REMIND Your Neural Network to Prevent Catastrophic Forgetting

Tyler L. Hayes (PhD Student)
Rochester Institute of Technology
Email: tlh6792@rit.edu
Web: https://tyler-hayes.github.io

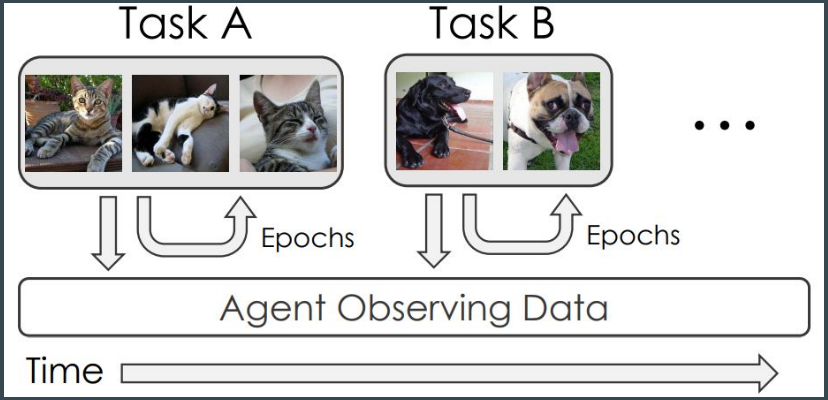
Continual AI Talk: May 29, 2020

Outline

- Incremental Learning Paradigms
- Replay to Mitigate Forgetting
- REMIND Architecture
- Experiments on Image Classification
- Experiments on Visual Question Answering
- Conclusions

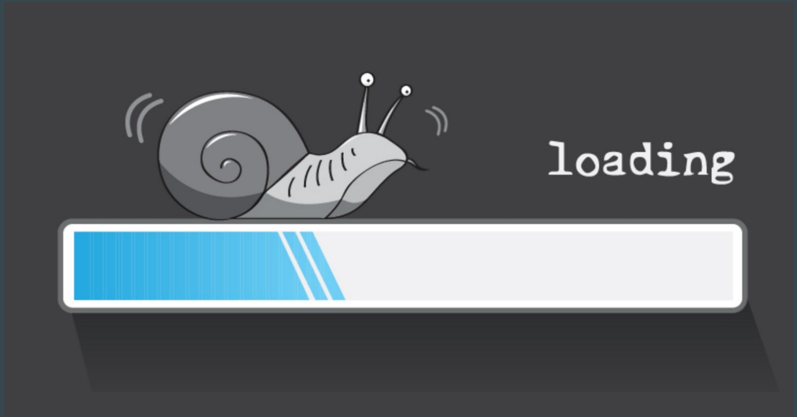
Incremental Batch Learning

- Dataset is broken into several batches
- At each time-step, the learner...
  - Receives a batch of data from one or more classes
  - Loops over the batch until learned
  - Is evaluated at the end of training the batch
- **Advantages:**
  - Recently demonstrated much success
  - Makes learning easier since batches are iid



Incremental Batch Learning

- **Caveats:**
  - It is slow...
    - Must wait for data batch to accumulate before learning
    - Looping makes learning time consuming
    - Cannot evaluate until a batch is learned
  - Batches often consist of multiple classes, reducing the problem to iid learning



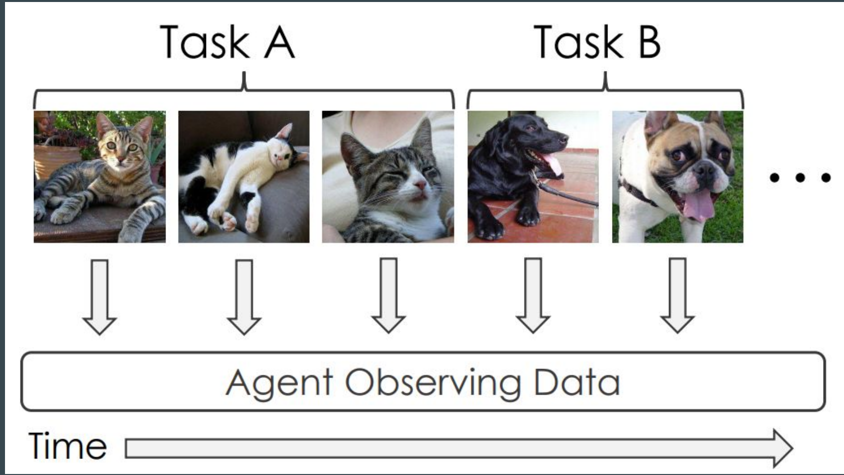
Why Doesn't Incremental Batch Learning Suffice?

- Real-world applications require agents to learn and evaluate new information immediately
- **Examples:** robotics/embedded agents, home appliances, web agents



Online Streaming Learning

- Instances often have temporal correlations and are non-iid, e.g., videos
- At each time-step, the learner...
  - Receives one new sample
  - Learns the sample and then is evaluated
- The learner is only allowed one loop through the entire dataset



