Two complementary perspectives to continual learning:

ask not only *what* to optimize, but also *how*

Gido van de Ven

E-mail: gido.vandeven@kuleuven.be

Website: https://gmvandeven.github.io

Barcelona

23 May 2024

Overview

Introduction

➔ What is continual learning?

➔ Why is continual learning difficult?

➔ Why is continual learning important?

Main part

➔ Current approach to continual learning

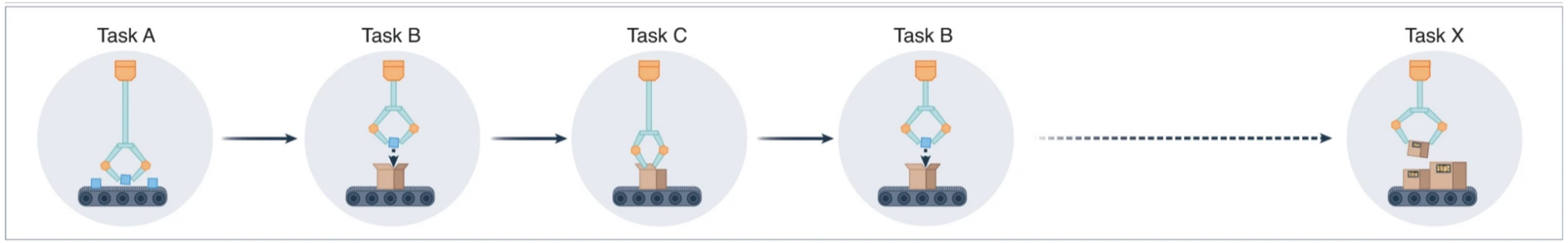
➔ Problem with the current approach to continual learning

➔ The road forward: continual learning needs a new perspective

What is continual learning?

What is continual learning?

- Continual learning is the problem of incrementally learning from a non-stationary stream of data
  - Non-stationary: distribution of data from which is learned changes over time
  - Incrementally: new learning should not overwrite what was learned before, but knowledge should be accumulated



Source: Kudithipudi et al. (2022, Nature Machine Intelligence)

What is continual learning?

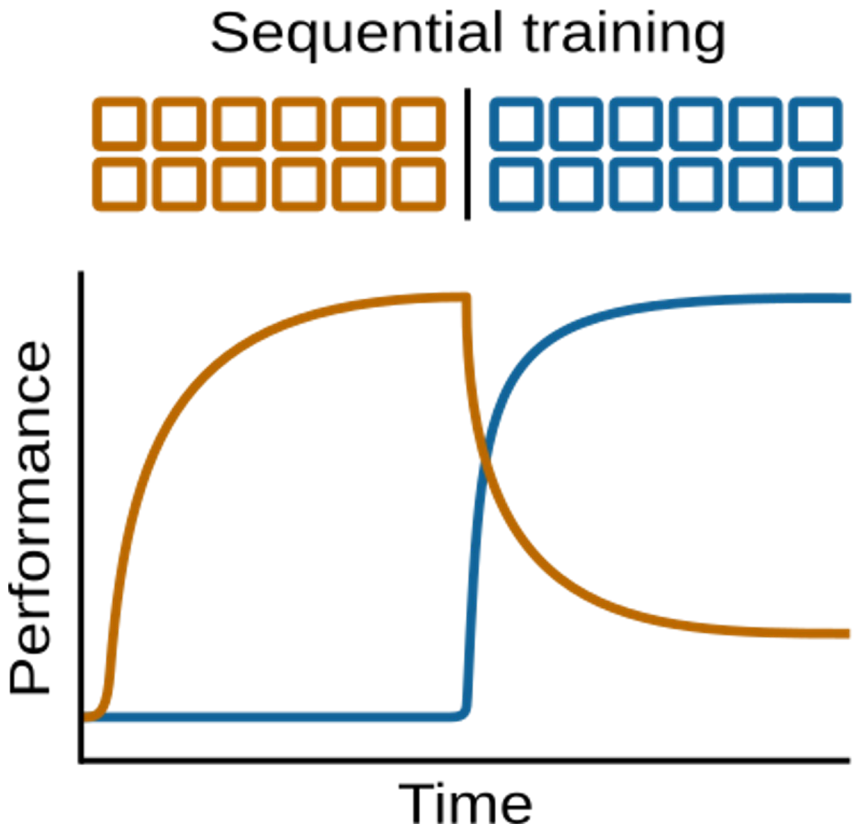
- Continual learning is the problem of incrementally learning from a non-stationary stream of data
  - Non-stationary: distribution of data from which is learned changes over time
  - Incrementally: new learning should not overwrite what was learned before, but knowledge should be accumulated

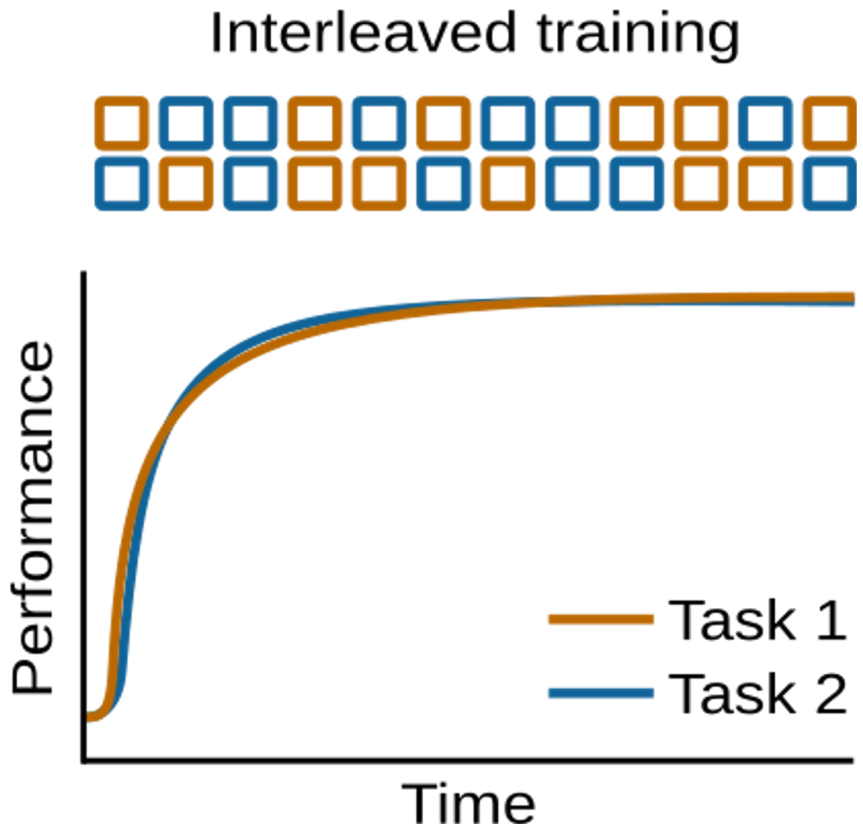
- Continual learning is a ***key aspect of intelligence***

- Deep neural networks largely lack continual learning abilities, especially compared to their biological counterparts

Why is continual learning difficult?

Catastrophic forgetting





Source: van de Ven, Soures et al. (2024, arXiv)

Catastrophic forgetting: expected or surprising?

- We first optimize the model for one task, and then we optimize it for another task. Why would the optimal solution for the second task still be good for the first one?

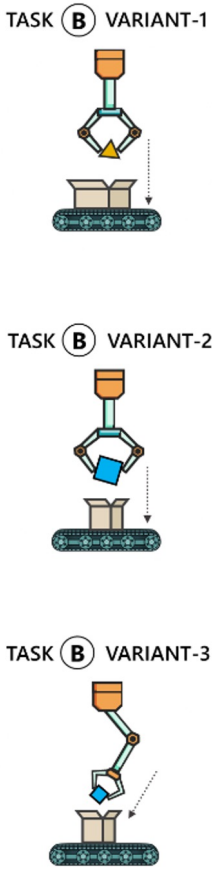
- One reasonable hypothesis:

  ``*If the modifications are all in the direction that helps the pattern that is being stored, there will be a conspiracy effect: The total help for the intended pattern will be the sum of all the small separate modifications. For unrelated patterns, however, there will be very little transfer of effect because some of the modifications will help and some will hinder. Instead of all the small modifications conspiring together, they will mainly cancel out.*'' (pp. 81-82; Hinton et al., 1986)

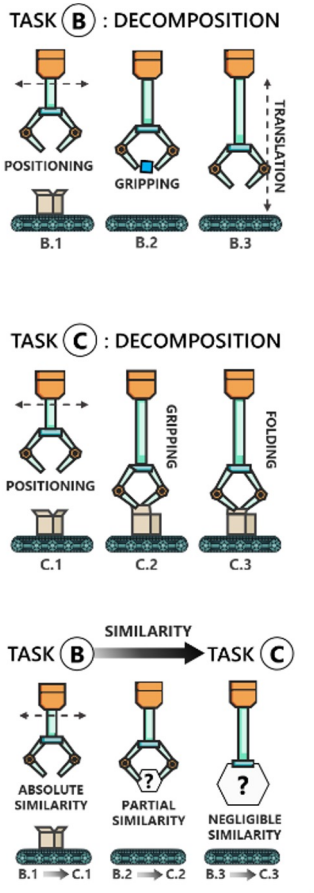
- McCloskey and Cohen (1989) and Ratcliff (1990) were the first to demonstrate catastrophic forgetting

