# The Expanding Scope of the Stability Gap: Unveiling its Presence in Joint Incremental Learning of Homogeneous Tasks

Sandesh Kamath[1,2] Albin Soutif-Cormerais [1,2] Joost van de Weijer [1,2] Bogdan Raducanu [1,2]
[1]Department of Computer Science, Universitat Autònoma de Barcelona
[2]Computer Vision Center, Barcelona {skamath, albin, joost, bogdan}@cvc.uab.es

## Abstract

*Recent research identified a temporary performance drop on previously learned tasks when transitioning to a new one. This drop is called the stability gap and has great consequences for continual learning: it complicates the direct employment of continually learning since the worse-case performance at task-boundaries is dramatic, it limits its potential as an energy-efficient training paradigm, and finally, the stability drop could result in a reduced final performance of the algorithm. In this paper, we show that the stability gap also occurs when applying joint incremental training of homogeneous tasks. In this scenario, the learner continues training on the same data distribution and has access to all data from previous tasks. In addition, we show that in this scenario, there exists a low-loss linear path to the next minima, but that SGD optimization does not choose this path. We perform further analysis including a finer batch-wise analysis which could provide insights towards potential solution directions.*

## 1. Introduction

Deep neural networks demonstrate remarkable performance across numerous machine-learning tasks. Nevertheless, when trained on non-IID streaming data these networks struggle to accumulate knowledge, and tend to forget previously acquired knowledge. Continual learning develops theory and methods to address this problem [3, 12]. It aims to develop algorithms that prevent *catastrophic forgetting* [13] and achieve a more favorable trade-off between stability and plasticity [14] while learning on a data stream.

A typical test setting that continual learning considers is learning from a sequence of tasks (each task with another data distribution) [3]. Usually, continual learning method performance is evaluated at the end of each of the tasks. Recently, researchers [1, 10] have observed an interesting phenomenon that went unnoticed in this standard evaluation setup: at the start of training a new task, the performance of previous tasks drastically drops, and only slowly recovers during the subsequent training of the new task. De Lange et al. [10] coined the term *stability gap* for this phenomenon. This observation should be taken into account for the application of continual learning systems (especially in safety-critical contexts) since it significantly lowers the worst-case performance of these algorithms. Furthermore, it can potentially worsen the final accuracy of the learner, since it might not recover totally from the knowledge loss incurred during the stability gap. Addressing the stability gap is therefore of utmost importance [6, 18].

The underlying mechanism responsible for the stability gap remains the subject of lively scientific debate, with no clear explanation available yet. Originally, Caccia et al. [1] hypothesized that the cause for the stability gap is because old class prototypes receive a large gradient from closely lying new class prototypes. However, this hypothesis could not fully explain the phenomenon, because the stability gap had also been observed in domain incremental learning (where the set of classes remains the same) [10]. A possible remaining explanation is the following. When optimizing on new data, the objective is to minimize the loss on both the available new data and unavailable old data. The loss on the unavailable previous data is then approximated with various continual learning strategies, such as regularization [9, 11] and data rehearsal [2, 16]. An explanation for the stability gap could be the failure to approximate this ideal joint loss on previous and current task data. Surprisingly, a recent paper [8], showed that even in the case of joint incremental training the stability gap occurred (in this case, we do have access to both old and new task data and can minimize the joint loss on both old and new data). They, therefore, came to the important realization that we should not only focus on *what* to optimize but more importantly on *how* to optimize our objective.

