Reconciling meta-learning and continual learning with online mixtures of tasks

*Ghassen Jerfel (Duke), *Erin Grant (Berkeley),
Tom Griffiths (Princeton), Katherine Heller (Duke, Google)

(* equal contribution)

Poster #175

Motivation

Motivation

Meta-learning algorithms often assume all tasks are **equally related**.

Motivation

Meta-learning algorithms often assume all tasks are **equally related**.

➔ Heterogeneity: How to exploit the varying degrees of similarity to

- encourage positive transfer between strongly related tasks?
- and avoid negative transfer from distractor tasks?

Motivation

Meta-learning algorithms often assume all tasks are **equally related**.

➔ Heterogeneity: How to exploit the varying degrees of similarity to

- encourage positive transfer between strongly related tasks?
- and avoid negative transfer from distractor tasks?

★ A **general-purpose similarity metric between tasks** is nontrivial for complex models such as neural networks!

Motivation

Motivation

Meta-learning algorithms can be brittle to **changes in the task distribution**, especially without access to previous training data.

Motivation

Meta-learning algorithms can be brittle to **changes in the task distribution**, especially without access to previous training data.

➔ Nonstationarity: How can we both detect and adapt to an evolving distribution over tasks in order to learn to learn without forgetting?

Motivation

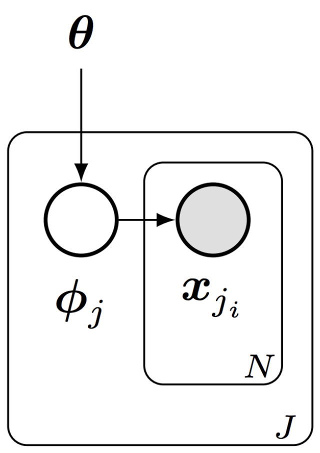
Meta-learning algorithms can be brittle to **changes in the task distribution**, especially without access to previous training data.

➔ Nonstationarity: How can we both detect and adapt to an evolving distribution over tasks in order to learn to learn without forgetting?

★ This is an instance of **task-agnostic continual learning**.

