

# EMO: EPISODIC MEMORY OPTIMIZATION FOR FEW-SHOT META-LEARNING

**Anonymous authors**

Paper under double-blind review

## A IMPLEMENTATION DETAILS

We follow (Finn et al., 2017; Yao et al., 2019) by adopting the standard four-block convolutional layers as the feature extractor for our episodic memory optimizer and all baselines. We also conduct our experiments by ANIL (Raghu et al., 2019), which removes the inner-loop updates for the feature extractor network, and applies inner-loop adaptation only to the classifier during training and testing. For all experiments, we keep the outer-loop optimizer consistent with the traditional optimization-based meta-learning approaches, e.g., Adam (Kingma & Ba, 2015). Our code will be publicly released.

## B PROOF OF CONVERGENCE

To analyze the convergence rate of the model, we first derive the upper bound for the expectation  $\mathbb{E} \|\Delta\theta_{t+1}\|^2$  with respect to the independent random noises for all previous gradients  $\{\epsilon_j\}_{j=1}^t$ , where  $\|\cdot\|$  is the spectral norm. We reformulate the aggregation process of our method as a linear multi-step system. Thus the gradient for the  $t$ -th iteration is  $\text{aggr}(\mathbf{g}_t, \mathcal{V}_t) = \sum_{s=0}^{S-1} w_{t,s} \mathbf{g}_{t-s}$ , where  $S$  is the number of step in the system. By incorporating the aggregation process into the update rule Eq. (??) and subtracting  $\theta^*$  from both sides, we obtain the recursive formulation about the difference  $\Delta\theta_t$  as:

$$\Delta\theta_{t+1} = \Delta\theta_t - \alpha \sum_{s=0}^{S-1} w_{t,s} \mathbf{g}_{t-s}. \quad (1)$$

In the paper, the gradient of each iteration is reformulated by adding its mean and the corresponding noise in Eq. (??). For clarity in the proof below, we define the average rate of the gradient changes from the  $t$ -th iteration of model parameters to the optimal as:

$$\mathcal{R}_t = \frac{\nabla f(\theta_t) - \nabla f(\theta^*)}{\Delta\theta_t} = \int_0^1 \nabla^2 f(\theta^* + u\Delta\theta_t) du. \quad (2)$$

With the assumptions about the objective function  $f$ , the average rate of gradient changes is also bounded between  $\mu$  and  $L$ . By incorporating Eq. (2) into Eq. (??), we simplify the recursive formulation about the difference  $\Delta\theta_t$  as:

$$\Delta\theta_{t+1} = \Delta\theta_t - \alpha \sum_{s=0}^{S-1} w_{t,s} \mathcal{R}_{t-s} \Delta\theta_{t-s} - \alpha \sum_{s=0}^{S-1} w_{t,s} \epsilon_{t-s}. \quad (3)$$

We take recursive formulations about  $\{\Delta\theta_{t+1-s}\}_{s=0}^{S-1}$  together and get the matrix version of the recursion below:

$$\begin{aligned}
\begin{bmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \\ \vdots \\ \Delta\theta_{t-S} \end{bmatrix} &= A_t \begin{bmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \\ \vdots \\ \Delta\theta_{t-S+1} \end{bmatrix} + \begin{bmatrix} -\alpha \sum_{s=0}^{S-1} w_{t,s} \epsilon_{t-s} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \\
\text{where } A_t &= \begin{bmatrix} I - \alpha w_{t,0} \mathcal{R}_t & -\alpha w_{t,1} \mathcal{R}_{t-1} & \cdots & -\alpha w_{t,S-2} \mathcal{R}_{t-S+2} & -\alpha w_{t,S-1} \mathcal{R}_{t-S+1} \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}. \tag{4}
\end{aligned}$$

