Adaptive Compositional Continual Meta-Learning

Bin Wu [1] Jinyuan Fang [2] Xiangxiang Zeng [3] Shangsong Liang [4 5] Qiang Zhang [1]

Abstract

This paper focuses on continual meta-learning, where few-shot tasks are heterogeneous and sequentially available. Recent works use a mixture model for meta-knowledge to deal with the heterogeneity. However, these methods suffer from parameter inefficiency caused by two reasons: (1) the underlying assumption of mutual exclusiveness among mixture components hinders sharing meta-knowledge across heterogeneous tasks. (2) they only allow increasing mixture components and cannot adaptively filter out redundant components. In this paper, we propose an **A**daptive **C**ompositional Continual **M**eta-**L**earning (ACML) algorithm, which employs a compositional premise to associate a task with a subset of mixture components, allowing meta-knowledge sharing among heterogeneous tasks. Moreover, to adaptively adjust the number of mixture components, we propose a component sparsification method based on evidential theory to filter out redundant components. Experimental results show ACML outperforms strong baselines, showing the effectiveness of our compositional meta-knowledge, and confirming that ACML can adaptively learn meta-knowledge.

1. Introduction

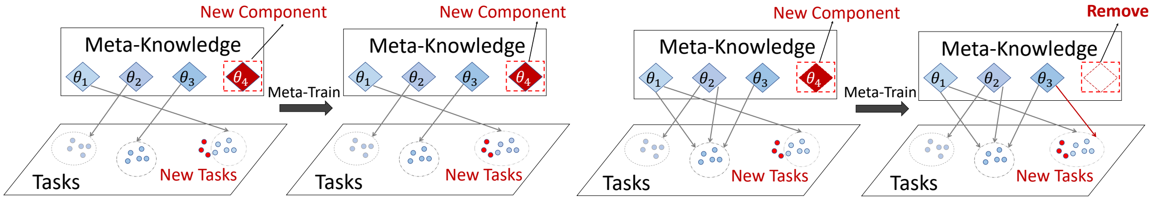
Meta-learning is an effective paradigm to deal with low-resource learning tasks where only a few labeled samples are available (Vanschoren, 2018; Hospedales et al., 2020), and has gained tremendous attention in recent years. The key idea of meta-learning is to inductively transfer meta-knowledge (i.e., the experience of how to learn) among different tasks to improve data efficiency and enhance model generalization (Raghu et al., 2020). Traditional meta-learning assumes all tasks are homogeneous and available instantly (Finn et al., 2017; 2018), which can be unrealistic in real-world scenarios. In this paper, we focus on a more practical and challenging setting, namely continual meta-learning where few-shot tasks are heterogeneous and arrive sequentially over time (Finn et al., 2019; Denevi et al., 2019). Such a heterogeneous and continual setting fits better for the real world and thus can be of practical significance to different domains, such as recommendation (Zhang et al., 2019) and personalization (Wu et al., 2022).

Despite the great potential of continual meta-learning in practical applications, there are two key challenges to be tackled in this setting: (i) how to capture the incremental meta-knowledge from the new tasks (Lee et al., 2017), (ii) how to avoid forgetting the learned meta-knowledge from previous tasks when dealing with the new heterogeneous tasks, i.e.,*catastrophic forgetting* (Kirkpatrick et al., 2017). Existing works (Jerfel et al., 2019; Yao et al., 2019; Zhang et al., 2021) mainly tackle these two challenges using a mixture model for meta-knowledge, where each component of the mixture model handles a task cluster, i.e., a set of homogeneous tasks. In order to learn incremental meta-knowledge and avoid catastrophic forgetting when facing a new task, they learn a new component for the mixture model if the task is dissimilar from previous tasks, otherwise, they update an existing component to which the task corresponds.

However, these works suffer from the parameter-inefficiency issue for two reasons. (1) They implicitly assume different components in the mixture meta-knowledge distribution are mutually exclusive. As heterogeneous tasks (i.e., tasks from different clusters) still have common information, the one-to-one mapping between tasks and mixture components overlooks the sharing of meta-knowledge among heterogeneous tasks, leading to meta-knowledge redundancy in different components. (2) They increase mixture components by some priors (e.g., Chinese Restaurant Prior) (Jerfel et al., 2019; Zhang et al., 2021) or a simple judgment on the similarities among tasks (Yao et al., 2019; 2020). These methods cannot adaptively adjust the actually required components from the data, because they only allow adding new meta-knowledge components but not removing redundant components. Consequently, one needs to maintain a large number of meta-knowledge components and face the param-

[1]Zhejiang University, Hangzhou, China [2]University of Glasgow, Glasgow, UK [3]Hunan University, Changsha, China [4]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE [5]Sun Yat-sen University, Guangzhou, China. Correspondence to: Qiang Zhang <qiang.zhang.cs@zju.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).



(a) Existing Works

(b) Our Work

Figure 1. The difference in incremental meta-knowledge between the existing works and ours. Previous methods maintain the mutual exclusive mixture meta-knowledge and only increase the number of components to add new meta-knowledge (the red dashed grid), which can lead to redundant components. Our algorithm provides tasks with a compositional meta-knowledge and employs an evidential sparsification to filter out the redundant meta-knowledge.

eter inefficiency issue (seen Fig. 1).

In this paper, we propose an **A**daptive **C**ompositional Continual **M**eta-**L**earning algorithm, abbreviated as ACML. Specifically, in order to allow meta-knowledge sharing among different clusters of tasks, we break the one-to-one mapping between a task and a meta-knowledge component; instead, we assume a task is composed of multiple aspects, each of which can be described by a different component. Accordingly, we build a one-to-many mapping between a task and meta-knowledge components. For example, in the image classification task, one component could learn how to detect colors and another component could learn how to capture object shapes. The meta-knowledge distribution for a task is therefore compositional, and different clusters of tasks are enabled to share the meta-knowledge via the overlapped components. This is achieved by an Indian Buffet Process (IBP) prior (Griffiths & Ghahramani, 2011) on the mixture components of meta-knowledge. Notwithstanding, it also faces the problem that we can only add new components but not remove redundant ones to meet the actual needs of available tasks. Filtering out the redundant components is necessary but non-trivial as components are mutually dependent in the one-to-many relationship between a task and meta-knowledge components. Therefore, we propose a component sparsification method based on the Evidential Theory (Dempster, 2008), which is a post hoc method after the update of meta-knowledge. In specific, we calculate the support and doubt degree for each component and remove the components which do not receive support from the tasks. Our contributions are summarized as:

- We propose a compositional meta-knowledge distribution via IBP, which enables meta-knowledge to be shared among heterogeneous tasks.
- We propose a post hoc evidential sparsification method to remove redundant meta-knowledge components.
- We conduct extensive experiments on four real-world datasets and the results show that our ACML outperforms the-state-of-art baselines under the heterogeneous continual learning setting.

2. Related Work

Meta-Learning. Meta-learning (Vanschoren, 2018; Hospedales et al., 2020) focuses on a few-shot setting. Recent works include metric-based (Snell et al., 2017; Oreshkin et al., 2018), model-based (Ha et al., 2016; Munkhdalai & Yu, 2017), optimization-based methods (Finn et al., 2017; 2018) and their Bayesian variants (Ravi & Beatson, 2018; Gordon et al., 2019; Iakovleva et al., 2020). However, most of them construct a globally-shared meta-knowledge, which can not fit the heterogeneous data distribution in the real world (Jerfel et al., 2019). To solve this problem, some works (Jerfel et al., 2019; Zhang et al., 2021) maintain a mixture of meta-knowledge, where a cluster of similar tasks is associated with a component of the meta-knowledge. This impedes the sharing of meta-knowledge between different clusters of tasks. In contrast, we break the one-to-one mapping between tasks and meta-knowledge components and build a one-to-many mapping based on the compositional premise to achieve more efficient parameter learning.

Continual Learning. Continual learning (Delange et al., 2021) typically overcomes the catastrophic forgetting issue via replay (Hu et al., 2019; Titsias et al., 2019), regularization (Benjamin et al., 2018; Pan et al., 2020) or incremental model selection (Kumar et al., 2021; Kessler et al., 2021). Moreover, there are some efforts in exploring compositional generalization in continual learning (Mendez & EATON, 2021) and meta-learning (Conklin et al., 2021; Requeima et al., 2019; Bronskill et al., 2020). They further guide many recent works focusing on continual meta-learning (Finn et al., 2019; Zhuang et al., 2020) to extend meta-knowledge when encountering new tasks, via increasing the number of mixture components (Yao et al., 2019) or adding a novel block to construct the mate-path (Yao et al., 2020). The Chinese Restaurant Process (CRP) has been used to determine the prior number of meta-knowledge components (Jerfel et al., 2019; Zhang et al., 2021). However, these methods only allow an increase of meta-knowledge components, and can not filter out redundant meta-knowledge. This would lead to parameter inefficiency and large computational consumption. In this paper, we employ an IBP prior to deciding whether to increase the number of components and more importantly, propose a post hoc sparsification method based on the Evidential Theory to filter the redundant component after the update of meta-knowledge.

Sparsification Method In recent years, a number of methods have been proposed to sparse the multi-modal space. Most of them (Martins & Astudillo, 2016; Laha et al., 2018) use a softmax alternative to sparse the large output space. Itkina et al. (2020) point out that the above methods are aggressive, and develop a post hoc evidential sparsification for conditional variational auto-encoder, based on the conclusion in (Denœux, 2019) that most existing classifiers can be seen as converting features into mass function and merging them to the final result. Following Itkina et al. (2020), Chen et al. (2021) present an evidential softmax method. However, these methods operate on mutual exclusiveness, which is different from the mutual dependency among the meta-knowledge components. Moreover, our evidential sparsification method provides a novel view of how to apply the evidential theory to continual learning.

3. Background

3.1. Bayesian Continual Meta-Learning

We focus on continual meta-learning. At each time step t, the meta-learning model receives a task τ_t which is sampled from a task distribution $p(\tau)$. The task is associated with a dataset $\mathcal{D}_t$, which is split into two sub-datasets, namely a support set $\mathcal{D}_t^S = \{x_i, y_i\}_{i=1}^{N_t}$ for training and a query set $\mathcal{D}_t^Q = \{x_i, y_i\}_{i=1}^{M_t}$ for validation. Following previous works (Yap et al., 2021), we assume $p(\tau)$ follows a non-stationary distribution, i.e., the task distribution shifts over time. Given sequentially arriving tasks from the non-stationary distribution, the goal of continual meta-learning is to adapt the meta-learning model to new tasks and avoid forgetting the learned knowledge from previous tasks.

One critical issue of continual meta-learning is the catastrophic forgetting problem when adapting models to sequential tasks from non-stationary distribution (Lee et al., 2017). To overcome the issue in the non-stationary task flow, some Bayesian meta-learning methods (Yap et al., 2021; Zhang et al., 2021) have been developed. They regard meta-knowledge as a latent variable and learn the posterior of meta-knowledge with the constraint of the prior in an online way following the principle of Variational Continual Learning (VCL) (Nguyen et al., 2018):

$$p(\boldsymbol{\theta}_t|\mathcal{D}_{1:t}) \propto p(\mathcal{D}_t|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t|\mathcal{D}_{1:t-1}), \quad (1)$$

where $\boldsymbol{\theta}_t$ is the meta-knowledge and used as the initialization following MAML (Finn et al., 2017). Note that VCL uses the posterior of meta-knowledge at $(t-1)$-th step as the prior of meta-knowledge at t-th step to serve as a regularization in order to alleviate the catastrophic forgetting problem. Moreover, the likelihood function at t-the step, i.e., $p(\mathcal{D}_t|\boldsymbol{\theta}_t)$, is defined in a probabilistic way (Gordon et al., 2019; Iakovleva et al., 2020):

$$p(\mathcal{D}_t|\boldsymbol{\theta}_t) = \int p(\mathcal{D}_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t)d\boldsymbol{\phi}_t, \quad (2)$$

where $\boldsymbol{\phi}_t$ is the task-specific parameter. Since the likelihood function is non-Gaussian, the exact posterior inference of meta-knowledge is intractable. To learn such intractable posterior, some approximate inference methods (e.g., variational inference (Kingma & Welling, 2013)) are applied. More details of inference are in Appendix A.