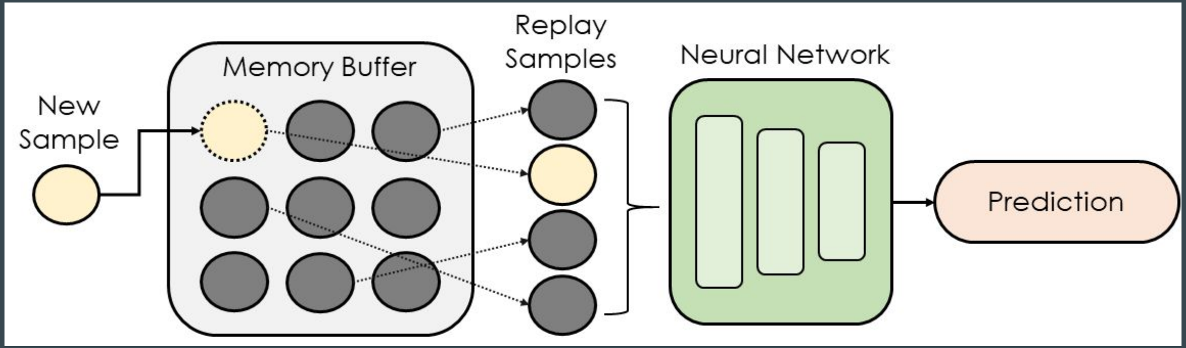
Online Streaming Learning

- **Advantages:**
  - Closer to how humans/animals learn
  - New instances are learned immediately and the agent can be evaluated immediately
  - Better suited for real-time applications

But what about catastrophic forgetting?

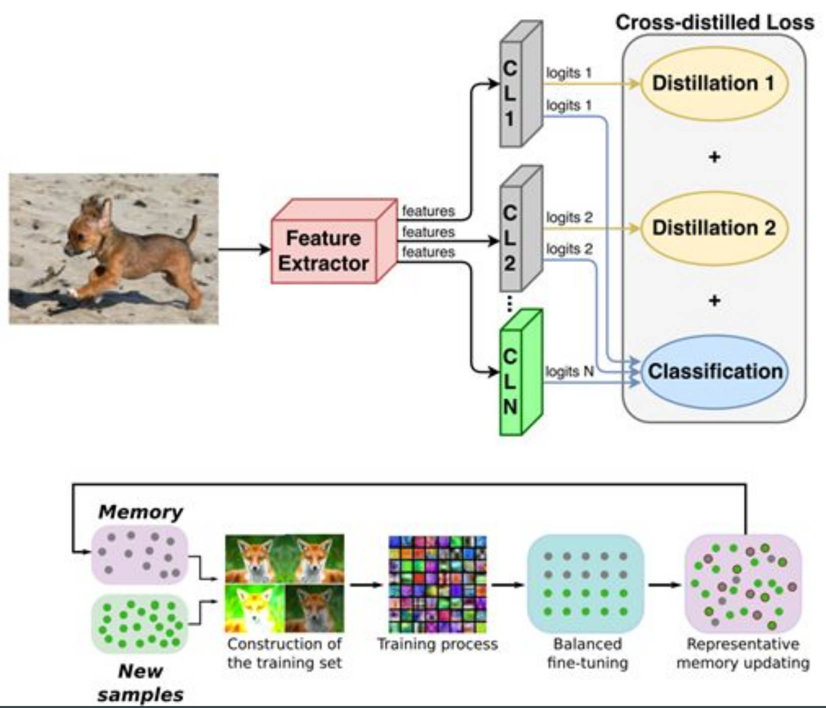
Replay in Artificial Neural Networks

- **Replay (rehearsal)** is one of the most effective methods for mitigating forgetting in neural networks
- Involves storing previous examples in a buffer and mixing new instances with old ones to fine-tune the network



Replay in Artificial Neural Networks

- Researchers have been upgrading replay to maximize efficiency
  - Store *N* raw images, e.g., 100 per class, for representation learning
  - Improve performance by using “soft” targets (**distillation**) and **data augmentation**
- But storing raw pixel images in a buffer is memory intensive and not biologically plausible



Castro et al. (ECCV-2018)

Gaps Between Biological and Artificial Replay

1. Hippocampal indexing theory* postulates that the hippocampus stores **compressed representations** of neocortical activity patterns, which are reactivated during consolidation
    a. Visual inputs are high in the visual processing hierarchy, e.g., not raw pixel representations
2. Animals perform immediate **online streaming learning** from non-iid experiences

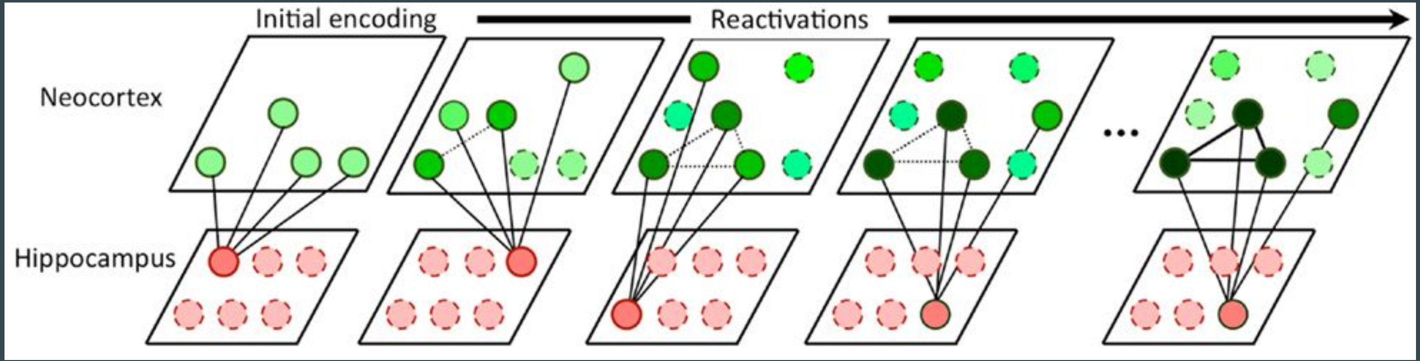


Image: Yassa and Reagh, 2013

*Teyler, T. J., & Rudy, J. W. (2007). The hippocampal indexing theory and episodic memory: updating the index. Hippocampus.