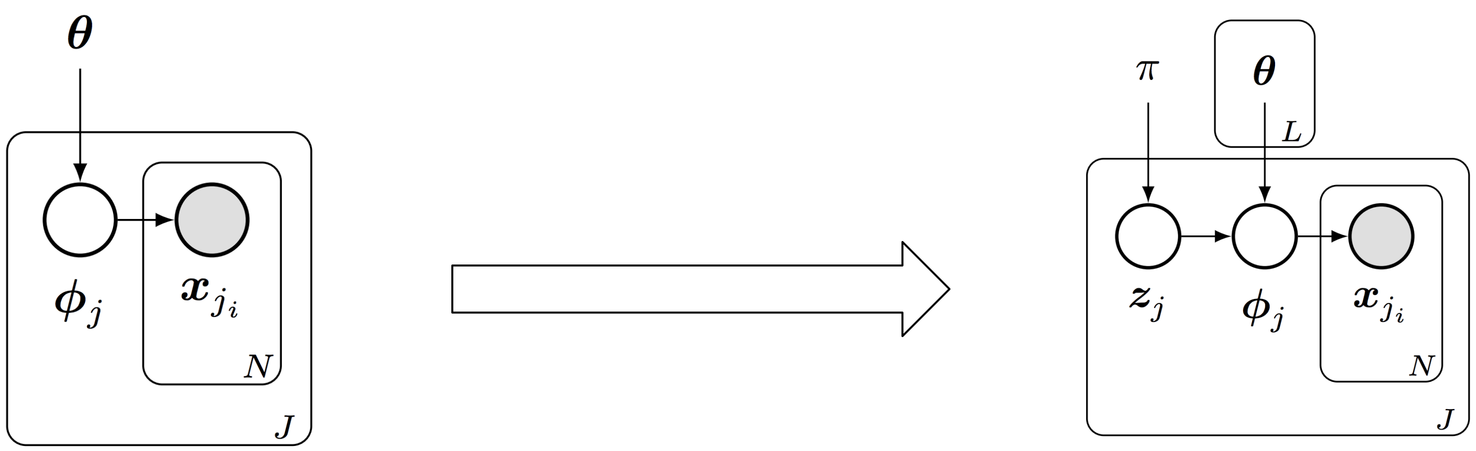
Contributions

hierarchical model of gradient-based meta-learning



Contributions

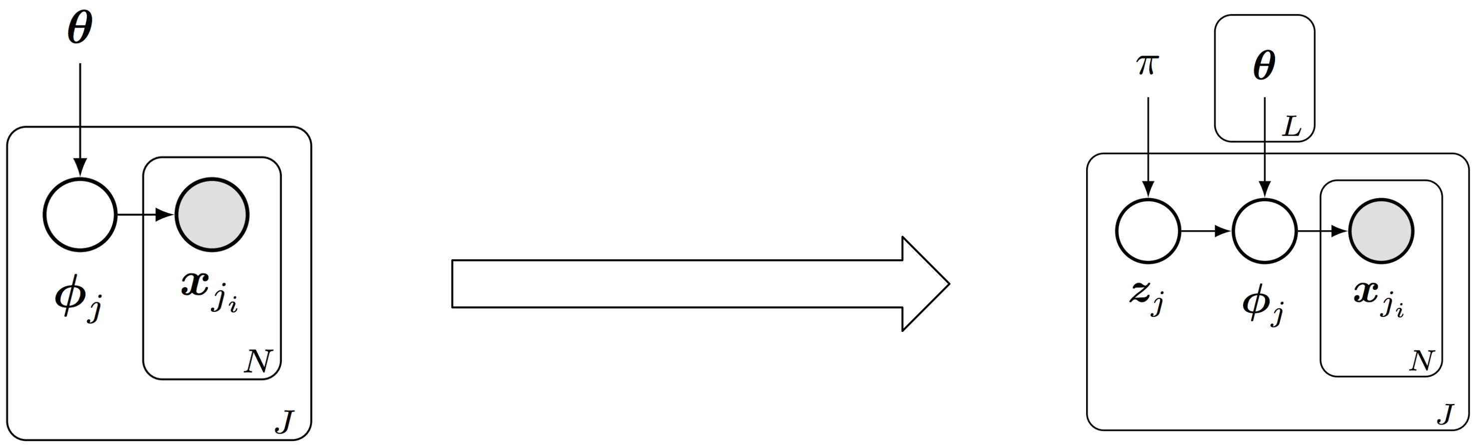
hierarchical model of gradient-based meta-learning



mixture of hierarchical models

Contributions

hierarchical model of gradient-based meta-learning

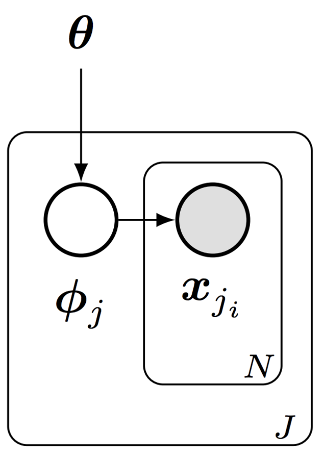


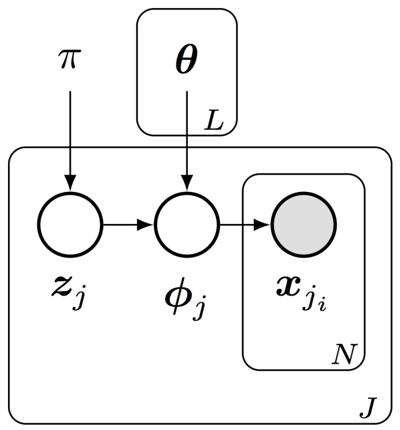
mixture of hierarchical models

➔ Estimation of latent task-specific parameters $\boldsymbol{\varphi}_j$ is performed by gradient-based expectation-maximization.

Contributions

hierarchical model of gradient-based meta-learning





mixture of hierarchical models

➔ Estimation of latent task-specific parameters $\boldsymbol{\varphi_j}$ is performed by gradient-based expectation-maximization.

★ The result is a **scalable** and **architecture-agnostic** algorithm that that **jointly estimates** task-specific cluster assignments and model parameters.