3.2. Evidential Theory

Evidential theory (i.e., The Dempster–Shafer (DS) theory) (Shafer, 1976; Yager & Liu, 2008) is a generalization of Bayesian theory (Dempster, 2008), which works on a discrete set of hypotheses (or equivalently, components of the mixture meta-knowledge distribution in this paper) and relaxes the constriction of exclusiveness in Bayesian theory. Such an relaxation fits well with the one-to-many relationship between tasks and components in our model.

Mathematically, let $Z = \{z_1, z_2, z_3, ..., z_K\}$ be a finite set, the element of which z_k is a binary variable indicating whether the current task is associated with the k-th component or not. The bayesian theory assumes belief is apportioned to each component in the finite set Z: $\sum_{k=1}^K p(z_k) = 1$, $p(z_k) + p(\overline{z_k}) = 1$, where $\overline{z_k}$ denotes all the components in Z other than z_k. Note that it implicitly assumes that components are mutually exclusive. In contrast, the evidential theory relaxes such constriction and allows belief to be assigned to a set of components by working over the power set of Z, denoted by 2^Z. The power set 2^Z represents any possible subset of Z, i.e., $2^Z = \{\emptyset, \{z_1\}, \{z_1, z_2\}, ..., Z\}$. Evidential theory defines a *mass function* on Z to construct the belief assignment, which is a mapping m: $2^Z \to [0, 1]$ and satisfies the following constraints:

$$m(\emptyset) = 0, \quad \sum_{A \subseteq Z} m(A) = 1, \quad (3)$$

where $\emptyset$ is an empty set and A is a subset of Z. Note that such a mass function is non-additive, since each subset to which belief is assigned, is not mutually exclusive. To merge mass functions deduced from different observations, Dempster (2008) proposes a Dempster's rule to help construct the fused mass function. The basic definition and more details about Dempster's rule are provided in Appendix B.

4. Adaptive Compositional Continual Meta-Learning

In this section, we present our Adaptive compositional Continual Meta-Learning algorithm (ACML). The overall framework of ACML is provided in Fig. 2.

4.1. Compositional Continual Meta-Learning

We first introduce the probabilistic framework of ACML. ACML relaxes the constraint that a task is associated with only a single component of meta-knowledge as such restriction of one-to-one mapping prevents the sharing of meta-knowledge among heterogeneous tasks. In contrast, we build a one-to-many mapping between a task and meta-knowledge components, which leads to a compositional distribution of meta-knowledge per task. The likelihood function at time t is:

$$\begin{aligned} p(\mathcal{D}_t|\boldsymbol{\theta}_t) &= \int p(\mathcal{D}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t)p(\boldsymbol{z}_t)d\boldsymbol{z}_t \\ &= \int \left[\int p(\mathcal{D}_t|\boldsymbol{\phi}_t)p(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t)d\boldsymbol{\phi}_t \right] p(\boldsymbol{z}_t)d\boldsymbol{z}_t, \end{aligned} \tag{4}$$

where $\boldsymbol{z}_t$ is the indicating vector consisting of binary elements, each element of which indicates whether the current task is associated with a meta-knowledge component or not. In this way, a subset of meta-knowledge components rather than a single component is leveraged to infer the task-specific parameter $p(\boldsymbol{\phi}_t|\boldsymbol{\theta_t}, \boldsymbol{z}_t)$, which is then used to tackle the few-shot tasks $p(\mathcal{D}_t|\boldsymbol{\phi}_t)$. Such a compositional premise, in which the meta-knowledge associated with a certain task consists of several components, enables the sharing of meta-knowledge among different clusters of tasks via the overlap components, relaxing the restriction of mutual exclusiveness in the conventional mixture meta-learning models.

4.2. Indian Buffet Process Prior

In the non-stationary regime, one important requirement is to capture incremental information when a newer task is encountered. Thus, the fixed meta-knowledge is not appropriate. To capture the incremental meta-knowledge and fit the compositional premise, we employ the Indian Buffet Process (IBP) (Griffiths & Ghahramani, 2011) to make a prior decision on the mixture components. Specifically, the number of mixture components at time t consists of two parts: $K_t = K_{t-1}+J_t$, where K_{t-1} is the number of mixture components at the previous time step and $J_t \sim Possion\left(\frac{\alpha}{t}\right)$ is the number of new components controlled by the hyperparameter α for capturing incremental knowledge. Therefore, the IBP prior for $\boldsymbol{z}_t$ is formulated based on a stick-breaking process (Teh et al., 2004):

$$\boldsymbol{v}_{t,k} \sim Beta(\alpha, 1),\ \boldsymbol{\pi}_{t,k} = \prod_{i=1}^{k} \boldsymbol{v}_{t,i},\ \boldsymbol{z}_{t,k} \sim Bern(\boldsymbol{\pi}_{t,k}), \tag{5}$$

for $k = 1, \ldots, K_t$, $Beta(\cdot)$ and $Bern(\cdot)$ represent the Beta distribution and the Bernoulli distribution respectively, and $\boldsymbol{z}_{t,k}$ is a binary value indicating whether the k-th component is associated to the task or not. Based on the IBP prior, the generative process of ACML is:

$$\boldsymbol{\theta}_{t,k} \sim \mathcal{N}(\boldsymbol{\mu}_{t,k}, \boldsymbol{\sigma}_{t,k}), \quad \boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t \sim p(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t), \tag{6}$$

for $k = 1, \ldots, K_t$, where $\boldsymbol{\theta}_{t,k}$ is the meta-knowledge of the k-th component, and the task-specific parameters $\boldsymbol{\phi}_t$ are inferred based on a subset of meta-knowledge components determined by $\boldsymbol{z}_t$. With the help of IBP prior, the coming tasks can reuse the meta-knowledge learned from the previous tasks and extend the meta-knowledge by adding additional components to fit well with the incremental requirement. The graphical model of our ACML is provided in Fig. 7 of the Appendix D. We introduce how to infer the posteriors of latent variables in the next section.

4.3. Structured Variational Inference

The exact inference is intractable because of non-conjugacy, thus, the approximation is required. In our work, we employ the variational inference (Blei et al., 2017) to approximate the posteriors. We capture the dependency among latent variables via structured mean-field variational inference (Hoffman & Blei, 2015). The variational distributions of latent variables are defined as:

$$\begin{aligned} &q(\boldsymbol{v}_t, \boldsymbol{z}_t, \boldsymbol{\theta}_t, \boldsymbol{\phi}_t|\mathcal{D}_t) \\ &\quad = q(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t, \mathcal{D}_t) \prod_{k=1}^{K_t} q(\boldsymbol{\theta}_{t,k})q(\boldsymbol{z}_{t,k}|\boldsymbol{v}_{t,k})q(\boldsymbol{v}_{t,k}), \end{aligned} \tag{7}$$

where we use Gaussian distributions to approximate the posteriors of $\boldsymbol{\phi}_t$ and $\boldsymbol{\theta}_{t,k}$, and use Bernoulli and Beta distributions to approximate the posteriors of $\boldsymbol{z}_{t,k}$ and $\boldsymbol{v}_{t,k}$, respectively. The parameters of these variational distributions are denoted as ψ_t. With the variational distributions, the training objective function, i.e, the evidence lower bound (ELBO) of the observation, at the current time t can be derived as follows:

$$\begin{aligned} &\mathcal{L}(\psi_t; \mathcal{D}_t) = -\mathbb{E}_{q(\boldsymbol{v}_t, \boldsymbol{z}_t, \boldsymbol{\theta}_t, \boldsymbol{\phi}_t|\mathcal{D}_t)}\left[\log p(\mathcal{D}_t|\boldsymbol{\phi}_t)\right] \\ &+ \sum_{k=1}^{K} \Big[\mathrm{KL}(q(\boldsymbol{v}_{t,k})\|p(\boldsymbol{v}_{t,k})) + \mathrm{KL}(q(z_{t,k}|\boldsymbol{v}_{t,k})\|p(z_{t,k}|\boldsymbol{v}_{t,k})) \\ &+ \mathrm{KL}(q(\boldsymbol{\theta}_{t,k})||p(\boldsymbol{\theta}_{t,k})) \Big] + \mathrm{KL}(q(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t, \mathcal{D}_t)||p(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t)), \end{aligned} \tag{8}$$

where $\mathrm{KL}(\cdot||\cdot)$ is the Kullback–Leibler divergence. Note that the expectation of likelihood in Eq. (8) can be computed

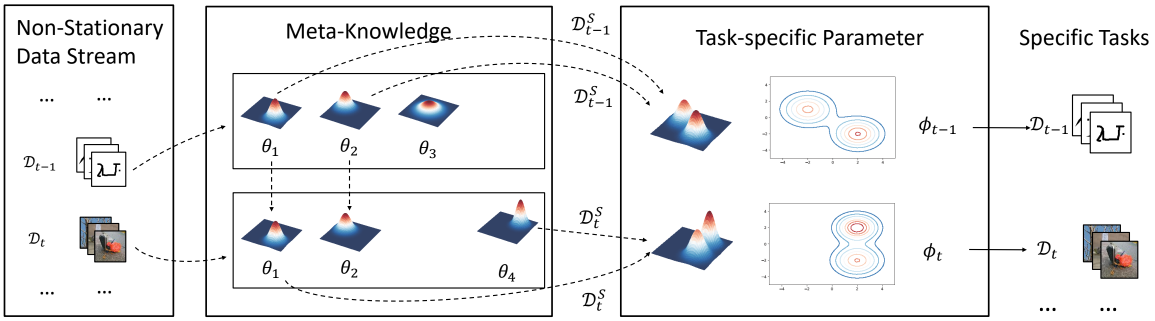


Figure 2. The ACML framework. The dashed line represents the parameter inference process and the solid line represents the data generation process. In this example, our proposed ACML maintains a mixture meta-knowledge distribution with three components from $\mathcal{D}_{t-1}$, and then creates an additional component $\boldsymbol{\theta}_4$ to capture the incremental information from $\mathcal{D}_t$. Note that the redundant component $\boldsymbol{\theta}_3$ is filtered out by the evidential sparsification method. Afterward, ACML associates a task with multiple meta-knowledge components to infer the task-specific parameter.

with Monte Carlo method by sampling from the variational distributions. However, it is not straightforward to sample from Bernoulli and Beta distributions while enabling back-propagation of gradients. To address this problem, we employ the implicit reparameterization (Figurnov et al., 2018) to enable the parameterization trick (Kingma & Welling, 2013) in these two distributions. The KL-terms can be computed in closed-form expressions. Details of the definition of variational distribution, the sampling gradient computation for the likelihood term, and the closed form expression for KL-terms are provided in Appendix D.

4.4. Evidential sparsification for Adaptive compositional Meta-Knowledge

The IBP prior cannot adaptively adjust the actually required components from the tasks, since they only increase the number of components as existing methods (Jerfel et al., 2019; Zhang et al., 2021) did. This would lead to redundant components when meeting a large number of tasks. However, it is non-trivial to identify useless meta-knowledge components, since the one-to-many relationship between a task and components assumes the non-exclusiveness among components, conflict with the common mutually-exclusive assumption in many existing sparsification methods (Itkina et al., 2020; Chen et al., 2021).

