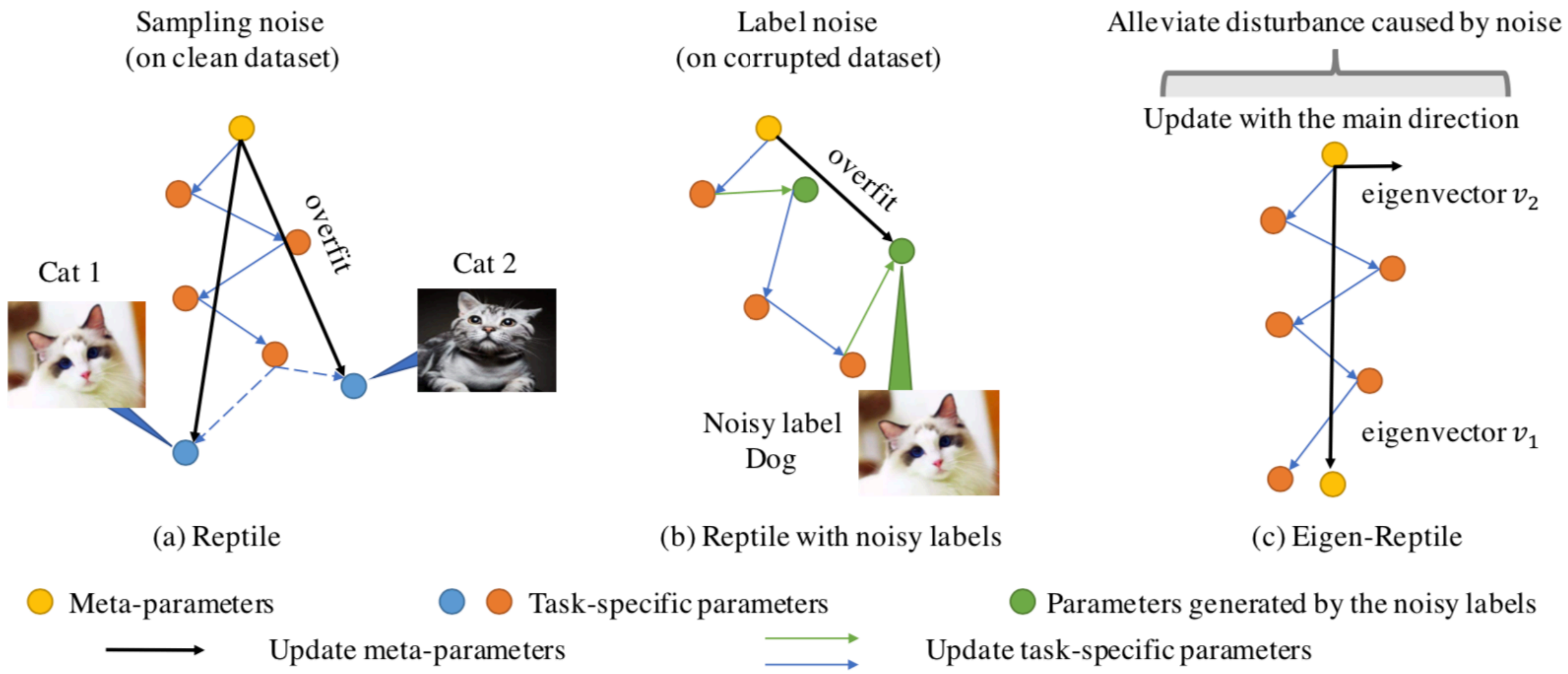
Robust Meta-learning with Sampling Noise and Label Noise via Eigen-Reptile

Dong Chen, Zhejiang University
Lingfei Wu, JD.COM Silicon Valley Research Center
Siliang Tang, Zhejiang University
Xiao Yun, JD.COM Silicon Valley Research Center
Bo Long, JD.COM Silicon Valley Research Center
Yueting Zhuang, Zhejiang University