Hess et al. [8] made their important observation when learning on heterogeneous tasks, referring to the fact that each task is drawn from a different distribution. In this paper, we show that the stability gap is even present in the case of joint incremental learning on homogeneous tasks (where

| Paper | Type | Tasks | |
|---|---|---|---|
| Caccia et al. [1] | CI: $c_1 \neq c_2$ | disjoint | heterogeneous |
| Lange et al. [10] | DI: $c_1 = c_2$ | disjoint | heterogeneous |
| Hess et al. [8] | CI: $c_1 \neq c_2$ | joint incr. | heterogeneous |
| Ours | DI: $c_1 = c_2$ | joint incr. | homogeneous |

Table 1. Summary of the expanding scope of the stability gap: from heterogeneous to homogeneous tasks.

each task is drawn from the same distribution). This result in presented in Figure 1. So, even in the case that both tasks have the same distribution, SGD optimization does not succeed in going to the 'nearby' optimal position without derailing through a high-loss region. The only difference between the new and old data is that the network has seen the old data (typically for 100 epochs here) and has not yet seen the newly arriving data. We think that this further confirms the fundamental nature of the stability gap in continual learning: it even occurs in the most simple continual learning setting when training from an increasing amount of data drawn from the same distribution. The main contributions of this work are:

1. We show that the stability gap also occurs during joint incremental learning from homogeneous tasks, arguably the least challenging continual learning setting.
2. We show that there exists a linear low-loss path to the optimal loss, but that SGD is not following this path (this was hypothesized in [8] but was not demonstrated).
3. We perform an analysis at mini-batch level, and discover that the gradient just after the task boundary successfully decreases the mini-batch loss but results in an overall loss increase on the test set. Addressing this might potentially lead to a solution to the stability gap problem.

This manuscript does not provide a new possible explanation for the stability gap. We think the observation that it occurs even for joint incremental learning of homogeneous tasks is relevant. Our results, confirm those of Hess et al. [8] and we agree with them that the focus should shift to how to optimize rather than what to optimize.

# 2. Stability Gap Analysis

## 2.1. Experimental setup

**Datasets:** We use the standard benchmark train-test split for all the datasets used in this work, that is publicly available. CIFAR-10 dataset consists of $60,000$ images of $32 \times 32$ size, divided into 10 classes: $50,000$ used for training and $10,000$ for testing. CIFAR-100 dataset consists of $60,000$ images of $32 \times 32$ size, divided into 100 classes: $50,000$ used for training and $10,000$ for testing.

**Architectures:** We consider two convolutional network architectures, VGG-16 [17] and ResNet-18 [7] for our study.

**Training Setup:** Our code base uses the pytorch library. For training we use the SGD optimizer with hyperparameters: learning rate (lr) of 0.01, momentum (m) of 0.9, batch size (bs) of 64.

**Notation:** In this work, we mainly study the two-task setting. All results reported will be in the *homogeneous task setting*, where the various tasks are drawn from the same distribution. We use the notation of $A$-$B$ to indicate task A will contain A% of the data and task B will contain B% of the data from the original training dataset. We will use the notation $A$-$B^*$ to identify the *joint incremental learning* setting. In this case when training task B, the algorithm has access to all the data of task A. In practice, for this setting for task B, we just combine the data of both tasks, and continue training on the combined dataset. Note, that the data of task A and B in our paper are disjoint data sets and do not contain the same data samples.

**Note on plots:** Most plots in this paper are with a warm-started model. This means a model trained on task A with the data as prescribed in the setting was used to continue training on task B. The starting point of the x-axis is then the iterations directly after the task-switch. This was done to better study the effect of the actual stability gap. Note, that we do not show the end of training on task B.

## 2.2. Stability Gap in Joint Incremental Learning of Homogeneous Tasks

To establish the occurrence of the stability gap in the setting of joint incremental learning of homogeneous tasks, we study the 50-50* setting. This setting divides the training set into two equally sized tasks, A and B. Both tasks are drawn from the same distribution. The test accuracy is provided for two datasets in Figure 1. We can observe that even in for this case, there is a clear stability gap. The performance drops from 0.89 to 0.74 on CIFAR-10 and from 0.65 to 0.38 on CIFAR-100. We posit a larger gap on CIFAR-100 to be related to the smaller number of samples per class. Note that for both these graphs performance has not returned to its task A level consistently even after the 2000 iterations showing the long-lasting impact of the stability gap. After continued training for around 3500-4500 iterations the models start to achieve more consistently a performance above 0.89 and 0.65, respectively.

