NICE: Supplementary Material

A. Experimental Details

A.1. Datasets and Episodes

We have developed class-incremental learning (CIL) scenarios by slicing common image classification benchmark datasets. For MNIST, FashionMNIST, and EMNIST (English letters, lower and upper cases correspond to the same class), as well as CIFAR10, each learning episode comprises two consecutive classes. In contrast, CIFAR100's episodes include 10 consecutive classes, while those of Tiny ImageNet encompass 40 classes. Consequently, MNIST, FashionMNIST, CIFAR10, and Tiny ImageNet each consist of 5 learning episodes, EMNIST has 13, and CIFAR100 has 10 episodes. This approach was chosen to present a range of scenarios, from those with longer sequences and easier episodes (EMNIST), and relatively longer sequences with challenging classes (CIFAR100), to those with more challenging, shorter sequences (TinyImageNet). While creating the learning episodes, we adhered to the original class order and refrained from data augmentation during training, limiting our data preprocessing to pixel normalization.

A.2. Architectures

In our experiments, we utilized three distinct architectures. Firstly, for the MNIST, FashionMNIST, and EMNIST datasets, we implemented two convolutional layers (each with a 3x3 kernel, stride of 1, and 32 filters) followed by max-pooling. These layers precede a fully connected layer with 500 neurons and an output layer, the size of which varies depending on the number of classes in each dataset. Secondly, for the CIFAR10 and CIFAR100 experiments, we employed a modified version of the VGG11 architecture, originally designed for the ImageNet dataset. To tailor it for smaller datasets, we halved the number of convolutional filters and equipped the hidden fully connected layers with 1024 neurons each. Finally, for the Tiny ImageNet dataset, we adopted the ResNet18 architecture, with the sole modification being the first convolutional layer, which uses a 3x3 kernel instead of the standard 7x7, better suiting smaller input sizes.

A.3. Baselines

We have compared NICE with eight established replay baselines:

- **Experience Replay (ER):** This straightforward replay algorithm enables sequential learning of new classes by augmenting the training batches with samples from all past classes stored in the memory.
- **Dark Experience Replay (DER):** A simple extension of the ER algorithm, DER also employs raw sample replay. However, unlike storing hard class labels, it retains the logits from the trained model as targets for replayed samples.
- **eXtended Dark Experience Replay (x-DER):** This is a follow-up method proposed by the authors of DER, analyzing its pitfalls. This extended version introduces multiple innovations such as memory content editing and adjusting logits tied to unseen classes.
- **Task-specific Attention Modules in Lifelong learning (TAMiL):** This approach encompasses both raw replay and the incremental addition of "task-specific attention modules" (TAMs) on the fly. First, a backbone learns current classes and utilizes replay to avoid forgetting. Next, representations output by this backbone are filtered using TAMs that capture task-specific information.
- **Function Distance Regularization (FDR):** Similar to DER, FDR involves replaying raw examples from previous classes. It uses past examples and network outputs to align current and previous outputs.
- **Incremental Classifier and Representation Learning (iCaRL):** iCaRL is a sophisticated raw replay method that also relies on revisiting all past classes. It is one of the most well-known replay methods and usually perform well across different datasets and architectures. It has three main components: classification by a nearest-mean-of-exemplars rule, prioritized exemplar selection based on herding, and representation learning using knowledge distillation and prototype rehearsal
- **Greedy Sampler and Dumb Learner (GDumb):** GDumb greedily stores samples in memory as they come and at test time, trains a model from scratch using samples only in the memory.
- **Averaged Gradient Episodic Memory (A-GEM):** A-GEM stands out among these baselines. While it stores raw samples, it doesn't use them directly in training. Instead, it projects gradients of new tasks based on gradients computed for memory samples. This approach necessitates storing raw examples for specific gradient computations during training.

As highlighted in the original paper, our benchmark focuses solely on "raw" replay methods. This decision is because generative and activation replay approaches typically aim to approximate "raw" replay. However, they often fall short in performance or depend on pretraining, which could lead to an unfair comparison.

A.4. Hyperparameter Selection

We trained all models using Stochastic Gradient Descent (SGD) with momentum. To ensure optimal performance, we tuned the hyperparameters for each method and reported the best results. Our tuning process involved a grid search of three different learning rates and batch sizes for all baselines. We did not employ learning rate decay (except for GDumb) or weight decay. The replay batch size was matched with the normal batch size. For method-specific hyperparameters, such as DER's alpha or GDumb's cutmix alpha, we conducted an additional hyperparameter search. This search included the baseline values suggested in the original papers, whenever available. We selected hyperparameters based on the validation accuracy, specifically focusing on a memory budget of $\times 2$ and seed 0. These parameters were then applied across other budgets and seeds. The training duration varied across datasets: 25 epochs for MNIST, FashionMNIST, and EMNIST; 50/75 epochs for CIFAR10/100; and 100 epochs for TinyImagenet.

Similar to some replay methods, NICE introduces an additional parameter, p, which determines the frequency of neuron selection, pruning, and updates for both memory and the context detector. Arguably, τ, which specifies the target activation fraction in Equation 2, can also be considered a hyperparameter. However, we found that setting it to 0.95 yielded consistent results across all datasets, so we kept it fixed. NICE also incorporates logistic regression within its context detector. For this, we primarily used the default parameters from Scikit-Learn. Nevertheless, we adjusted the maximum iterations and regularization coefficient according to the memory size. Refer to **Table** 1 and **Table** 2 for the best hyperparameters we used to report the results

Table 1. Hyperparameters for MNIST, FashionMNIST, and EMNIST: GDumb requires more epochs as it only trains on memory. We train it for 500 epochs on these datasets.

	Datasets		
Method	MNIST	FashionMNIST	EMNIST
Joint	BS: 128 LR: 0.01	BS: 128 LR: 0.01	BS: 128 LR: 0.01
NICE	BS: 32 LR: 0.01 p: 5	BS: 32 LR: 0.005 p: 5	BS: 32 LR: 0.005 p: 5
iCaRL	BS: 64 LR: 0.01	BS: 32 LR: 0.01	BS: 32 LR: 0.01
TAMiL	BS: 32 LR: 0.005 α: 0.1 β: 0.5 code_dim: 8	BS: 32 LR: 0.005 α: 0.1 β: 0.5 code_dim: 8	BS: 64 LR: 0.005 α: 0.1 β: 0.5 code_dim: 8
x-DER	BS: 128 LR: 0.005 α: 0.2 β: 0.8	BS: 128 LR: 0.001 α: 0.2 β: 0.8	BS: 128 LR: 0.001 α: 0.2 β: 0.8
DER	BS: 32 LR: 0.005 α: 0.2	BS: 128 LR: 0.001 α: 0.2	BS: 0.001 LR: 32 α: 0.2
GDumb	BS: 32 LR: (0.05, 0.005) cut-mix: n/a	BS: 32 LR: (0.05, 0.005) cut-mix: n/a	BS: 32 LR: (0.05, 0.005) cut-mix: n/a
ER	BS: 32 LR: 0.005	BS: 128 LR: 0.001	BS: 128 LR: 0.001
FDR	BS: 32 LR: 0.01 α 0.2:	BS: 64 LR: 0.001 α: 0.2	BS: 32 LR: 0.001 α: 0.2
SGD	BS: 128 LR: 0.01	BS: 128 LR: 0.01	BS: 128 LR: 0.01

Table 2. Hyperparameters for CIFAR10, CIFAR100, and Tiny ImageNet: GDumb requires more epochs as it only trains on memory. We train it for 500, 1000, and 3000 epochs on CIFAR10, CIFAR100, and Tiny ImageNet, respectively..

