

1 We thank you for the valuable reviews and the agreements on the novelty of our
2 method. We will fix the clarity issues and grammar problems in the final version.

3 **Reviewer 1** Thanks for the positive review, which encouraged us a lot.

4 **Confuse about the pretrain stage.** We have to clarify that the pretrain stage is to
5 train the meta-graph instead of the controller in offline learning with the initial task.
6 Then we jointly train both controller and meta-graph in an online learning manner.
7 We will make it clear in the final version.

8 **Computational complexity of update ... in online learning.** The computational
9 complexity of our method is high due to the NAS-based model. However, our
10 method "improve the accuracy" consistently in the scenario when the models are not able to store or see each mini-batch
11 of samples after doing several times of backward propagations, i.e. instance-level forgetting problem. Improving
12 computational complexity is a practical direction to study in the future.

13 **Why not compare to [1] and the performance difference.** We compare to A-GEM instead of [1] since A-GEM shows
14 better performance and is architecturally similar to [1]. The number of classes per task in our work is also used by [14,
15 17], but different from A-GEM. Hence, we cannot compare with the number in A-GEM directly.

16 **Storage of controller could be large.** We clarify that our controller only requires 80K number of parameters, which is
17 highly efficient comparing to storing the raw data.

18 **Potential for adapting to Single-head.** Most methods use the replay mechanism to achieve single-head. Note that
19 labeled examples are used for dealing with the imbalance logits at the last layer. Hsu et al.¹ shows that it is possible to
20 achieve single-head even with naive rehearsal, so it is possible to apply replay tricks as it is orthogonal to our idea.

21 **Reviewer 2** Thanks for agreeing that our idea is novel and providing more related works including brand new papers
22 accepted by ECCV'20 and CVPR-W'20. We will cite those in our final version.

23 **Value and difficulty of multi-head.** We agree that single-headed is interesting and more challenging. However, online
24 continual learning with multi-headed is still an unsolved problem, where our method achieves consistent improvements
25 over several methods on different datasets. Moreover, our method can be adapted to single-head by applying the replay
26 mechanism since replay is orthogonal to our proposed idea.

27 **Confuse with experiment setup and why it is better than replay buffer.** We store the learned parameters (80K) of the
28 controller for each task. This is more efficient than replay buffer which increases proportionally to the size of the dataset.
29 Directly storing data in the buffer also violates our setting of not able to see past examples, which might be restricted
30 when security is a concern.

31 **Comparison of number of parameters.** We have compared the additional parameters in line 241 of our main paper.
32 Our method only required 80K number of additional parameters per task compare to the standard replay-based method,
33 which required 3M for CIFAR, 12M for Tiny-ImageNet. Besides, the size of our model (meta-graph) can be shrunk from
34 43M to 15M by adopting Single-Path NAS² (see the performance in the table above), while other architectural-based
35 methods [14] required 80M, [13] required 66~74M.

36 **Reviewer 3** Thanks for appreciating the idea, we do several experiments and arguments to make it more convincing.
37 **Number of parameters.** Please refer to the table above and the response to R2, we have shrunk our model size from
38 43M to 15M by adopting Single-Path NAS. Another architectural-based method [14] has more parameters compare to
39 us. The number of parameters for autoencoders and expert networks in [13] will increase with the number of tasks, so it
40 is not as efficient as ours.

41 **Compare to [13,14].** We report the performance of [14] in the table above. Worse performance is expected as explained
42 in line 96-97 of the main paper. Note that we do not show the performance of [13] as we failed to achieve reasonable
43 performance with the authors' MatLab code in our setting. However, the worse result is also as expected since the
44 architectural-based methods often fail to handle online learning settings without proper design and tuning.

45 **Design choices of count-based exploration.** The design of $H^{\frac{1}{2}}$ is inspired by count-based exploration, sigmoid is to
46 mimic classifying overused blocks. ϵ is set to 0.5 since the input was ranged in [0,1]. Other hyperparameters analysis
47 will be included in the final paper.

48 **Not entirely Online Learning and confused about "epochs".** For the concern of not entirely online learning, please
49 refer to the answer to R1's 1st and 2nd questions. For pretraining, we train the meta-graph on the whole task with
50 multiple epochs. For jointly training, we will back prop the model with received mini-batch several times and not seeing
51 it anymore. We would like to clarify that the term "epoch" refers to iterating through a mini-batch instead of the whole
52 task, we will modify it as "iteration" in the final version.

53 **Reviewer 4** Thanks for appreciating our work, we compare with RPSNet [14] in the table above.

Method	Param.	Accuracy (%)	
		CIFAR10	CIFAR100
Ours	43M	83.8	54.4
Ours	15M	81.8	55.5
[14]	80M	67.4	40.1

¹Hsu, Yen-Chang, et al. "Re-evaluating continual learning scenarios: A categorization and case for strong baselines." arXiv preprint arXiv:1810.12488 (2018).

²Stamoulis, Dimitrios, et al. "Single-path nas: Designing hardware-efficient convnets in less than 4 hours." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2019.