# Meta-Curvature

Eunbyung Park
Junier B. Oliva
Department of Computer Science
University of North Carolina at Chapel Hill
UNC DEPARTMENT OF COMPUTER SCIENCE

## Second-order Optimization in MAML

- ❑ Model-Agnostic Meta-Learning (MAML) [1]

$$\underset{\theta}{\text{argmin}}\, \mathbb{E}_{\tau_i}[\mathcal{L}^{\tau_i}_{\text{val}}(\theta - \alpha\nabla\mathcal{L}^{\tau_i}_{\text{tr}}(\theta))]$$

- ❑ Meta-SGD [2]

$$\underset{\theta,\alpha}{\text{argmin}}\, \mathbb{E}_{\tau_i}[\mathcal{L}^{\tau_i}_{\text{val}}(\theta - \alpha\nabla\mathcal{L}^{\tau_i}_{\text{tr}}(\theta))]$$

- ❑ Meta-Curvature

$$\underset{\theta,\mathbf{M}}{\text{argmin}}\, \mathbb{E}_{\tau_i}[\mathcal{L}^{\tau_i}_{\text{val}}(\theta - \mathbf{M}\nabla\mathcal{L}^{\tau_i}_{\text{tr}}(\theta))]$$

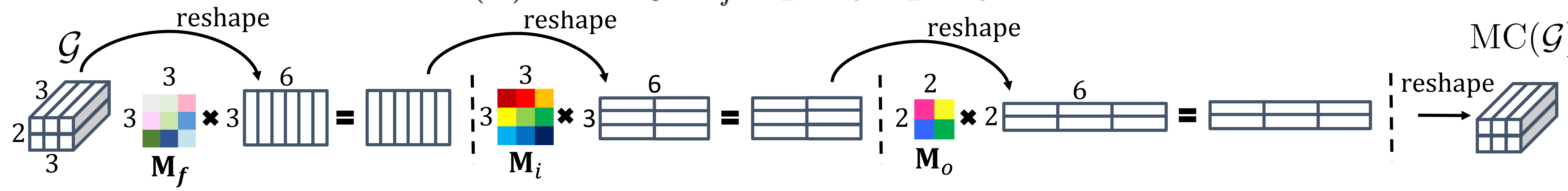
Learning a curvature matrix for better generalization and fast model adaptation in MAML framework.

Modeling the second-order dependencies $\Longrightarrow$ Second-order optimization in the inner optimization

Reducing space and computational complexity $\Longrightarrow$ Decomposition of the curvature matrix

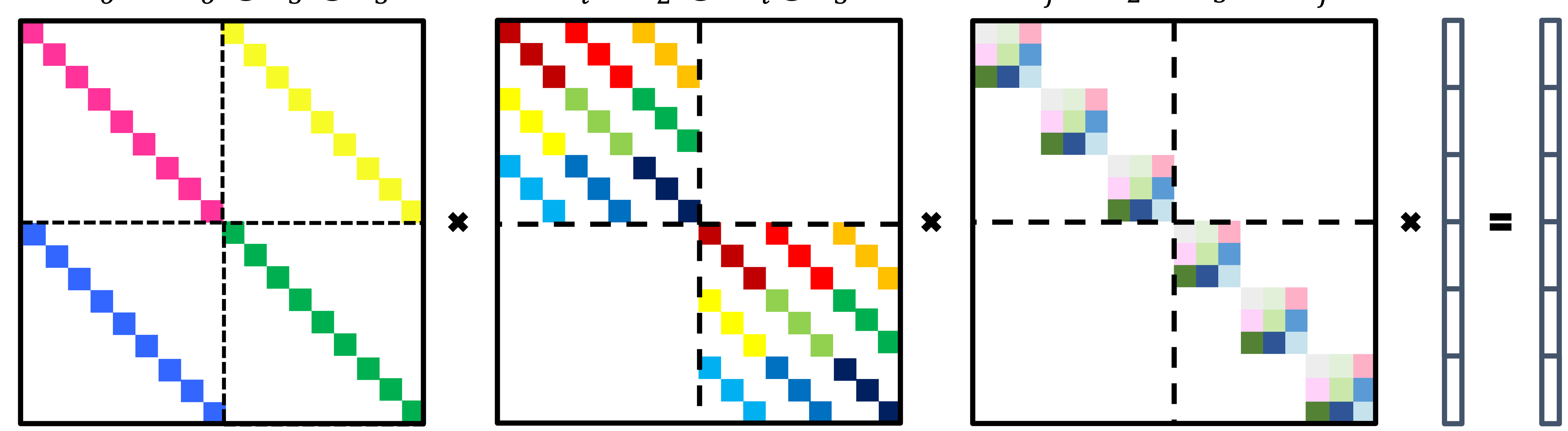
## Meta-Curvature Computation

**Tensor Product View** $\Longrightarrow$ $\text{MC}(\mathcal{G}) = \mathcal{G} \times_3 \mathbf{M}_f \times_2 \mathbf{M}_i \times_1 \mathbf{M}_o$




