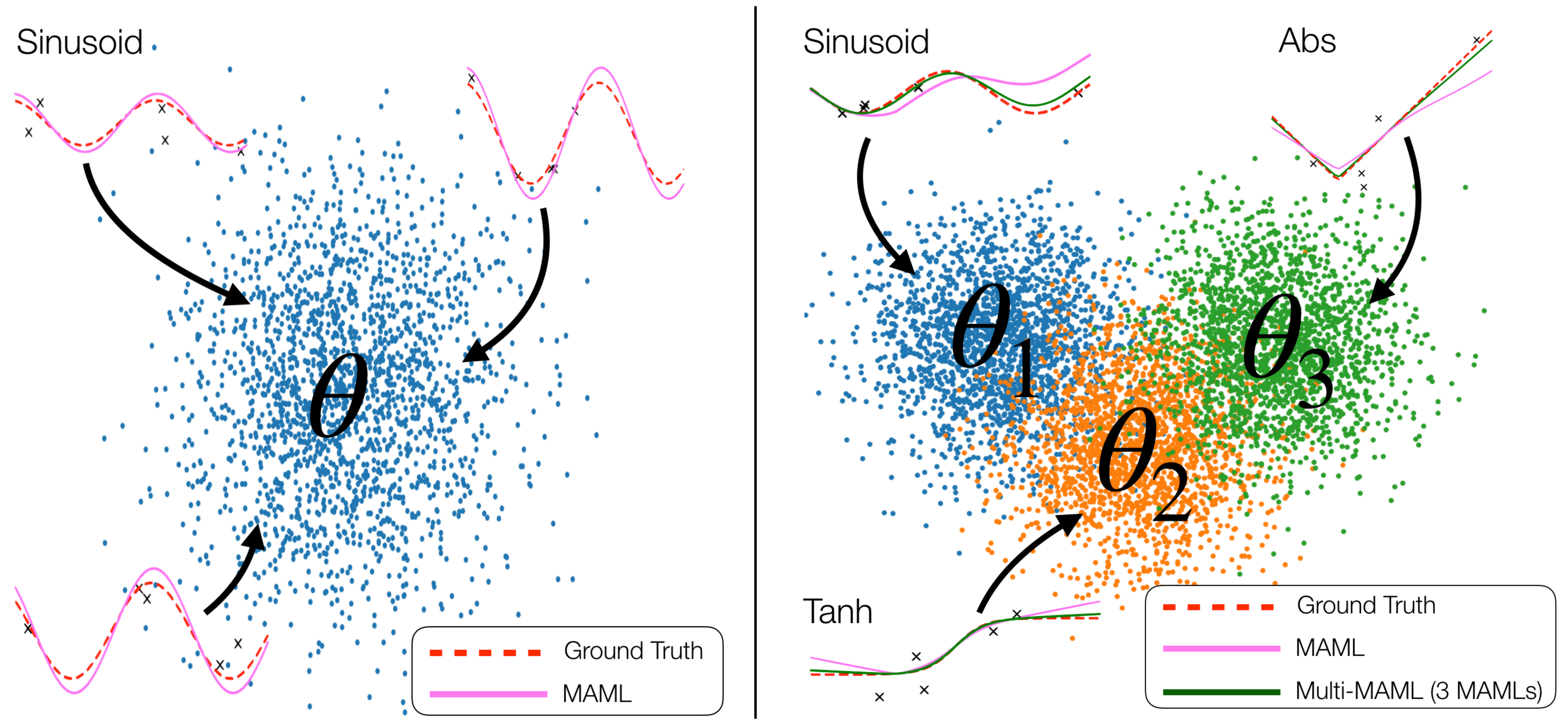
Multimodal Model-Agnostic Meta-Learning via Task-Aware Modulation

