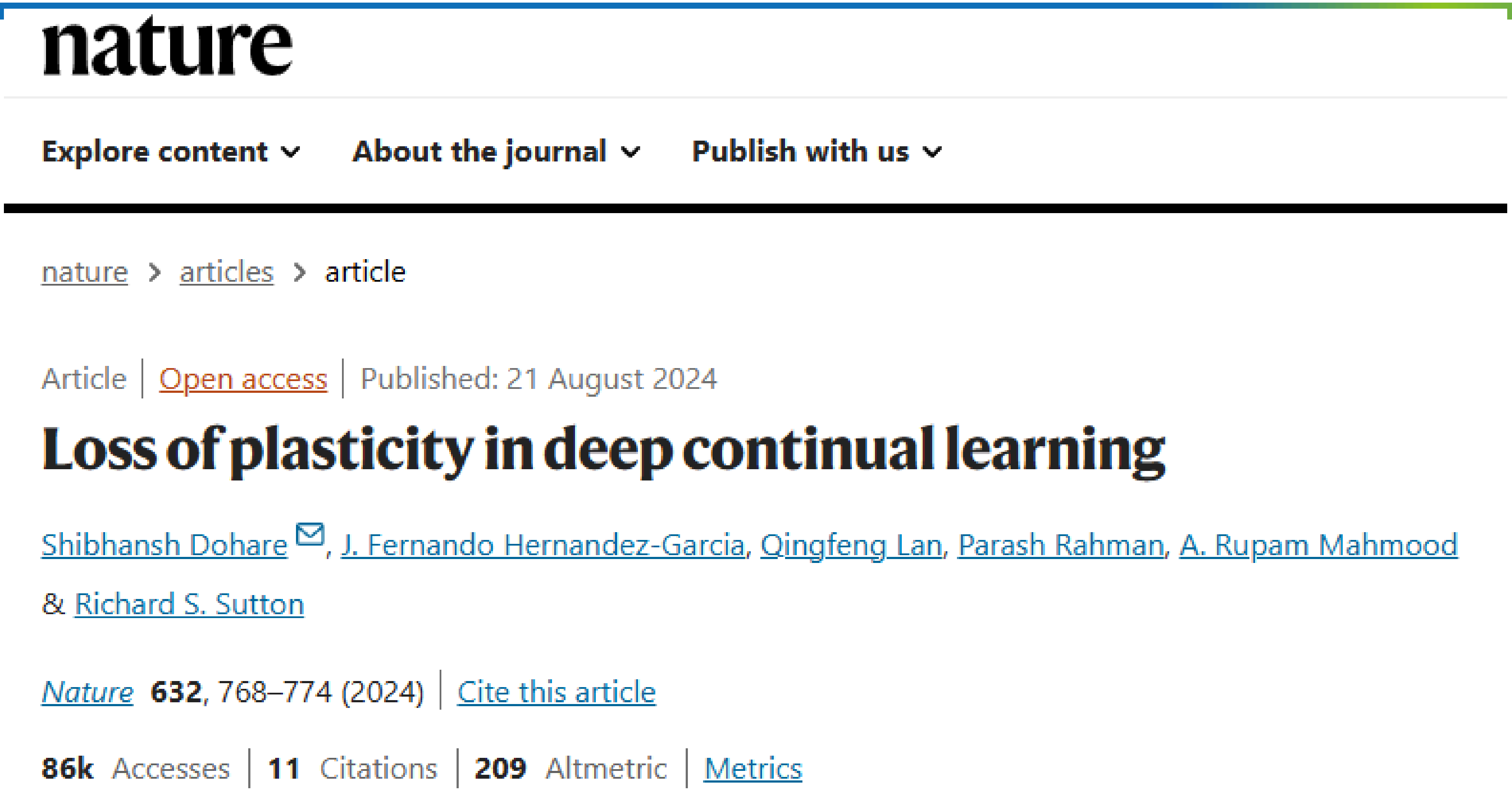
# Loss of Plasticity in Deep Learning

2025-03-28

KAIST Siit
Statistical Inference and
Information Theory Laboratory



Article | Open access | Published: 21 August 2024

# Loss of plasticity in deep continual learning

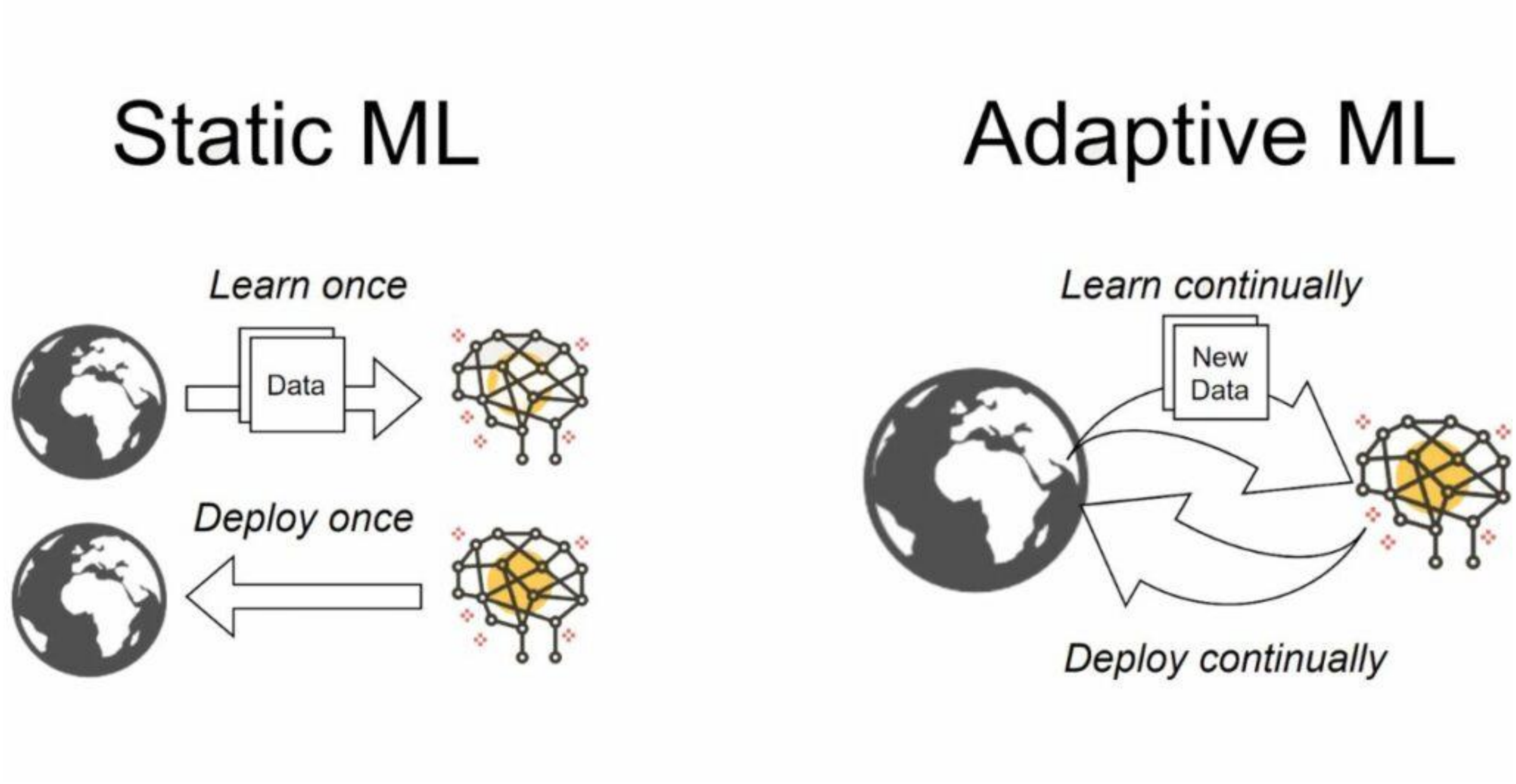
Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood & Richard S. Sutton

