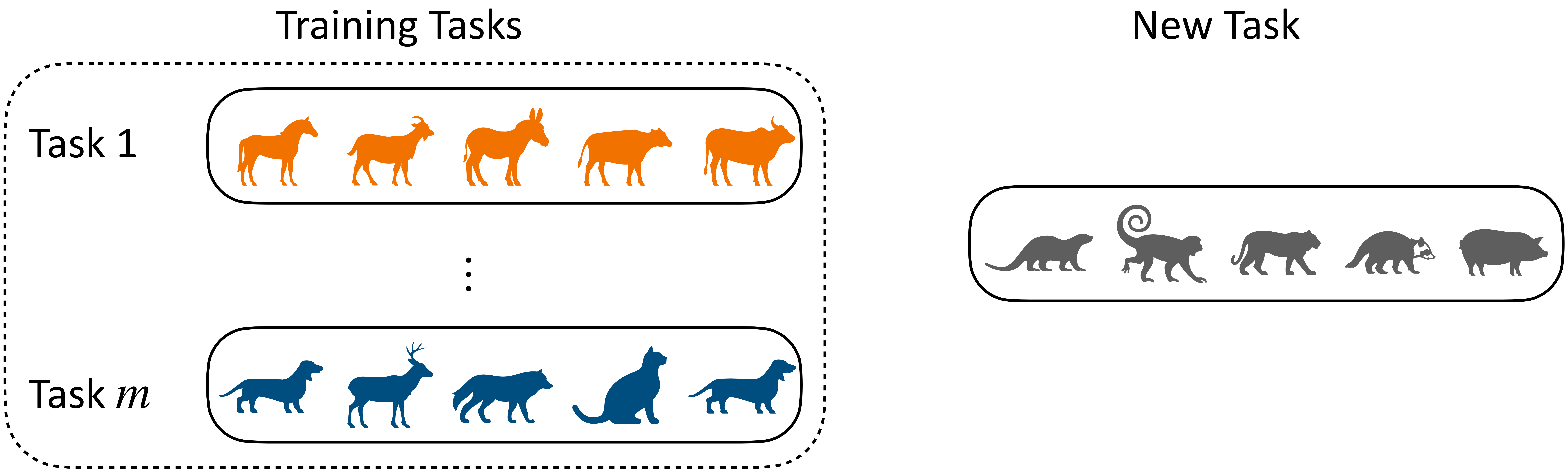
On the Stability and Generalization of Meta-Learning

Yunjuan Wang, Raman Arora
Johns Hopkins University

NeurIPS 2024

Motivation

- Large sets of training data from a single task are often lacking. Training data may stem from diverse tasks with shared similarities, while test data come from entirely new tasks.
- The challenge is to rapidly adapt to these unseen tasks without the need to train from scratch.



