Brain-inspired replay for continual learning with artificial neural networks

Gido van de Ven, Hava Siegelmann & Andreas Tolias

For full details:

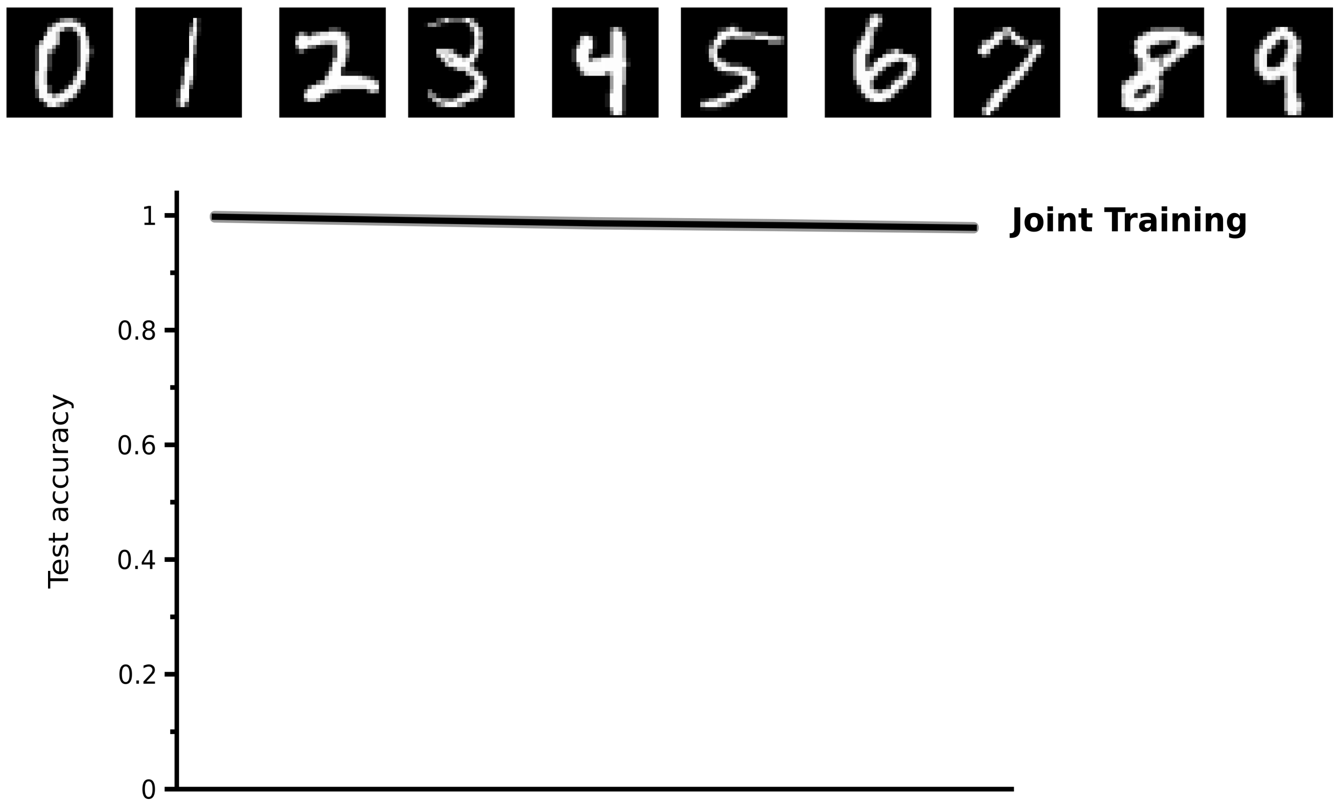
van de Ven GM, Siegelmann HT, Tolias AS (2020) Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, **11**: 4069.

Code: https://github.com/GMvandeVen/brain-inspired-replay

Motivation 1: alleviate catastrophic forgetting

- Artificial neural networks suffer from catastrophic forgetting:
  When trained on a new task, they rapidly forget previously learned tasks

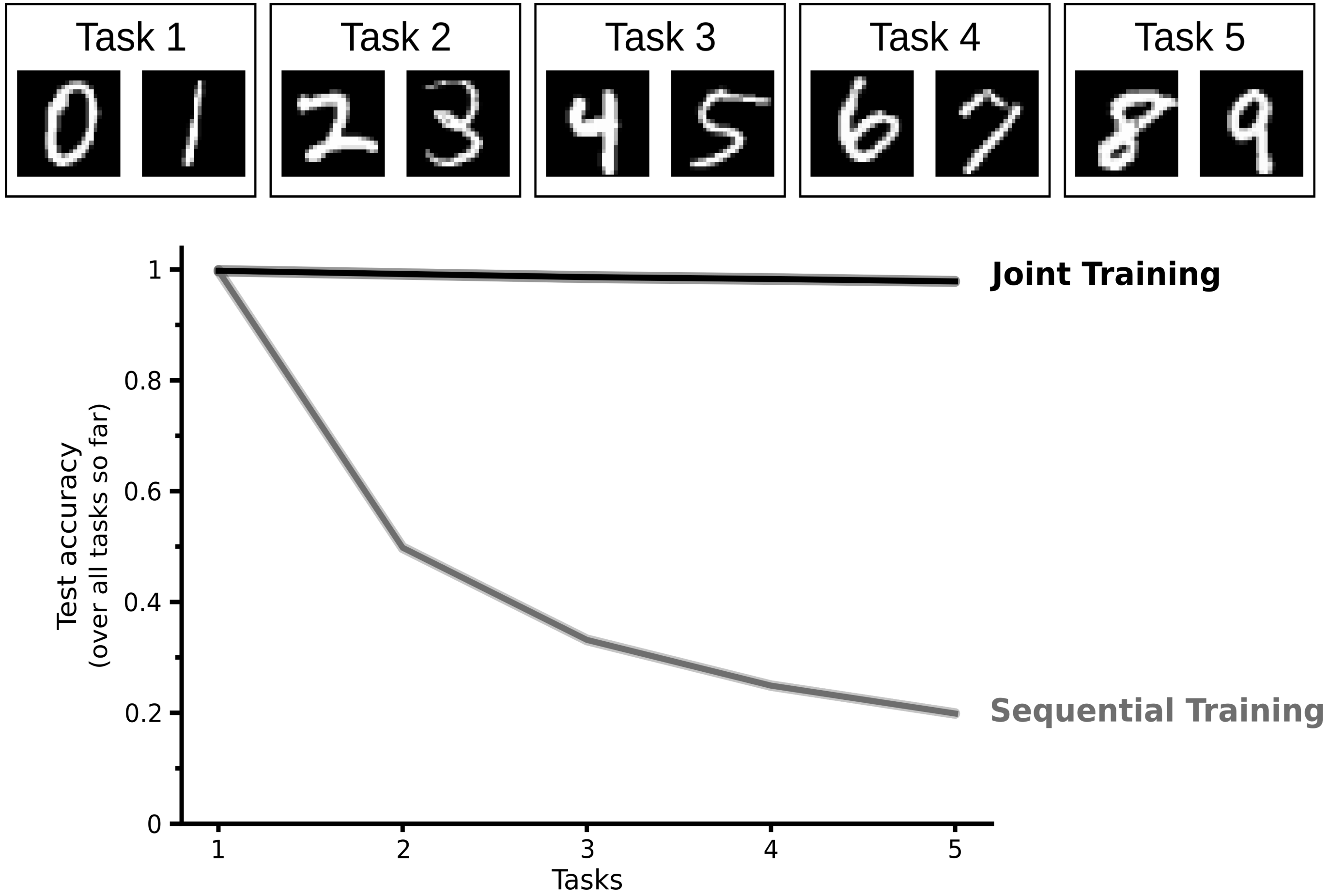
[McCloskey & Cohen, 1989 *Psych Learn Motiv*; Ratcliff, 1990 *Psych Rev*]



Motivation 1: alleviate catastrophic forgetting

- Artificial neural networks suffer from catastrophic forgetting:
  When trained on a new task, they rapidly forget previously learned tasks

[McCloskey & Cohen, 1989 *Psych Learn Motiv*; Ratcliff, 1990 *Psych Rev*]



Motivation 1: alleviate catastrophic forgetting

- Artificial neural networks suffer from catastrophic forgetting:
  When trained on a new task, they rapidly forget previously learned tasks
  [McCloskey & Cohen, 1989 *Psych Learn Motiv*; Ratcliff, 1990 *Psych Rev*]
- When it comes to continual learning, biological neural networks are far superior to their artificial counterparts

Motivation 2: computational model for replay

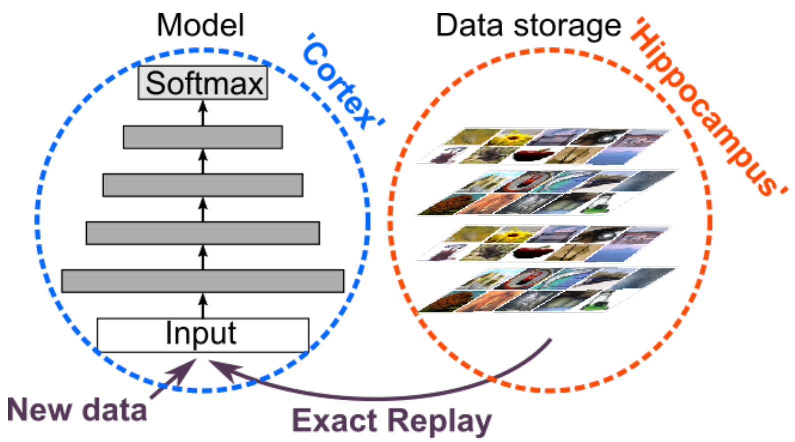
- In the brain, new memories are initially labile too
- Empirical evidence for a role of replay in memory consolidation
  [Wilson & McNaughton, 1994 *Science;* Rasch & Born, 2007 *Curr Opin Neurobiol*; van de Ven et al., 2016 *Neuron*]

--> Artificial neural networks as "*model organism*" for "*gain-of-function*" experiment:

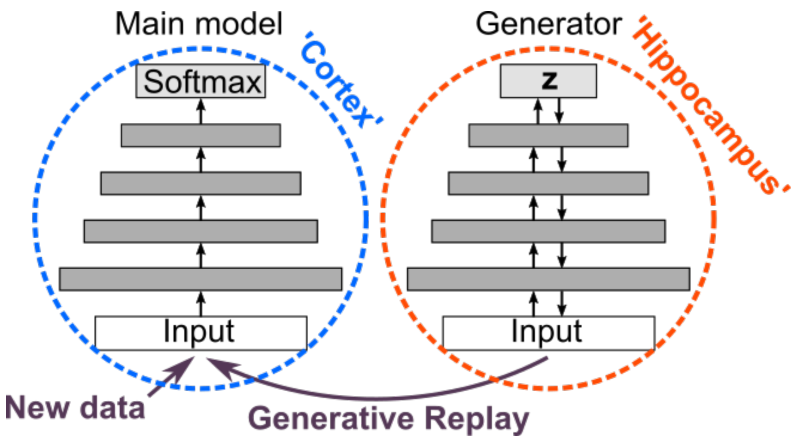
Could replay improve memory consolidation in artificial neural networks?

How to add replay to artificial neural networks?

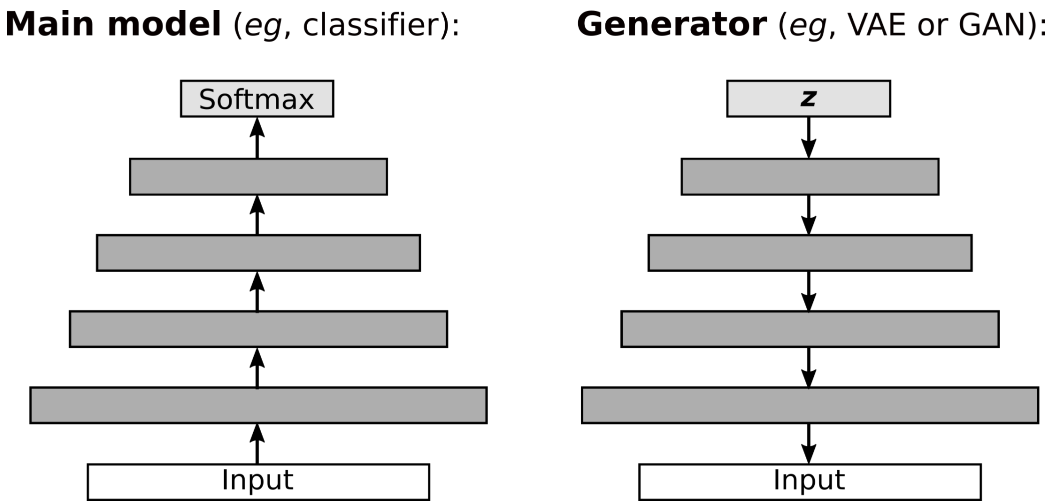
- Store data and interleave – *"exact"* or *"experience replay"*
  - *Initial argument for role of replay in memory consolidation in the brain* [McClelland et al., 1995 *Psych Rev*]
  - *Unclear how the brain could do directly store data*
  - Not always possible (e.g., privacy concerns, limited storage)
  - Problematic when scaling up to true lifelong learning



- Use a generative model – *"generative replay"*
  - *More realistic from neuroscience point of view*
  - *Views hippocampus as a generative neural network and replay as a generative process; see also* [Liu et al., 2018 *Neuron*; Liu et al., 2019 *Cell*]
  - Learning a generative model as a more scalable, privacy-preserving way of remembering previous seen data

