# Computationally Efficient Generalization without Forgetting on Streaming Data

Ameya Prabhu

UNIVERSITY OF OXFORD

# Continual Learning: Mixture of Problems

Continual Learning:

- Confluence of slightly different problems [Talk by Prof. Tinne at CoLLAs]
  - Biology Inspired Learning
  - Scalable Training of Foundation Models
  - Practical: Compute Limited Learning, Robotics
  - Federated Learning/User Personalization

# Continual Learning: Mixture of Problems

Continual Learning:

- Confluence of slightly different problems [Talk by Prof. Tinne at CoLLAs]
  - Biology Inspired Learning
  - Scalable Training of Foundation Models
  - Practical: Compute Limited Learning, Robotics
  - Federated Learning/User Personalization

Warning: Traditional CL has different Assumptions and Goals than my talk
(But I shall point wherever it is different..)

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Static Datasets

**Input:** Each dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ of *n* samples

**Goal :** Learn $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} L(f_\theta(\mathbf{x}), \mathbf{y})$$

(Standard) Supervised Classification

# Learning from a Firehose: Streaming Learning

## Supervised (Static) Classification

**Input:** Each dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ of *n* samples

**Goal :** Learn $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

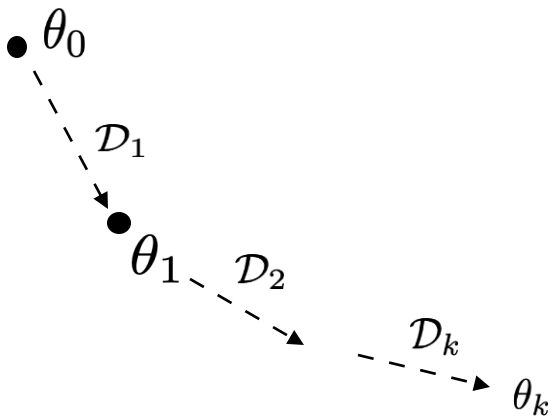
$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y} \sim \cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y})$$

It's the same process, repeated *k* times

## Continual (Streaming) Classification

**Input:** A stream of labeled data at each timestep *t*

**Goal :** Learn $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$




$$\min_\theta \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_\theta \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

## Objectives

- Scalable learning/adaptation to new data over time
- Mechanisms to add, consolidate & query knowledge ($\mathbb{K}$ )

# Learning from a Firehose: Streaming Learning

Supervised (Static) Classification

**Input:** Each dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$ of *n* samples

**Goal :** Learn $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y} \sim \cup_{i=1}^{k} \mathcal{D}_i} L(f$$

It's the same process, repe

Continual (Streaming) Classification

**Input:** A stream of labeled data at each timestep *t*

**Goal :** Learn $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

$$\mathcal{D}_2 \dashrightarrow \mathcal{D}_k \dashrightarrow \theta_k$$

$$\min_\theta \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_\theta \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

## Objectives

- Scalable learning/adaptation to new data over time
- Mechanisms to add, consolidate & query knowledge ($\mathbb{K}$ )

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Continual Learning is not Essential

Supervised (Static) Classification

$$\min_{\theta} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\cup_{i=1}^{k}\mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y})$$

Continual (Streaming) Classification

$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k}\mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

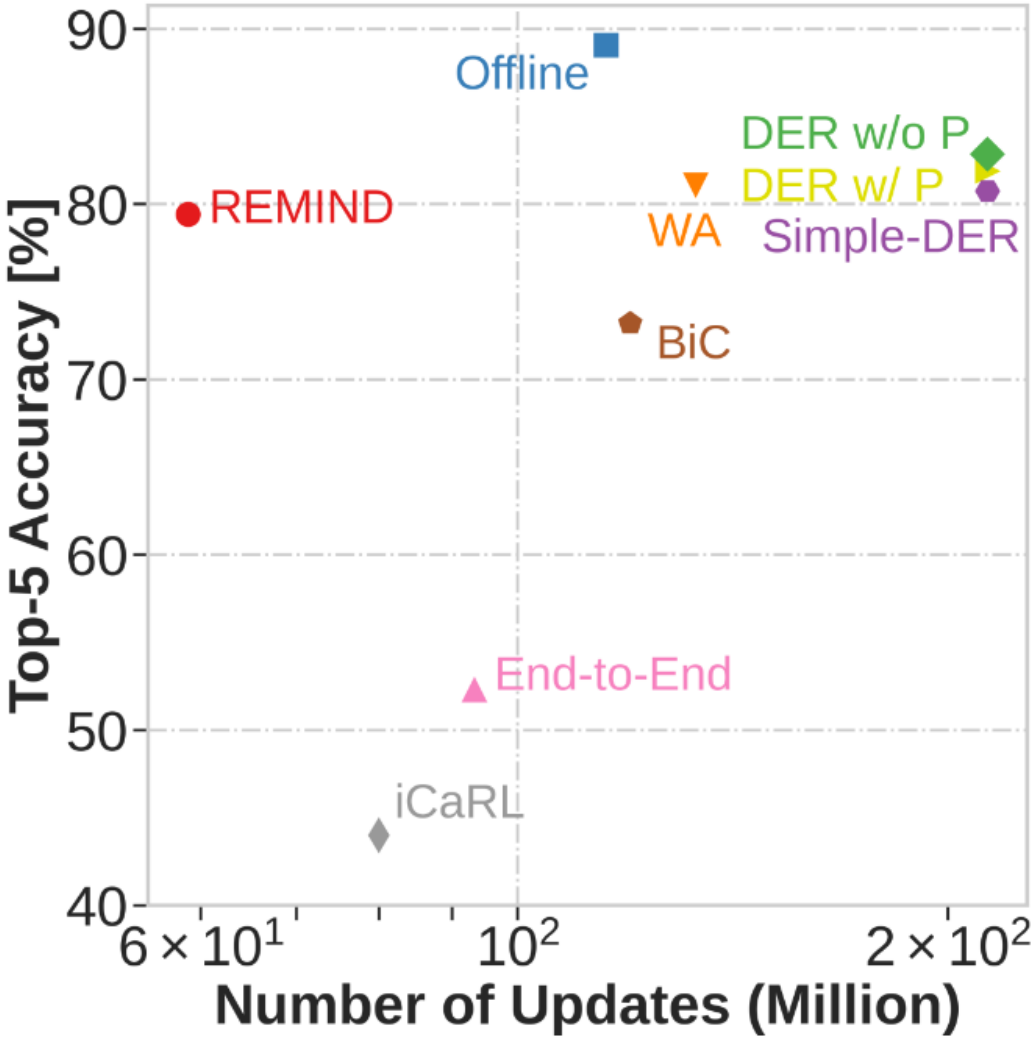
Field needs to show efficient learning in a streaming setting