Other features important for continual learning

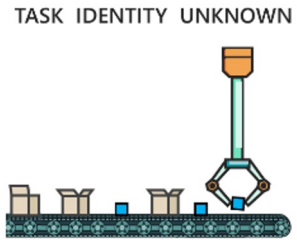
Rapid adaptation



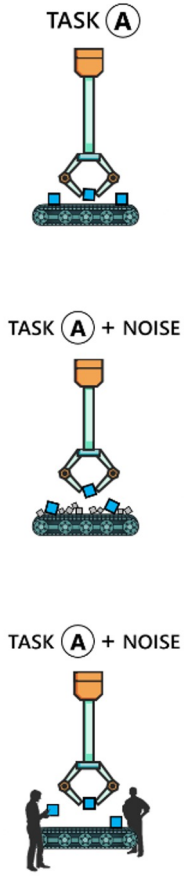
Exploiting task similarity



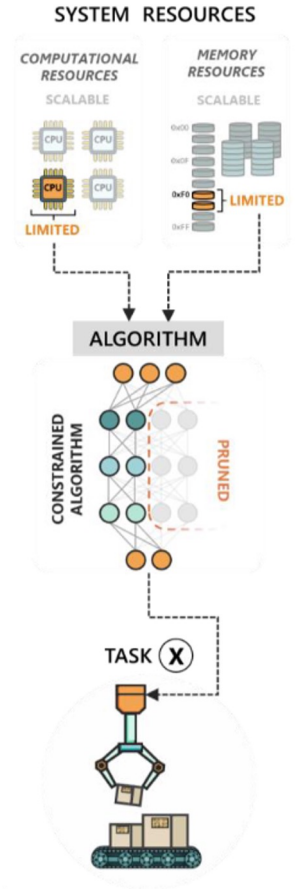
Task agnosticity



Noise tolerance



Resource efficiency and sustainability



Source: Kudithipudi et al. (2022, Nature Machine Intelligence)

Goal of continual learning

on both old and new tasks

e.g., in terms of memory and computation

Achieve an optimal trade-off between ***performance*** and ***resource efficiency***

Fine-tuning
Only train the model on the new data

- Bad performance
- Good resource efficiency

Continual learning strategies

Full retraining
Always fully train the model on all data so far

- Good performance
- Bad resource efficiency

Source: Verwimp et al. (2024, TMLR)

Why is continual learning important?

Potential applications of continual learning

1. Improve efficiency and substantially reduce required resources

Fine-tuning
Only train the model on the new data

- Bad performance
- Good resource efficiency

Continual learning strategies

Full retraining
Always fully train the model on all data so far

- Good performance
- Bad resource efficiency

Default approach in industry

Source: Verwimp et al. (2024, TMLR)

Potential applications of continual learning

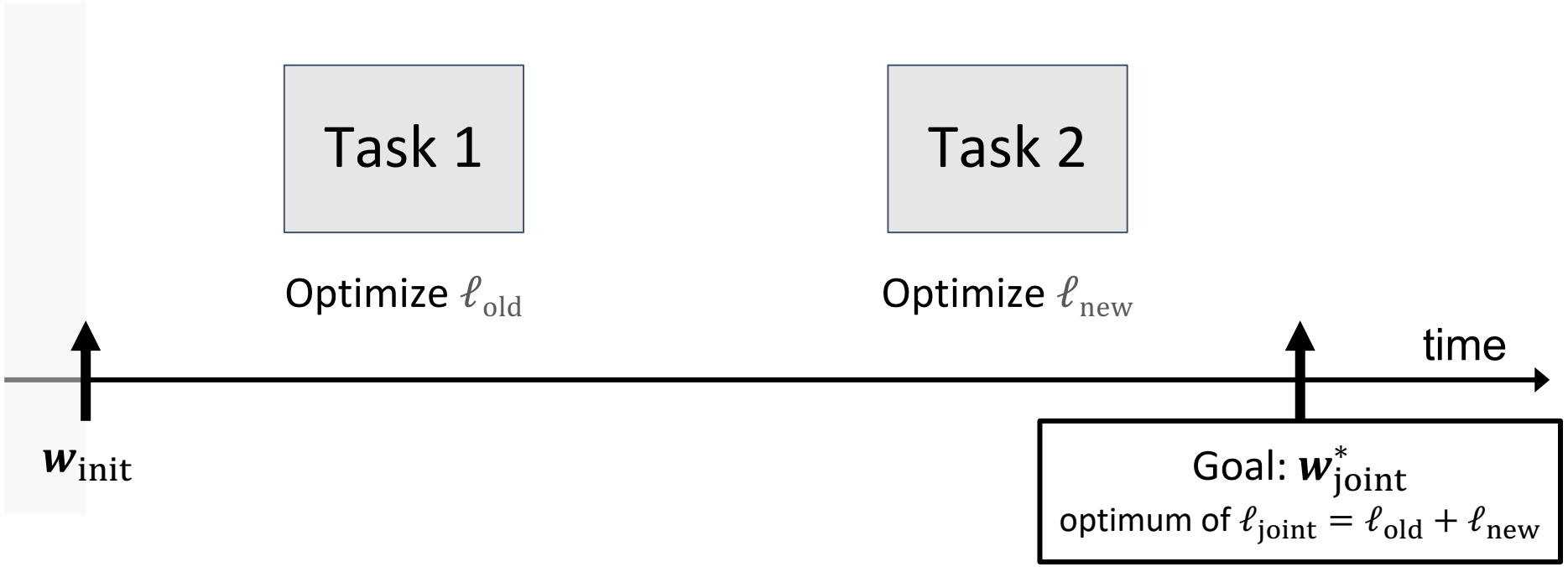
1. Improve efficiency and substantially reduce required resources
2. Correct mistakes or biases in trained networks (adapting locally)
3. On-device learning (e.g., personalization)
4. Reinforcement learning

Source: Verwimp et al. (2024, TMLR)

Current approach to continual learning

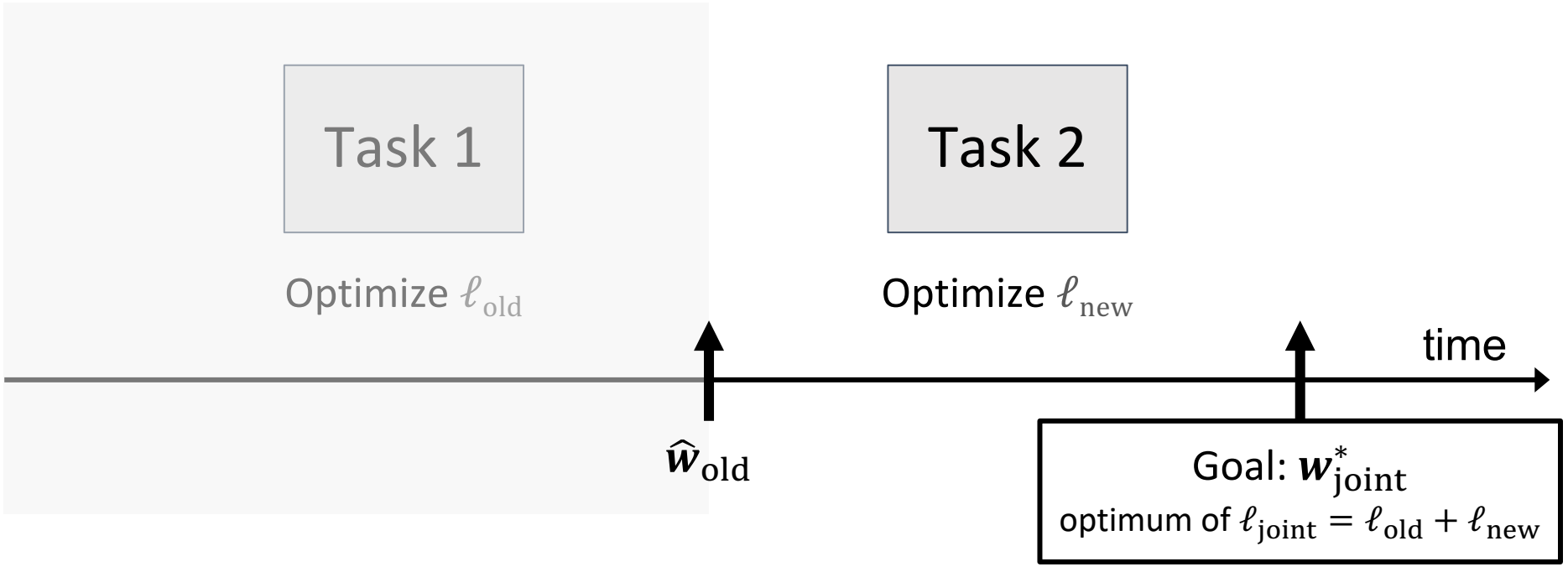
The core continual learning problem

→ Optimize the parameters w of a neural network f_w for two tasks that are observed one after the other