REMIND Your Neural Network to Prevent Catastrophic Forgetting

Tyler L. Hayes*, Kushal Kafle*, Robik Shrestha*, Manoj Acharya, & Christopher Kanan

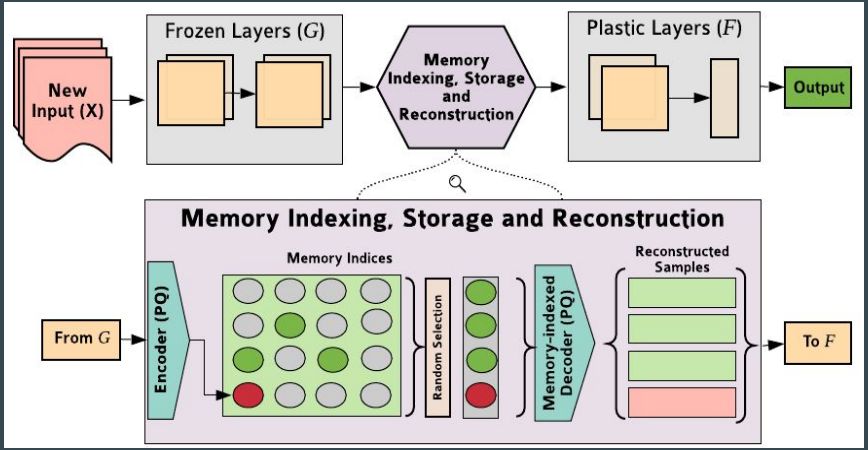
* denotes equal contribution

Available on arXiv: https://arxiv.org/abs/1910.02509

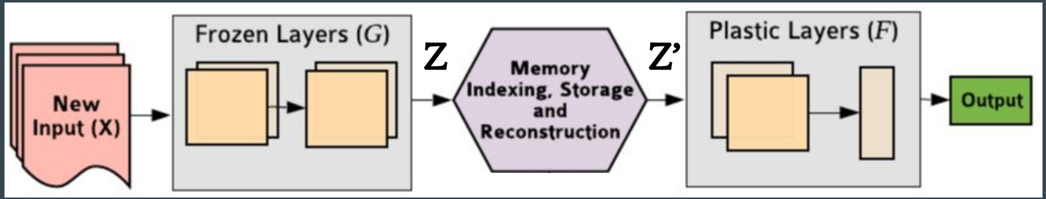
REMIND: **Re**play using **M**emory **Ind**exing

Perform brain-inspired streaming learning by:

- **Key Idea 1:** Store compressed representations of intermediate layers instead of raw inputs (e.g., CNN feature maps) for replay
  - Implement **hippocampal indexing theory** using tensor quantization
- **Key Idea 2:** Control stability-plasticity by freezing part of the network
  - Only **update high visual processing** areas



REMIND Components



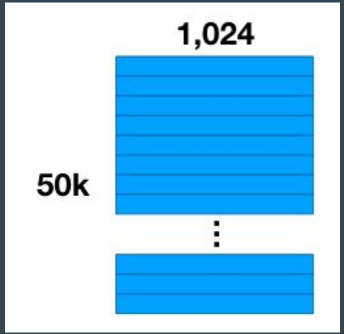
- Inputs are images denoted by X
- Main network is decomposed into **two sub-networks**
  - G consists of the early layers of the network and will be frozen during incremental training
  - F consists of the later layers of the network and will be plastic during incremental training
  - $Output = F(G(X))$
- CNN feature maps extracted from G are denoted by Z
  - $Z = G(X)$
- A **product quantizer** will be trained to encode Z as a compressed representation and decode the compressed representation as Z'

Before we talk about REMIND, we need to talk about Product Quantization*...

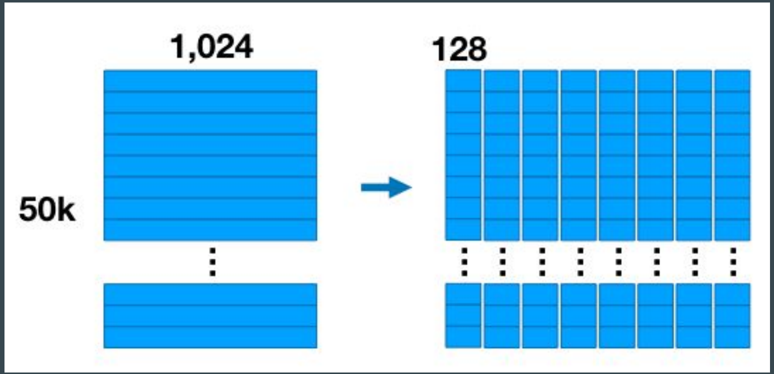
*Jegou, H., Douze, M., & Schmid, C. (2010). Product quantization for nearest neighbor search. IEEE TPAMI.

