A RELATIONSHIP BETWEEN IBCL AND MAML

In this section, we discuss the relationship between IBCL and the Model-Agnostic Meta-Learning (MAML) and Bayesian MAML (BMAML) procedures introduced in Finn et al. (2017); Yoon et al. (2018b), respectively. These are inherently different than IBCL, since the latter is a continual learning procedure, while MAML and BMAML are meta-learning algorithms. Nevertheless, given the popularity of these procedures, we feel that relating IBCL to them would be useful to draw some insights on IBCL itself.

In MAML and BMAML, a task i is specified by a n_i-shot dataset D_i that consists of a small number of training examples, e.g. observations $(x_{1_i}, y_{1_i}), \dots, (x_{n_i}, y_{n_i})$. Tasks are sampled from a task distribution $\mathbb{T}$ such that the sampled tasks share the statistical regularity of the task distribution. In IBCL, Assumption 1 guarantees that the tasks P_i share the statistical regularity of class $\mathcal{F}$. MAML and BMAML leverage this regularity to improve the learning efficiency of subsequent tasks.

At each meta-iteration i,

1. *Task-Sampling*: For both MAML and BMAML, a mini-batch T_i of tasks is sampled from the task distribution $\mathbb{T}$. Each task $\tau_i \in T_i$ provides task-train and task-validation data, $D^{\text{trn}}_{\tau_i}$ and $D^{\text{val}}_{\tau_i}$, respectively.

2. *Inner-Update*: For MAML, the parameter of each task $\tau_i \in T_i$ is updated starting from the current generic initial parameter θ_0, and then performing n_i gradient descent steps on the task-train loss. For BMAML, the posterior $p(\theta_{\tau_i} \mid D^{\text{trn}}_{\tau_i}, \theta_0)$ is computed, for all $\tau_i \in T_i$.

3. *Outer-Update*: For MAML, the generic initial parameter θ_0 is updated by gradient descent. For BMAML, it is updated using the Chaser loss (Yoon et al., 2018b, Equation (7)).

Notice how in our work $\bar{w}$ is a probability vector. This implies that if we fix a number of task k and we let $\bar{w}$ be equal to $(w_1, \dots, w_k)^\top$, then $\bar{w} \cdot \bar{P}$ can be seen as a sample from $\mathbb{T}$ such that $\mathbb{T}(P_i) = w_i$, for all $i \in \{1, \dots, k\}$.

Here lies the main difference between IBCL and BMAML. In the latter the information provided by the tasks is used to obtain a refinement of the (parameter of the) distribution $\mathbb{T}$ on the tasks themselves. In IBCL, instead, we are interested in the optimal parametrization of the posterior distribution associated with $\bar{w} \cdot \bar{P}$. Notice also that at time $k+1$, in IBCL the support of $\mathbb{T}$ changes: it is $\{P_1, \dots, P_{k+1}\}$, while for MAML and BMAML it stays the same.

Also, MAML and BMAML can be seen as ensemble methods, since they use different values (MAML) or different distributions (BMAML) to perform the Outer-Update and come up with a single value (MAML) or a single distributions (BMAML). Instead, IBCL keeps distributions separate via FGCS, thus capturing the ambiguity faced by the designer during the analysis.

Furthermore, we want to point out how while for BMAML the tasks τ_i are all "candidates" for the true data generating process (dgp) P_i, in IBCL we approximate P_i with the product $\prod_{h=1}^{i} L_h$ of the likelihoods up to task i. The idea of different candidates for the true dgp is beneficial for IBCL as well: in the future, we plan to let go of Assumption 1 and let each P_i belong to a credal set $\mathcal{P}_i$. This would capture the epistemic uncertainty faced by the agent on the true dgp.

