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

Ghassen Jerfel *
gj47@duke.edu
Duke University

Erin Grant *
eringrant@berkeley.edu
UC Berkeley

Thomas L. Griffiths
tomg@princeton.edu
Princeton University

Katherine Heller
kheller@stat.duke.edu
Duke University

Abstract

Learning-to-learn or *meta-learning* leverages data-driven inductive bias to increase the efficiency of learning on a novel task. This approach encounters difficulty when transfer is not advantageous, for instance, when tasks are considerably dissimilar or change over time. We use the connection between gradient-based meta-learning and hierarchical Bayes to propose a Dirichlet process mixture of hierarchical Bayesian models over the parameters of an arbitrary parametric model such as a neural network. In contrast to consolidating inductive biases into a single set of hyperparameters, our approach of task-dependent hyperparameter selection better handles latent distribution shift, as demonstrated on a set of evolving, image-based, few-shot learning benchmarks.

1 Introduction

Meta-learning algorithms aim to increase the efficiency of learning by treating task-specific learning episodes as examples from which to generalize [47]. The central assumption of a meta-learning algorithm is that some tasks are inherently related and so inductive transfer can improve sample efficiency and generalization [9, 8, 5]. In learning a single set of domain-general hyperparameters that parameterize a metric space [53] or an optimizer [40, 14], recent meta-learning algorithms make the assumption that tasks are equally related, and therefore non-adaptive, mutual transfer is appropriate. This assumption has been cemented in recent few-shot learning benchmarks, which comprise a set of tasks generated in a uniform manner [*e.g.,* 53, 14].

However, the real world often presents scenarios in which an agent must decide what degree of transfer is appropriate. In some cases, a subset of tasks are more strongly related to each other, and so non-uniform transfer provides a strategic advantage. On the other hand, transfer in the presence of dissimilar or outlier tasks worsens generalization performance [44, 12]. Moreover, when the underlying task distribution is non-stationary, inductive transfer to previously observed tasks should exhibit graceful degradation to address the catastrophic forgetting problem [28]. In these settings, the consolidation of all inductive biases into a single set of hyperparameters is not well-posed to deal with changing or diverse tasks. In contrast, in order to account for this degree of task heterogeneity, people detect and adapt to novel contexts by attending to relationships between tasks [10].

In this work, we learn a mixture of hierarchical models that allows a meta-learner to adaptively select over a set of learned parameter initializations for gradient-based adaptation to a new task. The method is equivalent to clustering task-specific parameters in the hierarchical model induced by

*Equal contribution.

Preprint. Work in progress.

recasting gradient-based meta-learning as hierarchical Bayes [21] and generalizes the model-agnostic meta-learning (MAML) algorithm introduced in [14]. By treating the assignment of task-specific parameters to clusters as latent variables, we can directly detect similarities between tasks on the basis of the task-specific likelihood, which may be parameterized by a complex model such as a neural network. Our approach therefore alleviates the need for explicit geometric or probabilistic modeling assumptions about the weights of a complex parametric model and provides a scalable method to regulate information transfer between episodes.

We additionally consider the setting of a non-stationary or evolving task distribution, which necessitates a meta-learning method that possesses adaptive complexity. We extend our latent variable model to the non-parametric setting and leverage stochastic point estimation in an infinite mixture [39] over model parameters; point estimation is scalable and requires no distributional assumptions, and is so the online gradient-based mixture approach is compatible with any meta-learning algorithm or neural network architecture that admits gradient-based optimization. We demonstrate the unexplored ability of this combination of non-parametric parameter priors with neural network models to automatically detect and adapt to task distribution shift in a naturalistic image dataset. Our work tackles the non-trivial setting of *task-agnostic* continual learning–where the task change is unobserved–thus addressing an unresolved challenge in *task-aware* continual learning [*e.g.,* 28].

2 Gradient-based meta-learning as hierarchical Bayes

Since our approach is grounded in the probabilistic formulation of meta-learning as hierarchical Bayes [4], our approach can be applied to any probabilistic meta-learner. In this work, we focus on model-agnostic meta-learning (MAML) [14], a gradient-based meta-learning approach that estimates global parameters to be shared among task-specific models as an initialization for a few steps of gradient descent. MAML admits a natural interpretation as parameter estimation in a hierarchical probabilistic model, where the learned initialization acts as data-driven regularization for the estimation of task-specific parameters $\hat{\boldsymbol{\phi}}_j$.

In particular, [21] cast MAML as posterior inference for task-specific parameters $\boldsymbol{\phi}_j$ given some samples of task-specific data $\boldsymbol{x}_{j_{1:N}}$ and a prior over $\boldsymbol{\phi}_j$ that is induced by early stopping of an iterative optimization procedure; truncation at K steps of gradient descent on the negative log-likelihood $-\log p(\, \boldsymbol{x}_{j_{1:N}} \mid \boldsymbol{\phi}_j \,)$ starting from $\boldsymbol{\phi}_{j_{(0)}} = \boldsymbol{\theta}$ can be then understood as mode estimation of the posterior $p(\, \boldsymbol{\phi}_j \mid \boldsymbol{x}_{j_{1:N}}, \boldsymbol{\theta} \,)$. The mode estimates $\hat{\boldsymbol{\phi}}_j = \boldsymbol{\phi}_{j_{(0)}} + \alpha \sum_{k=1}^{K} \nabla_{\boldsymbol{\phi}} \log p(\, \boldsymbol{x}_{j_{1:N}} \mid \boldsymbol{\phi}_{j_{(k-1)}} \,)$ are then combined to evaluate the marginal likelihood for each task as

$$p\left(\, \boldsymbol{x}_{j_{N+1:N+M}} \mid \boldsymbol{\theta} \,\right) = \int p\big(\boldsymbol{x}_{j_{N+1:N+M}} \mid \boldsymbol{\phi}_j\big) p\big(\boldsymbol{\phi}_j \mid \boldsymbol{\theta}\big) \, \mathrm{d}\boldsymbol{\phi}_j \approx p\big(\boldsymbol{x}_{j_{N+1:N+M}} \mid \hat{\boldsymbol{\phi}}_j\big) \,, \tag{1}$$

where $\boldsymbol{x}_{j_{N+1:N+M}}$ is another set of samples from the jth task. A training dataset can then be summarized in an empirical Bayes point estimate of $\boldsymbol{\theta}$ computed by gradient-based optimization of the joint marginal likelihood in (1) in across tasks, so that the likelihood of a datapoint sampled from a new task can be computed using only $\boldsymbol{\theta}$ and without storing the task-specific parameters.

3 Improving meta-learning by modeling latent task structure

If the task distribution is heterogeneous, assuming a single parameter initialization $\boldsymbol{\theta}$ for gradient-based meta-learning is not suitable because it is unlikely that the point estimate computed by a few steps of gradient descent will sufficiently adapt the task-specific parameters $\boldsymbol{\phi}$ to a diversity of tasks. Moreover, explicitly estimating relatedness between tasks has the potential to aid the efficacy of a meta-learning algorithm by modulating both positive and negative transfer [52, 59, 45, 61], and by identifying outlier tasks that require a more significant degree of adaptation [56, 23]. Nonetheless, defining an appropriate notion of task relatedness is a difficult problem in the high-dimensional parameter or activation space of models such as neural networks.