<u>Concretely: When is Continual Learning Useful?</u>

- Training time/compute is far shorter than others
- Performance is not lower than the static solutions

# Continual Learning is not Essential

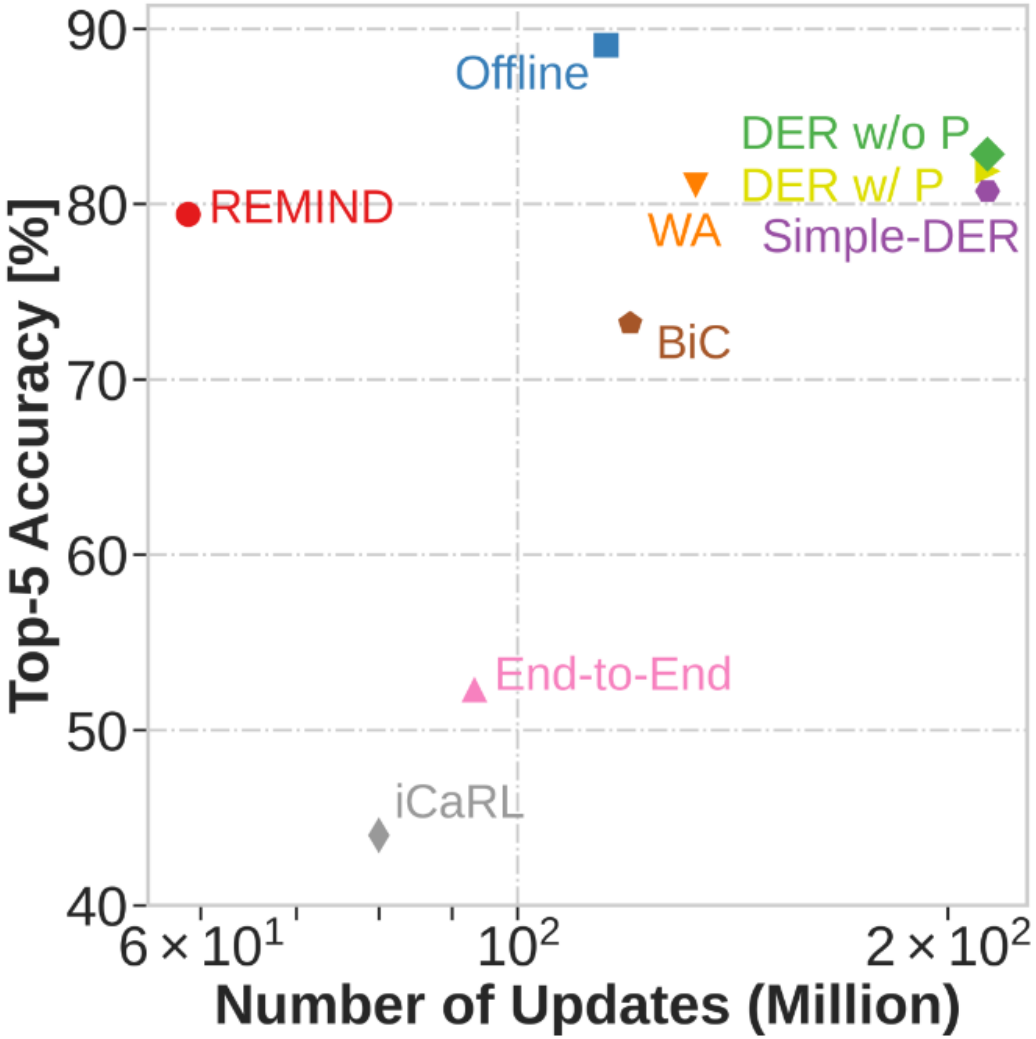
- Many recent CL methods for ImageNet use more compute than training a model from scratch!




Harun et. al, How Efficient Are Today's Continual Learning Algorithms?, CLVision 2023. Slide adapted from Chris.

# Continual Learning is not Essential

- Many recent CL methods for ImageNet use more compute than training a model from scratch!
- If you chat with people in industry, they think CL is a nice idea, but irrelevant for where it is today.