To summarize, IBCL is a continual learning technique whose aim is to find the correct parametrization of the posterior associated with $\bar{w} \cdot \bar{P}$. Here, $\bar{w}$ expresses the developer's preferences on the tasks. MAML and BMAML, instead, are meta-learning algorithms whose main concern is to refine the distribution $\mathbb{T}$ from which the tasks are sampled. While IBCL is able to capture the preferences of, and the ambiguity faced by, the designer, MAML and BMAML are unable to do so. On the contrary, these latter seem better suited to solve meta-learning problems. An interesting future research direction is to come up with imprecise BMAML, or IBMAML, where a credal set $\text{Conv}(\{\mathbb{T}_1, \dots, \mathbb{T}_k\})$ is used to capture the ambiguity faced by the developer in specifying the correct distribution on the possible tasks. The process of selecting one element from such credal set may lead to computational gains.

B REASON TO USE BAYESIAN CONTINUAL LEARNING

Let $q_0(\theta)$ be our prior pdf/pmf on parameter $\theta \in \Theta$ at time $t = 0$. At time $t = 1$, we collect data $(\bar{x}_1, \bar{y}_1)$ pertaining to task 1, we elicit likelihood pdf/pmf $l_1(\bar{x}_1, \bar{y}_1 \mid \theta)$, and we compute $q_1(\theta \mid \bar{x}_1, \bar{y}_1) \propto q_0(\theta) \times l_1(\bar{x}_1, \bar{y}_1 \mid \theta)$. At time $t = 2$, we collect data $(\bar{x}_2, \bar{y}_2)$ pertaining to task 2 and we elicit likelihood pdf/pmf $l_2(\bar{x}_2, \bar{y}_2 \mid \theta)$. Now we have two options.

(i) Bayesian Continual Learning (BCL): we let the prior pdf/pmf at time $t = 2$ be the posterior pdf/pmf at time $t = 1$. That is, our prior pdf/pmf is $q_1(\theta \mid \bar{x}_1, \bar{y}_1)$, and we compute $q_2(\theta \mid \bar{x}_1, \bar{y}_1, \bar{x}_2, \bar{y}_2) \propto q_1(\theta \mid \bar{x}_1, \bar{y}_1) \times l_2(\bar{x}_2, \bar{y}_2 \mid \theta) \propto q_0(\theta) \times l_1(\bar{x}_1, \bar{y}_1 \mid \theta) \times l_2(\bar{x}_2, \bar{y}_2 \mid \theta)$;[4]

(ii) Bayesian Isolated Learning (BIL): we let the prior pdf/pmf at time $t = 2$ be a generic prior pdf/pmf $q_0'(\theta)$. We compute $q_2'(\theta \mid \bar{x}_2, \bar{y}_2) \propto q_0'(\theta) \times l_2(\bar{x}_2, \bar{y}_2 \mid \theta)$. We can even re-use the original prior, so that $q_0' = q_0$.

As we can see, in option (i) we assume that the data generating process at time $t = 2$ takes into account both tasks, while in option (ii) we posit that it only takes into account task 2. Denote by $\sigma(X)$ the sigma-algebra generated by a generic random variable X. Let also Q_2 be the probability measure whose pdf/pmf is q_2, and Q_2' be the probability measure whose pdf/pmf is q_2'. Then, we have the following.

Proposition 3. *Posterior probability measure Q_2 can be written as a $\sigma(\bar{X}_1, \bar{Y}_1, \bar{X}_2, \bar{Y}_2)$-measurable random variable taking values in $[0, 1]$, while posterior probability measure Q_2' can be written as a $\sigma(\bar{X}_2, \bar{Y}_2)$-measurable random variable taking values in $[0, 1]$.*