Preliminaries: Product Quantization

- Let us assume that we have **50k vectors** that we can use to train a product quantization model
  - Each vector has **1024 dimensions**
- We would like to train a product quantization model that has **8 codebooks**
- First, divide our 50k vectors into 8 sub-vectors, each of size 128
  - 1024 / 8 = 128



Original Data

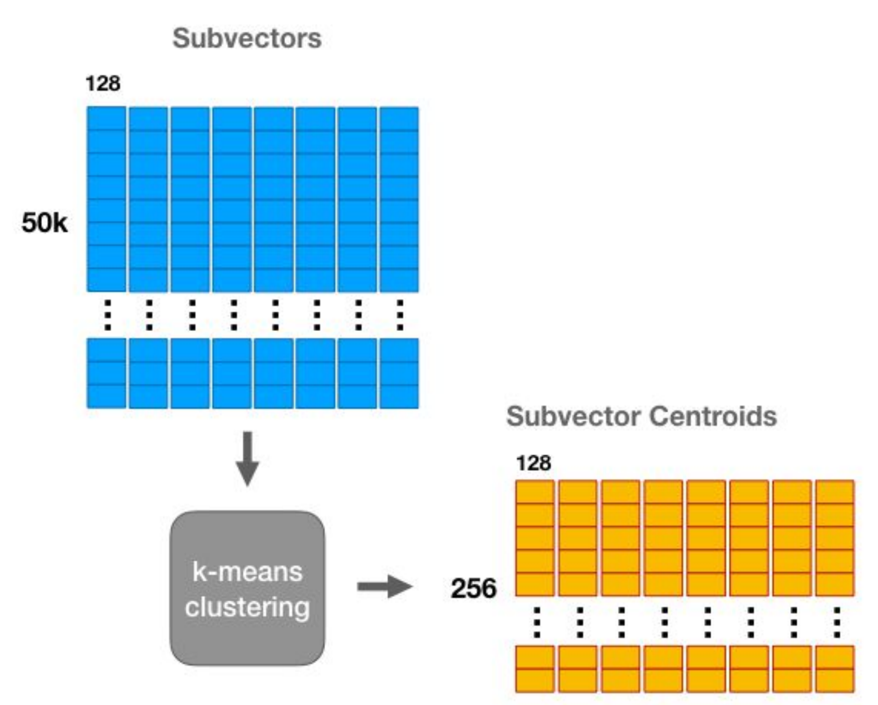


Data Sub-Vectors

Images from Chris McCormick

Preliminaries: Product Quantization

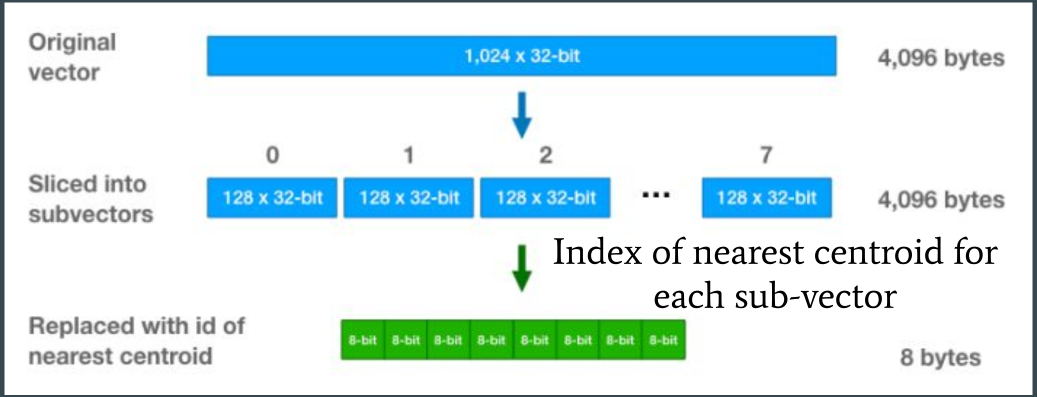
- Next, we perform *k*-means clustering 8 times to train our 8 codebooks, each of size 256
- Now our product quantizer has been trained!
  - Yellow sub-vector centroids are our codebooks



Images from Chris McCormick

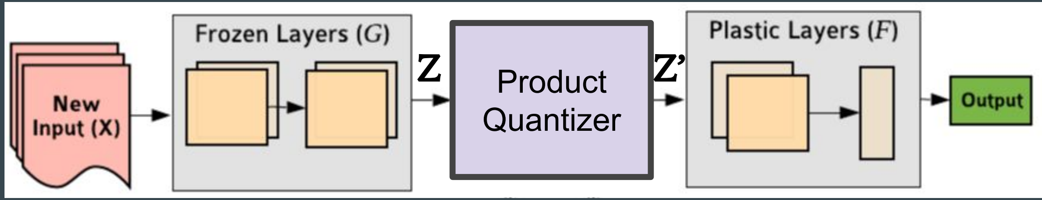
Preliminaries: Product Quantization

- Given a new 1024-dimensional vector
- Partition vector into 8 sub-vectors
- For each of the 8 sub-vectors, find its nearest centroid in the associated codebook and record the index of the centroid
- Now our vector can be represented by just 8 indices and a set of codebooks!



