Set-based Meta-Interpolation for Few-Tasks Meta-Learning

Seanie Lee [1*], Bruno Andreis [1*], Kenji Kawaguchi [2], Juho Lee [1,3], Sung Ju Hwang [1]

*: Equal Contribution, [1] KAIST, [2] NUS, [3] AITRICS

KAIST | AITRICS | NUS National University of Singapore, School of Computing | NEURAL INFORMATION PROCESSING SYSTEMS

Summary

Motivation

- Most of existing Meta-Learning methods requires **a large amount of** meta-training tasks.
- Data augmentations **require domain-specific knowledge** to design task augmentations.
- Manifold Mixup is **not effective** for non-image domain.

Contributions

- We propose **Meta-Interpolation**, utilizing **set function** to interpolate two tasks for augmentation.
- We **theoretically analyze** our model and show that it regularizes the meta-learner for **better generalization**.
- Meta-Interpolation significantly improves the performance of Prototypical Network on **various domains** for few-task meta-learning problem.

Background

Problem Statement

- Given a finite tasks $\{\mathcal{T}_t\}_{t=1}^{T}$, where each task consists of a support set $\mathcal{D}_t^s = \{(x_{t,i}^s, y_{t,i}^s)\}_{i=1}^{N_s}$ and query set $\mathcal{D}_t^s = \{(x_{t,i}^q, y_{t,i}^q)\}_{i=1}^{N_q}$.
- Given a predictive model, $f_{\theta,\lambda}$, we want to estimate the parameters such that it generalizes to unseen query set $\mathcal{D}_*^q$ using a support set $\mathcal{D}_*^s$
- We focus on few-task meta-learning problem, where T is small.

Metric-based Meta-Learning

- We focus on metric-based meta-learning, Prototypical Network.

$$\mathbf{c}_k := \frac{1}{N_k} \sum_{\substack{(\mathbf{x}_{t,i}^s, y_{t,i}^s) \in \mathcal{D}_t^s \\ y_{t,i}=k}} \hat{f}_{\theta,\lambda}(\mathbf{x}_{t,i}^s) \in \mathbb{R}^D$$

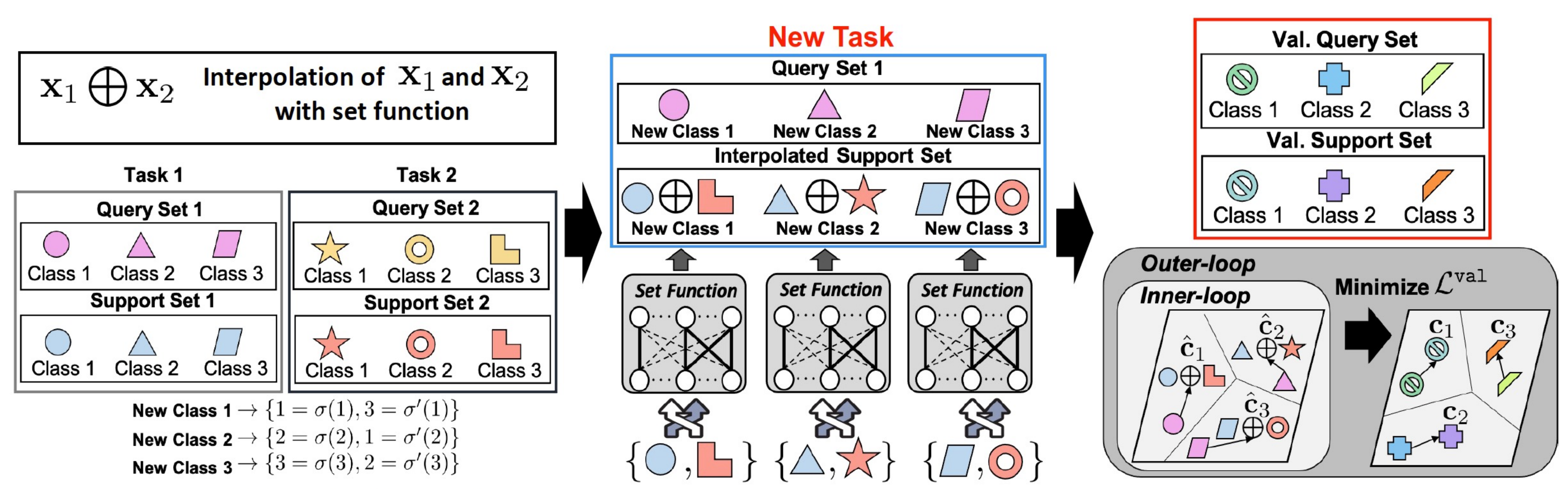
$$\mathcal{L}_{\text{singleton}}(\lambda, \theta; \mathcal{T}_t) := \sum_{i,k} \mathbb{1}_{\{y_{t,i}=k\}} \cdot \log \frac{\exp(-d(\hat{f}_{\theta,\lambda}(\mathbf{x}_{t,i}^q), \mathbf{c}_k))}{\sum_{k'} \exp(-d(\hat{f}_{\theta,\lambda}(\mathbf{x}_{t,i}^q), \mathbf{c}_{k'}))}$$