Note that  $A_t$  is the system matrix at the  $t$ -th iteration. By unrolling the recursion below, the upper bound of the expectation  $\mathbb{E} \|\Delta\theta_{t+1}\|^2$  can be derived :

$$\begin{aligned}
\mathbb{E} \|\Delta\theta_{t+1}\|^2 &\leq \mathbb{E}_{\epsilon_t, \dots, \epsilon_1} \left\| \begin{bmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \\ \vdots \\ \Delta\theta_{t-S} \end{bmatrix} \right\|^2 \\
&= \mathbb{E}_{\epsilon_t, \dots, \epsilon_1} \left\| A_t \begin{bmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \\ \vdots \\ \Delta\theta_{t-S+1} \end{bmatrix} + \begin{bmatrix} -\alpha \sum_{s=0}^{S-1} w_{t,s} \epsilon_{t-s} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|^2 \\
&= \|A_t\|^2 \mathbb{E}_{\epsilon_{t-1}, \dots, \epsilon_1} \left\| \begin{bmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \\ \vdots \\ \Delta\theta_{t-S+1} \end{bmatrix} \right\|^2 + \alpha^2 \sum_{s=0}^S w_{t,s}^2 \mathbb{E}_{\epsilon_{t-s}} \|\epsilon_{t-s}\|^2 \tag{5} \\
&\leq \|A_t\|^2 \mathbb{E}_{\epsilon_{t-1}, \dots, \epsilon_1} \left\| \begin{bmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \\ \vdots \\ \Delta\theta_{t-S+1} \end{bmatrix} \right\|^2 + \alpha^2 S \sigma^2 \\
&\dots \\
&\leq \prod_{j=1}^t \|A_j\|^2 \|\Delta\theta_1\|^2 + \alpha^2 S \sigma^2 \sum_{j=1}^S (\|A_t\|^2 \cdots \|A_{j+1}\|^2).
\end{aligned}$$

According to the definition of the spectral norm and the properties of block matrix (Polyak, 1964; Assran & Rabbat, 2020; McRae et al., 2022), we get the upper bound of the spectral norm below:

$$\begin{aligned}
\|A_t\| &\leq \lambda_t(\widehat{A}_t^\top \widehat{A}_t), \\
\text{where } \widehat{A}_t &= \begin{bmatrix} 1 - \alpha w_{t,0} \tau_t & -\alpha w_{t,1} \tau_{t-1} & \cdots & -\alpha w_{t,S-2} \tau_{t-S+2} & -\alpha w_{t,S-1} \tau_{t-S+1} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{6}
\end{aligned}$$

Note that  $\lambda_t(\widehat{A}_t^\top \widehat{A}_t)$  is the square root of the largest eigenvalue of the matrix  $\widehat{A}_t^\top \widehat{A}_t$ . The matrix  $\widehat{A}_t \in \mathbb{R}^{S \times S}$  has bounded hyperparameters:  $\tau_t \in [\mu, L]$  and  $w_t \in [0, 1]$ . We introduce  $\lambda_{\max}$  as the upper bound for all  $\lambda_t$  corresponding to all system matrices. Since the learning rate is chosen sufficiently small such that  $\lambda_{\max} < 1$ , we further simplify Eq. (5) below:

$$\mathbb{E} \|\Delta\theta_{t+1}\|^2 \leq \lambda_{\max}^{2t} \|\Delta\theta_1\|^2 + \frac{\alpha^2 \sigma^2 S}{1 - \lambda_{\max}^2}. \tag{7}$$

Recall that  $f(\cdot)$  is assumed to be  $L$ -smooth, we get the convergence rate of our model as

$$f(\theta_{t+1}) - f(\theta^*) \leq \frac{L}{2} (\lambda_{\max}^{2t} \|\Delta\theta_1\|^2 + \frac{\alpha^2 \sigma^2 S}{1 - \lambda_{\max}^2}). \quad (8)$$

## C MORE RESULTS

### C.1 COMPARISON WITH OTHER OPTIMIZERS

Table 1: Comparison with other optimizers on *Meta-Dataset-BTAF* under the 5-way 1-shot setting. EMO achieves better performance compared to other optimizers on all datasets.