In Table 1 we provide a summary of the main papers on the stability gap. The stability gap has been observed in increasingly general settings. Here, we show that it is also observed for joint incremental training of homogeneous tasks, which is arguably the most simple continual learning setting. This observation is relevant since it discards explanations for the stability gap which are based on characteristics that are not present (e.g. it cannot be uniquely explained by the presence of disjoint tasks or heterogeneous distributions).

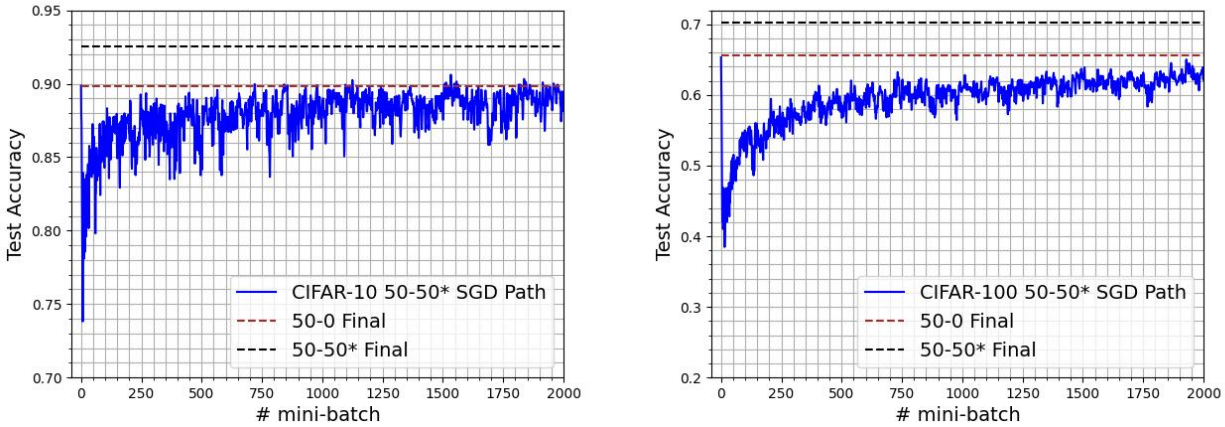


Figure 1. Occurrence of the stability gap in joint incremental learning with homogeneous tasks in the 50-50* setting on (left) CIFAR-10 and (right) CIFAR-100 datasets on a ResNet-18 model. This plot starts after training with task A, and the x-axis represents the number of iterations of training on task B.

### 2.3. Linear Mode Connectivity

Garipov et al. [5] were the first to study the mode connectivity properties of neural networks weights by connecting two independent minima obtained through differently seeded optimization processes using a simple curved path of low loss. Frankle et al. [4] later showed that a simpler kind of path naturally emerges early in training. They observed that models that are trained from a warm-started model version on the same dataset but with different SGD-noise lead to two checkpoints that are connected by a linear path of low loss. Mirzadeh et al. [15] later extended that property to optima of multitask models trained on incrementally larger datasets. Hess et al. [8] hypothesized that there exists a low-loss path between the optima when doing joint incremental learning of heterogeneous tasks. However, they do not demonstrate this in their paper. In this article, we investigate whether it is the case that training on incremental homogeneous tasks leads to linearly connected optimas or not (and we verify this). To do so, we take the initial checkpoint with weights $\theta_1$ and final checkpoint with weights $\theta_2$ and interpolate between the two by taking $\theta_\lambda = \lambda\theta_1 + (1-\lambda\theta_2)$ with $\lambda \in [0, 1]$. We later compute and report the test accuracy of each $\theta_\lambda$ to determine if the linear path is of low loss.

