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

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

Introduction

Unimodal Task Distribution  
Multimodal Task Distribution

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

Our Approach

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

Algorithm 4 MAMiLE Training Procedure
1. Input: Task distribution $P(T)$, hyperparameters $\alpha$ and $\beta$
2. Randomly initialize $\Theta_0$
3. While update do
4. Sample batches of tasks $T_\ell \sim P(T)$, for all $\ell$
5. Inner loop
6. Iterate $\cdot \equiv \cdot (\theta_0; \omega_h, \omega_g)$ with $k$ samples from $P(T)$
7. Compute parameters for $\omega_h, \omega_g$ as gradient descent:
   $\omega_h \leftarrow \omega_h - \alpha \nabla_{\omega_h} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
8. Update $\omega$ with $\nabla_{\omega_g} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
9. end for
10. Outer loop
11. Task Encoder: produce the task embedding
12. MLPs: modulate the task network blocks
13. Task network: fast adaptation through gradient updates

Parameters
\[ \omega_g, \omega_h \]

Experiment - Classification

Unimodal Task Distribution  
Multimodal Task Distribution

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

Model-Agnostic Meta-Learning 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 $\theta^T_\ell = \theta - \alpha \nabla_\theta \mathcal{L}(f(x, \theta); D_{\text{train}}^T)$
• Outer loop $\theta = \theta - \beta \nabla_\theta \sum_{T_\ell} P(T) \mathcal{L}(f(x, \theta_\ell); D_{\text{val}}^T)$

Method & Setup 2 Modes 5 Modes 20 Modes 5 Modes 20 Modes 5 Modes 20 Modes 5 Modes 20 Modes
Way 1-way 5-way 1-way 5-way 1-way 5-way 1-way 5-way 1-way 5-way
Classifiers MAML (all) 65.80% 77.79% 44.69% 54.55% 67.97% 28.32% 40.90% 54.41% 28.85% Multi-MAML 68.05% 73.07% 55.09% 62.39% 39.74% 41.46% 55.92% 37.79% MAMiLE (ours) 69.59% 78.73% 47.48% 57.48% 70.84% 36.72% 49.60% 68.83% 33.99%

Experiment - Reinforcement Learning

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

Model-Agnostic Meta-Learning 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 $\theta^T_\ell = \theta - \alpha \nabla_\theta \mathcal{L}(f(x, \theta); D_{\text{train}}^T)$
• Outer loop $\theta = \theta - \beta \nabla_\theta \sum_{T_\ell} P(T) \mathcal{L}(f(x, \theta_\ell); D_{\text{val}}^T)$

Algorithm 4 MAMiLE Training Procedure
1. Input: Task distribution $P(T)$, hyperparameters $\alpha$ and $\beta$
2. Randomly initialize $\Theta_0$
3. While update do
4. Sample batches of tasks $T_\ell \sim P(T)$, for all $\ell$
5. Inner loop
6. Iterate $\cdot \equiv \cdot (\theta_0; \omega_h, \omega_g)$ with $k$ samples from $P(T)$
7. Compute parameters for $\omega_h, \omega_g$ as gradient descent:
   $\omega_h \leftarrow \omega_h - \alpha \nabla_{\omega_h} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
8. Update $\omega$ with $\nabla_{\omega_g} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
9. end for
10. Outer loop
11. Task Encoder: produce the task embedding
12. MLPs: modulate the task network blocks
13. Task network: fast adaptation through gradient updates

Parameters
\[ \omega_g, \omega_h \]

Experiment - Regression

Unimodal Task Distribution  
Multimodal Task Distribution

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

Model-Agnostic Meta-Learning 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 $\theta^T_\ell = \theta - \alpha \nabla_\theta \mathcal{L}(f(x, \theta); D_{\text{train}}^T)$
• Outer loop $\theta = \theta - \beta \nabla_\theta \sum_{T_\ell} P(T) \mathcal{L}(f(x, \theta_\ell); D_{\text{val}}^T)$

Algorithm 4 MAMiLE Training Procedure
1. Input: Task distribution $P(T)$, hyperparameters $\alpha$ and $\beta$
2. Randomly initialize $\Theta_0$
3. While update do
4. Sample batches of tasks $T_\ell \sim P(T)$, for all $\ell$
5. Inner loop
6. Iterate $\cdot \equiv \cdot (\theta_0; \omega_h, \omega_g)$ with $k$ samples from $P(T)$
7. Compute parameters for $\omega_h, \omega_g$ as gradient descent:
   $\omega_h \leftarrow \omega_h - \alpha \nabla_{\omega_h} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
8. Update $\omega$ with $\nabla_{\omega_g} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
9. end for
10. Outer loop
11. Task Encoder: produce the task embedding
12. MLPs: modulate the task network blocks
13. Task network: fast adaptation through gradient updates

Parameters
\[ \omega_g, \omega_h \]

Experiment - Learned Task Embeddings

Unimodal Task Distribution  
Multimodal Task Distribution

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

Model-Agnostic Meta-Learning 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 $\theta^T_\ell = \theta - \alpha \nabla_\theta \mathcal{L}(f(x, \theta); D_{\text{train}}^T)$
• Outer loop $\theta = \theta - \beta \nabla_\theta \sum_{T_\ell} P(T) \mathcal{L}(f(x, \theta_\ell); D_{\text{val}}^T)$

Algorithm 4 MAMiLE Training Procedure
1. Input: Task distribution $P(T)$, hyperparameters $\alpha$ and $\beta$
2. Randomly initialize $\Theta_0$
3. While update do
4. Sample batches of tasks $T_\ell \sim P(T)$, for all $\ell$
5. Inner loop
6. Iterate $\cdot \equiv \cdot (\theta_0; \omega_h, \omega_g)$ with $k$ samples from $P(T)$
7. Compute parameters for $\omega_h, \omega_g$ as gradient descent:
   $\omega_h \leftarrow \omega_h - \alpha \nabla_{\omega_h} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
8. Update $\omega$ with $\nabla_{\omega_g} \mathcal{L}(\theta_0; \omega_g, \omega_h, \mathbf{y})$
9. end for
10. Outer loop
11. Task Encoder: produce the task embedding
12. MLPs: modulate the task network blocks
13. Task network: fast adaptation through gradient updates

Parameters
\[ \omega_g, \omega_h \]