ST-MAML : A Stochastic-Task based Method for Task-Heterogeneous Meta-Learning Supplementary Material

Zhe Wang[1] **Jake Grigsby**[1] **Arshdeep Sekhon**[1] **Yanjun Qi**[1]

[1]Computer Science Dept., University of Virginia, Charlottesville, Virginia, USA

S1 MODEL COMPARISON.

Table S1: Model comparison table. HoMAMLs are MAMLs designed for task homogeneity, and HeMAMLs are for heterogeneity. NPs describe methods in Neural Processes family. PMAMLs mean probabilistic extensions of MAML. Aug feature represents the augmented features.

Category	Tasks	Knowledge Set	Tailoring	Sampling	Inference on
HoMAMLs	MAML Finn et al. (2017)	Initialization			
	MetaSGD Li et al. (2017)	Initialization+lr			
HeMAMLs	MMAML Vuorio et al. (2019)	Initialization	✓		
	HSML Yao et al. (2019)	Initialization	✓		
NPs	NP Garnelo et al. (2018b)	Aug feature		✓	Representation
	CNP Garnelo et al. (2018a)	Aug feature			
PMAMLs	BMAML Yoon et al. (2018)	Initialization		✓	Parameters
	PLATIPUS Finn et al. (2018)	Initialization		✓	Parameters
	ST-MAML	Initialization+Aug feature	✓	✓	Representation

S2 APPROXIMATION FOR POSTERIOR DISTRIBUTION $q(Z_{\mathcal{T}})$.

Given the training set $\boldsymbol{D}^{tr}_{\mathcal{T}}$ of a task $\mathcal{T}$, the stochastic task variable $\boldsymbol{Z}_{\mathcal{T}}$ is supposed to infer its posterior distribution conditioned on $\boldsymbol{D}^{tr}_{\mathcal{T}}$ only, specifically, we have the true posterior:

$$p(\boldsymbol{Z}_{\mathcal{T}}|\mathcal{T}) = \frac{p(\boldsymbol{Z}_{\mathcal{T}}|\boldsymbol{D}^{tr}_{\mathcal{T}})p(\boldsymbol{Y}^{te}_{\mathcal{T}}|\boldsymbol{Z}_{\mathcal{T}}, \boldsymbol{X}^{te}_{\mathcal{T}}, \boldsymbol{D}^{tr}_{\mathcal{T}})}{p(\mathcal{T})} \tag{1}$$

the empirical distribution $p(\mathcal{T})$ is only known in the form of $\{(\boldsymbol{D}^{tr}_{\mathcal{T}}, \boldsymbol{D}^{te}_{\mathcal{T}})\}$ pairs. Thus, the true posterior distribution is intractable. Based on our design, we suppose the prior distribution $p(\boldsymbol{Z}_{\mathcal{T}}|\boldsymbol{D}^{tr}_{\mathcal{T}})$ is a multivariate Gaussian distribution, whose mean and variance is the output of a set operator acting on $(\boldsymbol{X}^{tr}_{\mathcal{T}}, \boldsymbol{Y}^{tr}_{\mathcal{T}})$ pairs. To ensure the posterior stays close to the prior, also the posterior is derived from $(\boldsymbol{D}^{tr}_{\mathcal{T}}, \boldsymbol{D}^{te}_{\mathcal{T}})$, we approximate it with the output of the same set operator acting on both $(\boldsymbol{X}^{tr}_{\mathcal{T}}, \boldsymbol{Y}^{tr}_{\mathcal{T}})$ and $(\boldsymbol{X}^{te}_{\mathcal{T}}, \boldsymbol{Y}^{te}_{\mathcal{T}})$ pairs.