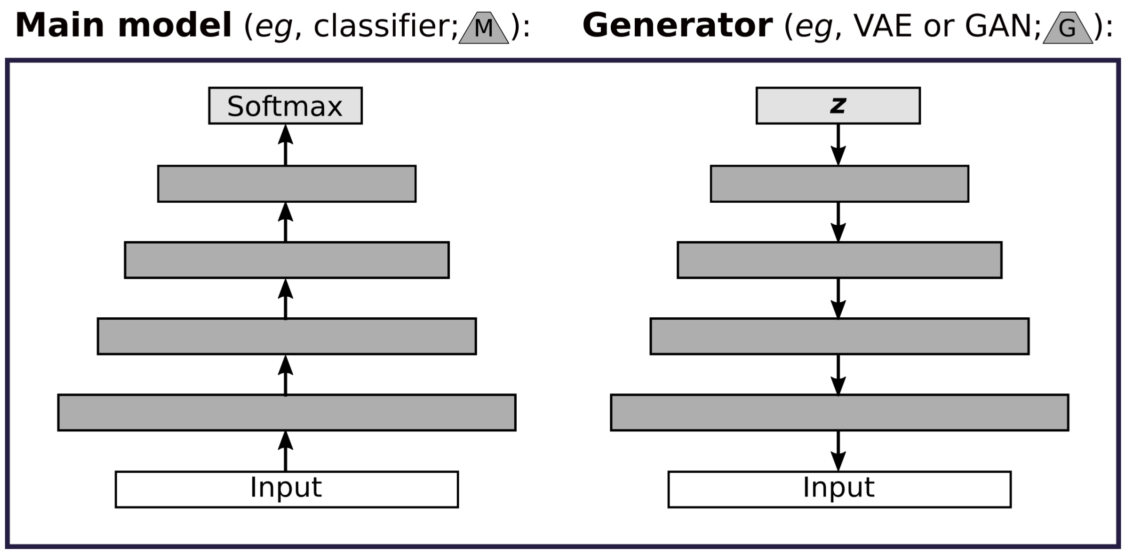


Generative replay

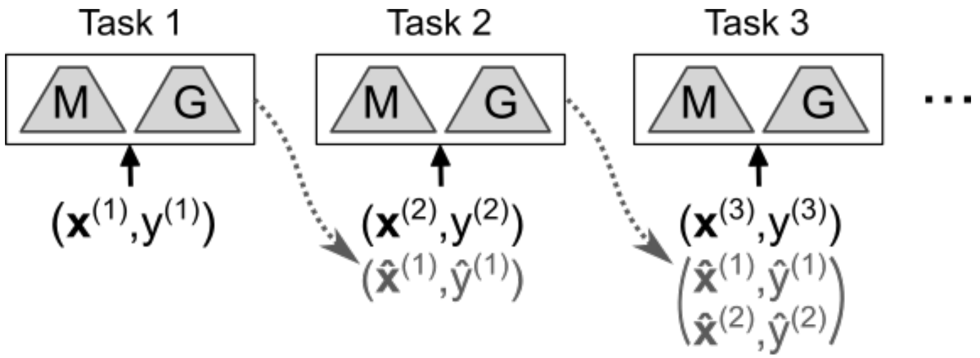


Shin et al. (2017) *NeurIPS*; van de Ven & Tolias (2018) *arXiv*

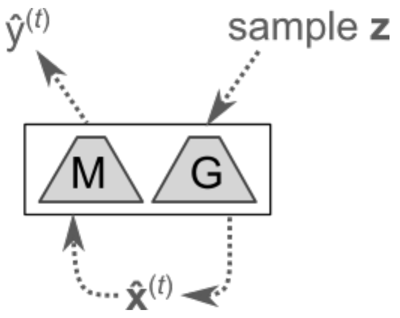
Generative replay



Incremental training protocol:



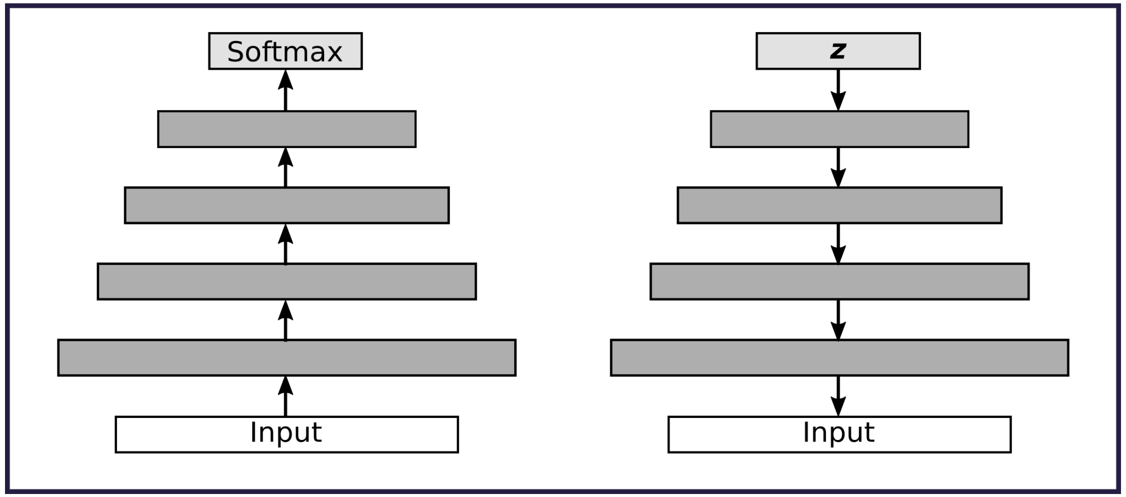
Generation of a sample to be replayed:



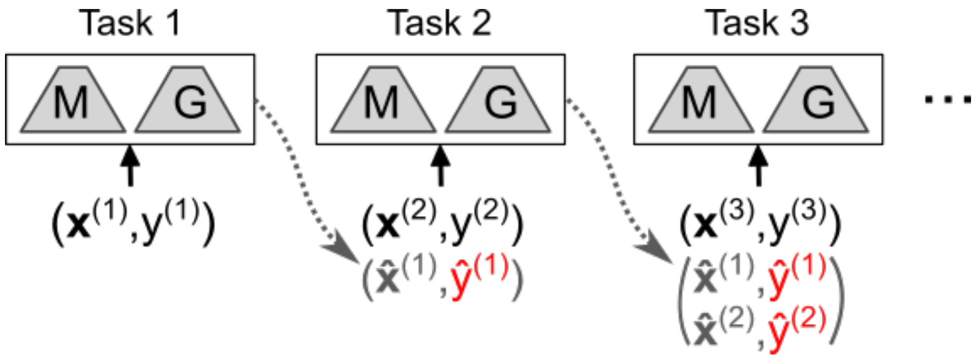
Shin et al. (2017) *NeurIPS*; van de Ven & Tolias (2018) *arXiv*

Generative replay

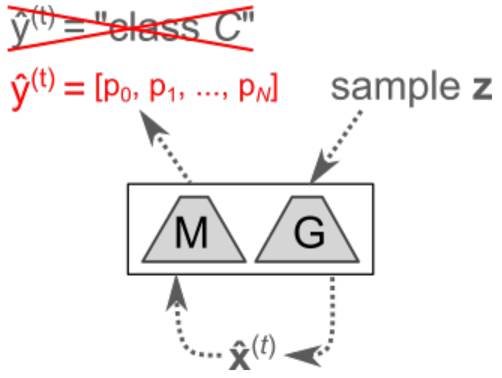
Main model (*eg*, classifier; M): **Generator** (*eg*, VAE or GAN; G):



Incremental training protocol:



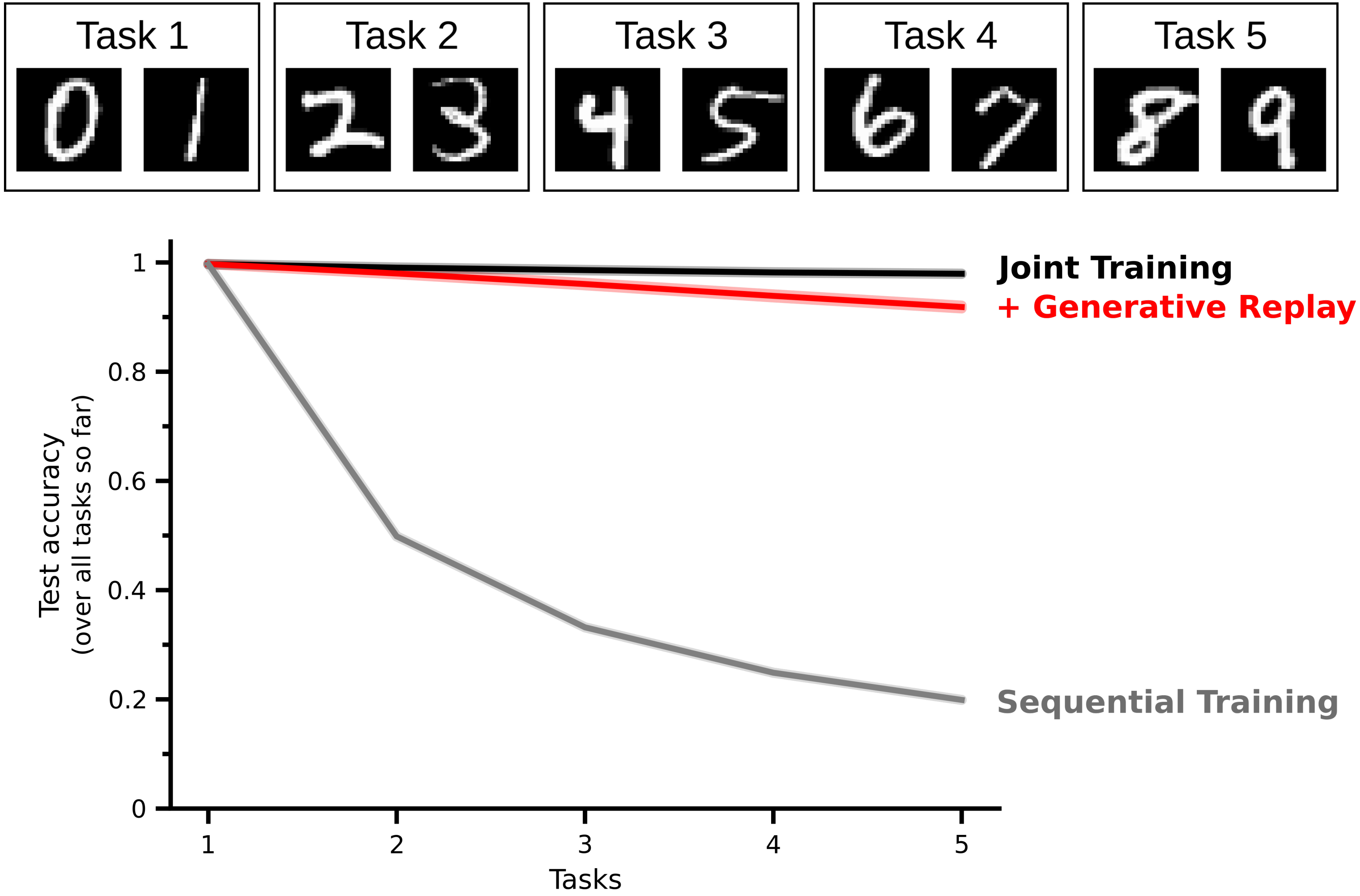
Generation of a sample to be replayed:



Use *distillation* for replayed data:
Label generated inputs with the by the previous model predicted probabilities for *all* classes ("soft targets"), instead of only with the predicted most likely class ("hard targets")

Shin et al. (2017) *NeurIPS*; van de Ven & Tolias (2018) *arXiv*

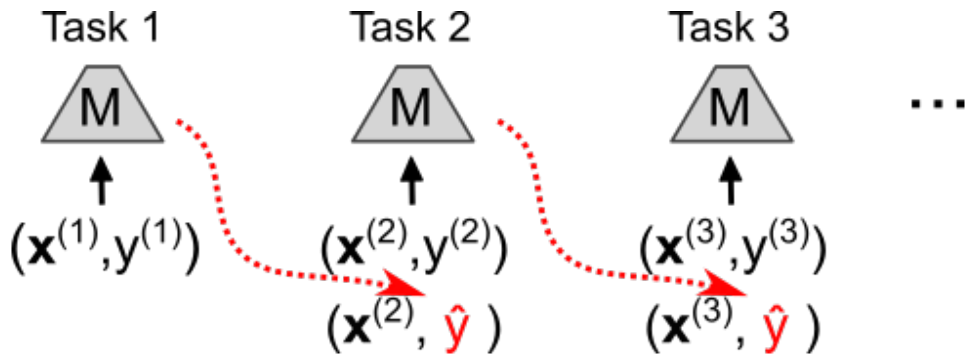
Generative replay can prevent Catastrophic Forgetting



Comparison with other methods

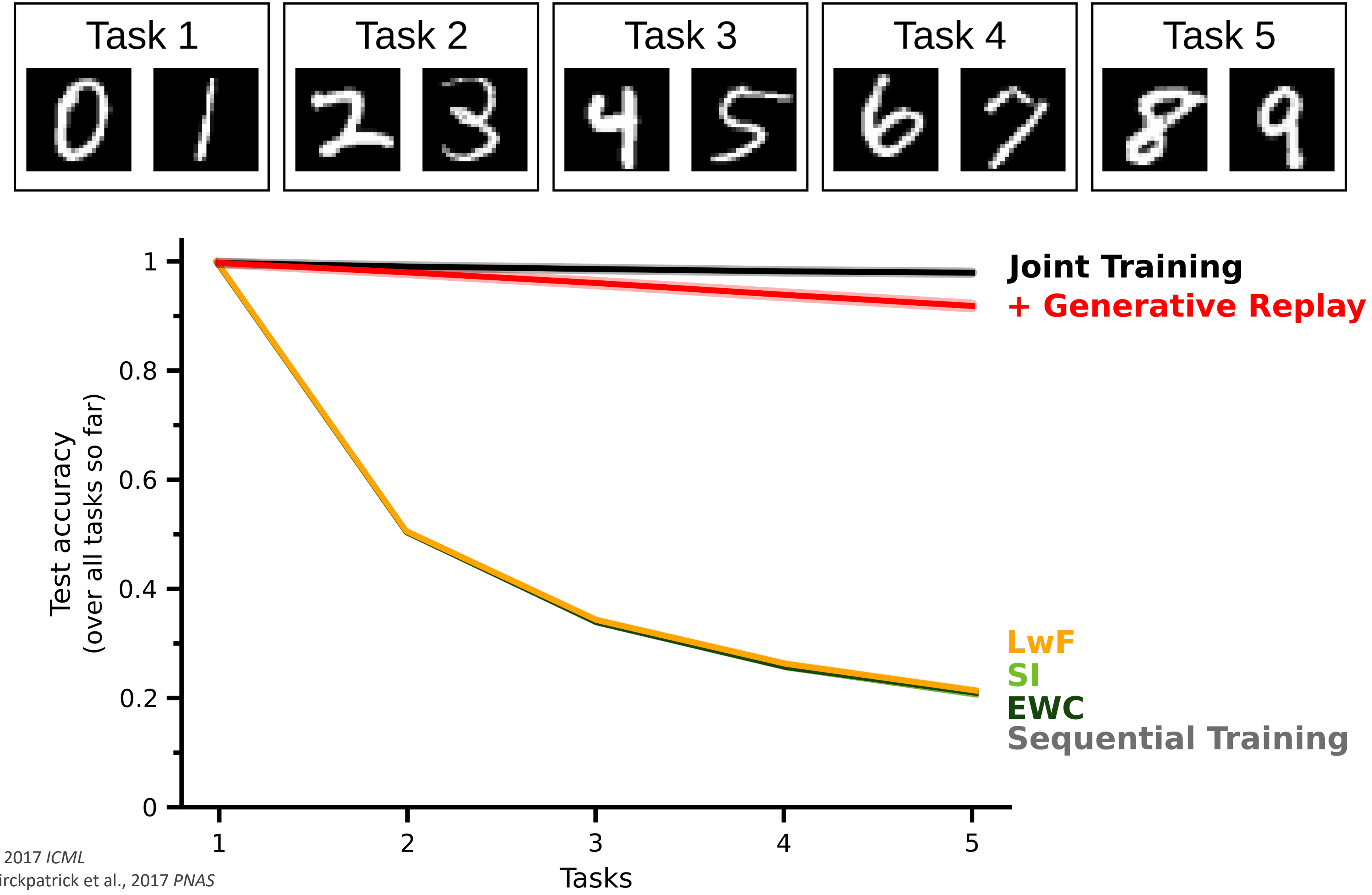
- Elastic Weight Consolidation (**EWC**) / Synaptic Intelligence (**SI**)
  - Estimate each parameter's importance for previously learned tasks'
  - Slow down learning for each parameter proportional to its estimated importance