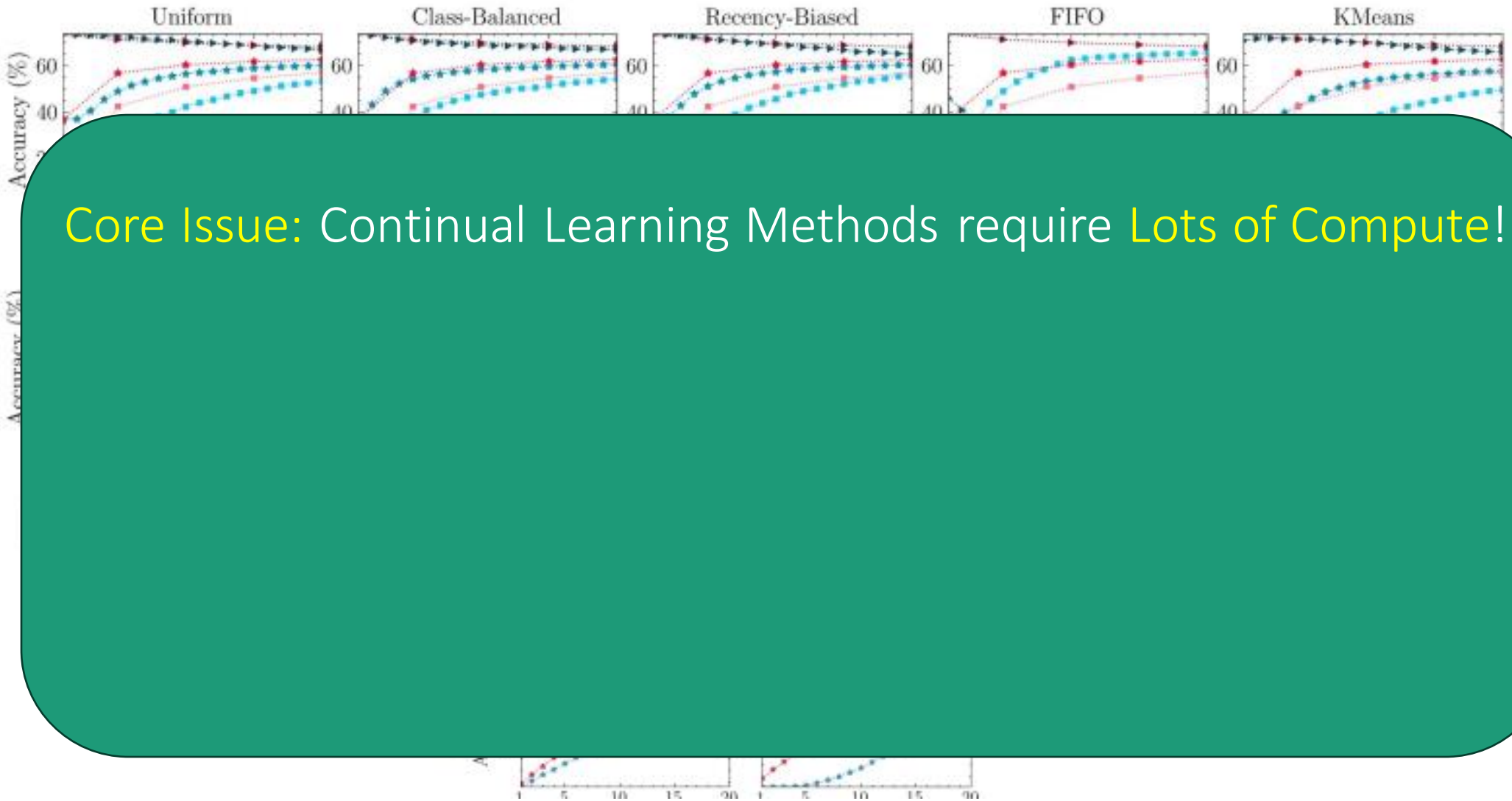



Harun et. al, How Efficient Are Today's Continual Learning Algorithms?, CLVision 2023. Slide adapted from Chris.