University of Michigan · USC

Risto Vuorio* Shao-Hua Sun* Hexiang Hu Joseph J. Lim

Introduction



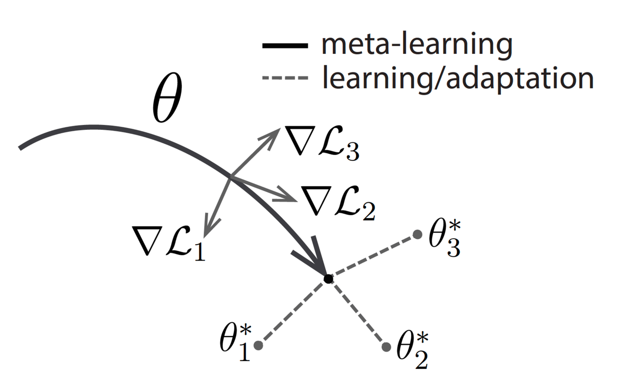
Real-world task distributions are often multimodal
- Have a rich structure (e.g. multiple modes)
- Some knowledge can be transferable across modes/tasks

Model-agnostic meta-learning (MAML) [1]
- Seek a common initialization parameter for all the modes

An ensemble of MAMLs (Multi-MAML)
- Mode labels are often not available
- Prevent sharing related knowledge among modes/tasks

Background

Model-Agnostic Meta-Learning [1]
- Meta-learn a parameter initialization that can be fine-tuned for new tasks in few gradient update steps



Model-Agnostic Meta-Learning Objective

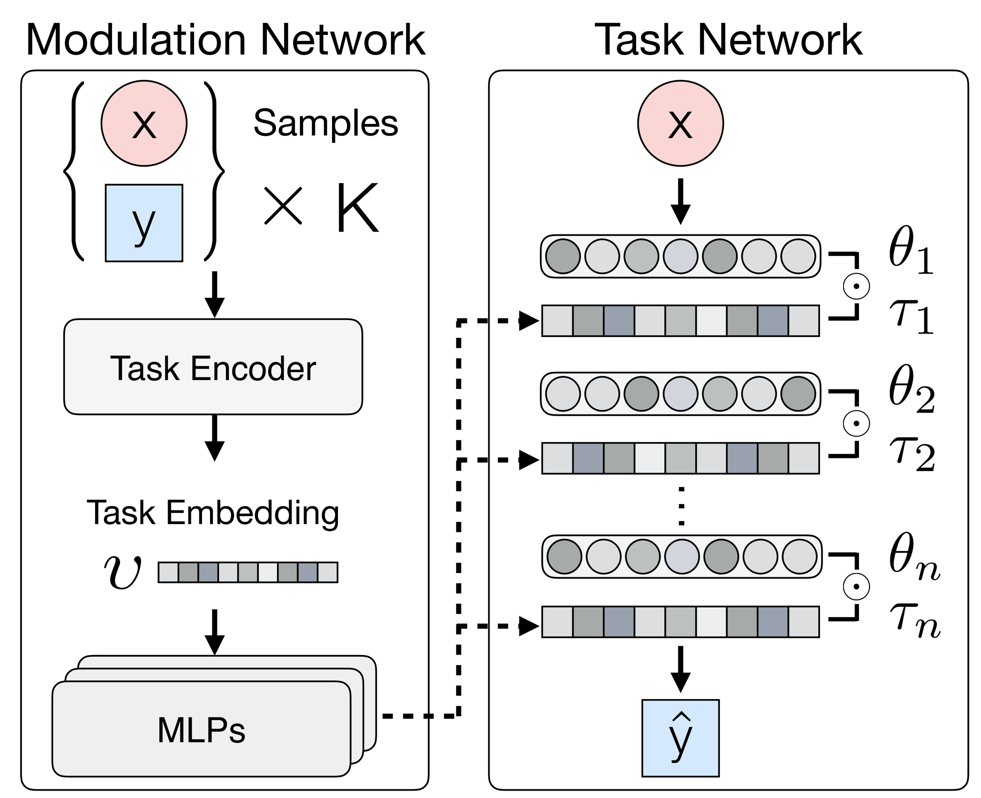
- Inner loop $\theta'_{\mathcal{T}_j} = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_j}(f(x,\theta); \mathcal{D}^{\text{train}}_{\mathcal{T}_j})$