	Datasets		
Method	CIFAR10	CIFAR100	Tiny ImageNet
Joint	BS: 32 LR: 0.01	BS: 32 LR: 0.005	BS: 128 LR: 0.01
NICE	BS: 32 LR: 0.01 p: 10	BS: 32 LR: 0.005 p: 15	BS: 64 LR: 0.01 p: 20
iCaRL	BS: 32 LR: 0.005	BS: 32 LR: 0.01	BS: 32 LR: 0.01
TAMiL	BS: 32 LR: 0.03 α: 0.1 β: 0.5 code_dim: 64	BS: 32 LR: 0.03 α: 0.1 β: 0.5 code_dim: 64	BS: 64 LR: 0.001 α: 0.1 β: 1.0 code_dim: 64
x-DER	BS: 64 LR: 0.005 α: 0.2 β: 0.8	BS: 32 LR: 0.001 α: 0.2 β: 0.8	BS: 32 LR: 0.001 α: 0.2 β: 0.8
DER	BS: 64 LR: 0.005 α: 0.2	BS: 32 LR: 0.001 α: 0.2	BS: 32 LR: 0.001 α: 0.2
GDumb	BS: 32 LR: (0.05, 0.005) cut-mix: 1.0	BS: 32 LR: (0.01, 0.0001) cut-mix: 1.0	BS: 32 LR: (0.01, 0.0001) cut-mix: 1.0
ER	BS: 32 LR: 0.01	BS: 32 LR: 0.001	BS: 32 LR: 0.001
FDR	BS: 32 LR: 0.005 α: 0.2	BS:32 LR: 0.001 α: 0.2	BS: 32 LR: 0.001 α: 0.2
SGD	BS: 128 LR: 0.01	BS: 128 LR: 0.01	BS: 128 LR: 0.01

A.5. NICE Parameter Count

The trainable parameter count in NICE undergoes minor variations during the training process. As connections are pruned from younger to older neurons throughout training, the total parameter count decreases. On the other hand, fitting logistic regression models to the context-detector leads to a modest increase in the number of parameters. To ensure fairness in our benchmarks, we compare the parameter count of NICE with that of standard dense networks used in other baseline

methods. Our analysis primarily focuses on the parameter count at the end of the training period for simplicity. Moreover, we estimate the maximum logistic regression parameter count by assuming models are fitted to all neurons, rather than just those exceeding a certain age. This results in $E-1$ logistic regression models, where E is the total number of episodes observed. The parameter count for each model equals the number of neurons in the architecture, including input and output neurons, plus an additional parameter for the bias term in logistic regressions. **Table** 3 presents these calculations in detail. Overall, NICE has a notably lower parameter count than dense baseline models across various datasets.

Table 3. NICE's parameter count compared to standard dense networks.

	Parameter Counts				
Datasets	NICE Network	Logistic Regressions	NICE Total	Dense Network	Difference
MNIST	$668,329$	4×576	$670,633$	$798,504$	$-127,871$
FashionMNIST	$656,412$	4×576	$658,716$	$798,504$	$-139,788$
EMNIST	$620,066$	12×592	$627,170$	$806,504$	$-179,334$
CIFAR10	$2,996,963$	4×3438	$3,010,715$	$4,412,256$	$-1,401,541$
CIFAR100	$2,995,482$	9×3528	$3,027,234$	$4,504,416$	$-1,477,182$
Tiny ImageNet	$9,192,402$	4×4108	$9,208,834$	$11,605,696$	$-2,396,862$

A.6. Adding Neurons to NICE on-the-fly

In the main paper, we mention that it is possible to add neurons to the NICE architecture on-the-fly. This can be achieved easily, but one should adhere to the rule: younger neurons do not input to older ones. For example, we can expand the layers by adding age-0 neurons connected to the previous layer and the next layer's age-0 neurons. For more depth, one should connect the penultimate age > 0 neurons to the output and insert new age-0 neuron layers in between. These changes can be efficiently implemented without disrupting the accuracy on seen classes.

A.7. Implementation and Compute Resources

All experiments were conducted on Ubuntu 20.04 using an NVIDIA GeForce RTX 3090 GPU with CUDA 12.0. We utilized PyTorch 1.13.1 for deep learning components, Scikit-Learn 1.2.1 for logistic regression training, and Python 3.10.9.

B. NICE Pseudo-code

Algorithm 1 NICE Algorithm for a single Episode

Require: D_e, f, M, C ▷ *D*ataset, Network, Activation Memory, Context-detector
Require: K, p, m, T ▷ # epochs, selection period, # activations to store, set of layer thresholds
for $i = 0$ to $\frac{K}{p}$ **do**
  if $i == 0$ **then**
    $f = \text{SelectAll}(f)$ ▷ *I*nitial selection. All age-0 neurons become age-1
  else
    $f = \text{Select}(f, \tau = 0.95)$ ▷ *S*elect top neurons to achieve τ fraction of total activaton
    $f = \text{UpdateConnections}(f)$ ▷ *E*nsure no connections from age-u to age-v where $u < v$
  end if
  $f = \text{Train}(f, D_e, \text{epochs} = p)$ ▷ *T*rain age-1 neurons for p epochs
  $M = \text{UpdateMemory}(f, D_e, M, T, m)$ ▷ *S*tore m episode activations to memory
  $C = \text{UpdateContextDetector}(C, M)$ ▷ *F*it logic regressions on M
end for
$f = \text{IncrementAges}(f)$ ▷ *A*ge all neurons above age-0 by one
$f = \text{FreezeConnections}(f)$ ▷ *F*reeze incoming connections of neurons age > 1
return f, M, C

C. Optimality Proof for Greedy Selection

Recall that we compute the activation of age-1 neurons at each layer on episode examples as follows:

$$A^l_{=1} = \sum_{x_k \sim D_e} \sum_{n^l_i \in N^l_{=1}} A(n^l_i, x_k, 1) \tag{1}$$