Therefore, we propose an evidential sparsification method for compositional meta-knowledge, which is a post hoc method after the update of meta-knowledge. Fig. 6 in Appendix C provides an intuitive explanation. Specifically, based on the evidential theory (Denœux, 2019; Itkina et al., 2020), we calculate a $w_{t,k}$ as the evidence weight indicating how much the k-th meta-knowledge component in the mixture distribution is useful to solve the task τ_t. According to Eq. (7), since the variational beta distribution $Beta(\boldsymbol{v}_{t,k}; \alpha_{t,k}, \beta_{t,k})$ determines the probability of the k-th component being selected for the task τ_t after the update of meta-knowledge, the $w_{t,k}$ is formulated as:

$$w_{t,k} = e^{\alpha_{t,k}} - \gamma \cdot e^{\beta_{t,k}}, \tag{9}$$

where $\alpha_{t,k}$ and $\beta_{t,k}$ are two parameters in the beta distribution and γ is the hyperparameter to adjust the sparsity of meta-knowledge. Note that this procedure only uses parameters of the beta distributions, without the need to memorize any task data. Following (Denœux, 2019), we assume the evidence weight of $\{\boldsymbol{z}_k\}$ and its complementary set$\overline{\{\boldsymbol{z}_k\}}$ equal to the positive part and the negative part of $w_{t,k}$, respectively:

$$w^+_{t,k} := \max(0, w_{t,k}), \quad w^-_{t,k} := \max(0, -w_{t,k}). \tag{10}$$

According to (Denoeux, 2008), the support degree and the doubt degree do not by themselves provide 100% certainty due to unknown information in the real world so the belief denoting the unknown information cannot point to any subset of components other than the universal set Z. Therefore, we define two mass functions for the support and doubt degrees, denoting the extent of certainty for the usefulness of $\{\boldsymbol{z}_k\}$ and $\overline{\{\boldsymbol{z}_k\}}$ respectively:

$$m^+_{t,k}(\{\boldsymbol{z}_k\}) = 1 - e^{-w^+_{t,k}},\ m^+_{t,k}(Z) = e^{-w^+_{t,k}}; \tag{11}$$

$$m^-_{t,k}(\overline{\{\boldsymbol{z}_k\}}) = 1 - e^{-w^-_{t,k}}, m^-_{t,k}(Z) = e^{-w^-_{t,k}}. \tag{12}$$

To reach a unified measure of how much a meta-knowledge component is useful for all occurring tasks, we need to merge all the mass functions. Finally, the merged mass function is as follows according to the evidential theory:

$$m(\{\boldsymbol{z}_k\}) = m^+_{1,1}(\{\boldsymbol{z}_k\}) \oplus ... \oplus m^-_{t,K_t}(\{\boldsymbol{z}_k\}) = \\ CC^+C^- \left\{ e^{-w^-_k} \left[e^{w^+_k} - 1 + \prod_{l \neq k} (1 - e^{-w^-_l}) \right] \right\}, \tag{13}$$

where C, C^+ and C^- are the normalization terms [1]. Note that there are $2 \times t \times K_t$ mass functions to be merged. To reduce the computational complexity, according to the computation rule in Dempster's rule, $w_k^+ = \sum_{t=0}^{T} w_{t,k}^+$ and $w_k^- = \sum_{t=0}^{T} w_{t,k}^-$, where T is the current time. The detailed derivation of Eq. (13) is in Appendix C. In the merged mass function, the components with a zero belief would be seen as redundant since they can not contribute to any existing tasks. These components ($m(\{\boldsymbol{z}_k\}) = 0$) are removed to reduce parameter redundancy.

To sum up, when encountering a new task, our ACML first determines whether to add new meta-knowledge components based on the IBP prior and then leverage the structured variational inference to update the meta-knowledge. Afterward, the evidential sparsification method is employed to adaptively filter out redundant components. The filtered meta-knowledge is used to infer the task-specific parameter $\boldsymbol{\phi}_t$ for a task during evaluation:

$$p(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t) = \frac{\sum_{k=1}^{K_t} \mathbb{1}\{m(\{\boldsymbol{z}_k\}) \neq 0\}\mathbb{1}\{\boldsymbol{z}_{t,k} \neq 0\} p(\boldsymbol{\phi}_{t,k}|\boldsymbol{\theta}_{t,k}; \lambda_{t,k})}{\sum_{k=1}^{K_t} \mathbb{1}\{m(\{\boldsymbol{z}_k\}) \neq 0\}\mathbb{1}\{\boldsymbol{z}_{t,k} \neq 0\}}. \tag{14}$$

4.5. Discussion

In contrast to recent works (Jerfel et al., 2019; Zhang et al., 2021), our work has two major differences that enhance the performance and confirm our contributions: (1) Different from the one-to-one matching between a task and a meta-knowledge component, our algorithm constructs a one-to-many matching via compositional meta-knowledge. It enables multiple task clusters to share meta-knowledge components, which improves the parameter efficiency. (2) Our algorithm employs evidential sparsification to adaptively filter out redundant components. Compared to those only using CPR as a prior, our method can meet the actual need of the arriving tasks, leading to higher parameter efficiency. The complexity analysis is shown in Appendix E.

5. Experiments

In what follows, we design experiments for three research questions to examine the effectiveness of ACML, which guides the remainder of the paper: **(RQ1)** Can our ACML achieve a better performance than the state-of-the-art baselines under the continual non-stationary setting? **(RQ2)** How does the number of components affect the performance? **(RQ3)** What is the impact of evidential sparsification on performance?

Experiments are conducted under continual non-stationary settings. We compare against the following baselines: (1) **Train-On-Everything (TOE)**: an intuitive method that re-initializes the meta-knowledge at each time t and trains on all the arriving data $\mathcal{D}_{1:t}$; (2) **Train-From-Scratch (TFS)**: another intuitive method that also re-initializes the meta-knowledge at each time t but trains only on the current data $\mathcal{D}_t$; (3) **Follow the Meta Leader (FTML)**(Finn et al., 2019): a method utilizing the Follow the Leader algorithm (Kalai & Vempala, 2005) to minimize the regret of meta-learner. (4) **Online Structured Meta-Learning (OSML)**(Yao et al., 2020): a method via conducting a pathway to extract meta-knowledge from a meta-hierarchical graph; (5) **Dirichlet Process Mixture Model (DPMM)**(Jerfel et al., 2019): an algorithm that employs CRP to conduct a mixture meta-knowledge using point estimation; (6) **Bayesian Online Meta-Learning with Variational Inference (BOMVI)**(Yap et al., 2021): a method that uses Bayesian meta-learning to address the catastrophic forgetting issue; (7) **Variational Continual Bayesian Meta-Learning (VC-BML)**(Zhang et al., 2021): a state-of-the-art method that aims to conduct a mixture meta-knowledge via a Bayesian method.

Following exiting works (Yap et al., 2021; Zhang et al., 2021), we conduct the experiments on four datasets: *VGG-Flowers*(Nilsback & Zisserman, 2008), *miniImagenet*(Ravi & Larochelle, 2017), *CIFAR-FS*(Bertinetto et al., 2018), and *Omniglot*(Lake et al., 2011). Tasks sampled from different datasets correspond to different task distribution, so that the continual non-stationary environment can be created via chronologically sampling tasks from different datasets. Specifically, the sampled task is a 5-way 5-shot task, and 5 classes are sampled randomly from a dataset for a task. In our experiment, we sequentially meta-train the model on tasks sampled from the meta-training dataset of these four datasets, which means that the model is first trained on the tasks sampled from *VGG-Flowers* dataset, and then proceeds to the next dataset. The performance is evaluated on the test set after tuning hyper-parameters on the validation set. More details about experimental and hyperparameter settings are in Appendix F. Our code is publicly available [2].

5.1. RQ1: Overall Performance Comparison

To examine the effectiveness of ACML, we present the mean meta-test accuracy on all the learned datasets at each meta-training stage in Tab. 1, and the details on each training stage are in Appendix F.4.1. Our ACML achieves the best performance at each meta-training stage (i.e., *VGG-Flowers*, *miniImagenet*, *CIFAR-FS* and *Omniglot*). Particularly, at the final stage (i.e., *Omniglot*), our ACML achieves an average performance improvement from nearly 2% (versus VC-BML) to 10% (versus FTML). This comparison result illustrates that ACML is more effective to capture incre-

[1] There normalization terms can be omitted.

[2] https://github.com/BinWu-Cs/AC-CML

Table 1. Mean meta-test accuracy (%) with 95% confidence interval of the learned dataset at each meta-training stage. The best performance is marked with boldface.

Algorithms	VGG-Flowers	miniImagenet	CIFAR-FS	Omniglot
FTML	76.84 ± 1.75	60.74 ± 1.85	66.71 ± 1.86	61.89 ± 1.49
OSML	79.61 ± 1.50	66.15 ± 1.73	68.24 ± 1.73	65.65 ± 1.40
DPMM	78.97 ± 1.52	66.55 ± 1.77	67.18 ± 1.86	68.26 ± 1.47
BOMVI	77.05 ± 1.80	60.44 ± 1.86	59.57 ± 1.77	69.04 ± 1.54
VC-BML	83.71 ± 1.58	68.09 ± 1.58	69.87 ± 1.74	69.48 ± 1.51
ACML	$\mathbf{85.11 \pm 1.46}$	$\mathbf{69.45 \pm 1.54}$	$\mathbf{70.72 \pm 1.61}$	$\mathbf{71.46 \pm 1.39}$

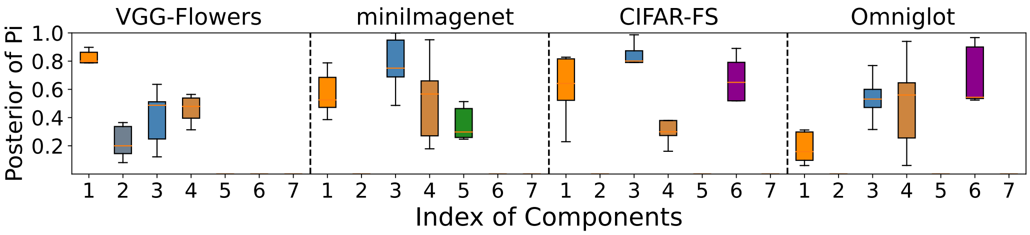


Figure 3. Each column represents the posterior probability π of the Bernoulli distribution of different components in the compositional meta-knowledge on different datasets.

Table 2. Mean meta-test accuracy (%) with 95% confidence interval under the sequential task setting, where the performance represents the average accuracy across the whole training tasks sequence on different dataset. The best performance is marked with boldface.

Algorithms	VGG-Flowers	miniImagenet	CIFAR-FS	Omniglot
FTML	57.29 ± 2.28	31.92 ± 1.58	39.21 ± 1.75	82.03 ± 1.42
OSML	56.07 ± 2.10	$32.41 \pm 1,36$	40.75 ± 1.85	82.89 ± 1.42
DPMM	64.21 ± 2.06	36.68 ± 1.46	47.47 ± 1.88	88.39 ± 1.48
BOMVI	64.71 ± 1.78	38.44 ± 1.41	48.19 ± 1.88	90.49 ± 1.62
VC-BML	65.28 ± 2.19	38.65 ± 1.83	47.07 ± 1.75	89.97 ± 1.11
ACML	$\mathbf{66.27 \pm 2.01}$	$\mathbf{40.04 \pm 1.48}$	$\mathbf{48.97 \pm 1.71}$	$\mathbf{91.13 \pm 0.20}$

mental knowledge as well as addressing the catastrophic forgetting issue. Moreover, the comparison between the performance of ACML and the baselines (i.e., DPMM and VC-CML), which maintain the mutually exclusive meta-knowledge components, confirming that our compositional meta-knowledge helps to improve performance via meta-knowledge sharing among tasks.

To further illustrate the association between tasks and meta-knowledge, we show each component's posterior of the Bernoulli distribution. As in Fig. 3, the probabilities of the Bernoulli distribution of each component are distinct. Moreover, the components are dynamically changing. For example, the fifth component is added at the *miniImagenet* stage and is filtered as redundant components since it only receives support from the *miniImagenet* and the support is not strong. As for the first component, although it is not strongly relevant to the final dataset, it remains because it contributes to the first three datasets. It further confirms that our method can add new components by IBP prior to capture the incremental information, and filters the redundant one using evidential sparsification.

