On Enforcing Better Conditioned Meta-Learning for Rapid Few-Shot Adaptation

Supplementary Material

Markus Hiller[1] **Mehrtash Harandi**[2] **Tom Drummond**[1]

[1]School of Computing and Information Systems, The University of Melbourne
[2]Department of Electrical and Computer Systems Engineering, Monash University
markus.hiller@student.unimelb.edu.au
mehrtash.harandi@monash.edu
tom.drummond@unimelb.edu.au

A Datasets used for experiments

This section provides additional background information about the five datasets we use to train and evaluate the baseline and our approach in the main paper.

CUB-200-2011. The Caltech-UCSD Birds-200-2011, *aka* CUB-200-2011 or simply 'CUB', is focused on fine-grained image classification tasks. It was proposed by Wah et al. (2011) and contains 11,788 images of 200 subcategories. We follow previous works like Chen et al. (2019) and use the evaluation protocol introduced by Hilliard et al. (2018), splitting the dataset into 100 classes for training, 50 for validation and 50 for testing.

***mini*ImageNet**. The *mini*ImageNet dataset has been initially proposed by (Vinyals et al., 2016) and the specific few-shot settings have been further refined in later work by (Ravi & Larochelle, 2017). It consists of a 100 class subset selected from the ImageNet dataset (Russakovsky et al., 2015) with 600 images for each class. The dataset is split into 64 training, 16 validation and 20 test classes.

***tiered*ImageNet**. The *tiered*ImageNet (Ren et al., 2018) dataset is equally a subset of classes selected from the bigger ImageNet (Russakovsky et al., 2015) dataset, however with a different structure and substantially larger set of classes. It is composed of 34 super-classes with a total of 608 categories that are split into 20, 6 and 8 super-classes totalling in 779,165 images. This unique split aims at achieving better separation between training, validation and testing, respectively.

CIFAR-FS. The CIFAR-FS dataset (Bertinetto et al., 2019) contains essentially the data from the CIFAR100 (Krizhevsky et al., 2009) dataset and splits the 100 categories of 600 images each into 64 training, 16 validation and 20 test classes.

FC-100. The FC-100 dataset (Oreshkin et al., 2018) is similarly derived from CIFAR100 (Krizhevsky et al., 2009) and split into 60 training, 20 validation and 20 test classes, but follows a splitting approach more similar to *tiered*ImageNet to increase separation between classes and difficulty.

B Effect of increased network depth

All experiments are conducted with equal contribution of all adaptation steps to the conditioning loss (as defined in Equation (9) of the main paper), with the conditioning constraint enforced with respect to the parameters of the classifier. To provide insights into the effect of increasing the depth and number of parameters onto the conditioning performance, we evaluate the following architectures on all five popular few-shot classification benchmarks for 5-shot and 1-shot settings: Convolutional networks with 4 layers (Conv4) and 6 layers (Conv6), as well as the two residual networks ResNet10

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and ResNet18. While a selection of the results has been discussed in the main paper, the test accuracies on all datasets across all architectures are presented in Table A1.

Table A1: **Increasing the network's depth and number of parameters.** Evaluations are conducted on all five popular FSL datasets: CUB-200-2011 (Wah et al., 2011), *mini*ImageNet (Vinyals et al., 2016), *tiered*ImageNet (Ren et al., 2018), CIFAR-FS (Bertinetto et al., 2019) and FC100 (Oreshkin et al., 2018). Reported are the classification accuracies on the unseen test set, averaged over 600 tasks following previous works like Chen et al. (2019).