S3 DERIVATION OF ELBO APPROXIMATION AS VARIATIONAL INFORMATION BOTTLENECK OBJECTIVE

For task $\mathcal{T}$, our fine-tuned task-specific knowledge set $\boldsymbol{\Theta}^1_{\mathcal{T}}$ contains two variables: model parameters $\boldsymbol{\theta}^1_{\mathcal{T}}$ and augmented features $\mathbf{h}^1_{\mathcal{T}}$. Given task inputs $\mathbf{X}_{\mathcal{T}} = [\boldsymbol{X}^{tr}_{\mathcal{T}}, \boldsymbol{X}^{te}_{\mathcal{T}}]$, we are seeking a task-specific knowledge set that is maximally informative

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

of test target $\boldsymbol{Y}_{\mathcal{T}}^{te}$, while being mostly compressive of training target $\boldsymbol{Y}_{\mathcal{T}}^{tr}$. Correspondingly, we would like to maximize the conditional mutual information $I(\boldsymbol{Y}_{\mathcal{T}}^{te}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$ and minimize $I(\boldsymbol{Y}_{\mathcal{T}}^{tr}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$. The information bottleneck objective is:

$$\mathcal{L}_{IB}(\mathcal{T}) = I(\boldsymbol{Y}_{\mathcal{T}}^{te}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) - \beta I(\boldsymbol{Y}_{\mathcal{T}}^{tr}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}). \tag{2}$$

We show the following lemma:

Lemma 1 *Given a task $\mathcal{T}$, maximizing the information bottleneck loss $\mathcal{L}_{IB}$ defined in* (2) *is equivalent to maximizing the weighted ELBO :*

$$\mathcal{L}_{wELBO}(\mathcal{T}) = \mathbf{E}_{\boldsymbol{\Theta}_{\mathcal{T}}^{1} \sim q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathcal{T})} \log p(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \boldsymbol{X}_{\mathcal{T}}^{te}) - \beta KL(q(\boldsymbol{Z}_{\mathcal{T}} | \mathcal{T}) || p(\boldsymbol{Z}_{\mathcal{T}} | \boldsymbol{D}_{\mathcal{T}}^{tr}). \tag{3}$$

Proof 1 *To lower bound IB objective defined in Eq. (2), we derive the lower bound for first term $I(\boldsymbol{Y}_{\mathcal{T}}^{te}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$ and upper bound for second term $I(\boldsymbol{Y}_{\mathcal{T}}^{tr}; \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$. Further, we assume a distribution $q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$ as a variational approximation of the true distribution $p(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})$.*

$$\begin{aligned} I(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}})}{p(\boldsymbol{Y}_{\mathcal{T}}^{te}) q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | X)} d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \\ &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})}{p(\boldsymbol{Y}_{\mathcal{T}}^{te})} d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \end{aligned} \tag{4}$$

$$\begin{aligned} q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) &= \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} \\ &= \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} d\boldsymbol{Y}_{\mathcal{T}}^{te} \end{aligned} \tag{5}$$

$$\begin{aligned} q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) &= \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}, \boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te} | \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} \\ &= \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}, | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te} | \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} \\ &= \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}, | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te} | \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} \end{aligned} \tag{6}$$

The last part follows from the fact that $\boldsymbol{\Theta}_{\mathcal{T}}^{1}$ is independent of $\boldsymbol{Y}_{\mathcal{T}}^{te}$ given $[\mathbf{X}_{\mathcal{T}}, \boldsymbol{Y}_{\mathcal{T}}^{tr}]$. Putting this together:

$$q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) = \frac{\int p(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{Y}_{\mathcal{T}}^{tr}) q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr}}{\int p(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{Y}_{\mathcal{T}}^{tr}) q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} d\boldsymbol{Y}_{\mathcal{T}}^{te}} \tag{7}$$

However, the above conditional distribution $q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})$ is intractable due to the unknown data distribution $p(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{Y}_{\mathcal{T}}^{tr})$. To derive the upper bound, we introduce a variational approximation $p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})$ for $q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})$. Take it into the Eq. (4), we have:

$$\begin{aligned} I(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) \log \frac{p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})}{p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{te})} d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \\ &\geq \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) \log \frac{p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})}{p(\boldsymbol{Y}_{\mathcal{T}}^{te})} d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \\ &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1} | \mathbf{X}_{\mathcal{T}}) \log p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} + C \\ &= \int q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) \log p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} d\mathbf{X}_{\mathcal{T}} + C \end{aligned} \tag{8}$$