Using the probabilistic interpretation of Section 2, we deal with the variability in the tasks by assuming that each set of task-specific parameters $\boldsymbol{\phi}_j$ is drawn from a mixture of base distributions each of which is parameterized by a hyperparameter $\boldsymbol{\theta}^{(\ell)}$. Accordingly, we capture task relatedness by estimating the likelihood of assigning each task to a mixture component based simply on the task negative log likelihood after a few steps of gradient-based adaptation. The result is a scalable meta-learning algorithm that jointly learns task-specific cluster assignments and model parameters,

Algorithm 1 `Stochastic gradient-based EM for finite and infinite mixtures(`
dataset $\mathscr{D}$, meta-learning rate β, adaptation rate α, temperature τ, initial cluster count L_0, meta-batch size J, training batch size N, validation batch size M, adaptation iteration count K, global prior G_0)

Initialize cluster count $L \leftarrow L_0$ and meta-level parameters $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(L)} \sim G_0$
while *not converged* **do**
  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, $\gamma_j \leftarrow$ `E-STEP` $(\boldsymbol{x}_{j_{1:N}}, \hat{\boldsymbol{\phi}}_j^{(1:L)})$
  Update each component ℓ in $1, \ldots, L$, $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} +$ `M-STEP` $(\{\boldsymbol{x}_{j_{N+1:N+M}}, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j\}_{j=1}^J)$
  Summarize $\{\boldsymbol{\theta}_1, \ldots\}$ to update global prior G_0
return $\{\boldsymbol{\theta}^{(1)}, \ldots\}$

`E-STEP(` $\{\boldsymbol{x}_{j_i}\}_{i=1}^N, \{\hat{\boldsymbol{\phi}}_j^{(\ell)}\}_{\ell=1}^L$`)`
  return τ-softmax$_\ell(\sum_i \log p(\boldsymbol{x}_{j_i} | \hat{\boldsymbol{\phi}}_j^{(\ell)}))$

`M-STEP(`$\{\boldsymbol{x}_{j_i}\}_{i=1}^M, \hat{\boldsymbol{\phi}}_j^{(\ell)}, \gamma_j$`)`
  return $\beta \nabla_{\boldsymbol{\theta}} [\sum_{j,i} \gamma_j \log p(\boldsymbol{x}_{j_i} | \hat{\boldsymbol{\phi}}_j^{(\ell)})]$

Top: **Algorithm 1:** Stochastic gradient-based expectation maximization (EM) for probabilistic clustering of task-specific parameters in a meta-learning setting. Bottom: **Subroutine 2:** The `E-STEP` and `M-STEP` for a finite mixture of hierarchical Bayesian models implemented as gradient-based meta-learners.

and is capable of modulating the transfer of information across tasks by clustering together related task-specific parameter settings.

Formally, let $\boldsymbol{z}_j$ be the categorical latent variable indicating the cluster assignment of each task-specific parameter $\boldsymbol{\phi}_j$. A direct maximization of the mixture model likelihood is a gubernatorial optimization problem that can grow intractable. This intractability is equally problematic for the posterior distribution over the cluster assignment variables $\boldsymbol{z}_j$ and the task-specific parameters $\boldsymbol{\phi}_j$, which are both treated as latent variables in the probabilistic formulation of meta-learning. A scalable approximation involves representing the conditional distribution for each latent variable with a *maximum a posteriori* (MAP) estimate. In our meta-learning setting of a mixture of hierarchical Bayes (HB) models, this suggests an augmented expectation maximization (EM) procedure [13] alternating between an `E-STEP` that computes an expectation of the task-to-cluster assignments $\boldsymbol{z}_j$, which itself involves the computation of a conditional mode estimate for the task-specific parameters $\boldsymbol{\phi}_j$, and an `M-STEP` that updates the hyperparameters $\theta^{(1:L)}$ (see Subroutine 2).

To ensure scalability, we use the minibatch variant of stochastic optimization [43] in both the `E-STEP` and the `M-STEP`; such approaches to EM are motivated by a view of the algorithm as optimizing a single free energy at both the `E-STEP` and the `M-STEP` [37]. In particular, for each task j and cluster ℓ, we follow the gradients to minimize the negative log-likelihood on the training data points $\boldsymbol{x}_{j_{1:N}}$, using the cluster parameters $\boldsymbol{\theta}^{(\ell)}$ as initialization. This allows us to obtain a modal point estimate of the task-specific parameters $\hat{\boldsymbol{\phi}}_j^{(\ell)}$. The `E-STEP` in Subroutine 2 leverages the connection between gradient-based meta-learning and HB [21] and the differentiability of our clustering procedure to employ the task-specific parameters to compute the posterior probability of cluster assignment. Accordingly, based on the likelihood of the same training data points under the model parameterized by $\hat{\boldsymbol{\phi}}_j^{(\ell)}$, we compute the cluster assignment probabilities as

$$\gamma_j^{(\ell)} := p\big(\boldsymbol{z}_j = \ell \mid \boldsymbol{x}_{j_{1:N}}, \boldsymbol{\theta}^{(1:L)}\big) \propto \int p(\boldsymbol{x}_{j_{1:N}} \mid \boldsymbol{\phi}_j)\, p(\boldsymbol{\phi}_j \mid \boldsymbol{\theta}^{(\ell)})\, \mathrm{d}\boldsymbol{\phi}_j \approx p(\boldsymbol{x}_{j_{1:N}} \mid \hat{\boldsymbol{\phi}}_j^{(\ell)})\,. \quad (2)$$

The cluster means $\boldsymbol{\theta}^{(\ell)}$ are then updated by gradient descent on the validation loss in the `M-STEP` in Subroutine 2; this `M-STEP` is analogous to the MAML algorithm in [14] with the addition of mixing weights $\gamma_j^{(\ell)}$.

Note that, unlike other recent approaches to probabilistic clustering [*e.g.,* 3] we adhere to the episodic meta-learning setup for both training and testing since only the task support set $\boldsymbol{x}_{j_{1:N}}$ is used to compute both the point estimate $\hat{\boldsymbol{\phi}}_j^{(\ell)}$ and the cluster responsibilities $\gamma_j^{(\ell)}$. See Algorithm 1 for the

Table 1: Meta-test set accuracy on the *mini*ImageNet **5-way**, **1-** and **5-shot** classification benchmarks from [53] among methods using a comparable architecture (the 4-layer convolutional network from [53]). For methods on which we report results in later experiments, we additionally report the total number of parameters optimized by the meta-learning algorithm. [a] Results reported by [40]. [b] We report test accuracy for models matching train and test "shot" and "way". [c] We report test accuracy for a comparable base (task-specific network) architecture.

Model	Num. param.	1-shot (%)	5-shot (%)
matching network [53][a]		43.56 ± 0.84	55.31 ± 0.73
meta-learner LSTM [40]		43.44 ± 0.77	60.60 ± 0.71
prototypical networks [49][b]		46.61 ± 0.78	65.77 ± 0.70
MAML [14]		48.70 ± 1.84	63.11 ± 0.92
MT-net [30]	$38,907$	51.70 ± 1.84	
PLATIPUS [15]	$65,546$	50.13 ± 1.86	
VERSA [20][c]	$807,938$	48.53 ± 1.84	
Our method: 2 components	$65,546$	49.60 ± 1.50	64.60 ± 0.92
3 components	$98,319$	51.20 ± 1.52	65.00 ± 0.96
4 components	$131,092$	50.49 ± 1.46	64.78 ± 1.43
5 components	$163,865$	51.46 ± 1.68	

full algorithm, whose high-level structure is shared with the non-parametric variant of our method detailed in Section 5.

4 Experiment: *mini*ImageNet few-shot classification

Clustering task-specific parameters provides a way for a meta-learner to deal with task heterogeneity as each cluster can be associated with a subset of the tasks that would benefit most from mutual transfer. While we do not expect existing tasks to present a significant degree of heterogeneity given the uniform sampling assumptions behind their design, we nevertheless conduct an experiment to validate that our method gives an improvement on a standard benchmark for few-shot learning.