			5-shot					1-shot				
	Network	**Method**	**step 1↑**	**step 2↑**	**step 3↑**	**step 4↑**	**step 5↑**	**step 1↑**	**step 2↑**	**step 3↑**	**step 4↑**	**step 5↑**
CUB	Conv4	MAML	20.04±0.02	24.16±0.38	64.30±0.84	74.71±0.78	77.06±0.69	50.53±0.92	57.94±0.97	60.40±0.98	60.90±0.98	61.09±0.98
		ours	61.78±0.75	73.28±0.71	75.64±0.71	76.75±0.68	77.24±0.67	55.93±0.92	61.62±0.96	62.82±0.98	63.20±0.99	63.26±0.98
	Conv6	MAML	20.00±0.00	34.14±0.62	73.58±0.82	79.40±0.70	80.52±0.65	20.06±0.04	25.85±0.48	62.54±1.02	66.78±0.98	67.67±0.98
		ours	75.42±0.72	79.21±0.65	80.01±0.65	80.44±0.63	80.65±0.63	65.15±1.06	68.28±1.06	68.54±1.07	68.76±1.06	68.87±1.06
	ResNet10	MAML	20.00±0.00	20.00±0.00	20.60±0.13	76.65±0.79	82.13±0.64	20.00±0.00	33.55±0.79	67.41±1.08	72.20±0.97	73.04±0.96
		ours	68.71±0.79	81.36±0.64	82.98±0.61	83.53±0.60	83.82±0.59	63.53±1.05	71.56±1.02	73.87±1.08	74.72±0.97	74.99±0.96
	ResNet18	MAML	20.00±0.00	20.48±0.13	43.96±0.86	79.56±0.74	83.56±0.61	20.02±0.02	25.57±0.54	72.20±0.98	74.06±0.95	74.52±0.94
		ours	81.57±0.61	84.00±0.55	84.47±0.54	84.63±0.54	84.66±0.54	66.55±1.03	72.93±0.98	74.38±0.97	74.96±0.97	75.22±0.97
*mini*ImageNet	Conv4	MAML	20.00±0.01	20.00±0.00	52.87±0.75	61.12±0.75	64.50±0.69	38.76±0.73	42.33±0.74	45.99±0.77	47.67±0.81	48.15±0.80
		ours	56.09±0.66	62.28±0.70	64.01±0.67	64.89±0.67	65.26±0.67	43.68±0.74	47.65±0.79	48.50±0.79	48.76±0.79	48.94±0.80
	Conv6	MAML	20.19±0.07	24.17±0.39	57.25±0.72	64.24±0.73	65.96±0.71	21.40±0.23	30.85±0.66	45.91±0.87	50.30±0.88	51.22±0.88
		ours	62.31±0.72	66.66±0.71	67.63±0.72	68.21±0.71	68.43±0.71	50.92±0.85	52.98±0.89	53.18±0.89	53.28±0.89	53.34±0.89
	ResNet10	MAML	20.06±0.05	20.03±0.04	41.74±0.74	63.28±0.72	69.43±0.71	20.02±0.04	22.14±0.30	50.99±0.84	56.47±0.82	57.35±0.80
		ours	53.93±0.75	68.96±0.72	71.40±0.70	72.18±0.69	72.46±0.67	50.44±0.82	55.86±0.84	57.44±0.84	57.97±0.85	58.20±0.85
	ResNet18	MAML	20.12±0.06	20.00±0.00	36.64±0.69	65.01±0.74	71.06±0.74	20.02±0.03	20.79±0.17	52.81±0.86	56.42±0.90	56.84±0.90
		ours	68.60±0.71	72.05±0.70	72.93±0.69	73.10±0.69	73.28±0.68	54.46±0.90	57.01±0.91	57.31±0.91	57.52±0.91	57.64±0.91
*tiered*ImageNet	Conv4	MAML	20.00±0.00	22.30±0.28	51.64±0.77	59.76±0.83	63.26±0.77	40.25±0.80	45.17±0.85	46.49±0.89	47.03±0.89	47.33±0.90
		ours	55.68±0.75	61.78±0.77	63.49±0.77	64.49±0.75	64.77±0.75	42.23±0.83	45.97±0.88	46.90±0.88	47.22±0.88	47.34±0.88
	Conv6	MAML	20.29±0.10	24.73±0.42	58.56±0.79	65.33±0.82	66.78±0.79	43.29±0.91	48.13±0.92	49.59±0.95	50.20±0.94	50.40±0.96
		ours	60.27±0.78	65.68±0.79	66.76±0.79	67.24±0.79	67.60±0.79	46.10±0.88	49.59±0.93	50.44±0.95	50.72±0.95	50.87±0.95
	ResNet10	MAML	20.01±0.01	20.27±0.10	36.18±0.76	67.61±0.90	73.03±0.85	20.00±0.00	24.68±0.50	52.25±0.92	57.34±0.94	57.80±0.94
		ours	58.15±0.83	71.83±0.75	74.40±0.74	75.40±0.72	75.77±0.71	52.88±0.97	58.09±0.96	59.19±0.96	59.54±0.96	59.65±0.96
	ResNet18	MAML	20.00±0.01	20.10±0.05	38.93±0.79	68.57±0.88	73.90±0.79	20.02±0.02	21.19±0.23	51.30±0.95	56.80±1.00	57.71±1.00
		ours	70.13±0.78	73.60±0.74	74.31±0.74	74.55±0.74	74.67±0.74	54.79±0.96	57.44±0.97	57.64±0.97	57.71±0.98	57.80±0.98
CIFAR-FS	Conv4	MAML	20.00±0.01	20.00±0.00	56.12±0.85	67.08±0.81	69.97±0.75	34.71±0.75	42.34±0.91	48.57±0.96	51.10±0.95	51.84±0.93
		ours	60.77±0.82	67.55±0.79	69.04±0.77	69.90±0.77	70.32±0.76	46.59±0.92	50.84±0.98	51.59±0.98	51.92±0.98	52.01±0.99
	Conv6	MAML	20.22±0.08	24.66±0.40	65.34±0.85	72.33±0.82	74.00±0.79	24.83±0.47	43.05±0.82	54.90±0.95	57.11±0.95	57.83±0.95
		ours	67.12±0.79	72.63±0.79	73.75±0.77	74.24±0.77	74.47±0.78	54.07±0.92	57.11±0.94	57.68±0.96	57.92±0.96	58.12±0.96
	ResNet10	MAML	20.04±0.03	20.00±0.00	40.23±0.87	71.29±0.87	77.12±0.77	20.00±0.00	57.47±0.97	63.57±1.01	64.93±1.00	65.42±0.98
		ours	60.93±0.87	75.65±0.76	77.67±0.72	78.45±0.72	78.90±0.70	57.58±0.96	63.97±0.94	65.63±0.94	66.07±0.94	66.17±0.95
	ResNet18	MAML	20.01±0.01	20.04±0.03	45.75±0.83	75.69±0.81	79.59±0.70	20.00±0.00	23.41±0.39	63.34±0.99	67.05±0.97	67.44±0.99
		ours	77.31±0.71	79.49±0.67	79.96±0.66	80.09±0.65	80.18±0.65	65.73±1.03	68.39±1.04	68.75±1.04	69.04±1.03	69.09±1.03
FC100	Conv4	MAML	20.00±0.00	20.00±0.00	38.03±0.66	42.96±0.74	47.61±0.72	20.04±0.03	27.98±0.57	31.12±0.68	35.17±0.76	36.08±0.76
		ours	40.46±0.68	44.43±0.72	46.90±0.72	47.51±0.73	48.01±0.73	33.84±0.72	36.16±0.75	36.68±0.76	36.77±0.76	36.82±0.76
	Conv6	MAML	20.00±0.00	20.70±0.14	39.62±0.59	42.72±0.71	46.48±0.70	30.31±0.66	32.49±0.69	34.67±0.74	35.23±0.76	35.64±0.76
		ours	41.16±0.62	45.25±0.68	46.57±0.70	47.13±0.71	47.45±0.72	33.55±0.64	35.55±0.65	36.15±0.67	36.36±0.66	36.40±0.67
	ResNet10	MAML	20.00±0.00	20.00±0.01	30.54±0.58	42.68±0.71	47.03±0.73	20.00±0.00	20.74±0.16	33.19±0.66	34.99±0.73	36.33±0.73
		ours	42.66±0.68	48.58±0.70	49.94±0.69	50.65±0.69	50.86±0.70	32.17±0.63	35.63±0.71	36.78±0.72	37.31±0.73	37.50±0.73
	ResNet18	MAML	20.00±0.00	20.00±0.00	20.00±0.00	48.46±0.73	48.56±0.76	20.00±0.00	20.04±0.02	32.30±0.64	34.37±0.69	35.19±0.70
		ours	47.11±0.73	50.16±0.73	50.82±0.74	51.08±0.74	51.22±0.74	33.24±0.69	36.05±0.71	36.70±0.72	36.90±0.72	37.02±0.72

C Adaptation beyond the training horizon

In this section, we provide further details regarding the behaviour of the baseline trained *without* ('MAML') and *with* the proposed condition loss $\mathcal{L}_\kappa$ ('ours') when the models are provided with the possibility to perform an increased number of adaptation steps beyond the training horizon **at test time only** – in our evaluated case up to 100 update steps. The training was in contrast performed with five adaptation steps. Results obtained on the test datasets for 5-way 5-shot and 5-way 1-shot scenarios are presented in Table A2 and follow the trends that have been discussed in the main paper.

Table A2: **Adaptation beyond the training horizon.** Evaluations are conducted on all five popular FSL datasets: CUB-200-2011 (Wah et al., 2011), *mini*ImageNet (Vinyals et al., 2016), *tiered*ImageNet (Ren et al., 2018), CIFAR-FS (Bertinetto et al., 2019) and FC100 (Oreshkin et al., 2018). Models have been trained with 5 inner-loop update steps, but are evaluated using additional update steps at inference time.