In the above equation, we use $KL(q(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) || p_\theta(\boldsymbol{Y}_{\mathcal{T}}^{te} | \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}})) \geq 0$ in the second step.
The second term is irrelevant to our objective so we can treat it as a constant. Note that:

$$q(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) = \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1} | \boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{Y}_{\mathcal{T}}^{te} | \mathbf{X}_{\mathcal{T}}) p(\mathbf{X}_{\mathcal{T}}) d\boldsymbol{Y}_{\mathcal{T}}^{tr} \tag{9}$$

Thus, an unbiased estimation of the first term is:

$$I(\boldsymbol{Y}_{\mathcal{T}}^{te}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) \geq \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) \log p_{\theta}(\boldsymbol{Y}_{\mathcal{T}}^{te}|\boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) d\boldsymbol{\Theta}_{\mathcal{T}}^{1}. \tag{10}$$

We derive the upper bound for second term:

$$\begin{aligned} I(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})}{p(\boldsymbol{Y}_{\mathcal{T}}^{tr}) q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})} d\boldsymbol{Y}_{\mathcal{T}}^{tr} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \\ &= \int p(\mathbf{X}_{\mathcal{T}}) \left[\int q(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}})}{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})} d\boldsymbol{Y}_{\mathcal{T}}^{tr} d\boldsymbol{\Theta}_{\mathcal{T}}^{1} \right] d\mathbf{X}_{\mathcal{T}} \end{aligned} \tag{11}$$

The denominator $q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) = \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}) d\boldsymbol{Y}_{\mathcal{T}}^{tr}$ is intractable for unknown $p(\boldsymbol{Y}_{\mathcal{T}}^{tr})$. We approximate it with $p_{\theta}(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})$. With similar derivation, the second term is upper bounded by:

$$I(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) \leq \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) p(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}})}{p_{\theta}(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})} d\boldsymbol{Y}_{\mathcal{T}}^{te} d\boldsymbol{Y}_{\mathcal{T}}^{tr} d\boldsymbol{\Theta}_{\mathcal{T}}^{1}. \tag{12}$$

Similarly, its unbiased estimation is given as:

$$I(\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}}) \leq \int q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) \log \frac{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}})}{p_{\theta}(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})} d\boldsymbol{\Theta}_{\mathcal{T}}^{1}. \tag{13}$$

Combining two terms, we get the total unbiased estimation of the IB loss:

$$L_{IB} = \mathbf{E}_{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}})} \log p_{\theta}(\boldsymbol{Y}_{\mathcal{T}}^{te}|\boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) - \beta KL(q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}}) || p_{\theta}(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathbf{X}_{\mathcal{T}})). \tag{14}$$

To incorporate target information, we inject the target variable $\boldsymbol{Y}_{\mathcal{T}}^{te}$ into posterior and $\boldsymbol{Y}_{\mathcal{T}}^{tr}$ into prior, and get the new approximation:

$$L_{IB} = \mathbf{E}_{q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathcal{T})} \log p_{\theta}(\boldsymbol{Y}_{\mathcal{T}}^{te}|\boldsymbol{\Theta}_{\mathcal{T}}^{1}, \mathbf{X}_{\mathcal{T}}) - \beta KL(q(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\mathcal{T}) || p_{\theta}(\boldsymbol{\Theta}_{\mathcal{T}}^{1}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \mathbf{X}_{\mathcal{T}})). \tag{15}$$