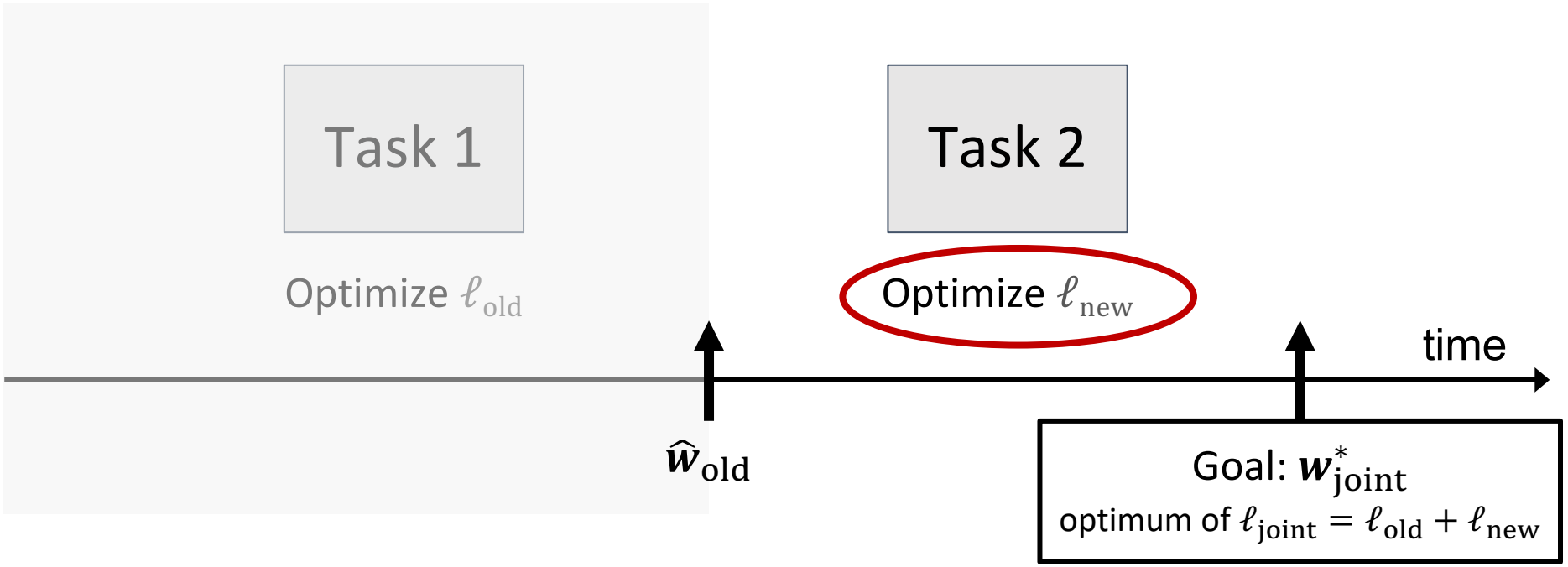
The core continual learning problem

→ Optimize the parameters w of a neural network f_w for two tasks that are observed one after the other



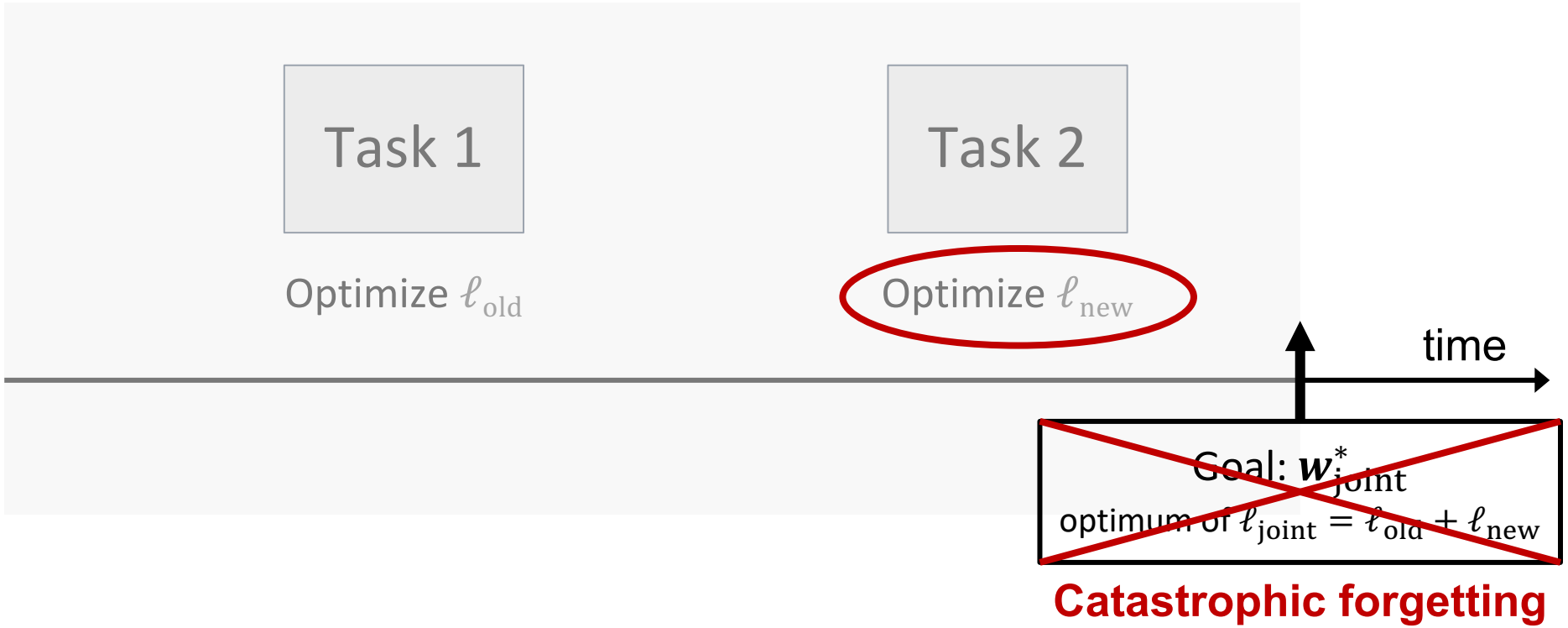
The core continual learning problem

→ Optimize the parameters w of a neural network f_w for two tasks that are observed one after the other



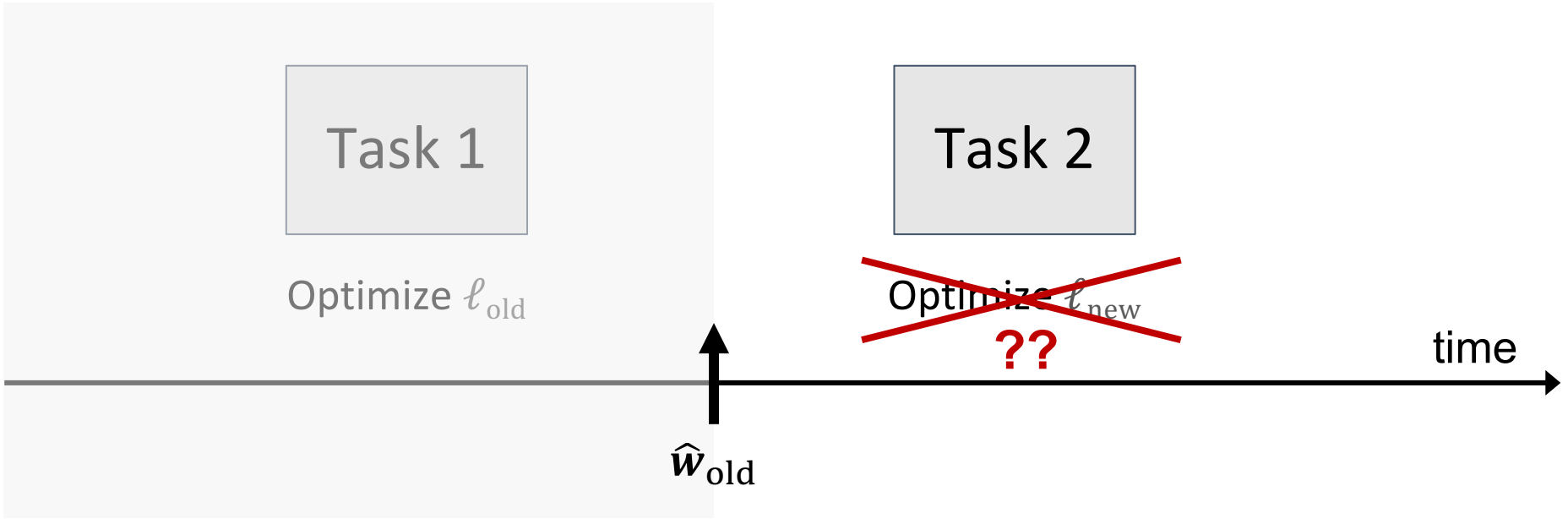
The core continual learning problem

→ Optimize the parameters w of a neural network f_w for two tasks that are observed one after the other

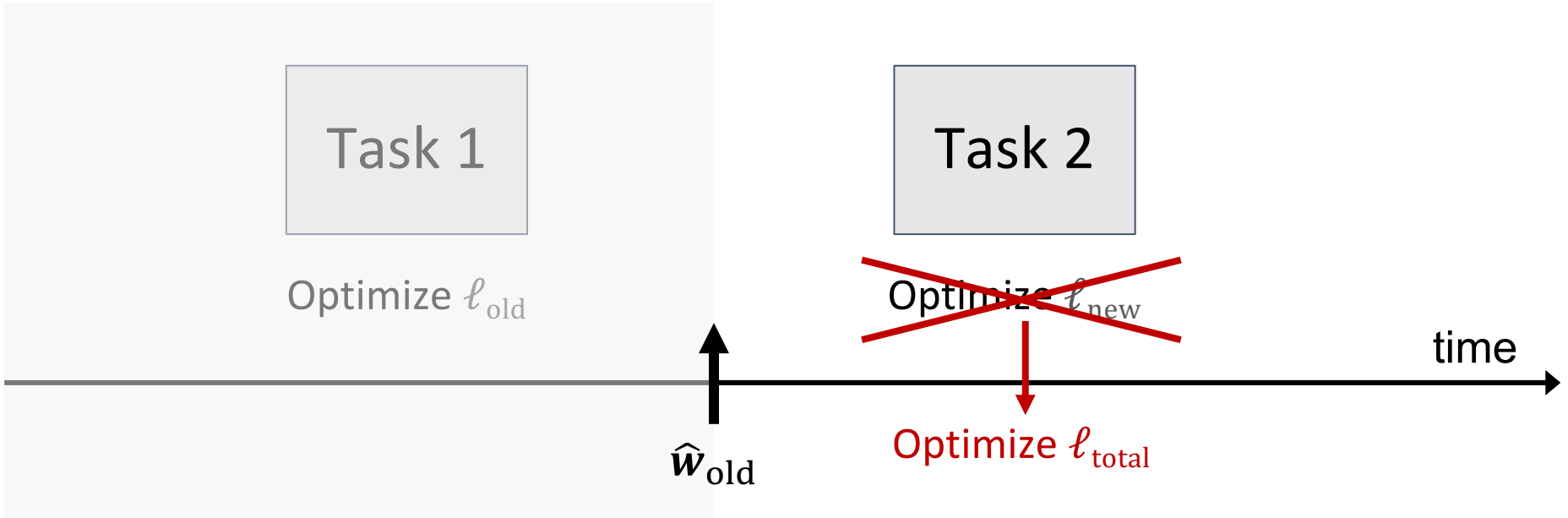


The core continual learning problem

→ Optimize the parameters w of a neural network f_w for two tasks that are observed one after the other