			5-shot					1-shot				
	Network	**Method**	**step 5↑**	**step 10↑**	**step 25↑**	**step 50↑**	**step 100↑**	**step 5↑**	**step 10↑**	**step 25↑**	**step 50↑**	**step 100↑**
CUB	Conv4	MAML	77.06±0.69	77.84±0.66	77.87±0.65	77.78±0.66	77.83±0.65	61.09±0.98	61.33±0.99	61.50±0.99	61.61±0.98	61.65±0.98
		ours	77.24±0.67	77.48±0.68	77.74±0.67	77.88±0.68	78.06±0.68	63.26±0.98	63.42±0.99	63.44±1.00	63.54±1.00	63.47±1.00
	Conv6	MAML	80.52±0.65	80.89±0.63	81.00±0.63	81.00±0.63	81.01±0.64	67.67±0.98	68.03±0.98	68.39±0.98	68.48±0.98	68.56±0.98
		ours	80.65±0.63	80.90±0.63	81.15±0.62	81.25±0.62	81.35±0.62	68.87±1.06	69.12±1.06	69.19±1.06	69.16±1.06	69.15±1.06
	ResNet10	MAML	82.13±0.64	83.90±0.59	83.99±0.59	84.09±0.59	84.08±0.59	73.04±0.96	73.56±0.96	73.80±0.96	73.97±0.96	74.01±0.96
		ours	83.82±0.59	84.09±0.58	84.11±0.59	84.26±0.57	84.28±0.57	74.99±0.96	74.99±0.97	75.19±0.97	75.24±0.97	75.33±0.97
	ResNet18	MAML	83.56±0.61	84.52±0.55	84.56±0.55	84.65±0.54	84.59±0.56	74.52±0.94	74.97±0.95	75.23±0.95	75.30±0.95	75.34±0.94
		ours	84.66±0.54	84.83±0.53	84.94±0.53	85.01±0.53	85.01±0.53	75.22±0.97	75.52±0.95	75.66±0.95	75.74±0.95	75.81±0.95
mini-ImageNet	Conv4	MAML	64.50±0.69	65.35±0.70	65.71±0.70	65.90±0.70	66.07±0.70	48.15±0.80	48.52±0.80	48.75±0.79	48.86±0.79	48.90±0.79
		ours	65.26±0.67	65.84±0.67	66.26±0.67	66.46±0.67	66.58±0.66	48.94±0.80	49.16±0.80	49.31±0.81	49.45±0.80	49.53±0.80
	Conv6	MAML	65.96±0.71	66.55±0.71	66.63±0.72	66.72±0.71	66.80±0.71	51.22±0.88	51.42±0.87	51.48±0.87	51.52±0.87	51.53±0.87
		ours	68.43±0.71	68.84±0.71	69.11±0.70	69.28±0.70	69.39±0.70	53.34±0.89	53.42±0.89	53.51±0.89	53.55±0.89	53.55±0.89
	ResNet10	MAML	69.43±0.71	71.90±0.68	72.29±0.66	72.28±0.67	72.21±0.67	57.35±0.80	57.80±0.83	57.86±0.84	57.91±0.84	58.17±0.85
		ours	72.46±0.71	73.10±0.67	73.33±0.68	73.28±0.68	73.35±0.68	58.20±0.85	58.28±0.87	58.27±0.87	58.29±0.87	58.38±0.88
	ResNet18	MAML	71.06±0.74	73.37±0.68	73.57±0.69	73.56±0.69	73.53±0.69	56.84±0.90	57.11±0.91	56.97±0.91	56.96±0.91	57.00±0.91
		ours	73.28±0.68	73.46±0.68	73.57±0.68	73.57±0.68	73.62±0.68	57.64±0.91	57.82±0.90	57.92±0.91	58.02±0.91	58.05±0.91
*tiered*ImageNet	Conv4	MAML	63.26±0.77	63.87±0.77	64.26±0.77	64.52±0.76	64.65±0.76	47.33±0.90	47.74±0.90	47.93±0.91	48.04±0.91	48.08±0.91
		ours	64.77±0.75	65.56±0.74	65.91±0.74	66.01±0.74	66.05±0.74	47.34±0.88	47.74±0.90	48.07±0.90	48.21±0.90	48.29±0.90
	Conv6	MAML	66.78±0.79	67.26±0.79	67.40±0.79	67.42±0.79	67.51±0.79	50.40±0.96	50.74±0.95	50.96±0.95	51.00±0.95	51.07±0.96
		ours	67.60±0.79	68.22±0.78	68.56±0.78	68.77±0.78	68.92±0.78	50.87±0.95	51.18±0.95	51.42±0.96	51.50±0.96	51.62±0.97
	ResNet10	MAML	73.03±0.85	75.14±0.79	75.22±0.79	75.19±0.79	75.20±0.80	57.80±0.94	57.95±0.96	58.07±0.95	58.16±0.96	58.22±0.96
		ours	75.77±0.71	76.29±0.70	76.45±0.70	74.42±0.70	76.45±0.70	59.65±0.96	59.94±0.95	60.22±0.94	60.42±0.93	60.53±0.93
	ResNet18	MAML	73.90±0.79	74.88±0.76	74.98±0.77	74.89±0.77	74.94±0.77	57.71±1.00	57.88±1.00	57.90±1.01	57.92±1.01	57.99±1.01
		ours	74.67±0.74	74.86±0.75	74.96±0.74	75.02±0.74	75.01±0.74	57.80±0.98	57.97±0.98	58.09±0.99	58.14±0.99	58.10±0.98
CIFAR-FS	Conv4	MAML	69.97±0.75	70.54±0.76	71.04±0.76	71.20±0.76	71.32±0.76	51.84±0.93	52.31±0.93	52.52±0.94	52.74±0.94	52.89±0.94
		ours	70.32±0.76	71.08±0.77	71.52±0.75	71.68±0.75	71.86±0.76	52.01±0.99	52.46±0.98	52.68±0.98	52.84±0.99	53.02±0.99
	Conv6	MAML	74.00±0.79	74.56±0.78	74.75±0.77	74.78±0.77	74.80±0.77	57.83±0.95	58.10±0.95	58.29±0.96	58.38±0.96	58.47±0.96
		ours	74.47±0.78	74.99±0.78	75.24±0.78	75.38±0.78	75.52±0.77	58.12±0.96	58.43±0.96	58.58±0.97	58.70±0.97	58.76±0.97
	ResNet10	MAML	77.12±0.77	78.75±0.69	78.77±0.69	78.72±0.70	78.69±0.71	65.42±0.98	65.82±0.99	65.81±0.99	65.98±0.98	66.03±0.97
		ours	78.90±0.70	79.30±0.68	79.39±0.68	79.34±0.69	79.37±0.69	66.17±0.95	66.04±0.96	66.07±0.97	66.16±0.96	66.24±0.96
	ResNet18	MAML	79.59±0.70	80.63±0.66	80.68±0.65	80.60±0.65	80.62±0.65	67.44±0.99	67.73±0.99	67.62±1.01	67.67±1.01	67.66±1.02
		ours	80.18±0.65	80.37±0.65	80.48±0.64	80.60±0.64	80.71±0.64	69.09±1.03	69.35±1.03	69.49±1.03	69.52±1.03	69.56±1.03
FC100	Conv4	MAML	47.61±0.72	48.40±0.71	48.79±0.73	49.14±0.73	49.25±0.73	36.08±0.76	36.45±0.76	36.66±0.77	36.82±0.77	36.89±0.76
		ours	48.01±0.73	48.59±0.72	49.01±0.72	49.30±0.72	49.52±0.72	36.82±0.76	37.08±0.76	37.24±0.76	37.31±0.76	37.45±0.75
	Conv6	MAML	46.48±0.70	47.30±0.70	47.48±0.71	47.69±0.71	47.82±0.71	35.64±0.76	35.86±0.76	35.99±0.75	36.04±0.76	36.05±0.75
		ours	47.45±0.72	48.08±0.73	48.56±0.72	48.77±0.72	48.90±0.72	36.40±0.67	36.51±0.67	36.72±0.67	36.74±0.67	36.80±0.67
	ResNet10	MAML	47.03±0.73	49.23±0.71	49.11±0.72	49.06±0.72	49.10±0.71	36.33±0.73	36.32±0.73	36.47±0.73	36.63±0.74	36.76±0.74
		ours	50.86±0.70	51.08±0.70	51.15±0.71	51.17±0.70	51.16±0.70	37.50±0.73	37.52±0.74	37.36±0.74	37.50±0.74	37.55±0.74
	ResNet18	MAML	48.56±0.76	50.25±0.76	50.29±0.76	50.16±0.75	50.18±0.76	35.19±0.70	35.36±0.72	35.44±0.72	35.48±0.72	35.57±0.72
		ours	51.22±0.74	51.41±0.74	51.55±0.74	51.52±0.74	51.63±0.74	37.02±0.72	37.27±0.72	37.26±0.72	37.37±0.72	37.40±0.72