Since $\boldsymbol{\theta}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{w}}^{Gate}(\boldsymbol{\theta}, \boldsymbol{Z}_{\mathcal{T}}), \mathbf{h}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{\beta}}^{Gate}(\boldsymbol{Z}_{\mathcal{T}})$, where $\boldsymbol{g}_{\boldsymbol{w}}^{Gate}, \boldsymbol{g}_{\boldsymbol{\beta}}^{Gate}$ are both deterministic and invertible mappings of $\boldsymbol{Z}_{\mathcal{T}}$. We have $p(\boldsymbol{\theta}_{\mathcal{T}}^{0}|\boldsymbol{\theta}) = \delta(\boldsymbol{\theta}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{w}}^{Gate}(\boldsymbol{Z}_{\mathcal{T}}, \boldsymbol{\theta})), p(\mathbf{h}_{\mathcal{T}}^{0}|\boldsymbol{Z}_{\mathcal{T}}) = \delta(\mathbf{h}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{\beta}}^{Gate}(\boldsymbol{Z}_{\mathcal{T}}))$. Moreover, $\mathbf{h}_{\mathcal{T}}^{0}, \boldsymbol{\theta}_{\mathcal{T}}^{0}$ are conditionally independent given $\boldsymbol{Z}_{\mathcal{T}}$. Similarly, $\mathbf{h}_{\mathcal{T}}^{1}, \boldsymbol{\theta}_{\mathcal{T}}^{1}$ are deterministic function of $\mathbf{h}_{\mathcal{T}}^{0}$ and $\boldsymbol{\theta}_{\mathcal{T}}^{0}$. Thus, the second term in Eq. (15) can be replaced with the divergence between the posterior and prior distribution of $\boldsymbol{Z}_{\mathcal{T}}$, i.e. $KL(q(\boldsymbol{Z}_{\mathcal{T}}|\mathcal{T}) || p(\boldsymbol{Z}_{\mathcal{T}}|\boldsymbol{Y}_{\mathcal{T}}^{tr}, \boldsymbol{X}_{\mathcal{T}}^{tr}))$. We now look into the log likelihood term in Eq. (14). Since the transitions $\boldsymbol{Z}_{\mathcal{T}} \rightarrow \boldsymbol{\theta}_{\mathcal{T}}^{0} \rightarrow \boldsymbol{\theta}_{\mathcal{T}}^{1}$ and $\boldsymbol{Z}_{\mathcal{T}} \rightarrow \mathbf{h}_{\mathcal{T}}^{0} \rightarrow \mathbf{h}_{\mathcal{T}}^{1}$ are deterministic:

$$\begin{aligned} \boldsymbol{\theta}_{\mathcal{T}}^{1} = \boldsymbol{\theta}_{\mathcal{T}}^{0} - \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathcal{T}}^{0}}, \mathbf{h}_{\mathcal{T}}^{0}, \boldsymbol{D}_{\mathcal{T}}^{tr})), \quad \boldsymbol{\theta}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{w}}^{Gate}(\boldsymbol{\theta}, \boldsymbol{z}), \quad \boldsymbol{z} \sim q(\boldsymbol{Z}_{\mathcal{T}}|\mathcal{T}) \\ \mathbf{h}_{\mathcal{T}}^{1} = \mathbf{h}_{\mathcal{T}}^{0} - \nabla_{\mathbf{h}} \mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}_{\mathcal{T}}^{0}}, \mathbf{h}_{\mathcal{T}}^{0}, \boldsymbol{D}_{\mathcal{T}}^{tr})), \quad \mathbf{h}_{\mathcal{T}}^{0} = \boldsymbol{g}_{\boldsymbol{\beta}}^{Gate}(\boldsymbol{z}). \end{aligned} \tag{16}$$

According to the analysis, the approximation to be optimized is:

$$L_{app} = \mathbf{E}_{\boldsymbol{\Theta}_{\mathcal{T}}^{1} \sim q(\boldsymbol{\Theta}_{\mathcal{T}}|\mathcal{T})} \log p_{\theta}(\boldsymbol{Y}_{\mathcal{T}}^{te}|\boldsymbol{\theta}_{\mathcal{T}}^{1}, X^{te}) - \beta KL(q(\boldsymbol{Z}_{\mathcal{T}}|\mathcal{T}) || p(\boldsymbol{Z}_{\mathcal{T}}|\boldsymbol{D}_{\mathcal{T}}^{tr})). \tag{17}$$

S4 HETEROGENEOUS FEW SHOT BINARY CLASSIFICATION RESULTS.