Besides, we consider another more challenging setting, where tasks from different datasets are mixed and randomly arrive one by one. Such a setting only allows accessibility for only one task each time, where the catastrophic forgetting issue is more severe. Tab. 2 shows the average accuracy results of tasks that belong to each dataset, to show the performance on different task distributions. Our ACML achieves the best performance on all four datasets with a performance improvement from nearly 1% to 9%, even in such a challenging setting. It further confirms that our algorithm is still effective on the long task sequence.

5.2. RQ2: The Impact of the Number of Components

To examine the effect of the number of meta-knowledge components in our ACML, we control the increased rate via α in IBP and the sparsification rate γ in evidential sparsification to limit the number of components (details are available in Appendix F.3). We conduct experiments with different

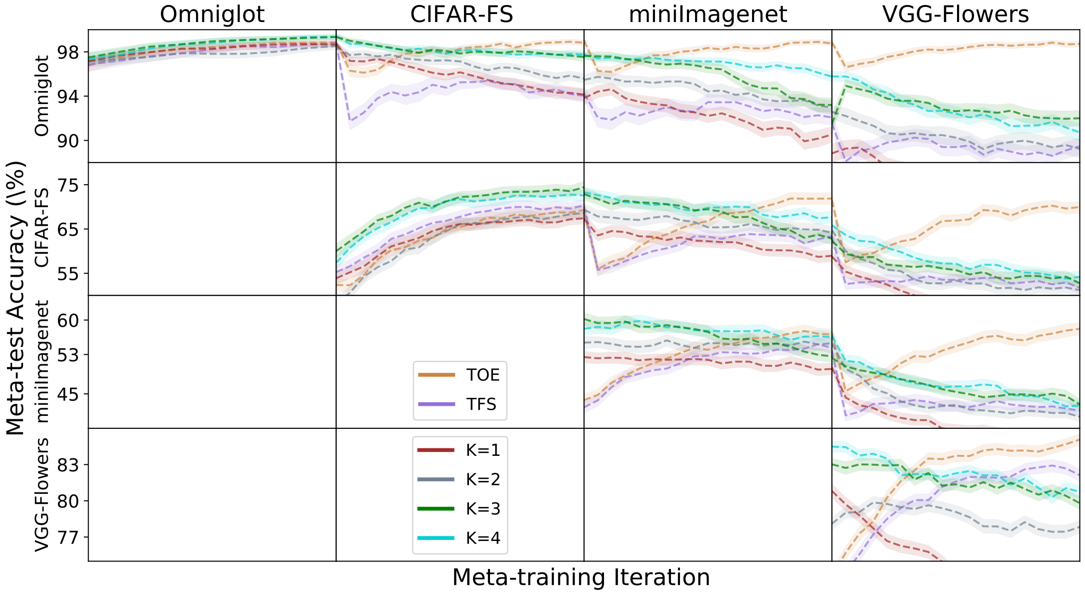
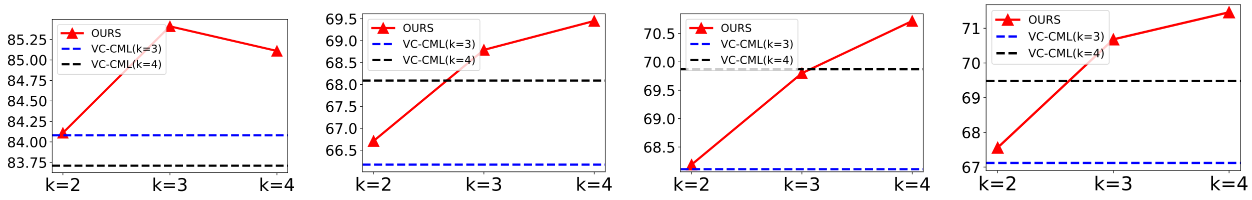


Figure 4. The evolution of meta-test accuracies (%) of ACML with different numbers of components when training on different datasets. TOE and TFS are two baselines for comparison.



(a) VGG-Flowers Stage (b) miniImagenet Stage (c) CIFAR-FS Stage (d) Omniglot Stage

Figure 5. The comparison between ACML and VC-BML with different numbers of components on each training stage.

Table 3. The meta-test accuracy (%) with 95% confidence interval before sparsification and after sparsification on each training stage

	original	sparse
Omniglot	99.31 ± 0.25	99.43 ± 0.21
CIFAR-FS	85.99 ± 1.07	86.53 ± 1.05
miniImagenet	76.07 ± 1.40	75.91 ± 1.38
VGG-Flowers	71.41 ± 1.48	71.22 ± 1.36

numbers of components (i.e., from 1 to 4) to test their effectiveness. The evolution of meta-test accuracy when training on different datasets is shown in Fig. 4. TOE has the best performance on most stages because it has access to all the available data. With the component number increasing, ACML has a better performance in both the learned and the new datasets. It further demonstrates that more components can capture the incremental meta-knowledge and alleviate the forgetting issue. However, when the component number increases from 3 to 4, the performance does not see a large improvement. This result reveals that the only ever-increasing number of components is not always helpful to alleviate the forgetting issue, and confirms the necessity of filtering the redundant components.

5.3. RQ3: The Effectiveness of Evidential Specification

To examine the impact of the evidential sparsification method, we compare performance before and after sparsification. The mean meta-test accuracy at each meta-training stage is shown in Tab. 3, and more results are shown in Appendix F.4.2. Compared to the full meta-knowledge, the sparse meta-knowledge at each time can achieve a comparative and even better performance in some stages (i.e., *Omniglot* and *CIFAR-FS*). This confirms that our method can reduce redundancy and computational cost without hurting model performance. We also observe that the datasets where the sparse meta-knowledge outperforms the original meta-knowledge are located in the initial stages, i.e., *Omniglot* and *CIFAR-FS*). We analyze it due to the fact that, in the beginning, the capacity of the components is adequate to deal with the existing occurring tasks, especially for the simple tasks (i.e., Omniglot). The sparsification is helpful to remove the redundant components that might be distractions or even noise to the occurring tasks. Moreover, we conduct experiments on different numbers of components with the appropriate γ. The results in Fig. 5 show that our model can outperform the SOTA baselines even with less number of components. For example, our ACML with three components achieves a competitive performance and even has a better performance, compared to VC-CML with four components. It confirms that our algorithm can filter out the redundant meta-knowledge component and is more parameter-efficiency.

6. Conclusion

This paper focuses on continual meta-learning, where tasks from a non-stationary distribution are sequentially available. We propose ACML, an Adaptive Compositional Meta-Learning algorithm that allows heterogeneous tasks to share meta-knowledge, which improves effectively the parameter efficiency. Moreover, an IBP prior is employed to determine whether to increase the number of components in the mixture meta-knowledge distribution, and an evidential sparsity method is proposed to adaptively filter out the redundant components so as to meet the actual need of all available tasks. The conducted experiments show the effectiveness of compositional meta-knowledge and confirm that our algorithm can learn the required meta-knowledge from tasks.

One limitation comes from the space complexity since our model still needs to increase the number of mixture components to cover more meta-knowledge. The proposed evidential sparsity method can help alleviate the required space complexity to a certain dedgree.

Acknowledgements

This work is supported by CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2022-052A), MBZUAI and MBZUAI-WIS funds. We want to express gratitude to the anonymous reviewers for their hard work and kind comments and Hangzhou AI Computing Center for their technical support.

References

Amit, R. and Meir, R. Meta-learning by adjusting priors based on extended pac-bayes theory. In *International Conference on Machine Learning*, pp. 205–214. PMLR, 2018.

Benjamin, A., Rolnick, D., and Kording, K. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2018.

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2018.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Bronskill, J., Gordon, J., Requeima, J., Nowozin, S., and Turner, R. Tasknorm: Rethinking batch normalization for meta-learning. In *International Conference on Machine Learning*, pp. 1153–1164. PMLR, 2020.

Chen, P., Itkina, M., Senanayake, R., and Kochenderfer, M. J. Evidential softmax for sparse multimodal distributions in deep generative models. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 11565–11576, 2021.

Conklin, H., Wang, B., Smith, K., and Titov, I. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3322–3335, 2021.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Dempster, A. P. A generalization of bayesian inference. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pp. 73–104. Springer, 2008.

Denevi, G., Stamos, D., Ciliberto, C., and Pontil, M. Online-within-online meta-learning. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 13089–13099, 2019.

Denoeux, T. A k-nearest neighbor classification rule based on dempster-shafer theory. In *Classic works of the Dempster-Shafer theory of belief functions*, pp. 737–760. Springer, 2008.

Denœux, T. Logistic regression, neural networks and dempster–shafer theory: A new perspective. *Knowledge-Based Systems*, 176:54–67, 2019.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 439–450, 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9537–9548, 2018.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

Gordon, J., Bronskill, J., Bauer, M., Turner, R. E., Stühmer, J., and Nowozin, S. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*, 2019.

Griffiths, T. L. and Ghahramani, Z. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12(4), 2011.

Ha, D., Dai, A., and Le, Q. V. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Hoffman, M. and Blei, D. Stochastic structured variational inference. In *Artificial Intelligence and Statistics*, pp. 361–369. PMLR, 2015.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

Hu, W., Lin, Z., Liu, B., Tao, C., Tao, Z. T., Zhao, D., Ma, J., and Yan, R. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International Conference on Learning Representations*, 2019.

Iakovleva, E., Verbeek, J., and Alahari, K. Meta-learning with shared amortized variational inference. In *International Conference on Machine Learning*, pp. 4572–4582. PMLR, 2020.

Itkina, M., Ivanovic, B., Senanayake, R., Kochenderfer, M. J., and Pavone, M. Evidential sparsification of multimodal latent spaces in conditional variational autoencoders. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Jerfel, G., Grant, E., Griffiths, T., and Heller, K. A. Reconciling meta-learning and continual learning with online mixtures of tasks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9119–9130, 2019.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Kessler, S., Nguyen, V., Zohren, S., and Roberts, S. J. Hierarchical indian buffet neural networks for bayesian continual learning. In *Uncertainty in Artificial Intelligence*, pp. 749–759. PMLR, 2021.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kumar, A., Chatterjee, S., and Rai, P. Bayesian structural adaptation for continual learning. In *International Conference on Machine Learning*, pp. 5850–5860. PMLR, 2021.

Laha, A., Chemmengath, S. A., Agrawal, P., Khapra, M. M., Sankaranarayanan, K., and Ramaswamy, H. G. On controllable sparse alternatives to softmax. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 6423–6433, 2018.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Lee, S., Kim, J., Jun, J., Ha, J., and Zhang, B. Overcoming catastrophic forgetting by incremental moment matching. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4652–4662, 2017.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Martins, A. and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pp. 1614–1623. PMLR, 2016.

Mendez, J. A. and EATON, E. Lifelong learning of compositional structures. In *International Conference on Learning Representations*, 2021.

Munkhdalai, T. and Yu, H. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563. PMLR, 2017.

Nalisnick, E. T. and Smyth, P. Stick-breaking variational autoencoders. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Oreshkin, B. N., López, P. R., and Lacoste, A. TADAM: task dependent adaptive metric for improved few-shot learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 719–729, 2018.

Pan, P., Swaroop, S., Immer, A., Eschenhagen, R., Turner, R. E., and Khan, M. E. Continual deep learning by functional regularisation of memorable past. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Ravi, S. and Beatson, A. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2018.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Shafer, G. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4077–4087, 2017.

Teh, Y., Jordan, M., Beal, M., and Blei, D. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17, 2004.

Titsias, M. K., Schwarz, J., de G. Matthews, A. G., Pascanu, R., and Teh, Y. W. Functional regularisation for continual learning using gaussian processes. *CoRR*, abs/1901.11356, 2019.

Vanschoren, J. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

Wu, B., Meng, Z., Zhang, Q., and Liang, S. Meta-learning helps personalized product search. In *Proceedings of the ACM Web Conference 2022*, pp. 2277–2287, 2022.