D Ablating the proposed condition loss function

We introduced in the main paper that computing the condition number as defined in Equation (4) would ignore the distribution of all but two eigenvalues and thus unnecessarily weaken the training signal if directly used as conditioning objective. In this section, we back up this intuition with empirical insights. In detail, we contrast both versions 1) using our loss defined via the variance of the logarithmic eigenvalues of the approximated Hessian as proposed in the main paper in Equation (9) to 2) simply using the logarithmic condition number computed via the maximum and minimum eigenvalues (Table A3). We find that while using the logarithmic condition number does still lead to a significant improvement of adaptation performance especially during the first few steps when compared to its unconstrained counterpart (MAML), it is notably outperformed by our proposed loss using the variance of the eigenvalues.

Table A3: **Ablating the condition loss function.** Reported are the step-wise classification accuracies on the validation set of *mini*ImageNet (Vinyals et al., 2016) for a 5-way 5-shot scenario (Conv6).

	5-shot				
Loss $\mathcal{L}_\kappa$	**step 1↑**	**step 2↑**	**step 3↑**	**step 4↑**	**step 5↑**
var(log(ev))	63.93±1.76	68.44±1.70	69.15±1.70	69.78±1.69	69.83±1.73
log(κ)	55.30±2.04	62.03±1.87	63.95±1.82	64.98±1.87	65.36±1.86

E Preconditioning – Number of parameters and performance

As discussed in the main paper, we compare our approach to other recently published methods that aim to achieve better convergence via preconditioning. Table A4 outlines the different parameter update procedures and highlights the additionally introduced parameters of other methods (blue). Note that these parameters are required at both training and inference time, and lead to a significant increase in parameter count ranging from 96% up to 2235%. In contrast, our proposed approach does not require any additional parameters to achieve preconditioning and thus allows to use more powerful backbones if increased parameter counts can be tolerated – enabling our method to outperform others across the entire parameter-accuracy spectrum. While we show a visualization outlining the interplay between the total number of parameters and achieved accuracies within the main paper, we provide extended details regarding the explicit parameter counts and accuracy values in Table A5.

Table A4: **Preconditioned parameter updates.** Detailed are the different ways of updating the parameters for recently published preconditioning methods. Additionally introduced parameters are highlighted in blue, and are required at both training and inference time (cf. Table A5).

Method	Inner Loop Param. Update
MAML (Finn et al., 2017)	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)})$
Ours	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)})$
Meta-SGD (Li et al., 2017)	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \,\mathrm{diag}(\phi) \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)})$
MC (Park & Oliva, 2019)	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \, M(\boldsymbol{\theta}_\tau^{(k-1)}, \psi) \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)})$
ModGrad (Simon et al., 2020)	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \, M_\tau^{(k-1)}(\Psi) \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)})$
Warp-MAML (Flennerhag et al., 2019)	$\boldsymbol{\theta}_\tau^{(k)} = \boldsymbol{\theta}_\tau^{(k-1)} - \alpha \nabla_{\boldsymbol{\theta}^{(k-1)}} \mathcal{L}(\boldsymbol{\theta}_\tau^{(k-1)}, \zeta)$

Table A5: **Preconditioning methods, number of parameters and accuracies.** Obtained for 5-way 5-shot evaluated on the *mini*ImageNet test set. Reported are results for MAML (Finn et al., 2017), Meta-SGD (Li et al., 2017), MC (Park & Oliva, 2019), ModGrad (Simon et al., 2020) and Warp-MAML (Flennerhag et al., 2019). †denotes reimplemented versions (cf. Table 1, main paper).

Method	Backbone Architecture	Parameter increase↓	Test Accuracy	Rel. Acc. increase↑	#Total Parameters	#Backbone Parameters
MAML	Conv4 (32)	–	63.11±0.92	–	32,901	32,901
ours	Conv4 (32)	+ 0%	63.33±0.72	+0.3%	32,901	32,901
Meta-SGD	Conv4 (32)	+100%	64.03±0.94	+1.5%	65,802	32,901
ours	Conv6 (32)	+ 57%	64.47±0.71	+2.2%	51,525	51,525
MAML†	Conv4 (64)	–	64.50±0.69	–	121,093	121,093
ours	Conv4 (64)	+ 0%	65.26±0.67	+ 1.2%	121,093	121,093
ModGrad	Conv4 (64)	+873%	69.17±0.69	+ 7.2%	1,178,019	121,093
ours	Conv6 (64)	+ 61%	68.43±0.71	+ 6.1%	195,205	195,205
ours	Conv6 (128)	+527%	71.00±0.68	+10.1%	759,045	759,045
MAML†	Conv4 (128)	–	66.06±0.71	–	463,365	463,365
ours	Conv4 (128)	+ 0%	68.07±0.70	+3.0%	463,365	463,365
Warp-MAML	Conv4 (128)	+ 96%	68.4 ±0.92	+3.5%	906,885	463,365
MC	Conv4 (128)	+2235%	68.01±0.73	+3.0%	10,818,928	463,365
ours	Conv6 (128)	+ 64%	71.00±0.68	+7.5%	759,045	759,045

F Algorithm for better conditioned meta-learning

Algorithm 1 shows the concise form of how the conditioning loss presented in the main paper is used in the context of gradient-based few-shot meta-learning. The algorithm follows the concept introduced by Finn et al. (2017) for MAML, with the addition of using our reformulated problem setting and in this way computing the condition information for each stage of the parameters updated during the inner loop (Lines 8 and 9). The outer loop then incorporates the conditioning constraint

(Line 12) as introduced in Equations (9) and (10) of the main paper and computes the overall task loss (Line 13). After completing all tasks in the current task batch, the network's parameters are then updated (Line 15) by considering both the classification and condition objectives, encouraging the model to learn a well-conditioned parameter space while solving the classification challenge.