Task design. In classification, task ambiguity is common when annotated data are limited. Images can share many attributes, and various combinations of them can be used for final decision-making. We evaluate our method on the ambiguous classification benchmark proposed in Finn et al. (2018). The CelebA dataset contains cropped images of celebrity faces and a list of attributes that describe their appearance. We split these attributes into training, validation, and test sets. During meta-training, we randomly sample two training attributes and form the positive class of images that share them. The negative class is formed by sampling the same number of images containing neither attribute. During meta-testing, training set images share three attributes. We construct three test sets by choosing two of the three attributes to define the positive class. The model learns to apply two attributes for decision making, but there are three combinations of two attributes for classification. Thus the task is ambiguous. We sample models from our distribution of solutions and assign them to the three test sets based on the loss values. If all test sets are covered with at least one model, the method can effectively discover all potential decision rules. The cover number is calculated as the average number of test sets that are covered. The coverage number for a deterministic method is 1. As Table S2 shows, our method can 1) achieve better accuracy, 2) reach lower NLL, and 3) discover more decision rules compared to MAML.

Table S2: 5-Shot Ambiguous Binary Classification.

Model	Accuracy	Coverage number	NLL
MAML	77.924	1.00	0.454
ST-MAML	79.698	1.13	0.439

S5 EXPERIMENT SETUP FOR SIMULATION.

2D Regression setup. Meta distribution $\mathcal{T}$ contains 6 function families. Input $X = [x_1, x_2] \sim U(0.0, 5.0)$. The value for x_2 is fixed as 1 if only x_1 is used. For *sinusoids* families : $y = asin(wx_1 + b) + \epsilon$, where $a \sim U[0.1, 5.0], b \sim U[0, 2\pi], w \sim U[0.8, 1.2]$; for *line* families: $y = ax_1 + b + \epsilon$, where $a \sim U[-3.0, 3.0], b \sim U[-3.0, 3.0]$; for *quadratic curves*: $y = ax_1^2 + bx_1 + c + \epsilon$, where $a \sim U[-0.2, 0.2], b \sim U[-2.0, 2.0], c \sim U[-3.0, 3.0]$; for cubic curves: $y = ax_1^3 + bx_1^2 + cx_1 + d + \epsilon$, where $a \sim U[-0.1, 0.1], b \sim U[-0.2, 0.2], c \sim U[-2.0, 2.0], d \sim U[-3.0, 3.0]$; for *quadratic surface*: $y = ax_1^2 + bx_2^2 + \epsilon$, where $a \sim U[-1.0, 1.0], b \sim U[-1.0, 1.0]$; for *ripple*: $y = sin(-a(x_1^2 + x_2^2)) + b + \epsilon$, where $a \sim U[-0.2, 0.2], b \sim U[-3.0, 3.0]$.

Model architecture for 2D regression. We adopt the same base model as in Yao et al. (2020); Finn et al. (2017), it contains 2 linear layer with 40 neurons followed by ReLU function. For the task representative module, we use 2 linear layers with 80 neurons.

Visualization for 2D regression. See Figure S1.

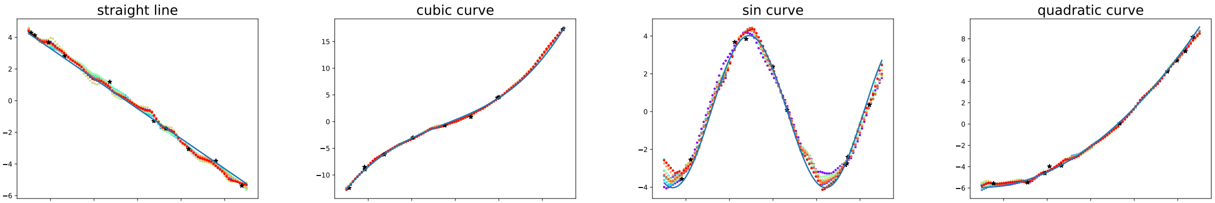

Figure S1: Qualitative Visualization of fitting curves. Black stars represent training set $\boldsymbol{D}_{\mathcal{T}}^{tr}$, 10 different samples of fitting curves are shown as colored dotted lines. The blue solid line is the true mapping.