Algorithm

Draw tasks $\mathcal{T}_1, \ldots, \mathcal{T}_J \sim p_{\mathscr{D}}(\mathcal{T})$

for j *in* $1, \ldots, J$ **do**

  Draw task-specific datapoints, $\boldsymbol{x}_{j_1} \ldots \boldsymbol{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\boldsymbol{x})$

  Draw a parameter initialization for a new cluster from the global prior, $\boldsymbol{\theta}^{(L+1)} \sim G_0$

  for ℓ *in* $\{1, \ldots, L, L+1\}$ **do**

    Initialize $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)}$

    Compute task-specific mode estimate, $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \hat{\boldsymbol{\phi}}_j^{(\ell)} + \alpha \sum_k \nabla_{\boldsymbol{\phi}} \log p(\, \boldsymbol{x}_{j_{1:N}} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)} \,)$

  Compute assignment of tasks to clusters, $\gamma_j \leftarrow \texttt{E-STEP}\ (\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)})$

Update each component ℓ in $1, \ldots, L$, $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} + \texttt{M-STEP}\ (\{\boldsymbol{x}_{j_{N+1:N+M}}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j\}_{j=1}^{J})$

Algorithm

Draw tasks $\mathcal{T}_1, \ldots, \mathcal{T}_J \sim p_{\mathscr{D}}(\mathcal{T})$
for j *in* $1, \ldots, J$ **do**
  Draw task-specific datapoints, $\boldsymbol{x}_{j_1} \ldots \boldsymbol{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\boldsymbol{x})$
  Draw a parameter initialization for a new cluster from the global prior, $\boldsymbol{\theta}^{(L+1)} \sim G_0$
  for ℓ *in* $\{1, \ldots, L, L+1\}$ **do**
    Initialize $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)}$
    Compute task-specific mode estimate, $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \hat{\boldsymbol{\phi}}_j^{(\ell)} + \alpha \sum_k \nabla_{\boldsymbol{\phi}} \log p(\, \boldsymbol{x}_{j_{1:N}} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}\,)$
  Compute assignment of tasks to clusters, $\gamma_j \leftarrow \texttt{E-STEP}\ (\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)})$
Update each component ℓ in $1, \ldots, L$, $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} + \texttt{M-STEP}\ (\{\boldsymbol{x}_{j_{N+1:N+M}}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j\}_{j=1}^{J})$

We compute ***L*** sets of fast weights via gradient-based adaptation from each global parameter $\boldsymbol{\theta}^{(\ell)}$.

Algorithm

Draw tasks $\mathcal{T}_1, \ldots, \mathcal{T}_J \sim p_{\mathscr{D}}(\mathcal{T})$
for j *in* $1, \ldots, J$ **do**
  Draw task-specific datapoints, $\boldsymbol{x}_{j_1} \ldots \boldsymbol{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\boldsymbol{x})$
  Draw a parameter initialization for a new cluster from the global prior, $\boldsymbol{\theta}^{(L+1)} \sim G_0$
  for ℓ *in* $\{1, \ldots, L, L+1\}$ **do**
    Initialize $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)}$
    Compute task-specific mode estimate, $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \hat{\boldsymbol{\phi}}_j^{(\ell)} + \alpha \sum_k \nabla_{\boldsymbol{\phi}} \log p(\boldsymbol{x}_{j_{1:N}} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)})$
  Compute assignment of tasks to clusters, $\gamma_j \leftarrow \texttt{E-STEP}\left(\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)}\right)$
Update each component ℓ in $1, \ldots, L$, $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} + \texttt{M-STEP}\left(\{\boldsymbol{x}_{j_{N+1:N+M}}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j\}_{j=1}^{J}\right)$

Based on the training losses for each set of weights, we estimate the task-to-cluster assignment probabilities.

Algorithm

Draw tasks $\mathcal{T}_1, \ldots, \mathcal{T}_J \sim p_{\mathscr{D}}(\mathcal{T})$
for j *in* $1, \ldots, J$ **do**
  Draw task-specific datapoints, $\boldsymbol{x}_{j_1} \ldots \boldsymbol{x}_{j_{N+M}} \sim p_{\mathcal{T}_j}(\boldsymbol{x})$
  Draw a parameter initialization for a new cluster from the global prior, $\boldsymbol{\theta}^{(L+1)} \sim G_0$
  for ℓ *in* $\{1, \ldots, L, L+1\}$ **do**
    Initialize $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)}$
    Compute task-specific mode estimate, $\hat{\boldsymbol{\phi}}_j^{(\ell)} \leftarrow \hat{\boldsymbol{\phi}}_j^{(\ell)} + \alpha \sum_k \nabla_{\boldsymbol{\phi}} \log p(\boldsymbol{x}_{j_{1:N}} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)})$
  Compute assignment of tasks to clusters, $\gamma_j \leftarrow \texttt{E-STEP}\ (\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)})$
Update each component ℓ in $1, \ldots, L$, $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} + \texttt{M-STEP}\ (\{\boldsymbol{x}_{j_{N+1:N+M}}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j\}_{j=1}^{J})$

Finally, we update the global parameters $\boldsymbol{\theta}^{(\ell)}$ with a weighted combination of gradient updates.