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

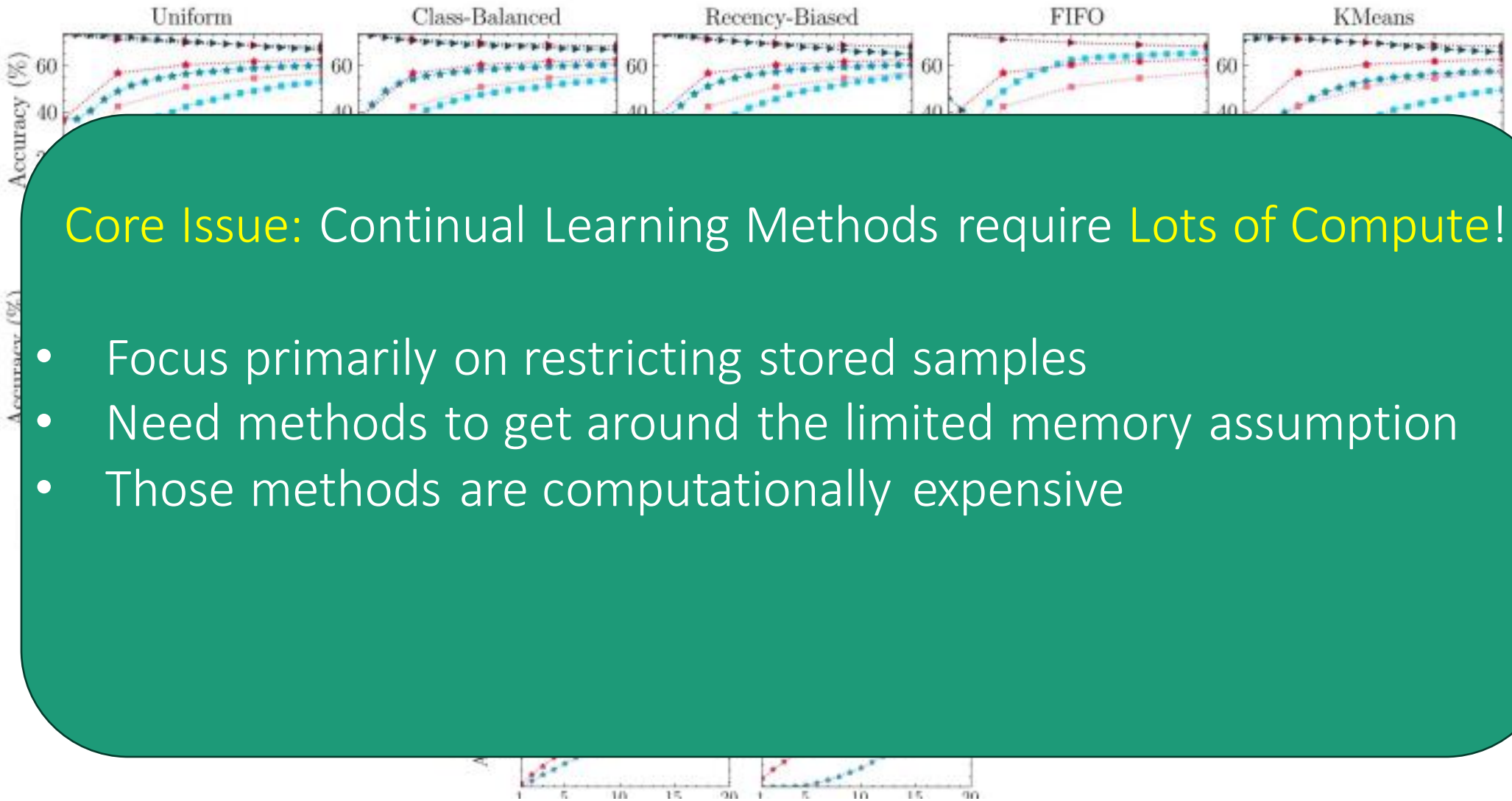
# Why Continual Learning Methods Fail: Core Issue




Core Issue: Continual Learning Methods require Lots of Compute!

Prabhu et. al., “Computationally Budgeted Continual Learning: What Does Matter?” CVPR23.