Problem Setup

Task Distribution

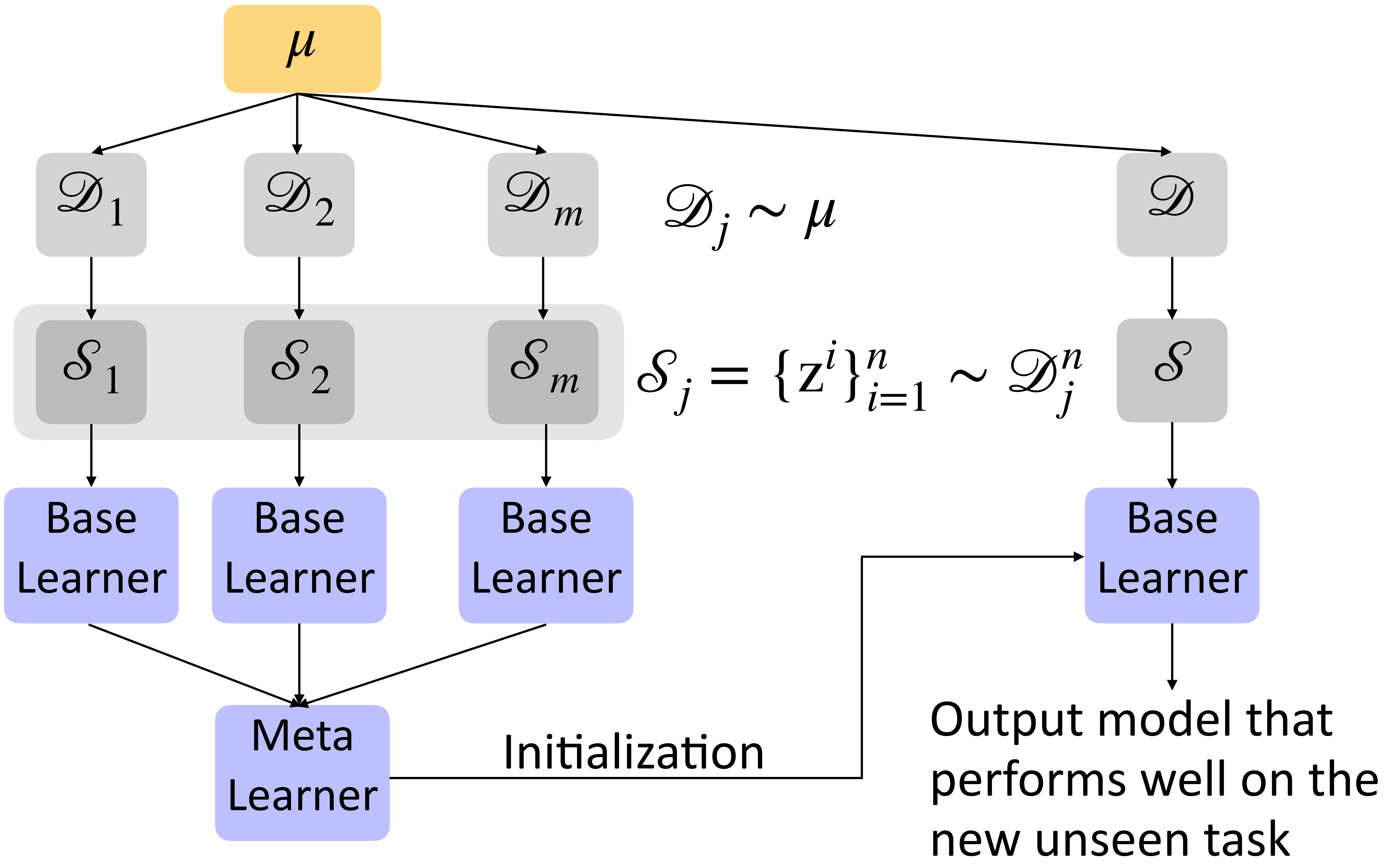
Sample m different tasks

n training data for each task

Meta-sample $\mathbf{S} = \{\mathcal{S}_j\}_{j=1}^{m}$

Task level learning

Meta level learning



Problem Setup

- Consider a supervised learning setting where each data point is denoted by $\mathrm{z} = (\mathrm{x}, y)$ drawn from some unknown distribution $\mathscr{D}$ over $\mathscr{Z} = \mathscr{X} \times \mathscr{Y}$, where input space $\mathscr{X}$, label space $\mathscr{Y}$.

- A meta-learning algorithm $\mathscr{A}$ takes the meta-sample $\mathbf{S}$ as input and outputs an algorithm $\mathscr{A}(\mathbf{S}) : (\mathscr{X} \times \mathscr{Y})^n \to \mathscr{H}$.

- **Goal:** learn a useful prior over tasks to help with rapid adaptation to new tasks.

- Transfer risk: $L(\mathscr{A}(\mathbf{S}), \mu) = \mathbb{E}_{\mathscr{D}\sim\mu}\mathbb{E}_{\mathscr{S}\sim\mathscr{D}^n}L(\mathscr{A}(\mathbf{S})(\mathscr{S}), \mathscr{D}) = \mathbb{E}_{\mathscr{D}\sim\mu}\mathbb{E}_{\mathscr{S}\sim\mathscr{D}^n}\mathbb{E}_{\mathrm{z}\sim\mathscr{D}}\ell(\mathscr{A}(\mathbf{S})(\mathscr{S}), z)$.

- Empirical multi-task risk: $L(\mathscr{A}(\mathbf{S}), \mathbf{S}) = \frac{1}{m}\sum_{j=1}^{m} L(\mathscr{A}(\mathbf{S})(\mathscr{S}_j), \mathscr{S}_j) = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{n}\sum_{i=1}^{n}\ell(\mathscr{A}(\mathbf{S})(\mathscr{S}_j), \mathrm{z}_j^i)$.