Here, $A(n^l_i, x_k, 1)$ represents the activation of the neuron n^l_i for the sample x_k, considering only incoming activations from previous layer neurons of age-1. Our selection strategy relies on solving the following discrete optimization problem:

$$\min_{S^l_1 \subseteq N^l_{=1}} \left|S^l_1\right| \text{ subject to } \sum_{x_k \sim D_e} \sum_{n^l_i \in S^l_1} A(n^l_i, x_k, 1) \geq \tau A^l_{=1} \tag{2}$$

where $0 \leq \tau \leq 1$ (set to 0.95 for experiments) and S^l_1 is the set of neurons we keep at age-1. Notice that the right side of the inequality does not contain the set in the minimization objective. Thus, we replace it with a constant:

$$\min_{S^l_1 \subseteq N^l_{=1}} \left|S^l_1\right| \text{ subject to } \sum_{x_k \sim D_e} \sum_{n^l_i \in S^l_1} A(n^l_i, x_k, 1) \geq T \tag{3}$$

Our goal is to pick a minimum-sized set of neurons S^l_1 to achieve or exceed the target T. We propose the following simple greedy algorithm:

1. Sort all neurons of age-1 in descending order.
2. Pick the neuron with the largest total activation and add it to the set until we reach or exceed the target T.

Proof: We will show that the greedy algorithm finds a solution with a sum of activations at least T, using fewer or the same number of neurons compared to any other optimal algorithm. Consider the optimal solution O consisting of k neurons, and assume that it does not include a particular neuron n^l_i from the top-k activations. Note that if O does include n^l_i, it is the same as the greedy solution. In the scenario where n^l_i is not part of O, we can substitute this neuron n^l_i with one or more neurons of lower or equal activation in the O. Because the neuron we are adding has a larger activation, the sum of activations will either stay the same or increase, while the number of units decreases or stays the same. This enhances the solution's quality, contradicting the assumption of O being optimal.

D. Additional Results

In the main paper, due to space constraints, we omitted some complementary results. Firstly, **Table** 4 offers a more detailed comparison with Replay methods, including standard deviations across multiple runs with different random seeds. Furthermore, **Table** 5 presents the average forgetting after learning has been completed for the top-performing methods (NICE, iCaRL, TAMiL, and x-DER). Following the approach described in [8], we measure forgetting for the j-th episode classes after the model has completed episode $k > j$ as:

$$f^k_j = \max_{l \in \{1,\ldots,k-1\}} a_{l,j} - a_{k,j}, \quad \forall j < k \tag{4}$$

where $a_{l,j}$ is the test accuracy for episode j classes after incrementally training the network from tasks 1 to l. Next, we calculate the average forgetting after the k-th episode as $F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f^k_j$. Lower F_k implies less forgetting on seen classes.

In the following subsections, we delve deeper into the replay baselines, specifically examining the "Forget and Relearn" phenomenon described in the Introduction. Subsequently, we provide a thorough analysis of neuron activations, demonstrating that neurons display distinct activation patterns for familiar and unfamiliar classes, extending this observation to datasets beyond CIFAR10. Finally, we explore alternatives to the logistic regression-based context detector.

D.1. Analyzing the "Forget and Relearn" Phenomenon in Replay Methods

In the Introduction, we discussed the phenomenon where DER and FDR methods completely forget digits 0 and 1 from the MNIST dataset when new classes are encountered and later relearn these digits using replay samples. This section expands that analysis to include various datasets and additional baseline methods such as ER and A-GEM. Our approach involves training models with DER, FDR, ER, and A-GEM, and closely tracking their accuracy on episode-1 classes throughout the training period. To effectively demonstrate the "forget and relearn" behavior, we conduct tests after every training update

Table 4. Accuracy across classes following all episodes on three memory budgets (standard, doubled, and quadrupled): Average accuracies reported across three seeds. Standard deviations are presented in parentheses, previously omitted due to space limitations in the main paper.

	Datasets					
Method	MNIST	Fashion	EMNIST	CIFAR10	CIFAR100	TinyImageNet
Joint	98.4 (±0.0)	86.8 (±0.5)	92.1 (±0.3)	75.8 (±4.7)	41.3 (±0.5)	38.0 (±0.2)
NICE	**83.4** (±0.8)	**73.9** (±0.8)	**66.3** (±0.4)	**55.1** (±0.5)	20.3 (±0.1)	**11.8** (±0.1)
iCaRL	79.7 (±1.8)	68.7 (±0.8)	65.3 (±0.5)	54.2 (±0.6)	23.5 (±0.5)	10.3 (±0.2)
TAMiL	81.1 (±1.2)	68.2 (±2.6)	58.1 (±2.7)	26.8 (±7.3)	16.8 (±1.0)	7.2 (±0.5)
x-DER	77.6 (±0.5)	65.3 (±0.7)	62.4 (±0.6)	37.2 (±1.6)	**24.7** (±0.4)	11.6 (±0.1)
DER	67.6 (±0.7)	65.4 (±1.5)	47.5 (±0.2)	27.3 (±2.5)	13.1 (±0.1)	9.6 (±0.1)
GDumb	60.3 (±7.9)	62.6 (±2.9)	43.8 (±1.8)	30.2 (±1.3)	10.4 (±0.2)	2.7 (±0.1)
ER	66.5 (±0.3)	58.3 (±4.2)	39.0 (±1.0)	23.1 (±2.2)	8.9 (±0.1)	8.8 (±0.1)
FDR	64.3 (±3.2)	52.7 (±4.9)	42.2 (±1.7)	27.5 (±2.3)	9.6 (±0.2)	10.2 (±0.1)
A-GEM	33.9 (±7.9)	42.6 (±3.5)	15.5 (±4.5)	21.2 (±4.3)	7.7 (±0.0)	10.2 (±0.1)
NICE (×2)	**85.9** (±0.9)	**75.1** (±0.8)	**69.4** (±0.3)	**56.5** (±0.6)	22.0 (±0.1)	12.2 (±0.2)
iCaRL (×2)	84.6 (±0.2)	74.5 (±0.6)	67.9 (±0.4)	56.3 (±0.7)	24.6 (±0.3)	13.6 (±0.2)
TAMiL (×2)	84.4 (±1.2)	70.6 (±1.7)	67.4 (±0.3)	34.1 (±2.5)	21.7 (±0.9)	11.6 (±0.2)
x-DER (×2)	85.6 (±0.3)	71.9 (±0.5)	68.6 (±0.5)	42.6 (±1.1)	**25.7** (±0.4)	**14.8** (±0.1)
DER (×2)	80.7 (±0.9)	73.4 (±1.5)	63.6 (±1.1)	35.4 (±0.6)	18.5 (±0.2)	9.8 (±0.1)
GDumb (×2)	73.4 (±1.6)	66.4 (±1.8)	56.7 (±1.0)	37.0 (±1.1)	14.5 (±0.6)	3.8 (±0.1)
ER (×2)	79.7 (±2.2)	66.6 (±1.7)	53.8 (±2.5)	31.5 (±1.8)	12.7 (±0.6)	8.6 (±0.3)
FDR (×2)	75.9 (±1.8)	64.8 (±4.5)	54.0 (±0.3)	35.5 (±1.2)	12.3 (±0.1)	10.1 (±0.1)
A-GEM (×2)	64.8 (±5.6)	45.5 (±5.0)	17.6 (±2.8)	21.9 (±4.0)	7.8 (±0.1)	10.4 (±0.1)
NICE (×4)	87.7 (±0.9)	76.3 (±0.6)	71.7 (±0.4)	57.6 (±0.9)	23.0 (±0.4)	12.3 (±0.1)
iCaRL (×4)	85.8 (±0.3)	76.1 (±0.2)	69.1 (±0.7)	**59.1** (±1.3)	25.8 (±0.8)	15.8 (±0.4)
TAMiL (×4)	**92.7** (±0.5)	77.2 (±0.9)	**78.6** (±0.6)	37.2 (±6.4)	24.8 (±1.3)	13.3 (±0.6)
x-DER (×4)	91.6 (±0.4)	75.8 (±0.7)	74.8 (±0.6)	50.0 (±0.3)	**28.5** (±1.0)	**16.8** (±0.2)
DER (×4)	90.6 (±1.0)	**77.7** (±0.4)	75.3 (±0.8)	41.1 (±1.7)	25.2 (±0.3)	10.0 (±0.1)
GDumb (×4)	86.4 (±0.5)	74.2 (±0.7)	65.8 (±1.1)	41.8 (±0.2)	20.8 (±0.8)	4.7 (±0.5)
ER (×4)	88.2 (±0.3)	74.9 (±0.9)	64.7 (±1.1)	38.7 (±1.6)	17.8 (±0.4)	8.8 (±0.2)
FDR (×4)	84.8 (±1.7)	72.6 (±2.9)	64.8 (±0.7)	45.0 (±0.7)	18.2 (±0.2)	10.1 (±0.1)
A-GEM (×4)	71.6 (±6.1)	44.5 (±3.9)	15.8 (±6.0)	21.1 (±4.3)	7.7 (±0.1)	10.5 (±0.2)
SGD	19.9 (±0.1)	20.0 (±0.0)	6.4 (±1.8)	19.1 (±0.2)	7.2 (±0.1)	10.2 (± 0.4)