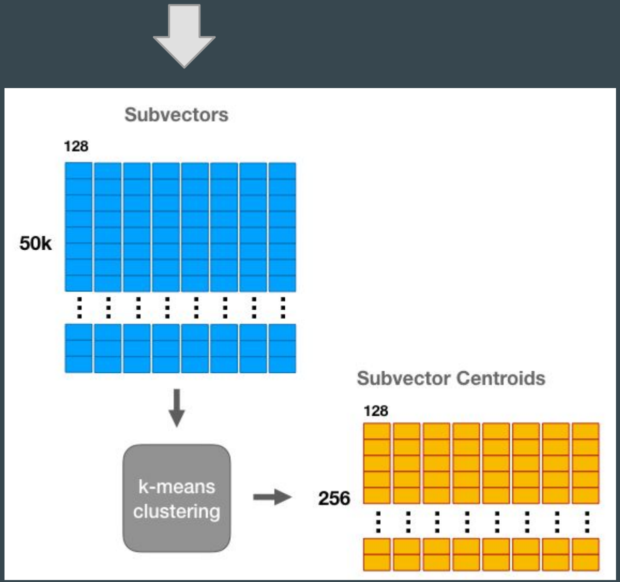
Images from Chris McCormick

How is REMIND Trained?



Stage 1: Base Initialization

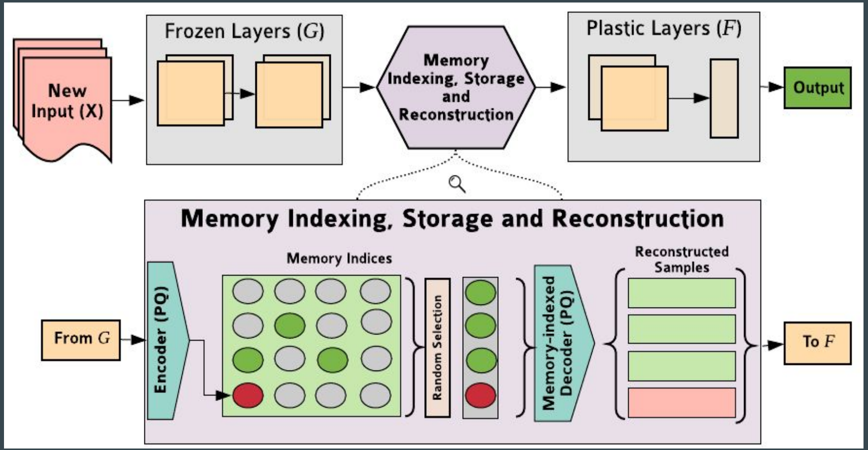
1. Train entire network **offline** (*F* and *G*) on a subset of the dataset (*X*) to initialize lower visual features
2. Push subset of data through frozen layers (*G*) of network and extract mid-level CNN maps for base init (*Z*)
3. Train product quantization model on base init CNN maps
4. Save codebook and memory indices of all data with ground truth labels



How is REMIND Trained?

Stage 2: Streaming Learning

1. Pass new data through “Frozen Layers” of CNN (G) to get feature map
2. Quantize feature map and store as indices relating to PQ codebook
3. Reconstruct subset of old samples from buffer and mix with current reconstructed example
4. Train the “Plastic Layers” of CNN (F) for one iteration



Classification Experiments

Datasets

- We evaluate performance on:
  - **ImageNet ILSVRC-2012**: 1,000 classes with ~1.28 million images
  - **CORe50:** 10 classes with 6,000 images
- For CORe50, we compare **four orderings** of the dataset

Example CORe50 Videos

iid

class iid

instance

class instance

Images from Lomonaco and Maltoni (CoRL-2017)