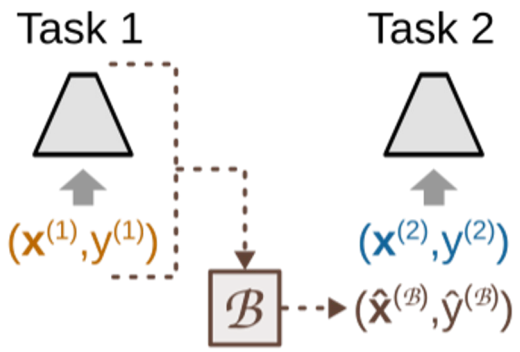


Current approach to continual learning: make changes to the loss



Current approach to continual learning: make changes to the loss

Replay

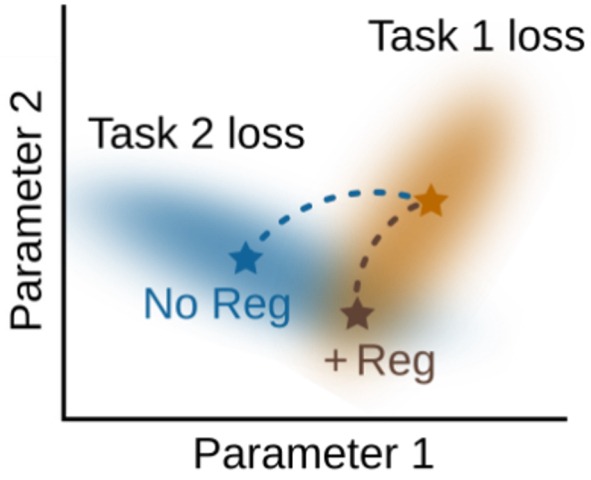


$$\ell_{\text{total}} = \ell_{\text{new}} + \ell_{\text{replay}}$$

$$\approx$$

$$\tilde{\ell}_{\text{old}}$$

Parameter regularization

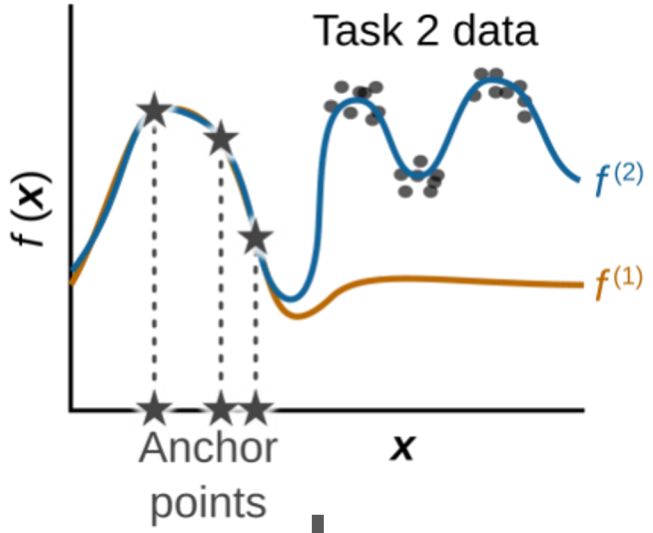


$$\ell_{\text{total}} = \ell_{\text{new}} + \|w - \hat{w}_{\text{old}}\|_{\Sigma}$$

$$\approx$$

$$\tilde{\ell}_{\text{old}}$$

Functional regularization



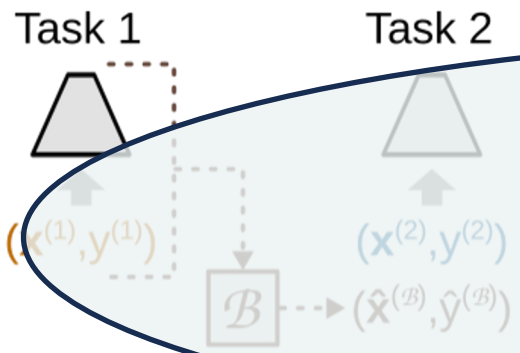
$$\ell_{\text{total}} = \ell_{\text{new}} + \langle f_w, f_{\hat{w}_{\text{old}}} \rangle_{\mathcal{A}}$$

$$\approx$$

$$\tilde{\ell}_{\text{old}}$$

Current approach to continual learning: make changes to the loss

Replay

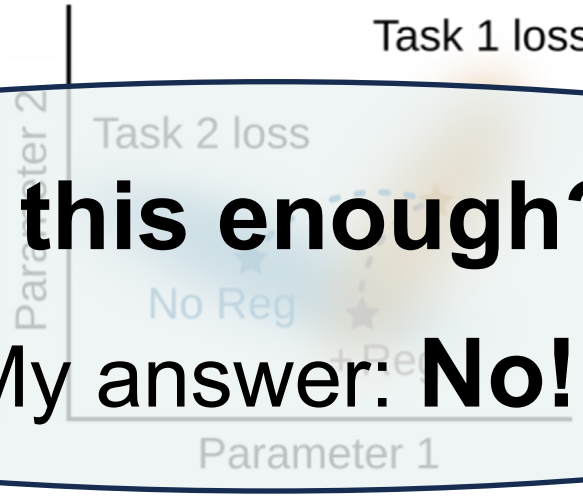


$$\ell_{\text{total}} = \ell_{\text{new}} + \ell_{\text{replay}}$$

$$\approx$$

$$\tilde{\ell}_{\text{old}}$$

Parameter regularization

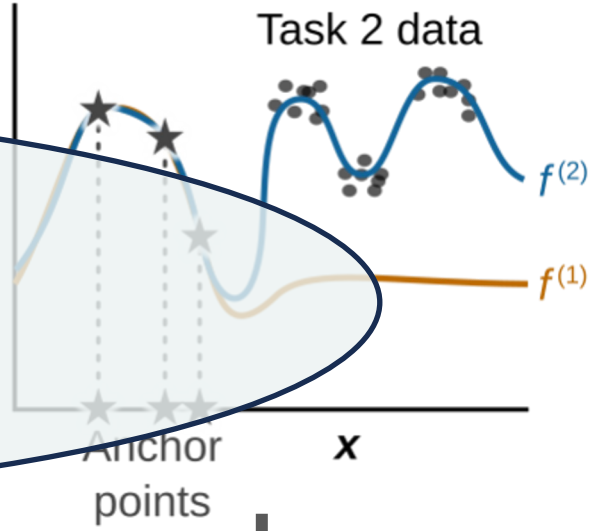


$$\ell_{\text{total}} = \ell_{\text{new}} + \|\boldsymbol{w} - \hat{\boldsymbol{w}}_{\text{old}}\|_{\Sigma}$$

$$\approx$$

$$\tilde{\ell}_{\text{old}}$$

Functional regularization



$$\ell_{\text{total}} = \ell_{\text{new}} + \langle f_{\boldsymbol{w}}, f_{\hat{\boldsymbol{w}}_{\text{old}}} \rangle_{\mathcal{A}}$$

$$\approx$$

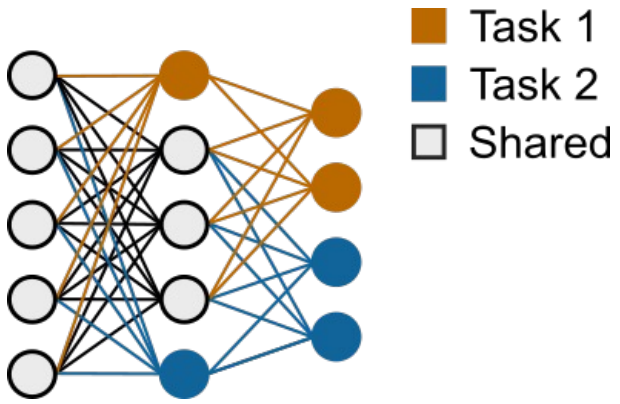
$$\tilde{\ell}_{\text{old}}$$

Is this enough?