during the initial epoch of each new training episode, followed by testing after the completion of each subsequent epoch. We used $\times 4$ memory budget to ensure that we have enough replay samples.

Figure 1 presents results for EMNIST, CIFAR10, and CIFAR100. Across all datasets and methods, we observe the "Forget and Relearn" phenomenon. Each time a new set of classes is introduced, the learner rapidly forgets old classes, followed by a partial recovery in performance. These results bolster our critiques of replay methods across various datasets and baselines. However, the family of replay methods is extensive, and it is possible that some replay methods do not suffer from this issue.

D.2. Relationship Between Activations and Context

In our paper, we discuss how NICE infers context through neuron activations, despite these neurons not being explicitly trained to respond differently to various episode classes. We asserted that certain neurons show stronger activations for learned classes and weaker ones for unfamiliar classes, facilitating the distinction between these two types of classes. In the main paper, due to space constraints, we only presented results for CIFAR10. However, here in **Figure** 2, we replicate the same experiment for MNIST, FashionMNIST, EMNIST, CIFAR100, Tiny ImageNet. Given the extensive number of neurons, our focus for CIFAR100 and Tiny ImageNet is on the top-250 neurons identified by a Random Forest. Specifically, we fit a Random Forest Classifier with 100 decision trees to determine whether neuron activations correspond to episode

Table 5. Forgetting across classes following all episodes on three memory budgets (standard, doubled, and quadrupled). Results are averaged across three seeds. Standard deviations across seeds are presented in parentheses.

	Datasets					
Method	MNIST	Fashion	EMNIST	CIFAR10	CIFAR100	TinyImageNet
NICE	**8.1** ($\pm$0.8)	**14.2** ($\pm$ 0.3)	**15.0** ($\pm$0.4)	**19.1** ($\pm$0.2)	**13.9** ($\pm$0.6)	**13.8** ($\pm$0.2)
iCaRL	12.9 ($\pm$1.9)	21.3 ($\pm$0.9)	16.6 ($\pm$0.1)	29.4 ($\pm$0.9)	17.0 ($\pm$0.4)	18.3 ($\pm$0.1)
TAMiL	11.2 ($\pm$3.1)	19.8 ($\pm$3.1)	20.7 ($\pm$0.8)	64.0 ($\pm$3.2)	40.5 ($\pm$0.5)	31.3 ($\pm$1.1)
x-DER	27.7 ($\pm$0.6)	29.9 ($\pm$6.0)	35.2 ($\pm$1.0)	53.1 ($\pm$5.4)	28.6 ($\pm$0.3)	49.5 ($\pm$0.2)
NICE ($\times$2)	**7.1** ($\pm$0.8)	**13.2** ($\pm$0.4)	**13.9** ($\pm$0.5)	**19.7** ($\pm$0.2)	**13.6** ($\pm$0.6)	**13.2** ($\pm$0.2)
iCaRL ($\times$2)	7.4 ($\pm$0.6)	13.4 ($\pm$0.2)	14.0 ($\pm$0.2)	27.0 ($\pm$0.4)	15.9 ($\pm$0.3)	13.4 ($\pm$0.1)
TAMiL ($\times$2)	12.2 ($\pm$1.3)	21.6 ($\pm$6.0)	14.9 ($\pm$1.5)	60.2 ($\pm$10.1)	31.3 ($\pm$1.6)	44.7($\pm$1.2)
x-DER ($\times$2)	17.4 ($\pm$0.4)	21.5 ($\pm$5.8)	27.1 ($\pm$0.4)	47.1 ($\pm$3.2)	19.8 ($\pm$1.0)	44.1 ($\pm$0.4)
NICE ($\times$4)	6.7 ($\pm$0.1)	12.7 ($\pm$0.7)	12.9 ($\pm$0.4)	**19.5** ($\pm$0.7)	13.8 ($\pm$0.1)	13.9 ($\pm$0.3)
iCaRL ($\times$4)	6.7 ($\pm$0.8)	12.7 ($\pm$1.0)	13.5 ($\pm$0.2)	24.4 ($\pm$1.5)	14.4 ($\pm$0.3)	**11.6** ($\pm$0.3)
TAMiL ($\times$4)	**5.5** ($\pm$0.6)	17.5 ($\pm$0.7)	**11.5** ($\pm$0.5)	55.8 ($\pm$7.5)	24.4 ($\pm$2.0)	40.1 ($\pm$1.9)
x-DER ($\times$4)	7.9 ($\pm$2.9)	**10.5** ($\pm$3.5)	20.6 ($\pm$0.6)	35.5 ($\pm$0.2)	**12.3** ($\pm$0.5)	40.1 ($\pm$0.1)

Table 6. Context detector accuracy (%) with different heuristics.