Sampling noise
(on clean dataset)
overfit
Cat 1
Cat 2
(a) Reptile
Label noise
(on corrupted dataset)
overfit
Noisy label
Dog
(b) Reptile with noisy labels
Alleviate disturbance caused by noise
Update with the main direction
eigenvector v_2
eigenvector v_1
(c) Eigen-Reptile
Meta-parameters
Task-specific parameters
Parameters generated by the noisy labels
Update meta-parameters
Update task-specific parameters

Retain the most information:

$$J(\boldsymbol{e}) = -\boldsymbol{e}^\top \boldsymbol{S}\boldsymbol{e} + \sum_{i=1}^{n} \| \boldsymbol{w_i} - \overline{\boldsymbol{w}} \|^2 \quad (1)$$

Direction of the main direction:

$$\max \boldsymbol{e}^\top \boldsymbol{S}\boldsymbol{e} \qquad \text{s.t.} \begin{cases} \overline{\boldsymbol{V}}\boldsymbol{e} > 0 \\ \boldsymbol{e}^\top \boldsymbol{e} = 1 \end{cases} \quad \text{, where} \qquad \overline{\boldsymbol{V}} = \frac{1}{\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} \boldsymbol{w}_{n-i+1} - \boldsymbol{w}_i \quad (2)$$

Get the objective function

$$g(\mu, \boldsymbol{e}, \lambda, \eta) = \boldsymbol{e}^\top \boldsymbol{S}\boldsymbol{e} - \lambda(\boldsymbol{e}^\top \boldsymbol{e} - 1) + \mu(-\overline{\boldsymbol{V}}\boldsymbol{e} + \eta^2), \quad \text{where} \qquad \lambda \neq 0, \mu \geq 0 \quad (3)$$

The cost of computing e of $Se = \lambda e$ is $O(d^3)$.

For fast computing:

$$\boldsymbol{W}\boldsymbol{W}^\top \underbrace{\boldsymbol{W}\widehat{\boldsymbol{e}}}_{\boldsymbol{e}} = \underbrace{\widehat{\lambda}}_{\lambda} \underbrace{\boldsymbol{W}\widehat{\boldsymbol{e}}}_{\boldsymbol{e}} \qquad (4)$$

When update meta-parameter:

$$\boldsymbol{\phi} \longleftarrow \boldsymbol{\phi} + \beta\nu\zeta\boldsymbol{e}, \quad \text{where} \quad \zeta = \frac{\lambda}{\sum_{m=1}^{n} \lambda_m} \qquad (5)$$