- Outer loop $\theta' = \theta - \beta\nabla_\theta \sum_{\mathcal{T}_j \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_j}(f(x,\theta'_{\mathcal{T}_j}); \mathcal{D}^{\text{val}}_{\mathcal{T}_j})$

[1] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." in International Conference on Machine Learning 2017

Our Approach

Intuition
- **Modulation network**: identify task modes and modulate the initialization accordingly
- **Task network**: further gradient adaptation via MAML steps

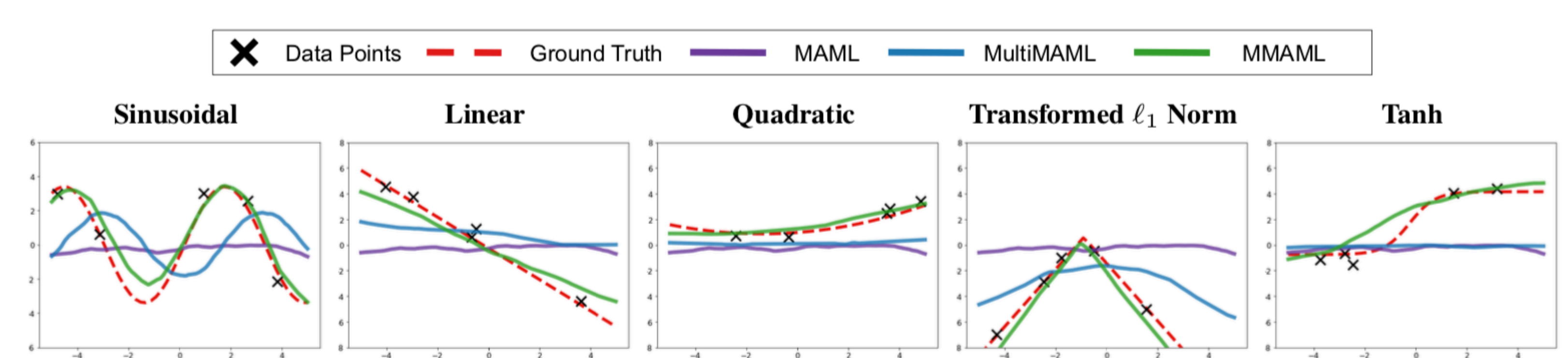


Algorithm 1 MMAML META-TRAINING PROCEDURE.

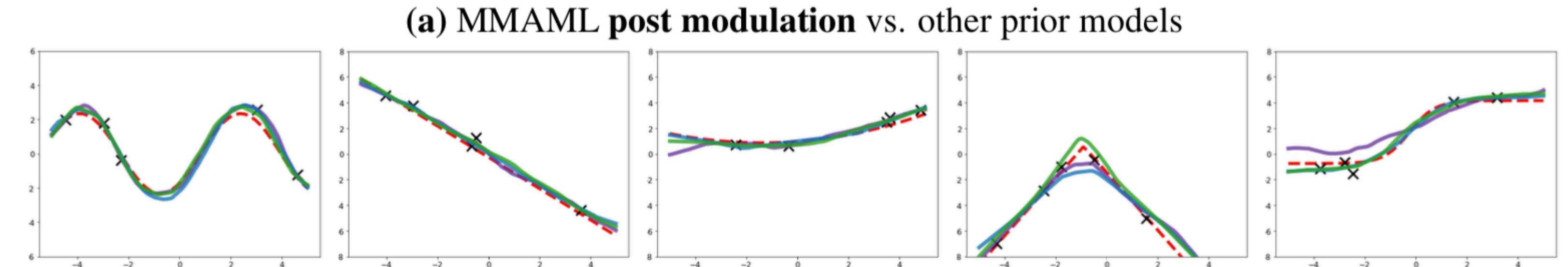
1: **Input:** Task distribution $P(\mathcal{T})$, Hyper-parameters α and β
2: Randomly initialize θ and ω.
3: **while** not DONE **do**
4: Sample batches of tasks $\mathcal{T}_j \sim P(\mathcal{T})$
5: **for all** j **do**
6: Infer $\upsilon = h(\{x,y\}_K; \omega_h)$ with K samples from $\mathcal{D}^{\text{train}}_{\mathcal{T}_j}$.
7: Generate parameters $\tau = \{g_i(\upsilon; \omega_g) \mid i = 1, \cdots, N\}$ to modulate each block of the task network f.
8: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_j}(f(x;\theta,\tau); \mathcal{D}^{\text{train}}_{\mathcal{T}_j})$ w.r.t the K samples
9: Compute adapted parameter with gradient descent: $\theta'_{\mathcal{T}_j} = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_j}(f(x;\theta,\tau); \mathcal{D}^{\text{train}}_{\mathcal{T}_j})$
10: **end for**
11: Update θ with $\beta\nabla_\theta \sum_{T_j \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_j}(f(x;\theta',\tau); \mathcal{D}^{\text{val}}_{\mathcal{T}_j})$
12: Update ω_g with $\beta\nabla_{\omega_g} \sum_{T_j \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_j}(f(x;\theta',\tau); \mathcal{D}^{\text{val}}_{\mathcal{T}_j})$
13: Update ω_h with $\beta\nabla_{\omega_h} \sum_{T_j \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{T}_j}(f(x;\theta',\tau); \mathcal{D}^{\text{val}}_{\mathcal{T}_j})$
14: **end while**