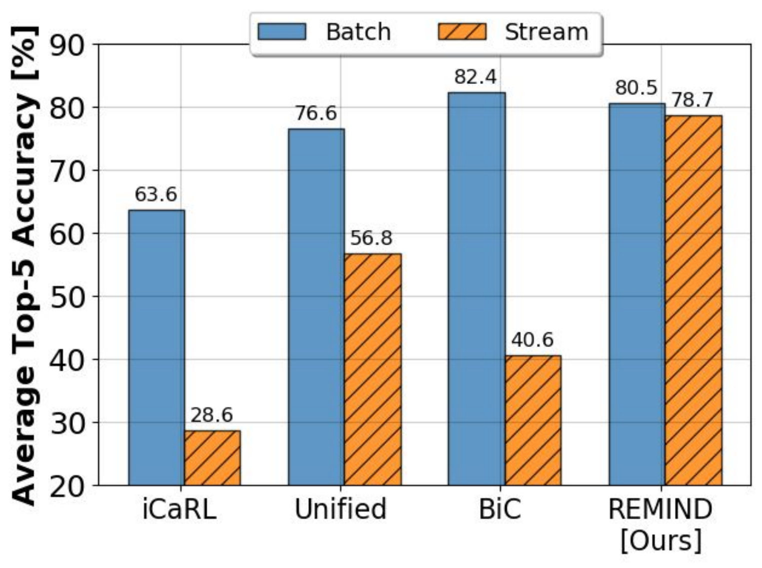
Comparison Models

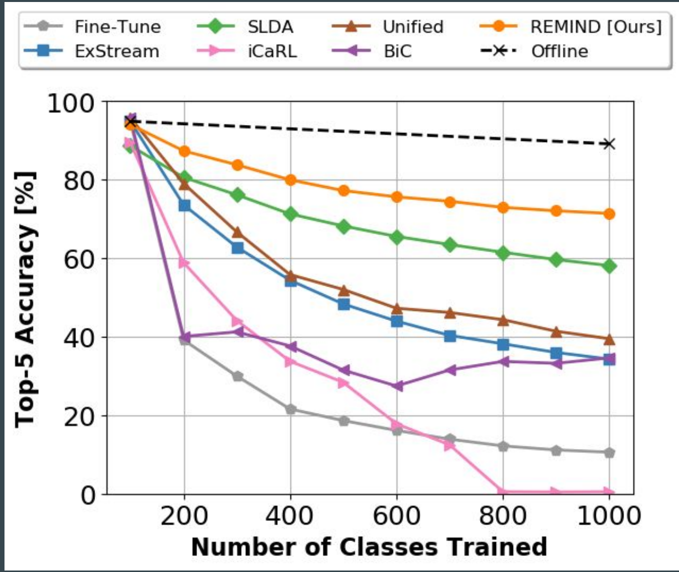
- We compare several **single-headed** baselines (i.e., not task aware):
  - **REMIND**: uses PQ and replay augmentation on quantized features
  - **SLDA** (Hayes & Kanan, CVPRW-2020): combination of a CNN with streaming LDA
  - **ExStream** (Hayes et al., ICRA-2019): uses stream clustering and replay
  - **iCaRL** (Rebuffi et al., CVPR-2017): combines replay with a distillation loss and nearest class mean classifier
  - **Unified Classifier** (Hou et al., CVPR-2019): extends iCaRL by using a cosine normalization layer, a class geometry preservation loss, a margin ranking loss, and a neural network classifier
  - **BiC** (Wu et al., CVPR-2019): extends iCaRL by training two bias correction parameters on the output layer and using a neural network classifier
  - **Offline (upper bound)**: an optimized offline learner

Stream

Batch

ImageNet ILSVRC-2012 Results



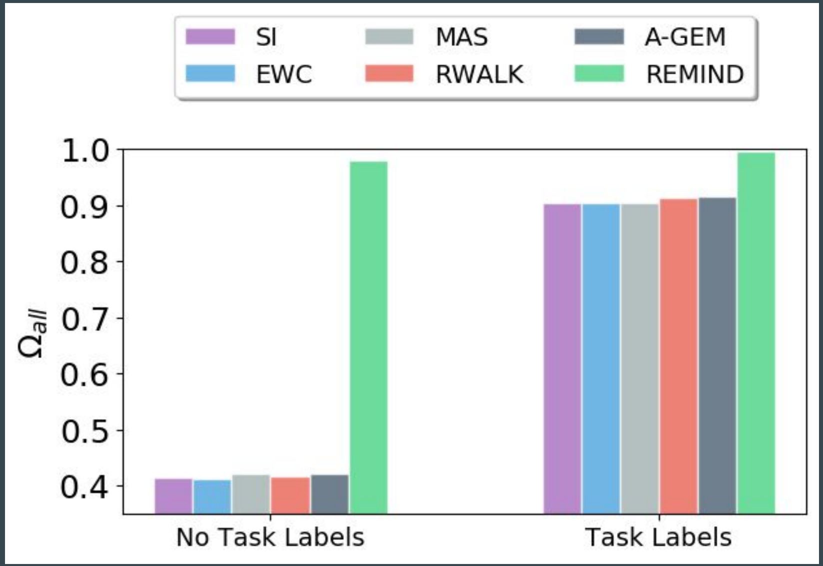


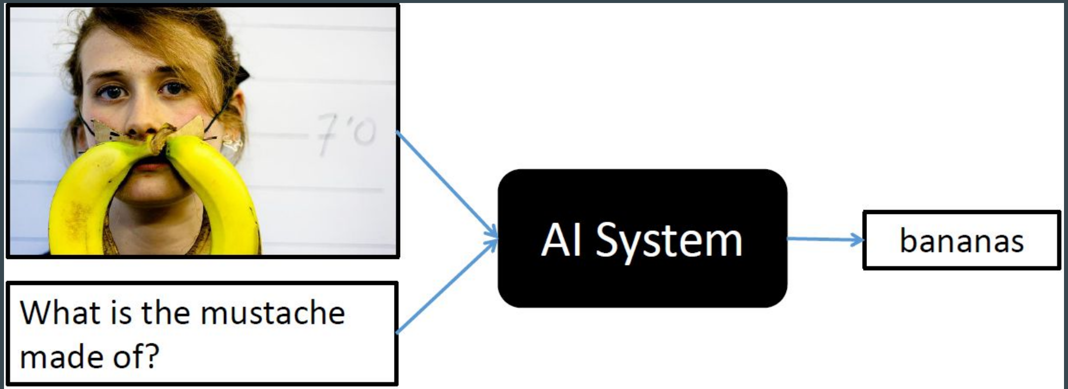
- REMIND achieves **state-of-the-art results** compared to recent methods in CVPR-2019 (BiC, Unified)
- Streaming mode for incremental batch methods: Small batch size of 50 and only a single epoch
- All models are given 1.5 GB of auxiliary memory. iCaRL, Unified, and BiC store raw images

CORe50 Results

- We compare **multi-headed** regularization baselines that require the task label during inference
- We show task-aware methods perform poorly when task labels are withheld during testing
- REMIND achieves the best results regardless of whether task labels are allowed

