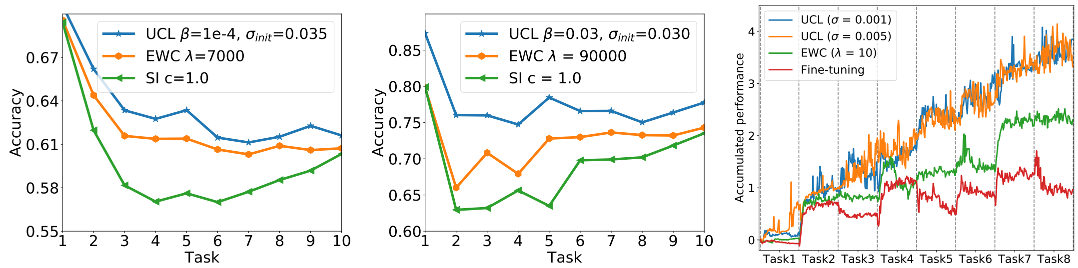

(a) Split CIFAR-100 (b) Omniglot (c) Accum. rewards for 8 RL tasks

Figure 1: Additional experiments on supervised learning and reinforcement learning tasks

Supervised learning [R1,R3] To check the effectiveness of UCL beyond the MNIST tasks, we experimented our UCL on two additional datasets, Split CIFAR-100 and Omniglot. For Split CIFAR-100, each task consists of 10 consecutive classes of CIFAR-100, and for Omniglot, each alphabet is treated as a single task, and we randomly sampled 10 tasks from all 50 alphabets. For Omniglot, we rescaled all images to 28×28 and augmented the dataset by including 20 random permutations (rotations and shifting) for each image. For both datasets, unlike the experiments in the manuscript, we used deeper CNN architectures, for which the notion of *uncertainty* in the convolution layer is defined for each *channel* (i.e., filter). For Split CIFAR-100, we used 6 3×3 convolution layers with 32-32-64-64-128-128 channels and 2 dense layers with 2048 and 256 nodes, and for Omniglot, we used 4 3×3 convolutional layers with 64 channels and 1 dense layer with 1024 nodes. We used multi-head outputs for both experiments, and 5 and 3 different random seed runs are averaged for Split CIFAR-100 and Omniglot, respectively. In Figure 1(a) and 1(b), we compared with EWC and SI and carried out extensive hyperparameter search for fair comparison. We did not compare with VCL since it did not have any results on vision datasets with CNN architecture. From the figures, we clearly observe that UCL outperforms the baselines for both tasks as well, stressing the effectiveness of UCL on diverse datasets. We could not carry out experiments on CUB and miniImageNet due to time constraint, and we will defer to the future work.

Reinforcement learning [R1-R3] We believe one of the important contributions of UCL is its strong performance in reinforcement learning setting. To that end, we conducted additional experiments on the Roboschool platform that expands the results in Section 4.2 of the manuscript. Namely, we randomly selected 8 tasks, {*Walker-HumanoidFlagrun-Hooper-Ant-InvertedDoublePendulum-Cheetah-Humanoid-InvertedPendulum*} and carried out continual learning (i.e., the past task data is not available once a new task is learned). We used two fully connected layers with 16 nodes and other hyperparameters were equal to the described in the manuscript. The hyperparamters were set to $\beta = 0.001$ and $\sigma_{\text{init}} = \{0.001, 0.005\}$ to show the influence of σ_{init}. Figure 1(c) shows the cumulated sum of normalized rewards up to the learned task, where the normalization was done for each task with the reward obtained by the task-dedicated network. Thus, the high cumulative sum corresponds to effectively combating the catastrophic forgetting (CF), and fine-tuning, which is known to suffer from CF, hovering around 1 makes sense. We observe UCL significantly outperforms EWC and different σ values have little effect on the final reward (Re:[**R3**]). We believe the reason why EWC does not excel as in Figure 4B of the original EWC paper, [Kirkpatrick *et.al*, Overcoming catastrophic forgetting in neural networks, *PNAS* 2017], is because we consider pure continual learning setting, while the original EWC paper allows learning tasks multiple times in recurring fashion. A possible reason why UCL works so well in RL setting may be due to the by-product of our weight sampling procedure; namely, it enables effective exploration as in [20, manuscript]. We stress that there are few algorithms in the literature that work well on both SL and RL continual learning setting, and our UCL is very competitive in that sense. We will include the reward trajectories for learning each task (that resulted in Fig 1(c)) in the Supplementary Material of the final version.

[R2] ① We apologize for any confusions we made while describing the sampling procedure in [Line 206, manuscript]. What we meant was that we sample model parameters every iteration, and the number of sampling is 1 for each iteration. At the beginning epoch of task t, we sample from $q(\mathcal{W}|\boldsymbol{\theta}_t)$ with $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1}$ (i.e., using the learned paramter up to task $t-1$), then continue to update $\boldsymbol{\theta}_t$ in the subsequent iterations. Hopefully, this resolves the confusion of the reviewer. ② We will make sure to correct some redundant "so-called" expressions in the final version.

[R3] We disagree that our work is only an incremental improvement over VCL for the following reasons. ① As **[R1]** has pointed out, our novel interpretation of KL term gives new insights and variations on online Bayesian learning. ② Since UCL dramatically reduces the number of parameters compared to VCL, we can apply UCL to much larger and deeper models as shown in above experiments with CNNs. Note VCL does not have any results on using deep CNNs. ③ Since UCL samples the weight parameters only once for each iteration, applying it to actor-critic based reinforcement learning algorithm becomes possible. We believe concrete regularization terms that we derive enables such efficient sampling scheme. In contrast, VCL needs to sample weights multiple times in each iteration for the Monte-Carlo simulation, and it is almost impossible to apply VCL in the RL continual learning setting as above.

While we do not have rigorous theoretical analyses on the formulation of UCL, the combination of ℓ_1 and ℓ_2 norms in the regularization term reminds of the *Elastic-net*, widely used in statistical learning. Also, our node-wise notion of uncertainty gives natural extension to the CNN models by defining uncertainty for each filter (channel) and leads to good performance for deep CNN models.