We apply Algorithm 3 with Subroutine 2 and $L \in \{2, 3, 4, 5\}$ components to the 1-shot and 5-shot, 5-way, *mini*ImageNet few-shot classification benchmarks [53]; Appendix C.2.1 contains additional experimental details. We demonstrate in Table 1 that a mixture of meta-learners improves the performance of gradient-based meta-learning on this task for any number of components. However, the performance of the parametric mixture does not improve monotonically with the number of components L. This leads us to the development of non-parametric clustering for continual meta-learning, where enforcing specialization to subgroups of tasks and increasing model complexity is in fact necessary to preserve performance on prior tasks due to significant heterogeneity.

5 Scalable online mixtures for task-agnostic continual learning

The mixture of meta-learners developed in Section 3 addresses a drawback of meta-learning approaches such as MAML that consolidate task-general information into a single set of hyperparameters. However, the method adds another dimension to model selection in the form of identifying the correct number of mixture components. While this may be resolved by cross-validation if the dataset is static and therefore the number of components can remain fixed, adhering to a fixed number of components throughout training is not appropriate in the non-stationary regime, where the underlying task distribution changes as different types of tasks are presented sequentially in a continual learning setting. In this regime, it is important to incrementally introduce more components that can each specialize to the distribution of tasks observed at the time of spawning.

To address this, we derive a scalable stochastic estimation procedure to compute the expectation of task-to-cluster assignments (`E-STEP`) for a growing number of task clusters in a *non-parametric* mixture model [39] called the Dirichlet process mixture model (DPMM). The formulation of the Dirichlet process mixture model (DPMM) that is most appropriate for incremental learning is the sequential draws formulation that corresponds to an instantiation of the Chinese restaurant process (CRP) [39]. A CRP prior over $\boldsymbol{z}_j$ allows some probability to be assigned to a new mixture component while the task identities are inferred in a sequential manner, and has therefore been key to recent online and stochastic learning of the DPMM [31]. A draw from a CRP proceeds as follows: For a sequence of tasks, the first task is assigned to the first cluster and the jth subsequent task is then

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

Subroutine 3: The E-STEP and M-STEP for an infinite mixture of hierarchical Bayesian models.

assigned to the ℓth cluster with probability

$$p\left(\boldsymbol{z}_j = \ell \mid \boldsymbol{z}_{1:j-1}, \zeta\right) = \begin{cases} n^{(\ell)}/n + \zeta & \text{for } \ell \leq L \\ \zeta/n + \zeta & \text{for } \ell = L + 1\,, \end{cases} \tag{3}$$

where L is the number of non-empty clusters, $n^{(\ell)}$ is the number of tasks already occupying a cluster ℓ, and ζ is a fixed positive concentration parameter. The prior probability associated with a new mixture component is therefore $p(\boldsymbol{z}_j = L + 1 \mid \boldsymbol{z}_{1:j-1}, \zeta)$.

In a similar spirit to Section 3, we develop a stochastic EM procedure for the estimation of the latent task-specific parameters $\boldsymbol{\phi}_{1:J}$ and the meta-level parameters $\boldsymbol{\theta}^{(1:L)}$ in the DPMM, which allows the number of observed task clusters to grow in an online manner with the diversity of the task distribution. While computation of the mode estimate of the task-specific parameters $\boldsymbol{\phi}_j$ is mostly unchanged from the finite variant, the estimation of the cluster assignment variables $\boldsymbol{z}$ in the E-STEP requires revisiting the Gibbs conditional distributions due to the potential addition of a new cluster at each step. For a DPMM, the conditional distributions for $\boldsymbol{z}_j$ are

$$p\left(\boldsymbol{z}_j = \ell \mid \boldsymbol{x}_{j_{1:M}}, \boldsymbol{z}_{1:j-1}\right) \propto \begin{cases} n^{(\ell)} \int p(\boldsymbol{x}_{j_{1:M}} \mid \boldsymbol{\phi}_j^{(\ell)}) p(\boldsymbol{\phi}_j^{(\ell)} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi}_j \, \mathrm{d}G_\ell(\boldsymbol{\theta}) & \text{for } \ell \leq L \\ \zeta \int p(\boldsymbol{x}_{j_{1:M}} \mid \boldsymbol{\phi}_j^{(0)}) p(\boldsymbol{\phi}_j^{(0)} \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\phi}_j \, \mathrm{d}G_0(\boldsymbol{\theta}) & \text{for } \ell = L + 1 \end{cases} \tag{4}$$

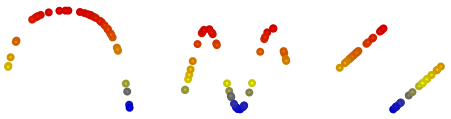
with G_0 as the base measure or global prior over the components of the CRP, G_ℓ is the prior over each cluster's parameters, initialized with a draw from a Gaussian centered at G_0 with a fixed variance and updated over time using summary statistics from the set of active components $\{\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(L)}\}$.

Taking the logarithm of the posterior over task-to-cluster assignments $\boldsymbol{z}_j$ in (4) and using a mode estimate $\hat{\boldsymbol{\phi}}_j^{(\ell)}$ for task-specific parameters $\boldsymbol{\phi}_j$ as drawn from the ℓth cluster gives the E-STEP in Subroutine 3. We may also omit the prior term $\log p(\hat{\boldsymbol{\phi}}_j^{(\ell)} \mid \boldsymbol{\theta}^{(\ell)})$ as it arises as an implicit prior resulting from truncated gradient descent, as explained in Section 3 of [21].

6 Experiments: *Task-agnostic* continual few-shot regression & classification

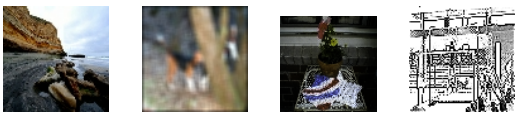
By treating the assignment of tasks to clusters as latent variables, the algorithm of Section 5 can adapt to a changing distribution of tasks, without any external information to signal distribution shift (*i.e.,* in a *task-agnostic* manner). Here, we present our main experimental results on both a novel synthetic regression benchmark as well as a novel evolving variant of *mini*ImageNet, and confirm the algorithm's ability to adapt to distribution shift by spawning a newly specialized cluster.

High-capacity baselines. As an ablation, we compare to the **non-uniform** parameteric **mixture** proposed in Section 3 with the number of components fixed at the total number of task distributions in the dataset (3). We also consider a **uniform** parametric **mixture** in which each component receives equal assignments; this can also be seen as the non-uniform mixture in the infinite temperature (τ) limit. Note that our meta-learner has a lower capacity than these two baselines for most of the training procedure, as it may decide to expand its capacity past one component only when the task



(a) polynomial **(b)** sinusoid **(c)** sawtooth

Figure 4: The diverse set of periodic functions used for few-shot regression in Section 6.1.



(a) plain **(b)** blur **(c)** night **(d)** pencil

Figure 5: Artistic filters **(b-d)** applied to *mini*ImageNet **(a)** to ensure non-homogeneity of tasks in Section 6.2.

distribution changes. Finally, for the large-scale experiment in Section 6.2, we compare with three recent meta-learning algorithms that report improved performance on the standard *mini*ImageNet benchmark of Section 3, but are not explicitly posed to address the continual learning setting of evolving tasks: **MT-net** [30], **PLATIPUS** [15], and **VERSA** [20].

