Thanks for all the valuable comments. Please check our responses below. We will address all minor comments.

R1 Q1: how the loss on the current task and on the memory change during training compared to other methods.

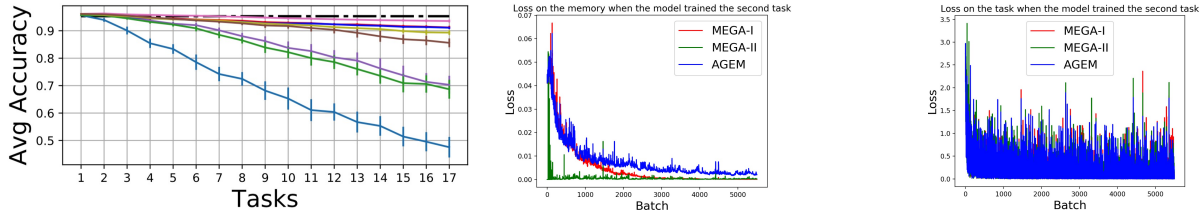
A1: In Figure 1b and 1c, the losses on the memory and on the task when the model is trained on the second task are plotted. On the task, all the methods share similar noisy pattern. On the memory, MEGA-I and MEGA-II achieve smaller error compared with A-GEM which reveals why MEGA-I and MEGA-II overcome forgetting.

R1 Q2: Is it the number of training examples that is limited or is it also the number of gradient updates?

A2: We agree. When the number of training examples is limited, if we fix the batch size, the consequence is that the number of gradient updates is smaller. The training loss converges slower and the ratio spends more time above 1, as the reviewer pointed out. This is also consistent with our derivation and empirical results in Appendix A.7. We will make it more clear in the revised version.

R1 Q3: Error bars.

A: The results are averaged across 5 runs with different random seeds. In Figure 1a, for an example we add error bars for the plot on MNIST and we will add all the missing error bars in the revised draft.



(a) (b) (c)

R1 Q4: The random variables xi and zeta are confusing. Typos/grammar. Include table of hyperparameters.

A:Thanks for the suggestions. We will fix them in the revised version.

R2 Q1: For a fairer picture of the topic, other more recent methods should also be included, e.g., [30] or [36].

A: Thanks for the comments. The paper [30] focuses on task-free continual learning which is a different setting from ours. In [36], the authors focus on sample selection for episodic memory based lifelong methods which is an orthogonal topic. In this paper, we use the same sample selection method (uniform sampling) for all episodic memory based lifelong learning methods for a fair comparison which is also the strategy employed in A-GEM [2].

R2 Q2: The multi-task baseline.

A: In the multi-task baseline, all the tasks are learned jointly (i.e., the examples of all the tasks are shuffled and the model is optimized over a single pass over the examples with SGD). The proposed methods only perform better than the multi-task baseline on the *Split CUB*. This can be possibly attributed to the joint effects of episodic memory and better optimization algorithms. First, by storing examples in the episodic memory, the examples of old tasks can be accessed *multiple times* instead of just *one time* as in the multi-task baseline. On the other hand, with the proposed better balancing schemes, the proposed methods outperform GEM and A-GEM, and also surpass the multi-task baseline on the *Split CUB*.

R2 Q3: Missing an important baseline—the performance when all the tasks are learned jointly.

A: Thanks for pointing this. The missing baseline is the multi-task baseline used in the paper. We will clarify this.

R2 Q4: Figure 4 also shows a curious behavior.

A: In Figure 4, we can observe MEGA-I significantly outperform A-GEM (71% vs. 63% with 600 examples per task).

R2 Q5: Additional feedback.

A: We mentioned the derivation of MEGA-II through α_1 and α_2, at line 202-203. Please refer to Appendix A.3 for details. And for the visualization of MEGA-II, we will add one figure to better illustrate the intuition.

R3 Q1: Especially surprising is the performance on CUB where this beats the multitask baseline.

A: Please refer to R2 Q2.

R3 Q2: Here are a few memory papers missing from the discussion.

A: We will add the missing citations and discussions.

R4 Q1: Theoretical Grounding and Empirical Evaluation.

A: The proposed view is used to point out one limitation of GEM and A-GEM which are the state-of-the-art lifelong learning methods. The proposed methods are motivated by the fact that GEM and A-GEM always choose $\alpha_1 = 1$. The results show the benefits of adjusting α_1 during training. It is shown in [2] that A-GEM has better or comparable performance than GEM, so we focus on comparing with A-GEM. We manage to finish the experiments of GEM, PROG-NN and MER on *Split AWA* dataset (20 tasks), and the results are (GEM: 44.7±2.47, PROG-NN: 41.34±4.03, MER: 36.34±3.94), all are inferior to MEGA-I (54.82±4.97). We will include all the results in the revised version.

R4 Q2: Why "MEGA-I with $\alpha_1 = 1$ and $\alpha_2 = 1$ " in table 1 do better than AGEM in 2. Typo/format.

A: As shown in Figure. 6 in A-GEM [2], the angle between the task gradient and the reference gradient is mostly acute (3500 / 5500), thus in A-GEM, $\alpha_2 = 0$ (as shown in Equation (4)) occurs frequently. In this case, the update totally ignores the reference memory (leads to forgetting). In contrast, MEGA-I with $\alpha_1 = 1$ and $\alpha_2 = 1$ considers both the reference memory and the current task in *each* update step which could better alleviate forgetting (although not as good as MEGA-I which dynamically adjusts α_1 and α_2). We will address the format issue in the revised draft.