Algorithm 1 Eigen-Reptile

```
Input: Distribution over tasks $P(\mathcal{T})$, outer step size $\beta$.
1:  Initialize meta-parameters $\boldsymbol{\phi}$
2:  while not converged do
3:      $\boldsymbol{W} = [\,], \nu = 0$
4:      Sample batch of tasks $\{\mathcal{T}_i\}_{i=1}^{B} \sim P(\mathcal{T})$
5:      for each task $\mathcal{T}_i$ do
6:          $\boldsymbol{\phi}_i = \boldsymbol{\phi}$
7:          Sample train set $D_{train}$ of $\mathcal{T}_i$
8:          for $j = 1, 2, 3, ..., n$ do
9:              $\boldsymbol{\phi}_i^j = U^j(D_{train}, \boldsymbol{\phi}_i)$
10:             $\boldsymbol{W}$ appends $\boldsymbol{W}_{:,j} = flatten(\boldsymbol{\phi}_i^j)$
11:         end for
12:         Mean centering, $\boldsymbol{W} = \boldsymbol{W} - \overline{\boldsymbol{w}}, \quad \overline{\boldsymbol{w}} \in R^{d\times 1}$
13:         Compute matrix $\widehat{\boldsymbol{\Lambda}}$ and eigenvector matrix $\widehat{\boldsymbol{P}}$ of
            scatter matrix $\boldsymbol{W}^\top \boldsymbol{W}$
14:         Eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ in $\widehat{\boldsymbol{\Lambda}}$
15:         Compute matrix of $\boldsymbol{W}\boldsymbol{W}^\top$, $\boldsymbol{P} = \boldsymbol{W}\widehat{\boldsymbol{P}}$
16:         Let the eigenvector corresponding to $\lambda_1$ be a unit
            vector, $\| \boldsymbol{e}_i^1 \|_2^2 = 1$
17:         for $j = 1, 2, 3, ..., n-1$ do
18:             $\nu = \nu + (\boldsymbol{W}_{:,j+1} - \boldsymbol{W}_{:,j})\boldsymbol{e}_i^1$
19:         end for
20:         $\boldsymbol{e}_i^1 = \frac{\lambda_1}{\sum_{m=1}^{n} \lambda_m} \times \boldsymbol{e}_i^1$
21:         Calculate the approximate direction of task-
            specific gradient update $\overline{\boldsymbol{V}}$:
22:         $\overline{\boldsymbol{V}} = \frac{1}{\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} \boldsymbol{W}_{:,n-i+1} - \boldsymbol{W}_{:,i}$
23:         if $\boldsymbol{e}_i^1 \cdot \overline{\boldsymbol{V}} < 0$ then
24:             $\boldsymbol{e}_i^1 = -\boldsymbol{e}_i^1$
25:         end if
26:     end for
27:     Average the main directions to get
            $\tilde{\boldsymbol{e}} = (1/B) \sum_{i=1}^{B} \boldsymbol{e}_i^1$
28:     Update meta-parameters $\boldsymbol{\phi} \longleftarrow \boldsymbol{\phi} + \beta \times \nu / B \times \tilde{\boldsymbol{e}}$
29: end while
```

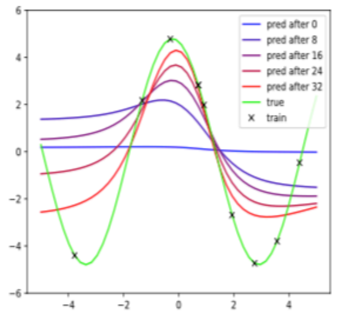
Theorem 1:
Alleviate sampling noise with the main direction.

$$\boldsymbol{C} = \frac{1}{n-1}\boldsymbol{S} = \boldsymbol{C}_t + \boldsymbol{C}_x = \boldsymbol{P}_t(\Lambda_t + \boldsymbol{\Lambda}_x)\boldsymbol{P}_t^\top = \boldsymbol{P}_t(\boldsymbol{\Lambda}_t + \sigma^2\boldsymbol{I})\boldsymbol{P}_t^\top = \boldsymbol{P}_t\Lambda\boldsymbol{P}_t^\top = \boldsymbol{P}\Lambda\boldsymbol{P}^\top$$

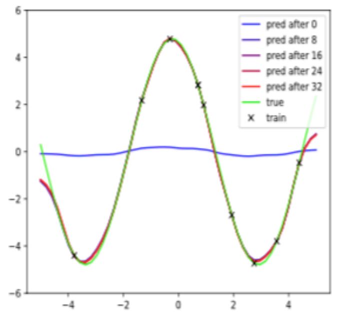
Theorem 2:
More accurate main direction with ISPL.

$$\frac{1}{\lambda}\mathbb{E}(\boldsymbol{C}_{tr})\boldsymbol{e} = \boldsymbol{P}_o(\boldsymbol{I} - \frac{\boldsymbol{\Lambda}_o}{\lambda})\boldsymbol{P}_o^\top\boldsymbol{e} \approx \boldsymbol{P}_o(\boldsymbol{I} - \frac{\lambda_o}{\lambda}\boldsymbol{I})\boldsymbol{P}_o^\top\boldsymbol{e} > \boldsymbol{P}_o(\boldsymbol{I} - \frac{\lambda_o - \xi}{\lambda - \xi}\boldsymbol{I})\boldsymbol{P}_o^\top\boldsymbol{e}$$