Dataset	MAML			
	w/ SGD	w/ Momentum	w/ Adam	w/ EMO
Bird	<b>53.94</b> $\pm$ 1.45	52.98 $\pm$ 1.42	52.55 $\pm$ 1.41	<b>56.32</b> $\pm$ 1.33
Texture	31.66 $\pm$ 1.31	31.38 $\pm$ 1.31	30.95 $\pm$ 1.34	<b>34.75</b> $\pm$ 1.41
Aircraft	<b>51.37</b> $\pm$ 1.38	51.09 $\pm$ 1.35	50.15 $\pm$ 1.33	<b>53.99</b> $\pm$ 1.33
Fungi	<b>42.12</b> $\pm$ 1.36	<b>41.54</b> $\pm$ 1.35	<b>41.04</b> $\pm$ 1.31	<b>43.15</b> $\pm$ 1.36

To show the benefit of the episodic memory optimizer, we compare MAML (Finn et al., 2017), Meta-SGD (Li et al., 2017), and ANIL (Raghu et al., 2019) with their EMO variants, where each variant uses EMO as the inner-loop optimizer. Table 1 shows each method with EMO achieves better performance by a large margin than the original methods on four different datasets. More importantly, the most challenging, which has the largest domain gap Texture, delivers 34.75%, surpassing the Meta-SGD by 2.09%. We attribute the improvements to our model’s ability to leverage the episodic memory to adjust the model parameters, allowing the model to update the test task model using the most training task-like update, and thus leading to improvements over original models.

### C.2 EFFECT OF DIFFERENT AGGREGATION FUNCTIONS

We also give the ANIL with EMO for ablating the effect of EMO’s aggregation function used to compute the new gradients. We report the performance of ANIL with EMO using different aggregation functions in Table 2. The best-suited aggregation function for ANIL with EMO is the Transformer. To ensure consistency of implementation on each dataset, we choose the Transformer aggregation function for ANIL with EMO.

Table 2: Effect of ANIL with different aggregation functions. Mean achieves better performance than alternatives.

Dataset	ANIL with EMO		
	sum	Mean	Transformer
Bird	54.91 $\pm$ 1.33	<b>55.18</b> $\pm$ 1.34	54.78 $\pm$ 1.33
Texture	32.71 $\pm$ 1.30	33.14 $\pm$ 1.40	<b>33.15</b> $\pm$ 1.41
Aircraft	<b>53.16</b> $\pm$ 1.40	52.11 $\pm$ 1.38	52.79 $\pm$ 1.33
Fungi	43.17 $\pm$ 1.34	43.07 $\pm$ 1.31	<b>43.75</b> $\pm$ 1.36

### C.3 EFFECT OF MEMORY CONTROLLER

We further assess the effect of the memory controller with ANIL with EMO and Meta-SGD with EMO in Table 3. With CLOCK-EM, Meta-SGD with EMO achieves better performance on all datasets, while ANIL with EMO leads to a small but consistent gain under all the datasets with LRU-EM. To ensure consistency of implementation on each dataset, we choose the LRU-EM function for ANIL with EMO, and CLOCK-EM is used for Meta-SGD with EMO.

Table 3: Effect of ANIL with different memory controllers. LRU-EM achieves better performance than alternatives.

Dataset	ANIL with EMO		
	FIFO-EM	CLOCK-EM	LRU-EM
Bird	50.11 $\pm$ 1.31	53.91 $\pm$ 1.34	<b>54.78</b> $\pm$ 1.43
Texture	29.11 $\pm$ 1.41	32.94 $\pm$ 1.40	<b>33.15</b> $\pm$ 1.31
Aircraft	47.96 $\pm$ 1.40	<b>53.91</b> $\pm$ 1.35	52.79 $\pm$ 1.33
Fungi	40.97 $\pm$ 1.35	43.17 $\pm$ 1.35	<b>43.75</b> $\pm$ 1.31

Table 4: Effect of Meta-SGD with different memory controllers. LRU-EM achieves better performance than alternatives.

Dataset	Meta-SGD with EMO		
	FIFO-EM	CLOCK-EM	LRU-EM
Bird	53.05 $\pm$ 1.34	<b>58.95</b> $\pm$ 1.41	57.31 $\pm$ 1.34
Texture	32.13 $\pm$ 1.41	<b>36.26</b> $\pm$ 1.33	35.95 $\pm$ 1.41
Aircraft	49.16 $\pm$ 1.41	55.21 $\pm$ 1.35	<b>56.19</b> $\pm$ 1.34
Fungi	41.61 $\pm$ 1.34	<b>45.24</b> $\pm$ 1.35	44.75 $\pm$ 1.36

#### C.4 ANALYSIS OF EPISODIC MEMORY

In this section, we further analysis of our proposed episodic memory with the other three datasets. In this experiment, we meta-train MAML and MAML with EMO on the Texture, Aircraft, and Fungi datasets, respectively, and meta-test on *Meta-Dataset-BTAF*. Therefore the episodes saved in the memory are from the Texture, Aircraft, and Fungi, respectively. The results are shown in Figure 1. Consistent with the results in the Figure ??, MAML with EMO has a significant performance improvement when the meta-training dataset is the same as the meta-test dataset. Interestingly, the memory of Aircraft can also help Bird to achieve better performance in Figure 1 (b). Similarly, when the test task has large distribution shifts with the training task, the memory will not be useful or even harmful.