6.1 Continual few-shot regression

We first consider an explanatory experiment in which three regression tasks are presented sequentially with no overlap. For input $\boldsymbol{x}$ sampled uniformly from $[-5, 5]$, each regression task is generated, in a similar spirit to the sinusoidal regression setup in [14], from one of a set of simple but distinct one-dimensional functions (polynomial Figure 4a, sinusoid wave Figure 4b, and sawtooth wave Figure 4c). For the experiment in Figure 6 and Figure 7, we presented the polynomial tasks for 4000 iterations, followed by sinusoid tasks for 3000 iterations, and finally sawtooth tasks. Additional details on the experimental setup can be found in Appendix C.2.2.

Results: Distribution shift detection. The cluster responsibilities in Figure 7 on the meta-test dataset of tasks, from each of the three regression types in Figure 4, indicates that the non-parametric algorithm recognizes a change in the task distribution and spawns a new cluster at iterations 4000 and a bit after 7000. Each newly created cluster is specialized to the task distribution observed at the time of spawning and remains as such throughout training, since the majority of assignments for each type of regression remains under a given cluster from the time of its introduction.

Results: Improved generalization and slower degradation of performance. We investigate the progression of the meta-test mean-squared error (MSE) for the three regression task distributions in Figure 6. We first note the clear advantage of non-uniform assignment both in improved generalization, when testing on the active task distribution, and in slower degradation, when testing on previous distributions. This is due to the ability of these methods to modulate the transfer of information in order to limit negative transfer. In contrast, the uniform method cannot selectively adapt specific clusters to be responsible for any given task, and thus inevitably suffers from catastrophic forgetting.

The adaptive capacity of our non-parametric method allows it to spawn clusters that specialize to newly observed tasks. Accordingly, even if the overall capacity is lower than that of the comparable non-uniform parametric method, our method achieves similar or better generalization, at any given training iteration. More importantly, specialization allows our method to better modulate information transfer as the clusters are better differentiated. Consequently, each cluster does not account for many assignments from more than a single task distribution, throughout training. Therefore, we observed a significantly slower rate of degradation of the MSE on previous task distributions as new tasks are introduced. This is especially evident from the performance on the first task in Figure 6.

6.2 Continual few-shot classification

Next, we consider an evolving variant of the *mini*ImageNet few-shot classification task. In this variant, one of a set of artistic filters are applied to the images during the meta-training procedure to simulate a changing distribution of few-shot classification tasks. For the experiment in Figure 8 and Figure 9 we first train using images with a "blur" filter (Figure 5b) for 7500 iterations, then with a "night" filter (Figure 5c) for another 7500 iterations, and finally with a "pencil" filter (Figure 5d). Additional details on the experimental setup can be found in Appendix C.2.3.

Results: Meta-test accuracy. In Figure 9, we report the evolution of the meta-test accuracy for two variants of our non-parametric meta-learner in comparison to the parametric baselines introduced

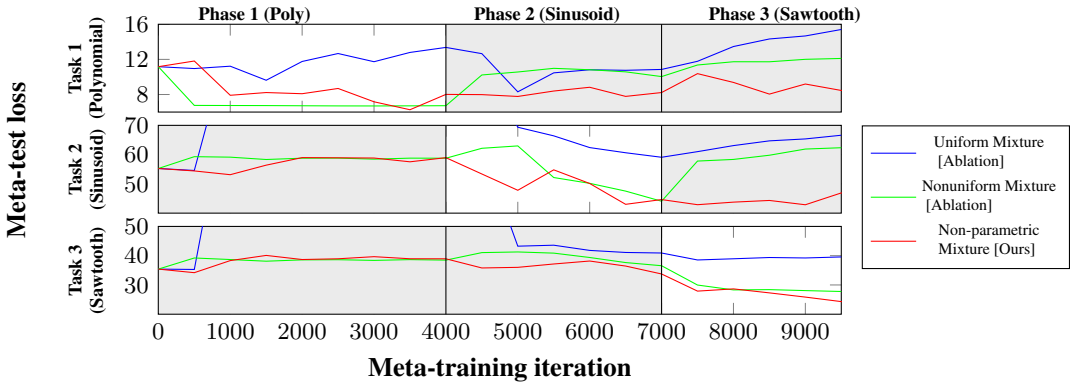


Figure 6: Results on the evolving dataset of few-shot regression tasks (lower is better). Each panel (row) presents, for a specific task type (polynomial, sinusoid or sawtooth), the average meta-test set accuracy of each method over cumulative number of few-shot episodes. We additionally report the degree of loss in backward transfer (i.e., catastrophic forgetting) to the tasks in each meta-test set in the legend; all methods but the non-parametric method experience a large degree of catastrophic forgetting during an inactive phase.

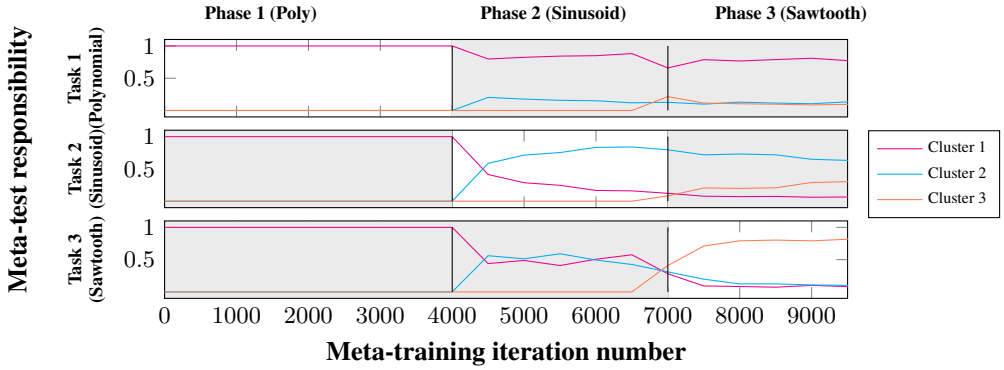


Figure 7: Each panel (row) presents task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ over time. A higher responsibility entails a higher degree of specialization of a particular cluster (color) to a particular task (row).

in Section 6, *high-capacity baselines*. The *task-agnostic* variant is the core algorithm described in previous sections, as used for the regression tasks. The *task-aware* variant augments the core algorithm with a cool-down period that prevents overspawning for the duration of a training phase. This requires some knowledge of the duration which is external to the meta-learner, thus the *task-aware* nomenclature (see Appendix D.1 for further details).

It is clear from Figure 8 that neither of our algorithms suffer from catastrophic forgetting to the same degree as the parametric baselines. In fact, at the end of training, both of our methods outperform all the parametric baselines on the first and second task.

Results: Specialization. Given the higher capacity of the parametric baselines, and the inherent degree of similarity between the filtered *mini*ImageNet task distributions (unlike the regression tasks in the previous section), the parametric baselines perform better on each task distribution while during its active phase. However, they quickly suffer from degradation once the task distribution shifts. Our approach does not suffer from this phenomenon and can handle non-stationarity owing to the credit assignment of a single task distribution to a specialized cluster. This specialization is illustrated in Figure 9, where we track the evolution of the average cluster responsibilities on the meta-test dataset from each of the three *mini*ImageNet few-shot classification tasks. Each cluster is specialized so as to acquire the majority of a single task distribution's test set assignments, despite the degree of similarity between tasks originating from the same source (*mini*ImageNet). We observed this difficulty with the non-monotone improvement of parametric clustering, as a function of components, in Section 4.