*Nature* **632**, 768–774 (2024) | Cite this article

**86k** Accesses | **11** Citations | **209** Altmetric | Metrics

# Lifelong Learning

- Train a model with incremental data
  - Must mitigate catastrophic forgetting
  - Must mitigate loss of plasticity



# Deep learning doesn’t work for continual updates

- Modern deep learning relies on replay buffers for continual learning
- Earlier problems with deep continual learning
  - Catastrophic Forgetting
  - Loss of Plasticity
  - The failure of warm-starting
  - In the context of deep reinforcement learning …
    - Primacy Bias and resetting in deep RL
    - Capacity loss in RL
  - But no one has done a direct test or demonstration of loss of plasticity in supervised learning
- Plasticity loss = losing the ability to learn after multiple rounds of training

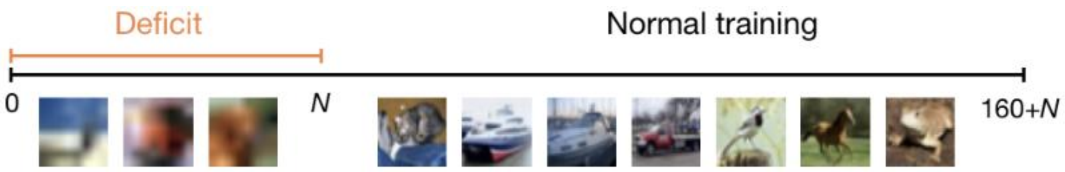
# History of related topics

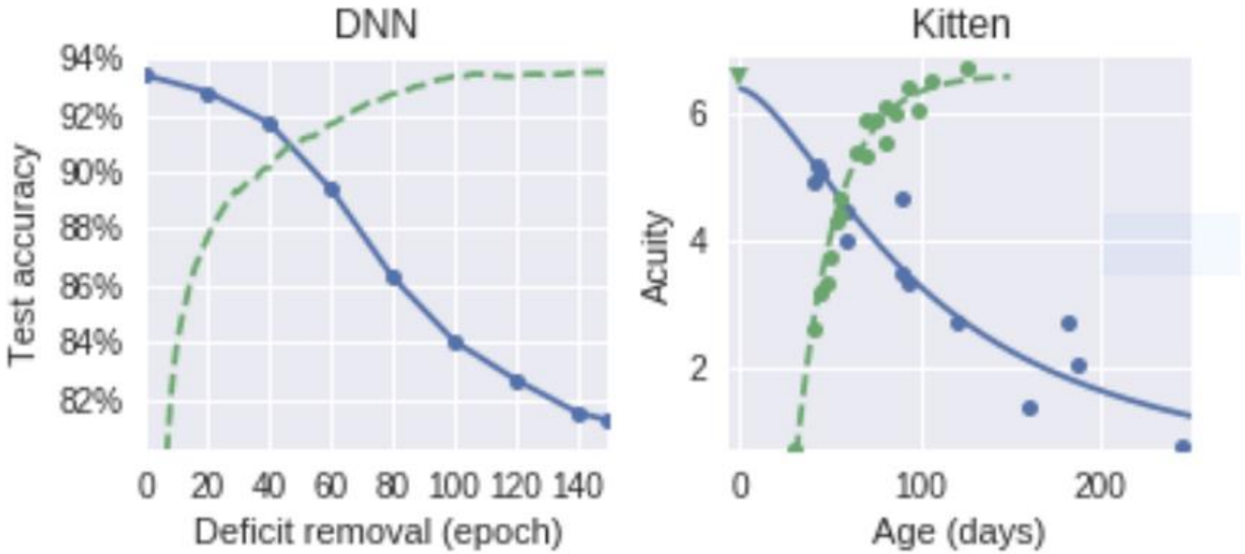
- In the context of pre-training
  - Critical learning period (Achille 2017)
  - Warm-starting DNN training (Ash&Adams 2020)
- In the context of continual learning
  - Intransigence (Chaudry 2018)
  - Degradation in backprop’s ability to learn (Dohare 2021)
- In the context of reinforcement learning
  - Primacy bias (Nikishin 2022)
  - Dormant neuron phenomenon (Sokar 2023)
  - Implicit under-parameterization (Kumar 2021)
  - Capacity loss (Lyle 2022)
- Loss of plasticity (Lyle 2023) = the all-encompassing of above

[Intransigence] Chaudhry, Arslan, et al. "Riemannian walk for incremental

# Critical learning periods in DNNs

- Studied the effect of sensory deficit at early training phase
  - Image blurring simulates a cataract(백내장)-like sensory deficit.
  - First few epochs are critical to create strong connection to input distribution. Once created, they do not appear to change during training.
- Training on blurred CIFAR images reduced its ability to subsequently learn on clean CIFAR images.





[Critical] Achille, Alessandro, Matteo Rovere, and Stefano Soatto. "Critical learning periods in deep networks." *ICLR* 2018.

# Warm-starting is actually damaging

- “Warm-starting”: warm-up the training with a portion of the training samples in hopes to accelerate convergence.
- Somehow, warm-starting yields worse generalization.

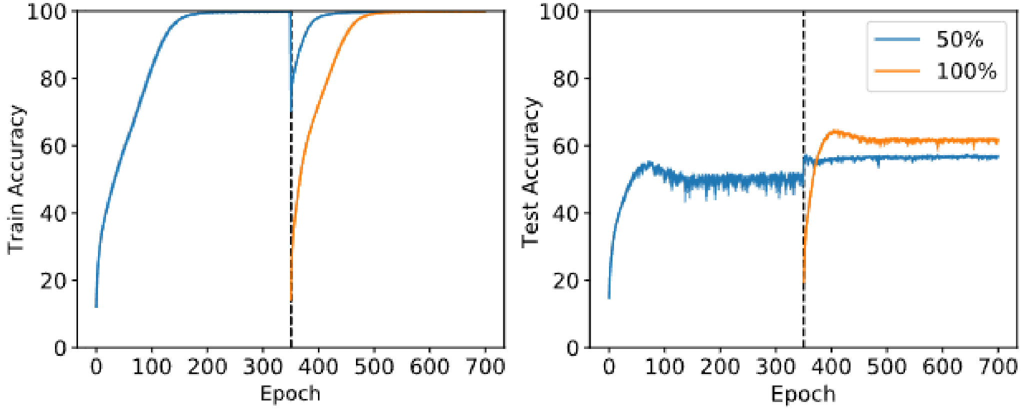
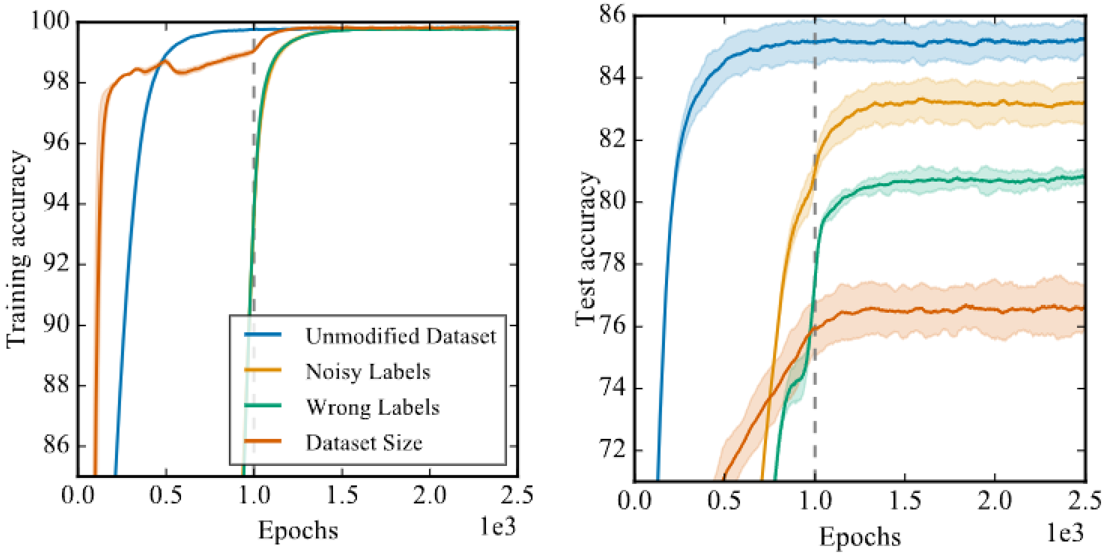


Figure 1: A comparison between ResNets trained using a warm start and a random initialization on CIFAR-10. Blue lines are models trained on 50% of CIFAR-10 for 350 epochs then trained on 100% of the data for a further 350 epochs. Orange lines are models trained on 100% of the data from the start. The two procedures produce similar training performance but differing test performance.