**Matrix Product View** $\Longrightarrow$ $\text{vec}(\text{MC}(\mathcal{G})) = \widehat{\mathbf{M}_o}\widehat{\mathbf{M}_i}\widehat{\mathbf{M}_f}\text{vec}(\mathcal{G})$

$\widehat{\mathbf{M}_o} = \mathbf{M}_o \otimes \mathbf{I}_3 \otimes \mathbf{I}_3$ $\quad$ $\widehat{\mathbf{M}_i} = \mathbf{I}_2 \otimes \mathbf{M}_i \otimes \mathbf{I}_3$ $\quad$ $\widehat{\mathbf{M}_f} = \mathbf{I}_2 \otimes \mathbf{I}_3 \otimes \mathbf{M}_f$



## Experimental Results

**[ Few-shot classification - Omniglot ]**

| | 5w-1s | 5w-5s | 20w-1s | 20w-5s |
|---|---|---|---|---|
| MAML [1] | 98.7 ± 0.4 | 99.9 ± 0.1 | 95.8 ± 0.3 | 98.9 ± 0.2 |
| Meta-SGD [2] | 99.53 ± 0.26 | 99.93 ± 0.09 | 95.93 ± 0.38 | 98.97 ± 0.19 |
| MAML++¶ [3] | 99.47 | 99.93 | 97.65 | 99.33 |
| MC | **99.77 ± 0.17** | 99.79 ± 0.10 | **97.86 ± 0.26** | **99.24 ± 0.07** |
| MC¶ | **99.97 ± 0.06** | 99.89 ± 0.06 | **99.12 ± 0.16** | **99.65 ± 0.05** |

*5w-1s: 5way 1 shot* ¶ *: 3 model ensemble*

**[ Few-shot classification – Imagenet]**

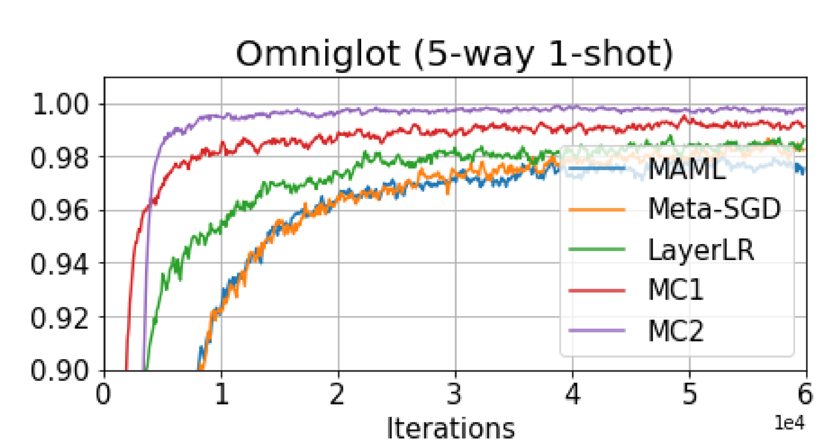
| | mini-Imagenet | | tiered-Imagenet | |
|---|---|---|---|---|
| | 5w-1s | 5w-1s | 5w-5s | 5w-5s |
| LEO – *center* [4] | 61.76 ± 0.08 | 77.59 ± 0.12 | 66.33 ± 0.05 | 81.44 ± 0.09 |
| LEO – *multiview* [4] | 63.97 ± 0.20 | 79.49 ± 0.70 | | |
| MetaOptNet◊ [5] | 64.09 ± 0.62 | 80.00 ± 0.45 | 65.81 ± 0.74 | 81.75 ± 0.53 |
| Meta-SGD [2] | 56.58 ± 0.21 | 68.84 ± 0.19 | 59.75 ± 0.25 | 69.04 ± 0.22 |
| MC – *center* | 61.85 ± 0.10 | 77.02 ± 0.11 | **67.21 ± 0.10** | **82.61 ± 0.08** |
| MC – *multiview* | **64.40 ± 0.10** | **80.21 ± 0.10** | | |