7 Related Work

Meta-learning. In this work, we show how changes to the hierarchical Bayesian model assumed in meta-learning [21, Fig. 1(a)] can be realized as changes to a meta-learning algorithm. In contrast,

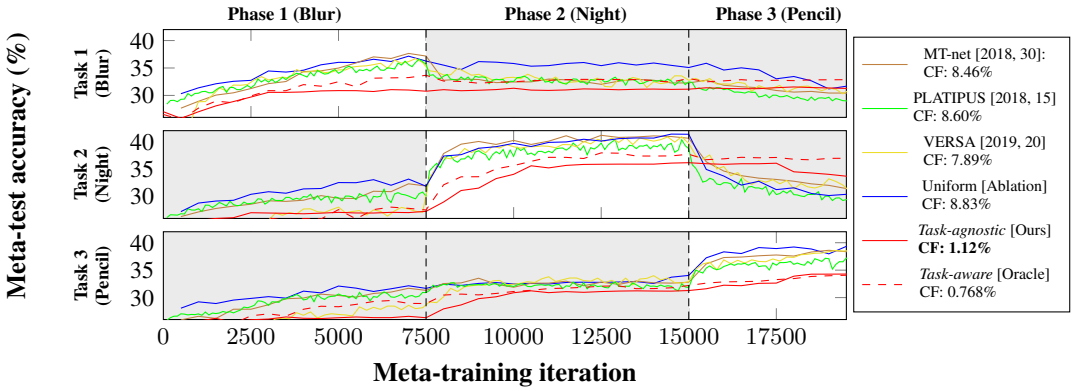


Figure 8: Results on the evolving dataset of filtered *mini*ImageNet few-shot classification tasks (higher is better). Each panel (row) presents, for a specific task type (filter), the average meta-test set accuracy over cumulative number of few-shot episodes. We additionally report the degree of loss 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).

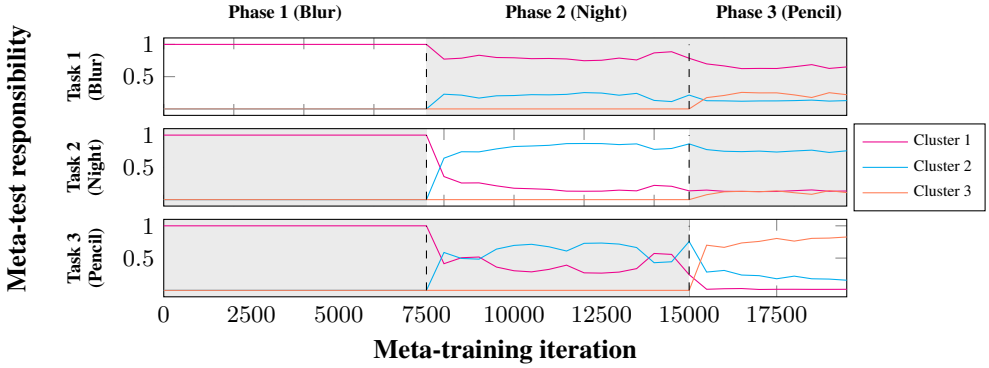


Figure 9: Each panel (row) presents task-specific per-cluster meta-test responsibilities $\gamma^{(\ell)}$ over time. A higher responsibility entails a higher degree of specialization of a particular cluster (color) to a particular task (row).

follow-up approaches to improving the performance of meta-learning algorithms [*e.g.,* 30, 15, 20] do not change the underlying probabilistic model; what differs is the inference procedure to infer values of the global (shared across tasks) and local (task-specific) parameters; for example, [20] consider feedforward conditioning while [15] employ variational inference. Due to consolidation into one set of global parameters shared uniformly across tasks, none of these methods inherently accommodate heterogeneity or non-stationarity.

Continual learning. Techniques developed to address the catastrophic forgetting problem in continual learning, such as elastic weight consolidation (EWC) [28], synaptic intelligence (SI) [58], variational continual learning (VCL) [38], and online Laplace approximation [42] require access to an explicit delineation between tasks that acts as a catalyst to grow model size, which we refer to as *task-aware*. In contrast, our non-parametric algorithm tackles the *task-agnostic* setting in which the meta-learner recognizes a latent shift in the task distribution and adapts accordingly.

8 Conclusion

Meta-learning is a source of learned inductive bias. Occasionally, this inductive bias is harmful because the experience gained from solving a task does not transfer. Here, we present an approach that allows a probabilistic meta-learner to explicitly modulate the amount of transfer between tasks, as well as to adapt its parameter dimensionality when the underlying task distribution evolves. We formulate this as probabilistic inference in a mixture model that defines a clustering of task-specific parameters. To ensure scalability, we make use of the recent connection between gradient-based meta-learning and hierarchical Bayes [21] to perform approximate *maximum a posteriori* (MAP) inference in both a finite and an infinite mixture model. Our work is a first step towards more realistic settings

of diverse task distributions, and crucially, *task-agnostic* continual meta-learning. The approach stands to benefit from orthogonal improvements in posterior inference beyond MAP estimation (*e.g.,* variational inference [27], Laplace approximation [32], or stochastic gradient Markov chain Monte Carlo [33]). as well as scaling up the base model (*e.g.,* trading the four-layer convolutional network for a more complex architecture).

References

[1] Abadi, Martín, Barham, Paul, Chen, Jianmin, Chen, Zhifeng, Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Irving, Geoffrey, Isard, Michael, *et al.* . 2016. Tensorflow: A system for large-scale machine learning. *Pages 265–283 of: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*.

[2] Bakker, Bart, & Heskes, Tom. 2003. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, **4**(May), 83–99.

[3] Bauer, Matthias, Rojas-Carulla, Mateo, Świkatkowski, Jakub Bartłomiej, Schölkopf, Bernhard, & Turner, Richard E. 2017. Discriminative k-shot learning using probabilistic models. *arXiv preprint arXiv:1706.00326*.

[4] Baxter, Jonathan. 1997. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, **28**(1), 7–39.

[5] Baxter, Jonathan. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, **12**(1), 149–198.

[6] Besag, Julian. 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 259–302.

[7] Blei, David M, Jordan, Michael I, *et al.* . 2006. Variational inference for Dirichlet process mixtures. *Bayesian analysis*, **1**(1), 121–143.

[8] Caruana, Rich. 1998. Multitask learning. *Pages 95–133 of: Learning to learn*. Springer.

[9] Caruana, Richard. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. *In: Proceedings of the 10th International Conference on Machine Learning (ICML)*.

[10] Collins, Anne GE, & Frank, Michael J. 2013. Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological review*, **120**(1), 190.

[11] Daumé III, Hal. 2009. Bayesian multitask learning with latent hierarchies. *Pages 135–142 of: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*.

[12] Deleu, Tristan, & Bengio, Yoshua. 2018. The effects of negative adaptation in Model-Agnostic Meta-Learning. *arXiv preprint arXiv:1812.02159*.

[13] Dempster, Arthur P, Laird, Nan M, & Rubin, Donald B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.

[14] Finn, Chelsea, Abbeel, Pieter, & Levine, Sergey. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *In: Proceedings of the 34th International Conference on Machine Learning (ICML)*.

[15] Finn, Chelsea, Xu, Kelvin, & Levine, Sergey. 2018. Probabilistic model-agnostic meta-learning. *Pages 9516–9527 of: Advances in Neural Information Processing Systems*.

[16] Gao, Jing, Fan, Wei, Jiang, Jing, & Han, Jiawei. 2008. Knowledge transfer via multiple model local structure mapping. *Pages 283–291 of: Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM.

[17] Gershman, Samuel J, & Blei, David M. 2012. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, **56**(1), 1–12.

[18] Ghahramani, Zoubin, & Beal, Matthew J. 2000. Variational inference for Bayesian mixtures of factor analysers. *Pages 449–455 of: Advances in neural information processing systems*.