Proof. Pick any $A \subset \Theta$. Then, $Q_2[A \mid \sigma(\bar{X}_1, \bar{Y}_1, \bar{X}_2, \bar{Y}_2)] = \mathbb{E}_{Q_2}[\mathbb{1}_A \mid \sigma(\bar{X}_1, \bar{Y}_1, \bar{X}_2, \bar{Y}_2)]$, a $\sigma(\bar{X}_1, \bar{Y}_1, \bar{X}_2, \bar{Y}_2)$-measurable random variable taking values in $[0, 1]$. Notice that $\mathbb{1}_A$ denotes the indicator function for set A. Similarly, $Q_2'[A \mid \sigma(\bar{X}_2, \bar{Y}_2)] = \mathbb{E}_{Q_2'}[\mathbb{1}_A \mid \sigma(\bar{X}_2, \bar{Y}_2)]$, a $\sigma(\bar{X}_2, \bar{Y}_2)$-measurable random variable taking values in $[0, 1]$. This is a well-known result in measure theory. □

Of course Proposition 3 holds for all $t \geq 2$. Recall that the sigma-algebra $\sigma(X)$ generated by a generic random variable X captures the idea of information encoded in observing X. An immediate corollary is the following.

Corollary 4. *Let $t \geq 2$. Then, if we opt for BIL, we lose all the information encoded in $\{(\bar{X}_i, \bar{Y}_i)\}_{i=1}^{t-1}$.*

In turn, if we opt for BIL, we obtain a posterior that is not measurable with respect to $\sigma(\{(\bar{X}_i, \bar{Y}_i)\}_{i=1}^{t}) \setminus \sigma(\bar{X}_t, \bar{Y}_t)$. If the true data generating process P_t is a function of the previous data generating processes $P_{t'}$, $t' \leq t$, this leaves us with a worse approximation of the "true" posterior $Q^{\text{true}} \propto Q_0 \times P_t$.

The phenomenon in Corollary 4 is commonly referred to as *catastrophic forgetting*. Continual learning literature is unanimous in labeling catastrophic forgetting as undesirable – see e.g. Farquhar and Gal (2019); Li et al. (2020). For this reason, in this work we adopt a BCL approach. In practice, we cannot compute the posterior pdf/pmf exactly, and we will resort to variational inference to approximate them – an approach often referred to as Variational Continual Learning (VCL) Nguyen et al. (2018). As we shall see in Appendix E, Assumption 1 is needed in VCL to avoid catastrophic forgetting.

B.1 RELATIONSHIP BETWEEN IBCL AND OTHER BCL TECHNIQUES

Like Farquhar and Gal (2019); Li et al. (2020), the weights in our Bayesian neural networks (BNNs) have Gaussian distribution with diagonal covariance matrix. Besides capturing the designer's ambiguity, $\mathcal{Q}^{\text{co}}(1, \ldots, k)$ is also useful because its convexity allows to remove the components of the knowledge base that are redundant, that is, that can be written as convex combination of the elements of $\text{ex}[\mathcal{Q}^{\text{co}}(1, \ldots, k)]$. Because IBCL is rooted in Bayesian continual learning, we can initialize IBCL with a much smaller number of parameters to solve a complex task as long as it can solve a set of

[4] Here we tacitly assume that the likelihoods are independent.

simpler tasks. In addition, IBCL does not need to evaluate the importance of parameters by measures such as computing the Fisher information, which are computationally expensive and intractable in large models.

C HIGHEST DENSITY REGION

Some scholars indicate HDRs as the Bayesian counterpart of the frequentist concept of confidence interval. In dimension 1, $R_\alpha(Q)$ can be interpreted as the narrowest interval – or union of intervals – in which the value of the (true) parameter falls with probability of at least $1-\alpha$ according to distribution Q. We give a simple visual example in Figure 3.

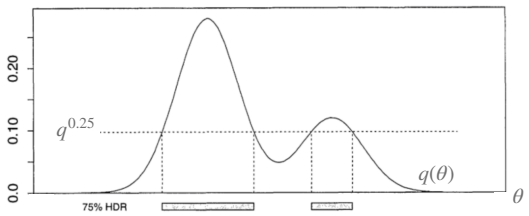

Figure 3: The 0.25-HDR from a Normal Mixture density. This picture is a replica of (Hyndman, 1996, Figure 1). The geometric representation of "75% probability according to Q" is the area between the pdf curve $q(\theta)$ and the horizontal bar corresponding to $q^{0.25}$. A higher probability coverage (according to Q) would correspond to a lower constant, so $q^\alpha < q^{0.25}$, for all $\alpha < 0.25$. In the limit, we recover 100% coverage at $q^0 = 0$.