Heuristics	Fashion	EMNIST	C10	C100
Logistic Regression	74.9	66.5	**57.7**	**28.4**
MLP-100	75.8	68.7	57.4	27.4
RF-100	**77.2**	**69.2**	55.3	27.5
Nearest Hamming	70.3	67.1	30.7	21.8
Ave. Hamming	30.3	33.2	30.1	10.8
Random	20	7.6	20.1	9.8

one, utilizing the feature importance from the Random Forest to select the top-250 neurons. For smaller scale experiments (MNIST, FashionMNIST, and EMNIST), we include all neurons but order them based on importance to enhance clarity in visualization. These additional results support our observation on CIFAR10.

D.3. Context Detector Ablation Experiment

We conducted an additional experiment, comparing chained logistic regressions to various heuristics. **Table** 6 shows results using standard memory budget. Nearest and average Hamming distances determine context by using the closest memory entry and the mean distance to episode entries, respectively. We also tested chained MLPs (single 100-unit hidden layer) and Random Forests (100-tree). The findings indicate that while complex heuristics may outperform on some datasets, chained logistic regression is an efficient, and robust choice.

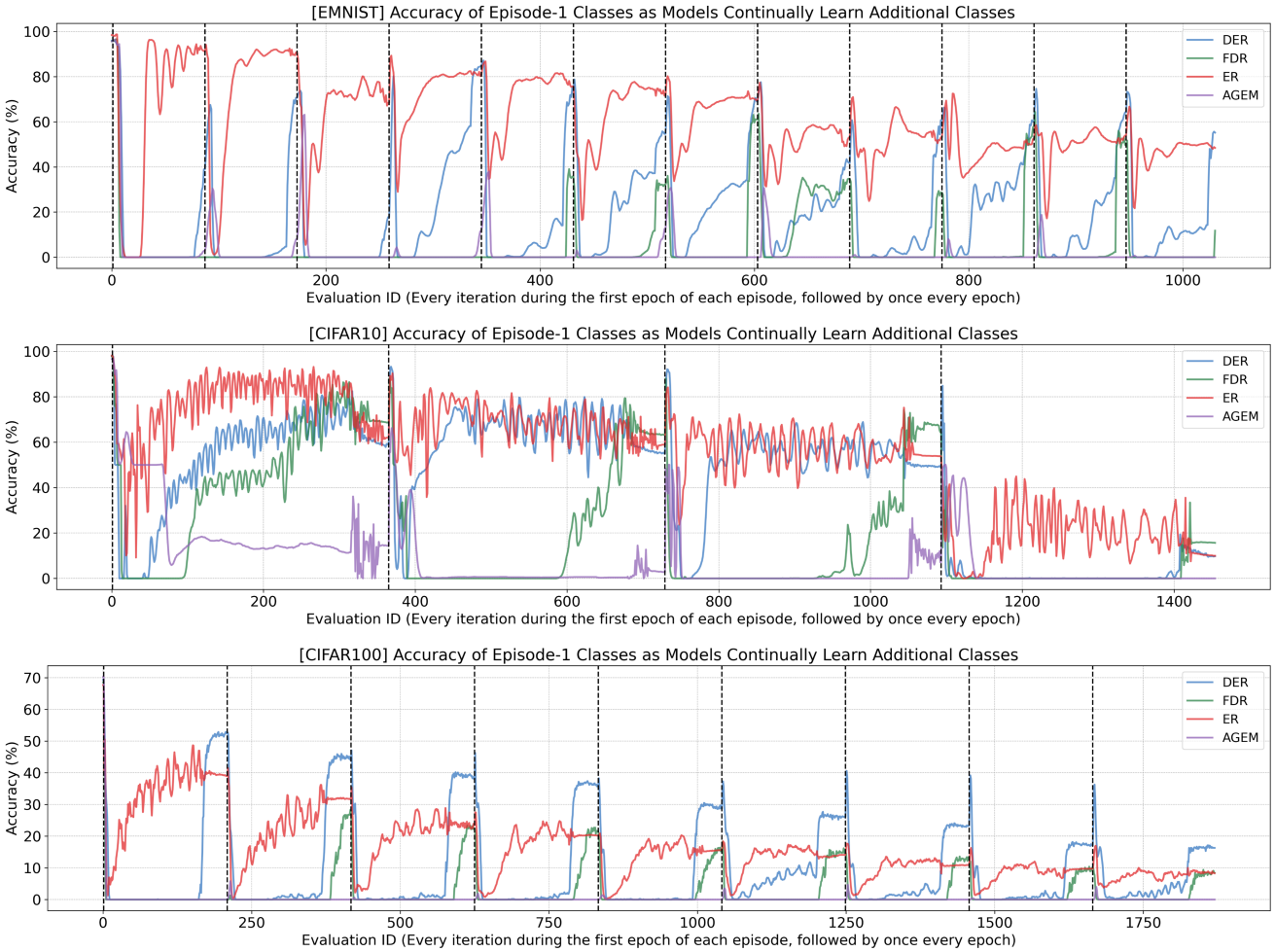

Figure 1. DER, FDR, ER, and A-GEM are trained on EMNIST (row 1), CIFAR10 (row 2), and CIFAR100 (row 3), with vertical dashed lines indicating the start of a new episode. Note that the x-axis shows evaluation IDs rather than training iterations. We run evaluations after every update during the first epoch of each episode, followed by one evaluation after each subsequent epoch.