# Why Continual Learning Methods Fail: Core Issue

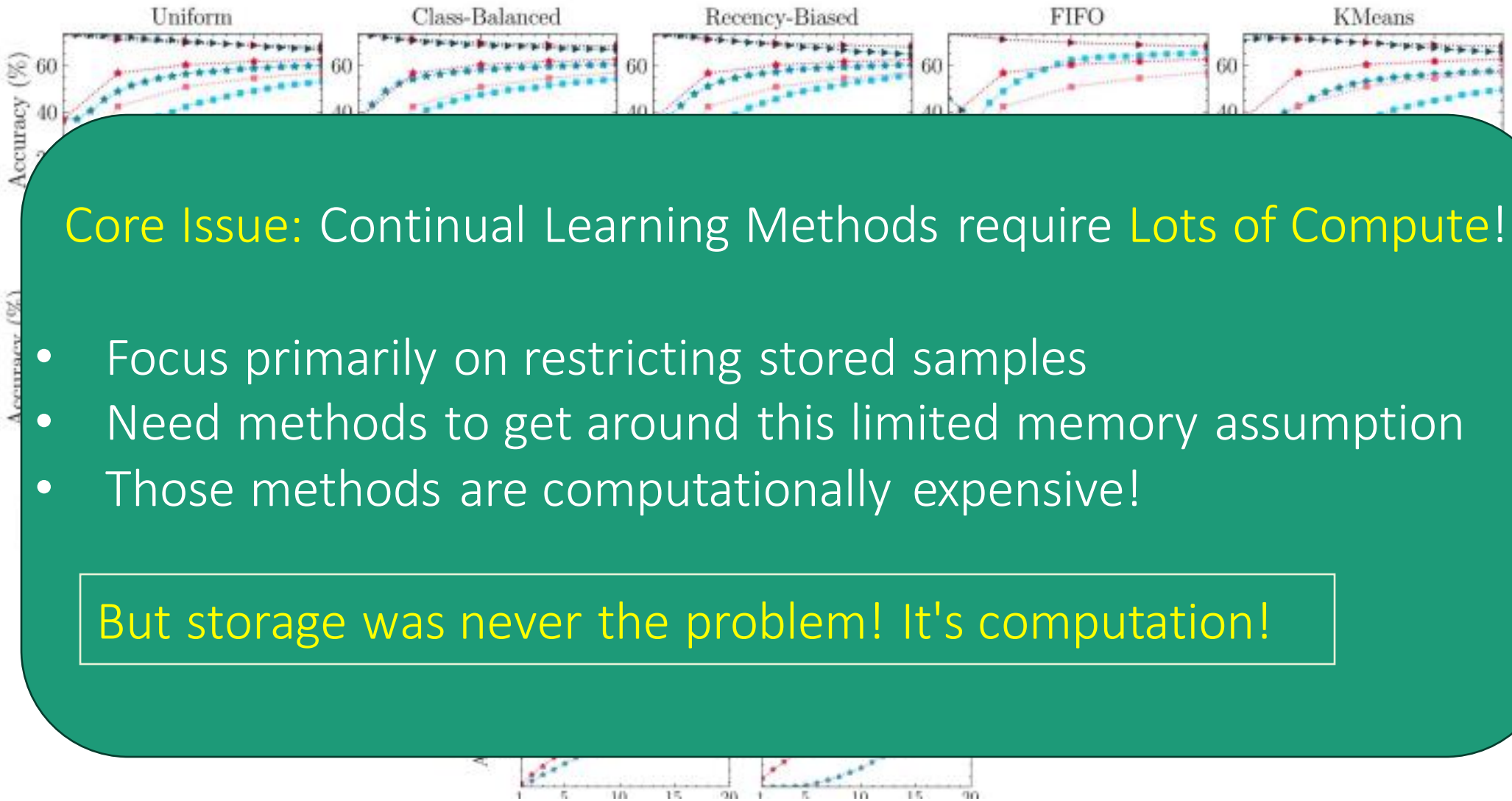



Core Issue: Continual Learning Methods require Lots of Compute!

- Focus primarily on restricting stored samples
- Need methods to get around the limited memory assumption
- Those methods are computationally expensive

Prabhu et. al., "Computationally Budgeted Continual Learning: What Does Matter?" CVPR23.

# Why Continual Learning Methods Fail: Core Issue



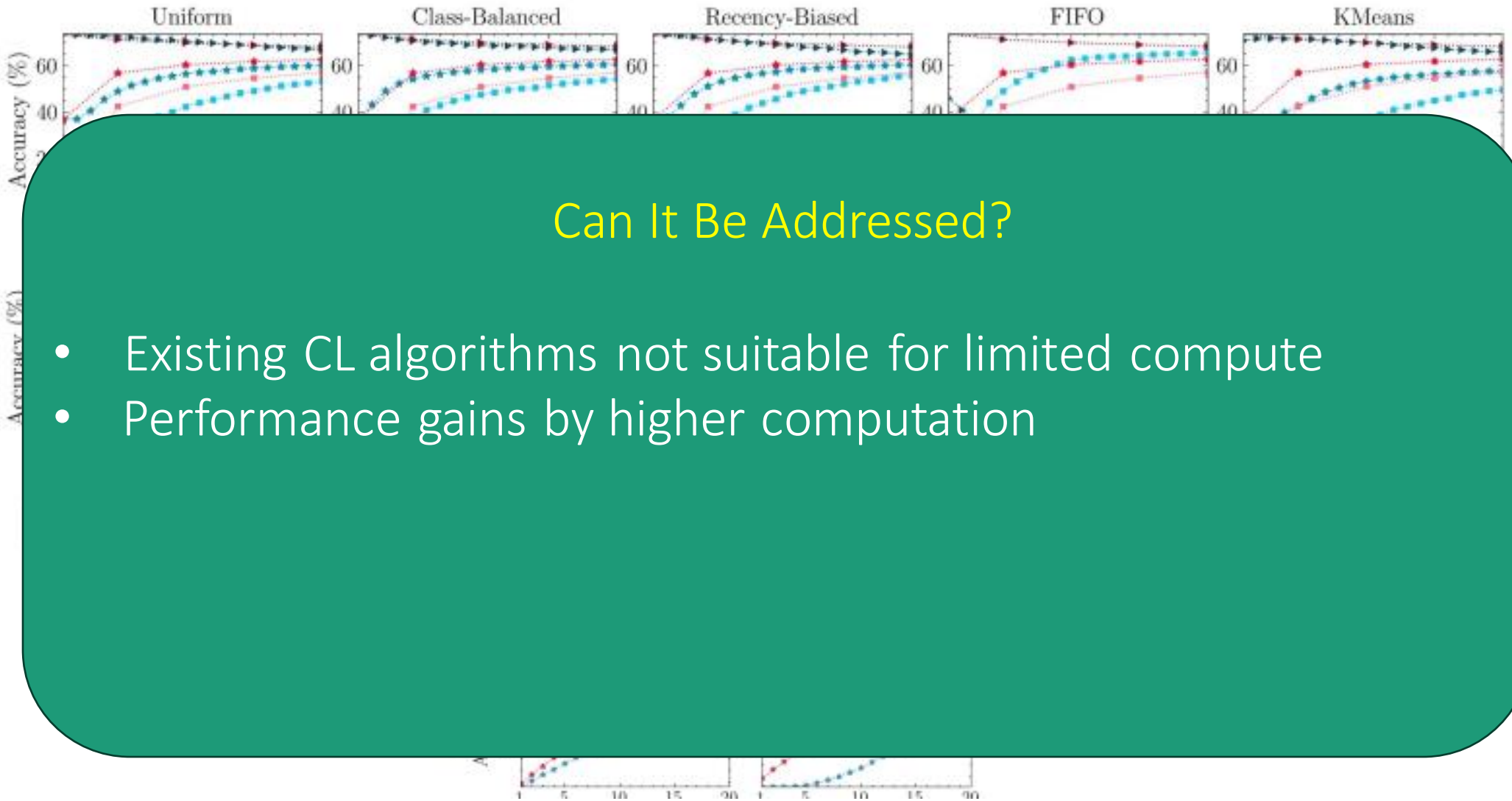

Core Issue: Continual Learning Methods require Lots of Compute!

- Focus primarily on restricting stored samples
- Need methods to get around this limited memory assumption
- Those methods are computationally expensive!

But storage was never the problem! It's computation!