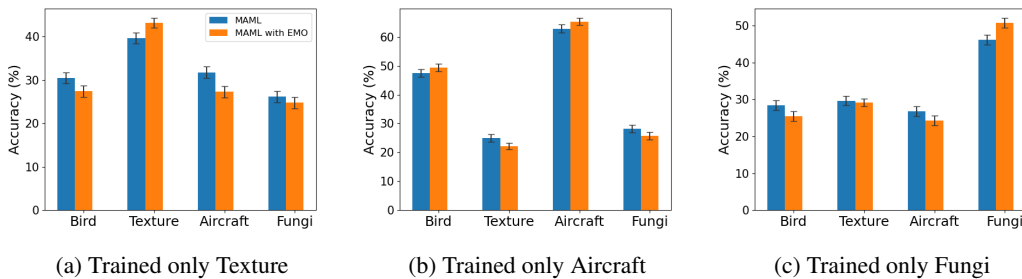


Figure 1: Analysis of episodic memory.

#### C.5 COMPARISON WITH THE STATE-OF-THE-ART ON FEW-SHOT LEARNING DATASETS

We further conduct experiments on the *Meta-Dataset-BTAF* and *miniImageNet* under the 5-way 5-shot setting in Table 5. We also give the comparative results for few-shot learning on *miniImageNet* and *tiredImageNet* using a ResNet-12 back in the Table 6. In these comparison, we apply ARML Yao et al. (2020) with EMO to do the experiment. Our method achieves state-of-the-art performance on all benchmarks under the 5-way 5-shot setting.

Table 5: Comparative results of different algorithms on the *Meta-Dataset-BTAF* using a Conv-4 backbone under the 5-way 5-shot setting. The results of other methods are provided by (Yao et al., 2019; Jiang et al., 2022). Equipping ARML with EMO makes it a consistent top-performer.

Method	Bird	Texture	Aircraft	Fungi	<i>miniImageNet</i>
MAML (Finn et al., 2017)	68.52 $\pm$ 0.79	44.56 $\pm$ 0.68	66.18 $\pm$ 0.71	51.85 $\pm$ 0.85	63.11 $\pm$ 0.92
Meta-SGD (Li et al., 2017)	67.87 $\pm$ 0.74	45.49 $\pm$ 0.68	66.84 $\pm$ 0.70	52.51 $\pm$ 0.81	64.03 $\pm$ 0.94
HSML (Yao et al., 2019)	71.68 $\pm$ 0.73	48.08 $\pm$ 0.69	73.49 $\pm$ 0.68	56.32 $\pm$ 0.80	65.91 $\pm$ 0.95
ARML (Yao et al., 2020)	73.34 $\pm$ 0.70	49.67 $\pm$ 0.67	74.88 $\pm$ 0.64	57.55 $\pm$ 0.82	66.87 $\pm$ 0.93
TSA-MAML (Zhou et al., 2021)	72.31 $\pm$ 0.71	49.50 $\pm$ 0.68	74.01 $\pm$ 0.70	56.95 $\pm$ 0.80	65.52 $\pm$ 0.92
ANIL (Raghu et al., 2019)	70.67 $\pm$ 0.72	44.67 $\pm$ 0.95	66.05 $\pm$ 1.07	52.89 $\pm$ 0.30	61.50 $\pm$ 0.92
BMG (Flennerhag et al., 2021)	71.56 $\pm$ 0.76	49.44 $\pm$ 0.73	66.83 $\pm$ 0.79	52.56 $\pm$ 0.89	66.73 $\pm$ 0.91
MUSML (Jiang et al., 2022)	76.69 $\pm$ 0.72	52.41 $\pm$ 0.75	77.76 $\pm$ 0.82	57.74 $\pm$ 0.81	65.12 $\pm$ 1.48
<b>ARML with EMO</b>	<b>77.17</b> $\pm$ 0.65	<b>53.25</b> $\pm$ 0.68	<b>77.83</b> $\pm$ 0.63	<b>59.15</b> $\pm$ 0.79	<b>71.05</b> $\pm$ 0.91

Table 6: Comparative results for few-shot learning on *miniImagenet* and *tieredImagenet* using a ResNet-12 backbone. ARML with EMO can also improve performance for traditional few-shot learning.