D 2-WASSERSTEIN METRIC

In the main portion of the paper, we endowed $\Delta_{\mathcal{XY}}$ with the 2-Wasserstein metric. It is defined as

$$\|P - P'\|_{W_2} \equiv W_2(P, P') := \sqrt{\inf_{\gamma \in \Gamma(P,P')} \mathbb{E}_{((x_1,y_1),(x_2,y_2))\sim\gamma}[d((x_1,y_1),(x_2,y_2))^2]}, \quad \text{where}$$

1. $P, P' \in \Delta_{\mathcal{XY}}$;
2. $\Gamma(P, P')$ is the set of all couplings of P and P'. A coupling γ is a joint probability measure on $(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$ whose marginals are P and P' on the first and second factors, respectively;
3. d is the product metric endowed to $\mathcal{X} \times \mathcal{Y}$ (Deza and Deza, 2013, Section 4.2).[5]

E IMPORTANCE OF ASSUMPTION 1

We need Assumption 1 in light of the results in Kessler et al. (2023). There, the authors show that misspecified models can forget even when Bayesian inference is carried out exactly. By requiring that $\text{diam}(\mathcal{F}) = r$, we control the amount of misspecification via r. In Kessler et al. (2023), the authors design a new approach – called Prototypical Bayesian Continual Learning, or ProtoCL – that allows dropping Assumption 1 while retaining the Bayesian benefit of remembering previous tasks. Because the main goal of this paper is to come up with a procedure that allows the designer to express preferences over the tasks, we retain Assumption 1, and we work in the classical framework of Bayesian Continual Learning. In the future, we plan to generalize our results by operating with ProtoCL.[6]

[5] We denote by $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ the metrics endowed to $\mathcal{X}$ and $\mathcal{Y}$, respectively.

[6] In Kessler et al. (2023), the authors also show that if there is a task dataset imbalance, then the model can forget under certain assumptions. To avoid complications, in this work we tacitly assume that task datasets are balanced.

F AN EXAMPLE OF A PARAMETRIZED FAMILY $\mathcal{F}$

Let us give an example of a parametrized family $\mathcal{F}$. Suppose that we have one-dimensional data points and labels. At each task i, the marginal on $\mathcal{X}$ of P_i is a Normal $\mathcal{N}(\mu, 1)$, while the conditional distribution of label $y \in \mathcal{Y}$ given data point $x \in \mathcal{X}$ is a categorical Cat(ϑ). Hence, the parameter for P_i is $\theta = (\mu, \vartheta)$, and it belongs to $\Theta = \mathbb{R} \times \mathbb{R}^{|\mathcal{Y}|}$. In this example, family $\mathcal{F}$ can be thought of as the convex hull of distributions that can be decomposed as we just described, and whose distance according to the 2-Wasserstein metric does not exceed some $r > 0$.

G PREFERENCES INDUCE A PARTIAL ORDER ON THE TASKS

Notice how $\bar{w}$ induces a preference relation $\preceq_{\bar{w}}$ on the elements of $\mathcal{T}_k := \{P_1, P_2, \ldots, P_k\}$, $k \geq 2$. We have that $P_i \preceq_{\bar{w}} P_j$ if and only if $w_i \leq w_j$, $i \neq j$. In other words, we favor task j over task i if the weight w_j assigned to task j is larger than the one assigned to task i. In turn, $(\mathcal{T}_k, \preceq_{\bar{w}})$ is a poset, for all $k \geq 2$.

