Rebuttal - Meta-Curvature (Paper ID: 1842)

To all reviewers:

We performed more hyperparameter search, e.g. dropout rates, learning rates, the number of hidden units, etc., and the results had shown that the proposed method outperformed LEO [2] and the recent state-of-the-art results from [3]. For fair comparison to [3] that used extensive data augmentations, we also reported the results with the multiview features provided by LEO [2], where features were averaged over representations of 4 corner and central crops and their horizontal mirrored versions. We did not make any algorithmic changes and we are hoping to update the results with the corresponding hyperparameters in the final version. We promise to release the code and trained models in order to encourage reproducibility.

Table 1: The results on miniImagenet and tieredImagenet with WRN-28-10 features. ‡ indicates that both meta-train and meta-validation are used during meta-training. † denotes indicates that 15-shot meta-training was used for both 1-shot and 5-shot testing. MetaOptNet [3] used ResNet-12 backbone and trained end-to-end manner while we used the fixed features provided by [2].

	miniImagenet		tieredImagenet	
	1-shot	5-shot	1-shot	5-shot
[1]‡	59.60 ± 0.41	73.74 ± 0.19	·	·
LEO (center)‡ [2]	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
LEO (multiview)‡ [2]	63.97 ± 0.20	79.49 ± 0.70	·	·
MetaOptNet-SVM‡† [3]	64.09 ± 0.62	80.00 ± 0.45	65.81 ± 0.74	81.75 ± 0.53
Meta-SGD (center)	56.58 ± 0.21	68.84 ± 0.19	59.75 ± 0.25	69.04 ± 0.22
MC2 (center)	61.22 ± 0.10	75.92 ± 0.17	66.20 ± 0.10	82.21 ± 0.08
MC2 (center)‡	$\mathbf{61.85 \pm 0.10}$	77.02 ± 0.11	$\mathbf{67.21 \pm 0.10}$	$\mathbf{82.61 \pm 0.08}$
MC2 (multiview)‡	$\mathbf{64.40 \pm 0.10}$	$\mathbf{80.21 \pm 0.10}$	·	·

To R1:

Thanks for the reference, we will cite it with the discussion. In short, the big difference is that their matrix to transform the gradient is a simple binary mask whose rows are either 0 or 1 vector. With the updated experimental results, we hope we resolve your concerns about the performance of WRN experiments.

To R2:

Eq (10): Given a new task, it does not directly follow the gradients of training loss, which might lead the model to overfit (or underfit). Instead, it finds the most similar tasks in the meta-training set and follows the gradients of validation losses in those similar tasks.

To R3:

For clarity question 1: The second order optimization methods are mainly for speeding up the convergence. However, there is no notion of generalization. Faster convergence could mean faster overfitting, which may lose the opportunity to get out of local minima.

2: In convolutional layers, we collapsed height and width into one dimension. The filter size is usually very small (3x3), the second-order matrix (9x9) might not be a big issue. In fully-connected layers, for example, a weight matrix 10x20 needs two curvature matrices, 10x10 and 20x20.

3-4: We do really appreciate your comments about tensor-train decomposition and batch normalization. Both are really interesting aspects. We will leave them as future works.

5: We agree with your point. Here, 'test' set is probably better choice than 'validation' set.

References

[1] Siyuan Qiao, Chenxi Liu, Wei Shen, Alan Yuille, Few-Shot Image Recognition by Predicting Parameters from Activations, In *CVPR* 2018.

[2] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, Raia Hadsell, Meta-Learning with Latent Embedding Optimization, In *ICLR* 2019.

[3] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, Stefano Soatto, Meta-Learning with Differentiable Convex Optimization, In *CVPR* 2019.