Algorithm 1 Learning a Better Conditioned Parameter Space

Require: $p(\mathcal{T})$; α, β, γ ▷ Distribution over tasks; Hyperparameters

1: $\boldsymbol{\theta}^* \leftarrow$ Random initialization
2: **while** not done **do**
3: $\quad \{\tau_1, \dots, \tau_B\} \sim p(\mathcal{T})$ ▷ Sample a batch of tasks
4: $\quad$ **for all** τ_i **do**
5: $\quad\quad \boldsymbol{\theta}^0_{\tau_i} \leftarrow \boldsymbol{\theta}^*$
6: $\quad\quad (\mathcal{D}^{\text{train}}_{\tau_i}, \mathcal{D}^{\text{val}}_{\tau_i}) \sim \tau_i$ ▷ Sample train and validation set
7: $\quad\quad$ **for** k in $\{1, \dots, K\}$ inner-loop update steps **do** ▷ Inner-loop adaptation
8: $\quad\quad\quad$ Compute $\mathbf{J}^{(k)}$ via $\mathcal{L}(\mathcal{D}^{\text{train}}_{\tau_i}, \boldsymbol{\theta}^{(k-1)}_{\tau_i})$ ▷ Following Equations (5) - (8)
9: $\quad\quad\quad$ Compute and temporarily store $\boldsymbol{\lambda}(\mathbf{J}^{(k)}\mathbf{J}^{(k)\top})$ ▷ Eigenvalues of approx. Hessian
10: $\quad\quad\quad \boldsymbol{\theta}^{(k)}_{\tau_i} \leftarrow \boldsymbol{\theta}^{(k-1)}_{\tau_i} - \alpha \nabla_{\boldsymbol{\theta}^{(k-1)}_{\tau_i}} \mathcal{L}(\mathcal{D}^{\text{train}}_{\tau_i}, \boldsymbol{\theta}^{(k-1)}_{\tau_i})$ ▷ Inner-loop parameter update
11: $\quad\quad$ **end for**
12: $\quad\quad \mathcal{L}_\kappa\left(\boldsymbol{\theta}^{(K)}_{\tau_i}(\mathcal{D}^{\text{train}}_{\tau_i}, \boldsymbol{\theta}^*)\right) = \frac{1}{K}\sum_{k=1}^{K} \text{Var}\left(\log_{10}\left(\boldsymbol{\lambda}\left(\mathbf{J}^{(k)}\mathbf{J}^{(k)^\top}\right)\right)\right)$ ▷ Cond. loss
13: $\quad\quad \mathcal{L}_{\tau_i} = \mathcal{L}\left(\mathcal{D}^{\text{val}}_{\tau_i},\ \boldsymbol{\theta}^{(K)}_{\tau_i}(\mathcal{D}^{\text{train}}_{\tau_i}, \boldsymbol{\theta}^*)\right) + \gamma \mathcal{L}_\kappa\left(\boldsymbol{\theta}^{(K)}_{\tau_i}(\mathcal{D}^{\text{train}}_{\tau_i}, \boldsymbol{\theta}^*)\right)$ ▷ Overall task loss
14: $\quad$ **end for**
15: $\quad \boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* - \beta \nabla_{\boldsymbol{\theta}^*} \sum_{i=1}^{B} \mathcal{L}_{\tau_i}$ ▷ Meta update overall parameter set
16: **end while**

G Details on many-way multi-shot scenarios

A detailed version of the results used for the visualization of different 5-way K-shot and N-way 5-shot scenarios depicted in the main paper are presented in Table A6, including the 95% confidence intervals. While enforcing a well-conditioned parameter space for the inner-loop optimization leads to significantly better first-step adaptation results, it can also be observed that the conditioning seems to additionally improve the overall results achieved after 5 updates. The results further indicate that the adaptation of the baseline parameters during the initial steps (mainly 1-3) differs dependent on the number of shots, and seems to be increasingly delayed to the last steps for settings with a higher number of shots (e.g., 42.10% vs. 21.14% vs. 20.32% after 3 updates for $k = 10$, $k = 15$ and $k = 20$, respectively).

H Constraining parameter subsets

As discussed in the main paper, we choose to apply our proposed conditioning constraint only to a subset of the network's parameters to increase efficiency and scalability. We demonstrated that the development of the condition number calculated with respect to only the parameters of the classifier is representative for the condition number calculated with respect to the full set of network parameters. In this section, we provide the visualisations of the development of all evaluated subsets. It is to be noted that for all depicted results, the condition constraint is enforced to the parameter subset denoted in the respective legend. As can be observed in Figure A1, all subsets except for the batchnorm of the embedding layer '*eBN*' demonstrate a development of the condition number that is very similar to the one of the condition number *w.r.t.* the full parameter set. For completeness, we additionally provide the development of the condition number *w.r.t.* the full parameter set if the model is trained *without* our proposed conditioning loss in Figure A1 (h) (i.e., conventional MAML baseline like proposed by Finn et al. (2017)) – demonstrating the significantly higher condition number and thus

Table A6: **Many-way multi-shot experiments.** Average test accuracy for various 5-way K-shot and N-way 5-shot scenarios evaluated on the *mini*ImageNet (Vinyals et al., 2016) test set using a Conv6 architecture.

	Setting	Method	step 1↑	step 2↑	step 3↑	step 4↑	step 5↑
5-way	1-shot	MAML	21.40±0.23	30.85±0.66	45.91±0.87	50.30±0.88	51.22±0.88
		ours	50.92±0.85	52.98±0.89	53.18±0.89	53.28±0.89	53.34±0.89
	5-shot	MAML	20.19±0.07	24.17±0.39	57.25±0.72	64.24±0.73	65.96±0.71
		ours	62.31±0.72	66.66±0.71	67.63±0.72	68.21±0.71	68.43±0.71
	10-shot	MAML	20.00±0.00	20.07±0.04	42.10±0.70	65.48±0.74	70.66±0.67
		ours	64.82±0.72	70.35±0.70	71.65±0.67	72.68±0.68	73.13±0.67
	15-shot	MAML	20.00±0.00	20.00±0.00	21.14±0.18	68.00±0.67	71.22±0.63
		ours	63.91±0.69	70.18±0.65	72.30±0.65	73.58±0.63	74.01±0.63
	20-shot	MAML	20.00±0.00	20.00±0.00	20.32±0.08	69.41±0.65	72.98±0.61
		ours	64.83±0.66	71.56±0.63	73.56±0.60	74.85±0.60	75.55±0.58
5-shot	5-way	MAML	20.19±0.07	24.17±0.39	57.25±0.72	64.24±0.73	65.96±0.71
		ours	62.31±0.72	66.66±0.71	67.63±0.72	68.21±0.71	68.43±0.71
	10-way	MAML	10.00±0.00	14.11±0.30	41.50±0.43	47.38±0.47	49.59±0.46
		ours	42.20±0.42	48.94±0.45	50.79±0.46	51.67±0.46	52.02±0.47
	15-way	MAML	6.67±0.00	7.66±0.27	35.32±0.32	38.29±0.31	41.08±0.31
		ours	32.55±0.30	39.81±0.32	41.83±0.32	42.84±0.32	43.30±0.21

worse-conditioned parameter space that is learned by the unconstrained method. In stark contrast, it can further be observed that the trajectories of the methods actively enforcing conditioning are very close for all subsets where the parameters of the classifier '*cls*' are involved in the conditioning constraint, and that the condition numbers of the actual network ('*all*') is particularly low for all these cases, justifying our choice of using the '*cls*' subset throughout all major experiments in the main paper.