[19] Glorot, Xavier, & Bengio, Yoshua. 2010. Understanding the difficulty of training deep feedforward neural networks. *Pages 249–256 of: Proceedings of the thirteenth international conference on artificial intelligence and statistics*.

[20] Gordon, Jonathan, Bronskill, John, Bauer, Matthias, Nowozin, Sebastian, & Turner, Richard. 2019. Meta-Learning Probabilistic Inference for Prediction. *In: ICLR*.

[21] Grant, Erin, Finn, Chelsea, Levine, Sergey, Darrell, Trevor, & Griffiths, Thomas. 2018. Recasting Gradient-based meta-learning as hierarchical Bayes. *In: Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[22] Greff, Klaus, van Steenkiste, Sjoerd, & Schmidhuber, Jürgen. 2017. Neural expectation maximization. *Pages 6694–6704 of: Advances in Neural Information Processing Systems*.

[23] Gupta, Sunil, Phung, Dinh, & Venkatesh, Svetha. 2013. Factorial multi-task learning: a Bayesian nonparametric approach. *Pages 657–665 of: International conference on machine learning*.

[24] Heskes, Tom. 1998. Solving a Huge Number of Similar Tasks: A Combination of Multi-Task Learning and a Hierarchical Bayesian Approach. *Pages 233–241 of: Proceedings of the 15th International Conference on Machine Learning (ICML)*.

[25] Hughes, Michael C, Fox, Emily, & Sudderth, Erik B. 2012. Effective split-merge monte carlo methods for nonparametric models of sequential data. *Pages 1295–1303 of: Advances in neural information processing systems*.

[26] Jain, Sonia, & Neal, Radford M. 2004. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of computational and Graphical Statistics*, **13**(1), 158–182.

[27] Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, & Saul, Lawrence K. 1999. An introduction to variational methods for graphical models. *Machine learning*, **37**(2), 183–233.

[28] Kirkpatrick, James, Pascanu, Razvan, Rabinowitz, Neil, Veness, Joel, Desjardins, Guillaume, Rusu, Andrei A, Milan, Kieran, Quan, John, Ramalho, Tiago, Grabska-Barwinska, Agnieszka, *et al.* . 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, **114**(13), 3521–3526.

[29] Lawrence, Neil D, & Platt, John C. 2004. Learning to learn with the informative vector machine. *Page 65 of: Proceedings of the 21st International Conference on Machine Learning (ICML)*.

[30] Lee, Yoonho, & Choi, Seungjin. 2018. Gradient-based meta-learning with learned layerwise metric and subspace. *In: ICML*.

[31] Lin, Dahua. 2013. Online learning of nonparametric mixture models via sequential variational approximation. *Pages 395–403 of: Advances in Neural Information Processing Systems (NIPS)*.

[32] MacKay, David. 1992. A practical Bayesian framework for backpropagation networks. *Neural computation*, **4**(3), 448–472.

[33] Metropolis, Nicholas, & Ulam, Stanislaw. 1949. The monte carlo method. *Journal of the American statistical association*, **44**(247), 335–341.

[34] Müller, Peter, & Insua, David Rios. 1998. Issues in Bayesian analysis of neural network models. *Neural Computation*, **10**(3), 749–770.

[35] Neal, Radford M. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, **9**(2), 249–265.

[36] Neal, Radford M. 2012. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media.

[37] Neal, Radford M, & Hinton, Geoffrey E. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Pages 355–368 of: Learning in graphical models*. Springer.

[38] Nguyen, Cuong V, Li, Yingzhen, Bui, Thang D, & Turner, Richard E. 2017. Variational continual learning. *arXiv preprint arXiv:1710.10628*.

[39] Rasmussen, Carl Edward. 2000. The infinite Gaussian mixture model. *Pages 554–560 of: Advances in neural information processing systems*.

[40] Ravi, Sachin, & Larochelle, Hugo. 2017. Optimization as a model for few-shot learning. *In: Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

[41] Raykov, Yordan P, Boukouvalas, Alexis, Baig, Fahd, & Little, Max A. 2016. What to do when K-means clustering fails: a simple yet principled alternative algorithm. *PloS one*, **11**(9), e0162259.

[42] Ritter, Hippolyt, Botev, Aleksandar, & Barber, David. 2018. Online structured Laplace approximations for overcoming catastrophic forgetting. *Pages 3738–3748 of: Advances in Neural Information Processing Systems*.

[43] Robbins, Herbert, & Monro, Sutton. 1951. A stochastic approximation method. *The annals of mathematical statistics*, **22**(3), 400–407.

[44] Rosenstein, Michael T, Marx, Zvika, Kaelbling, Leslie Pack, & Dietterich, Thomas G. 2005. To transfer or not to transfer. *Pages 1–4 of: NIPS 2005 workshop on transfer learning*, vol. 898.

[45] Rothman, Adam J, Levina, Elizaveta, & Zhu, Ji. 2010. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, **19**(4), 947–962.

[46] Salakhutdinov, Ruslan, Tenenbaum, Joshua B, & Torralba, Antonio. 2013. Learning with hierarchical-deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(8), 1958–1971.

[47] Schmidhuber, Jürgen. 1987. *Evolutionary principles in self-referential learning*. Ph.D. thesis, Institut für Informatik, Technische Universität München.

[48] Sculley, David. 2010. Web-scale k-means clustering. *Pages 1177–1178 of: Proceedings of the 19th international conference on World wide web*. ACM.

[49] Snell, Jake, Swersky, Kevin, & Zemel, Richard S. 2017. Prototypical Networks for Few-shot Learning. *In: Advances in Neural Information Processing Systems (NIPS) 30*.

[50] Srivastava, Nitish, & Salakhutdinov, Ruslan R. 2013. Discriminative transfer learning with tree-based priors. *Pages 2094–2102 of: Advances in Neural Information Processing Systems*.

[51] Tank, Alex, Foti, Nicholas, & Fox, Emily. 2015. Streaming variational inference for Bayesian nonparametric mixture models. *Pages 968–976 of: Artificial Intelligence and Statistics*.

[52] Thrun, Sebastian. 1996. Discovering structure in multiple learning tasks: The TC algorithm.

[53] Vinyals, Oriol, Blundell, Charles, Lillicrap, Tim, Wierstra, Daan, *et al.* . 2016. Matching networks for one shot learning. *Pages 3630–3638 of: Advances in Neural Information Processing Systems (NIPS) 29*.

[54] Wan, Jing, Zhang, Zhilin, Yan, Jingwen, Li, Taiyong, Rao, Bhaskar D, Fang, Shiaofen, Kim, Sungeun, Risacher, Shannon L, Saykin, Andrew J, & Shen, Li. 2012. Sparse Bayesian multi-task learning for predicting cognitive outcomes from neuroimaging measures in Alzheimer's disease. *Pages 940–947 of: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[55] Welling, Max, & Kurihara, Kenichi. 2006. Bayesian K-means as a "maximization-expectation" algorithm. *Pages 474–478 of: Proceedings of the 2006 SIAM international conference on data mining*. SIAM.

[56] Xue, Ya, Liao, Xuejun, Carin, Lawrence, & Krishnapuram, Balaji. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, **8**(Jan), 35–63.

[57] Yu, Kai, Tresp, Volker, & Schwaighofer, Anton. 2005. Learning Gaussian processes from multiple tasks. *Pages 1012–1019 of: Proceedings of the 22nd International Conference on Machine Learning (ICML)*.

[58] Zenke, Friedemann, Poole, Ben, & Ganguli, Surya. 2017. Continual learning through synaptic intelligence. *arXiv preprint arXiv:1703.04200*.