H PROOFS OF THE THEOREMS

Proof of Theorem 1. Without loss of generality, suppose we have encountered $i = 2$ tasks so far, so the FGCS is $\mathcal{Q}_2^{\text{co}}$. Assume (again without loss of generality) that all the elements in posterior sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ cannot be written as a convex combination of one another. Let $\hat{Q}$ be any element in the convex hull $\mathcal{Q}_2^{\text{co}}$. Then, there exists a probability vector $\bar{\beta} = (\beta_1^1, \ldots, \beta_1^{m_1}, \beta_2^1, \ldots, \beta_2^{m_2})^\top$ such that

$$\hat{Q} = \sum_{j=1}^{m_1} \beta_1^j Q_1^j + \sum_{j=1}^{m_2} \beta_2^j Q_2^j \propto L_1 \sum_{j=1}^{m_1} \beta_1^j Q_0^j + L_1 L_2 \sum_{j=1}^{m_2} \beta_2^j Q_0^j. \tag{4}$$

This proportional relationship is based on the Bayesian inference (line 4) in Algorithm 1. Hence, there exists an equivalent preference $\bar{w} = (w_1 = \sum_{j=1}^{m_1} \beta_1^j, w_2 = \sum_{j=1}^{m_2} \beta_2^j)^\top$. □

Proof of Theorem 2. For maximum generality, assume Θ is uncountable. Let $\hat{q}_{\bar{w}}$ denote the pdf of $\hat{Q}_{\bar{w}}$. The *α-level Highest Density Region* $R_\alpha(\hat{Q}_{\bar{w}})$ is defined in (Coolen, 1992) as a subset of the output space such that

$$\int_{R_\alpha(\hat{Q}_{\bar{w}})} \hat{q}_{\bar{w}}(\theta) \mathrm{d}\theta \geq 1 - \alpha \quad \text{and} \quad \int_{R_\alpha(\hat{Q}_{\bar{w}})} \mathrm{d}\theta \text{ is a minimum.}$$

We need $\int_{R_\alpha(\hat{Q}_{\bar{w}})} \mathrm{d}\theta$ to be a minimum because we want $R_\alpha(\hat{Q}_{\bar{w}})$ to be the smallest possible region that gives us the desired probabilistic coverage. Equivalently, from Definition 2 we know that we can write that $R_\alpha(\hat{Q}_{\bar{w}}) = \{\theta \in \Theta : \hat{q}_{\bar{w}}(\theta) \geq \hat{q}_{\bar{w}}^\alpha\}$, where $\hat{q}_{\bar{w}}^\alpha$ is a constant value. In particular, it is the largest constant such that $\Pr_{\hat{Q}_{\bar{w}}}[\theta \in R_\alpha(\hat{Q}_{\bar{w}})] \geq 1 - \alpha$ (Hyndman, 1996). Equation 3, then, comes from the fact that $\Pr_{\hat{Q}_{\bar{w}}}[\theta^\star_{\bar{w}} \in R_\alpha(\hat{Q}_{\bar{w}})] = \int_{R_\alpha(\hat{Q}_{\bar{w}})} \hat{q}_{\bar{w}}(\theta) \mathrm{d}\theta$, a well-known equality in probability theory (Billingsley, 1986). The integral is greater than or equal to $1 - \alpha$ by the definition of HDR. □

I DETAILS OF EXPERIMENT SETUP

Our experiment code is available at an anonymous GitHub repo: `https://github.com/ibcl-anon/ibcl`.

I.1 BENCHMARKS

We select 15 tasks from CelebA. All tasks are binary image classification on celebrity face images. Each task i is to classify whether the face has an attribute such as wearing eyeglasses or having a mustache. The first 15 attributes (out of 40) in the attribute list Liu et al. (2015) are selected for our

tasks. The training, validation and testing sets are already split upon download, with 162,770, 19,867 and 19,962 images, respectively. All images are annotated with binary labels of the 15 attributes in our tasks. We use the same training, validation and testing set for all tasks, with labels being the only difference.