I Condition number and few-step performance

As discussed in the main paper, the development of the condition number and the validation accuracy are directly related. While we presented the validation accuracies for a Conv4 and Conv6 architecture together with the condition number of inner-loop update step 1 in the main paper, we herein show the detailed development of all five inner-loop update steps. The corresponding visualisations of the classification accuracy achieved on the validation set during training are presented in Figure A2 for a Conv4 and Conv6 architecture trained *without* ('MAML') and *with* ('ours') the proposed conditioning constraint enforced via $\mathcal{L}_\kappa$. We further show the development of the condition number with respect to the parameters of the classifier using the support sets of the training data (left column, $\kappa(\boldsymbol{\theta}^{(k)}_{\text{train}})$) and the validation data (right column, $\kappa(\boldsymbol{\theta}^{(k)}_{\text{valid}})$) in Figure A3 for steps $k = 0$ up to $k = 4$, i.e., all parameter sets that will be updated during the course of the 5 inner-loop update steps. Note that the condition property of the initial parameter space at step 0 is important to perform the first inner-loop update (step 1), which is why we investigate the condition numbers of the parameter sets before each update (i.e., sets at stages 0 - 4 for update steps 1 - 5). Both architectures have been trained on the *tiered*ImageNet dataset (Ren et al., 2018).

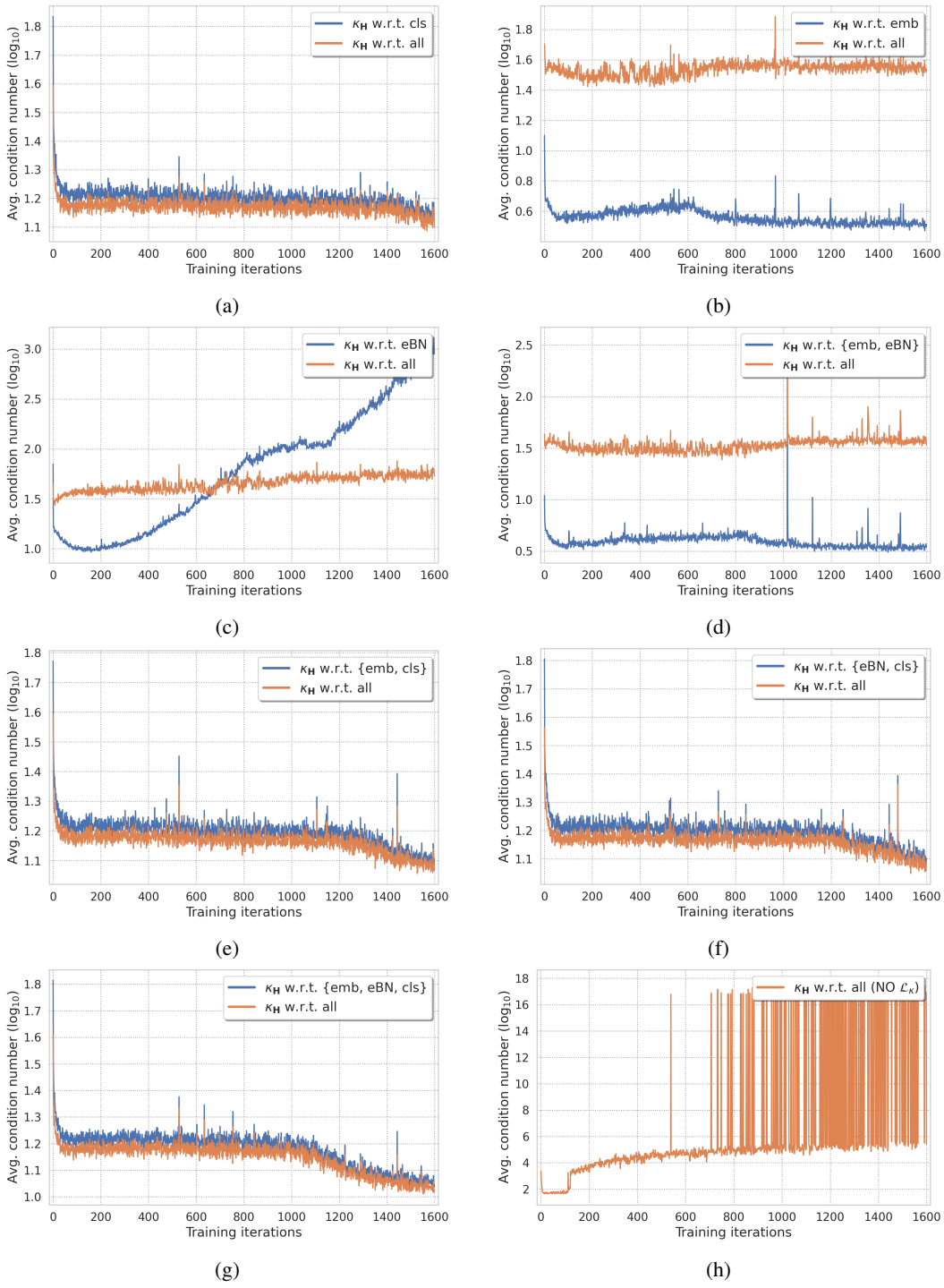
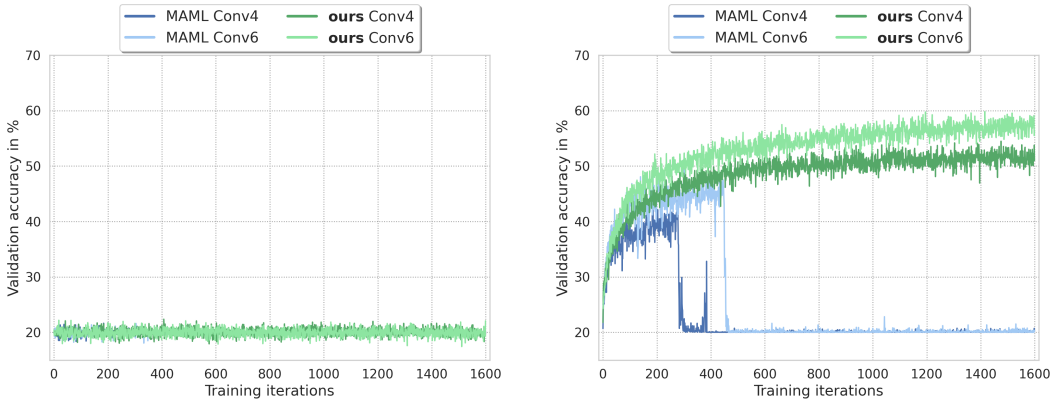
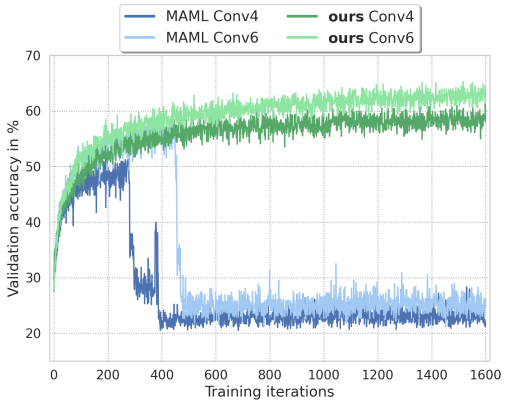