[Shrink&Perturb] Ash, Jordan, and Ryan P. Adams. "On warm-starting neural network training." *NeurIPS 2020*

# Warm-starting is actually damaging

- Not only that, warm-starting with the ground truth labels yields poorer accuracy than warm-starting with wrong labels.
- Shows that non-stationarity early in training has a permanent effect.



Noisy labels: incorrect labels are drawn at each step
Wrong labels: labels are replaced by random labels
Dataset size: gradually add more training samples

**Figure 1:** Accuracy on CIFAR-10 when the training data is non-stationary over the first 1000 epochs (dashed line). The remaining epochs are trained on the full, unaltered training data. Testing is performed on unaltered data throughout. While final training performance (left) is almost unaffected, test accuracy (right) is significantly reduced by initial, transient non-stationarity.

Igl, Maximilian, et al. "Transient non-stationarity and generalisation in deep reinforcement learning." *ICLR 2021*

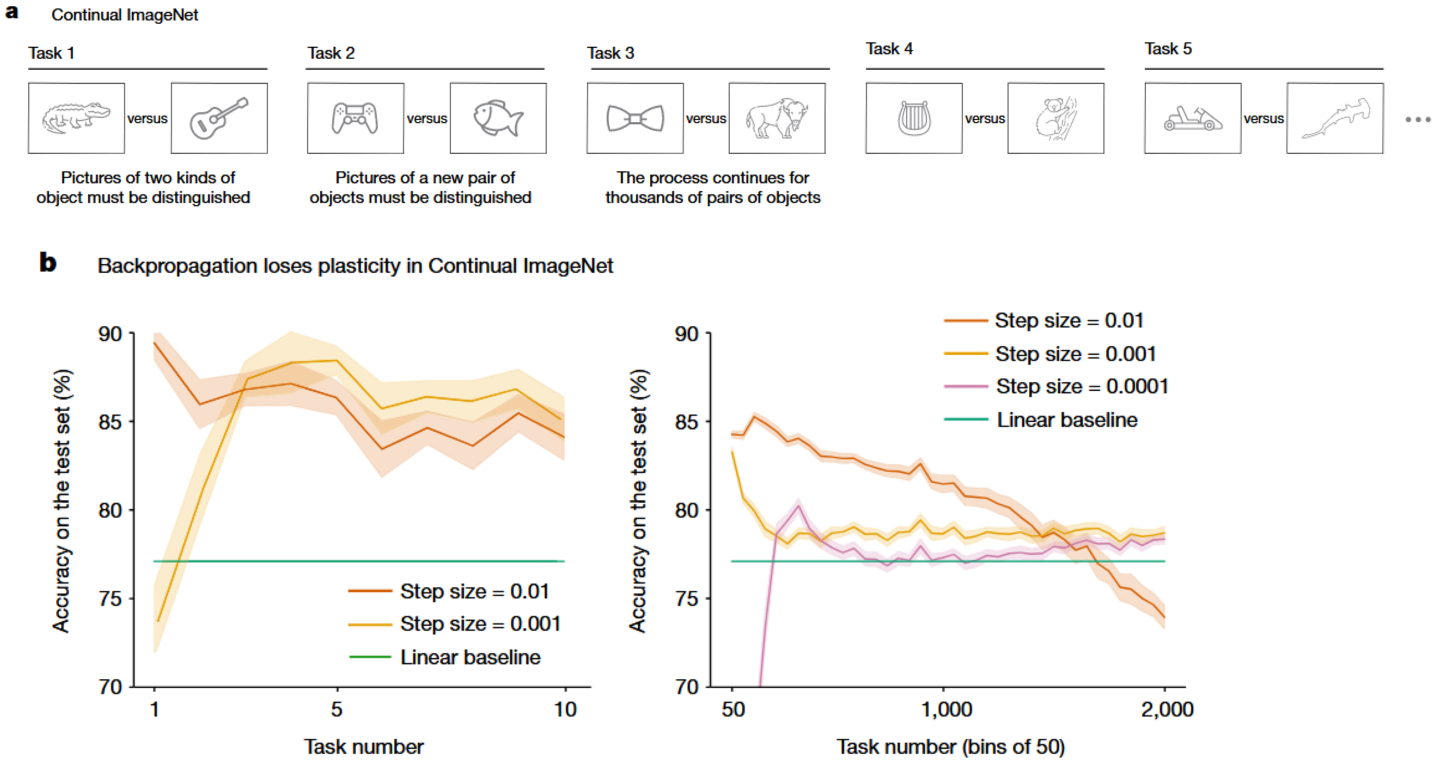
# Plasticity loss in neural nets

- Plasticity loss becomes evident only after a long sequence of tasks (Dohare 2021)
- Continual learning: This becomes an important challenge in today’s agentic AI, where the agent encounter many non-stationary data stream over a long lifetime.
- Reinforcement learning: (1) observation is non-stationary (2) common RL methods using temporal difference learning bootstrap off of the predictions of a periodically updating target network. (DQN) The changing target introduces non-stationarity.

Dohare, Shibhansh, Richard S. Sutton, and A. Rupam Mahmood. "Continual backprop: Stochastic gradient descent with persistent randomness." *arXiv* 2021

# Demonstrating plasticity loss

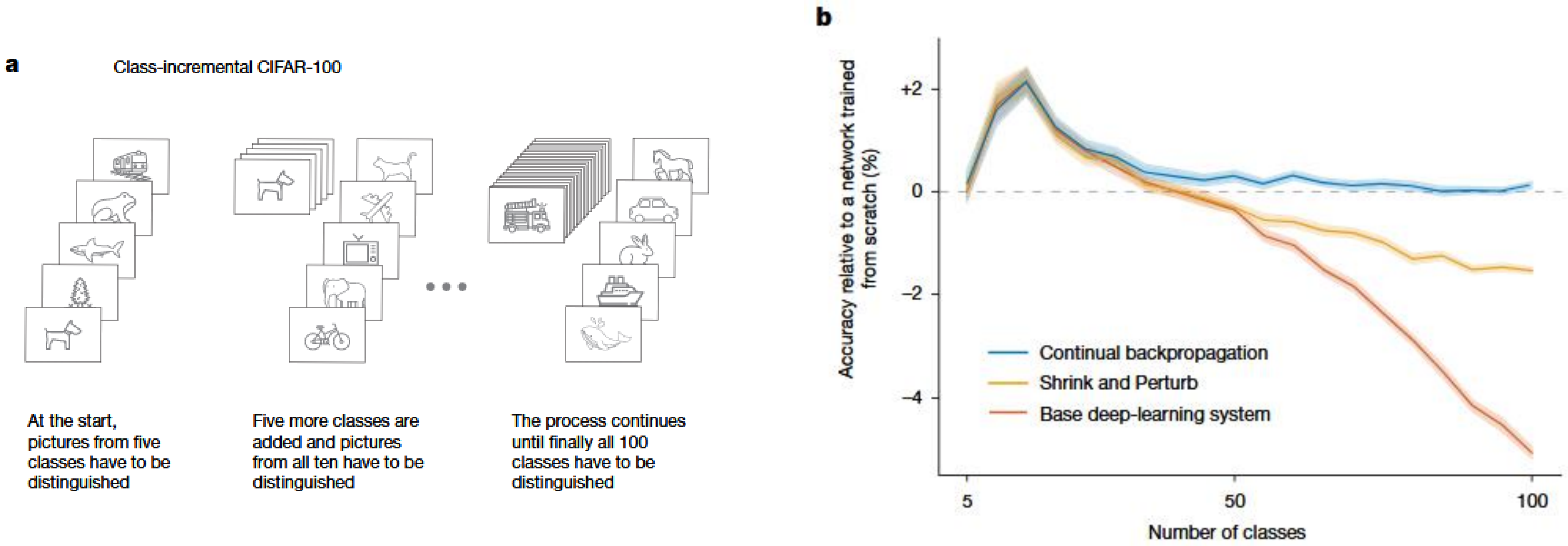
- *Continual ImageNet* benchmark
- The plasticity gradually degrades until the model performs no better than the linear baseline. This effect accumulates over tasks.



Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

# Demonstrating plasticity loss

- *Class-incremental CIFAR-100* benchmark



At the start, pictures from five classes have to be distinguished

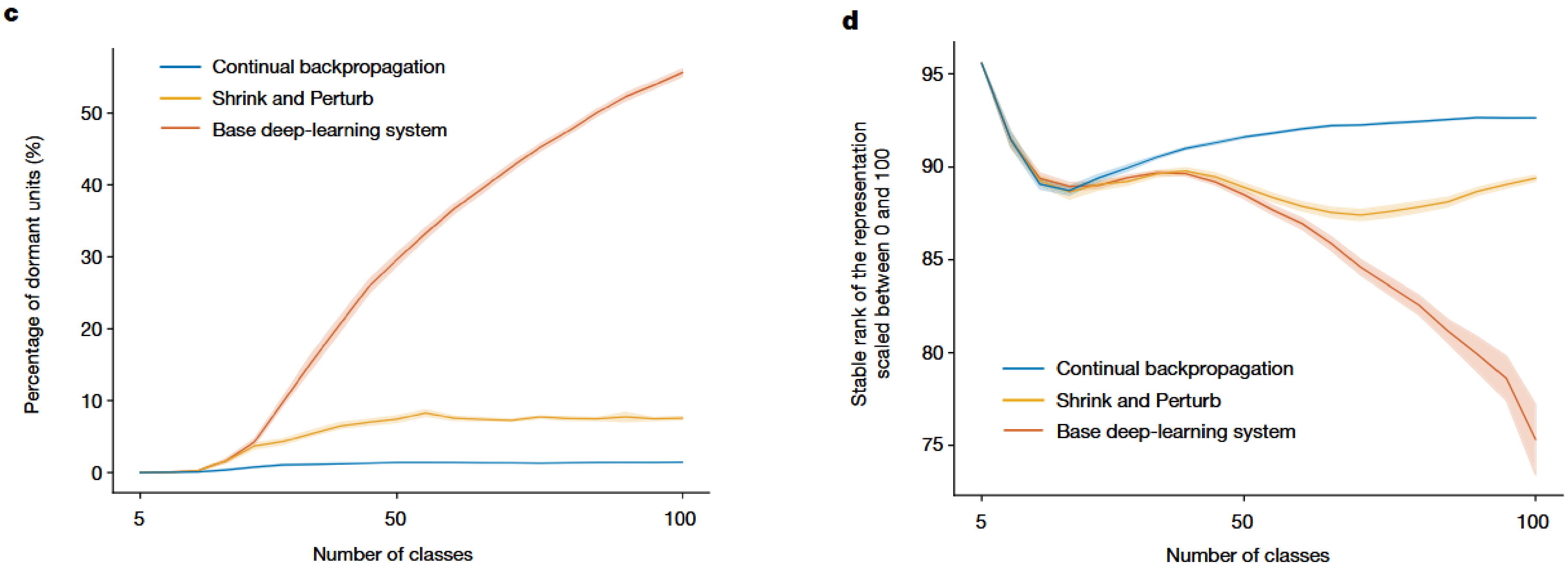
Five more classes are added and pictures from all ten have to be distinguished

The process continues until finally all 100 classes have to be distinguished

Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

# Demonstrating plasticity loss

- *Class-incremental CIFAR-100* benchmark
- Plasticity loss is characterized by increased dead units and feature rank collapse.



Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

# Plasticity loss in deep RL

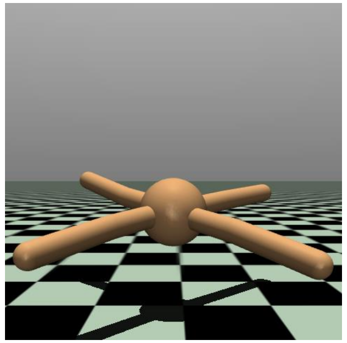
- Deep RL agents may gradually lose the ability to learn from new experiences
- Degradation in backprop's ability to learn (Dohare 2021)
- "Implicit under-parameterization" – loss of expressivity characterized by a drop in the rank of the learned value network features (Kumar 2021)
- "Capacity loss" – deep RL agent trained on a sequence of target values lose their ability to quickly update their predictions over time. (Lyle 2022)
- "Primacy bias" – a tendency to overfit initial experiences that damages the rest of the learning process (Nikishin 2022)

[CBP] Dohare, Shibhansh, Richard S. Sutton, and A. Rupam Mahmood. "Continual backprop: Stochastic gradient descent with persistent randomness." *arXiv* 2021.

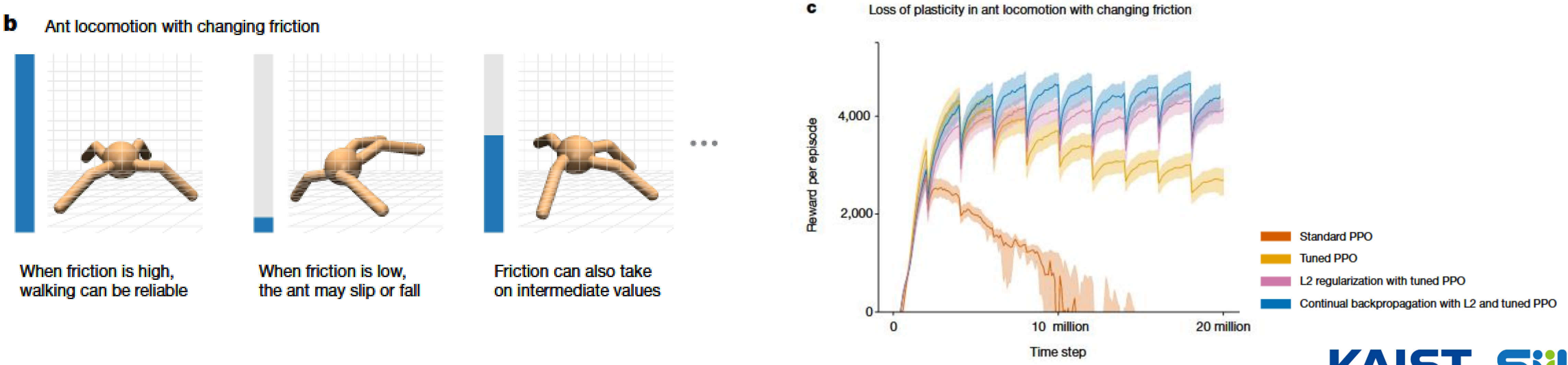
[Implicit] Kumar, Aviral, et al. "Implicit under-parameterization inhibits data-efficient deep reinforcement learning." ICLR 2021

# Demonstrating plasticity loss in RL problems

- *MuJoCo-Ant* environment
  - Action space = torque applied to the leg joints
  - Observation space = orientation of the leg joints
  - Reward = forward distance travelled
  - The friction is changed regularly for non-stationarity



