Appendix

The supplementary materials include:

- **Appendix A**: Additional experimental results for MNIST and CIFAR10.
- **Appendix B**: Proof of arguments in GEM discussion.
- **Appendix C**: Comprehensive reproducibility details.

A. Additional results

Due to space constraints in the main paper, in this section we report the additional results.

Learning trajectory MNIST. Figure 1 illustrates the learning trajectory projection in parameter space for MNIST. The CIFAR10 and Mini-Imagenet results are reported in the main paper, for which the findings extend to this MNIST sequence as well.

Loss of linear interpolations MNIST and CIFAR10. Figure 2 reports the loss for linear interpolations in parameter space for MNIST and CIFAR10. Results for Mini-Imagenet are reported in the main paper. Notably, CIFAR10 w_1 to w_2 does not overfit significantly on the rehearsal memory. However, training is only done for one epoch per task and after training three more tasks with the rehearsal memory, w_5 reports near zero loss for the rehearsal memory, indicating overfitting. This shows that rehearsal may overfit more on the rehearsal memory as the training sequence length increases, either by more tasks (e.g. CIFAR10) or more epochs per task (e.g. Mini-Imagenet with 10 epochs per task).

MNIST high-loss ridge aversion. We provide additional experiments for other commonly used rehearsal memory sizes (0.2k and 2k) in the MNIST setup [1, 3, 8].

Experiment	**Rehearsal memory size $\|\mathcal{M}\|$**	
	200	*2000*
MNIST		
ER	81.8 ± 0.7	91.8 ± 0.4
ER-step	87.6 ± 1.1 (n=20)	92.6 ± 0.3 (n=5)

Table 1: Additional MNIST avg. accuracy results for memory sizes 200 and 2000, comparing ER and ER-step for the high-loss ridge aversion experiment with n the number of steps.

B. Loss constraints: GEM vs. Rehearsal

In GEM [4], the updates of the model are restricted to the directions where the loss on the memory samples decreases or remains equal. This is imposed by requiring $g_n \cdot g_i \geq 0, \forall i$ with g_n the gradient on the new batch and g_i on sample i in the rehearsal memory. This is a first-order approximation, hence it is only exact where the loss surface is linear.

In contrast, in rehearsal an increase of the loss on the memory samples is allowed, as long as it is smaller or equal than the decrease in loss on the new batch. We proof this in the following.

A model update with SGD is calculated as:

$$w' \leftarrow w - \alpha g \tag{1}$$

with gradient g and learning rate α. If we assume the first order approximation to hold in an α-region around w, then because the negative gradient is either zero or points in a direction with decreasing loss, it follows that

$$\begin{aligned} \mathcal{L}(w') &\leq \mathcal{L}(w) \,, \\ \mathcal{L}_m(w') + \mathcal{L}_n(w') &\leq \mathcal{L}_m(w) + \mathcal{L}_n(w) \,, \\ \mathcal{L}_m(w') - \mathcal{L}_m(w) &\leq \mathcal{L}_n(w) - \mathcal{L}_n(w') \,, \end{aligned} \tag{2}$$

with $\mathcal{L}$ the average of the loss on the memory $\mathcal{L}_m$ and the loss of the new batch $\mathcal{L}_n$. Therefore, based on the same first order approximation as in [4], Eq. 2 shows that rehearsal only allows increases in loss on the memory as large as the decrease in loss on the new batch.

C. Reproducibility details

This section provides all the details to maintain reproducibility of our experiments. Furthermore, our codebase provides the original implementation in Pytorch to reproduce our results[1].

C.1. Empirical evidence Hypotheses 1 and 2

MNIST is trained with a two-layer MLP, with each layer 400 ReLU nodes. Optimization of the model uses vanilla stochastic gradient descent (SGD), with a constant learning rate of 0.01. Each update is performed on a batch of 10 new and 10 memory samples. The rehearsal memory has a fixed capacity of 50 samples per task. This fixed capacity is allocated before training to enable analyzing overfitting for static task-specific rehearsal memory's. Online training is performed as each sample is only seen once during training, except for the memory samples. The MNIST split results in $T1$ containing 0's and 1's and $T2$ containing 2's and 3's.

