We thank all reviewers for the insightful feedback. We are encouraged to note that our method, MERLIN, is novel (**R2**,**R3**,**R5**); our experimental setting captures the method's effectiveness (**R3**,**R5**); we achieve 'outstanding performance compared to strong baselines'(**R2**); our approach is 'well-grounded' (**R1**), 'clearly different from prior works' (**R2**). **R2** notes that the task-free scenario used is a 'realistic direction to the area'. Further, all reviewers unanimously agree that 'the paper is well-written'. We address all comments below. We will use the additional page allowed in the final version to incorporate feedback and address any lack of clarity in presentation (**R2**).

Methods	Split MNIST	Permuted MNIST	Split CIFAR10	Split CIFAR100	Mini-ImageNet
Single	44.89 ± 0.30	73.13 ± 2.27	73.24 ± 3.08	30.81 ± 3.57	27.57 ± 2.64
EWC	45.01 ± 0.14	74.98 ± 2.04	74.28 ± 2.2	29.23 ± 3.38	28 ± 2.59
GEM	86.79 ± 1.56	82.05 ± 4.95	79.13 ± 1.68	40.65 ± 1.95	34.17 ± 1.23
iCaRL	89.91 ± 0.92	NA	72.65 ± 1.33	27.13 ± 2.99	38.86 ± 1.63
GSS	88.39 ± 0.81	81.44 ± 1.27	57.9 ± 2.65	19.19 ± 0.7	14.81 ± 0.98
MERLIN	90.67 ± 0.80	$\mathbf{85.54 \pm 0.5}$	$\mathbf{82.93 \pm 1.16}$	$\mathbf{43.55 \pm 0.61}$	$\mathbf{40.05 \pm 2.94}$
CN-DPM (ICLR'20)	$\mathbf{92.12 \pm 0.14}$	-	46.01 ± 1.23	14.29 ± 0.14	-
MERLIN - SN Prior	23.34 ± 0.24	32.51 ± 1.57	28.23 ± 2.21	12.32 ± 1.45	14.76 ± 0.23

(**R1**) **"Comparison with Bayesian Continual Leaning (BCL) methods":** Thanks for this question, it gives more completeness to our work. We compared MERLIN against the most recent BCL work, CN-DPM (Lee et al., ICLR'20) as **R1** suggested. As shown in the table, CN-DPM performs better than MERLIN on Split-MNIST, but drastically fails on harder datasets. We note that the baseline methods considered in this work also perform better than CN-DPM on non-MNIST datasets. We'd also like to add that not all VAE-based CL methods learn a posterior over model params, or operate in an online CL setting. For eg., VCL (Nguyen et al, ICLR'18) learns a posterior over the data distribution (also not an online CL method), and not model param distribution. This is a subtle difference to be noted. MERLIN performs variational CL in the model param space, can be adapted easily to class + domain incremental setting, and work in task-aware + task-agnostic settings.

(**R1**) **"Why task-specific learned priors, not std normal prior?":** As correctly noted in **R1**'s 'Summary', the learned task-specific priors are necessary to generate task-specific weights and consolidate meta-model on previous task params, as well as to sample models for ensembling at inference. As suggested, we ran a study where we replaced the task-specific learned prior with a standard Normal prior and finetuned the corresponding generated model on task-specific exemplars. The last row of table above (MERLIN - SN Prior) shows the result (very poor), validating the usefulness of task-specific learned priors. We also visualized the learned-task specific prior in Fig 3 (Appendix), where we see good separability across tasks.

Methods	HAT	AudioMNIST
Single	47.79 ± 0.94	76.37 ± 2.25
EWC	47.19 ± 2.58	79.89 ± 16.14
GEM	67.23 ± 0.97	89.45 ± 1.14
iCaRL	62.34 ± 2.45	84.73 ± 2.22
GSS	69.79 ± 1.51	92.81 ± 0.19
MERLIN	$\mathbf{73.54 \pm 1.71}$	$\mathbf{96.47 \pm 1.79}$

(**R2**, **R5**)**"Expts on heterogeneous datasets from HAT?; Results on more domains?":** Thanks for the suggestion. We ran expts on the Heterogeneous dataset from HAT as well as the AudioMNIST dataset (Becker et al, 2018) (shown in adjoining table). We comfortably outperform baselines on these datasets and domains too.