Yager, R. R. and Liu, L. *Classic works of the Dempster-Shafer theory of belief functions*, volume 219. Springer, 2008.

Yao, H., Wei, Y., Huang, J., and Li, Z. Hierarchically structured meta-learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97, pp. 7045–7054. PMLR, 2019.

Yao, H., Zhou, Y., Mahdavi, M., Li, Z., Socher, R., and Xiong, C. Online structured meta-learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yap, P., Ritter, H., and Barber, D. Addressing catastrophic forgetting in few-shot problems. In *International Conference on Machine Learning*, pp. 11909–11919. PMLR, 2021.

Zhang, Q., Fang, J., Meng, Z., Liang, S., and Yilmaz, E. Variational continual bayesian meta-learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24556–24568, 2021.

Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

Zhuang, Z., Wang, Y., Yu, K., and Lu, S. No-regret non-convex online meta-learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3942–3946. IEEE, 2020.

A. Variational Inference for Meta-Learning

Following MAML (Finn et al., 2017), many Bayesian variant (Ravi & Beatson, 2018; Gordon et al., 2019; Iakovleva et al., 2020) are proposed. To fit well with the bi-level optimization architecture, most of them consider a hierarchical bayesian inference (Amit & Meir, 2018), where the Evidence Lower Bound (ELBO) of likelihood can be derived as follows:

$$\begin{aligned}
&\log\left[\prod_{i=1}^{T} p(\mathcal{D}_i)\right] \qquad (15)\\
&=\log\left[\int p(\boldsymbol{\theta})\left[\prod_{i=1}^{T}\int p(\mathcal{D}_i|\boldsymbol{\phi}_i)p(\boldsymbol{\phi}_i|\boldsymbol{\theta})d\boldsymbol{\phi}_i\right]d\boldsymbol{\theta}\right]\\
&\geq\mathbb{E}_{q(\boldsymbol{\theta};\psi)}\left[\log\left(\prod_{i=1}^{T}\int p(\mathcal{D}_i|\boldsymbol{\phi}_i)p(\boldsymbol{\phi}_i|\boldsymbol{\theta})d\boldsymbol{\phi}_i\right)\right]\\
&\quad-D_{KL}(q(\boldsymbol{\theta};\psi)||p(\boldsymbol{\theta}))\\
&\geq\mathbb{E}_{q(\boldsymbol{\theta};\psi)}\left[\sum_{i=1}^{T}\mathbb{E}_{q(\boldsymbol{\phi}_i;\lambda_i)}\left[\log p(\mathcal{D}_i|\boldsymbol{\phi}_i)\right]\right.\\
&\quad\left.-D_{KL}(q(\boldsymbol{\phi}_i;\lambda)||p(\boldsymbol{\phi}_i|\boldsymbol{\theta}))\right]-D_{KL}(q(\boldsymbol{\theta};\psi)||p(\boldsymbol{\theta})),
\end{aligned}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}=\{\boldsymbol{\phi}_i\}$ are the global parameter and task-specific parameter, respectively. Note that the low bound is derived based on the Jensen's equation and the variational distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, i.e., $q(\boldsymbol{\theta};\psi)$ and $q(\boldsymbol{\phi}_i;\lambda_i)$, are introduced to approximate the intractable posterior. Then, the bi-level optimization is transformed as:

$$\begin{aligned}
\boldsymbol{\phi}^*,\lambda^*=&\arg\max_{\psi,\lambda}\mathbb{E}_{q(\boldsymbol{\theta};\psi)}\left[\sum_{i=1}^{T}\mathbb{E}_{q(\boldsymbol{\phi}_i;\lambda_i)}\left[\log p(\mathcal{D}_i|\boldsymbol{\phi}_i)\right]\right.\\
&\left.-D_{KL}(q(\boldsymbol{\phi}_i;\lambda)||p(\boldsymbol{\phi}_i|\boldsymbol{\theta}))\right]\\
&-D_{KL}(q(\boldsymbol{\theta};\psi)||p(\boldsymbol{\theta})), \qquad (16)
\end{aligned}$$

where the goal of the optimization is to seek the optimal variational distribution of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, parameterized by ψ and λ, respectively.

B. Introduction in Evidential Theory

B.1. The Basic Definition

Evidential theory (Denœux, 2019) works on a discrete set of hypotheses (or equivalently, components of meta-knowledge in this paper). Let $Z=\{\boldsymbol{z}_1,\boldsymbol{z}_2,\boldsymbol{z}_3,...,\boldsymbol{z}_K\}$ be a finite set, the element of which $\boldsymbol{z}_k$ is a binary variable indicating whether the current task is associated with k-th component or not, and the power set of Z, denoted by $2^Z=\{\emptyset,\{\boldsymbol{z}_1\},\{\boldsymbol{z}_1,\boldsymbol{z}_2\},...,Z\}$. A *mass function* on Z is a mapping m: $2^Z\to[0,1]$ and satisfies the following constraints:

$$m(\emptyset)=0,\quad\sum_{A\subseteq Z}m(A)=1, \qquad (17)$$

where $\emptyset$ is an empty set and A is a subset of Z. The mass function $m(\cdot)$ represents the support to each potential subset of components, and any subset A is called *focal set* if $m(A)>0$. As a particular case, the vacuous mass function (i.e., $m(Z)=1$) indicates that it does not focus on any subset in the case of complete ignorance. One mass function is said *simple* when:

$$m(A)=s,\quad m(Z)=1-s,\quad w=-\ln(1-s), \qquad (18)$$

where A is a single strict subset $A\subset Z$, $s\in[0,1]$ represents the support degree of A, and w denotes the *evidential weight* of A.

Given a mass function, there are two corresponding functions, called *belief and plausibility function*, respectively, which are defined as follows:

$$Bel(A)=\sum_{B\subseteq A}m(B), \qquad (19)$$

$$Pl(A)=\sum_{C\cap A\neq\emptyset}m(C)=1-Bel(\bar{A}), \qquad (20)$$

where $\bar{A}$ denotes the complemented set of A. $Bel(A)$ can be interpreted as the total support degree to A, while the $1-Pl(A)$ can be interpreted as the total doubt degree to A. Besides, when the plausibility function is restricted to singletons, i.e., a single element $\boldsymbol{z}_k$ of Z, then it is called *contour function* $Pl:\boldsymbol{z}_k\to[0,1]$.

B.2. Dempster's Rule

Given two mass functions m_1 and m_2, their combination is defined according to Dempster's rule:

$$(m_1\oplus m_2)(A)=\frac{1}{1-\kappa}\sum_{B\cap C=A}m_1(B)\cdot m_2(C), \qquad (21)$$

where κ is the degree of conflict between two evidences, which is defined as:

$$\kappa=\sum_{B\cap A=\emptyset}m_1(B)\cdot m_2(C). \qquad (22)$$

Note that Dempster's rule for the combination of mass functions is commutative and associative. Based on Dempster's rule for the combination between two mass functions, the combination of two corresponding contour functions pl_1 and pl_2 can be computed as:

$$(pl_1\oplus pl_2(\boldsymbol{z}_k))=\frac{pl_1(\boldsymbol{z}_k)\cdot pl_2(\boldsymbol{z}_k)}{1-\kappa}. \qquad (23)$$

And if both mass functions are simple with the same strict subset, their fusion can be defined as:

$$A^{w_1} \oplus A^{w_1} = A^{w_1+w_2}, \tag{24}$$

where both A^{w_1} and A^{w_2} represent the simple mass functions with a single strict subset and their evidential weights are w_1 and w_2, respectively.

C. The Computational Details of Fusing Mass Function

We try to combine all the positive mass functions and all the negative mass functions, respectively. And then the two can be fused to produce the final result.

C.1. The Fusion Across Time

Before positive fusion and negative fusion, we need to merge evidence supporting the same focal elements at different times. Since the simple mass functions have the same focal set, their fusion can be calculated following Eq. 24 and the weight is:

$$w_k^+ = \sum_{i=0}^{t} w_{i,k}^+, \quad w_k^- = \sum_{i=0}^{t} w_{i,k}^- \tag{25}$$

where $w_{i,k}^+$ and $w_{i,k}^-$ are the evidential weight of the positive and negative mass function at time i. respectively. In this way, the evidence supporting the same focal element from different time can be merged first:

$$m_k^+(\{\boldsymbol{z}_k\}) = 1 - exp(-w_k^+), \;\; m_k^+(Z) = exp(-w_k^+); \tag{26}$$

$$m_k^-(\overline{\{\boldsymbol{z}_k\}}) = 1 - exp(-w_k^-), \;\; m_k^-(Z) = exp(-w_k^-). \tag{27}$$

C.2. The Fusion of m^+

As we define above, all the positive mass functions have the only two focal elements, $\{\boldsymbol{z}_k\}$ and Z. Then the combination of them can be computed according to the Dempster's rule:

$$\begin{aligned} m^+(\{\boldsymbol{z}_k\}) &\propto [1 - exp(-w_k^+)] \prod_{l \neq k} exp(-w_k^+) \\ &= [exp(w_k^+) - 1] \prod_{l=1}^{K} exp(-w_k^+), \end{aligned} \tag{28}$$

$$m^+(Z) \propto \prod_{k=1}^{K} exp(-w_k^+). \tag{29}$$

As the fused mass function constraint to the sum of one, the results can be computed by normalizing the terms. So that the sum of all terms is:

$$\begin{aligned} & m^+(Z) + \prod_{l=1}^{K} m^+(\{\boldsymbol{z}_k\}) \\ & \propto \left(\prod_{k=1}^{K} exp(-w_k^+) \right) \\ & \quad + \sum_{k=1}^{K} \left\{ [exp(w_k^+) - 1] \prod_{l=1}^{K} exp(-w_k^+) \right\} \\ & = \left(\prod_{k=1}^{K} exp(-w_k^+) \right) \cdot \left[\left(\sum_{k=1}^{K} exp(w_k^+) \right) - K + 1 \right]. \end{aligned} \tag{30}$$

And the terms can be normalized as:

$$\begin{aligned} & m^+(\{\boldsymbol{z}_k\}) \\ & = \frac{[exp(w_k^+) - 1] \prod_{l=1}^{K} exp(-w_k^+)}{\left(\prod_{k=1}^{K} exp(-w_k^+) \right) \cdot \left[\left(\sum_{k=1}^{K} exp(w_k^+) \right) - K + 1 \right]} \\ & = \frac{exp(w_k^+) - 1}{\left(\sum_{k=1}^{K} exp(w_k^+) \right) - K + 1}, \end{aligned} \tag{31}$$

$$\begin{aligned} & m^+(Z) \\ & = \frac{\prod_{k=1}^{K} exp(-w_k^+)}{\left(\prod_{k=1}^{K} exp(-w_k^+) \right) \cdot \left[\left(\sum_{k=1}^{K} exp(w_k^+) \right) - K + 1 \right]} \\ & = \frac{1}{\left(\sum_{k=1}^{K} exp(w_k^+) \right) - K + 1}. \end{aligned} \tag{32}$$

C.3. The Fusion of m^-

Different from the positive mass functions, the negative mass functions have the only two focal elements, $\overline{\{\boldsymbol{z}_k\}}$ and Z. To compute the combination of all negative mass functions, we need to compute the conflict firstly:

$$\kappa^- = \prod_{k=1}^{K} \left(1 - exp(-w_k^-) \right). \tag{33}$$

Thus, for any strict subset A of Z, its belief can be computed as:

$$\begin{aligned} & m^-(A) \\ & = \frac{\left[\prod_{\boldsymbol{z}_k \notin A} \left(1 - exp(-w_k^-) \right) \right] \cdot \left[\prod_{\boldsymbol{z}_k \in A} exp(-w_k^-) \right]}{1 - \prod_{k=1}^{K} \left(1 - exp(-w_k^-) \right)}. \end{aligned} \tag{34}$$

And the mass belief of the complete set Z is:

$$m^-(Z) = \frac{\prod_{k=1}^{K} exp(-w_k^-)}{1 - \prod_{k=1}^{K} \left(1 - exp(-w_k^-) \right)}. \tag{35}$$

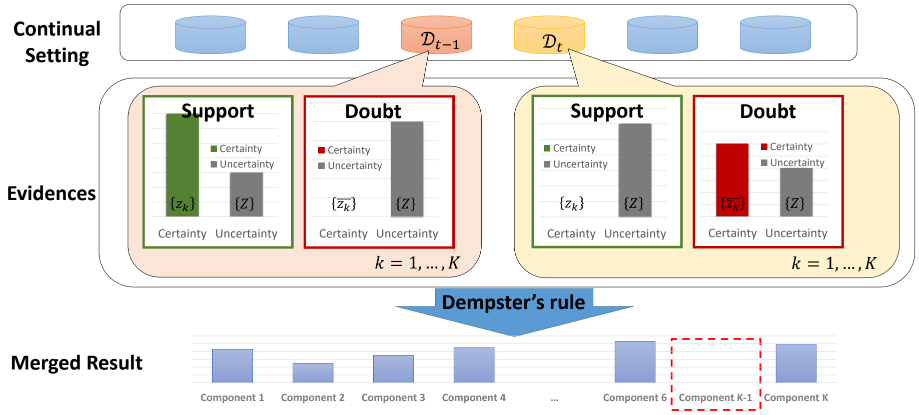


Figure 6. An intuitive explanation of our proposed evidential sparsification method. The tasks from the previous and current provide support and doubt information with uncertainty for meta-knowledge components. Such information can be merged by Dempster's rule to provide a unified relationship between the occurring tasks and the components. The components not receiving support are removed (i.e., the component $K-1$ in the figure).