$$y_*^q = \arg\min_k d(\hat{f}_{\theta,\lambda}(\mathbf{x}_*^q), \mathbf{c}_k)$$

Proposed Method: Meta-Interpolation

Task Interpolation

- We sample two tasks $\mathcal{T}_{t_1} = \{\mathcal{D}_{t_1}^s, \mathcal{D}_{t_1}^q\}$, $\mathcal{T}_{t_2} = \{\mathcal{D}_{t_2}^s, \mathcal{D}_{t_2}^q\}$, we **interpolate the two tasks** with Set Transformer, $\varphi_\lambda: \mathbb{R}^{n \times d} \to \mathbb{R}^d$.
- For support set, we sample two permutations σ_{t_1}, σ_{t_2} on $[K]$ and pair **two instances from class $\sigma_{t_1}(k)$ and $\sigma_{t_2}(k)$**, and interpolate their hidden representations with φ_λ for each $k \in [K]$.
- With the interpolated support set, we get class prototype $\hat{c}_k$.
- For query set, we do not interpolate them. Instead, we measure a distance between the $x_{t_1,i}^q$ with $y_{t_1,i}^q = \sigma_{t_1}(k)$ and the interpolated prototype $\hat{c}_k$



(a) Class Assignment (b) Task Augmentation (c) Bilevel Optimization

Bilevel Optimization

- We consider the parameter of Set Transformer λ as **hyperparameter**.
- We use **Implicit Function Theorem** (Lorraine et al., 2020) to solve the bilevel optimization problem.

$$\lambda^* := \arg\min_{\lambda} \frac{1}{T'} \sum_{t=1}^{T'} \mathcal{L}_{\text{singleton}}(\lambda, \theta^*(\lambda); \mathcal{T}_t^{\text{val}})$$

$$\theta^*(\lambda) := \arg\min_{\theta} \frac{1}{2T} \sum_{t=1}^{T} \mathcal{L}_{\text{singleton}}(\lambda, \theta; \mathcal{T}_t^{\text{train}}) + \mathcal{L}_{\text{mix}}(\lambda, \theta; \hat{\mathcal{T}}_t)$$

Theoretical Analysis

Implicit Regularization by Task Interpolation

- The loss with task interpolation is **approximation** of the original loss with **regularization**.
- In simple **logistic regression**, task interpolation induces data-dependent regularization, which **reduces Rademachar complexity.**