EM Subroutines

E-STEP($\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)}$, *concentration* ζ, *threshold* ϵ)

DPMM log-likelihood for all ℓ in $1, \ldots, L$, $\rho_j^{(\ell)} \leftarrow \sum_i \log p(\, \boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}\,) + \log n^{(\ell)}$

DPMM log-likelihood for new component, $\rho_j^{(L+1)} \leftarrow \sum_i \log p(\, \boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(L+1)}\,) + \log \zeta$

DPMM assignments, $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \ldots, \rho_j^{(L+1)})$

if $\gamma_j^{(L+1)} > \epsilon$ **then**

Expand the model by incrementing $L \leftarrow L + 1$

else

Renormalize $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \ldots, \rho_j^{(L)})$

return γ_j

M-STEP($\{\boldsymbol{x}_{j_i}\}_{i=1}^{M}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j$, *concentration* ζ)

return $\beta \, \nabla_{\boldsymbol{\theta}}[\sum_{j,i} \gamma_j \log p(\, \boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}\,) + \log n^{(\ell)}]$

EM Subroutines

E-STEP($\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)}$, *concentration* ζ, *threshold* ϵ)

DPMM log-likelihood for all ℓ in $1, \dots, L$, $\rho_j^{(\ell)} \leftarrow \sum_i \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}) + \log n^{(\ell)}$

DPMM log-likelihood for new component, $\rho_j^{(L+1)} \leftarrow \sum_i \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(L+1)}) + \log \zeta$

DPMM assignments, $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \dots, \rho_j^{(L+1)})$

if $\gamma_j^{(L+1)} > \epsilon$ **then**

  Expand the model by incrementing $L \leftarrow L + 1$

else

  Renormalize $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \dots, \rho_j^{(L)})$

return γ_j

M-STEP($\{\boldsymbol{x}_{j_i}\}_{i=1}^M, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j$, *concentration* ζ)

return $\beta \, \nabla_{\boldsymbol{\theta}}[\sum_{j,i} \gamma_j \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}) + \log n^{(\ell)}]$

Computes soft task-to-cluster assignments $\boldsymbol{\gamma}$ based on a conditional mode estimate of the task-specific parameter $\boldsymbol{\varphi}_j$.

★ Note the **CRP prior penalties** (**log $n^{(\ell)}$** and **log ζ**).

EM Subroutines

E-STEP($\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)}$, *concentration* ζ, *threshold* ϵ)

- DPMM log-likelihood for all ℓ in $1, \ldots, L$, $\rho_j^{(\ell)} \leftarrow \sum_i \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}) + \log n^{(\ell)}$
- DPMM log-likelihood for new component, $\rho_j^{(L+1)} \leftarrow \sum_i \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(L+1)}) + \log \zeta$
- DPMM assignments, $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \ldots, \rho_j^{(L+1)})$
- **if** $\gamma_j^{(L+1)} > \epsilon$ **then**
  - Expand the model by incrementing $L \leftarrow L + 1$
- **else**
  - Renormalize $\gamma_j \leftarrow \tau\text{-softmax}(\rho_j^{(1)}, \ldots, \rho_j^{(L)})$
- **return** γ_j

Computes soft task-to-cluster assignments $\boldsymbol{\gamma}$ based on a conditional mode estimate of the task-specific parameter $\boldsymbol{\varphi}_j$.

★ Note the **CRP prior penalties** (**log $n^{(\ell)}$** and **log ζ**).