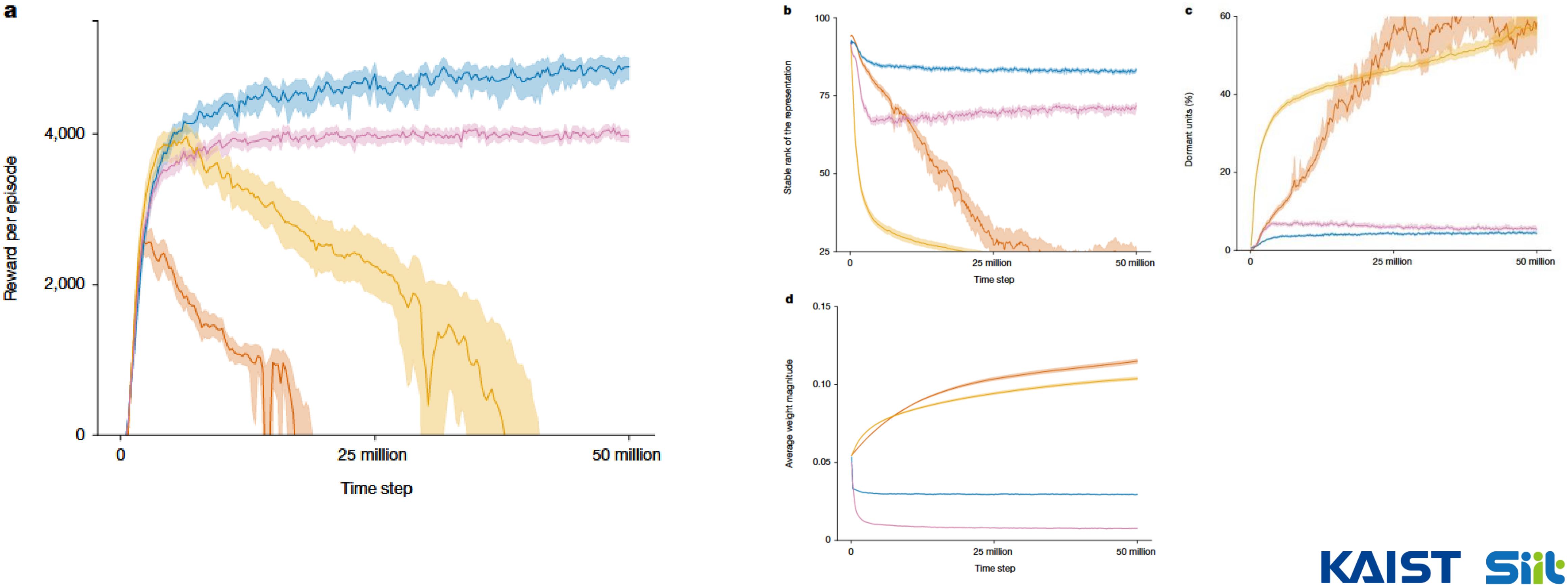
- PPO agent show substantial loss of plasticity after friction change



Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

# Demonstrating plasticity loss in RL problems

- Plasticity loss affects even a non-stationary problem (constant friction) after extended training



Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

# Probing plasticity loss by dormant neurons

- Dormant neuron phenomenon

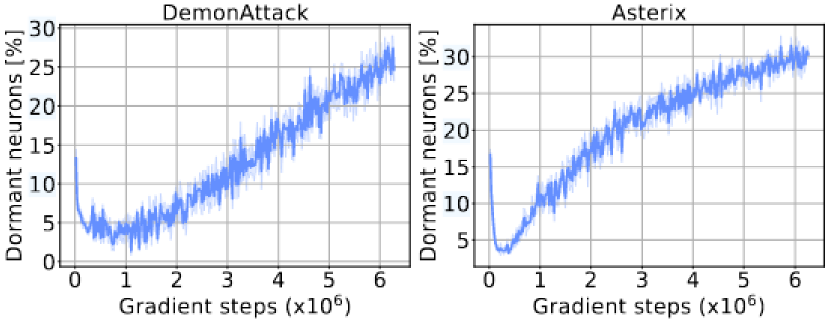
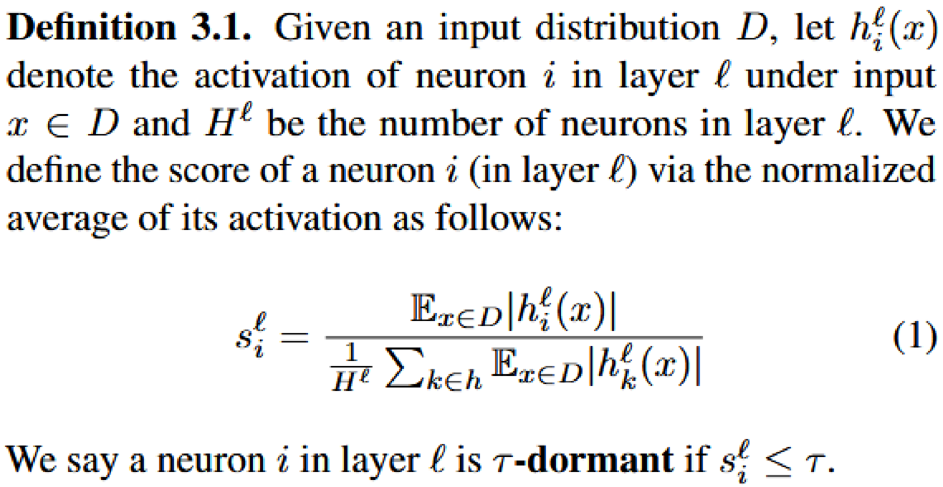


*Figure 2.* The percentage of dormant neurons increases throughout training for DQN agents.



**Definition 3.1.** Given an input distribution $D$, let $h_i^\ell(x)$ denote the activation of neuron $i$ in layer $\ell$ under input $x \in D$ and $H^\ell$ be the number of neurons in layer $\ell$. We define the score of a neuron $i$ (in layer $\ell$) via the normalized average of its activation as follows:

$$s_i^\ell = \frac{\mathbb{E}_{x\in D}|h_i^\ell(x)|}{\frac{1}{H^\ell}\sum_{k\in h}\mathbb{E}_{x\in D}|h_k^\ell(x)|} \tag{1}$$

We say a neuron $i$ in layer $\ell$ is $\tau$-**dormant** if $s_i^\ell \leq \tau$.