Methods	Split CIFAR10	Split CIFAR100	Mini-ImageNet
Single	69.65 ± 0.79	18.8 ± 2.21	18.57 ± 2.31
EWC	67.98 ± 2.96	16.89 ± 3.95	19.29 ± 3.58
GEM	72.23 ± 1.56	26.71 ± 1.75	27.71 ± 2.56
iCaRL	69.23 ± 2.24	24.81 ± 2.88	23.84 ± 1.95
GSS	49.82 ± 2.01	13.99 ± 0.56	12.92 ± 0.17
MERLIN	$\mathbf{82.93 \pm 1.16}$	$\mathbf{43.55 \pm 0.61}$	$\mathbf{40.05 \pm 2.94}$

(**R1**) **"Use of bigger architectures in baselines may hurt their performance":** We re-ran all baselines with the same smaller ResNet used in MERLIN (L248), and report the results in adjoining table. We see that MERLIN outperform all baselines here again; the performance of the baseline model drops significantly, possibly due to the smaller capacity.

(**R1**) **"For CIFAR100/miniImageNet, 10 classes/task corresponds to 5000 samples/class and not 2500? Cited works used 5 classes/task":** For these datasets, we randomly sampled 2500 samples from 5000, and used the same 2500 across all baselines, for fair comparison (results reported across 5 such trials). To further clarify, we ran experiments with 5 samples per task (20 tasks) and report the results in adjoining table. We note that baseline accuracy matches with values reported in GEM (Tab 2, Col 3). We perform better than baselines even in this setting.

Methods	Split CIFAR100	Mini-ImageNet
Single	36.44 ± 3.44	35.85 ± 2.08
EWC	37.03 ± 2.51	35.36 ± 2.07
GEM	57.02 ± 1.41	52.28 ± 1.53
iCaRL	50.23 ± 1.37	53.22 ± 1.56
GSS	18.74 ± 0.82	16.34 ± 0.12
MERLIN	$\mathbf{64.83 \pm 1.78}$	$\mathbf{57.35 \pm 1.92}$

Other clarifications: - (**R3**) *"Scale of subsets":* Sec 4.1.1 has these details. The base model is trained with 1000 (MNIST) and 2500 (all datasets other than MNIST) samples per task; - (**R1**,**R3**) *"Amount of episodic memory used for baselines":* All baselines had access to same amount of exemplar memory as in MERLIN: 100 for MNIST and 600 for all other datasets. In Appendix Sec C, we study effect of varying exemplar memory size of MERLIN and two of its best competitors (GEM and iCaRL); - (**R1**) *"Inference time":* MERLIN take 745 ms while baseline methods take $\sim$ 300ms for CIFAR datasets on a single 1080Ti GPU. It takes more time than baselines, but is still real-time; - (**R5**) *"Why forgetting measure is worse for MERLIN on two datasets.":* No method is perfect. As discussed in L308-312, iCaRL uses distillation loss to ensure that logits of previous task don't alter much while learning a new task. This brings down the forgetting measure. We still outdo all baselines on 3 other datasets; - (**R1**) *"Baselines in task-aware setting":* All reported results of baselines in the paper are task-aware. Task-agnostic MERLIN was put under disadvantage while being compared to task-aware counterparts, still outperforming them. We believe that this confusion arose because of L 331, which should have been "All results *of MERLIN* in Tab 1 do not assume task information"; - (**R3**) *"vote of basic model...trick to improve performance":* Our method design allows ensembling of models for CL, though each model may be weak by itself (i.e. number of models is 1) - each individual model is upto $8\times$ smaller in param size than baseline models (L 353). Such an ensembling approach has not been tried before, and cannot be done easily with existing methods too.