M-STEP($\{\boldsymbol{x}_{j_i}\}_{i=1}^{M}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j$, *concentration* ζ)

- **return** $\beta \nabla_{\boldsymbol{\theta}}[\sum_{j,i} \gamma_j \log p(\boldsymbol{x}_{j_i} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)}) + \log n^{(\ell)}]$

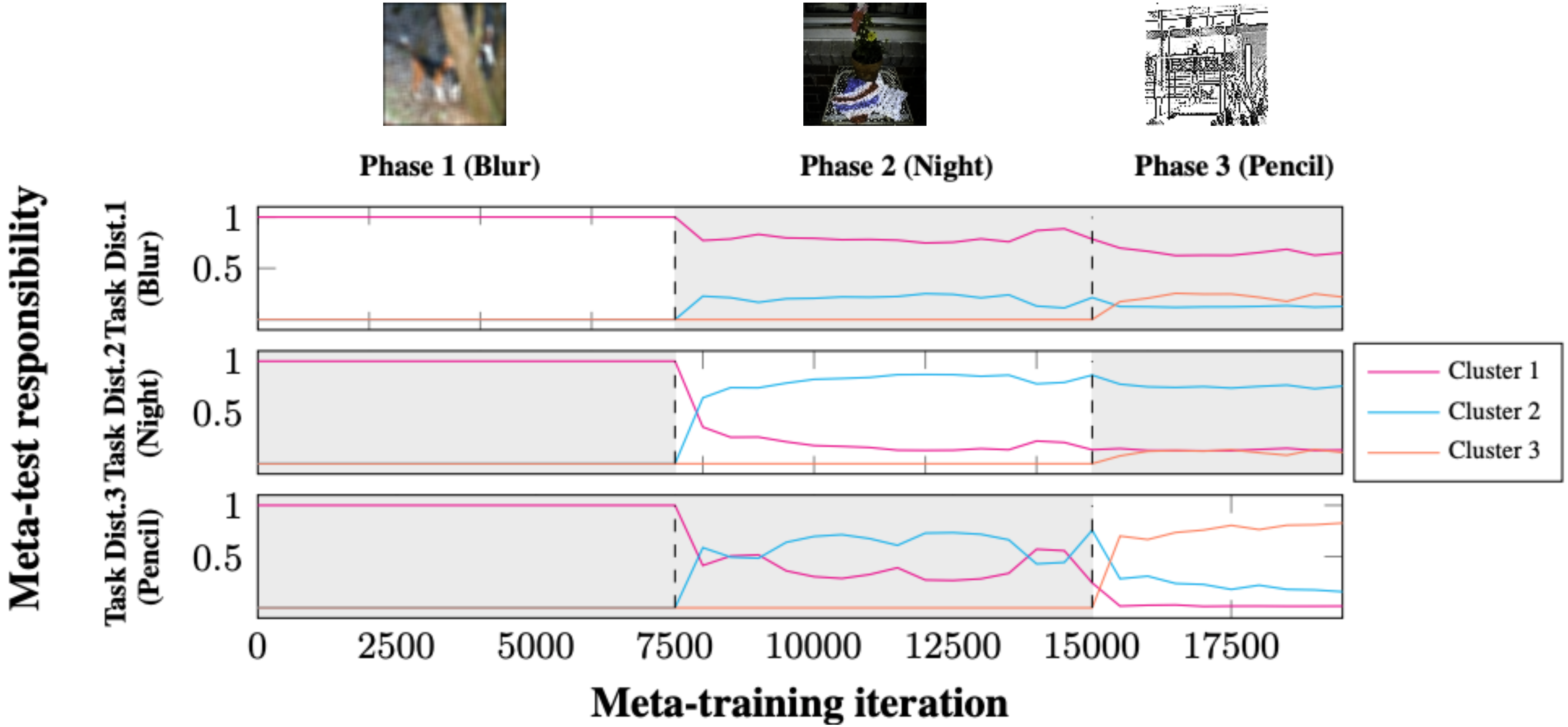
Updates global parameters $\boldsymbol{\theta}^{(\ell)}$ by gradient descent on the task-specific validation loss.

★ This is a **weighted** version of the MAML [Finn 2017] outer loop update.

- ✓ Heterogeneity: Task relatedness can be inferred from the likelihood of assigning each task to a hyperparameter set based on the likelihood after a few steps of gradient-based adaptation to data from a specific task.

- ✓ Non-stationarity: The nonparametric mixture allows for adaptive capacity and change detection, thus alleviating catastrophic forgetting even in the task-agnostic setting (no task boundaries).