More results for 2D regression. During meta-training, we fixed the size of training set $|\boldsymbol{D}_{\mathcal{T}}^{tr}|$ as 10, the standard deviation for Gaussian noise σ to be 0.3. During meta-testing, we can decrease the size of training set or increase the noise level such that tasks ambiguity can be more concerning. We visualize them in Figure 4. The model can effectively reason over ambiguity as we vary the size of the training data or noise level. The sampled functions tend to span wider space as $|\boldsymbol{D}_{\mathcal{T}}^{tr}|$ decreases or the noise level increases. However, they stay faithful around those annotated training data.

NOAA GSOD Dataset Details. The data is available at `https://data.noaa.gov/dataset/dataset/global-surface-summary-of-the-day-gsod`. The dataset is large, so we reduce the size while preserving a wide range of years by using every 10th year from $1969 - 2019$. Each file in the unzipped dataset corresponds to one year of data at a particular station. Files that do not contain at least 40 days of data are ignored. Task number i is created in the following way:

1. We sample 40 days of data that have valid temperature entires.
2. We drop the columns ("STATION", "NAME", "TEMP_ATTRIBUTES", "DEWP", "DEWP_ATTRIBUTES", "PRCP_ATTRIBUTES", "SLP_ATTRIBUTES", "STP_ATTRIBUTES", "VISIB_ATTRIBUTES", "WDSP_ATTRIBUTES", "MAX", "MIN", "MAX_ATTRIBUTES", "MIN_ATTRIBUTES", "LATITUDE", and "LONGITUDE")
3. We convert the date column from (MM/DD/YYYY) to a float [0, 1] representing the time since the first day of that year.
4. The “FRSHTT" is a 6 bit binary string where each digit indicates the presence of fog, rain, snow, hail, thunder, and tornadoes respectively. We transform the “FRSHTT" column into 6 binary columns.
5. The GSOD dataset reports missing values with all 9s, e.g. 99.99, or 999.9. We find and replace these values with 0.0. We also replace NaN entries with 0.0.
6. The units of some input variables are adjusted to bring their values down to a smaller range. Pressure variables (“SLP" and “STP") are converted from millibars to bars. Elevation is changed from meters to kilometers.
7. The “TEMP" variable is split from the data to become our target value.

We use a 42k/5k/1k split to divide the files into train, val and test sets.

Model architecture for weather prediction. Similar to 2D regression, the feature learner has two linear layers with 100 neurons followed by ReLU activation funcion. The mapping to task representation $\boldsymbol{Z}_{\mathcal{T}}$ contains 3 layers with hidden dimension 40. 80, 200. The augmented dimension is set to be 20.

Model runtime and compute. The model trains on one GTX 2080 card. Training times vary by experiment, ranging from a few hours to a day.

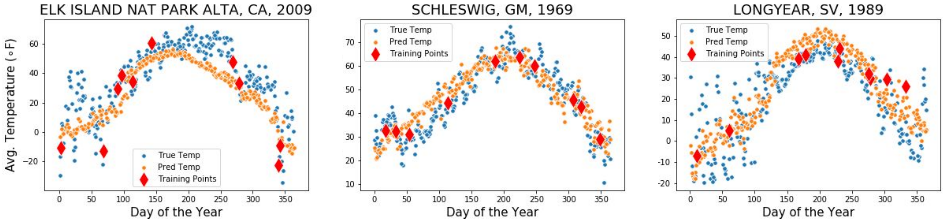

Figure S2: A visualization of trained ST−MAML on the NOAA-GSOD temperature prediction task. The model is given 10 training points (red) and predicts the remaining days of the year (orange). The true temperatures are shown in blue.