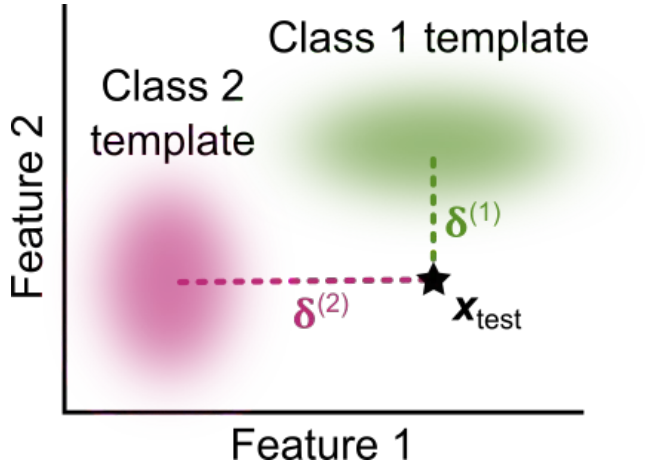
My answer: **No!**

Other approaches to continual learning

Context-specific components



Template-based classification

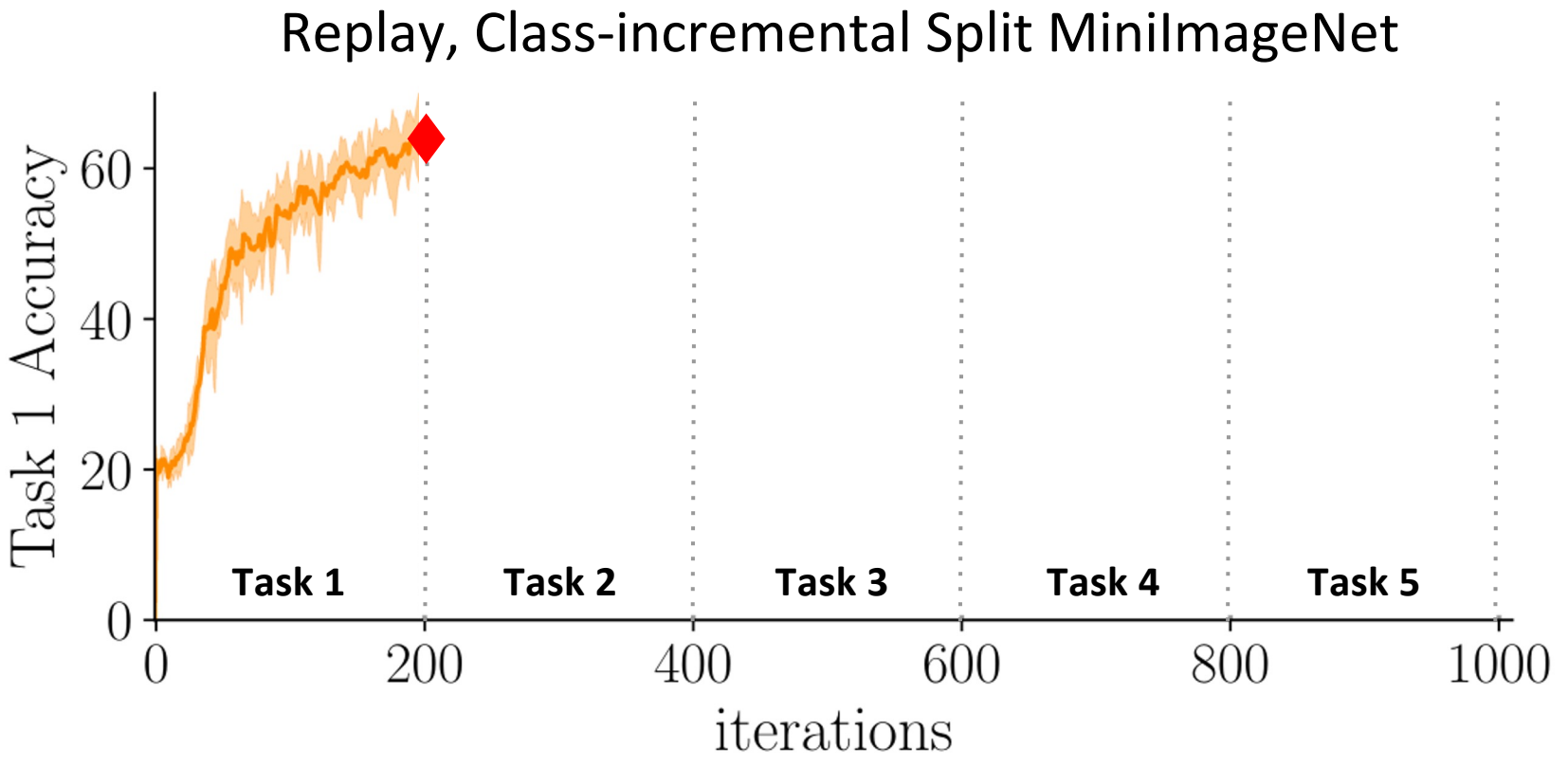


However, these specialized approaches are not approaches for optimizing a shared set of parameters in a continual manner

These approaches are useful and important, but not what I mean here

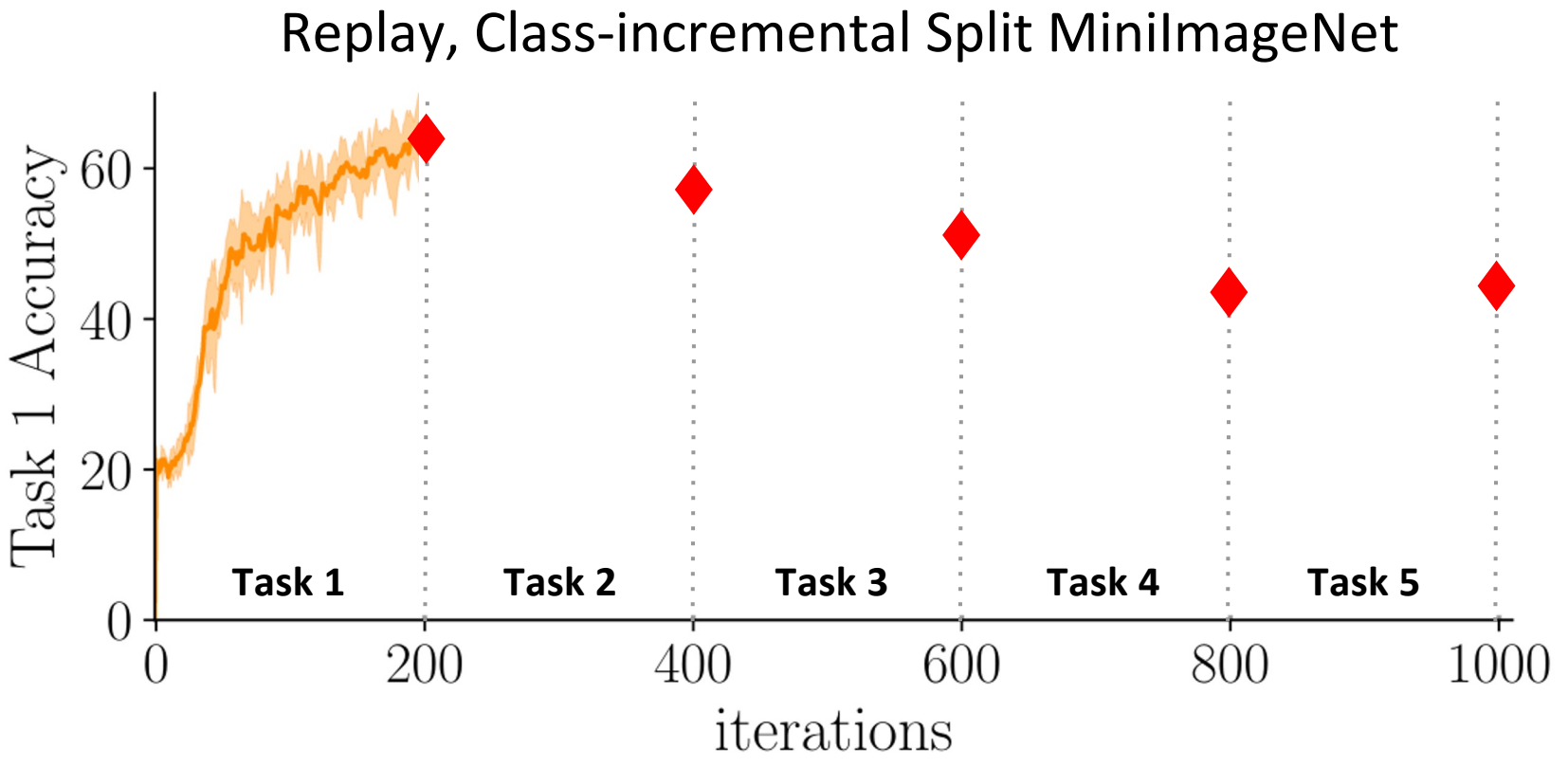
The Stability Gap:
A problem with the current approach to CL

Does replay prevent forgetting?



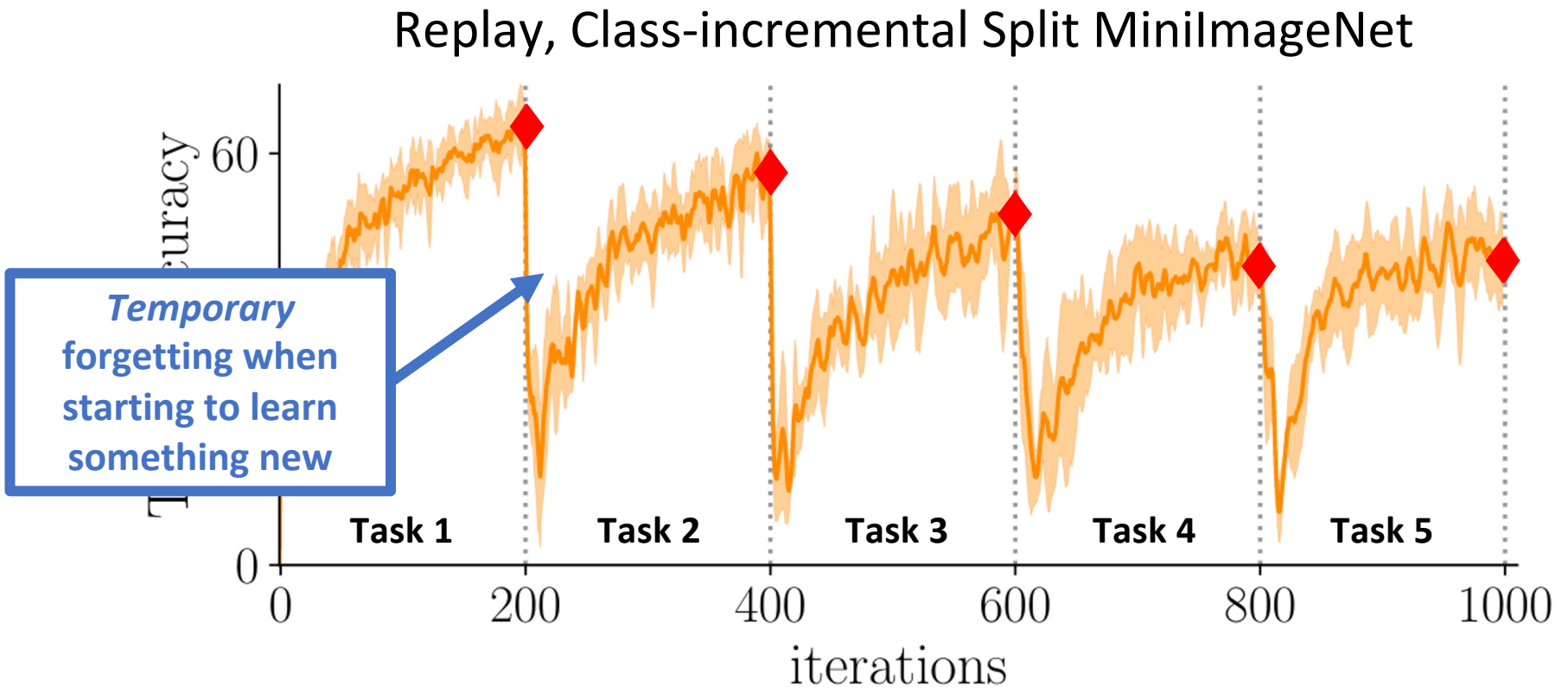
De Lange, **van de Ven** & Tuytelaars (2023) "Continual evaluation for lifelong learning: Identifying the stability gap" *ICLR, spotlight*