[59] Zhang, Yi, & Schneider, Jeff G. 2010. Learning multiple tasks with a sparse matrix-normal penalty. *Pages 2550–2558 of: Advances in Neural Information Processing Systems*.

[60] Zhang, Yongyue, Brady, Michael, & Smith, Stephen. 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, **20**(1), 45–57.

[61] Zhang, Yu, & Yeung, Dit-Yan. 2014. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **8**(3), 12.

A Extended related work

Multi-task learning. [44] demonstrated that negative transfer can worsen generalization performance, and avoidance of negative transfer has motivated much work on hierarchical Bayes in transfer learning and domain adaptation [*e.g.,* 29, 57, 16, 11, 54]. Closest to our proposed approach is early work on hierarchical Bayesian multi-task learning with neural networks that places a prior only on the output layer [24, 2, 46, 50]. In contrast, we place a non-parametric prior on the full set of neural network weights. Furthermore, none of these approaches were applied to the episodic training setting of meta-learning. Similar to our point estimation procedure, [24] and [50] propose training a mixture model over the output layer weights of a neural network using MAP inference. However, these approaches do not scale well to all the layers in a network as performing full passes on the dataset for inference of the full set of weights is computationally intractable in general.

Clustering. Incremental or stochastic clustering was considered in the EM setting in [37]. and in the K-means setting in [48]. [31] conducted online learning of a non-parametric mixture model using sequential variational inference. A key distinction between our work and these approaches is that we leverage the connection between empirical Bayes in a hierarchical model and gradient-based meta-learning [21] to use a MAML-like [14] objective as a log posterior surrogate. This allows our algorithm to make use of a scalable stochastic gradient descent optimizer instead of alternating a maximization step with an inference pass over the full dataset [*c.f.,* 50, 3].

Our approach is also distinct from recent work on gradient-based clustering [22] since we employ the episodic batching of [53]. This can be a challenging setting for a clustering algorithm, as the assignments need to be computed using, for example, $K = 1$ examples per class in the 1-shot setting.

Contrasting the batch and stochastic settings. In the stochastic setting, access to past data is unavailable, and so none of the standard algorithms and heuristics for inference in non-parametric models are applicable [*e.g.,* 26, 25]. In particular, our proposed algorithm does not refine the cluster assignments of previously observed points by way of multiple expensive passes over the whole dataset.

In contrast, we incrementally infer model parameters and add components during episodic training based on noisy estimates of the gradients of the marginal log-likelihood. Moreover, we avoid the need to preserve task assignments, which is potentially harmful due to stale parameter values, since the task assignments in our framework are meant to be easily reconstructed on-the-fly using the `E-STEP` with updated parameters $\boldsymbol{\theta}^{(0)}, \ldots, \boldsymbol{\theta}^{(L)}, G$.

***Maximum a posteriori* estimation as iterated conditional modes.** Due to the high-dimensionality of the parameter set of neural networks, we consider a mode estimation procedure based on iterated conditional modes (ICM) [6, 60, 55, 41] that can leverage gradient computation instead of the expensive process of Gibbs sampling. iterated conditional modes (ICM) is a greedy strategy that iteratively maximizes the full conditional distribution for each variable (*i.e.,* computes the MAP estimate), instead of sampling from the conditional as is done in Gibbs sampling [55]. This leads to a fast point-estimation of the DPMM parameters in which we only need to track the means of the cluster priors.

Alternative inference procedures in probabilistic mixtures. A standard approach for estimation in latent variable models, such as probabilistic mixtures, is to represent the distribution using samples produced via some sampling algorithm.The most widely used is the Gibbs sampler [35, 17], which draws from the conditional distribution of each latent variable, given the others, until convergence to the posterior distribution over all the latents. However, in the setting of latent variables defined over high-dimensional parameter spaces such as those of neural network models, using a sampling algorithm such as the Gibbs sampler is prohibitively expensive [36, 34]. Instead of sampling, one can fit factorized variational distributions to the exact distribution $p(\phi, z|x) \approx q(\phi)q(z)$ [18, 7]. It should be noted that we do not claim that our method of point estimation in the DPMM is the most accurate method for posterior inference but we leave improved approximate inference extensions to future work.

The main drawback of using point estimates for a non-parametric mixture estimation is the inability to leverage the diffusion of the global prior G_0 when computing the likelihood of a new cluster. Highly concentrated parameter estimates for non-empty clusters should lead to low likelihoods for outlier tasks, whereas the diffused global prior should be better at capturing a wider variety of tasks.

Nonetheless, point estimation is a necessary trade-off between computation and accuracy. To allow for a more accurate estimate of the likelihood, we experimented with simulating a normal centered at the global prior mean with a variance hyperparameter that can be annealed over time to account for increased certainty about the prior choice. We can then compare the average cluster responsibility to the threshold. Another interesting extension we experimented with was to compute the gradient for each of the samples and average over the number of samples as to approximate the expectation of the gradient under the global prior. However, we found this to be less stable than simply comparing the cluster responsibilities to the threshold.

B *Maximum a posteriori* estimation in the Dirichlet process mixture model

From (4) and using a conditional mode estimate for task-specific parameters $\boldsymbol{\phi}_j$,

$$\log p\left(\boldsymbol{z}_j = \ell \mid \boldsymbol{x}_{j_{1:M}}, \boldsymbol{z}_{1:j-1}, \boldsymbol{\theta}^{(\ell)}\right) \approx \begin{cases} \log n^{(\ell)} + \log p(\boldsymbol{x}_{j_{1:M}} | \hat{\boldsymbol{\phi}}_j^{(\ell)}) + \\ \log p(\hat{\boldsymbol{\phi}}_j^{(\ell)} | \boldsymbol{\theta}^{(\ell)}) & \text{for } \ell \leq L \\ \log \zeta \ + \ \log p(\boldsymbol{x}_{j_{1:M}} | \hat{\boldsymbol{\phi}}_j^{(\ell)}) \ + \\ \log(\hat{\boldsymbol{\phi}}_j^{(\ell)} | \boldsymbol{\theta}^{(0)}) & \text{for } \ell = L+1 \ . \end{cases} \tag{5}$$

C Experimental setup

C.1 Dataset details

Few-shot regression

- Polynomial wave (Figure 4a):
$$y = \sum_i a_i x^{p_i}$$
and $a \sim \mathcal{U}(-5.0, 5.0)$.
- Sinusoid wave (Figure 4b):
$$y = a \sin(x - \phi)$$
where $\phi \sim \mathcal{U}(0, \pi)$ and $a \sim \mathcal{U}(0.1, 5.0)$.
- Sawtooth wave (Figure 4c):
$$y = -\frac{2a}{\pi} \arctan(\cot(\frac{x\pi}{\phi}))$$
where $\phi \sim \mathcal{U}(0, \pi)$, $a \sim \mathcal{U}(0.1, 5.0)$.

C.2 Hyperparameter choices

C.2.1 *Mini*ImageNet few-shot classification.

We use the same data split, neural network architecture, and hyperparameter values as in [14] for common components. We use $\tau = 1$ for the softmax temperature and the same initialization as [14] for the global prior G_0. We determine an iteration number for early stopping using the validation set.

C.2.2 Continual few-shot regression.

Our architecture is a feedforward neural network with 2 hidden layers with ReLU nonlinearities, each of size 40. We use a meta-batch size of 10 tasks (both for the inner updates and the meta-gradient updates) for 5-shot regression. Our non-parametric algorithm starts with a single cluster ($L_0 = 1$ in Algorithm 3). In these experiments, we set the spawning threshold $\epsilon = 0.95T/(L+1)$, with L the number of non-empty clusters and T the size of the meta-batch. We use the mean-squared error for each task as the inner loop and meta-level objectives.