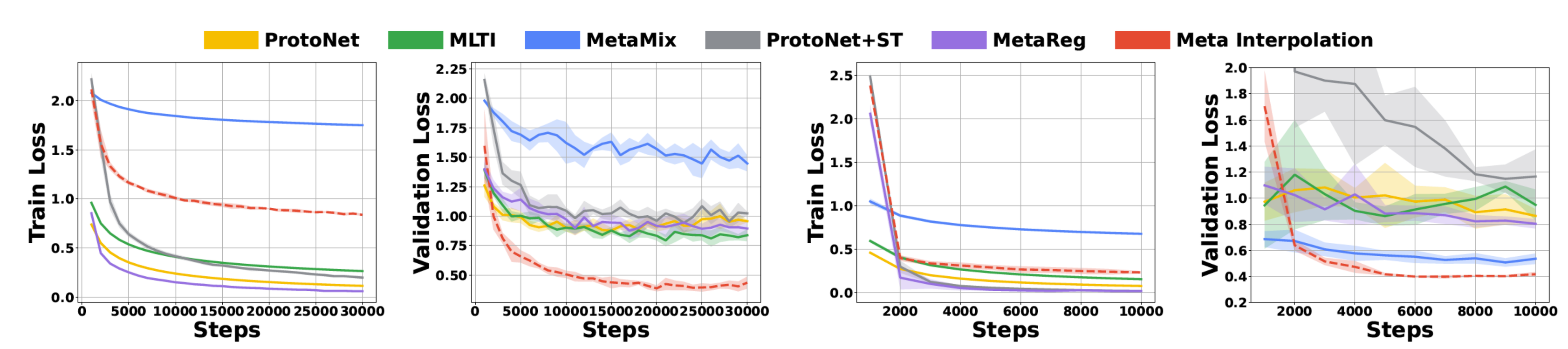
Experimental Results

Table 1: Average accuracy of 5 runs and ±95% confidence interval for few shot classification on non-image domains – Tox21, NCI, GLUE-SciTail dataset, and ESC-50 datasets. ST stands for Set Transformer.

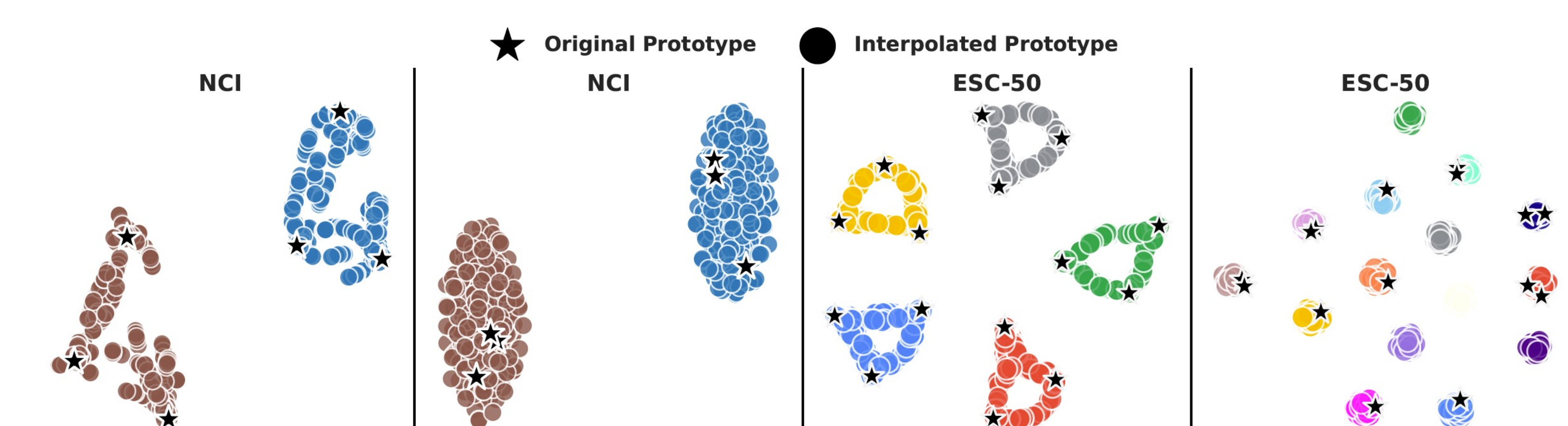
	Chemical			Text	Speech
	Metabolism	Tox21	NCI	GLUE-SciTail	ESC-50
Method	5-shot	5-shot	5-shot	4-shot	5-shot
ProtoNet	$63.62 \pm 0.56\%$	$64.07 \pm 0.80\%$	$80.45 \pm 0.48\%$	$72.59 \pm 0.45\%$	$69.05 \pm 1.48\%$
MetaReg	$66.22 \pm 0.99\%$	$64.40 \pm 0.65\%$	$80.94 \pm 0.34\%$	$72.08 \pm 1.33\%$	$74.95 \pm 1.78\%$
MetaMix	$68.02 \pm 1.57\%$	$65.23 \pm 0.56\%$	$79.46 \pm 0.38\%$	$72.12 \pm 1.04\%$	$71.99 \pm 1.41\%$
MLTI	$65.44 \pm 1.14\%$	$64.16 \pm 0.23\%$	$81.12 \pm 0.70\%$	$71.65 \pm 0.70\%$	$70.62 \pm 1.96\%$
ProtoNet+ST	$66.26 \pm 0.65\%$	$64.98 \pm 1.25\%$	$81.20 \pm 0.30\%$	$72.37 \pm 0.56\%$	$71.54 \pm 1.56\%$
Meta-Interpolation	$\mathbf{72.92} \pm 1.89\%$	$\mathbf{67.54} \pm 0.40\%$	$\mathbf{82.86} \pm 0.26\%$	$\mathbf{73.64} \pm 0.59\%$	$\mathbf{79.22} \pm 0.84\%$