Percentage of performance with respect to offline baseline

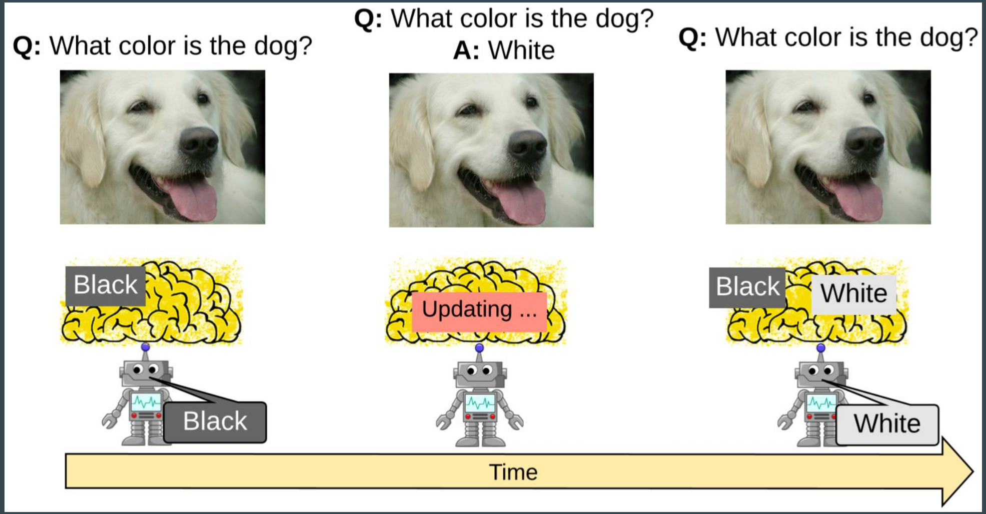




Visual Question Answering Experiments

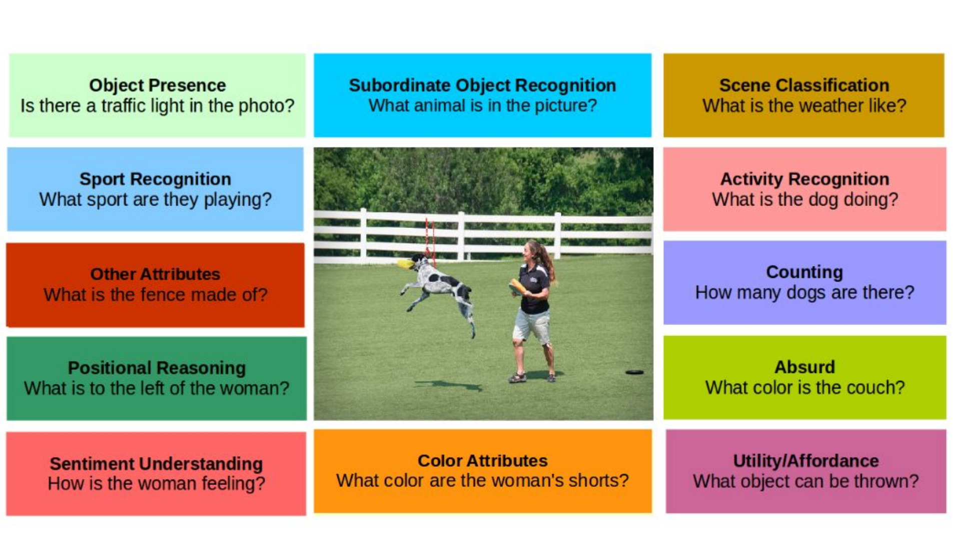
Streaming Visual Question Answering

- REMIND can be easily extended to other tasks such as visual question answering (VQA)
- **Input Stream:** Image + Question
- **Output:** Answer to Question about the Image
- Inputs are either randomly ordered (iid) or ordered by question type (q-type)



VQA on the TDIUC and CLEVR Datasets

TDIUC test for generalization across different underlying tasks required for VQA

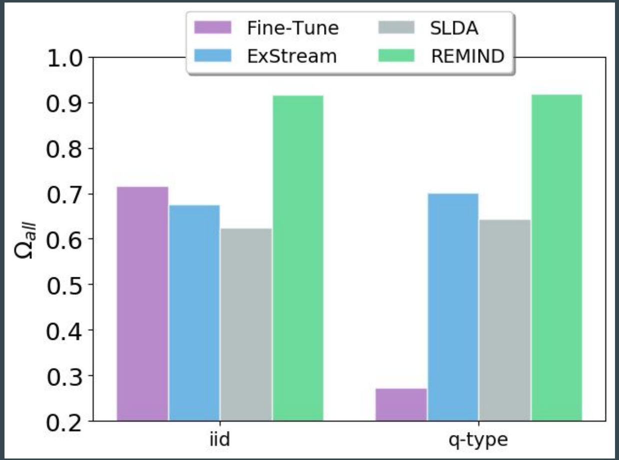


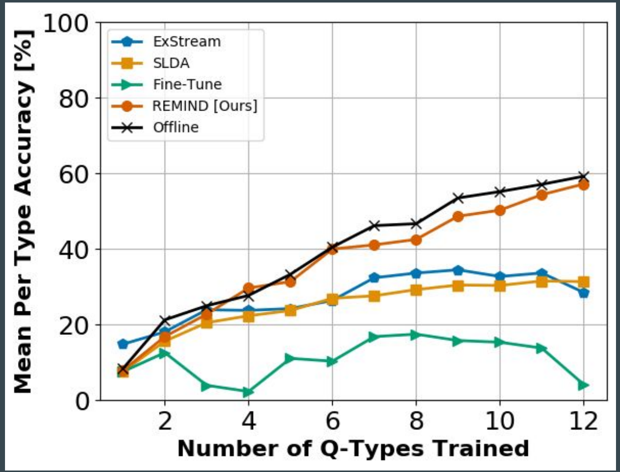
Kafle and Kanan (ICCV-2017)

CLEVR tests for multi-step compositional reasoning

Johnson et al. (CVPR-2017)