We select 20 classes from CIFAR100 (Krizhevsky et al., 2009) to construct 10 Split-CIFAR100 tasks (Zenke et al., 2017). Each task is a binary image classification between an animal classes (label 0) and a non-animal class (label 1). The classes are (in order of tasks):

1. Label 0: aquarium fish, beaver, dolphin, flatfish, otter, ray, seal, shark, trout, whale.
2. Label 1: bicycle, bus, lawn mower, motorcycle, pickup truck, rocket, streetcar, tank, tractor, train.

That is, the first task is to classify between aquarium fish images and bicycle images, and so on. We want to show that the continual learning model incrementally gains knowledge of how to identify animals from non-animals throughout the task sequence. For each class, CIFAR100 has 500 training data points and 100 testing data points. We hold out 100 training data points for validation. Therefore, at each task we have 400 * 2 = 800 training data, 100 * 2 = 200 validation data and 100 * 2 = 200 testing data.

We also select 20 classes from TinyImageNet (Le and Yang, 2015). The setup is similar to Split-CIFAR100, with label 0 being animals and 1 being non-animals.

1. Label 0: goldfish, European fire salamander, bullfrog, tailed frog, American alligator, boa constrictor, goose, koala, king penguin, albatross.
2. Label 1: cliff, espresso, potpie, pizza, meatloaf, banana, orange, water tower, via duct, tractor.

The dataset already splits 500, 50 and 50 images for training, validation and testing per class. Therefore, each task has 1000, 100 and 100 images for training, validation and testing, respectively.

20NewsGroups (Lang, 1995) contains news report texts on 20 topics. We select 10 topics for 5 binary text classification tasks. Each task is to distinguish whether the topic is computer-related (label 0) or not computer-related (label 1), as follows.

1. Label 0: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x.
2. Label 1: misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey.

Each class has different number of news reports. On average, a class has 565 reports for training and 376 for testing. We then hold out 100 reports from the 565 for validation. Therefore, each binary classification task has 930, 200 and 752 data points for training, validation and testing, on average respectively.

I.2 Training Configurations

All data points are first preprocessed by a feature extractor. For images, the feature extractor is a pre-trained ResNet18 (He et al., 2016). We input the images into the ResNet18 model and obtain its last hidden layer's activations, which has a dimension of 512. For texts, the extractor is TF-IDF (Aizawa, 2003) succeeded with PCA to reduce the dimension to 512 as well.

Each Bayesian network model is trained with evidence lower bound (ELBO) loss, with a fixed feed-forward architecture (input=512, hidden=64, output=1). The hidden layer is ReLU-activated and the output layer is sigmoid-activated. Therefore, our parameter space Θ is the set of all values that can be taken by this network's weights and biases.

The three variational inference priors, learning rate, batch size and number of epcohs are tuned on validation sets. The tuning results are as follows.

1. CelebA: priors = $\{\mathcal{N}(0, 0.2^2 I), \mathcal{N}(0, 0.25^2 I), \mathcal{N}(0, 0.3^2 I)\}$, lr = $1e-3$, batch size = 64, epochs = 10.
2. Split-CIFAR100: priors = $\{\mathcal{N}(0, 2^2 I), \mathcal{N}(0, 2.5^2 I), \mathcal{N}(0, 3^2 I)\}$, lr = $5e-4$, batch size = 32, epochs = 50.

3. TinyImageNet: priors = $\{\mathcal{N}(0, 2^2I), \mathcal{N}(0, 2.5^2I), \mathcal{N}(0, 3^2I)\}$, lr = $5e-4$, batch size = 32, epochs = 30.
4. 20NewsGroup: priors = $\{\mathcal{N}(0, 2^2I), \mathcal{N}(0, 2.5^2I), \mathcal{N}(0, 3^2I)\}$, lr = $5e-4$, batch size = 32, epochs = 100.

For the baseline methods, we use exactly the same learning rate, batch sizes and epochs. For probabilistic baseline methods (VCL and VCL-reg), we use the prior with the median standard deviation. For example, on CelebA tasks, VCL and VCL-reg uses the normal prior $\mathcal{N}(0, 0.25^2I)$.