- Learning without Forgetting (**LwF**)
  - Replay inputs from current task, labeled according to the predictions of the model trained on the previous tasks



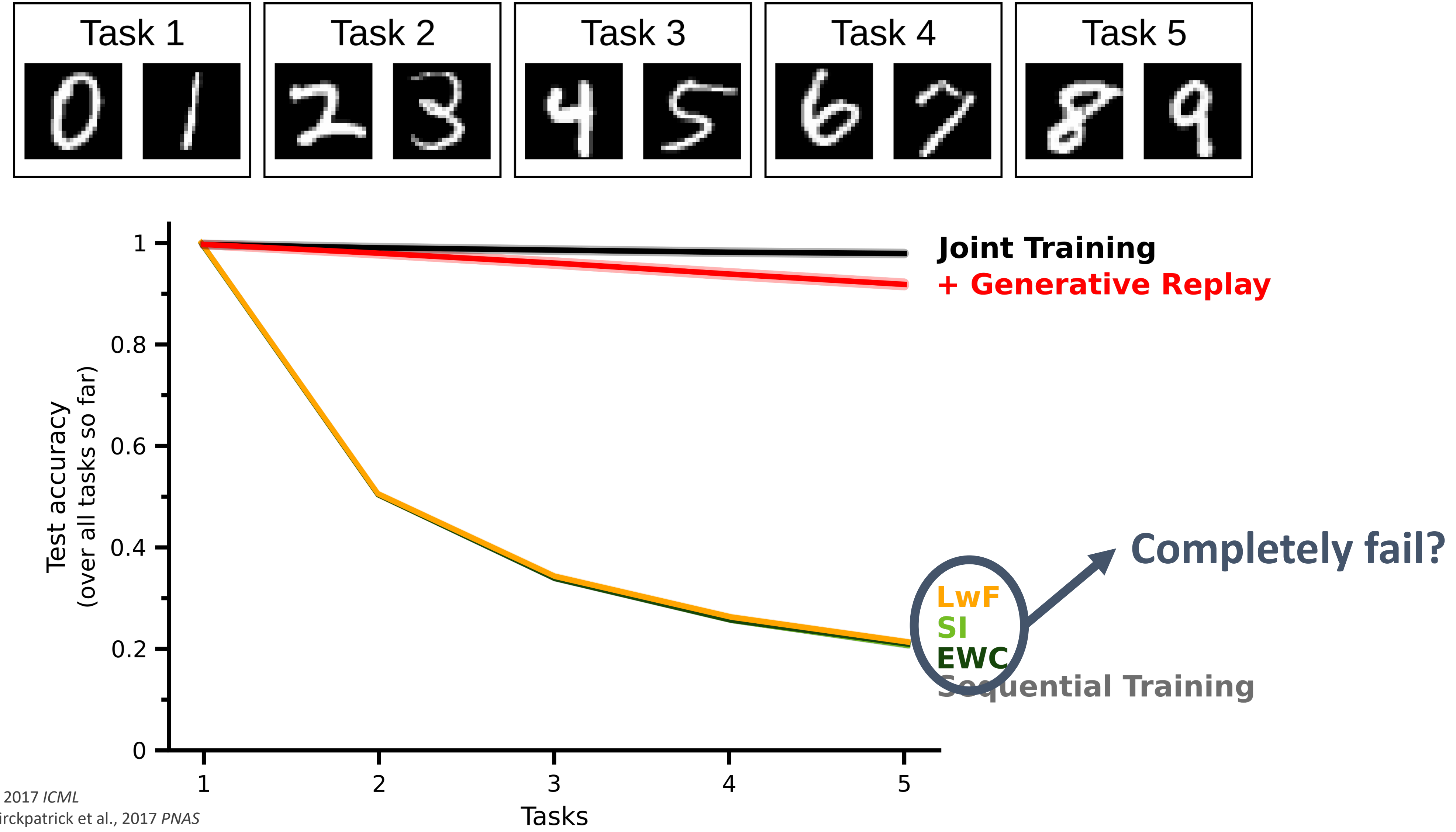
Synaptic Intelligence (SI): Zenke et al., 2017 *ICML*
Elastic Weight Consolidation (EWC): Kirckpatrick et al., 2017 *PNAS*
Learning without Forgetting (LwF)*:* Li & Hoiem, 2017 *IEEE T Pattern Anal*

Comparison with other methods



Synaptic Intelligence (SI): Zenke et al., 2017 *ICML*
Elastic Weight Consolidation (EWC): Kirckpatrick et al., 2017 *PNAS*
Learning without Forgetting (LwF): Li & Hoiem, 2017 *IEEE T Pattern Anal*

Comparison with other methods



Synaptic Intelligence (SI): Zenke et al., 2017 *ICML*
Elastic Weight Consolidation (EWC): Kirckpatrick et al., 2017 *PNAS*
Learning without Forgetting (LwF): Li & Hoiem, 2017 *IEEE T Pattern Anal*

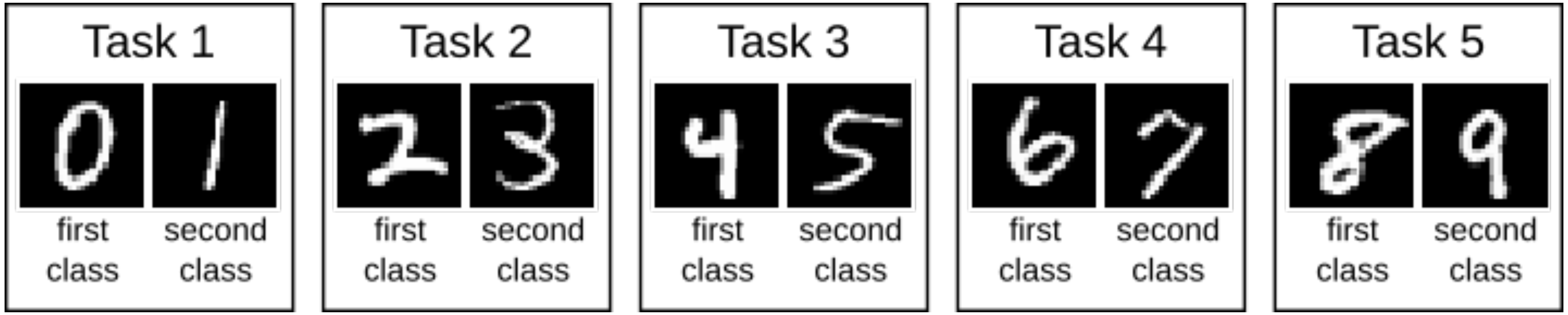
Important differences in evaluation protocols

- Is task identity provided?
- If it is not, does task identity need to be inferred?

Three continual learning scenarios

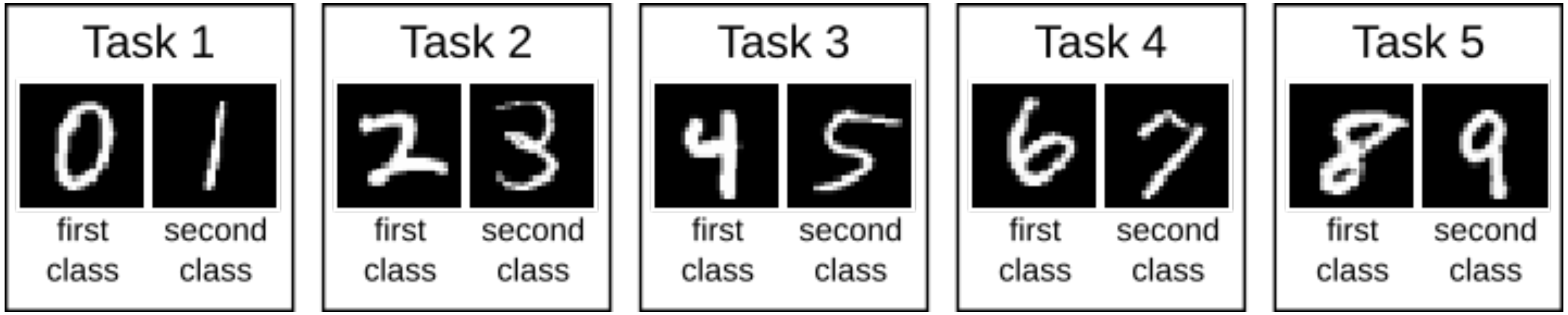
Scenario		*Required at test time*
Task-Incremental Learning	(Task-IL)	Solve tasks so far, task-ID provided
Domain-Incremental Learning	(Domain-IL)	Solve tasks so far, task-ID not provided
Class-Incremental Learning	(Class-IL)	Solve tasks so far *and* infer task-ID

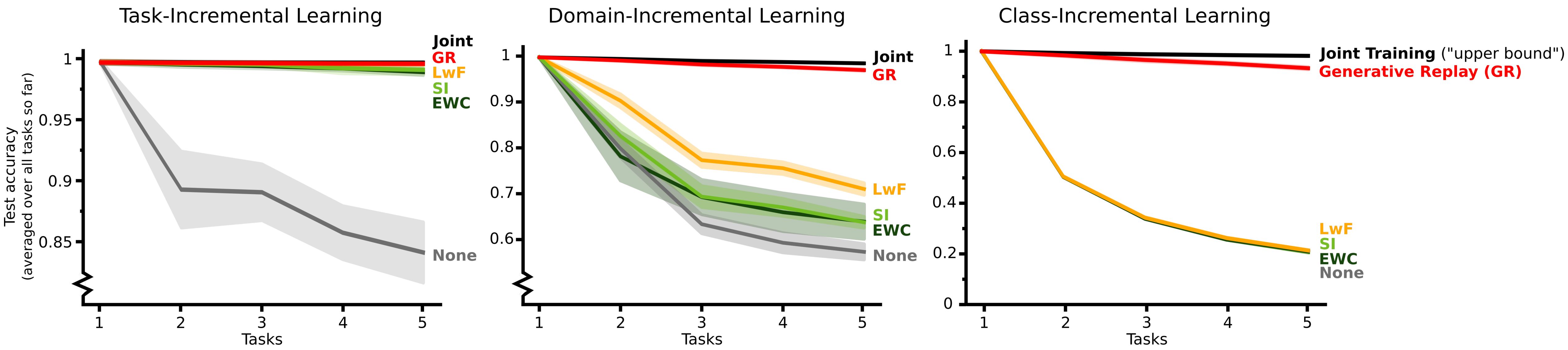
Three continual learning scenarios – *split MNIST*



Task-IL	With task given, is it the first or second class? (*e.g.*, '0' or '1')
Domain-IL	With task unknown, is it a first or second class? (*e.g.*, in ['0','2','4','6','8'] or in ['1','3','5','7','9'])
Class-IL	With task unknown, which digit is it? (*i.e.*, choice from '0' to '9')

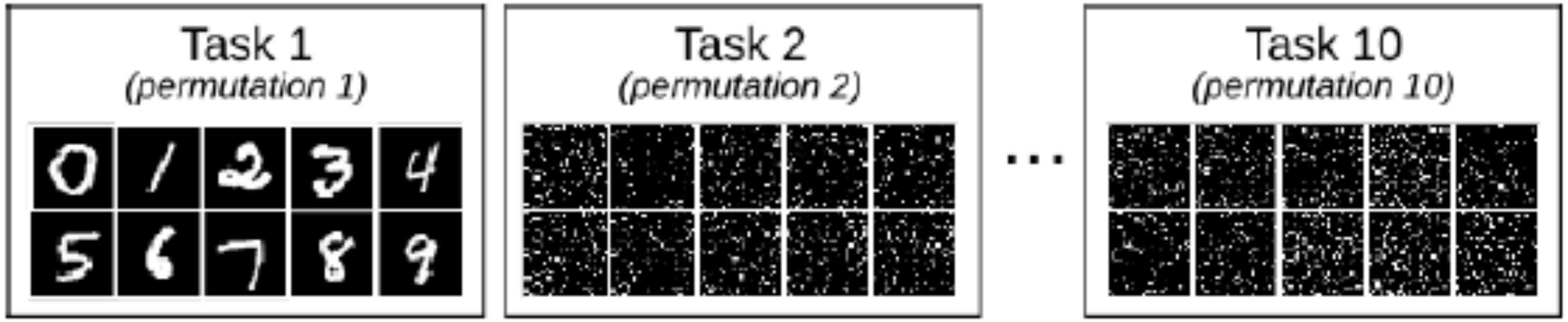
Three continual learning scenarios – *split MNIST*

