NIPS*, 2019. 2

[8] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 1, 2

[9] 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

[10] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020. 1

[11] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Evci, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *ICLR*, 2019. 1

[12] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 2

[13] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. 1

[14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014. 2