Cluster assignments on *stylized miniImageNet*



Above: An evolving dataset of stylized *mini*ImageNet few-shot classification tasks using a sequence of filters; each panel gives task-specific per-cluster responsibilities over time.
Unique cluster (color) has high responsibility for each different type of task (row).

Accuracy on stylized miniImageNet

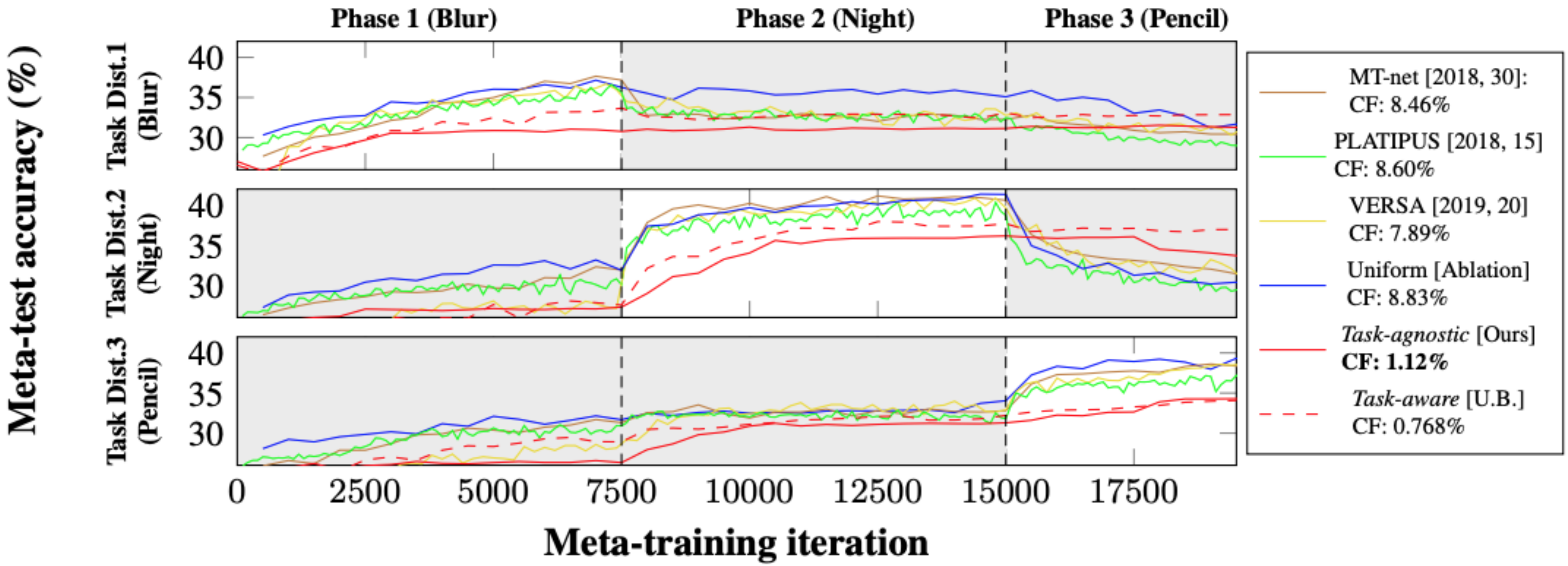


Figure 8: Results on the evolving dataset of filtered *mini*ImageNet few-shot classification tasks (higher is better). Each panel (row) presents, for a specific task type (filter), the average meta-test set accuracy over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (catastrophic forgetting, **CF**) in the legend. This is calculated for each method as the average drop in accuracy on the first two tasks at the end of training (lower is better; U.B.: upper bound).

Summary

- ★ **Task-specific latent structure** regulates transfer in a **heterogeneous** (highly varied) and potentially **non-stationary** (evolving) distribution of tasks, without explicitly modeling task relatedness (*e.g.*, geometrically).
- ★ We **scale Bayesian nonparametrics** to the full set of NN weights with a stochastic point-estimation algorithm in order to **detect distribution shift** and **adapt model capacity**.
- ★ We report **improved accuracy** on the static *mini*ImageNet dataset.
- ★ We report improved performance on a **catastrophic forgetting evaluation** (*i.e.*, accuracy on prior tasks is preserved while learning new tasks).

Poster #175

05:00 -- 07:00 PM

@ East Exhibition Hall B + C