Table 2: Average accuracy of 5 runs and ±95% confidence interval for few shot classification on image domains — Rainbow MNIST, Mini-ImageNet, and CIFAR100. ST stands for Set Transformer.

	RMNIST	Mini-ImageNet-S		CIFAR-100-FS	
Method	1-shot	1-shot	5-shot	1-shot	5-shot
ProtoNet	$75.35 \pm 1.43\%$	$39.14 \pm 0.78\%$	$51.17 \pm 0.57\%$	$38.05 \pm 1.56\%$	$52.63 \pm 0.74\%$
MetaReg	$76.40 \pm 0.56\%$	$39.36 \pm 0.45\%$	$50.94 \pm 0.67\%$	$37.74 \pm 0.70\%$	$52.73 \pm 1.26\%$
MetaMix	$76.54 \pm 0.72\%$	$38.25 \pm 0.09\%$	$52.38 \pm 0.52\%$	$36.13 \pm 0.63\%$	$52.52 \pm 0.89\%$
MLTI	$79.40 \pm 0.75\%$	$39.69 \pm 0.47\%$	$52.73 \pm 0.51\%$	$38.81 \pm 0.55\%$	$53.41 \pm 0.83\%$
ProtoNet+ST	$77.38 \pm 2.05\%$	$38.93 \pm 1.03\%$	$48.92 \pm 0.67\%$	$38.03 \pm 0.85\%$	$50.72 \pm 0.92\%$
Meta Interpolation	$\mathbf{83.24} \pm 1.39\%$	$\mathbf{40.28} \pm 0.48\%$	$\mathbf{53.06} \pm 0.33\%$	$\mathbf{41.48} \pm 0.45\%$	$\mathbf{54.94} \pm 0.80\%$



(a) Train RMNIST (b) Val. RMNIST (c) Train NCI (d) Val. NCI



(a) MLTI (b) Meta-Interp. (c) MLTI (d) Meta-Interp.

Figure 3: Visualization of original and interpolated tasks from NCI ((**a**) and (**b**)) and ESC-50 ((**c**) and (**d**)).