C.2.3 Continual few-shot *mini*ImageNet classification.

We use the same data split, neural network architecture, and hyperparameter values as in [14] for common components. We use a meta-batch size of 4 tasks, start with a single cluster, and set the spawning threshold to the same formula as in Section C.2.2. We use the multi-class cross entropy error for each task as the inner loop and meta-level objectives. More details on the the practical implementation for image datasets of the non-parametric algorithm can found in Section D.

D Practical and implementational details

D.1 *Task-aware vs. task-agnostic*

Since a cluster is not well-tuned immediately after its creation, we consider a cool-down period after the spawning of each new cluster where we do not consider the creation of new clusters for a fixed number of iterations, and we freeze the updating of existing clusters for a same number of iterations. This allows the newly-created cluster to take enough gradient updates in order to move from its global prior initialization, allowing it to sufficiently differentiate from the global prior.

This experimental paradigm also allows us to approximate the *task-aware* algorithms of prior work [*e.g.,* 28, 58, 38, 42] which require access to an explicit delineation between tasks that acts as a catalyst to grow model size. For the *task-aware* non-parametric mixture results reported in the experiments, we set this cool-down period to be exactly the length of the training phase for the appropriate dataset; therefore, clusters which are not meant to be specialized for the active dataset are not updated. In contrast, the *task-aware* results consider a cool-down period of $1k$ iterations, which is less than 15% of the active period for each dataset. Extensions to this fixed cool-down period could consider the rate of learning in the active cluster in order to detect when the new component has been sufficiently fit to the new task.

D.2 Practical extensions to the non-parametric algorithm

The penalty term of $\log n^{(\ell)}$ or $\log \zeta$ is necessary to regularize the likelihood of a potential new cluster in order to limit overspawning. However, in the setting where the likelihood is approximated by the loss function of a complex neural network, as in the case for most meta-learning applications, there is a large difference in orders of magnitude between the loss value (especially for the cross-entropy function) and the penalty term, even after a single batch of assignments. Furthermore, the classical log observation count $\log n$ term is misaligned with our stochastic setting for two reasons. First, since we do not re-evaluate over the whole dataset for every meta-learning episode, we are thus more concerned with the relative number of task assignments over recent iterations than the total number of assignments over the duration of training. Second, the number of tasks to be assigned can grow too large in the stochastic setting (e.g. $60k$ for *mini*ImageNet) which exacerbates the already large difference in orders of magnitudes between the loss function and the penalty term.

Accordingly, we propose two changes; First, we compute the observation based on a moving window of fixed size (5 in the experiments). Second, we apply a coefficient, which can be tuned, to the log observation count in (4). This provides more flexibility to our meta-learner as it allows it to apply to any black-box function approximator which might exhibit losses of orders of magnitudes smaller than those expected of classical probabilistic models. While the moving window size and CRP penalty coefficient terms are somewhat interdependent, we propose them as a simple starting point to tune this non-parametric meta-learner beyond what is empirically explored in this paper.

Note that without such changes in the stochastic setting of meta-learning, a nonparametric algorithm would be unable to spawn a new cluster after the first handful of iterations. Even if we were to lower the threshold ϵ, multiple almost identical clusters would be spawned in the first few iterations before it would be impossible to spawn anymore. Furthermore, the clusters would be nearly identical given the small step size of a gradient update for each meta-learning episode. Finally, this would be computationally intensive since unlike the typical applications of non-parametric mixture learning where one can afford to spawn hundreds of components then prune them over the training procedure.

D.3 Thresholding

A marked difference that is not immediate from the Gibbs conditionals is the use of a threshold on the cluster responsibilities, detailed in the `E-STEP` in Subroutine 4, to account for noise from stochastic optimization when spawning a cluster on the basis of a single batch. This threshold is necessary for the stochastic mode estimation procedure of Algorithm 3, as it ensures that a new cluster's responsibility needs to exceed a certain value before being permanently added to the set of components.

If a cluster has close to an equal share of responsibilities as compared to existing clusters after accounting for the CRP penalty $\log n^{(\ell)}$ or $\log \zeta$, it is spawned. Accordingly, this approximate inference routine still preserves the preferential attachment ("rich-get-richer") dynamics of Bayesian nonparametrics [41]. A sequential approximation for non-parametric mixtures with a similar threshold was proposed in [31] and [51], in which variational Bayes was used instead of point estimation in a DPMM.

D.4 Pruning heuristics

None of the results reported in our experiments used a pruning heuristic as we used a rather conservative hyper parameter setting that deters overspawning. We did however explore different heuristics which could work in more general settings, especially in the presence of many more latent clusters of tasks than considered in the experimental settings in this work. One such heuristic is to prune small clusters that have received disproportionately few assignments over a certain number of past iterations. Another is to evaluate the functional similarity of two clusters by computing an odds-ratio statistic for the assignment probabilities to each cluster over a set of validation tasks. If the odds-ratio statistic is below a certain threshold, the smaller cluster can be pruned.

D.5 Estimating the CRP hyperparameters

We fixed α at the size of the meta-batch. An alternative is to place a $\Gamma(1,1)$ on the concentration parameter. Based on the likelihood , the posterior is then proportional to $p(\alpha|N,K) \propto \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)}\alpha^K e^{-\alpha}$ This is not a standard distribution but [39] have shown that $\log p(\alpha|N,K)$ is log-concave and methods such as L-BFGS have been used successfully in prior works. Alternatively, if we have some prior knowledge about the expected number of clusters, we can compute α based on $E[K] = \alpha \log N$. For the window-size, we considered an initial size of 20 iterations that can grow as more cluster are considered.

D.6 Implementation details

We implemented both of our parametric and non-parametric meta-learners in TensorFlow (TF) [1]. We considered 2 different settings for the `M-STEP` optimization:

- Train each cluster's parameters separately based on its corresponding loss function in an alternating manner closest to the classic EM algorithm.
- Train all cluster weights simultaneously using a surrogate loss over all validation batches.

Since the latter better leverages the differentiability of softmax-clustering and performed better empirically, we used it to report all experimental results.

D.6.1 Nonparametric Implementation

For the nonparametric algorithm, we chose the first approach to the `M-STEP` by constructing separate optimizers for each cluster's parameters. We pre-allocate a set of weights and use a mask during training to discard the parameters of empty clusters due to the static nature of TF graphs. When the algorithm exhausts the set of pre-allocated weights, we simply construct more network weight and reinitialize our optimizers.

D.6.2 CRP global prior

The likelihood of a new cluster is sensitive to the choice of a base measure or prior prior, G_0 on the cluster hyperparameters. Our gradient-based point estimation does not make any modeling assumption on the distribution of the weights, rendering the problem of principally updating the base measure, after or during training, non-trivial. We chose to initialize all weights with zero-mean normals in the fully-connected layers. For the convolutional layers, we leveraged Xavier initialization [19] similarly to prior work [14] in meta-learning.

However, such initialization is poor in the non-parametric for most non-trivial regression or classification tasks. Therefore, in the nonparametric setting, we start with a single cluster for a fixed number of iterations. We then initialize all clusters with the weights of the first clusters. This set of weights can be considered as the mean of the base measure or global prior in our setting.

We periodically update the global prior using a uniform average of the parameters of the existing clusters. This can be done by simply averaging over the parameter of the non-empty clusters as weighted by their sizes. Note that, we found that performing weighted KDE smoothing with a small bandwidth hyperparameter to perform slightly better than the average which is to be expected for neural network parameters. The number of iterations between updates of the global prior is a hyperparameter that we tune on the validation set. It is also possible to continuously, but less frequently over time, update this global prior as more data is encountered.