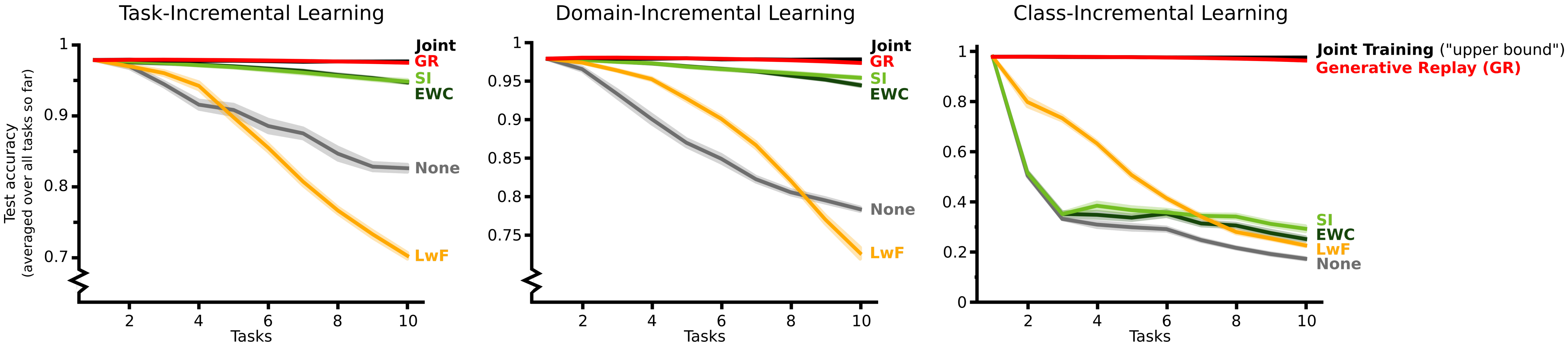




Code to replicate these experiments: https://github.com/GMvandeVen/continual-learning

Three continual learning scenarios – *permuted MNIST*





Code to replicate these experiments: https://github.com/GMvandeVen/continual-learning

Three continual learning scenarios & replay

INTERIM SUMMARY 1:

- In continual learning, a critical experimental design consideration is whether task identity is provided / must be inferred
- Generative replay works very well for MNIST-based problems
- Only replay-based methods seem to be capable of learning to distinguish classes that are never observed together

Further details: - van de Ven & Tolias (2018) Generative replay with feedback connections as a general strategy for continual learning. *arXiv:1809.10635*.
- van de Ven & Tolias (2019) Three scenarios for continual learning. *NeurIPS Continual Learning workshop*

Three continual learning scenarios & replay

INTERIM SUMMARY 1:

- In continual learning, a critical experimental design consideration is whether task identity is provided / must be inferred
- Generative replay works very well for MNIST-based problems
- Only replay-based methods seem to be capable of learning to distinguish classes that are never observed together

Further details: - van de Ven & Tolias (2018) Generative replay with feedback connections as a general strategy for continual learning. *arXiv:1809.10635*.
- van de Ven & Tolias (2019) Three scenarios for continual learning. *NeurIPS Continual Learning workshop*

But… (1) MNIST digits are relatively easy to generate

(2) constantly retraining on all previous tasks seems very inefficient

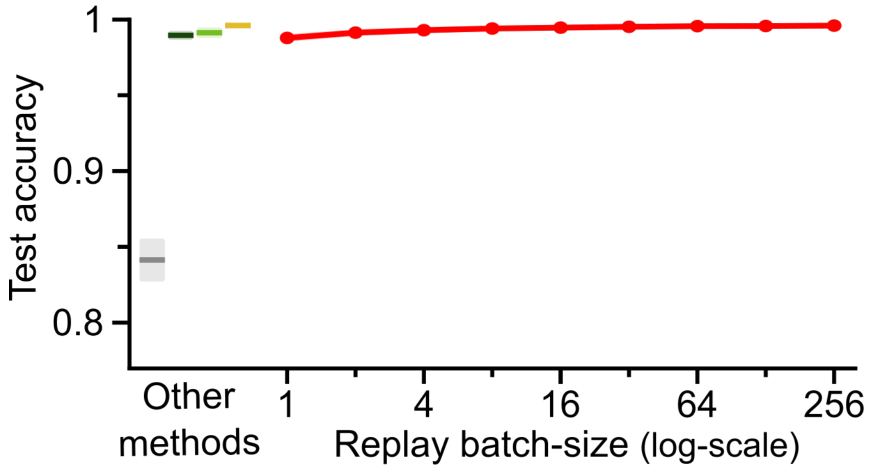
Efficiency: How much replay is needed?

- Previous tasks' datasets do not need to be replayed "fully"
- How far could the number of replayed sampled per batch be reduced?

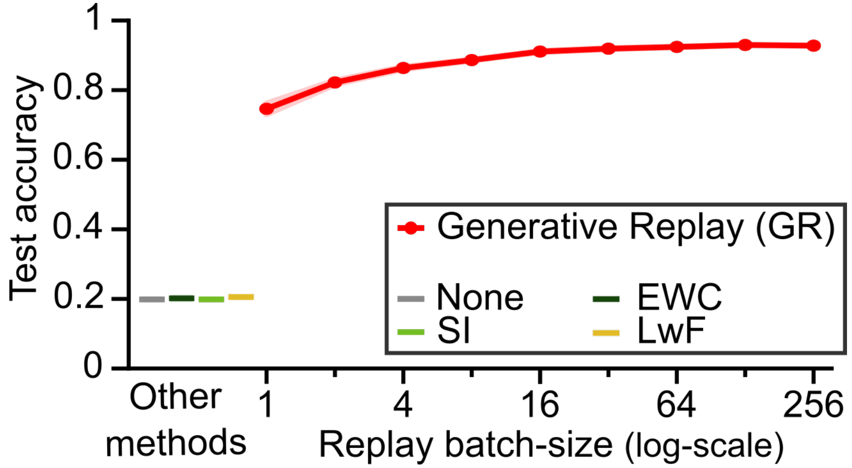
$$\mathcal{L}_{\text{total}} = \frac{1}{N_{\text{tasks so far}}}\mathcal{L}_{\text{current}} + \left(1 - \frac{1}{N_{\text{tasks so far}}}\right)\mathcal{L}_{\text{replay}}$$

$N_{\text{current-batch}} = 128$ $\qquad$ $N_{\text{replay-batch}} = ?$

Task-incremental learning



Class-incremental learning



→ ***Fully replaying previous tasks is not needed, replaying only a few examples could suffice***

(these experiments are on Split MNIST)

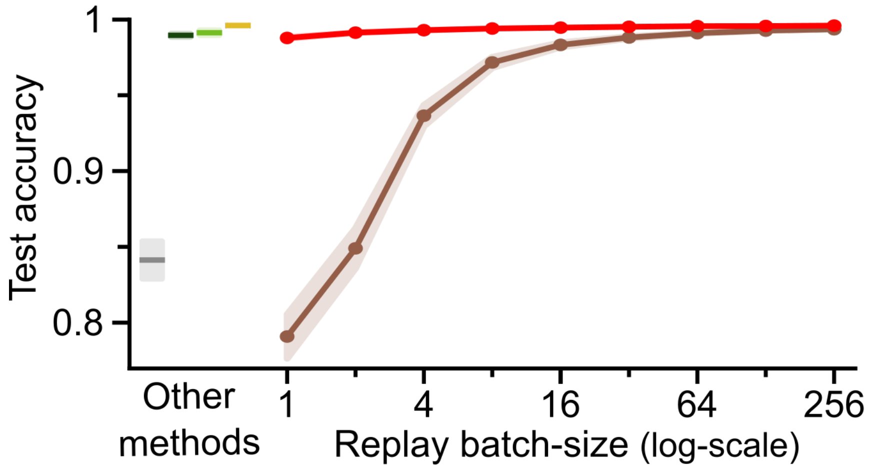
Efficiency: How much replay is needed?

- Previous tasks' datasets do not need to be replayed "fully"
- How far could the number of replayed sampled per batch be reduced?

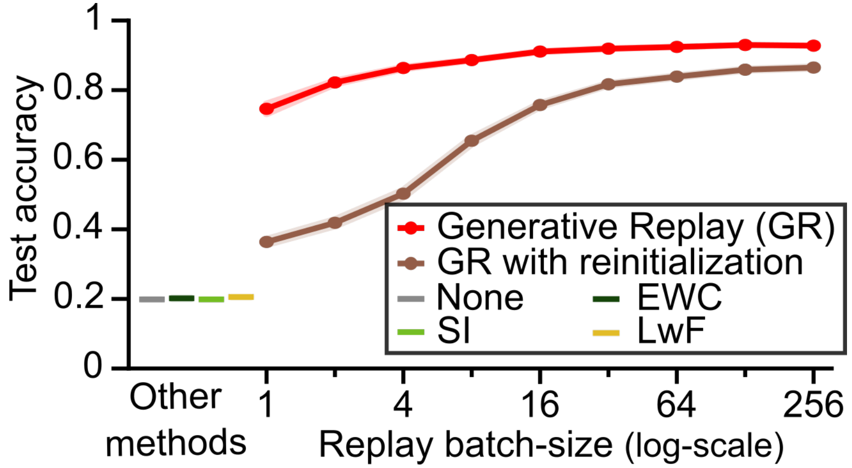
$$\mathcal{L}_{\text{total}} = \frac{1}{N_{\text{tasks so far}}}\mathcal{L}_{\text{current}} + \left(1 - \frac{1}{N_{\text{tasks so far}}}\right)\mathcal{L}_{\text{replay}}$$

$N_{\text{current-batch}} = 128$ $\qquad$ $N_{\text{replay-batch}} = ?$

Task-incremental learning



Class-incremental learning

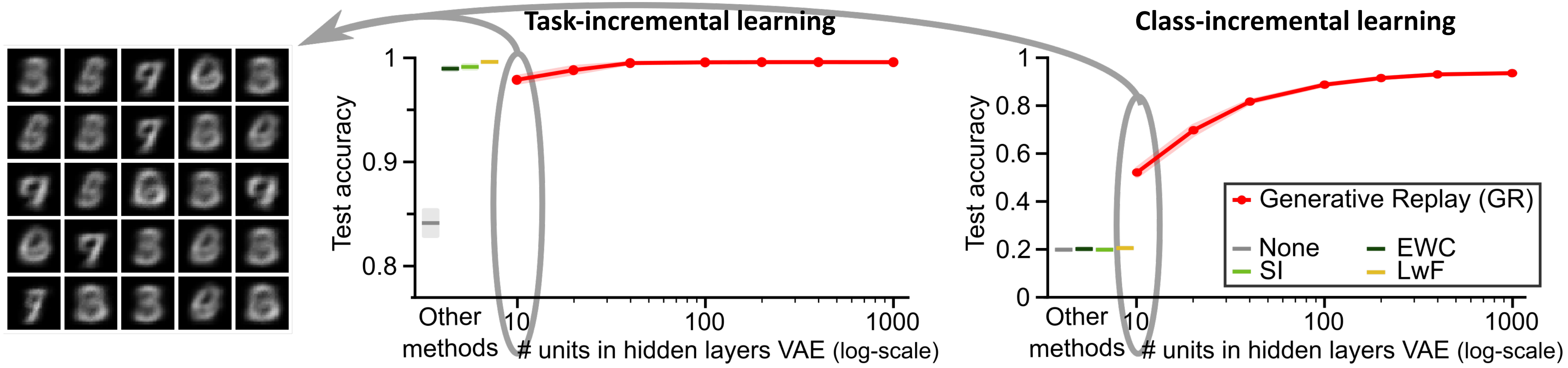


→ ***Fully replaying previous tasks is not needed, replaying only a few examples could suffice***

(these experiments are on Split MNIST)

Robustness: How good does the replay need to be?

- Generating MNIST-digits is relatively easy; could this scale to more complicated inputs?
- Replaying inputs from current task (i.e., LwF) already helps under certain conditions
- Performance of generative replay is evaluated as function of the size of the generator

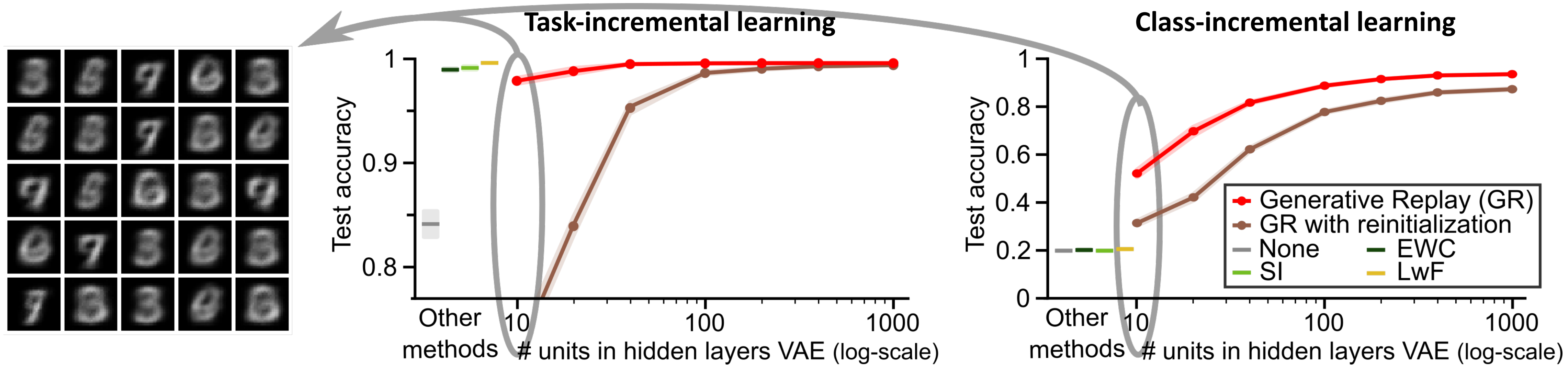


→ *A perfect memory (storing everything) is not needed, a low-quality generative model could suffice*

(these experiments are on Split MNIST)

Robustness: How good does the replay need to be?