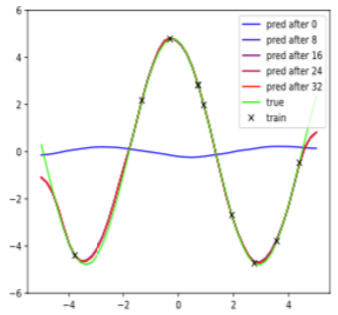
Toy test



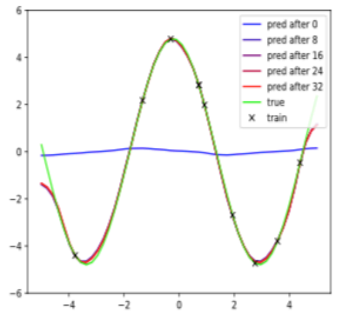
(a) ER of iteration 1



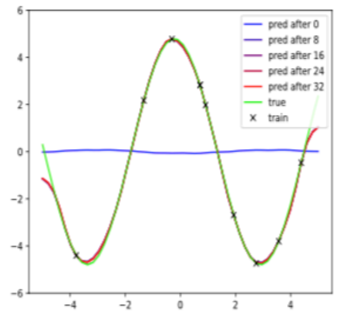
(b) ER of iteration 10000



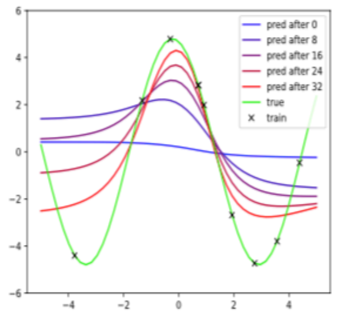
(c) ER of iteration 15000



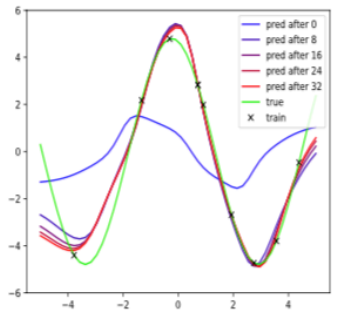
(d) ER of iteration 20000



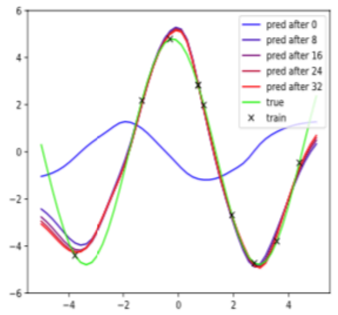
(e) ER of iteration 30000