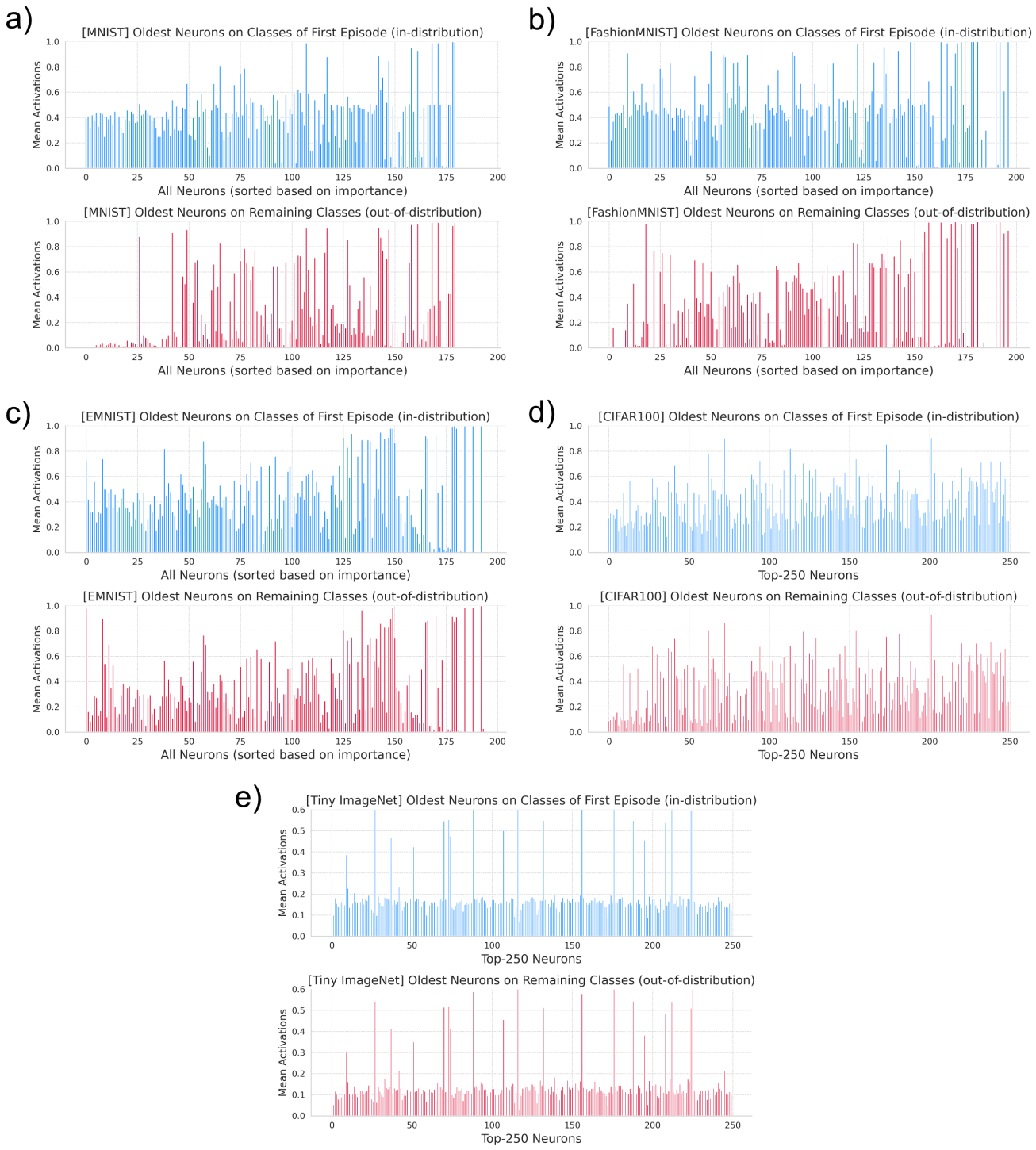

Figure 2. Average activations of the neurons for Episode-1 classes (blue, top) and for the remaining classes (red, bottom) are shown. a) MNIST, b) FashionMNIST, c) EMNIST, d) CIFAR100, e) Tiny ImageNet.

E. From Neuroscience to Deep Learning

Machine learning has already gained much from the emulation of basic features of neural circuits. While the brain is too complex to recreate fully, there are fundamental principles at work which may be applied to machine learning. NICE aims to provide stable memory capabilities inspired by neuroscience observations of context-dependent network partitioning in the hippocampus. However, NICE is not intended to be a model of the brain, but instead a machine learning algorithm loosely inspired by the functioning of the brain. In this section, we draw on neuroscience literature to compare and contrast the structure of NICE with the hippocampus. More specifically, we highlight (1) neurogenesis in the dentate gyrus and (2) contextual discrimination between old and young neurons as key inspirations from hippocampal memory encoding processes.

E.1. Biological Underpinnings of NICE

Of all the brain regions, the hippocampus has long been theorized to play a central role in memory encoding [21, 26, 29, 30]. In corroboration with this, hippocampal-lesioned patients are unable to form new declarative memories (e.g., events or historical episodes). Thus, the hippocampus represents a uniquely important brain region for the study of memory encoding and for the research into memory mechanisms in general.

E.1.1 Dentate Gyrus Neurogenesis

Within the hippocampus, one region, the dentate gyrus (DG), displays a unique ability to produce new neurons during adulthood, being one of a few brain regions to do so [11, 12, 14, 17, 27]. This continual turnover of DG neurons has been proposed to serve an important role in the ability of the hippocampus to encode new memories throughout adulthood [2, 9, 11, 13, 14, 17, 18, 27]

The majority of the neurons of the dentate gyrus, and the neurons which are continually grown in adulthood, are known as granule cells (GCs). These neurons are developmentally stacked upon each other so that older neurons are located deeper (towards the stratum oriens), and younger neurons are packed on top of these older neurons (towards the stratum moleculare) [1, 21]. The dendrites (analogous to input connections in machine learning) of older GCs are more extensive and receive more synaptic connections than their younger counterparts.

In the dentate gyrus, newborn granule cells undergo extended maturation through developmental stages driven by specific transcription factors [11, 14]. A substantial portion of newborn GCs die within the first couple weeks of their birth. This death occurs in two waves, one when the neurons are first born and one when synaptic connections begin to form. At this second stage, only neurons which are able to receive contextual input survive [11]. These selected neurons then go through a maturational stage from 4-6 weeks where they exhibit increased synaptic plasticity [2, 9, 11, 13]. Beyond this point, their threshold for activation and their input specificity begin to increase as they become immature granule cells. Thus, synaptic plasticity in the dentate gyrus is largely confined to a subset of newly generated neurons. This characteristic corresponds to NICE's use of a developmental subset to learn synaptic associations, as described in section 3.2. A natural consequence of different generations of neurons growing and maturing together is that these neurons will become interconnected [10]. This functional feature corresponds to NICE's wiring of same-age neurons together, as described in section 3.2.

As immature GCs age, they gradually integrate into their surroundings as they become functionally coupled into excitatory and inhibitory subnetworks. Immature and mature GCs in the dentate gyrus are heavily inhibited relative to newborn neurons. This heavy direct inhibition can be interpreted as a mechanism for network partitioning [5, 20, 31]. This characteristic corresponds to NICE's use of thresholded network activity to inhibit network inference, as described in section 3.4. Fully mature neurons contribute to recognition of known patterns, i.e., pattern completion, unlike the learning capabilities provided by newborn neurons [2, 17, 18].

In NICE, we approximate the continuous process of neural development by categorizing neurons into three discrete maturation stages: progenitor (age-0), immature (age-1), and mature neurons (age-2 and older), corresponding to newborn, immature, and mature granule cells, respectively. Furthermore, while mature neurons have greatly reduced synaptic potentiation, we approximate this by freezing connections to mature neurons.