	Parameters
Outer loop	
• **Task Encoder**: produce the task embedding	ω_g
• **MLPs**: modulate the task network blocks	ω_h
Inner loop	
• **Task network**: fast adapt through gradient updates	θ

Experiment - Regression



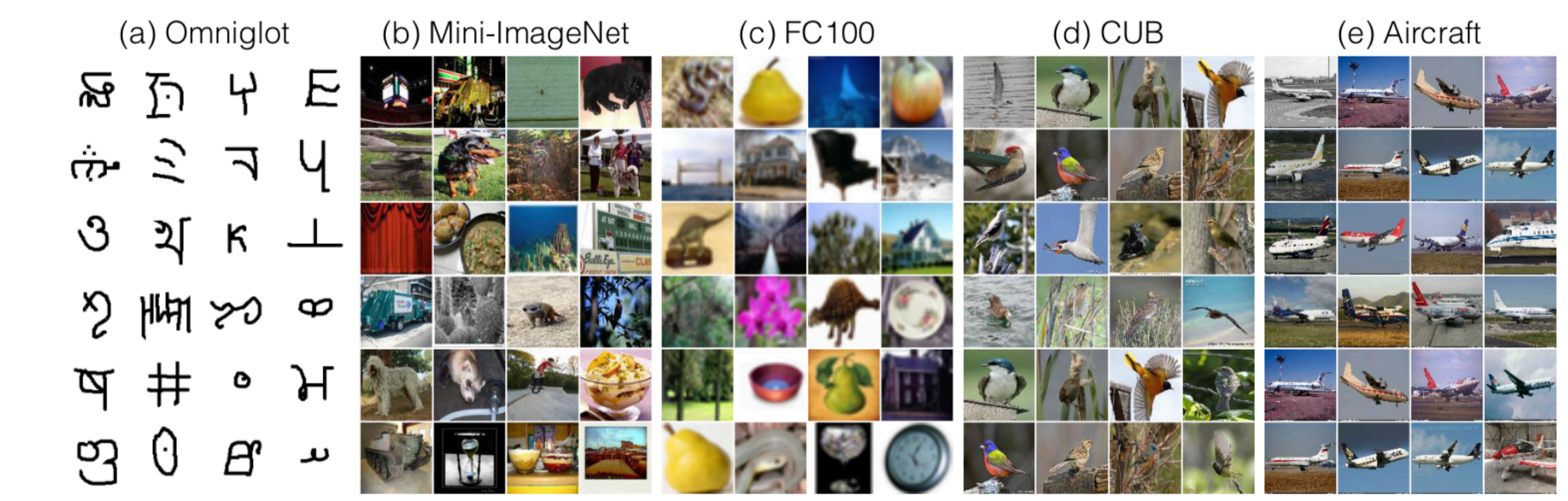
(a) MMAML **post modulation** vs. other prior models