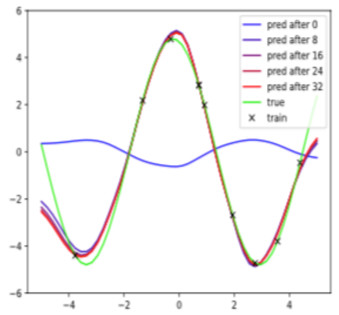
(f) Reptile of iteration 1



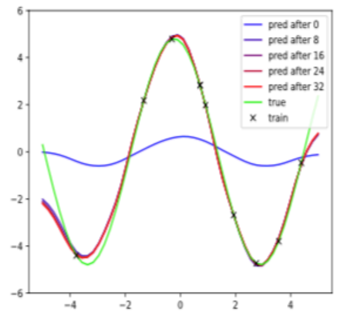
(g) Reptile of iteration 10000



(h) Reptile of iteration 15000



(i) Reptile of iteration 20000



(j) Reptile of iteration 30000

Compare with other directions

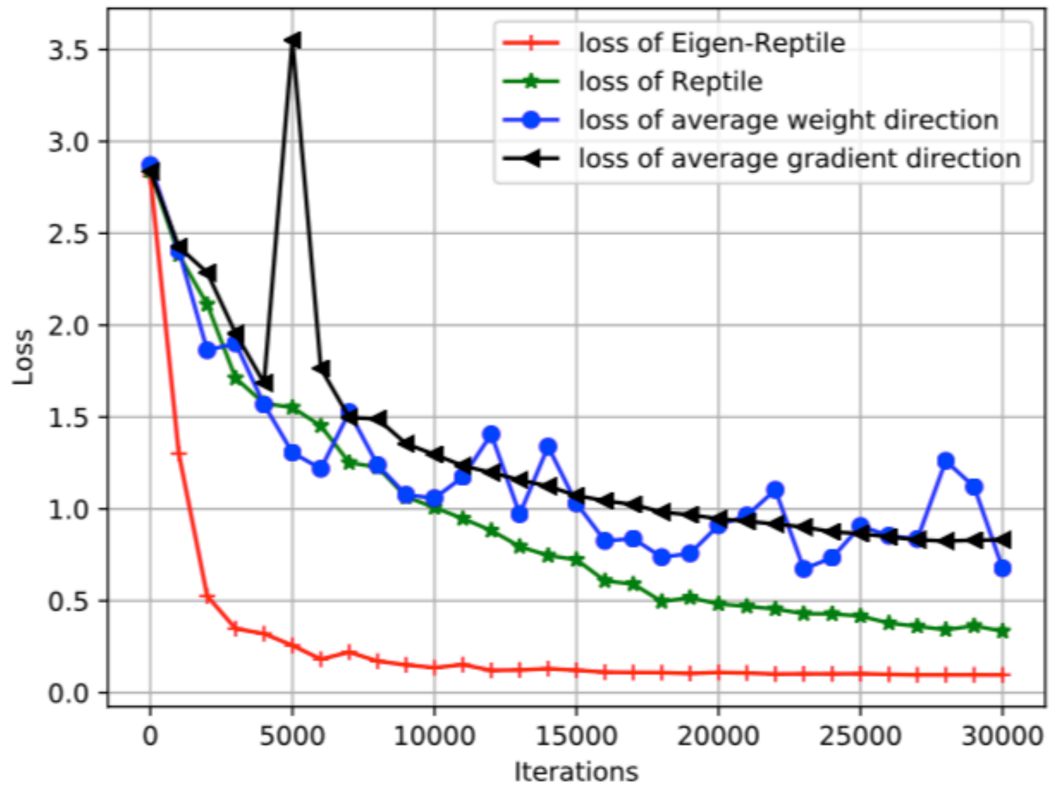
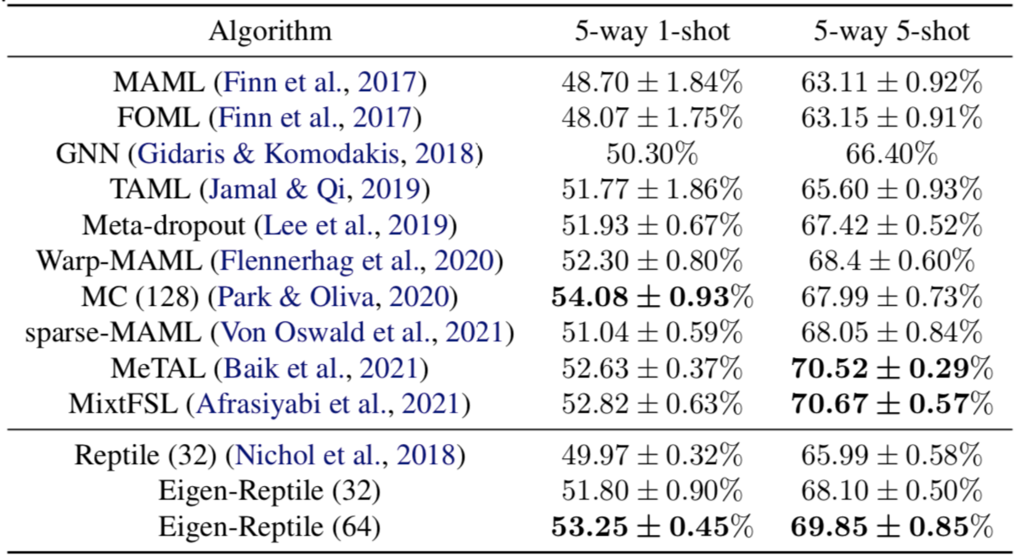


Figure 4. Loss of different update direction.