Method	<i>miniImagenet</i> 5-way		<i>tieredImagenet</i> 5-way	
	1-shot	5-shot	1-shot	5-shot
SNAIL (Mishra et al., 2018)	55.71 $\pm$ 0.99	68.88 $\pm$ 0.92	-	-
Dynamic FS (Gidaris & Komodakis, 2018)	55.45 $\pm$ 0.89	70.13 $\pm$ 0.68	-	-
TADAM (Oreshkin et al., 2018)	58.50 $\pm$ 0.30	76.70 $\pm$ 0.30	-	-
MTL (Sun et al., 2019)	61.20 $\pm$ 1.80	75.50 $\pm$ 0.80	-	-
VariationalFSL (Zhang et al., 2019)	61.23 $\pm$ 0.26	77.69 $\pm$ 0.17	-	-
TapNet (Yoon et al., 2019)	61.65 $\pm$ 0.15	76.36 $\pm$ 0.10	63.08 $\pm$ 0.15	80.26 $\pm$ 0.12
MetaOptNet (Lee et al., 2019)	62.64 $\pm$ 0.61	78.63 $\pm$ 0.46	65.81 $\pm$ 0.74	81.75 $\pm$ 0.53
CTM (Li et al., 2019)	62.05 $\pm$ 0.55	78.63 $\pm$ 0.06	64.78 $\pm$ 0.11	81.05 $\pm$ 0.52
CAN (Hou et al., 2020)	63.85 $\pm$ 0.48	79.44 $\pm$ 0.34	69.89 $\pm$ 0.51	84.23 $\pm$ 0.37
HVM (Du et al., 2022)	67.83 $\pm$ 0.57	83.88 $\pm$ 0.51	73.67 $\pm$ 0.71	88.05 $\pm$ 0.44
<b>ARML with EMO</b>	<b>69.15</b> $\pm$ 0.34	<b>84.13</b> $\pm$ 0.25	<b>75.17</b> $\pm$ 0.35	<b>89.05</b> $\pm$ 0.25

## REFERENCES

- Mahmoud Assran and Michael Rabbat. On the convergence of nesterov’s accelerated gradient method in stochastic settings. In *ICML*, 2020.
- Yingjun Du, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Hierarchical variational memory for few-shot learning across domains. In *ICLR*, 2022.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pp. 1126–1135, 2017.
- Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. In *ICLR*, 2021.
- Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pp. 4367–4375, 2018.
- Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, pp. 4005–4016, 2020.
- Weisen Jiang, James Kwok, and Yu Zhang. Subspace learning for effective meta-learning. In *ICML*, pp. 10177–10194. PMLR, 2022.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- Kwonjoon Lee, Subhansu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, pp. 10657–10665, 2019.
- Hongyang Li, David Eigen, Samuel Dodge, Matthew Zeiler, and Xiaogang Wang. Finding task-relevant features for few-shot learning by category traversal. In *CVPR*, pp. 1–10, 2019.

- Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.
- Paul-Aymeric McRae, Prasanna Parthasarathi, Mahmoud Assran, and Sarath Chandar. Memory augmented optimizers for deep learning. In *ICLR*, 2022.
- Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.
- Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, pp. 721–731, 2018.
- Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.
- Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, pp. 403–412, 2019.
- Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *ICML*, pp. 7045–7054. PMLR, 2019.
- Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. *arXiv preprint arXiv:2001.00745*, 2020.
- Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *ICML*, pp. 7115–7123, 2019.
- Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *ICCV*, pp. 1685–1694, 2019.
- Pan Zhou, Yingtian Zou, Xiao-Tong Yuan, Jiashi Feng, Caiming Xiong, and Steven Hoi. Task similarity aware meta learning: Theory-inspired improvement on maml. In *UAI*, pp. 23–33. PMLR, 2021.