Prabhu et. al., “Computationally Budgeted Continual Learning: What Does Matter?” CVPR23.

# Why Continual Learning Methods Fail: Core Issue

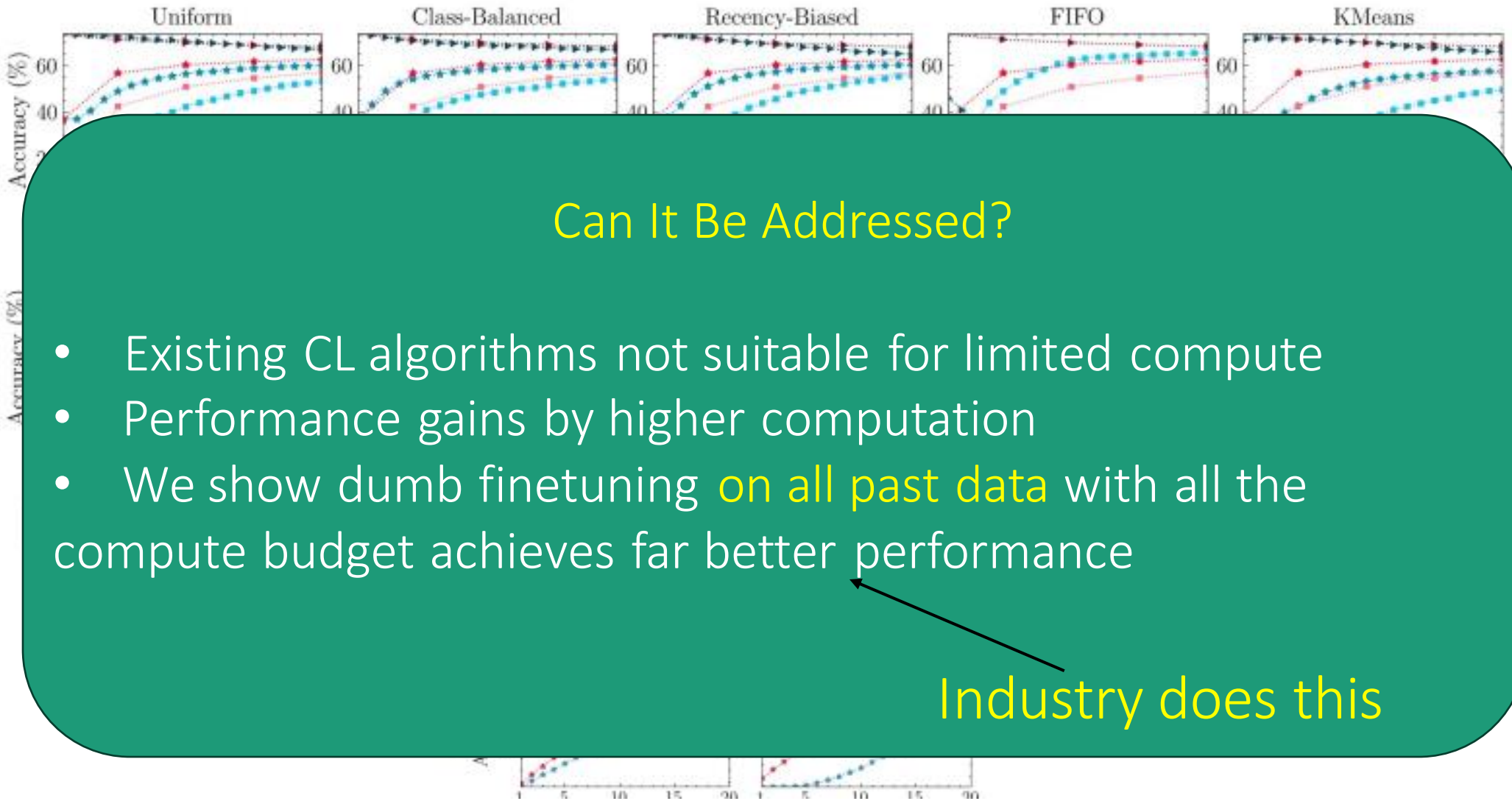



## Can It Be Addressed?

- Existing CL algorithms not suitable for limited compute
- Performance gains by higher computation

Prabhu et. al., “Computationally Budgeted Continual Learning: What Does Matter?” CVPR23.

# Why Continual Learning Methods Fail: Core Issue




Can It Be Addressed?

- Existing CL algorithms not suitable for limited compute
- Performance gains by higher computation
- We show dumb finetuning on all past data with all the compute budget achieves far better performance

Industry does this

Prabhu et. al., "Computationally Budgeted Continual Learning: What Does Matter?" CVPR23.

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Continual Learning Needs to Rethink Assumptions

- Many plausible assumptions
- Assumptions conflict: Choose wisely!
  - Tradeoff between Compute vs Memory Limitations
  - Tradeoff between Training and Inference Costs

# Continual Learning Needs to Rethink Assumptions

- Some assumptions are useful
  - Important to make assumptions to simplify hard problem statements.
    - (Limited Compute/Time Budget): GPUs are expensive!

# Continual Learning Needs to Rethink Assumptions

- Some assumptions are useful
  - Important to make assumptions to simplify hard problem statements.
- Conversely, important that these assumptions reflect the real world
  - Otherwise, no-one will use the solution proposed
  - (HDD space is virtually free, why space-efficient continual learning?)