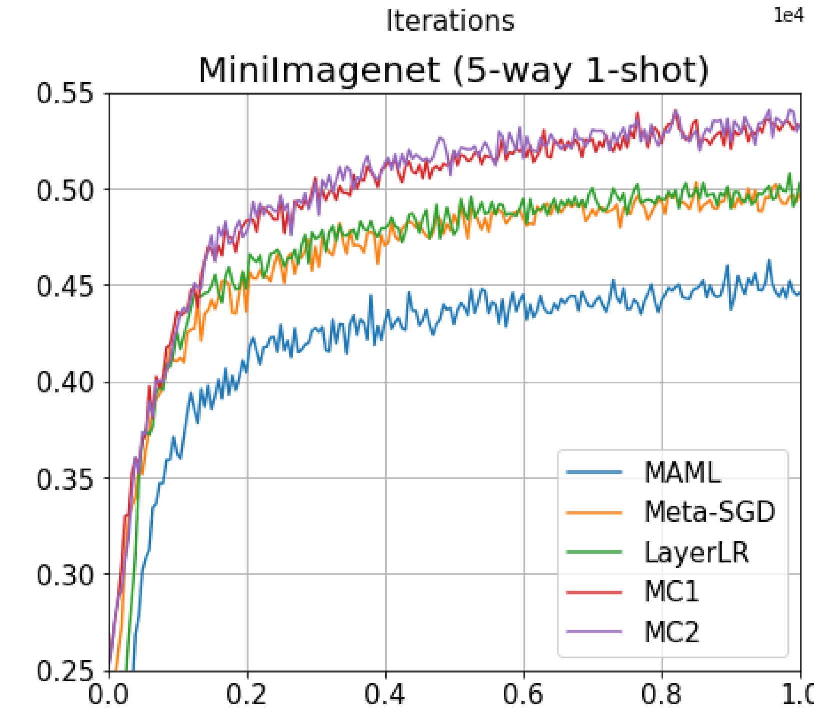
*WRN-28-10 pretrained features + MLP* ◊ *: ResNet-12 and 15-shot meta-training*
*multiview:* features averaged over four corners, central crops, and horizontal mirrored

**[ Few-shot regression]**

| | 5-shot | 10-shot |
|---|---|---|
| MAML [1] | 0.69 ± 0.07 | 0.44 ± 0.04 |
| Meta-SGD [2] | 0.48 ± 0.06 | 0.26 ± 0.03 |
| MC | **0.41 ± 0.05** | **0.20 ± 0.02** |

*Sinusoidal : amp [0.1, 5.0], phase [0, π]*




## Analysis

Meta-gradient w.r.t M : $\theta^{\tau_i}(\mathbf{M}) = \theta - \alpha\mathbf{M}\nabla_\theta\mathcal{L}^{\tau_i}_{\text{tr}}(\theta)$

$$\nabla_{\mathbf{M}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i}(\mathbf{M})) = -\alpha\nabla_{\theta^{\tau_i}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i})\nabla_\theta\mathcal{L}^{\tau_i}_{\text{tr}}(\theta)^\top$$

Given a fixed point $\theta$ and a meta training set $T = \{\tau_i\}$, gradient descents from an initial $\mathbf{M_0}$ :

$$\mathbf{M}_T = \mathbf{M}_0 - \beta\sum_{i=1}^{|\mathcal{T}|}\nabla_{\mathbf{M}_{i-1}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i}) = \mathbf{M}_0 + \alpha\beta\sum_{i=1}^{|\mathcal{T}|}\nabla_{\theta^{\tau_i}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i})\nabla_\theta\mathcal{L}^{\tau_i}_{\text{tr}}(\theta)^\top$$

Applying $M_T$ to the gradients of a new task $\tau_{new}$ :

$$\begin{aligned}\mathbf{M}_T\nabla_\theta\mathcal{L}^{\tau_{\text{new}}}_{\text{tr}}(\theta) &= \Big(\mathbf{M}_0 + \alpha\beta\sum_{i=1}^{|\mathcal{T}|}\nabla_{\theta^{\tau_i}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i})\nabla_\theta\mathcal{L}^{\tau_i}_{\text{tr}}(\theta)^\top\Big)\nabla_\theta\mathcal{L}^{\tau_{\text{new}}}_{\text{tr}}(\theta) \\ &= \mathbf{M}_0\nabla_\theta\mathcal{L}^{\tau_{\text{new}}}_{\text{tr}}(\theta) + \beta\sum_{i=1}^{|\mathcal{T}|}\big(\nabla_\theta\mathcal{L}^{\tau_i}_{\text{tr}}(\theta)^\top\nabla_\theta\mathcal{L}^{\tau_{\text{new}}}_{\text{tr}}(\theta)\big)\alpha\nabla_{\theta^{\tau_i}}\mathcal{L}^{\tau_i}_{\text{val}}(\theta^{\tau_i}) \\ &= \mathbf{M}_0\nabla_\theta\mathcal{L}^{\tau_{new}}_{tr}(\theta) + \beta\sum_{i=1}^{|\mathcal{T}|}\underbrace{\big(\nabla_\theta\mathcal{L}^{\tau_i}_{tr}(\theta)^\top\nabla_\theta\mathcal{L}^{\tau_{new}}_{tr}(\theta)\big)}_{\text{Gradient similarity}}\underbrace{\big(\alpha\nabla_\theta\mathcal{L}^{\tau_i}_{val}(\theta) + \mathcal{O}(\alpha^2)\big)}_{\text{Taylor expansion}}.\end{aligned}$$

### References

[1] Meta-SGD: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, Finn et al, ICML 2017
[2] Meta-SGD: Learning to Learn Quickly for Few-Shot Learning, Li et al, arXiv 2017
[3] How to train your MAML, Antoniou et al, ICLR 2019
[4] Meta-Learning with Latent Embedding Optimization, Rusu et al, ICLR 2019
[5] Meta-Learning with Differentiable Convex Optimization, Lee et al, CVPR 2019