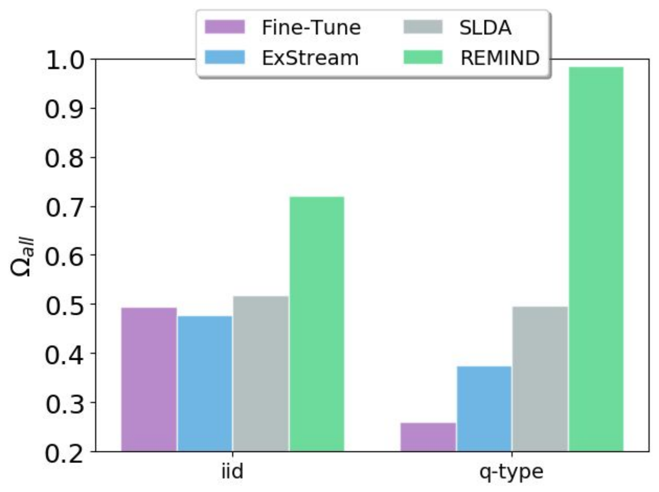
TDIUC Results

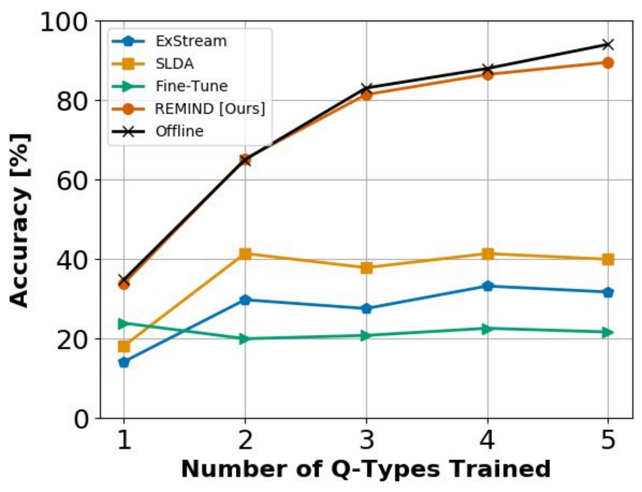




- REMIND achieves strong performance on TDIUC and outperforms baselines
- REMIND closely tracks offline performance when inputs during streaming are organized by q-type

CLEVR Results



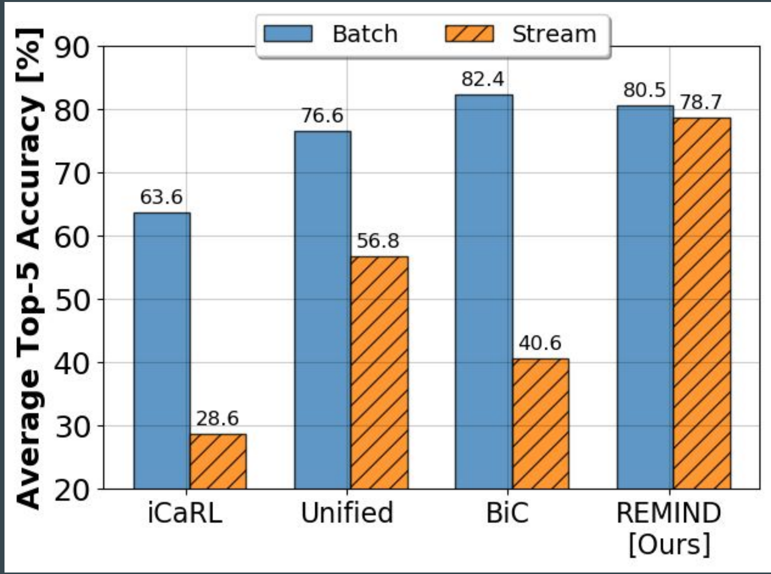


- REMIND achieves strong performance, demonstrating its ability to perform complex, multi-step reasoning in a streaming setting
- Ordering the input stream by q-type appears to be beneficial to REMIND's learning

Conclusions

- We proposed REMIND, which stores compressed mid-level CNN features as indices in a replay buffer
- REMIND outperforms existing models for object classification on ImageNet and CORe50, while remaining robust to the order in which data are presented
- We used REMIND to pioneer Visual Question Answering and achieved strong results, especially on the CLEVR dataset

Incremental Learning on 1,000 classes of ImageNet ILSVRC-2012

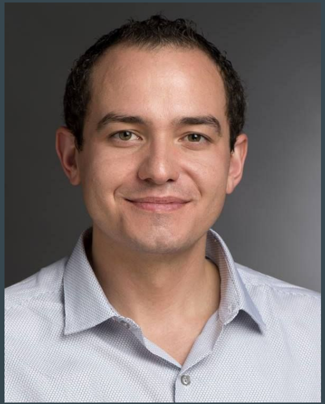


Acknowledgements: Co-Authors and Sponsors

Kushal Kafle

Robik Shrestha

Manoj Acharya



Chris Kanan

NSF DARPA AIR FORCE OFFICE OF SCIENTIFIC RESEARCH UNITED STATES AIR FORCE Adobe Research NVIDIA

Thank You!

Questions?

Tyler Hayes
tlh6792@rit.edu
https://tyler-hayes.github.io

- REMIND is a novel streaming model that uses **tensor quantization** to efficiently store hidden representations for replay
- REMIND outperforms existing incremental learning models on **ImageNet ILSVRC-2012** and **CORe50**
- REMIND achieves strong performance on **streaming Visual Question Answering**
- **arXiv:** https://arxiv.org/abs/1910.02509