For further fusion of the positive and negative mass functions, we need to compute $pl^-(\boldsymbol{z}_k)$, which can be defined as:

$$pl^-(\{\boldsymbol{z}_k\}) = \frac{\prod_{k=1}^K pl_k^-(\{\boldsymbol{z}_k\})}{1 - \prod_{k=1}^K \left(1 - exp(-w_k^-)\right)}, \tag{36}$$

where the plausibility of negative mass function is:

$$pl_l^-(\{\boldsymbol{z}_k\}) = \begin{cases} exp(-w_l^-) & if\ k = l \\ 1 & otherwise \end{cases}. \tag{37}$$

Thus, the result of the fused plausibility is:

$$pl^-(\{\boldsymbol{z}_k\}) = \frac{exp(-w_k^-)}{1 - \prod_{k=1}^K \left(1 - exp(-w_k^-)\right)}. \tag{38}$$

C.4. The Final Fusion

To clarify the following derivation, we assume that:

$$C^+ = \frac{1}{\left(\sum_{k=1}^K exp(w_k^+)\right) - K + 1}, \tag{39}$$

$$C^- = \frac{1}{1 - \prod_{k=1}^K \left(1 - exp(-w_k^-)\right)}. \tag{40}$$

Similarly, to combine the positive and negative mass function, we need to compute the conflict between them at first:

$$\begin{aligned} \kappa &= \sum_{k=1}^K \left\{ m^+(\{\boldsymbol{z}_k\}) \left[\sum_{\boldsymbol{z}_k \notin A} m^-(A) \right] \right\} \\ &= \sum_{k=1}^K \left\{ m^+(\{\boldsymbol{z}_k\}) \cdot \left[1 - pl^-(\{\boldsymbol{z}_k\})\right] \right\} \\ &= \sum_{k=1}^K \left\{ C^+ \left[exp(w_k^+) - 1\right] \cdot \left[1 - C^-(exp(-w_k^-))\right] \right\}, \end{aligned} \tag{41}$$

where $A \subseteq Z$. To make the following derivation clarified, let:

$$\begin{aligned} C &= \frac{1}{1-\kappa} \\ &= \frac{1}{1 - \sum_{k=1}^K \left\{ C^+ \left[e^{w_k^+} - 1\right] \cdot \left[1 - C^-(e^{-w_k^-})\right] \right\}}. \end{aligned} \tag{42}$$

Then for any $k \in \{1, 2, ..., K\}$, the mass belief of each singleton can be computed as:

$$m(\{\boldsymbol{z}_k\}) \tag{43}$$

$$\begin{aligned} =& C \left\{ m^+(\{\boldsymbol{z}_k\}) \cdot \left[\sum_{\boldsymbol{z}_k \in A} m^-(A) \right] \right. \\ & \left. + m^+(Z) \cdot m^-(\{\boldsymbol{z}_k\}) \right\} \\ =& C \left\{ m^+(\{\boldsymbol{z}_k\}) \cdot pl^-(\{\boldsymbol{z}_k\}) + m^+(Z) \cdot m^-(\{\boldsymbol{z}_k\}) \right\}, \end{aligned} \tag{44}$$

where $A \subseteq Z$. Combining Eq. 31, Eq. 32 Eq. 34 and Eq. 38, , the final result of the mass singleton belief is:

$$\begin{aligned} & m(\{\boldsymbol{z}_k\}) \\ =& C \left\{ C^+ \left[e^{w_k^+} - 1\right] \cdot C^- [e^{-w_k^-}] \right. \\ & \left. + C^+ \cdot C^- \left[e^{-w_k^-} \cdot \prod_{l \neq k} \left(1 - e^{-w_l^-}\right) \right] \right\} \\ =& CC^+C^- \left\{ e^{-w_k^-} \left[e^{w_k^+} - 1 + \prod_{l \neq k} (1 - e^{-w_l^-}) \right] \right\}. \end{aligned} \tag{45}$$

D. Details of Inference

In this section, we present the details of our structured variational inference for our proposed ACML. The pseudo-code

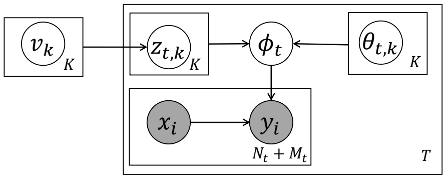


Figure 7. The probability model of ACML. The solid line denotes the generative process, and the white circle and the grey circle denote the latent variant and the observed variant, respectively.

and the probability model are shown in Alg.1 and Fig. 7, respectively.

D.1. Variational Distribution

Because of the intractability of posterior, we introduce the variational distribution to approximate the true posterior (seen in Eq. (7)). The variational distributions are parameterized as:

$$q(\boldsymbol{v}_{t,k}) = Beta(\alpha_{t,k}, \beta_{t,k}), \tag{46}$$

$$q(\boldsymbol{z}_{t,k}|\boldsymbol{\pi}_{t,k}) = Bern(\boldsymbol{\pi}_{t,k}), \quad \text{where } \boldsymbol{\pi}_{t,k} = \boldsymbol{v}_{t,k}, \tag{47}$$

$$q(\boldsymbol{\theta}_{t,k}) = \mathcal{N}(\mu_{t,k}, \sigma^2_{t,k}\mathbb{1}), \tag{48}$$

$$q(\boldsymbol{\phi}_t|\boldsymbol{\theta}_t, \boldsymbol{z}_t, \mathcal{D}_t) = \frac{\sum_{k=1}^{K} \mathbb{1}\{\boldsymbol{z}_{t,k} \neq 0\} q(\boldsymbol{\phi}_{t,k}|\boldsymbol{\theta}_{t,k}; \lambda_{t,k})}{\sum_{k=1}^{K} \mathbb{1}\{\boldsymbol{z}_{t,k} \neq 0\}}, \tag{49}$$

where $\lambda_{t,k} = SGD_J(\boldsymbol{\theta}^*_{t,k}, \mathcal{D}^S_t, \epsilon)$, and $SGD_J(\cdot)$ represents the stochastic gradient descent with J steps. That is, the required variational parameters are $\psi_t = \{\alpha_{t,k}, \beta_{t,k}, \mu_{t,k}, \sigma_{t,k}, \lambda_{t,k}\}$ for all $k = 1, ..., K_t$. Note that we replace $\prod_{i=1}^{k} \boldsymbol{v}_{t,i}$ with $\boldsymbol{v}_{t,k}$ in the posterior, to remove the implicit order constraint in the prior. So that the optimization aims to search for the optimal variational parameter to maximize the ELBO in Eq. 8.

We summarize the meta-training process of our proposed ACML. Note that we assume the algorithm uses a single batch for simplicity, and it can be easily extended to the mini-batch setting. The pseudo-code is in Alg.1.

D.2. Reparameterization

The variational posterior is obtained by optimizing the ELBO using structured variational inference. To make inference tractable, we utilize three reparameterizations, to infer the Gaussian distribution, beta distribution and Bernoulli distribution, respectively.

D.2.1. The variational Gaussian distribution reparameterization

As we mentioned above, the variational distributions of meta-knowledge from each clusters are diagonal Gaussian $\boldsymbol{\theta}_{t,k} \sim \mathcal{N}(\mu_{t,k}, \sigma_{t,k})$. We employ the reparameterization, which can represent the meta-knowledge using a deterministic function $\boldsymbol{\theta}_{t,k} = g(\varepsilon; \mu_{t,k}, \sigma_{t,k})$, where $\epsilon \sim \mathcal{N}(0, I)$. To apply the reparameterization, we define the standardization function and its inverse as:

$$\mathcal{S}_\psi(\boldsymbol{\theta}) = \frac{\boldsymbol{\theta} - \mu}{\sigma} = \varepsilon \sim q(\varepsilon), \quad \text{where } q(\varepsilon) = \mathcal{N}(0, I),$$
$$\boldsymbol{\theta} = \mathcal{S}^{-1}_{\boldsymbol{\phi}}(\varepsilon) = \varepsilon \cdot \sigma + \mu. \tag{50}$$

Note that we omit the subscripts for clarity and the remainder of this section omits them as well. Then we can represent the objective in ELBO w.r.t $q(\boldsymbol{\theta})$ as follows:

$$\mathbb{E}_{q_\psi(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] = \mathbb{E}_{q(\varepsilon)}[f(\mathcal{S}^{-1}_\psi(\varepsilon))]. \tag{51}$$

This allows us to compute the gradient of the expectation in another way:

$$\nabla_\psi \mathbb{E}_{q_\psi(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] = \mathbb{E}_{q(\varepsilon)}[\nabla_\psi f(\mathcal{S}^{-1}_\psi(\varepsilon))]$$
$$= \mathbb{E}_{q(\varepsilon)}[\nabla_{\boldsymbol{\theta}} f(\mathcal{S}^{-1}_\psi(\varepsilon)) \nabla_\psi \mathcal{S}^{-1}_\psi(\varepsilon)], \tag{52}$$

D.2.2. The variational Beta distribution reparameterization

There is no simple inverse of the standardization function when using the reparameterization for Beta distribution, which makes it impossible to apply the explicit reparameterization directly. Instead, there are two ways to tackle the problem: the implicit reparameterization and the Kumaraswamy reparameterization.