Does replay prevent forgetting?



De Lange, **van de Ven** & Tuytelaars (2023) "Continual evaluation for lifelong learning: Identifying the stability gap" *ICLR, spotlight*

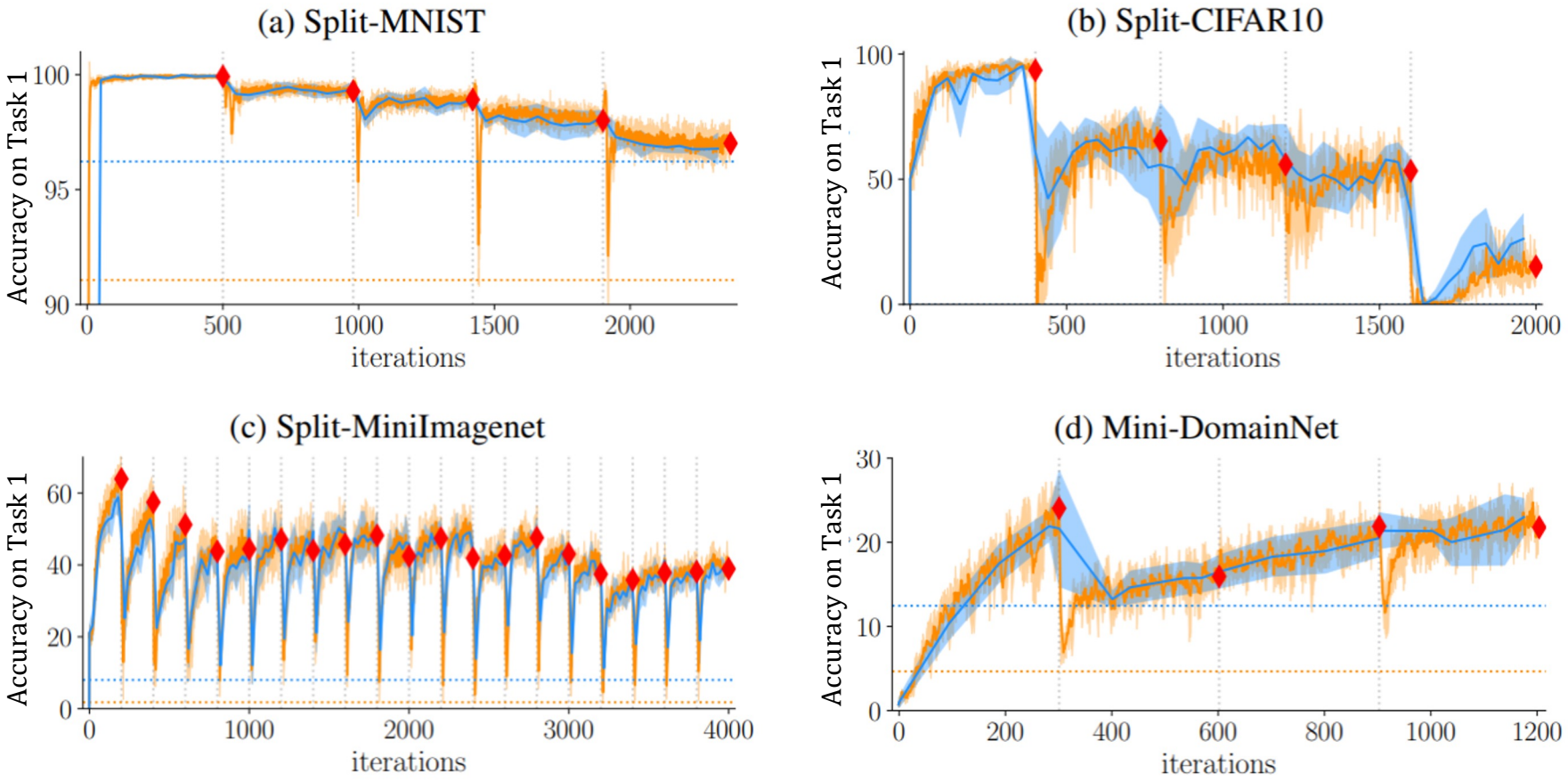
Does replay prevent forgetting?



De Lange, **van de Ven** & Tuytelaars (2023) "Continual evaluation for lifelong learning: Identifying the stability gap" *ICLR, spotlight*

The stability gap is consistently observed

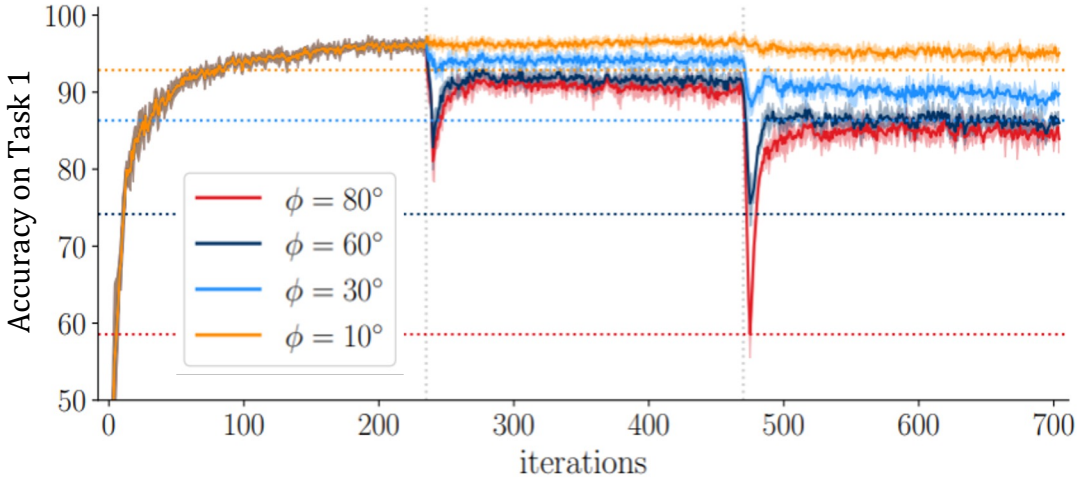
Replay, Class-incremental on …



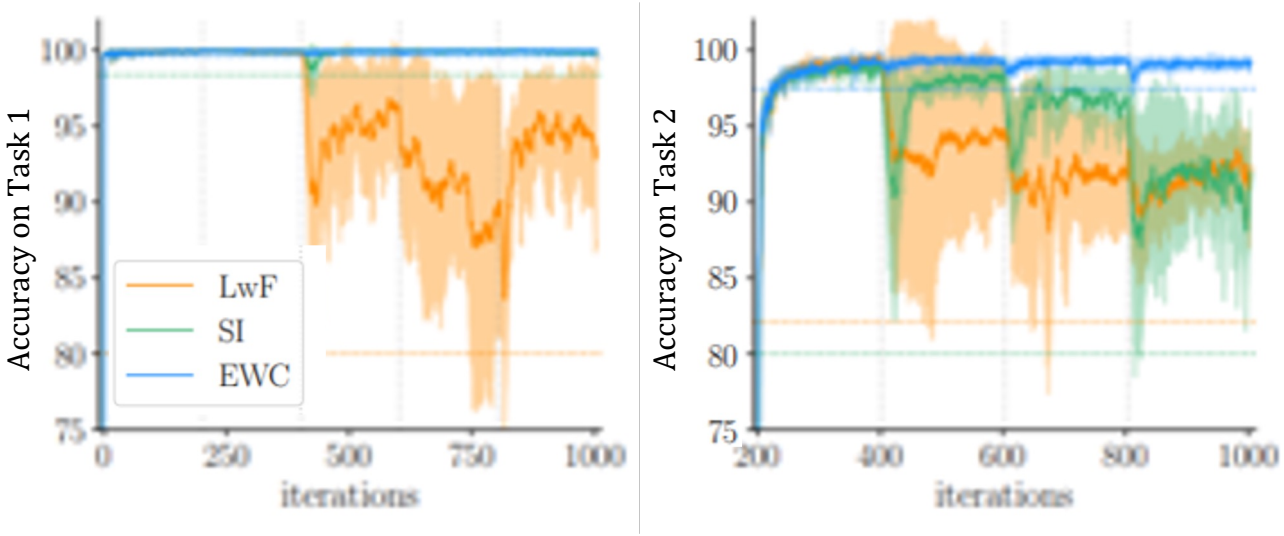
De Lange, **van de Ven** & Tuytelaars (2023) "Continual evaluation for lifelong learning: Identifying the stability gap" *ICLR, spotlight*