# Continual Learning Needs to Rethink Assumptions

On Assumptions/Problem Definitions

- Important that these assumptions reflect the real world
  - Some other assumptions which are too simplistic in "GDumb: A Simple Approach that Questions Our Progress in Continual Learning", ECCV 2020

**GDumb: A Simple Approach that Questions Our Progress in Continual Learning**

Ameya Prabhu[1] Philip H.S. Torr[1] Puneet K. Dokania[1,2]
{ameya,phst,puneet}@robots.ox.ac.uk

**2.1 Simplifying Assumptions in Continual Learning**

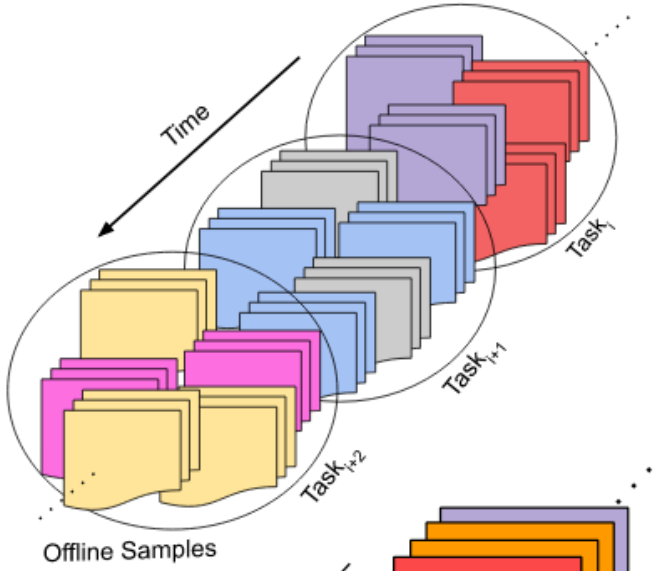
The above discussed formulation is general in the sense that it does not put any constraints whatsoever on the growing nature of the label space, nature of test samples, and size of the output space $|\lim_{t\to\infty} \mathcal{Y}_t|$. It does not put any restrictions on the resources (compute and storage) one might pick to get a reliable mapping

# Outline

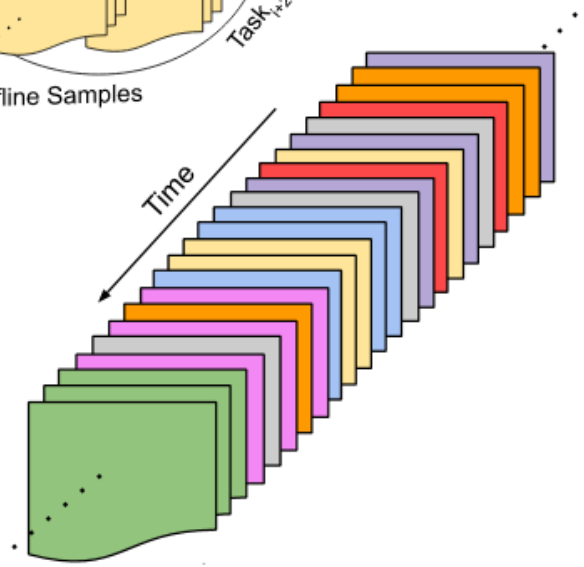
- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Quick Odyssey thru Continual Learning: Online/Offline

Offline CL

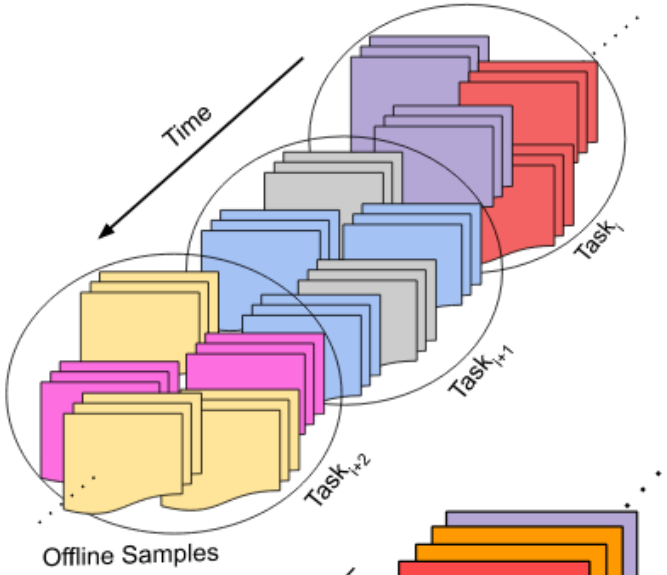



Online CL

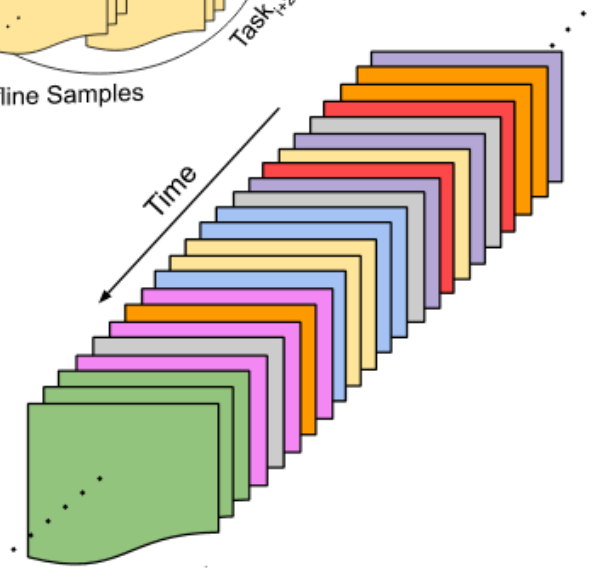


# Quick Odyssey thru Continual Learning: Online/Offline

Offline CL




Online CL




## What Differentiates?

### Does it Require Rapid Adaptation?

Yes => Time-Sensitive Model Updates
(Online CL)

- Must Measure Rapid Adaptation!