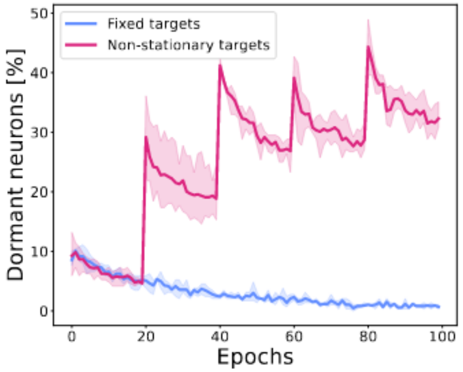
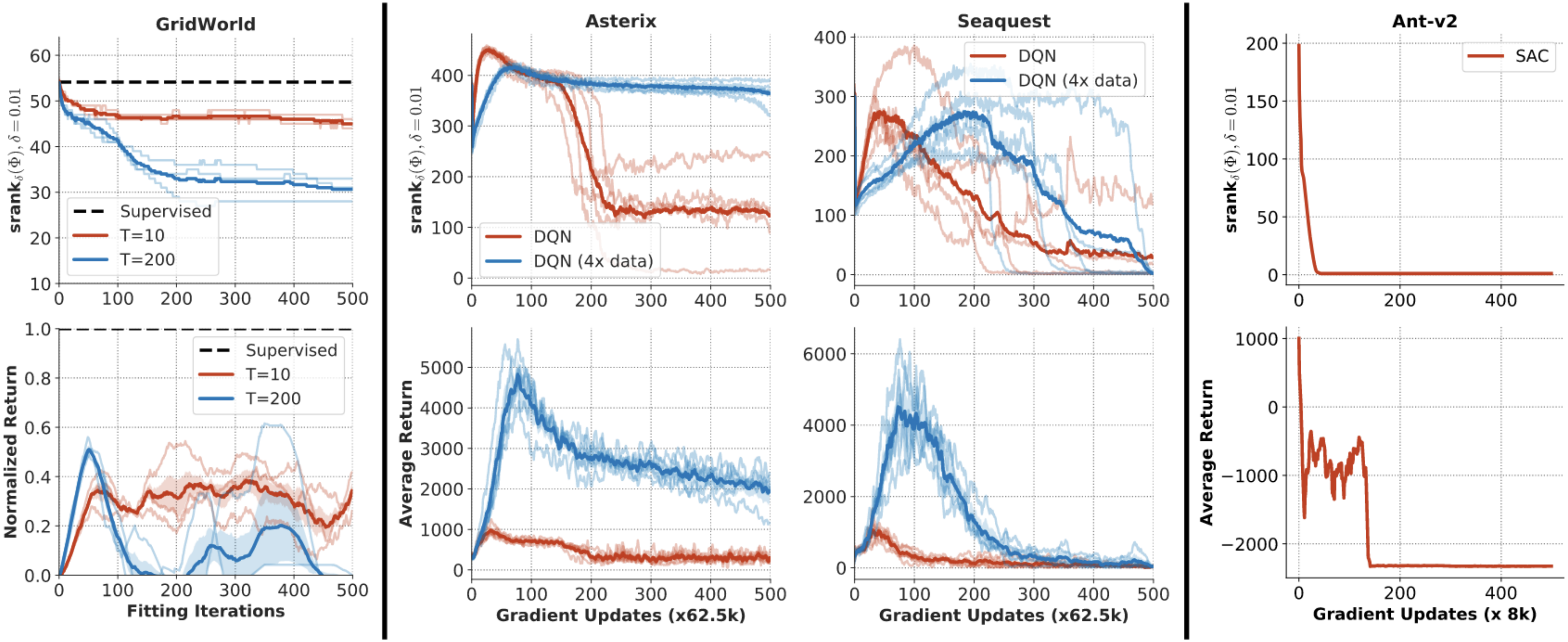


*Figure 3.* Percentage of dormant neurons when training on CIFAR-10 with fixed and non-stationary targets. Averaged over 3 independent seeds with shaded areas reporting 95% confidence intervals. The percentage of dormant neurons increases with non-stationary targets.

[ReDo] Sokar, Ghada, et al. "The dormant neuron phenomenon in deep reinforcement learning." *ICML* 2023 (oral)

# Probing plasticity loss by decreased feature rank



[Implicit] Kumar, Aviral, et al. "Implicit under-parameterization inhibits data-efficient deep reinforcement learning." ICLR 2021

# How to measure redundancy in activations

- Generalization of matrix rank to a continuous scale
- Effective rank (Roy and Vetterli 2007)
- For a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$ with its normalized singular values $\mathbf{p} = \left\{\frac{\sigma_k}{\sum \sigma_i}\right\}_{k=1}^{\min\{m,n\}}$, $\mathrm{erank}(\boldsymbol{\Phi}) = e^{\mathrm{H}(\mathbf{p})}$
- Erank ranges between 1 and $\min\{m, n\}$
- Erank quantifies how uniform the singular values are.
- No hyperparameter is required to define the numerical rank.

[erank] Roy, Olivier, and Martin Vetterli. "The effective rank: A measure of effective dimensionality." *2007 15th European signal processing conference*. IEEE, 2007.

# How to measure redundancy in activations

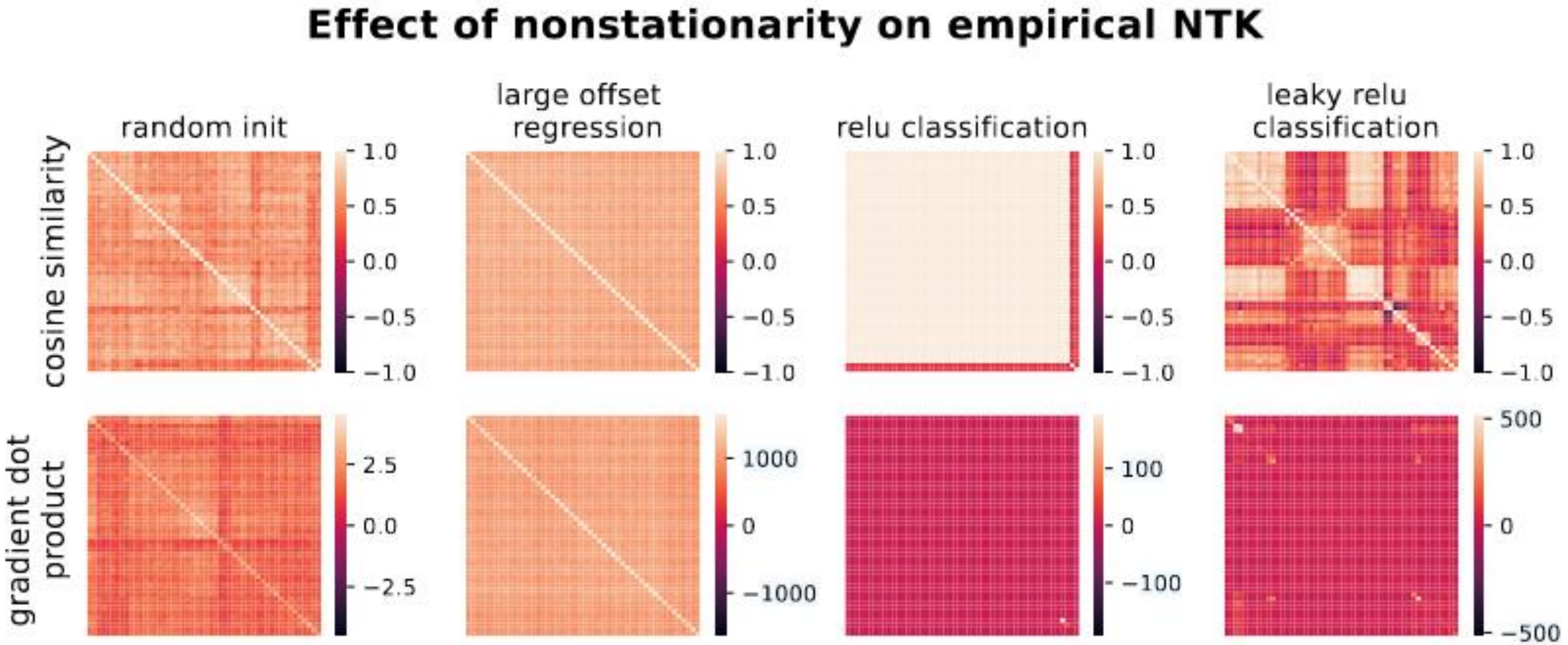
- There are several ways to quantify numerical rank
- Stable rank (Kumar 2021)
- $\mathrm{srank}(\boldsymbol{\Phi}) = \min\left\{k \middle| \frac{\sum_{i=1}^{k} \sigma_i}{\sum \sigma_i} \geq 1 - \delta\right\}$
- The number of rank that make up the $(1 - \delta)$% of the total norm
- Requires a hyperparameter $\delta$ to stabilize the rank. (e.g., $\delta = 0.01$)

[Implicit] Kumar, Aviral, et al. "Implicit under-parameterization inhibits data-efficient deep reinforcement learning." *ICLR 2021*

[Capacity loss] Lyle, Clare, Mark Rowland, and Will Dabney. "Understanding and Preventing Capacity Loss in Reinforcement Learning." *ICLR 2022 (spotlight)*

# Visualizing generalization in a neural net

- Empirical NTK matrix: $\kappa_\theta(x_i, x_j) = \langle \nabla_\theta f(x_i), \nabla_\theta f(x_j) \rangle$
- The off-diagonals show how the neural net generalizes.
- A good initialization should have large off-diagonal magnitudes



Lyle, Clare, et al. "Disentangling the causes of plasticity loss in neural networks." *CoLLAs* 2024.