- Generating MNIST-digits is relatively easy; could this scale to more complicated inputs?
- Replaying inputs from current task (i.e., LwF) already helps under certain conditions
- Performance of generative replay is evaluated as function of the size of the generator

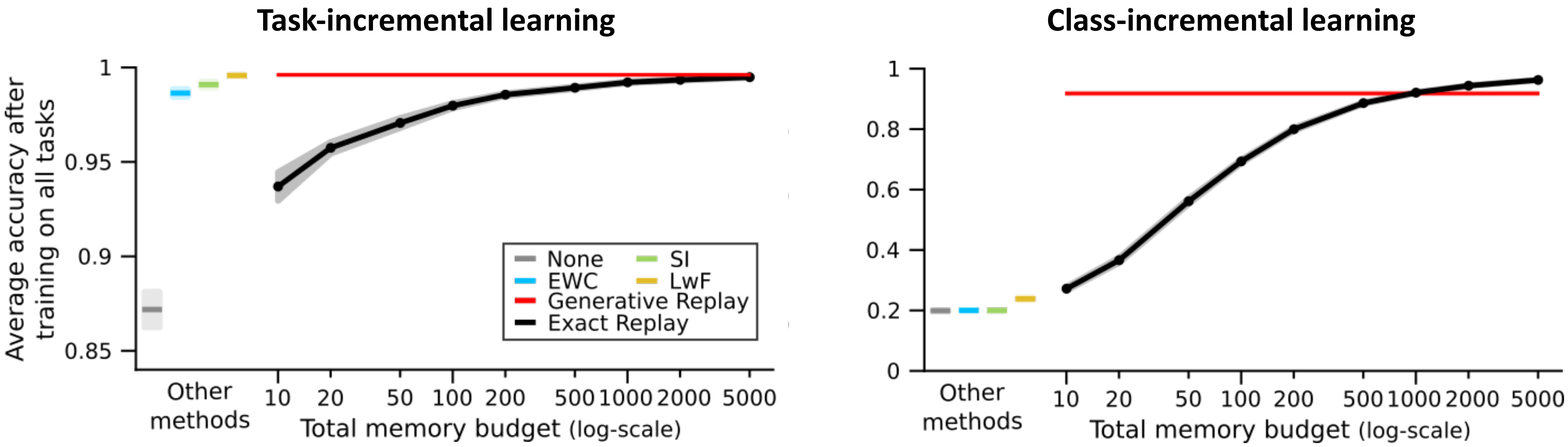


→ *A perfect memory (storing everything) is not needed, a low-quality generative model could suffice*

(these experiments are on Split MNIST)

Generative replay *vs.* replaying stored data

- Performance of generative replay is compared with that of exact replay as a function of the total number of datapoints allowed to be stored in memory



→ ***Variety of what is replayed does seem to be important***

(these experiments are on Split MNIST)

Robustness and efficiency of replay

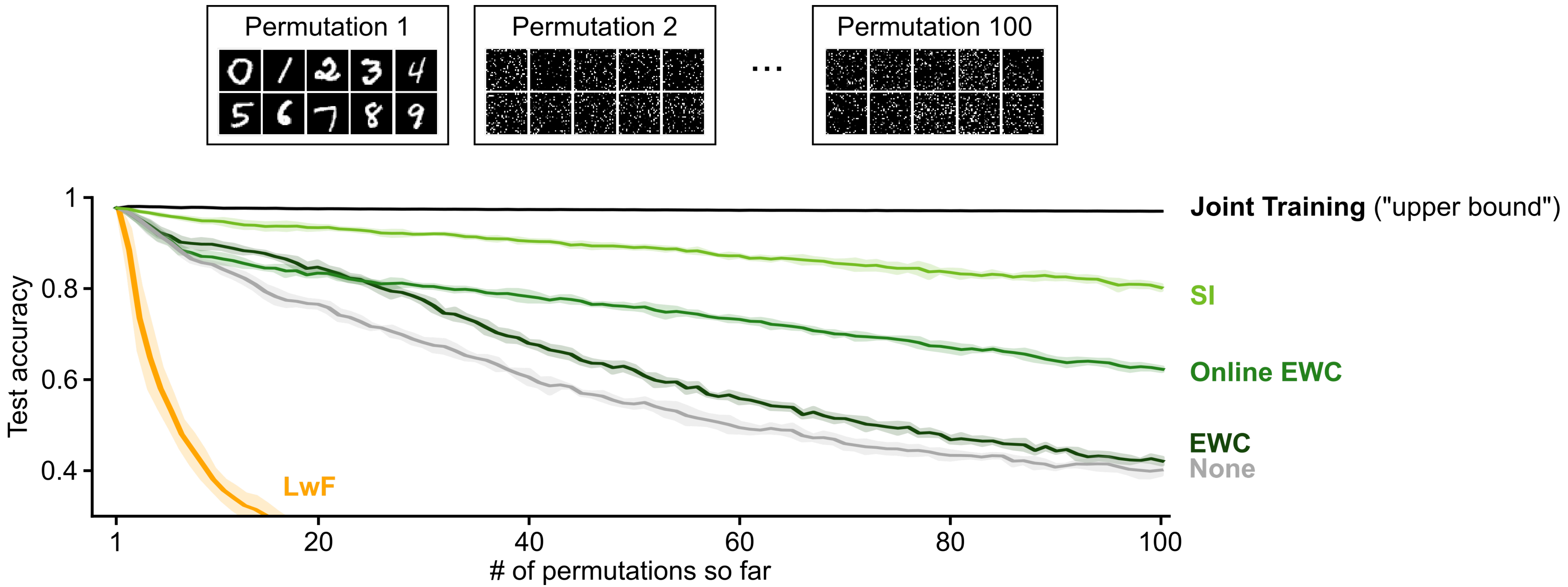
INTERIM SUMMARY 2:

- Even replaying a few or poor-quality samples can substantially boost lifelong learning performance
- Although the number of replayed samples can be relatively low, variety of what is replayed does seem to be important

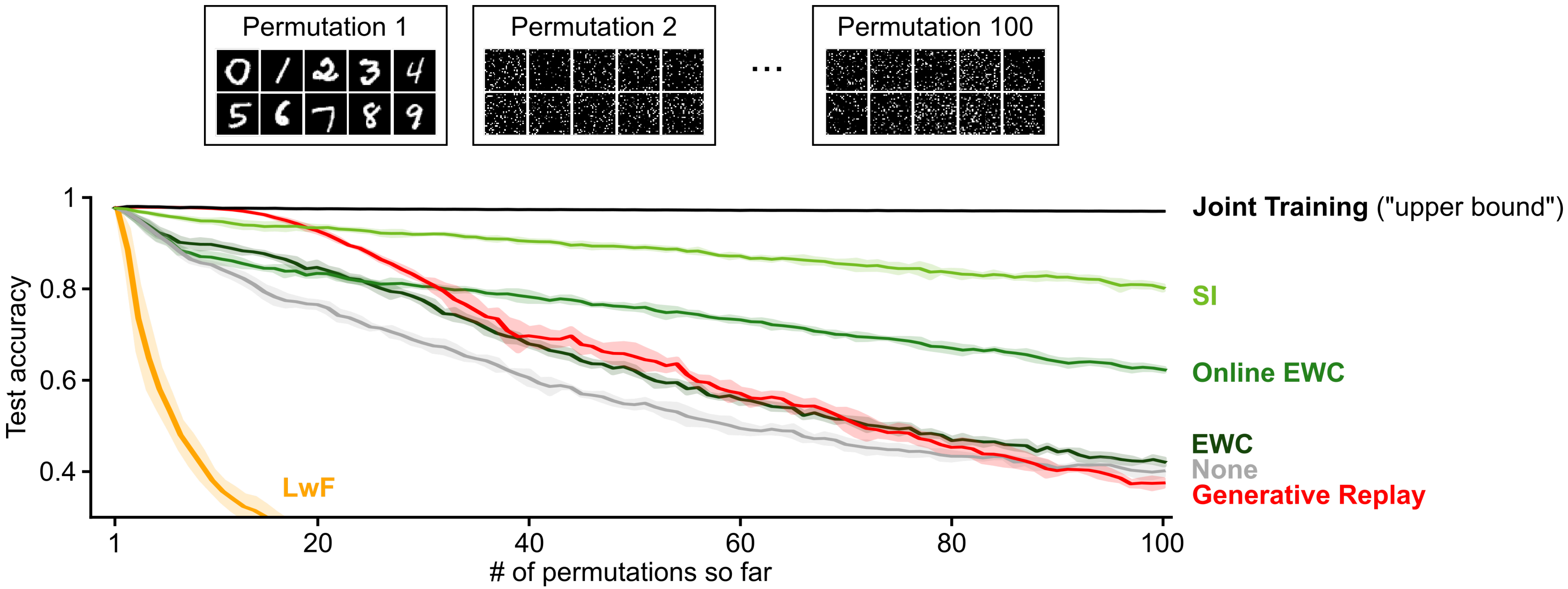
Next step:

→ Scale up experiments to problems with many tasks or more complicated inputs

What about scaling up to many tasks?

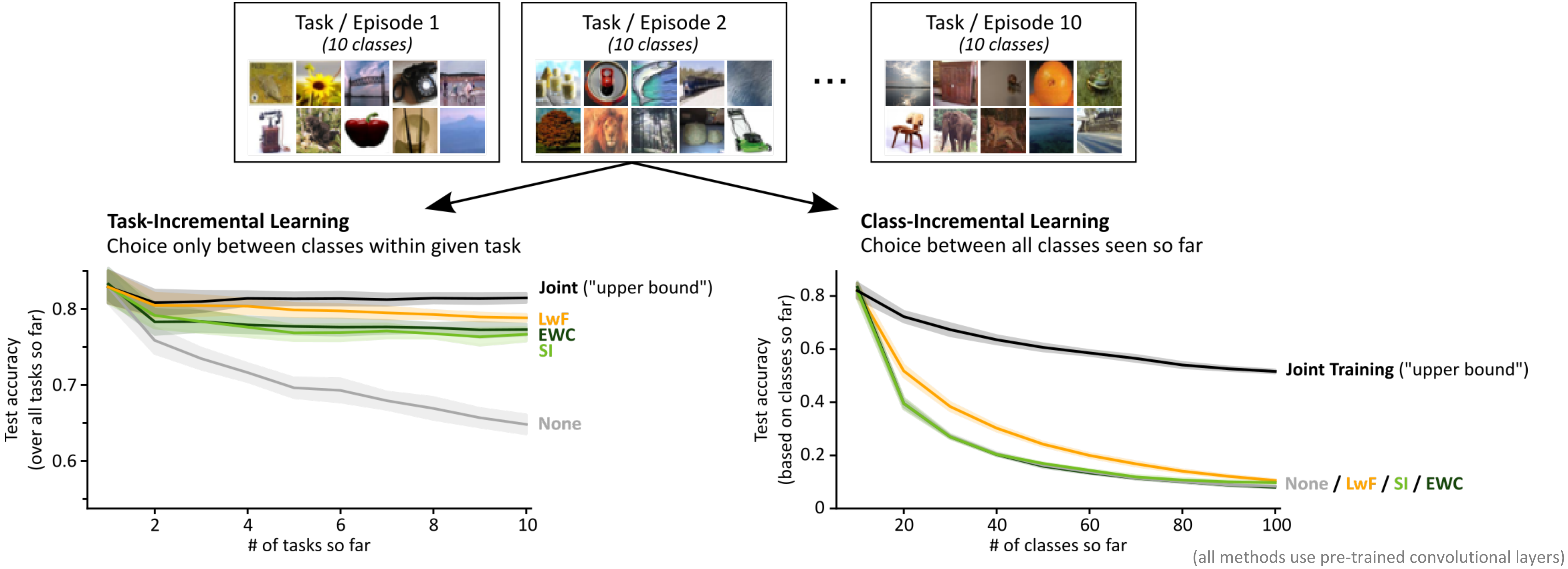


What about scaling up to many tasks?

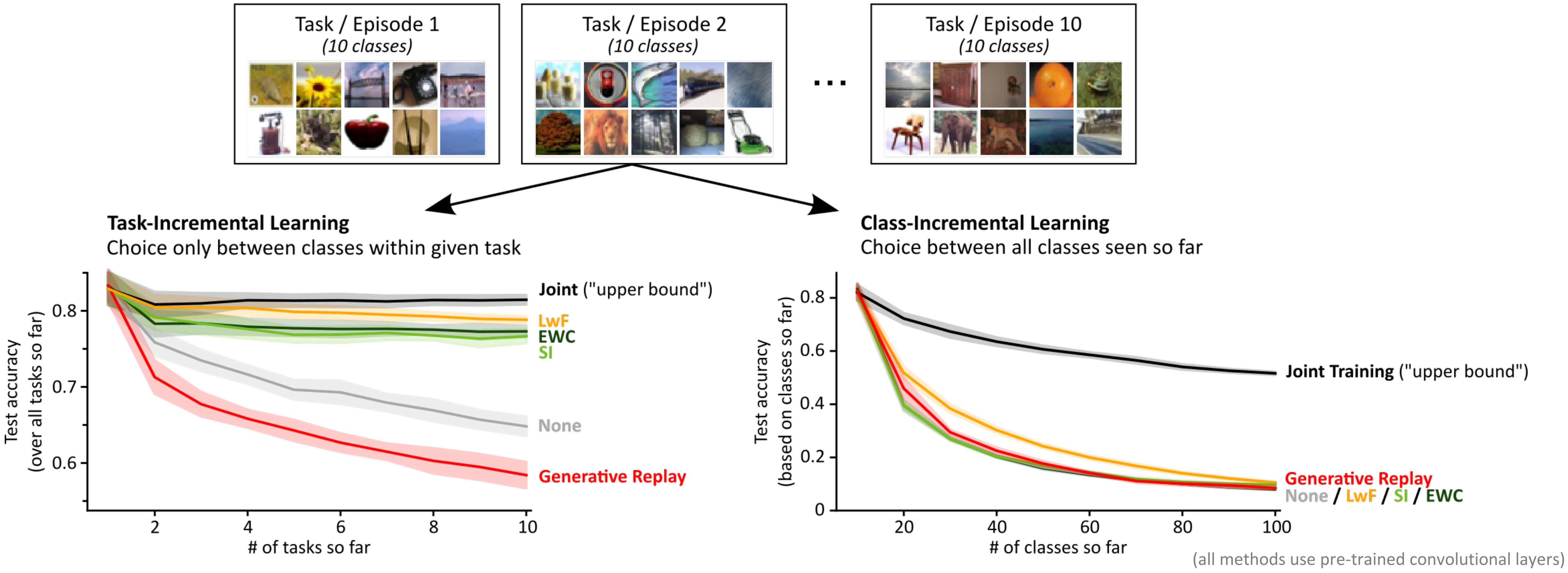


Standard versions of generative replay break down on problems with many tasks

What about scaling up to more complex inputs?



What about scaling up to more complex inputs?



Standard versions of generative replay break down on problems with more complex inputs (e.g., natural images) [see also Lesort et al., 2019 *IJCNN*; Aljundi et al., 2019 *NeurIPS*]

What about scaling up?

