We thank the reviewers for their feedback. R2 and R4 recommend accepting our paper based on a novel, mathematically well motivated approach and convincing results. R2 says, *" .. a fundamentally different way of diminishing catastrophic forgetting.. proposed solution is backed by solid mathematical framework .. "*; R4 says, *"..an approach which somehow bridges the two problems and proposes a dual view on these problems is great.. experiments are quite convincing.."*R1 raised concerns regarding comparison to prior work and specific contributions of our work, which we address first.

R1: "Re-implement CIFAR-100 experiments .. compare with EWC, GEM"
A: We initially followed the experimental protocol of Zenke et. al. 2017 for the CIFAR experiments. We re-implemented the variant in Lopez-Paz 2017. Our method achieves 65.2% average accuracy across 5 runs for ResNet18, significantly outperforming EWC (49.8% accuracy), iCARL (Rebuffi et al., 54.6%) and comparable to GEM result of 67.8%. Note that the results of prior work were computed in the GEM paper using a non-standard ResNet architecture and required a separate output layer for each task. In contrast, we use the standard ResNet18 and only require a single output layer.

R1: " .. several important earlier works have not been discussed .."
A: We acknowledge the brevity of our discussion of past work in our paper and promise to significantly expand this discussion in the next draft. The key differences/benefits of our work are:

- We require significantly less number of additional parameters to add a new task. For instance, is a fully connected layer has MxN parameters, we only require M additional parameters (binaryPSP) for each new task. In contrast, EWC stores the diagonal Fisher Matrix of size MxN parameters for each task.
- Lopez-Paz et al. necessarily need to store data from *all* the previous tasks. Our method has no such requirements.
- Masse et al.'s idea of gating is conceptually different from our method. Masse et al., gate by zeroing out a subset of features (i.e. a bitwise multiplication with a vector such as 1, 0, 1, 0), whereas we use the rotation of features to nearly orthogonal vectors. E.g. in binary superposition, a context vector might be: -1, 1, 1, -1. However, we can choose the context to be ternary (e.g. -1, 0, 1, 0), where a few elements are set to 0. The gating method of Masse et al. is therfore a special case of our framework. We will include our results showing that ternary superposition is superior to gating by simply zeroing out features. Additionally, we have provided theoretically motivated guidelines for choosing these basis/context vectors and the effect of these choices on the amount of catastrophic forgetting.
- Methods like EWC, GEM constraint the amount of change in parameters for a new task in a manner that ensures performance on past tasks is maintained. With increase in tasks, constraints accumulate, making it impossible to add new tasks once optimization slows. Our method, however, shows a different property – its much more flexible in the sense that it forgets the older tasks when network capacity is reached in order to accommodate newer tasks.

R1: " ..context variables are fixed and random; not trainable .. one can train the context variables.."
A: Great question and suggestion. We will clarify this further in the text. We chose random context variables because randomness is enough to guarantee (near) orthogonality of vectors in high-dimensional vector spaces. However, our framework is fully differentiable and it is possible to learn the context variables as you suggested. In fact, we have been working in this direction and the initial results are very promising.

R1: ".. automatic inference of task identity .."
A: This question is of great interest and we are working on it, but is complementary to our main contribution of demonstrating that parameter superposition is a powerful tool to combat catastrophic forgetting.

R1: "If possible, implement the Atari benchmarks and compare the results with EWC."
A: Unfortunately, we could not train on ATARI due to time constraints – but will include in the final version.

R1, R2: A few typos and inconsistencies in present in the results.
A: Sorry for the confusion. We promise to revise the text to improve clarity and remove typos and any inconsistencies.

R2: " ..how many models can be stored in superposition is a very interesting question .."
A: Yes, indeed. On permuting MNIST, we tested networks with 2000, 1000, 500, 250 units and each storing 1600, 800, 400, 200, 100 different tasks. Performance degrades gracefully with number of tasks: 2000 unit network has an average accuracy of (94.06%, 100 tasks); (92.35%, 200 tasks); (85.21%, 400 tasks); (65.15%, 800 tasks). For a 250 unit network: (83.68%, 100 tasks); (72.46%, 200 tasks); (53.46%, 400 tasks); (35.65%, 800 tasks). We will include these results in the final version of the paper.

R3: "..real tasks with available data for which it would make sense to use such a model..authors to comment.."
A: Continual learning methods are very useful in resource constrained online -learning settings, where it is not possible to store past data to form batches or in situations where data statistics are continuously changing. An example is mini-drone performing remote surveillance as lighting changes from day to night (analogous to rotating MNIST).

R3: "..how this work could influence or guide research in model compression?"
A: Model compression at training time is an unsolved problem. We present one way to use excess capacity in networks that solve multiple tasks during training time – we hope this instigates more interest in online compression of models.