Figure A1: **Constraining a reduced parameter set.** Condition number with respect to the reduced parameter subset denoted in the respective legend, and to all parameters of the model over 1600 iterations on the *mini*ImageNet dataset with a Conv4 architecture for a 5-way 5-shot scenario. Models in (a) - (g) are trained with $\mathcal{L}_{\kappa}$ *w.r.t.* the respective subset, while (h) shows the development for the model trained without the use of the proposed $\mathcal{L}_{\kappa}$.

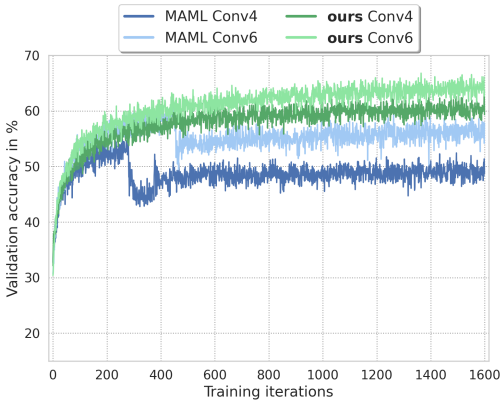


(a) Step 0: $\boldsymbol{\theta}^{(0)}$

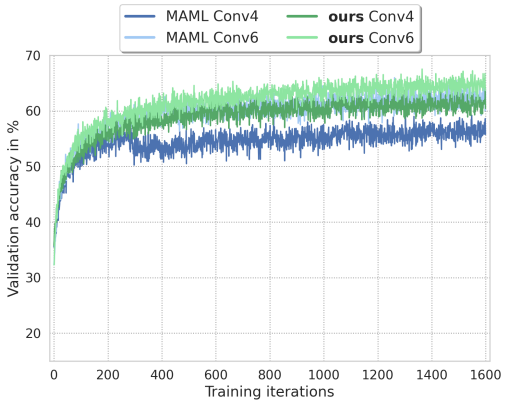
(b) Step 1: $\boldsymbol{\theta}^{(1)}$



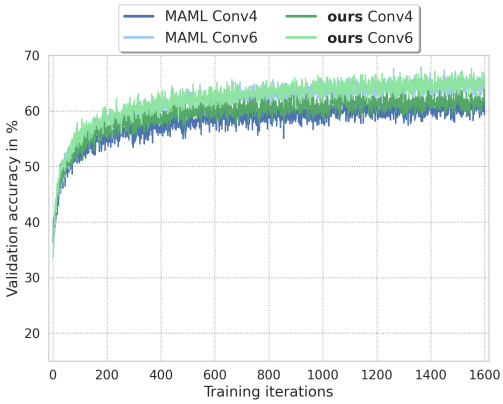
(c) Step 2: $\boldsymbol{\theta}^{(2)}$



(d) Step 3: $\boldsymbol{\theta}^{(3)}$

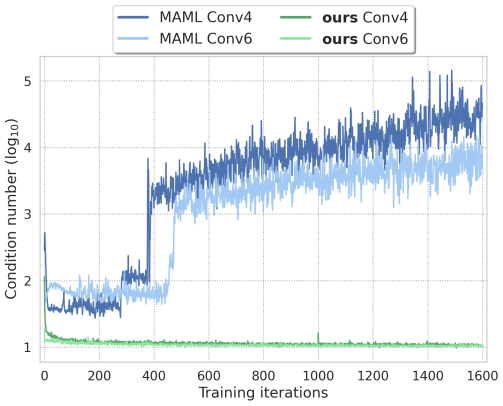


(e) Step 4: $\boldsymbol{\theta}^{(4)}$

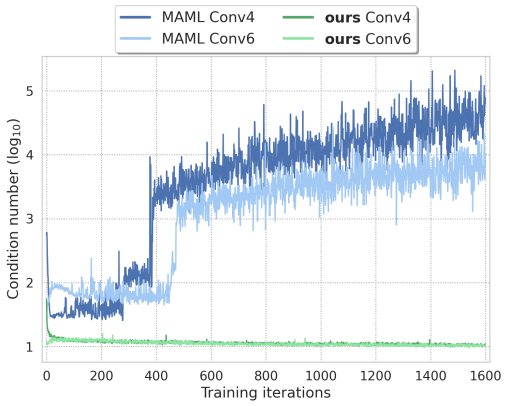


(f) Step 5: $\boldsymbol{\theta}^{(5)}$

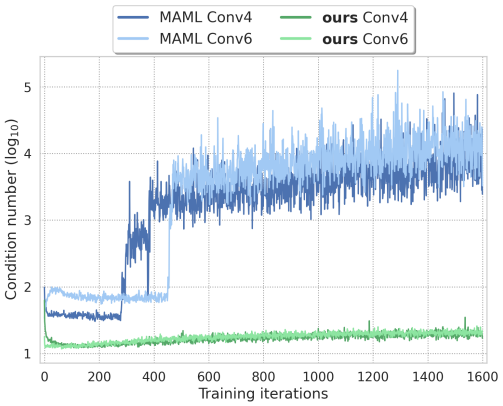
Figure A2: **Validation accuracy over inner-loop update steps.** Reported results obtained by training the baseline *without* ('MAML') and *with* our proposed conditioning constraint ('ours') with respect to the parameters of the model's classifier. Training has been conducted over 1600 iterations in a 5-way 5-shot scenario on the *tiered*ImageNet dataset with a Conv4 and Conv6 architecture.