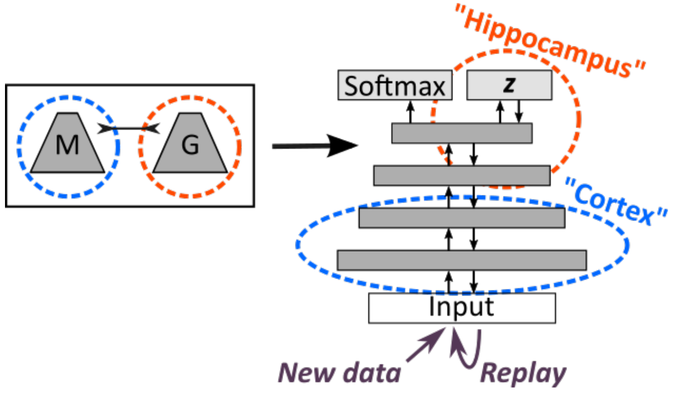
INTERIM SUMMARY 3:

- Standard versions of generative replay break down when either many tasks must be learned or when the inputs become more complex

Possible solutions...

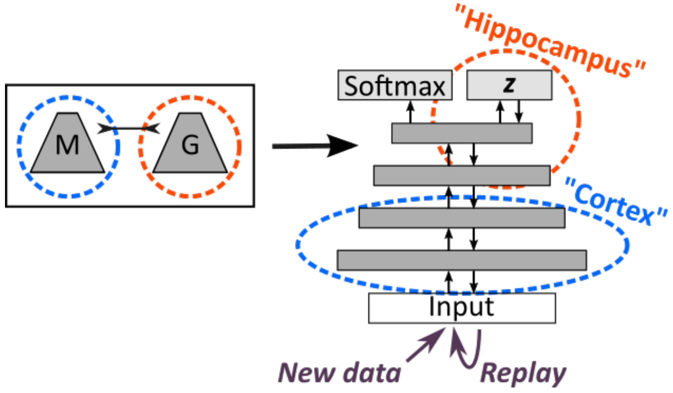
- Use recent progress in deep generative modelling to improve quality of generator?
  - → Incrementally training state-of-the-art generative models is very challenging
  - → Computationally very costly
- Model generative replay after the brain

Brain-inspired modifications to Generative Replay



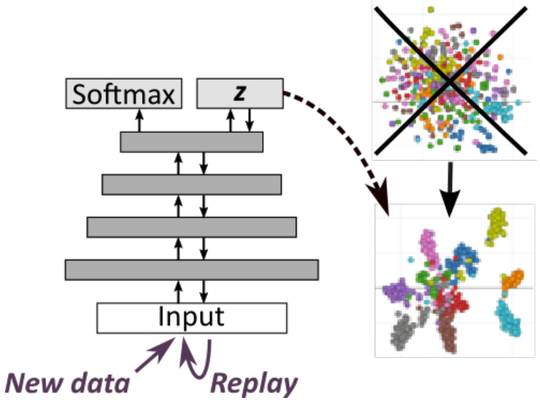
- ***Replay-through-Feedback:*** Merge generator into main model; replay is now generated by the feedback / backward connections

Brain-inspired modifications to Generative Replay

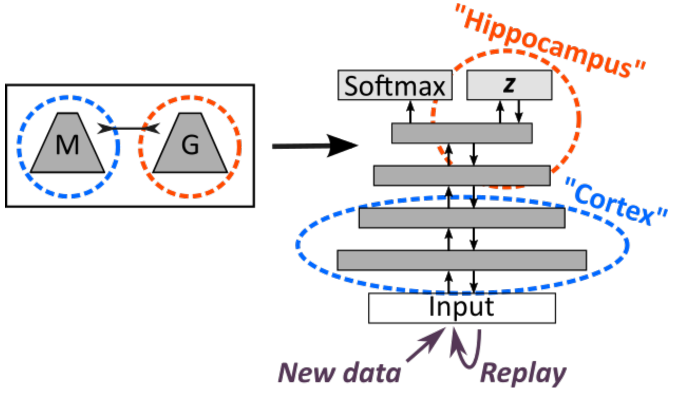


- ***Replay-through-Feedback:*** Merge generator into main model; replay is now generated by the feedback / backward connections

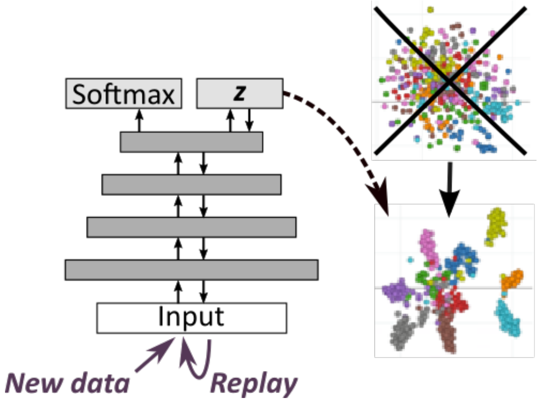
- ***Conditional Replay:*** Enable model to generate specific classes, by replacing the standard normal prior by a Gaussian mixture with a separate mode for each class



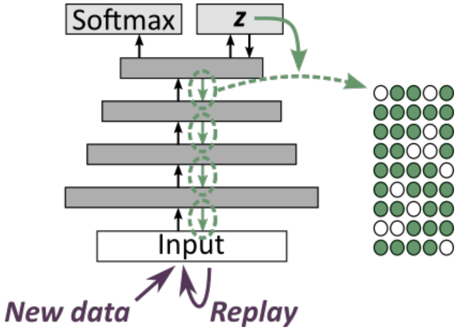
Brain-inspired modifications to Generative Replay



- ***Replay-through-Feedback:*** Merge generator into main model; replay is now generated by the feedback / backward connections

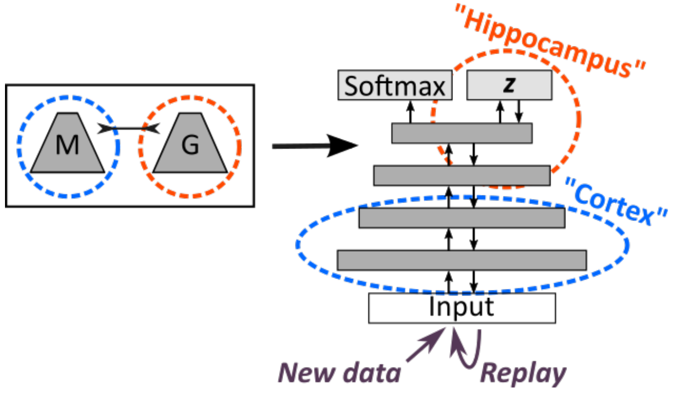


- ***Conditional Replay:*** Enable model to generate specific classes, by replacing the standard normal prior by a Gaussian mixture with a separate mode for each class

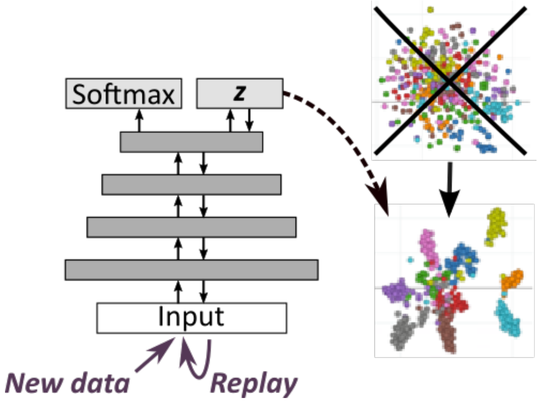


- ***Gating based on Internal Context:*** For each class, inhibit (or gate) a different subset of neurons during the generative backward pass

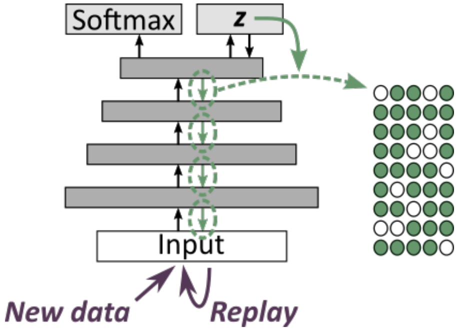
Brain-inspired modifications to Generative Replay



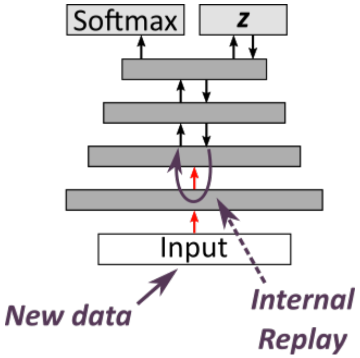
- ***Replay-through-Feedback:*** Merge generator into main model; replay is now generated by the feedback / backward connections



- ***Conditional Replay:*** Enable model to generate specific classes, by replacing the standard normal prior by a Gaussian mixture with a separate mode for each class

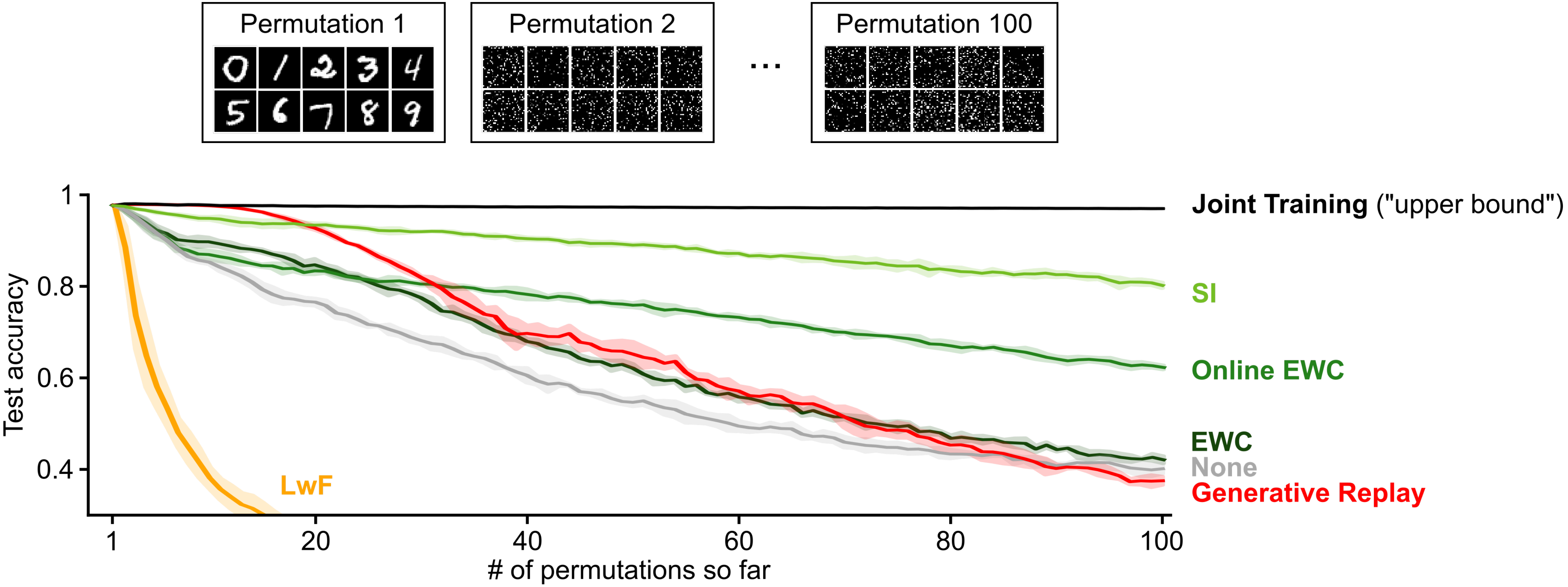


- ***Gating based on Internal Context:*** For each class, inhibit (or gate) a different subset of neurons during the generative backward pass

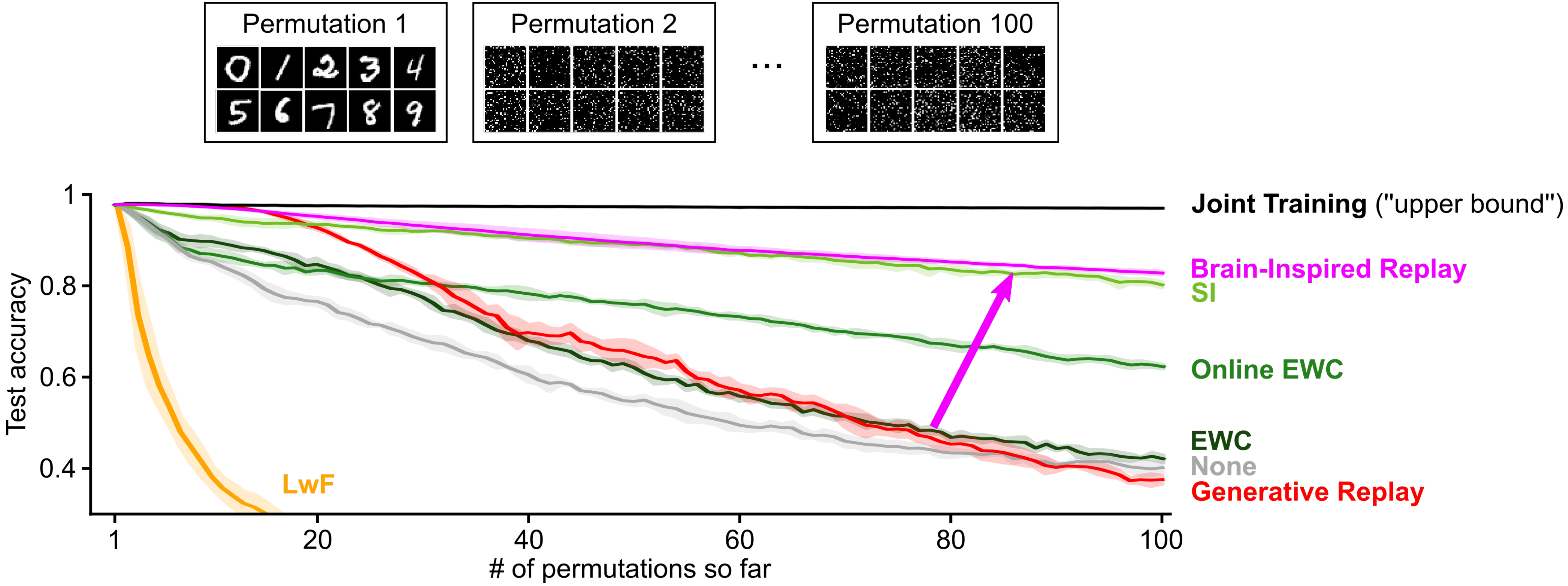


- ***Internal Replay:*** Replay internal or hidden representations, instead of at the input level (e.g., pixel level)