E.1.2 Hippocampal Network Context

In neuroscience, the term "context" has many connotations, as it can refer to the application of prior knowledge to current processing [3], as well as the identification of environments [15] and features [32]. One definition for context is neuron-level input which modulates neuron excitability, not to be confused with feed-forward driving input which directly causes neuron activity [16]. Anatomically, driving input and neural context correspond to synaptic connections onto proximal and

distal dendrites, respectively. Moreover, we observe that neurogenesis allows for an inherent form of network context in that early-born neurons drive network activity while late-born neurons are context-modulated [4, 6, 23, 24]. This network context inspires NICE's use of network activity to inform network partitioning along developmental boundaries.

In the brain, a network's ability to differentiate between contexts relies on developmental differences between subpopulations of neurons [2, 4, 6, 7, 10, 17, 18, 23, 24, 28]. Early-born hippocampal neurons are heavily interconnected and play a driving role in stable ensemble formation and the processing of familiar environments [4, 10, 23–25]. These early-born neurons are active at regular locations in the environment, forming a stable grid of activity that "remaps" upon changes in environment or context [15, 17, 32]. Middle-born neurons are moderately connected and heavily inhibited, facilitating pattern separation, i.e., the orthogonalization of similar inputs, via increased sparsity [4, 11]. This inhibitory activity can be interpreted as a subnetwork which learns to partition excitatory network activity [4, 5, 20, 31]. Late-born hippocampal neurons facilitate contextual discrimination and pattern separation via sparse and plastic synapses [2, 6, 9, 11, 13, 17, 18, 27, 32].

In both cases, developmentally differentiated neuron subpopulations play distinct roles in memory encoding processes, with the older, high-activity neurons retaining past knowledge and younger, low-activity neurons learning new associations. As older neurons are more heavily interconnected, they are more likely to respond to sensory changes corresponding to novelty [5]. This is akin to how NICE uses thresholded unit activity to identify an in-distribution subnetwork, as discussed in section 4.2 (see figure 5). While the strong interconnectivity of old neurons provides them with stability, it means that stimuli may not be uniquely represented since multiple stimuli may engender the same pattern of activity [2, 18, 23]. In contrast, the sparse connectivity of low-activity neurons means they can encode different features. This is akin to how NICE recruits low-activity units to encode new contexts.

Thus, in the brain, contextual factors underlie the persistent storage of information. In the dentate gyrus, the growth of new neurons supports contextual discrimination throughout adulthood. Specifically, neural pruning and neurogenesis in the dentate gyrus represent the refreshing of memory capacity to allow for new learning. This developmental learning is informed by excitatory and inhibitory "contextual input" which determines the allocation of network resources per a specific scenario. NICE takes inspiration from these neuroscientific observations to design a method for continual learning which does not inherently rely on replay but instead modulates unit activity based on network context to prevent catastrophic forgetting.

E.2. Biological Plausibility of NICE

Modeling a network as interconnected and complicated as the mammalian brain requires the distillation of biologically complex processes to functional features that can be grossly approximated by ML techniques. Naturally, these simplifications fail to capture all of the functionality and efficiency of neurological circuitry. The aim of NICE is not to model the brain, but instead to create a useful algorithm based on principles inspired by the brain. Therefore, in this section, we discuss the limitations of NICE with regards to neuroscience.

As mentioned, context detection in the brain is the result of an interplay of specialized developmental populations. However, in NICE neurons of the same age are wired together, but unit differentiation is limited to the developmental association of synaptic connectivity. Therefore, NICE draws a direct link between neurogenesis and context detection, whereas in the brain these processes are in fact decoupled. More specifically, context detection in the hippocampus relies on an interplay of stable activity driven by older neurons and dynamic learning provided by younger neurons [6, 17, 18, 23, 27]. This process is fundamentally facilitated by developmental differentiation, but is also a product of the inputs that the hippocampus receives [14, 19, 22] and the neural circuitry which has evolved to enable context detection [3, 24, 28].

Most of the literature on the hippocampus focuses on spatial navigation [3, 23, 25], and not on its potential role in image classification [29]. This is in part due to the practical nature of experimental paradigms studying spatial navigation. Therefore, more work needs to be done to understand how hippocampal processes could be involved in non-locomotor tasks such as image recognition. Nevertheless, NICE presumes that image classification tasks can be addressed in an equivalent manner to spatial navigation tasks in that network activity is leveraged for context detection. However, this simplification ignores the differential processing of separate sensory modalities within the hippocampus [7, 19, 23, 24, 28, 30].

Another limitation of NICE lies in the distinction between neural context and network context. Since NICE aggregates information from across the network to determine episodic identity, it represents a form of network context similar to the dynamic between low-activity and high-activity neurons of the hippocampus. However, NICE excludes a formulation of neural context, meaning that individual units do not separately modulate their inputs, unlike neurons in the brain [3, 16]. On a more abstract level, this represents the implementation of context solely based on proximal feed-forward flow. However, the context detector in NICE allows for information from downstream regions of the network to partition upstream network activity, serving as a form of contextual feedback. Therefore, NICE employs a mechanism for neural cooperation insofar as the context detector operates on a global network signal to partition local network activity.

As described in section 3.3, older neurons in NICE have their connections from younger neurons pruned. This connection pruning was done to prevent catastrophic forgetting from the updating of prior knowledge with new learning. However, this does not preserve the biological characteristic of older neurons having more incoming connections then their younger counterparts as outlined above. Were a mechanism for unit context to be integrated into NICE, it is conceivable that older neurons could receive connections from younger neurons given that their prior knowledge may be preserved via different contexts. This would allow for prior knowledge to be leveraged for future tasks (old-to-new), as NICE already incorporates, but also for new learning to potentially shape prior knowledge (new-to-old) in an informed manner.

References

[1] David G. Amaral, Helen E. Scharfman, and Pierre Lavenex. The dentate gyrus: fundamental neuroanatomical organization (dentate gyrus for dummies). In *The Dentate Gyrus: A Comprehensive Guide to Structure, Function, and Clinical Implications*, pages 3–790. Elsevier, 2007. 9

[2] Diego M. Arribas, Antonia Marin-Burgin, and Luis G. Morelli. Adult-born granule cells improve stimulus encoding and discrimination in the dentate gyrus. *bioRxiv*, 2022. 9, 10

[3] Jaan Aru, Moritz Drüke, Juhan Pikamäe, and Matthew Larkum. Mental navigation and the neural mechanisms of insight. *Trends in Neurosciences*, 46, 2022. 9, 10

[4] Hugo Balleza-Tapia, Luis Enrique Arroyo-García, Arturo G. Isla, Raúl Loera-Valencia, and André Fisahn. Functionally-distinct pyramidal cell subpopulations during gamma oscillations in mouse hippocampal area ca3. *Progress in Neurobiology*, 210:102213, 2022. 10

[5] Mia Borzello, Steve Ramirez, Alessandro Treves, Inah Lee, Helen Scharfman, Craig Stark, James J. Knierim, and Lara M. Rangel. Assessments of dentate gyrus function: discoveries and debates. *Nature Reviews Neuroscience*, 24:502–517, 2023. 9, 10