(b) MMAML **post adaptation** vs. other posterior models

Method	2 Modes		3 Modes		5 Modes	
	Post Modulation	Post Adaptation	Post Modulation	Post Adaptation	Post Modulation	Post Adaptation
MAML [8]	-	1.085	-	1.231	-	1.668
Multi-MAML	-	0.433	-	0.713	-	1.082
LSTM Learner	0.362	-	0.548	-	0.898	-
Ours: MMAML (Softmax)	1.548	0.361	2.213	**0.444**	2.421	0.939
Ours: MMAML (FiLM)	2.421	**0.336**	1.923	**0.444**	2.166	**0.868**

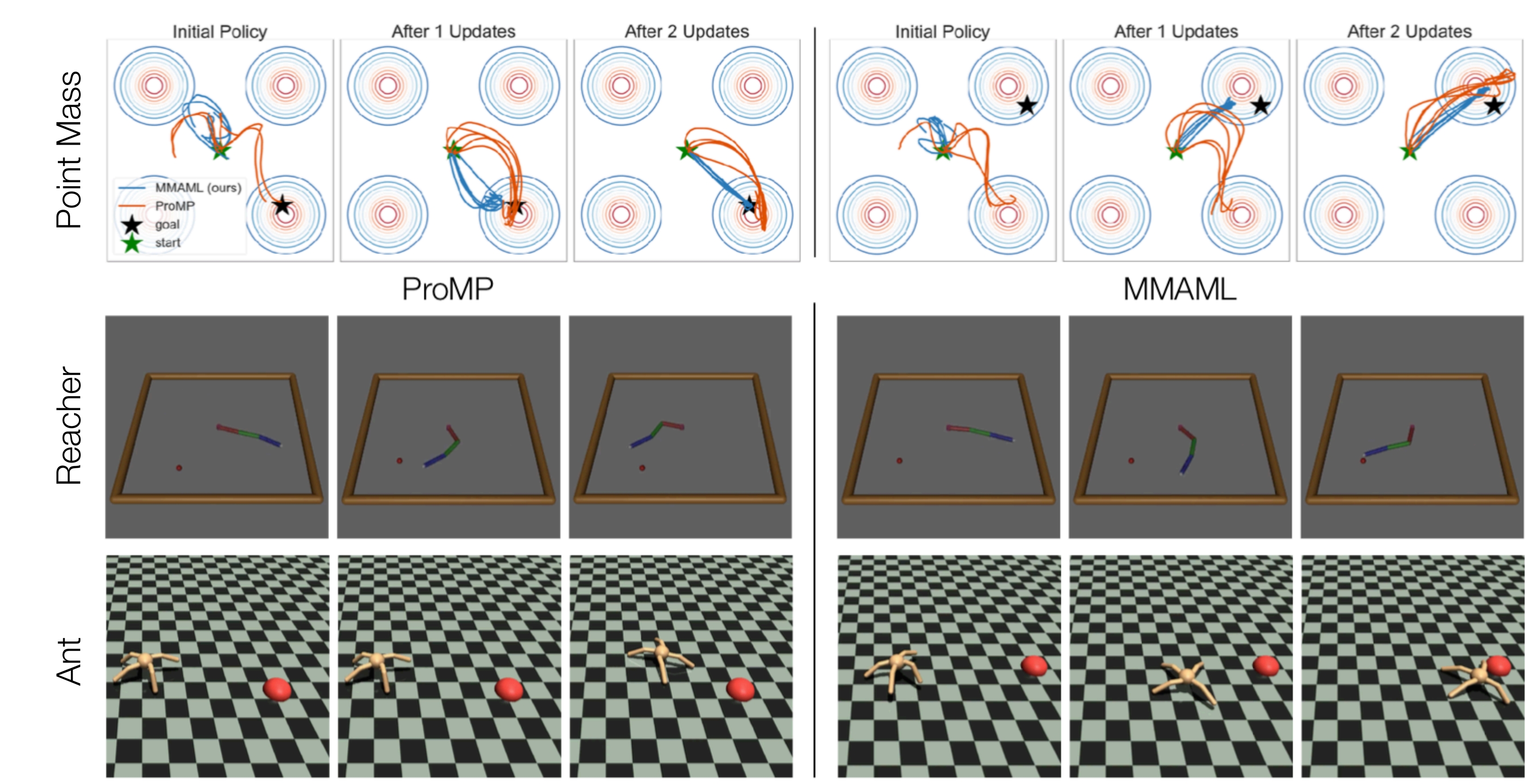
Experiment - Classification



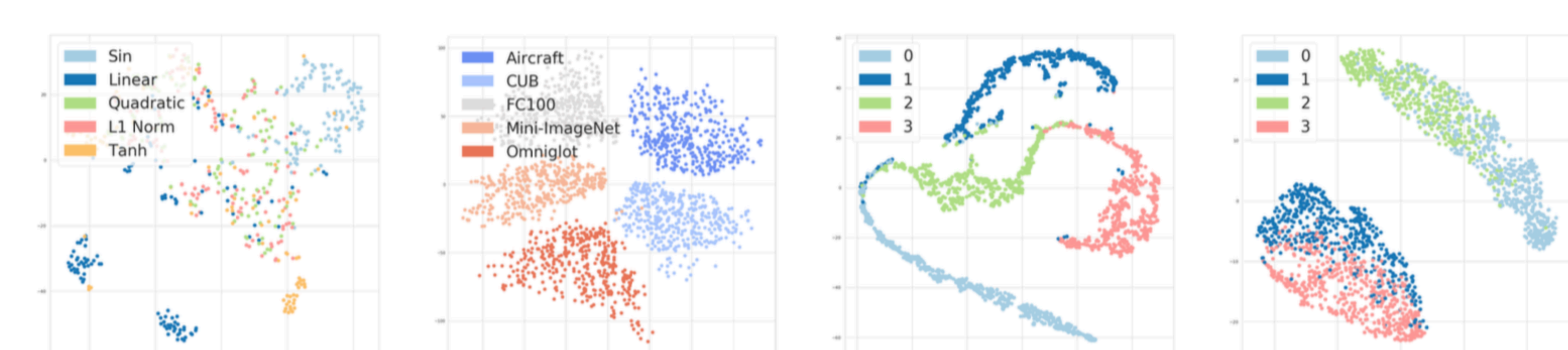
Method & Setup	2 Modes			3 Modes			5 Modes		
Way	5-way		20-way	5-way		20-way	5-way		20-way
Shot	1-shot	5-shot	1-shot	1-shot	5-shot	1-shot	1-shot	5-shot	1-shot
MAML [8]	66.80%	77.79%	44.69%	54.55%	67.97%	28.22%	44.09%	54.41%	28.85%
Multi-MAML	66.85%	73.07%	**53.15%**	55.90%	62.20%	**39.77%**	45.46%	55.92%	33.78%
MMAML (ours)	**69.93%**	**78.73%**	47.80%	**57.47%**	**70.15%**	36.27%	**49.06%**	**60.83%**	**33.97%**

Experiment - Reinforcement Learning

Method	POINT MASS 2D			REACHER			ANT	
	2 Modes	4 Modes	6 Modes	2 Modes	4 Modes	6 Modes	2 Modes	4 Modes
ProMP [42]	-397 ± 20	-523 ± 51	-330 ± 10	-12 ± 2.0	-13.8 ± 2.5	-14.9 ± 2.9	-761 ± 48	-953 ± 46
Multi-ProMP	-109 ± 6	-109 ± 6	-92 ± 4	-4.3 ± 0.1	-4.3 ± 0.1	-4.3 ± 0.1	-624 ± 38	-611 ± 31
Ours	-136 ± 8	-209 ± 32	-169 ± 48	-10.0 ± 1.0	-11.0 ± 0.8	-10.9 ± 1.1	-711 ± 25	-904 ± 37



Experiment - Learned Task Embeddings



(a) Regression (b) Image classification (c) RL Reacher (d) RL Point Mass