I.3 Evaluation Method

We use widely adopted continual learning metrics, (1) average per-task accuracy and (2) peak per-task accuracy to evaluate performance, as well as (3) backward transfer (Díaz-Rodríguez et al., 2018) to evaluate resistance to catastrophic forgetting. These metrics are computed from all accuracies acc_{ij} of a model at the end of task i on the testing data on a previous task $j \in \{1, \dots i\}$. Specifically,

$$\begin{aligned} avg\_per\_task\_acc_i &= \frac{1}{i}\sum_{l=1}^{i} acc_{ij},\ i \in \{1, \dots, N\} \\ peak\_per\_task\_acc_i &= \max_{j \in \{1,\dots,i\}} acc_{ij},\ i \in \{1, \dots, N\} \\ avg\_per\_task\_bt_i &= \frac{1}{i-1}\sum_{l=2}^{i}(acc_{ij} - acc_{i(j-1)}),\ i \in \{2, \dots, N\} \end{aligned} \tag{5}$$

To obtain an acc_{ij} that evaluates preference-addressing capability, at each task i, we randomly sample $K = 10$ preferences, $\bar{w}_{i1}, \dots, \bar{w}_{iK}$, over all tasks encountered so far. Therefore, GEM-reg, VCL-reg and IBCL need to generate K models, one for each preference. All K models are evaluated on testing data of task $j \in \{1, \dots i\}$, resulting in accuracy acc_{ijk}, with $k \in \{1, \dots, K\}$. We use preference as weights to compute acc_{ij} as a weighted sum

$$acc_{ij} = \sum_{k=1}^{K} \frac{\bar{w}_{ik}[j]}{W_{ik}} acc_{ijk} \tag{6}$$

where $W_{ik} = \sum_{k=1}^{K} \bar{w}_{ik}[j]$ is the normalization factor to ensure the resulting accuracy value is in $[0, 1]$. Here, $\bar{w}_{ik}[j]$ denotes the j-th scalar entry of preference vector $\bar{w}_{ik}$. For GEM and VCL, we only learn 1 model per task to address all preferences. To evaluate that one model's capability in preference addressing, we use its testing accuracy in place of acc_{ijk} in equation 6. By this computation, all accuracy scores are preference-weighted and reflect an algorithm's ability to produce preference-addressing models.

Recall that models generated by VCL, VCL-reg and IBCL are probabilistic (BNNs for VCL and VCL-reg and HDRs for IBCL). Therefore, we sample 100 deterministic models from each of the output probabilistic models to compute acc_{ijk}. We record the maximum, mean and minimum values of acc_{ijk} across all the sampled models. The maximum value is the estimated Pareto optimality.

J Ablation Studies

We conduct two ablation studies. The first one is on different significance level α in Algorithm 2.

In Figure 4, we evaluate testing accuracy on three different α's over five different preferences (from $[0.1, 0.9]$ to $[0.9, 0.1]$) on the first two tasks of 20NewsGroup. For each preference, we uniformly sample 200 deterministic models from the HDR. We use the sampled model with the maximum L2 sum of the two accuracies to estimate the Pareto optimality under a preference. We can see that, as α approaches 0, we tend to sample closer to the Pareto front. This is because, with a smaller α, HDRs becomes wider and we have a higher probability to sample Pareto-optimal models according to Theorem 2. For instance, when $\alpha = 0.01$, we have a probability of at least 0.99 that the Pareto-optimal solution is contained in the HDR.