Problem Setup

Task level learning: Given a meta-model (parameterized by meta-parameter w), the hope is that it can be adapted easily to a new task $\mathscr{D} \sim \mu$; in particular, a task-specific model u can be quickly learned from a task-specific training set $\mathscr{S} \sim \mathscr{D}^n$ of size n using the following proximal update:

$$\mathrm{u} = \arg\min_{\mathrm{u} \in \mathscr{W}} L(\mathrm{u}, \mathscr{S}) + \frac{\lambda}{2} \|\mathrm{u} - \mathrm{w}\|^2$$

Meta level learning: w itself is learned on the given meta-sample $\mathbf{S} = \{\mathscr{S}_j\}_{j=1}^m$ by minimizing a regularized empirical loss averaged over tasks:

$$\hat{\mathrm{w}} = \arg\min_{\mathrm{w} \in \mathscr{W}} \frac{1}{m} \sum_{j=1}^{m} \min_{\mathrm{u} \in \mathscr{W}} \left[L(\mathrm{u}, \mathscr{S}_j) + \frac{\lambda}{2} \|\mathrm{u} - \mathrm{w}\|^2 \right]$$

Proximal Meta Learning Algorithm

Algorithm 1 Prox Meta-Learning Algorithm $\mathcal{A}$

Input: Meta-sample $\mathbf{S}=\{\mathcal{S}_j\}_{j=1}^m$, epochs T, K, step sizes γ, η, regularization parameter λ

$\mathrm{w}_1 = 0$.
for $t = 1, 2, \ldots, T$ **do**
  for $j = 1, \ldots, m$ **do**
    $\mathrm{u}(\mathrm{w}_t, \mathcal{S}_j) = \mathcal{A}_{\text{task}}(\mathrm{w}_t, \mathcal{S}_j, K, \eta, \lambda)$
    % Using Algorithm 2
  end for
  Calculate the gradient, $\forall j \in [m]$,
  $\nabla F_{\mathcal{S}_j}(\mathrm{u}(\mathrm{w}_t, \mathcal{S}_j), \mathrm{w}_t) = -\lambda(\mathrm{u}(\mathrm{w}_t, \mathcal{S}_j) - \mathrm{w}_t)$.
  Update $\mathrm{w}_{t+1} = \mathrm{w}_t - \frac{\gamma}{m}\sum_{j=1}^m \nabla F_{\mathcal{S}_j}(\mathrm{u}(\mathrm{w}_t, \mathcal{S}_j), \mathrm{w}_t)$
  $\mathrm{w}_{t+1} = \Pi_{\mathcal{W}}(\mathrm{w}_{t+1})$
end for
return $@\mathcal{A}_{\text{task}}(\mathrm{w}_{T+1}, \cdot, K, \eta, \lambda)$

Algorithm 2 Task-specific Algorithm $\mathcal{A}_{\text{task}}$

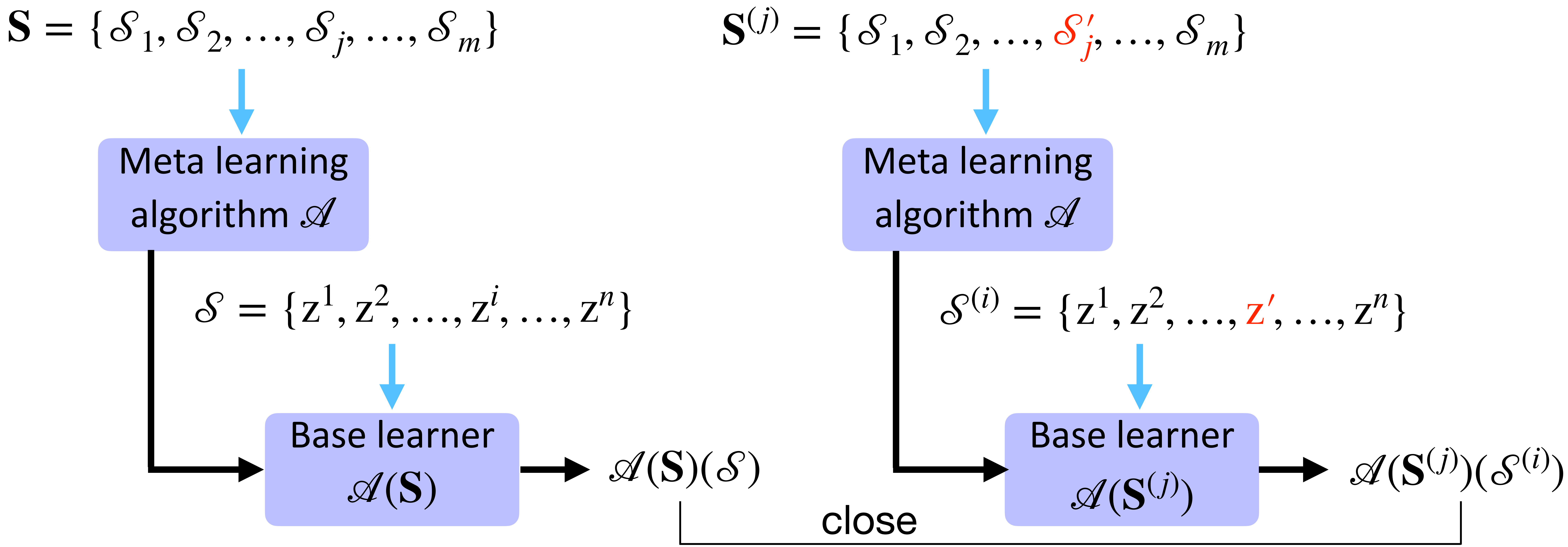
Input: Pretrained model w, training data $\mathcal{S}$, #epochs K, step size η, reg. parameter λ