Figure 2 compares the loss of the models obtained by linearly interpolating between the initial and final model to the ones of the model checkpoints along the SGD optimization trajectory. Unsurprisingly, the path taken by SGD during optimization is not linear. More surprisingly, it goes through areas of higher loss especially during the initial period that corresponds to the *stability gap*, while the linear path between the initial and final model is of low loss. The linear path results confirm that a low-loss path exists between the minima achieved after training task A, and the minima after training task B. Surprisingly, SGD does not take this path and instead passes through a high-loss area before converging towards the minima which is optimal for task A and B data. We have shown here the first few iterations after the task-switch.

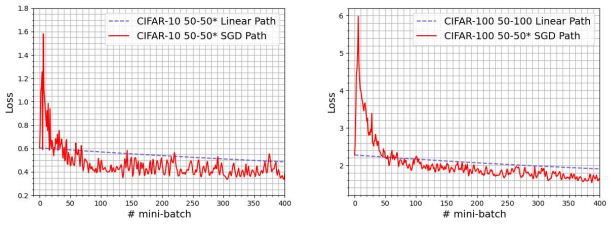


Figure 2. In the 50-50* setting, we present the loss path with SGD and the linear connectivity loss path between the warm-start and final models using with ResNet-18 model on (left) CIFAR-10, (right) CIFAR-100 dataset. In order to observe the stability gap, we zoom in on the first few iterations of the new task.

**Per mini-batch loss analysis.** In Figure 3, we observe with a microscope the learning of the model per mini-batch. In this plot we show the training batch accuracy for the current mini-batch before (blue line) and after (red line) the SGD update. We observe that the SGD update results in a loss decrease(or accuracy increase) for the particular mini-batch (the blue line is below the red line). However, when we look at the test accuracy (black line), we see that even though initial steps lead to a lower loss on the mini-batch, they do not result in better test performance. The black line goes down in the initial iterations. This means that the SGD update moves the network parameters away from the optimal path.

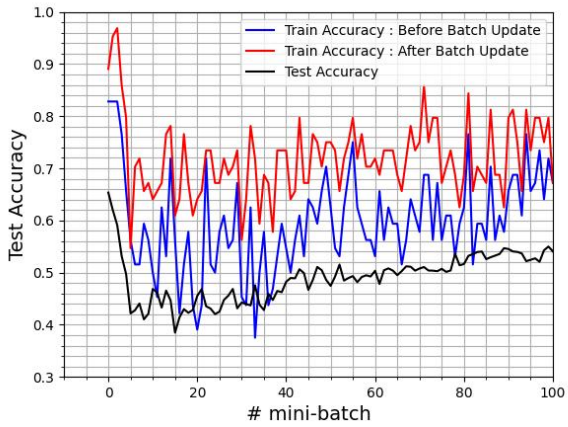


Figure 3. Using CIFAR-100 with ResNet-18, we present the finer analysis of the **local** improvement obtained at the batch level by observing the train accuracy per batch before (blue line) and after (red line) SGD update is applied for the batch in the 50-50* setting. The black line is the corresponding test accuracy.

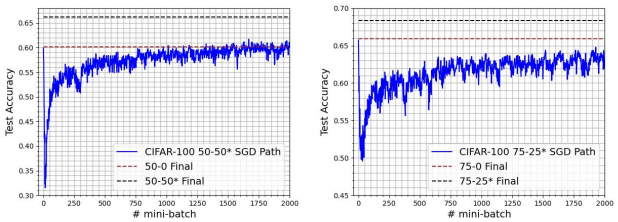


Figure 4. Using CIFAR-100 with VGG-16, stability gap in (left) 50-50* (right) 75-25* setting.

## 2.4. Additional Analysis

Here we verify if the stability gap also occurs for several other settings.

**Stability gap using other architectures.** While we present a detailed study of the stability gap on ResNet-18 architecture, in Figure 4 we show this phenomenon is not restricted to a specific architecture by using another well-known VGG-16 architecture on the CIFAR-100 dataset.

**Stability gap in other settings.** In Section 2.2, we mainly considered the 50-50* setting which is the joint incremental training with homogeneous task. Here, we look at the stability gap with different first task size and include results for the setting 10-90* and 75-25* in Figure 5. We observe that the gap is larger when starting from a smaller first task.