CIFAR10 training details are equal to those of MNIST, except for the model and the memory capacity. The model used is the reduced Resnet18, introduced by by Lopez-Paz et al. [4]. $T1$ of CIFAR10 consists of planes and cars and $T2$ contains birds and cats, following the standard split. The memory capacity is 100 samples per task.

Mini-Imagenet training details are equal to those of CIFAR10, except for training 10 epochs per task rather than

[1] Code released upon paper acceptance: github.com/*******

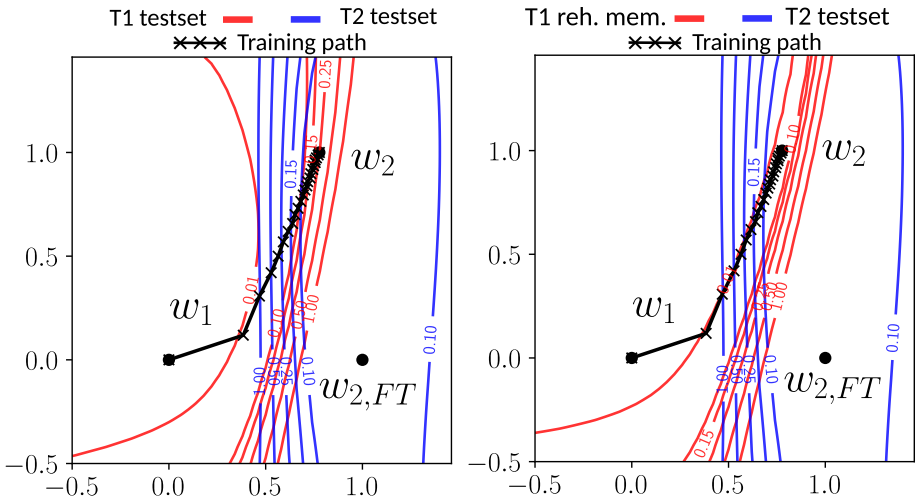

Figure 1: Projection of MNIST learning trajectories in parameter space on the plane defined by w_1, w_2 and $w_{2,FT}$. For the same $T2$ test loss (blue), the loss for $T1$ (red) is calculated in two different ways. *left*: $T1$ loss for the vast test set. *right*: $T1$ loss for the limited rehearsal memory.