Option 1 (RERM):
$\mathrm{u}(\mathrm{w}, \mathcal{S}) = \mathrm{argmin}_{\mathrm{u} \in \mathcal{W}} L(\mathrm{u}, \mathcal{S}) + \frac{\lambda}{2}\|\mathrm{u} - \mathrm{w}\|^2$.
Option 2 (GD): Set $\mathrm{u}^{(1)}(\mathrm{w}, \mathcal{S}) = \mathrm{w}$
for $t = 1, 2, \ldots, K-1$ **do**
  $\mathrm{u}^{(k+1)}(\mathrm{w}, \mathcal{S}) = \mathrm{u}^{(k)}(\mathrm{w}, \mathcal{S}) - \eta(\nabla L(\mathrm{u}^{(k)}(\mathrm{w}, \mathcal{S}), \mathcal{S}) + \lambda(\mathrm{u}^{(k)}(\mathrm{w}, \mathcal{S}) - \mathrm{w}))$
  $\mathrm{u}^{(k+1)}(\mathrm{w}, \mathcal{S}) = \Pi_{\mathcal{W}}(\mathrm{u}^{(k+1)}(\mathrm{w}, \mathcal{S}))$
end for
return Option 1 (RERM): $\mathrm{u}(\mathrm{w}, \mathcal{S})$
  Option 2 (GD): $\frac{1}{K}\sum_{k=1}^K \mathrm{u}^{(k)}(\mathrm{w}, \mathcal{S})$

Uniform meta-stability

Definition: A meta-learning algorithm $\mathscr{A}$ is $\bar{\beta}$-uniformly meta-stable if for any neighboring meta-samples $\mathbf{S}, \mathbf{S}^{(j)}$, and neighboring samples $\mathscr{S}, \mathscr{S}^{(i)}$, for any task $\mathscr{D} \sim \mu$ and any $\mathrm{z} \sim \mathscr{D}$, we have

$$|\ell(\mathscr{A}(\mathbf{S})(\mathscr{S}), \mathrm{z}) - \ell(\mathscr{A}(\mathbf{S}^{(j)})(\mathscr{S}^{(i)}), \mathrm{z})| \leq \bar{\beta}$$



Uniform meta-stability

Definition: A meta-learning algorithm $\mathscr{A}$ is $\bar{\beta}$-uniformly meta-stable if for any neighboring meta-samples $\mathbf{S}, \mathbf{S}^{(j)}$, and neighboring samples $\mathscr{S}, \mathscr{S}^{(i)}$, for any task $\mathscr{D} \sim \mu$ and any $\mathrm{z} \sim \mathscr{D}$, we have

$$|\,\ell(\mathscr{A}(\mathbf{S})(\mathscr{S}), \mathrm{z}) - \ell(\mathscr{A}(\mathbf{S}^{(j)})(\mathscr{S}^{(i)}), \mathrm{z})\,| \leq \bar{\beta}$$

Theorem: Consider a meta-learning problem for some M-bounded loss function ℓ and task distribution μ. For any $\bar{\beta}$-uniformly meta-stable learning algorithm $\mathscr{A}$, we have that with probability at least $1-\delta$,

$$L(\mathscr{A}(\mathbf{S}), \mu) \lesssim L(\mathscr{A}(\mathbf{S}), \mathbf{S}) + \bar{\beta}\log(mn)\log(1/\delta) + M\sqrt{\log(1/\delta)/(mn)}$$