… also in other settings or with other methods

Replay, Domain-incremental Rotated MNIST



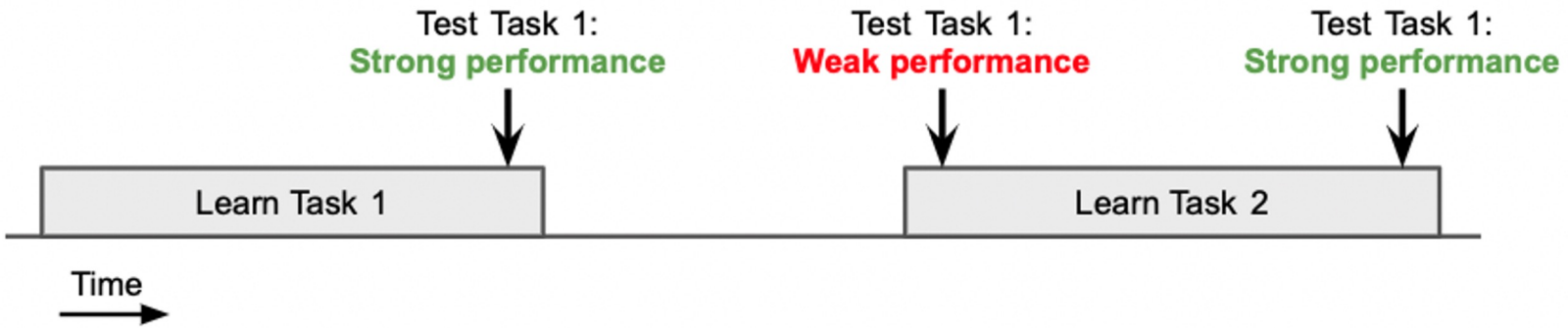
Regularization, Task-incremental Split MNIST



De Lange, **van de Ven** & Tuytelaars (2023) "Continual evaluation for lifelong learning: Identifying the stability gap" *ICLR, spotlight*

Why should we care?

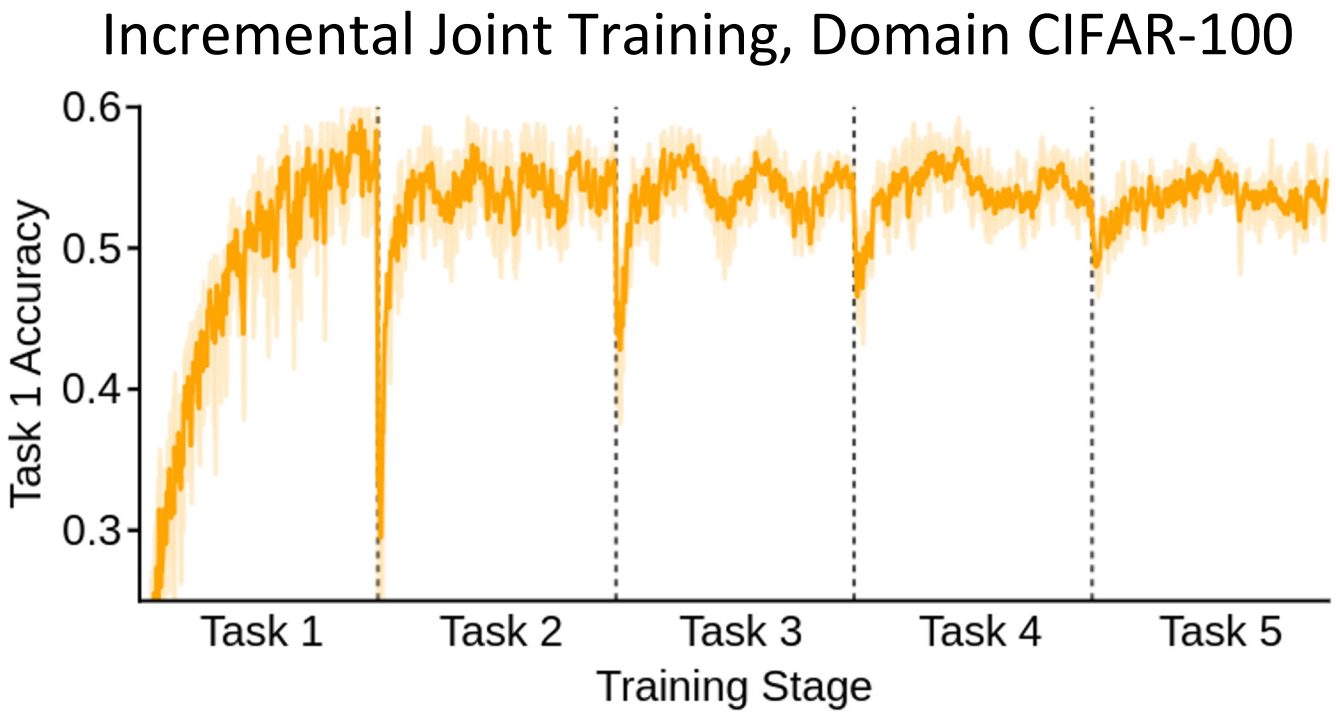
- Problematic for safety-critical applications
  - Worst-case performance might be important
  - Could be exploitable by adversarial agent with control over the training stream
- Could avoiding the stability gap lead to better *final performance*?
  - Preventing forgetting seems more efficient than having to re-learn
- Scientifically interesting
  - Do humans suffer from transient forgetting upon learning something new?



How to avoid the stability gap?

- Continue to improve the quality of replay?

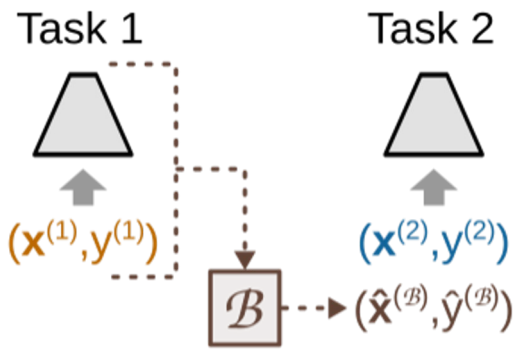
The stability gap occurs even with “perfect” replay!



Hess, Tuytelaars & **van de Ven** (2023) “Two Complementary Perspectives to Continual Learning” *Pre-registered Report, 1st ContinualAI Unconference*

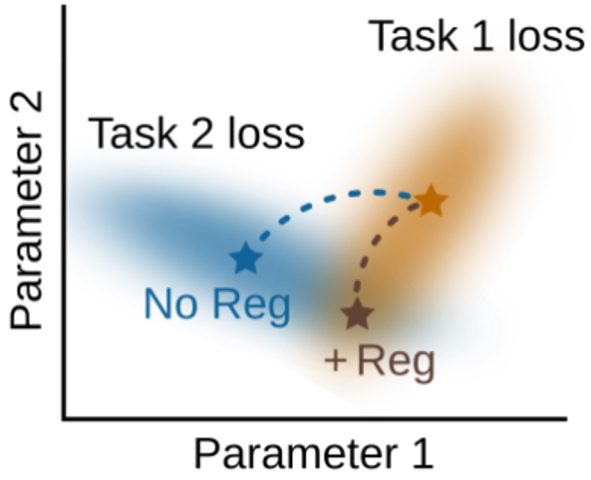
Current approach to continual learning: make changes to the loss

Replay



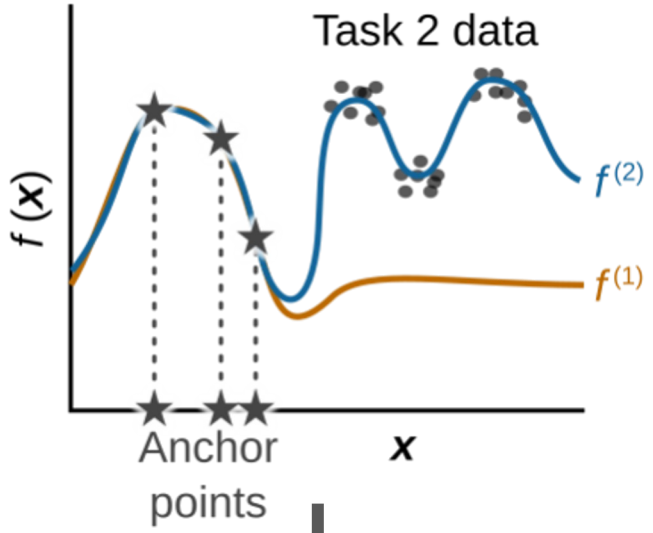
$$\ell_{\text{total}} = \ell_{\text{new}} + \ell_{\text{replay}}$$

Parameter regularization



$$\ell_{\text{total}} = \ell_{\text{new}} + \|\boldsymbol{w} - \hat{\boldsymbol{w}}_{\text{old}}\|_{\Sigma}$$