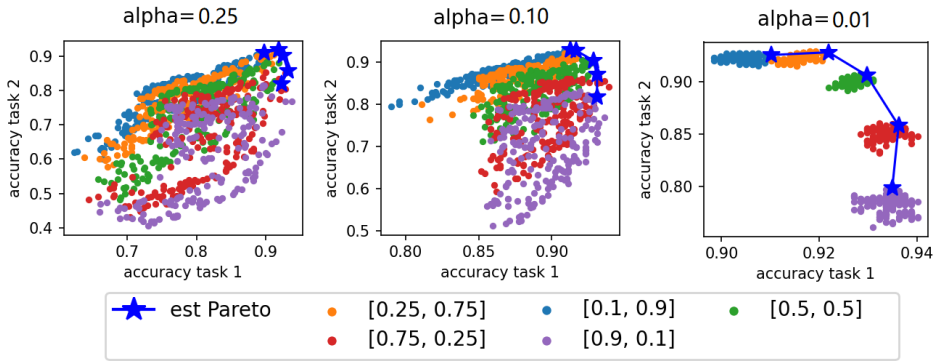

Figure 4: Different α's on different preferences over the first two tasks in 20NewsGroup.

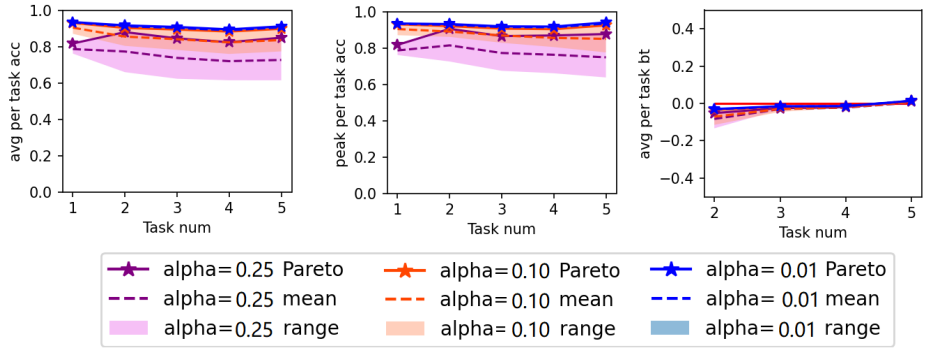

Figure 5: Different α's on randomly generated preferences over all tasks in 20NewsGroup.

We then evaluate the three α's in the same way as in the main experiments, with 10 randomly generated preferences per task. Figure 5 shows that the performance drops as α increases, because we are more likely to sample poorly performing models from the HDR.

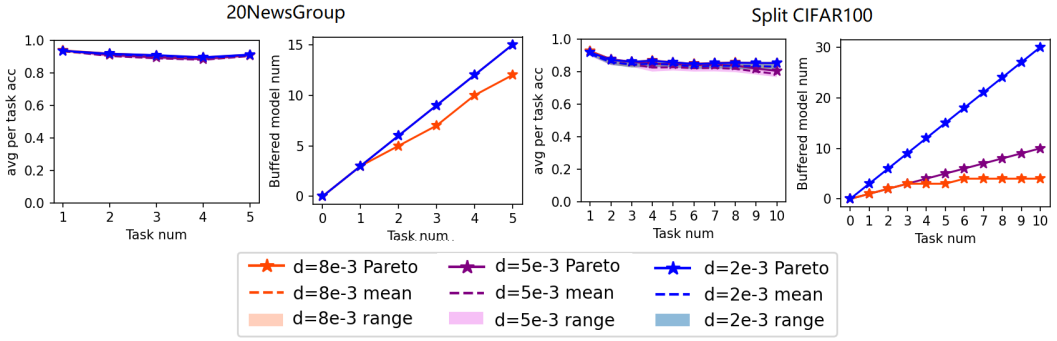

Figure 6: Different d's on 20NewsGroup and Split-CIFAR100. The buffer growth curves of $d = 5e-3$ and $d = 2e-3$ of 20NewsGroup are overlapping.

The second ablation study is on different thresholds d in Algorithm 1. As d increases, we are allowing more posteriors in the knowledge base to be reused. This will lead to memory efficiency at the cost of a performance drop. Figure 6 supports this trend. We can see how performance barely drops by reusing posteriors, while the buffer growth speed becomes sublinear. For Split-CIFAR100, when $d = 8e-3$, the buffer size stops growing after task 6.