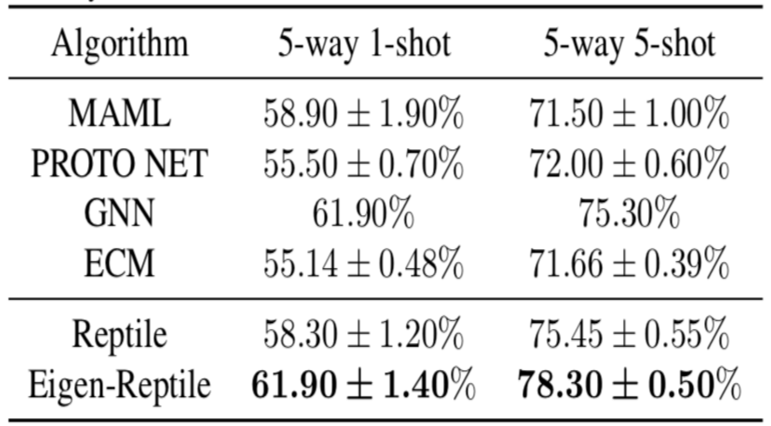
Experiments on clean Mini-Imagenet and CIFAR-FS

Table 1. Accuracy of FSL on Mini-Imagenet N-way K-shot. The ± shows 95% confidence interval over tasks. The number in (·) denotes the number of filters.



Algorithm	5-way 1-shot	5-way 5-shot
MAML (Finn et al., 2017)	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
FOML (Finn et al., 2017)	$48.07 \pm 1.75\%$	$63.15 \pm 0.91\%$
GNN (Gidaris & Komodakis, 2018)	50.30%	66.40%
TAML (Jamal & Qi, 2019)	$51.77 \pm 1.86\%$	$65.60 \pm 0.93\%$
Meta-dropout (Lee et al., 2019)	$51.93 \pm 0.67\%$	$67.42 \pm 0.52\%$
Warp-MAML (Flennerhag et al., 2020)	$52.30 \pm 0.80\%$	$68.4 \pm 0.60\%$
MC (128) (Park & Oliva, 2020)	$\mathbf{54.08 \pm 0.93}\%$	$67.99 \pm 0.73\%$
sparse-MAML (Von Oswald et al., 2021)	$51.04 \pm 0.59\%$	$68.05 \pm 0.84\%$
MeTAL (Baik et al., 2021)	$52.63 \pm 0.37\%$	$\mathbf{70.52 \pm 0.29}\%$
MixtFSL (Afrasiyabi et al., 2021)	$52.82 \pm 0.63\%$	$\mathbf{70.67 \pm 0.57}\%$
Reptile (32) (Nichol et al., 2018)	$49.97 \pm 0.32\%$	$65.99 \pm 0.58\%$
Eigen-Reptile (32)	$51.80 \pm 0.90\%$	$68.10 \pm 0.50\%$
Eigen-Reptile (64)	$\mathbf{53.25 \pm 0.45}\%$	$\mathbf{69.85 \pm 0.85}\%$

Table 2. Few Shot Classification on CIFAR-FS N-way K-shot accuracy. The ± shows 95% confidence interval over tasks.