# What causes plasticity loss?

- We suspect that multiple mechanisms are involved.
- Distribution shift in preactivations -> dead neurons (Sokar 2023, Dohare et al., Lyle et al.)
- Rank collapse in activations (Kumar 2021, Dohare et al.)
- Linearization (Lyle 2024)
- Growth in parameter norm -> training instabilities (Lyle 2024)
- Low-rank structure in empirical NTK (Lyle 2024)

[Implicit] Kumar, Aviral, et al. "Implicit under-parameterization inhibits data-efficient deep reinforcement learning." ICLR 2021
Lyle, Clare, et al. "Disentangling the causes of plasticity loss in neural networks." *CoLLAs* 2024.

# What causes plasticity loss? – preactivation shift

- (1) dormant neurons – Rapid distribution shift in preactivations causes dormant neurons (Sokar 2023, Dohare et al., Lyle et al.)
    - Dead neuron is unable to backpropagate the error signal
- (2) linear neurons – limits the expressivity of neural net. A ReLU with always-positive preactivation.

Lyle, Clare, et al. "Disentangling the causes of plasticity loss in neural networks." *CoLLAs* 2024.
[ReDo] Sokar, Ghada, et al. "The dormant neuron phenomenon in deep reinforcement learning." *ICML* 2023 (oral)

# What causes plasticity loss? – parameter norm

- Growth in parameter norm
- -> numerical instabilities during training.
- -> Increased sharpness of loss landscape (Hessian norm)
- -> saturation effect in softmax (e.g., attention head) or normalization layer (e.g., layernorm)

[Primacy bias] Nikishin, Evgenii, et al. "The primacy bias in deep reinforcement learning." *ICML* 2022.
Lyle, Clare, et al. "Disentangling the causes of plasticity loss in neural networks." *CoLLAs* 2024.

# What causes plasticity loss? – optimizer state

- Adam optimizer state: first- and second-moment of gradient
- The default config for first-moment decay rate ($\beta_1 = 0.9$) and second-moment decay rate ($\beta_2 = 0.999$) cause a large loss in plasticity.
- Mismatch in two decay rate makes the training unstable when the optimizer receives a sudden large gradient.

Lyle, Clare, et al. "Understanding plasticity in neural networks." *ICML* 2023 (oral)

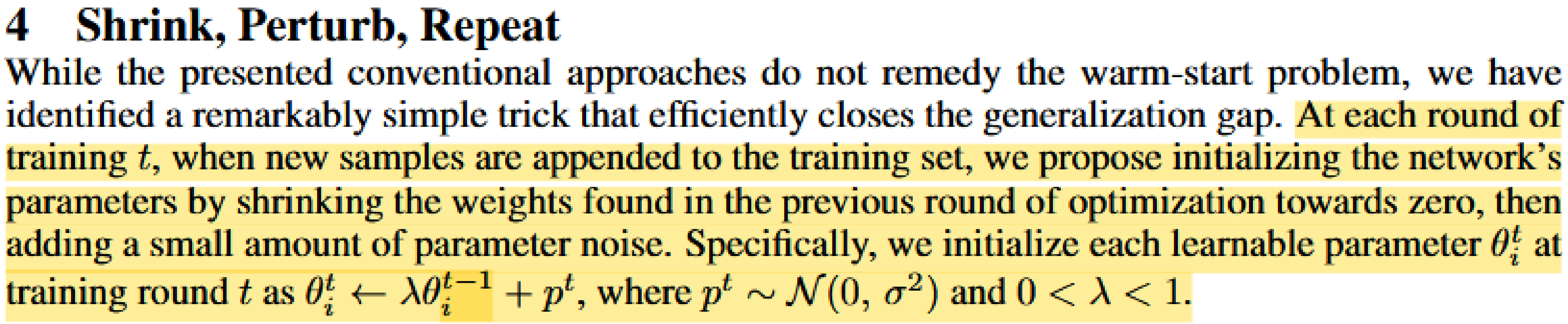
# Strategies for mitigation

- The naïve way
  - Stack all training data in a memory and re-train from scratch (Igl 2020)
- Weight resetting
  - Continual backprop (Dohare 2021), ReDo (Sokar 2023)
- Regularization
  - L2 regularization, Shrink-and-perturb (Ash & Adams 2020), L2 Init (Kumar 2024)
- Architectural mitigations
  - Layernorm (Lyle 2023), Concatenated ReLU (Abbas 2023)
- Optimizer design
  - Modifications to the default Adam optimizer config (Lyle 2023)
- Re-initializing parameters appears to be a viable strategy

Kumar et al. "Maintaining plasticity in continual learning via regenerative regularization." *CoLLAs* 2024.

# Shrink & Perturb

- When new samples are added to the training set, apply L2 regularization (shrink) + random noise (perturb) to weights.
- The noise inject randomness to the network which prevents units from dying.



## 4 Shrink, Perturb, Repeat

While the presented conventional approaches do not remedy the warm-start problem, we have identified a remarkably simple trick that efficiently closes the generalization gap. At each round of training $t$, when new samples are appended to the training set, we propose initializing the network's parameters by shrinking the weights found in the previous round of optimization towards zero, then adding a small amount of parameter noise. Specifically, we initialize each learnable parameter $\theta_i^t$ at training round $t$ as $\theta_i^t \leftarrow \lambda\theta_i^{t-1} + p^t$, where $p^t \sim \mathcal{N}(0, \sigma^2)$ and $0 < \lambda < 1$.

[Shrink&Perturb] Ash, Jordan, and Ryan P. Adams. "On warm-starting neural network training." *NeurIPS 2020*

# Continual Backprop

- Re-initialize a few under-utilized units at each training step
- Units are selected upon the smallest “contribution utility”
  - Contribution utility = moving average of (activation) x (sum of outgoing weights)

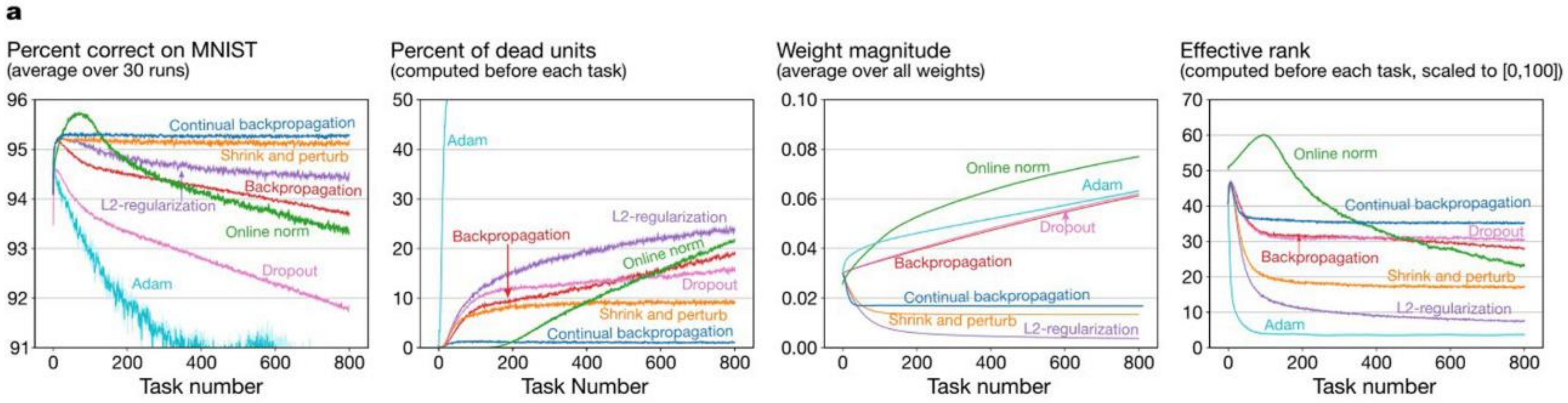
$$\mathbf{u}_l[i] = \eta \times \mathbf{u}_l[i] + (1-\eta) \times |\mathbf{h}_{l,i,t}| \times \sum_{k=1}^{n_{l+1}} |\mathbf{w}_{l,i,k,t}|,$$

- Hyperparameters: replacement rate $\rho$, maturity threshold $m$
  - Replacement rate is typically set to a very small value (~10E-5)
  - Maturity threshold protects the new unit from immediate re-initialization

[CBP] Dohare, Shibhansh, Richard S. Sutton, and A. Rupam Mahmood. "Continual backprop: Stochastic gradient descent with persistent randomness." *arXiv* 2021.