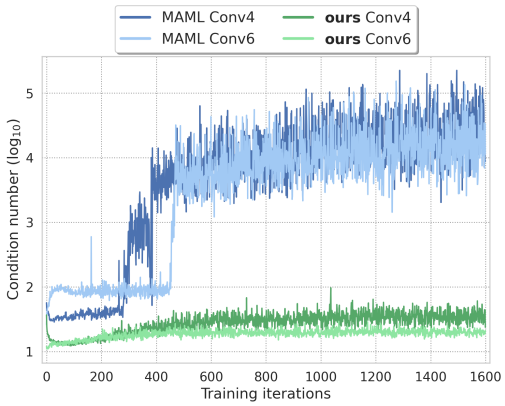
(a) Step 0: $\kappa\left(\boldsymbol{\theta}_{\text{train}}^{(0)}\right)$



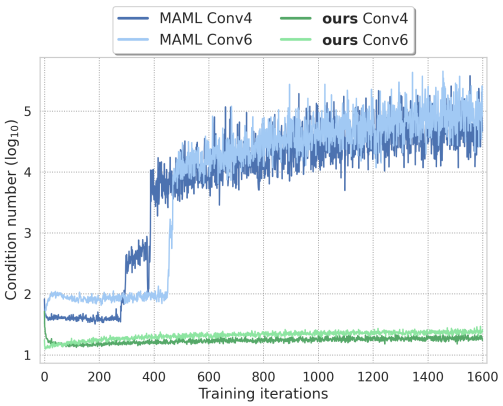
(b) Step 0: $\kappa\left(\boldsymbol{\theta}_{\text{valid}}^{(0)}\right)$



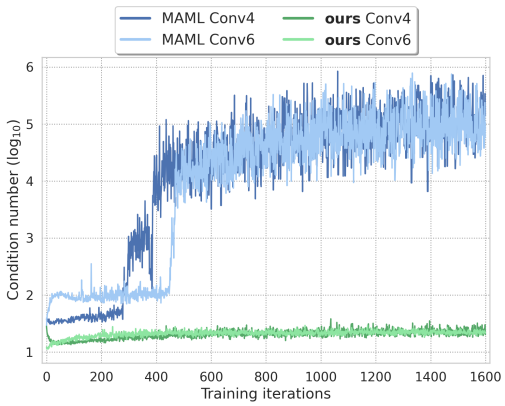
(c) Step 1: $\kappa\left(\boldsymbol{\theta}_{\text{train}}^{(1)}\right)$



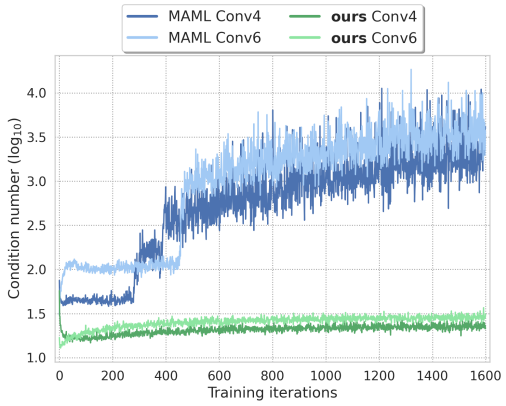
(d) Step 1: $\kappa\left(\boldsymbol{\theta}_{\text{valid}}^{(1)}\right)$



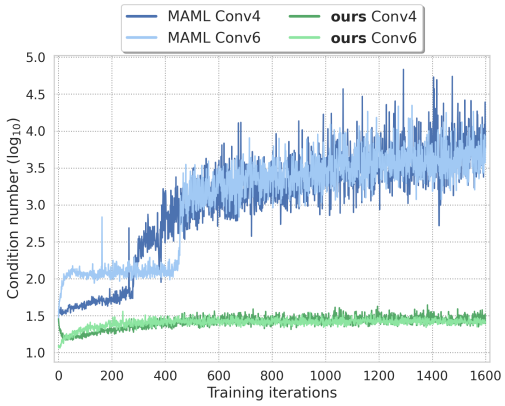
(e) Step 2: $\kappa\left(\boldsymbol{\theta}_{\text{train}}^{(2)}\right)$



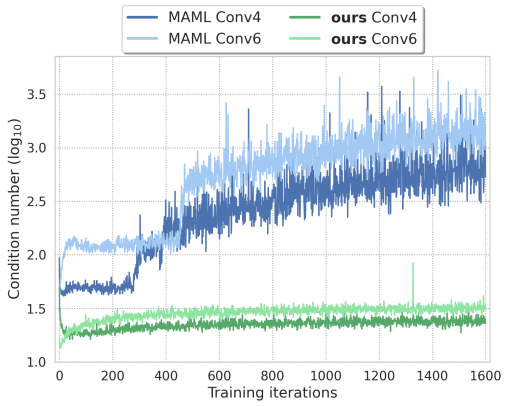
(f) Step 2: $\kappa\left(\boldsymbol{\theta}_{\text{valid}}^{(2)}\right)$



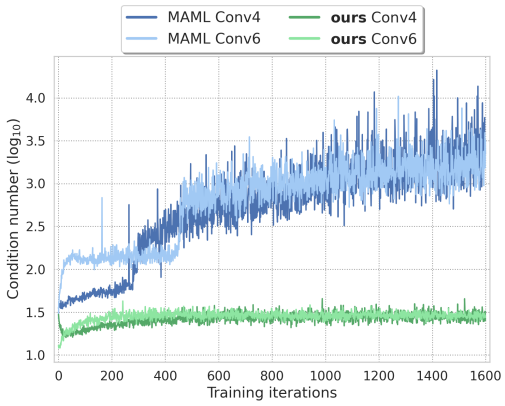
(g) Step 3: $\kappa\left(\boldsymbol{\theta}^{(3)}_{\text{train}}\right)$



(h) Step 3: $\kappa\left(\boldsymbol{\theta}^{(3)}_{\text{valid}}\right)$



(i) Step 4: $\kappa\left(\boldsymbol{\theta}^{(4)}_{\text{train}}\right)$



(j) Step 4: $\kappa\left(\boldsymbol{\theta}^{(4)}_{\text{valid}}\right)$

Figure A3: **Condition numbers over inner-loop update steps.** Reported results were obtained by training the baseline *without* ('MAML') and *with* our proposed conditioning constraint ('ours') with respect to the parameters of the model's classifier. Training has been conducted over 1600 iterations in a 5-way 5-shot scenario on the *tiered*ImageNet dataset with a Conv4 and Conv6 architecture. For each update step, we report the condition number computed via either the support set of the training data $\kappa(\boldsymbol{\theta}^{(k)}_{\text{train}})$ or validation data $\kappa(\boldsymbol{\theta}^{(k)}_{\text{valid}})$.

References

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C., and Huang, J.-B. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 2017.

Flennerhag, S., Rusu, A. A., Pascanu, R., Visin, F., Yin, H., and Hadsell, R. Meta-learning with warped gradient descent. In *International Conference on Learning Representations*, 2019.

Hilliard, N., Phillips, L., Howland, S., Yankov, A., Corley, C. D., and Hodas, N. O. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Oreshkin, B., Rodríguez López, P., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Park, E. and Oliva, J. B. Meta-curvature. *Advances in Neural Information Processing Systems*, 32, 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 2015.

Simon, C., Koniusz, P., Nock, R., and Harandi, M. On modulating the gradient for meta-learning. In *European Conference on Computer Vision*. Springer, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.