Algorithm	5-way 1-shot	5-way 5-shot
MAML	$58.90 \pm 1.90\%$	$71.50 \pm 1.00\%$
PROTO NET	$55.50 \pm 0.70\%$	$72.00 \pm 0.60\%$
GNN	61.90%	75.30%
ECM	$55.14 \pm 0.48\%$	$71.66 \pm 0.39\%$
Reptile	$58.30 \pm 1.20\%$	$75.45 \pm 0.55\%$
Eigen-Reptile	$\mathbf{61.90 \pm 1.40}\%$	$\mathbf{78.30 \pm 0.50}\%$

Experiments on corrupted Mini-Imagenet

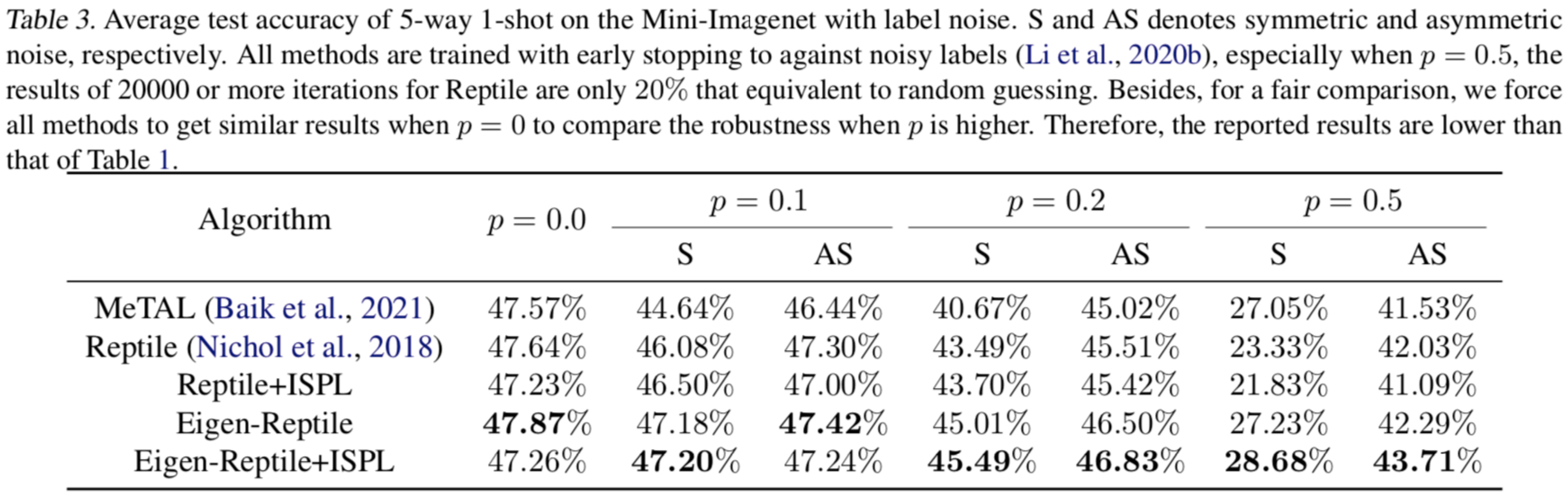


Table 3. Average test accuracy of 5-way 1-shot on the Mini-Imagenet with label noise. S and AS denotes symmetric and asymmetric noise, respectively. All methods are trained with early stopping to against noisy labels (Li et al., 2020b), especially when $p = 0.5$, the results of 20000 or more iterations for Reptile are only 20% that equivalent to random guessing. Besides, for a fair comparison, we force all methods to get similar results when $p = 0$ to compare the robustness when p is higher. Therefore, the reported results are lower than that of Table 1.

Algorithm	$p = 0.0$	$p = 0.1$		$p = 0.2$		$p = 0.5$	
		S	AS	S	AS	S	AS
MeTAL (Baik et al., 2021)	47.57%	44.64%	46.44%	40.67%	45.02%	27.05%	41.53%
Reptile (Nichol et al., 2018)	47.64%	46.08%	47.30%	43.49%	45.51%	23.33%	42.03%
Reptile+ISPL	47.23%	46.50%	47.00%	43.70%	45.42%	21.83%	41.09%
Eigen-Reptile	**47.87**%	47.18%	**47.42**%	45.01%	46.50%	27.23%	42.29%
Eigen-Reptile+ISPL	47.26%	**47.20**%	47.24%	**45.49**%	**46.83**%	**28.68**%	**43.71**%

Thanks