Implicit reparameterization. This way also utilizes the reparameterization to tackle the intractable gradient in Beta distribution:

$$\nabla_\gamma \mathbb{E}_{q_\gamma(\boldsymbol{v}_k)}[f(\boldsymbol{v}_k)] = \mathbb{E}_{q(\varepsilon)}[\nabla_\gamma f(\boldsymbol{v}_k)] \tag{53}$$
$$= \mathbb{E}_{q(\varepsilon)}[\nabla_{\boldsymbol{v}_k} f(\boldsymbol{v}_k) \nabla_\gamma \boldsymbol{v}_k],$$

without the inverse of the standardization function, the term $\nabla_\gamma \boldsymbol{v}_k$ is difficult to compute. Inspired by (Figurnov et al., 2018), we employ the implicit reparameterization to compute the gradient, the idea of which is to differentiate the standardization function $\mathcal{S}_\gamma(\boldsymbol{v}_k) = \varepsilon$ using the chain rule instead of searching its inverse:

$$\nabla_{\boldsymbol{v}_k} \mathcal{S}_\gamma(\boldsymbol{v}_k) \nabla_\gamma(\boldsymbol{v}_k) + \nabla_\gamma \mathcal{S}_\gamma(\boldsymbol{v}_k) = \mathbf{0}, \tag{54}$$

$$\nabla_\gamma \boldsymbol{v}_k = -(\nabla_{\boldsymbol{v}_k} \mathcal{S}_\gamma(\boldsymbol{v}_k))^{-1} \nabla_\gamma \mathcal{S}_\gamma(\boldsymbol{v}_k). \tag{55}$$

Note that the standardization function can be the CDF of the Beta distribution and $\varepsilon \sim Unif[0, 1]$. Then the implicit gradient is:

$$\nabla_\gamma \boldsymbol{v}_k = \frac{\nabla_\gamma F(\boldsymbol{v}_k; \gamma)}{-(\nabla_{\boldsymbol{v}_k} F(\boldsymbol{v}_k; \gamma))} = \frac{\nabla_\gamma F(\boldsymbol{v}_k; \gamma)}{-p(\boldsymbol{v}_k; \gamma)}, \tag{56}$$

where $p(\boldsymbol{v}_k; \gamma)$ is the PDF of the Beta distribution.

Algorithm 1 The meta-training process of ACML.

```
Input: Task distribution $p(\tau)$, data distribution $p(\mathcal{D}|\tau)$,
    the initial number of component $K_0$, concentration parameter $\alpha$,
    the number of inner update step $J$, the inner learning rate $\epsilon$,
    and the outer learning rate $\zeta$
for t=1,.. do
  Determine the added number: $J_t = Possion(\frac{\alpha}{t})$
  Determine the number of component: $K_t = K_{t-1} + J_t$
  Initialize the variational beta distribution: $\alpha_{t,k}, \beta_{t,k}, \forall k = 1, ..., K_t$
  Initialize the variational distribution of meta-knowledge: $\mu_k, \sigma_k, \forall k = 1, .., K_t$
  while not converge do
    Sample $\boldsymbol{v}_{t,k} \sim q(\boldsymbol{v}_{t,k}; \alpha_{t,k}, \beta_{t,k}), \forall k = 1, ..., K_t$
    Compute the ELBO according to Eq. 8
    Compute the gradient: $\nabla\mu_{t,k}, \nabla\sigma_{t,k}, \forall k = 1, ..., K_t$ via explicit reparameterization according to Eq. 52
    Compute the gradient: $\nabla\alpha_{t,k}, \nabla\beta t, k, \forall k = 1, ..., K_t$ via implicit reparameterization according to Eq. 56
    Update the variational parameters: $\alpha_{t,k} \leftarrow \alpha_{t,k} - \zeta\nabla\alpha_{t,k}, \forall k = 1, ..., K_t$
    Update the variational parameters: $\beta_{t,k} \leftarrow \beta_{t,k} - \zeta\nabla\beta_{t,k}, \forall k = 1, ..., K_t$
    Update the variational parameters: $\mu_{t,k} \leftarrow \mu_{t,k} - \zeta\nabla\mu_{t,k}, \forall k = 1, ..., K_t$
    Update the variational parameters: $\sigma_{t,k} \leftarrow \sigma_{t,k} - \zeta\nabla\sigma_{t,k}, \forall k = 1, ..., K_t$
  end while
  Update prior: $p(\boldsymbol{v}_{t,k}) \leftarrow q(\boldsymbol{v}_{t,k}), \forall k = 1, ..., K_t$
  Compute the evidential weight $w^+_{t,k}, w^-_{t,k}, \forall k = 1, ..., K_t$ according to Eq. 10
  Compute the mass function according to Eq. 13
  Remove the components without support information according to Eq. 14
end for
```

Kumaraswamy distribution. The Beta distribution of $\boldsymbol{v}_k$ also can be reparameterized using a Kumaraswamy distribution (Nalisnick & Smyth, 2017). The Kumaraswamy distribution can be defined as:

$$p(\boldsymbol{v}_k; \alpha, \beta) = \alpha\beta\boldsymbol{v}_k^{\alpha-1}(1 - \boldsymbol{v}_k^{\alpha})^{\beta-1}, \tag{57}$$

and then the inverse of standardization function can be computed as:

$$\mathcal{S}_\gamma(\boldsymbol{v}_k) = (1 - \varepsilon^{1/\beta})^{1/\alpha}, \ \ where\ \varepsilon \sim Unif[0, 1]. \tag{58}$$

The KL-Divergence between the Kumaraswamy distribution and the Beta distribution in ELBO can be written as:

$$\begin{aligned}
&D_{KL}\left(q(\boldsymbol{v}_k; \alpha_k, \beta_k)||p(v; \alpha, \beta)\right) && (59)\\
&= \frac{\alpha_k - \alpha}{\alpha_k}\left(-\gamma - \Psi(\beta_k) - \frac{1}{\beta_k}\right) + \log \alpha_k\beta_k\\
&+ \log\left[B(\alpha, \beta)\right] - \frac{\beta_k}{1 - \beta_k}\\
&+ (\beta - 1)\beta_k \sum_{m=1}^{\infty} \frac{1}{m + \alpha_k\beta_k} B\left(\frac{m}{\alpha_k}, \beta_k\right), && (60)
\end{aligned}$$

where γ is the Euler constant, $\Psi(\cdot)$ is the digamma function, and $B(\cdot, \cdot)$ is the beta function. Following the existing work (Nalisnick & Smyth, 2017), the above the infinite term in the formula can be approximated using a infinite sum of the first 11 terms.

D.2.3. THE VARIATIONAL BERNOULLI DISTRIBUTION REPARAMETERIZATION

As the Bernoulli distribution is one of the classic discrete distributions, the sampling requires performing an $argmax$ operation. But the $argmax$ operation is not differentiable.

We employ the Concrete distribution (Maddison et al., 2017), also named Gumbel-softmax distribution (Jang et al., 2017), to address the above issue. Then, we can sample a random variable as follows:

$$x_j = \sigma\left(\frac{\log(\boldsymbol{\pi}_k) + \log\left(\frac{u_k}{1-u_k}\right)}{\lambda}\right), \ u \sim U(0, 1), \tag{61}$$

where $\lambda \in (0, \infty)$ is a temperature hyper-parameter, $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{\pi}_k$ is the parameter of the Bernoulli distribution and u_k is sampled from a uniform distribution U. To guarantee a lower bound on the ELBO, both posterior and prior Bernoulli distribution need to be replaced with concrete distribution:

$$\begin{aligned}
&D_{KL}\left[q(\mathbf{z}_t|\boldsymbol{\pi}_{k,t})||p(\mathbf{z}_t|\boldsymbol{\pi}_{k,t})\right]\\
\geq &D_{KL}\left[q(\mathbf{z}_t|\boldsymbol{\pi}_{k,t}, \lambda)||p(\mathbf{z}_t|\boldsymbol{\pi}_{k,t}, \lambda)\right]. && (62)
\end{aligned}$$

E. The Analysis of Complexity

We discuss the computational cost of our proposed ACML as follows, including the time complexity and space com-

Table 4. The convolution neural network architecture in ACML and baselines.

Layers	**Output Size**
Input image	$28 \times 28 \times 3$
The first convolution layers	$14 \times 14 \times 64$
The second convolution layers	$7 \times 7 \times 64$
The third convolution layers	$3 \times 3 \times 64$
The forth convolution layers	$1 \times 1 \times 64$

plexity.

For time complexity, the *de facto* bi-level optimization mechanism in meta-learning requires $O(n^2)$ when updating one meta-knowledge component, where an algorithm with time complexity $O(n)$ is a linear time algorithm. If without any sparsification or constraint on the number of components, it will see an unlimited increase as the existing work (Jerfel et al., 2019; Zhang et al., 2021), and thus the time complexity will be up to $O(n^3)$. If with our evidential sparsification, the number of components will be limited to a small constant C with an appropriate hyperparameter γ in Eq. 9, so that the time complexity will be down to $O(C * n^2) \approx O(n^2)$.

Similarly, as each component of meta-knowledge contains the parameter of the model, its space complexity is $O(n)$. And the total space complexity of models without sparsification will be up to $O(n^2)$ for the unlimited number of meta-knowledge components when encountering many tasks. But our algorithm can alleviate this issue using the evidential sparsification to reduce down to $O(n)$ with an appropriate hyperparameter γ.

F. Details of Experiment

F.1. The details of baselines

For a fair comparison, we use the widely-applied network architecture following (Yap et al., 2021; Zhang et al., 2021). In what follows, we describe the details of the baselines:

TOE: Training-On-Everything method (TOE) is an intuitive method, that re-initializes the meta-knowledge and trains them on all the having arrived datasets at each time. We use the same Bayesian meta-learning architecture as our algorithm. The difference between TOE and ACML is that ACML is only trained on the current dataset at each time instead of all the having arrived dataset in TOE and ACML does not re-initialize the meta-knowledge at each time as what TOE do.

TFS: Train-From-Scratch (TFS) is another intuitive method, which also re-initializes meta-knowledge but only trains them on the current dataset. Similarly, it also uses the same Bayesian meta-learning architecture as our algorithm. The difference between TFS and ACML is that our algorithm maintains the posterior meta-knowledge at last time as the prior at the current time instead of re-initializing them as TFS.

FTML: Follow the Meta Leader (FTML) proposed by (Finn et al., 2019) uses the Follow the Leader algorithm to fill the gap between meta-learning and online learning. However, it assumes that all the having arrived datasets are available, which is memory-consuming and conflicts with the continual meta-learning. For a fair comparison, we only train FTML on the current dataset as same as our algorithm.

OSML: Online Structured Meta-Learning (OSML) (Yao et al., 2020) maintains a meta-hierarchical graph with different knowledge blocks and conducts a meta-knowledge pathway for the encountered new task. However, it employs a well pre-trained convolution network to initialize the model in the original paper. As ACML and other baselines are randomly initialized, it would be unfair to use the original initializing way. Therefore, we also randomly initialize the OSML model.

DPMM: Dirichlet Process Mixture Model (DPMM) (Jerfel et al., 2019) employs a Chinese Restaurant Process to conduct the mixture meta-knowledge with a dynamic number of components. Note that it is not a Bayesian method and employs the point estimation to update the meta-knowledge.

BOMVI: Bayesian Online Meta-Learning with Variational Inference (BOMVI) (Yap et al., 2021) is a state-of-the-art algorithm, which conducts a meta-knowledge distribution to address the catastrophic forgetting issue in continual meta-learning. Similarly, it also employs variational inference to update the meta-knowledge.

VC-BML: Variation Continual Bayesian Meta-Learning (VC-BML) (Zhang et al., 2021) is another state-of-the-art algorithm, which also employs a truncated Chinese Restaurant Process to conduct the mixture meta-knowledge. Different from DPMM, it uses the Bayesian inference to conduct the mixture distribution of meta-knowledge and places an upper bound on the number of components to reduce the computational consumption.

All the baselines and our proposed ACML follow the experimental setting as described in Sec. F.3.

F.2. The datasets

VGG-Flowers: VGG-Flowers(Nilsback & Zisserman, 2008) consists of 102 flower categories. Also, we randomly choose 66 categories for meta-training, 16 categories for validation and the remained 20 categories for meta-test.

miniImagenet: miniImagenet(Ravi & Larochelle, 2017) is designed for few-shot learning, which consists of 100 different classes. Similarly, we also split the dataset into

Table 5. Some important hyper-parameters used in our experiments.

Hyper-parameter	**VGG-Flowers**	**miniImagenet**	**CIFAR-FS**	**Omniglot**
The number of outer update step	2000	2000	2000	2000
The outer learning rate	0.001	0.001	0.001	0.001
The number of outer update step	3	3	3	1
The inner learning rate	0.05	0.1	0.1	0.1

Table 6. Performance of our ACML and the baselines on each datasets at each meta-training stage. The best performance on each dataset is marked with boldface and the second best is marked with underline.