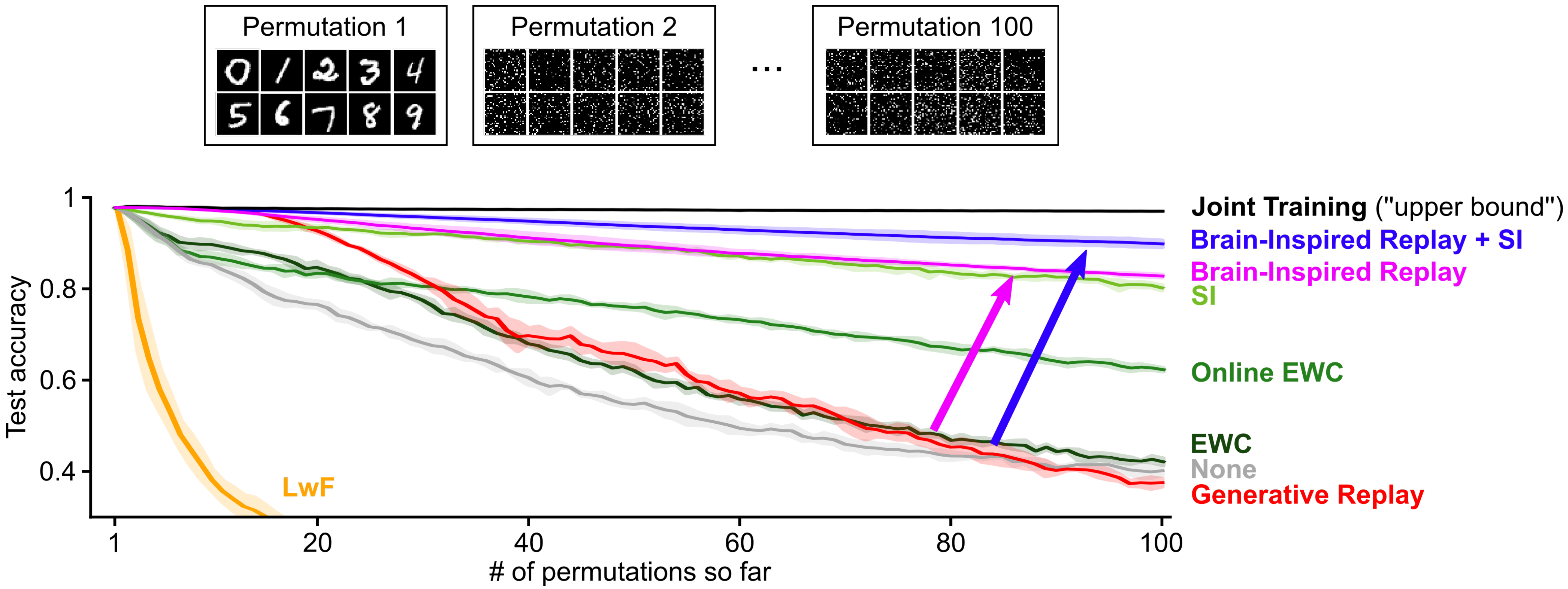
Brain-inspired replay with many tasks



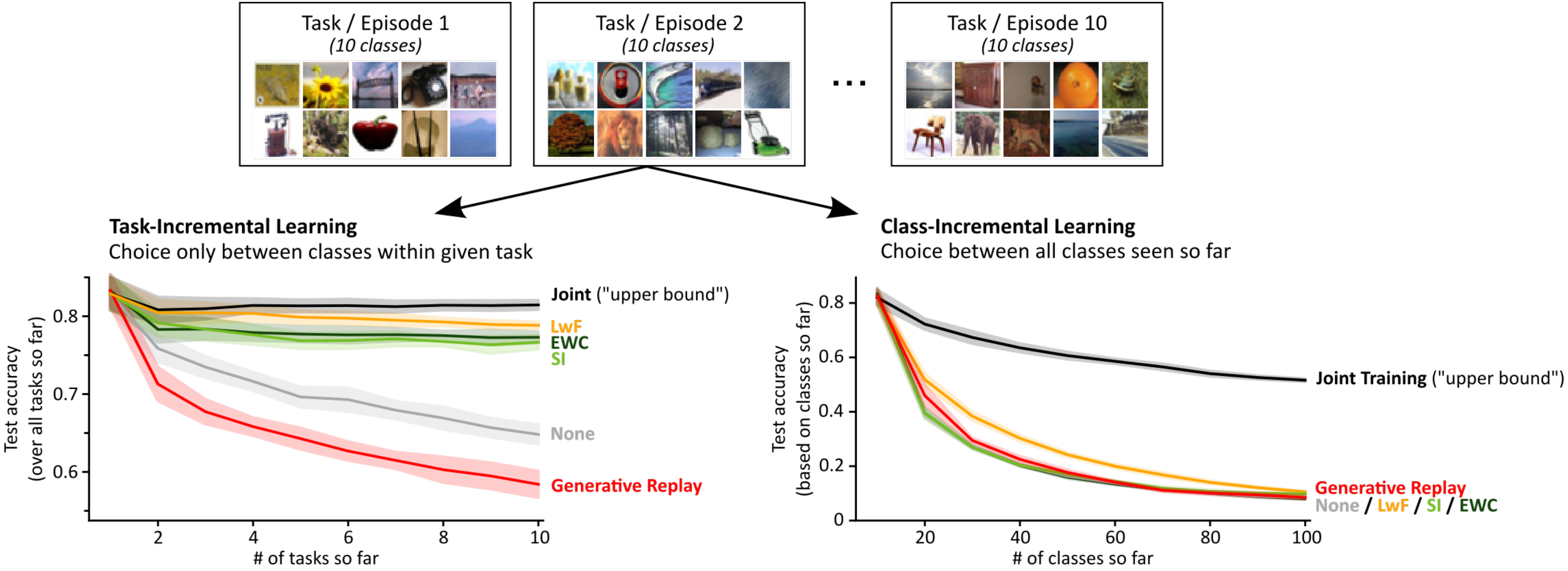
Brain-inspired replay with many tasks



Brain-inspired replay with many tasks

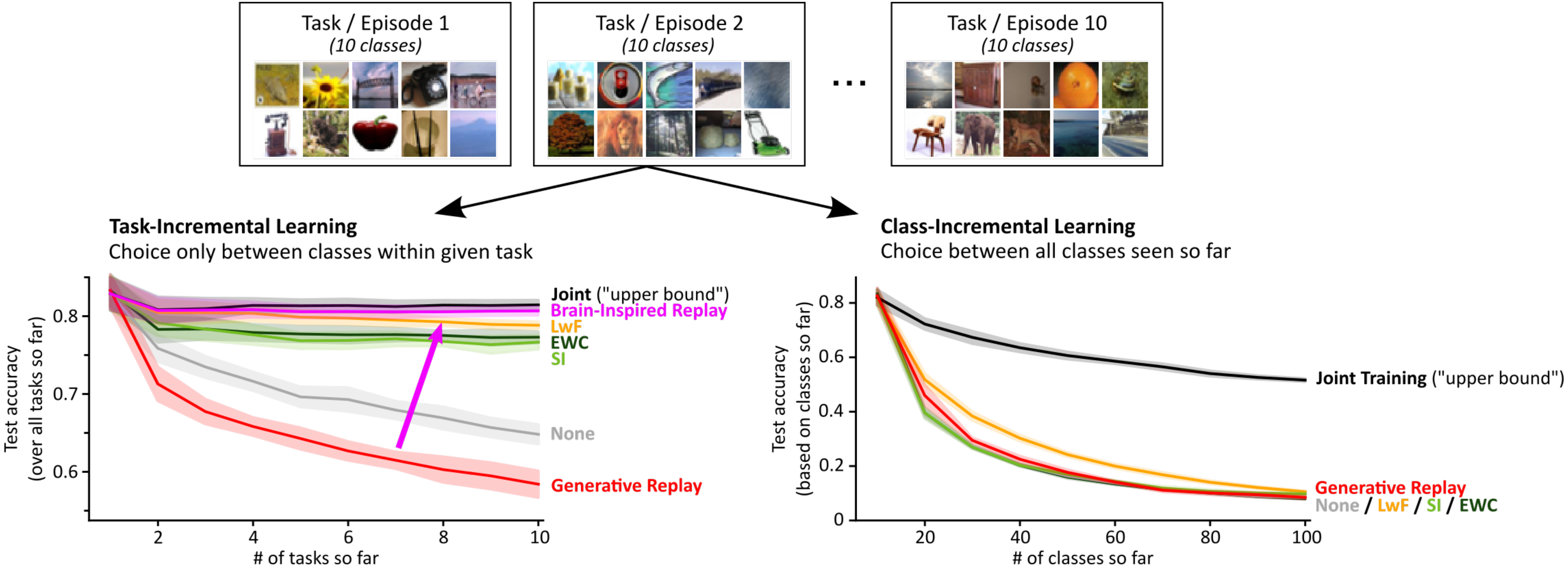


Brain-inspired replay on natural images



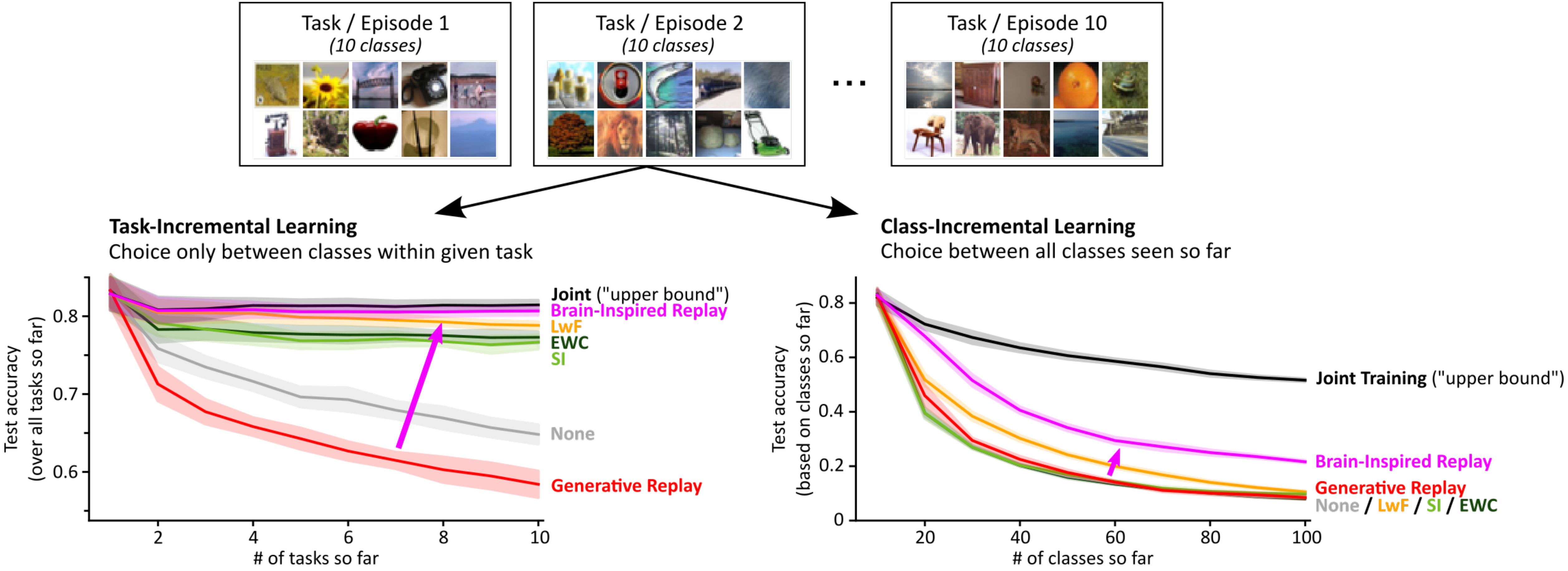
(all methods use pre-trained convolutional layers)

Brain-inspired replay on natural images



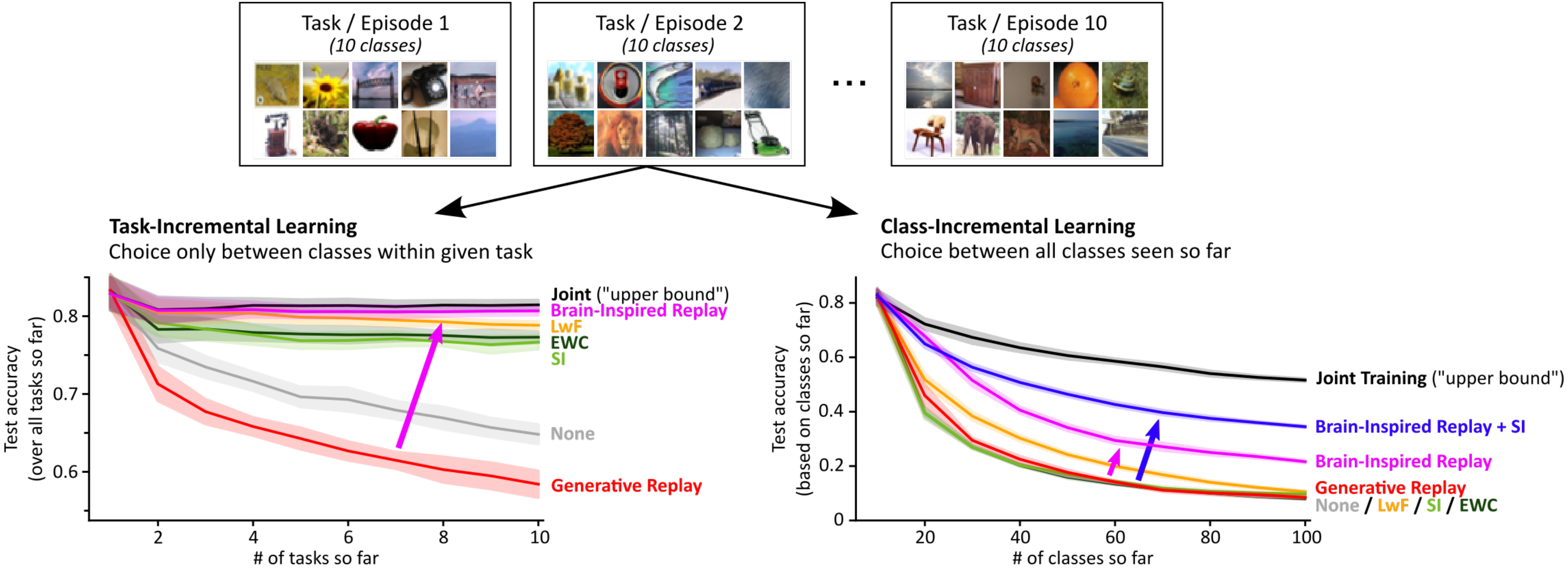
(all methods use pre-trained convolutional layers)