[6] Davide Cavalieri, Alexandra Angelova, Anas Islah, Catherine Lopez, Marco Bocchio, Agnès Baude, and Rosa Cossart. Ca1 pyramidal cell diversity is rooted in the time of neurogenesis. *bioRxiv*, 2021. 10

[7] Mark Cembrowski and Nelson Spruston. Heterogeneity within classical cell types is the rule: lessons from hippocampal pyramidal neurons. *Nature Reviews Neuroscience*, 20, 2019. 10

[8] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 4

[9] John Darby Cole, Delane F. Espinueva, Désirée R. Seib, Alyssa M. Ash, Matthew B. Cooke, Shaina P. Cahill, Timothy P. O'Leary, Sharon S. Kwan, and Jason S. Snyder. Adult-born hippocampal neurons undergo extended development and are morphologically distinct from neonatally-born neurons. *Journal of Neuroscience*, 40(30):5740–5756, 2020. 9, 10

[10] Rosa Cossart and Roustem Khazipov. How development sculpts hippocampal circuits and function. *Physiological Reviews*, 102(1): 343–378, 2022. 9, 10

[11] Annina Denoth and Sebastian Jessberger. Formation and integration of new neurons in the adult hippocampus. *Nature Reviews Neuroscience*, 22:1–14, 2021. 9, 10

[12] P. S. Eriksson, E. Perfilieva, T. Björk-Eriksson, A. M. Alborn, C. Nordborg, D. A. Peterson, and F. H. Gage. Neurogenesis in the adult human hippocampus. *Nature Medicine*, 4(11):1313–1317, 1998. 9

[13] Shaoyu Ge, Chih-Hao Yang, Kuei-Sen Hsu, Guo-Li Ming, and Hongjun Song. A critical period for enhanced synaptic plasticity in newly generated neurons of the adult brain. *Neuron*, 54(4):559–566, 2007. 9, 10

[14] Gerd Kempermann, Fred Gage, Ludwig Aigner, Hongjun Song, Maurice Curtis, Sandrine Thuret, Georg Kuhn, Sebastian Jessberger, Paul Frankland, Heather Cameron, Elizabeth Gould, Rene Hen, Djoher Abrous, Nicolas Toni, Alejandro Schinder, Xinyu Zhao, Paul Lucassen, and Jonas Frisén. Human adult neurogenesis: Evidence and remaining question. *Cell Stem Cell*, 23, 2018. 9, 10

[15] John L Kubie, Eliott R J Levy, and André A Fenton. Is hippocampal remapping the physiological basis for context? *Hippocampus*, 30(8):851–864, 2020. 9, 10

[16] Matthew Larkum, J. Zhu, and Bert Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398:338–41, 1999. 9, 10

[17] Matías Mugnaini, Mariela F. Trinchero, Alejandro F. Schinder, Verónica C. Piatti, and Emilio Kropff. Unique potential of immature adult-born neurons for the remodeling of ca3 spatial maps. *Cell Reports*, 42(9):113086, 2023. 9, 10

[18] Toshiaki Nakashiba, Jesse Cushman, Kenneth Pelkey, Sophie Renaudineau, Derek Buhl, Thomas Mchugh, Vanessa Rodriguez Barrera, Ramesh Chittajallu, Keisuke Iwamoto, Chris McBain, Michael Fanselow, and Susumu Tonegawa. Young dentate granule cells mediate pattern separation, whereas old granule cells facilitate pattern completion. *Cell*, 149:188–201, 2012. 9, 10

[19] Eirik S. Nilssen, Thanh P. Doan, Maximiliano J. Nigro, Shinya Ohara, and Menno P. Witter. Neurons and networks in the entorhinal cortex: A reappraisal of the lateral and medial entorhinal subdivisions mediating parallel cortical pathways. *Hippocampus*, 29(12): 1238–1254, 2019. 10

[20] Kenneth A. Pelkey, Ramesh Chittajallu, Michael T. Craig, Ludovic Tricoire, Jason C. Wester, and Chris J. McBain. Hippocampal gabaergic inhibitory interneurons. *Physiological Reviews*, 97(4):1619–1747, 2017. 9, 10

[21] Edmund T. Rolls. A theory of hippocampal function in memory. *Hippocampus*, 6(6):601–620, 1996. 9

[22] Maximilian Schlecht, Maanasa Jayachandran, Gabriela E. Rasch, and Timothy A. Allen. Dual projecting cells linking thalamic and cortical communication routes between the medial prefrontal cortex and hippocampus. *Neurobiology of Learning and Memory*, 188: 107586, 2022. 10

[23] Farnaz Sharif, Behnam Tayebi, György Buzsáki, Sébastien Royer, and Antonio Fernandez-Ruiz. Subcircuits of deep and superficial ca1 place cells support efficient spatial coding across heterogeneous environments. *Neuron*, 109(2):363–376.e6, 2021. 10

[24] Ivan Soltesz and Attila Losonczy. Ca1 pyramidal cell diversity enabling parallel information processing in the hippocampus. *Nature Neuroscience*, 21:484–493, 2018. 10

[25] Marielena Sosa, Anna Gillespie, and Loren Frank. *Neural Activity Patterns Underlying Spatial Coding in the Hippocampus*. 2016. 10

[26] E. Tulving and H. J. Markowitsch. Episodic and declarative memory: role of the hippocampus. *Hippocampus*, 8(3):198–204, 1998. 9

[27] Sebnem Nur Tuncdemir, Clay Orion Lacefield, and Rene Hen. Contributions of adult neurogenesis to dentate gyrus network activity and computations. *Behavioural Brain Research*, 374:112112, 2019. 9, 10

[28] Sebnem N. Tuncdemir, Andres D. Grosmark, Gergely F. Turi, Amei Shank, John C. Bowler, Gokhan Ordek, Attila Losonczy, Rene Hen, and Clay O. Lacefield. Parallel processing of sensory cue and spatial information in the dentate gyrus. *Cell Reports*, 38(3): 110257, 2022. 10

[29] Nicholas B. Turk-Browne. The hippocampus as a visual area organized by space and time: A spatiotemporal similarity hypothesis. *Vision Research*, 165:123–130, 2019. 9, 10

[30] Niels van Strien, Natalie Cappaert, and Menno Witter. The anatomy of memory: An interactive overview of the parahippocampal-hippocampal network. *Nature reviews. Neuroscience*, 10:272–82, 2009. 9, 10

[31] Bert Vancura, Tristan Geiller, and Attila Losonczy. Organization and plasticity of inhibition in hippocampal recurrent circuits. *bioRxiv*, 2023. 9, 10

[32] Xinyu Zhao, Yingxue Wang, Nelson Spruston, and Jeffrey Magee. Membrane potential dynamics underlying context-dependent sensory responses in the hippocampus. *Nature Neuroscience*, 23:1–11, 2020. 9, 10