Meta-Training Stage	**Algorithms**	**VGG-Flowers**	**miniImagenet**	**CIFAR-FS**	**Omniglot**	**Average**
VGG-Flowers	FTML	76.84 ± 1.75	-	-	-	76.84 ± 1.75
	OSML	79.61 ± 1.50	-	-	-	79.61 ± 1.50
	DPMM	78.97 ± 1.52	-	-	-	78.97 ± 1.52
	BOMVI	77.05 ± 1.80	-	-	-	77.05 ± 1.80
	VC-BML	$\underline{83.71 \pm 1.58}$	-	-	-	$\underline{83.71 \pm 1.58}$
	ACML	$\mathbf{85.11 \pm 1.46}$	-	-	-	$\mathbf{85.11 \pm 1.46}$
miniImagenet	FTML	76.51 ± 1.92	44.97 ± 1.77	-	-	60.74 ± 1.85
	OSML	76.19 ± 1.68	56.11 ± 1.77	-	-	66.15 ± 1.73
	DPMM	$\underline{76.65 \pm 1.79}$	56.45 ± 1.74	-	-	66.55 ± 1.77
	BOMVI	75.75 ± 1.97	45.12 ± 1.74	-	-	60.44 ± 1.86
	VC-BML	76.47 ± 1.41	$\mathbf{59.71 \pm 1.75}$	-	-	$\underline{68.09 \pm 1.58}$
	ACML	$\mathbf{81.71 \pm 1.42}$	$\underline{57.19 \pm 1.66}$	-	-	$\mathbf{69.45 \pm 1.54}$
CIFAR-FS	FTML	75.11 ± 1.84	54.89 ± 1.66	70.13 ± 2.07	-	66.71 ± 1.86
	OSML	78.29 ± 1.63	57.36 ± 1.54	69.07 ± 2.01	-	68.24 ± 1.73
	DPMM	75.60 ± 1.76	55.79 ± 1.75	70.15 ± 2.07	-	67.18 ± 1.86
	BOMVI	74.08 ± 1.60	47.55 ± 1.84	57.07 ± 1.86	-	59.57 ± 1.77
	VC-BML	$\underline{79.04 \pm 1.54}$	$\mathbf{59.17 \pm 1.74}$	$\underline{71.40 \pm 1.93}$	-	$\underline{69.87 \pm 1.74}$
	ACML	$\mathbf{79.29 \pm 1.48}$	$\underline{58.98 \pm 1.65}$	$\mathbf{73.89 \pm 1.69}$	-	$\mathbf{70.72 \pm 1.61}$
Omniglot	FTML	63.04 ± 2.01	37.27 ± 1.69	47.95 ± 1.99	99.31 ± 0.28	61.89 ± 1.49
	OSML	70.68 ± 1.83	40.67 ± 1.50	51.89 ± 2.04	$\underline{99.35 \pm 0.24}$	65.65 ± 1.40
	DPMM	65.20 ± 1.67	$\underline{48.53 \pm 1.63}$	$\underline{60.15 \pm 2.30}$	99.16 ± 0.29	68.26 ± 1.47
	BOMVI	$\mathbf{73.19 \pm 1.86}$	46.28 ± 1.62	58.99 ± 2.14	97.71 ± 0.53	69.04 ± 1.54
	VC-BML	71.02 ± 1.76	48.53 ± 1.82	59.14 ± 2.01	99.21 ± 0.47	$\underline{69.48 \pm 1.52}$
	ACML	$\underline{71.92 \pm 1.86}$	$\mathbf{50.07 \pm 1.66}$	$\mathbf{64.50 \pm 1.83}$	$\mathbf{99.36 \pm 0.22}$	$\mathbf{71.46 \pm 1.39}$

three datasets (i.e., 64 classes for meta-training, 16 classes for validation and 20 classes for meta-test) following the existing works.

CIFAR-FS: CIFAR-FS(Bertinetto et al., 2018) dataset used in our experiment is adapted from the CIFAR-100 dataset (Krizhevsky et al., 2009) for few-shot learning, which consists of 100 classes. Following the existing works (Yap et al., 2021; Zhang et al., 2021), we also randomly split the datasets, where 64 classes are used for meta-training, 16 classes are used for validation and the remained 20 classes are used for meta-test, respectively.

Omniglot: Omniglot(Lake et al., 2011) is a widely-used dataset, which contains 1,623 different handwritten characters from 50 different alphabets. Following the previous works (Yap et al., 2021; Zhang et al., 2021), we randomly split the dataset into three subsets, 1,100 characters for meta-training, 100 characters for validation and the remaining 423 characters for meta-test.

To simulate the online non-stationary setting, we assume the above datasets are available sequentially. Moreover, we focus 5-way 5-shot task, which conducts the low-resource environment. For each dataset, we form the streaming tasks via randomly sampling 5 classes with replacement as a task. And we randomly sample 5 examples for each class in a support set and 15 examples for each class in a query set.

In our experiment, we also consider another more challeng-

Table 7. Performance of our ACML and the baselines on each datasets at each meta-training stage. The best performance (without 'original') on each dataset is marked with boldface and the second best (without 'original') is marked with underline.

Meta-Training Stage	Algorithms	Omniglot	CIFAR-FS	miniImagenet	VGG-Flowers	Average
Omniglot	FTML	99.25 ± 0.24	-	-	-	99.25 ± 0.24
	OSML	98.20 ± 0.39	-	-	-	98.20 ± 0.39
	DPMM	97.15 ± 0.48	-	-	-	97.15 ± 0.48
	BOMVI	97.35 ± 0.73	-	-	-	97.35 ± 0.73
	VC-BML	$\underline{99.28 \pm 0.48}$	-	-	-	$\underline{99.28 \pm 0.48}$
	ACML	$\mathbf{99.43 \pm 0.21}$	-	-	-	$\mathbf{99.43 \pm 0.21}$
	original	99.31 ± 0.25	-	-	-	99.31 ± 0.25
CIFAR-FS	FTML	96.12 ± 0.76	67.08 ± 1.87	-	-	81.60 ± 1.32
	OSML	96.09 ± 0.53	66.20 ± 2.02	-	-	81.15 ± 1.28
	DPMM	93.31 ± 0.80	60.88 ± 2.03	-	-	77.10 ± 1.42
	BOMVI	$\underline{97.68 \pm 0.43}$	56.29 ± 2.00	-	-	76.99 ± 1.22
	VC-BML	$\mathbf{97.72 \pm 0.38}$	$\underline{72.80 \pm 1.74}$	-	-	$\underline{85.26 \pm 1.06}$
	ACML	97.66 ± 0.39	$\mathbf{75.39 \pm 1.71}$	-	-	$\mathbf{86.53 \pm 1.05}$
	original	98.15 ± 0.39	73.82 ± 1.75	-	-	85.99 ± 1.07
miniImagenet	FTML	96.63 ± 0.58	68.60 ± 1.79	54.68 ± 1.9	-	73.30 ± 1.42
	OSML	95.04 ± 0.79	$\underline{69.20 \pm 1.72}$	55.13 ± 1.81	-	73.12 ± 1.44
	DPMM	95.01 ± 0.67	64.93 ± 2.14	55.49 ± 1.74	-	71.81 ± 1.52
	BOMVI	$\underline{97.01 \pm 0.70}$	59.25 ± 1.76	46.21 ± 1.66	-	67.49 ± 1.37
	VC-BML	96.29 ± 0.58	69.05 ± 1.68	$\underline{59.25 \pm 1.86}$	-	$\underline{74.86 \pm 1.37}$
	ACML	$\mathbf{97.10 \pm 0.46}$	$\mathbf{70.67 \pm 1.97}$	$\mathbf{59.97 \pm 1.70}$	-	$\mathbf{75.91 \pm 1.38}$
	original	96.74 ± 0.69	71.35 ± 1.72	60.13 ± 1.80	-	76.07 ± 1.40
VGG-Flowers	FTML	93.69 ± 0.83	58.27 ± 1.81	45.75 ± 1.52	80.32 ± 1.77	69.51 ± 1.48
	OSML	91.79 ± 1.14	$\underline{59.05 \pm 1.80}$	46.51 ± 1.64	81.71 ± 1.69	69.77 ± 1.57
	DPMM	93.21 ± 1.03	$\mathbf{61.55 \pm 1.82}$	45.01 ± 1.56	80.71 ± 1.72	70.12 ± 1.53
	BOMVI	$\mathbf{97.13 \pm 0.54}$	58.77 ± 1.89	$\underline{47.24 \pm 1.81}$	75.59 ± 2.04	69.68 ± 1.57
	VC-BML	92.80 ± 0.82	58.36 ± 1.87	47.09 ± 1.78	$\underline{82.92 \pm 1.46}$	$\underline{70.29 \pm 1.48}$
	ACML	$\underline{94.07 \pm 0.84}$	58.76 ± 1.64	$\mathbf{48.74 \pm 1.47}$	$\mathbf{83.31 \pm 1.50}$	$\mathbf{71.22 \pm 1.36}$
	original	93.25 ± 0.81	58.94 ± 1.82	49.29 ± 1.74	84.16 ± 1.56	71.41 ± 1.48

ing setting, where tasks from different datasets are mixed and arrive one by one. In this setting, we conduct different tasks stream with a length of 100, and then train the model on each task one by one and evaluate the performance on each dataset.

F.3. The details of experiment setting

For each task, we employ the same convolution network following the previous works (Yap et al., 2021; Zhang et al., 2021), which is shown in Tab. 4. For our model, we use the Adam optimizer as the outer optimizer and the SGD optimizer as the inner optimizer. For the Monte Carlo sampling used in our algorithm, we set the number of sampling as 5. For the initial number of components in the compositional meta-knowledge, we set it as 4. All the important hyperparameters can be seen in Tab. 5. We ran our algorithm on NVIDIA Tesla V100 32GB GPU. It took about 54 hours to train.

In practice, since the additional meta-knowledge components can easily lead to large computational consumption, we set the α of IBP as 1. It ensures that the expectation of the number of additional components at the beginning is 1 and decreases with time. As for the hyperparameter γ in our evidential sparsification method, it controls the sparsification rate. Instead of searching for an appropriate value, we can simply filter out the components with the lowest value in the final mass function in practice, which equals filtering the zero value with an appropriate γ.

F.4. Additional Experimental Result

In what follows, we present the full result on the streaming datasets (i.e., VGG-Flowers, miniImagenet, CIFAR-FS and Omniglot), and change the order of datasets to verify the generality of our algorithm.

F.4.1. Meta-test Accuracies on each Dataset at Different Meta-training Stage

We only show the average result at each meta-training stage and the performance on each dataset at the last meta-training stage in the main text. We additionally show the full results in Tab. 6. Although ACML can not achieve the best performance on all having arriving datasets at some meta-training stages (i.e., *CIFAR-FS* and *miniImage*), it outperforms all the baselines on the average results, which confirms the effectiveness of ACML. Additionally, ACML can not only maintain the performance on the old datasets, but also achieve better results on the new datasets, which illustrates that it can alleviate better catastrophic forgetting than other baselines. Note that ACML achieves the best performance on the current datasets at each stage (especially compared to VC-BML, where it assumes that each component is mutually exclusive), which shows that our proposed model can resolve the conflict between the learned meta-knowledge and the incremental meta-knowledge, and it is expected that the compositional can utilize the shared meta-knowledge to improve the performances.

Tab. 6 also shows the detailed results of ACML before and after evidential sparsification. The results show that ACML still achieves a comparative performance on most datasets at each meta-training stage, compared to before evidential sparsification. It further confirms the effectiveness of our proposed evidential sparsification.

F.4.2. Additional experimental in different order

To further confirm the generality of our model, we change the order of datasets in the streaming tasks. We conduct the experiments on a new order, where the model is trained chronologically on *Omniglot*, *CIFAR-FS* , *miniImagenet* and *VGG-Flowers*. The results are shown in Tab. 7. The result on the streaming tasks with a different order shows that ACML still outperforms other baselines, which further confirms the generality of ACML.