Brain-inspired replay on natural images



(all methods use pre-trained convolutional layers)

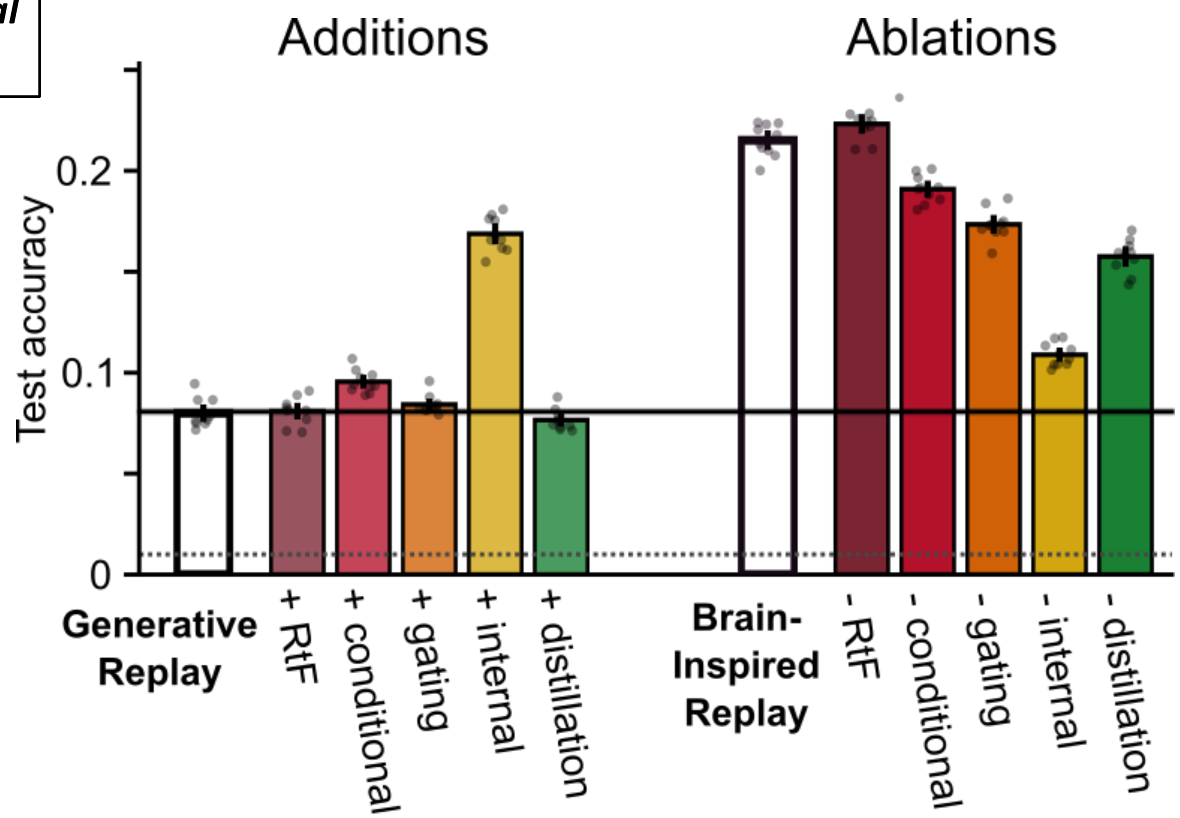
Brain-inspired replay on natural images



(all methods use pre-trained convolutional layers)

Lesion experiments

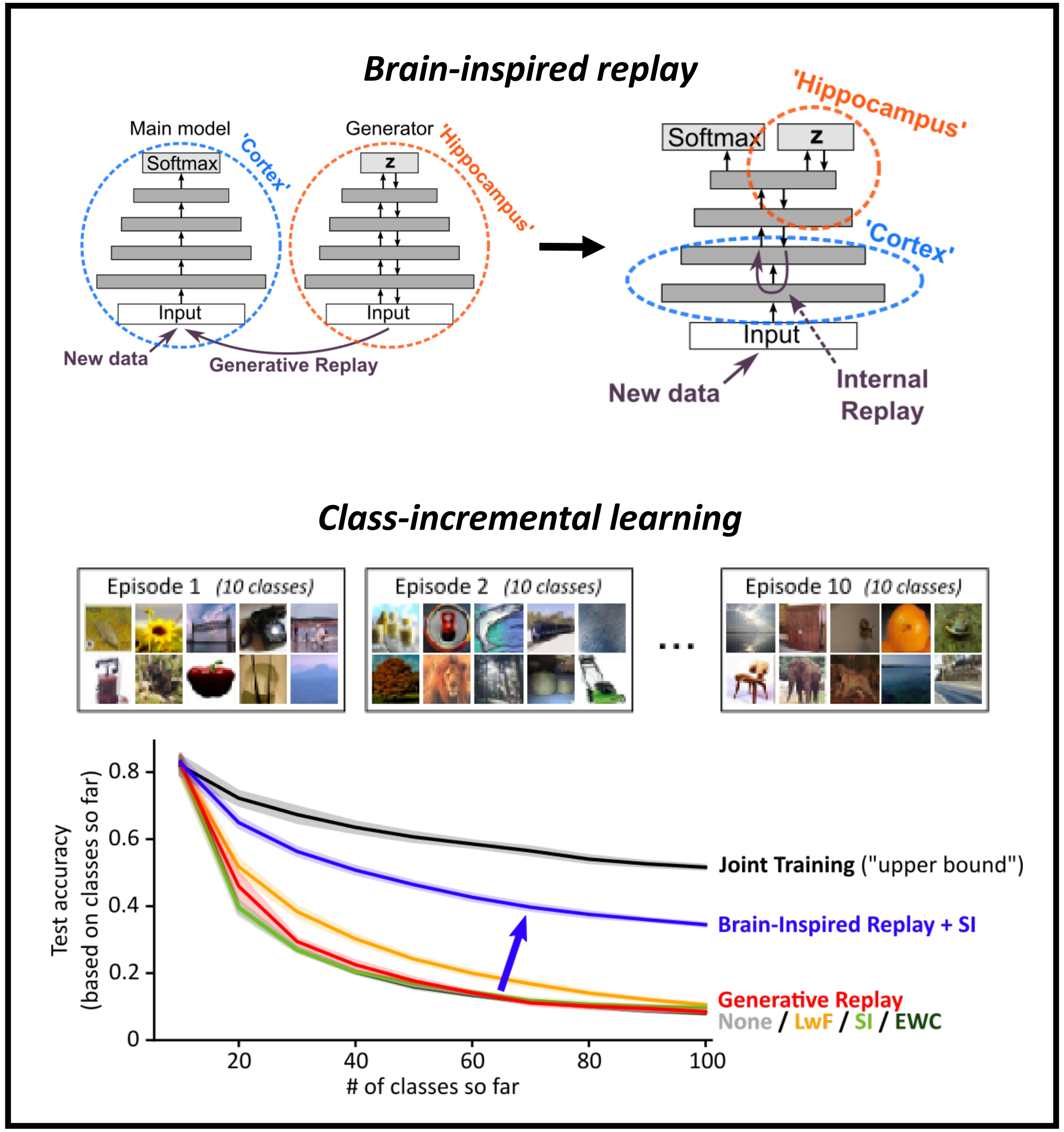
Class-incremental CIFAR-100:



- Internal replay is most influential, but all modifications contribute

Summary

- Replay is especially important for class-incremental learning (i.e., learning to distinguish between classes that are not observed together)
- Even replaying a few or poor-quality samples can substantially boost continual learning performance
- Scaling generative replay up to problems with many tasks or more complicated inputs is nevertheless not straight-forward
- Modelling generative replay after the brain can substantially increase performance while lowering computational costs
- Our brain-inspired replay method replays internal or hidden representations that are generated by the network's own, context-modulated feedback connections



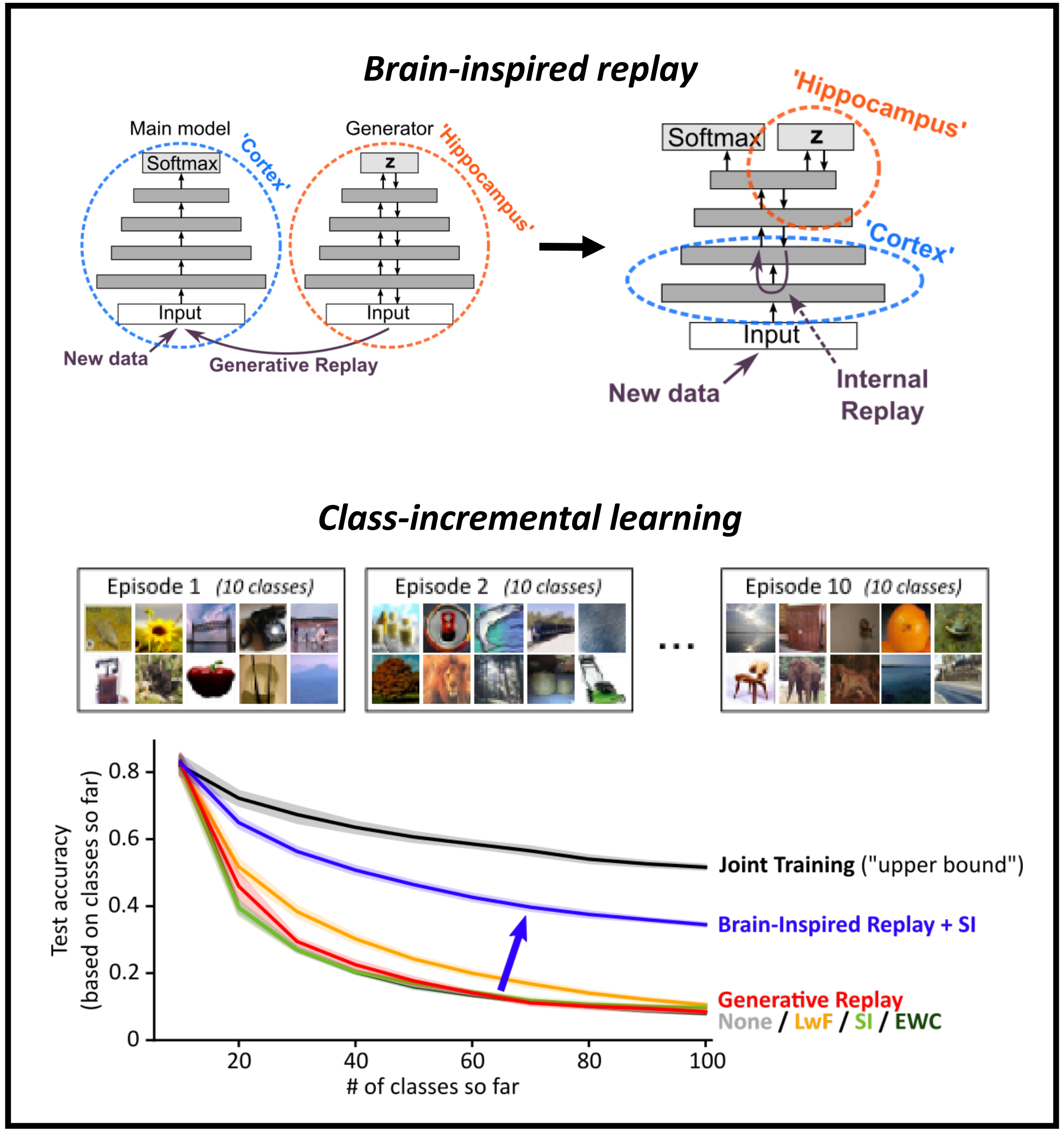
For full details:

van de Ven GM, Siegelmann HT, Tolias AS (2020) Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, **11**: 4069.

Code: https://github.com/GMvandeVen/brain-inspired-replay

Acknowledgements

We thank Mengye Ren, Zhe Li and Máté Lengyel for comments on various parts of this work, and Johannes Oswald and Zhengwen Zeng for useful suggestions. This research project 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 the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. Disclaimer: The views and conclusions contained in this presentation were those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, IARPA, DoI/IBC, or the U.S. Government.



References

- Aljundi R, Lin M, Goujaud B, Bengio Y (2019) Gradient based sample selection for online continual learning. *NeurIPS:* 11816-11825.
- Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2017) Overcoming catastrophic forgetting in neural networks. *PNAS* **114**: 3521-3526.
- Lesort T, Caselles-Dupré H, Garcia-Ortiz M, Stoian A, Filliat D (2019). Generative models from the perspective of continual learning. *IJCNN:* 1-8.
- Li Z, Hoiem D (2017) Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40***:* 2935-2947.
- Liu K, Sibille J, Dragoi G (2018) Generative predictive codes by multiplexed hippocampal neuronal tuplets. *Neuron* **99**: 1329-1341.
- Liu Y, Dolan RJ, Kurth-Nelson Z, Behrens TE (2019). Human replay spontaneously reorganizes experience. *Cell* **178**: 640-652.
- Masse NY, Grant GD, Freedman DJ (2018) Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *PNAS* **115**: E10467-E10475.
- McClelland JL, McNaughton BL, O'Reilly RC (1995) Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review* **102**: 419-457.
- McCloskey M, Cohen NJ (1989) Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation* **24**: 109-165.
- Rasch B, Born J (2007) Maintaining memories by reactivation. *Current Opinion in Neurobiology* **17**: 698–703.
- Ratcliff R (1990) Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review* **97**: 285-308.
- Shin H, Lee JK, Kim J, Kim J (2017) Continual learning with deep generative replay. *NeurIPS*: 2994-3003.
- van de Ven GM, Trouche S, McNamara CG, Allen K, Dupret D (2016) Hippocampal offline reactivation consolidates recently formed cell assembly patterns during sharp wave-ripples. *Neuron* **92**: 968-974.
- van de Ven GM, Siegelmann HT, Tolias AS (2020) Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications* **11**: 4069.
- van de Ven GM, Tolias AS (2019) Three scenarios for continual learning. *ArXiv preprint*: 1904.07734.
- van de Ven GM, Tolias AS (2018) Generative replay with feedback connections as a general strategy for continual learning. *ArXiv preprint*: 1809.10635.
- Wilson MA, McNaughton BL (1994) Reactivation of hippocampal ensemble memories during sleep. *Science* **265**: 676-679.
- Zenke F, Poole B, Ganguli S (2017) Continual learning through synaptic intelligence. *ICML*: 3987-3995.