Bound Transfer Generalization Gap

Algorithm	Loss	Conditions	Uniform meta-stability $\bar{\beta}$
Algo. 1 with RERM	convex, G-Lipschitz	$\gamma \leq \frac{1}{\lambda}$	$\frac{G^2}{\lambda m} + \frac{G^2}{\lambda n}$
Algo. 1 with RERM	convex, H-smooth, M-bounded	$\gamma \leq \frac{1}{\lambda}, \lambda \geq H$	$\frac{HM}{\lambda(2n-1)} + \frac{HM}{\lambda(m+1)}$
Algo. 1 with GD	convex, G-Lipschitz, H-smooth	$\eta \leq \frac{2}{H+2\lambda}, \gamma \leq \frac{1}{\lambda T}$	$\frac{G^2}{\lambda m} + \frac{G^2}{\lambda n}$
Algo. 1 with GD	ρ-weakly convex, G-Lipschitz	$\eta \leq \frac{1}{\lambda}, \gamma \leq \frac{1}{\lambda T}, \lambda \geq 2\rho$	$G^2\sqrt{\frac{\eta}{\lambda}} + \frac{G^2}{\lambda m} + \frac{G^2}{\lambda n}$
Algo. 3 with GD	ρ-weakly convex, G-Lipschitz	$\eta \leq \frac{1}{\lambda}, \gamma \leq \frac{1}{\lambda T}, \lambda \geq 2\rho$	$G^2\sqrt{\frac{\eta}{\lambda}} + \frac{G^2}{\lambda m} + \frac{G^2}{\lambda n}$, w.h.p.

Table 1: Bounds on uniform meta-stability $\bar{\beta}$ for different families of learning problems. Here, η is the step-size for GD for task-specific learning, γ is the step-size for GD for meta-parameter learning, m is the number of tasks during training, n is the number of training data for the task at test time.

Extension Proximal Meta-Learning with Stochastic Optimization

Robust Adversarial Proximal Meta-Learning

Excess Transfer Risk

$$\underbrace{L(\mathscr{A}(\mathbf{S})(\mathscr{S}),\mathscr{D}) - L(\mathrm{u}_*,\mathscr{D})}_{\text{Excess Transfer Risk } \mathscr{E}_{\text{risk}}(\mathscr{A})} = \underbrace{L(\mathscr{A}(\mathbf{S})(\mathscr{S}),\mathscr{D}) - \frac{1}{m}\sum_{j=1}^{m} L(\mathscr{A}(\mathbf{S})(\mathscr{S}_j),\mathscr{S}_j)}_{\text{Generalization Gap } \mathscr{E}_{\text{gen}}(\mathscr{A})}$$

$$+ \underbrace{\frac{1}{m}\sum_{j=1}^{m} L(\mathscr{A}(\mathbf{S})(\mathscr{S}_j),\mathscr{S}_j) - L(\mathrm{u}_j^*,\mathscr{S}_j)}_{\text{Optimization and Approximation Error } \mathscr{E}_{\text{opt+app}}(\mathscr{A})}$$

$$+ \underbrace{L(\mathrm{u}_j^*,\mathscr{S}_j) - L(\mathrm{u}_*,\mathscr{S}_j)}_{\leq 0} + \underbrace{L(\mathrm{u}_*,\mathscr{S}_j) - L(\mathrm{u}_*,\mathscr{D})}_{\mathbb{E}_{\forall j\in[m],\mathscr{S}_j\sim\mathscr{D}_j^n,\mathscr{D}_j\sim\mu,\mathscr{D}\sim\mu}=0}$$

$\mathrm{u}_* = \arg\min_{\mathrm{u}\in\mathscr{W}} L(\mathrm{u},\mathscr{D})$ optimal task-specific hypothesis for the unseen task;

$\mathrm{u}_j^* = \arg\min_{\mathrm{u}\in\mathscr{W}} L(\mathrm{u},\mathscr{S}_j)$ optimal task-specific hypothesis for the given training tasks.

Excess Transfer Risk

Convex and smooth loss:

Setting $\eta = \mathcal{O}\left(\frac{1}{\lambda\sqrt{K}}\right)$ gives us $\mathbb{E}\left[\mathcal{E}_{\text{risk}}(\mathscr{A})\right] \leq \mathcal{O}\left(\frac{1}{\lambda\sqrt{K}} + \frac{1}{\lambda m} + \frac{1}{\lambda n} + \frac{\lambda}{T} + \lambda\sigma^2\right)$, where $\sigma^2 = \frac{1}{m}\sum_{j=1}^{m}\|\hat{\mathrm{w}} - \mathrm{u}_j^*\|^2$ is the approximation error.

Convex and non-smooth loss:

Setting $\eta = \mathcal{O}\left(\frac{1}{\lambda K^{2/3}}\right)$ gives us $\mathbb{E}\left[\mathcal{E}_{\text{risk}}(\mathscr{A})\right] \leq \mathcal{O}\left(\frac{1}{\lambda K^{1/3}} + \frac{1}{\lambda m} + \frac{1}{\lambda n} + \frac{\lambda}{T} + \lambda\sigma^2\right)$

Thank you!