Functional regularization

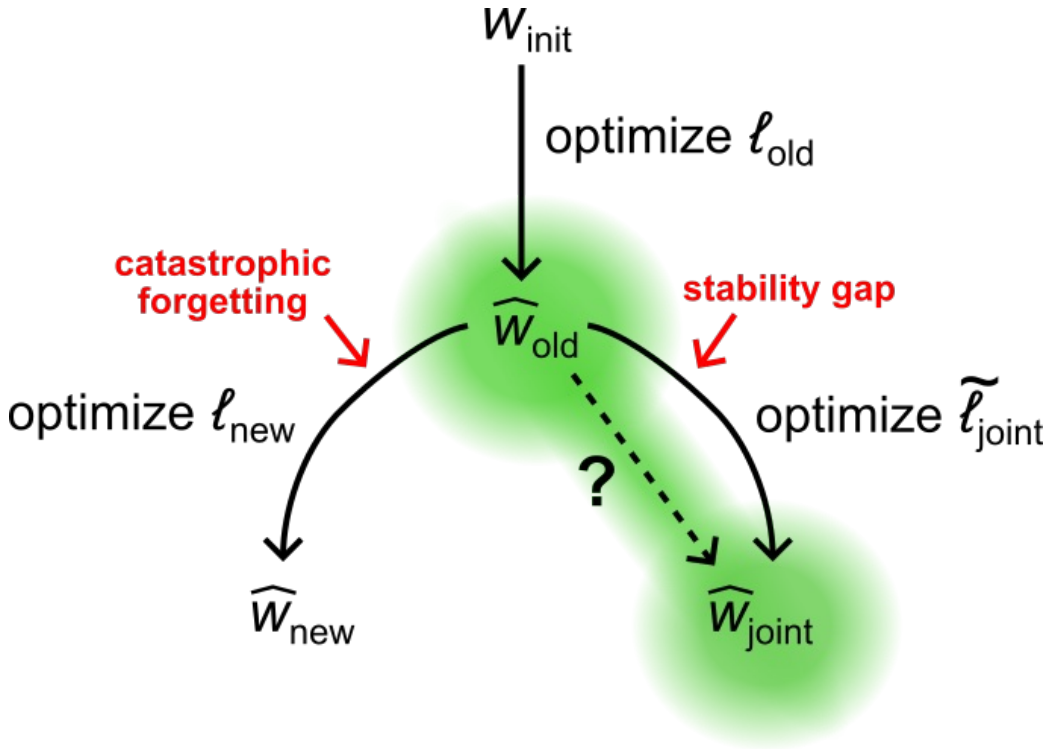


$$\ell_{\text{total}} = \ell_{\text{new}} + \langle f_{\boldsymbol{w}}, f_{\hat{\boldsymbol{w}}_{\text{old}}} \rangle_{\mathcal{A}}$$

Even with a perfect approximation to the joint loss, the stability gap persists!

How to solve this?

Can the stability gap be solved?



= Area in parameter space with low loss on old tasks

Work on mode connectivity suggests there is a low-loss path from $\widehat{\boldsymbol{w}}_{\text{old}}$ to $\widehat{\boldsymbol{w}}_{\text{joint}}$
(Draxler et al., 2018; Garipov et al., 2018; Mirzadeh et al., 2021)

Continual learning needs a new perspective

- To overcome the stability gap, changes must be made to *how* the loss function is optimized
- Standard optimization routines for deep learning (SGD and its variants) have been developed for the stationary setting
- No guarantees in continual setting, yet widely used
- Fundamental difference between both settings:
  - Stationary → start from random initialization
  - Continual → start from partial solution

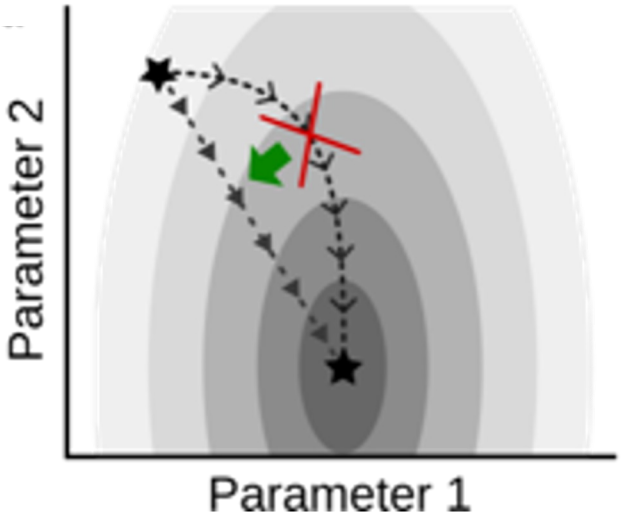
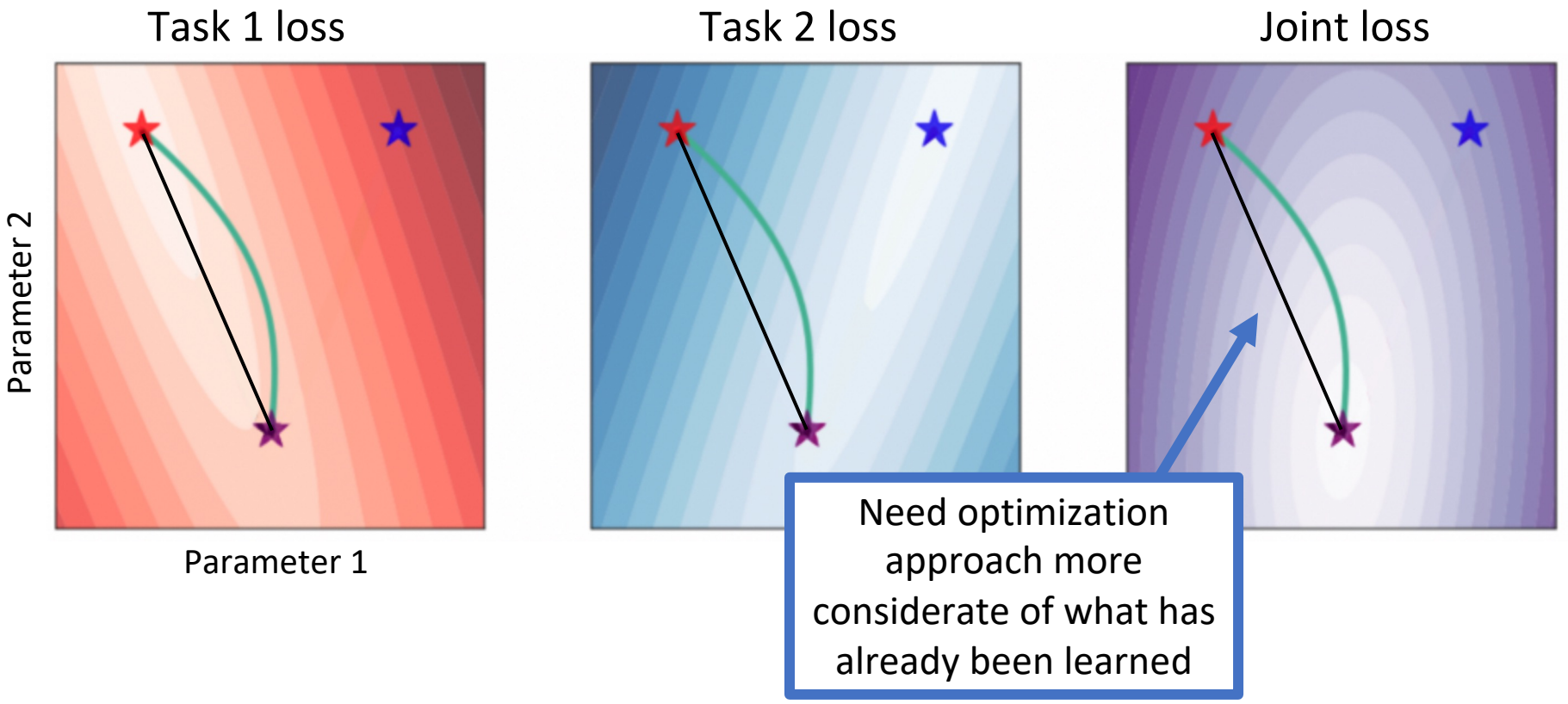


Illustration with a toy example



Kao, Jensen, **van de Ven**, Bernacchia & Hennequin (2021) "Natural continual learning: success is a journey, not (just) a destination" *NeurIPS*

How to improve optimization for continual learning?

- An exciting open question!

Existing CL work that focuses on the optimization process:

- Explore influence of optimizer (Adam, SGD, …) on CL performance
- Change optimization routine to encourage flatter minima
- Gradient projection

Gradient projection-based optimization

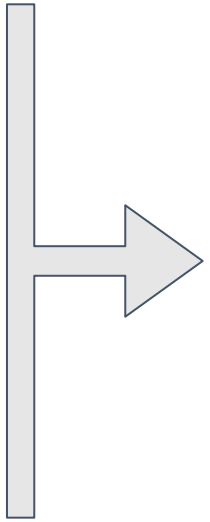
- Modifies the way a given loss function $\ell(w)$ is optimized
- Rather than basing parameter updates on original gradient $g = \nabla_w \ell(w)$, they are based on a projected version $\bar{g}$ of that gradient

Orthogonal gradient projection (e.g., OWM, OGD, GPM)

- Project g to the orthogonal complement of the 'gradient subspaces' of old tasks
- Trains each task in a different subspace of the network

Gradient episodic memories (e.g., GEM, A-GEM)

- Motivated by the constrained optimization problem of optimizing ℓ_{new} without increasing ℓ_{old}



These approaches use a modified optimization routine to optimize ℓ_{new}

My proposal is that they should be used to optimize $\tilde{\ell}_{\text{joint}}$

Funding acknowledgements

The work and research towards preparing this presentation has been supported by an IBRO-ISN Research Fellowship, by the *Lifelong Learning Machines* (L2M) program of the Defence Advanced Research Projects Agency (DARPA) via contract number HR0011-18-2-0025 and by funding from the European Union under the Horizon 2020 research and innovation program (ERC project *KeepOnLearning*, grant agreement No. 101021347) and under Horizon Europe (Marie Skłodowska-Curie fellowship, grant agreement No. 101067759).