# What causes plasticity loss?

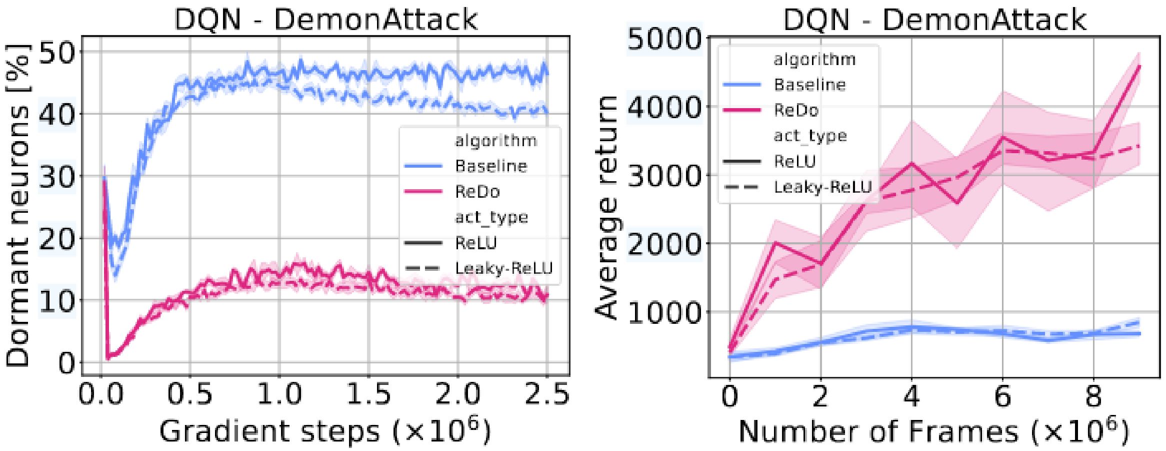
- Some existing techniques such as Adam, Dropout, and normalization actually amplifies plasticity loss under non-stationarity.
- Regularizing the weight norm and variability injection seems to be important for plasticity.



Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)

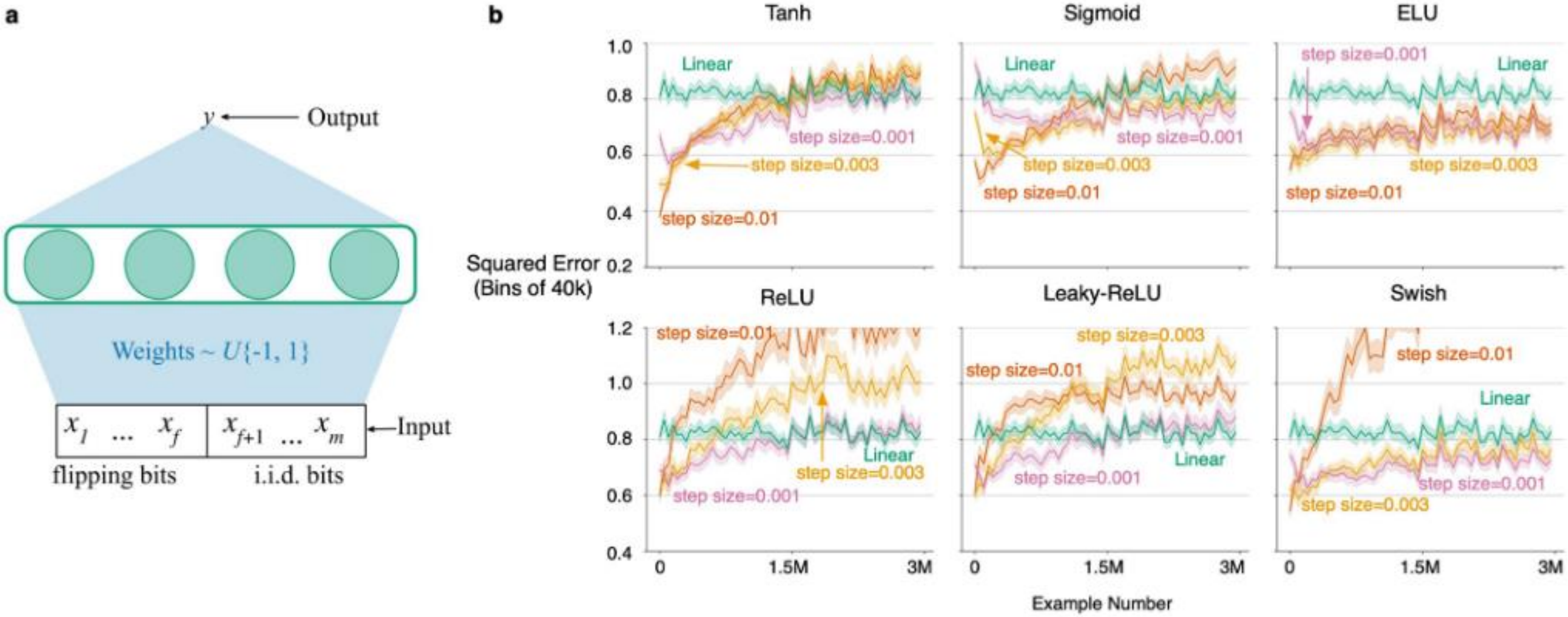
# What causes plasticity loss? – ReLU activation

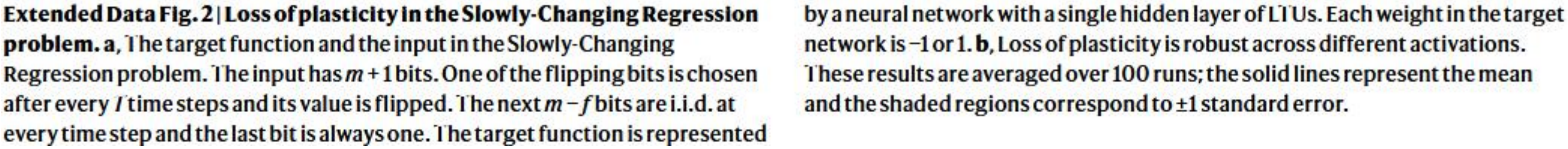
- LeakyReLU slightly improves dormant neurons, but does not mitigate the issue. So this is not just about the dying ReLU problem.
- LeakyReLU (non-saturating nonlinearity) can prevent dying, however preactivation deviation from initialization can still harm signal propagation and cause negative effect to training dynamics.



Lyle, Clare, et al. "Disentangling the causes of plasticity loss in neural networks." *CoLLAs* 2024.
[ReDo] Sokar, Ghada, et al. "The dormant neuron phenomenon in deep reinforcement learning." *ICML* 2023 (oral)

# What causes plasticity loss? – ReLU activation





**Extended Data Fig. 2 | Loss of plasticity in the Slowly-Changing Regression problem. a,** The target function and the input in the Slowly-Changing Regression problem. The input has $m + 1$ bits. One of the flipping bits is chosen after every $T$ time steps and its value is flipped. The next $m - f$ bits are i.i.d. at every time step and the last bit is always one. The target function is represented by a neural network with a single hidden layer of LTUs. Each weight in the target network is −1 or 1. **b,** Loss of plasticity is robust across different activations. These results are averaged over 100 runs; the solid lines represent the mean and the shaded regions correspond to ±1 standard error.

Dohare, Shibhansh, et al. "Loss of plasticity in deep continual learning." *Nature* (2024)