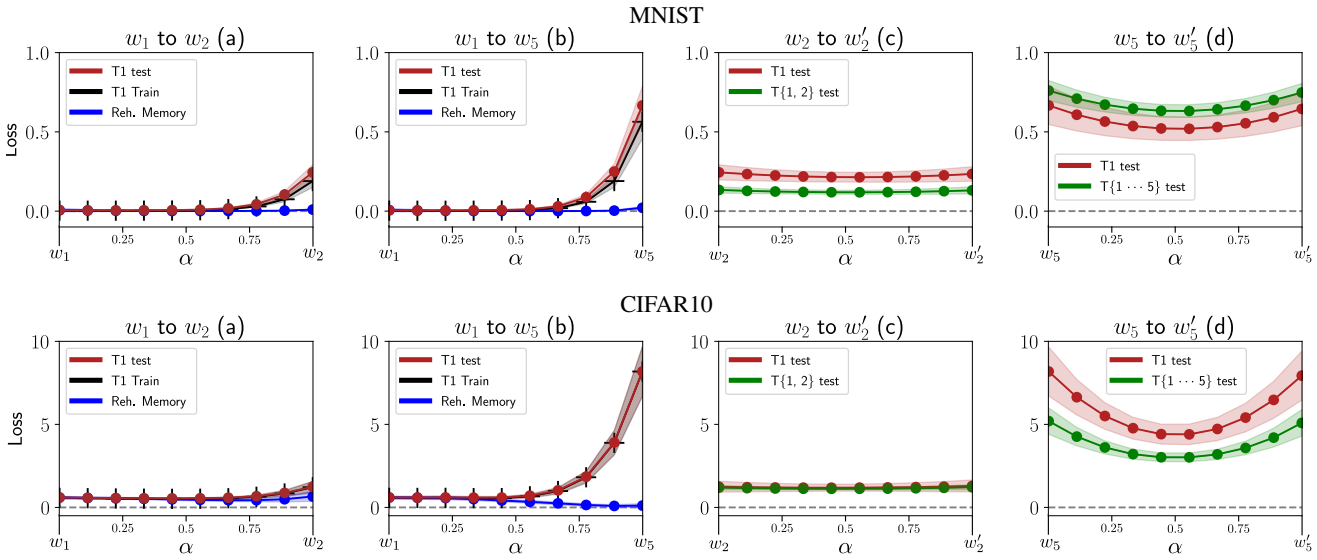

Figure 2: Avg. loss and standard deviation on linear paths between the models used in the empirical evidence of hypotheses one and two. Training is performed on MNIST (*top*) and CIFAR10 (*bottom*) and sampled 100 times for different model initializations and memory populations. A path from w_i to w_j is calculated as $(1-\alpha)w_i + \alpha w_j$. *(a) and (b)*: Loss on the linear path between the model after learning $T1$ (w_1) and the model after learning with rehearsal on $T2$ and $T5$ respectively. *(c) and (d)*: Loss of the path between two models learned with different memory populations, after $T2$ and $T5$ respectively. Red is the loss on the T1 testset, green the average loss of all tasks up to T2 and T5 respectively. Results contained no outliers with a higher loss on the path compared to the loss of the models.

training online. For Mini-Imagenet there is no standard split and the categories were assigned randomly to a task, but remained the same in all experiments. As in CIFAR10, the memory capacity is 100 samples per task.

C.2. High-loss ridge aversion

This section details the learning details for the high-loss ridge aversion experiments. All benchmarks use a gridsearch for the number of steps $n \in \{0, 1, 2, 3, 4, 5, 10, 20, 50\}$, with $n = 0$ the Experience Re-

play (ER) baseline. We report the best results from this gridsearch for *ER-step* with $n > 0$, following the procedure in [4]. All results are obtained with 10 epochs per task. In contrast to the hypothesis experiments discussed in Appendix C.1, this experiment allows for a dynamically subdivided rehearsal memory instead of a fixed allocation over all tasks. We use this memory policy in this experiment as it is commonly used in literature and allows exploiting the full memory capacity [7, 1, 2].

MNIST has the same setup as in Appendix C.1, except with 10 epochs per task and learning rate 0.001 to allow smaller steps when training only on the rehearsal memory.

CIFAR10 and Mini-Imagenet follow the reduced Resnet18 setup with Stable-SGD [6] for CIFAR100 in [5]. That is, Stable-SGD is used to obtain wider minima, with momentum 0.8 and initial learning rate 0.1, where we used a decaying factor per task t of 0.8^t. The fixed dropout rate 0.1 is obtained from gridsearch in values $[0.1, 0.25]$.

C.3. Projection plots

The loss contour plots in the parameter space as in Figure 1 are inspired by recent work [5]. They show a hyperplane in the parameter space, defined by three points w_1, w_2 and w_3. Orthogonalizing $w_2 - w_1$ and $w_3 - w_1$ gives a two dimensional coordinate system with base vectors u and v. The value at point (x, y) is then calculated as the loss of a model with parameters $w_1 + u \cdot x + v \cdot y$. For more details, we refer to our code or the appendix in [5]. The training trajectories shown in these figures are from a single run and are the projections of the points in the parameter space to this hyperplane. The projection of w' is calculated as $u \cdot (w' - w_1)$ and $v \cdot (w' - w_1)$. The indicated points are each 50, 50 and 100 steps apart for respectively MNIST, CIFAR10 and Mini-Imagenet.

References

[1] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. *arXiv preprint arXiv:2009.00919*, 2020. 1, 3

[2] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 3

[3] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 1

[4] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pages 6467–6476, 2017. 1, 3

[5] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. 3

[6] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *arXiv preprint arXiv:2006.06958*, 2020. 3

[7] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 3

[8] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1