In addition, we conduct experiments with the splits 50-50 and 75-25 which is equal to incremental training with new data from the same distribution (without access to all previous data). We observe in Figure 6 that the stability gap occurs in this setting too and is more pronounced than the corresponding 50-50* and 75*-25* setting studied before. The gap is larger from 0.65 to 0.20 and 0.70 to 0.23 as against 0.65 to 0.38 and 0.70 to 0.44, respectively.

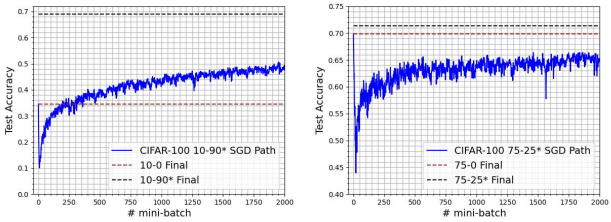


Figure 5. Using CIFAR-100 with ResNet-18, stability gap in (left) 10-90* (right) 75-25* setting. We can see that the stability gap increases for a smaller-sized first task.

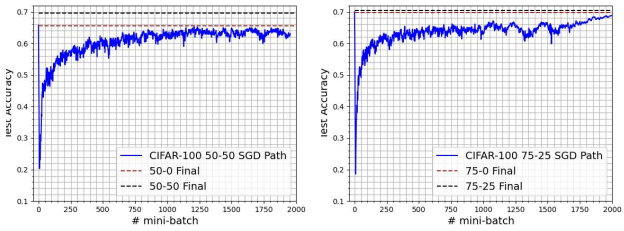


Figure 6. Using CIFAR-100 with ResNet-18, stability gap in (left) 50-50, (right) 75-25 setting. We can see that the stability gap increases when comparing (left) with the 50-50* setting in Fig. 1(right) and (right) with the 72-25* setting in Fig. 5(right).

## 3. Conclusions

In this article, we present compelling insights into the stability gap phenomenon. In particular, we show that it also manifests when applying joint incremental training on a sequence of homogeneous tasks, which is often considered the simplest scenario for continual learning. Through experimental evidence, we demonstrate that while the loss along the SGD path displays a stability gap, this discrepancy is not mirrored in the loss along the linear trajectory between checkpoints. An analysis at the mini-batch level showed that the gradient computed on the initial mini-batches (after the task-switch) does reduce the loss for each mini-batch but it results in an increased loss on the test data. We also observe that in the incremental learning with homogeneous tasks, when we remove rehearsal (going 50-50* to 50-50), the stability gap increases. In further research, we will explore this direction to possibly discover the cause of the stability gap and possible remedies.

**Acknowledgement.** We acknowledge projects TED2021-132513B-I00 and PID2022-143257NB-I00 funded by MCIN/AEI/10.13039/501100011033, by European Union NextGenerationEU/PRTR, by ERDF A Way of Making Europe, and by Generalitat de Catalunya CERCA Program.

## References


[1] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. 1, 2

[2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019. 1

[3] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 1

[4] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. 3

[5] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, 2018. 3

[6] Md Yousuf Harun and Christopher Kanan. Overcoming the stability gap in continual learning, 2023. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[8] Timm Hess, Tinne Tuytelaars, and Gido M. van de Ven. Two complementary perspectives to continual learning: Ask not only what to optimize, but also how, 2023. 1, 2, 3

[9] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2016. 1

[10] Matthias De Lange, Gido M van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2

[11] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2935–2947, 2016. 1

[12] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022. 1

[13] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989. 1

[14] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 2013. 1

[15] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning, 2020. 3

[16] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017. 1

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[18] Albin Soutif-Cormerais, Antonio Carta, and Joost van de Weijer. Improving online continual learning performance and stability with temporal ensembles. *CoRR*, abs/2306.16817, 2023. 1