No => Can Collect Samples & Update (Efficiently!)
(Offline CL)

# Odyssey: Compute Limited Offline Setup

Learn a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ from a stream $\mathcal{S}$ revealing data sequentially over steps $t \in \{1, 2, \ldots, \infty\}$ where at every step:

1. $\mathcal{S}$ reveals a set of image-label pairs $\{(x_i^t, y_i^t)\}_{i=1}^{n_t} \sim \mathcal{D}_{j \leq t}$
2. Memory is updated to $\mathcal{T}_t = \cup_{r=1}^{t} \{(x_i^r, y_i^r)\}_{i=1}^{n_r}$
3. Continual learner updates $\theta_t$ to $\theta_{t+1}$ spending a computational budget $\mathcal{C}_t$

Prabhu et. al., "Computationally Budgeted Continual Learning: What Does Matter?" CVPR23.

# Odyssey: Compute Limited Offline Setup

Learn a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ from a stream $\mathcal{S}$ revealing data sequentially over steps $t \in \{1, 2, \ldots, \infty\}$ where at every step:

1. $\mathcal{S}$ reveals a set of image-label pairs $\{(x_i^t, y_i^t)\}_{i=1}^{n_t} \sim \mathcal{D}_{j \leq t}$
2. Memory is updated to $\mathcal{T}_t = \cup_{r=1}^{t} \{(x_i^r, y_i^r)\}_{i=1}^{n_r}$
3. Continual learner updates $\theta_t$ to $\theta_{t+1}$ spending a computational budget $\mathcal{C}_t$

Prior art fails to constrain computation to train the model

Constraint by adjusting gradient updates according to cost of each method

Prabhu et. al., "Computationally Budgeted Continual Learning: What Does Matter?" CVPR23.

# Odyssey: Compute Limited Offline Setup

Learn a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ from a stream $\mathcal{S}$ revealing data sequentially over steps $t \in \{1, 2, \ldots, \infty\}$ where at every step:

No memory limitation assumption!

1. $\mathcal{S}$ reveals a set of image-label pairs $\{(x_i^t, y_i^t)\}_{i=1}^{n_t} \sim \mathcal{D}_{j \leq t}$
2. Memory is updated to $\mathcal{T}_t = \cup_{r=1}^{t} \{(x_i^r, y_i^r)\}_{i=1}^{n_r}$
3. Continual learner updates $\theta_t$ to $\theta_{t+1}$ spending a computational budget $\mathcal{C}_t$

Prabhu et. al., “Computationally Budgeted Continual Learning: What Does Matter?” CVPR23.

# Odyssey: Compute Limited Online Setup

1. *One* data point $x_t \sim \pi_t$ sampled from a non-stationary distribution $\pi_t$ is revealed.
2. The learner makes the scalar prediction $\hat{y}_t = f(x_t; \theta_t)$ using a compute budget, $B_t^{pred}$.
3. Learner receives the true label $y_t$.
4. Learner updates the model $\theta_{t+1}$ using a compute budget, $B_t^{learn}$

Prabhu et. al., "Online Continual Learning Without the Storage Constraint" Arxiv.

# Odyssey: Compute Limited Online Setup

1. *One* data point $x_t \sim \pi_t$ sampled from a non-stationary distribution $\pi_t$ is revealed.
2. The learner makes the scalar prediction $\hat{y}_t = f(x_t; \theta_t)$ using a compute budget, $B_t^{pred}$.
3. Learner receives the true label $y_t$.
4. Learner updates the model $\theta_{t+1}$ using a compute budget, $B_t^{learn}$

Prior art fails to equalize compute (but far better than offline CL)

No memory limitation assumption!

Prabhu et. al., "Online Continual Learning Without the Storage Constraint" Arxiv.

With Memory Limits: "Real-Time Evaluation in Online Continual Learning: A New Hope" Ghunaim et al, CVPR23.

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Scenarios restrict Algorithms and that's bad!

Source: Gido's Sony Talk

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

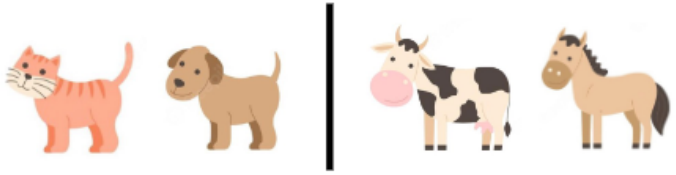
**Important challenge:** achieve positive transfer between tasks

- Domain-incremental learning *(Domain-IL)*
  - Learn the same type of problem in different contexts

**Important challenge:** alleviate catastrophic forgetting

- Class-incremental learning *(Class-IL)*
  - Incrementally learn a growing number of classes

**Important challenge:** learn to discriminate between objects not observed together



Images designed by Freepik

*Sources:* van de Ven & Tolias (2018, NeurIPS workshop), van de Ven et al. (2022, Nature Machine Intelligence)

Why? Incentivizes bad algorithm design by encouraging a two-level hierarchy

Litmus Test: If your algorithm relies on tasks being independent, it's bad!

# Our Approach: No Distributional Assumptions

UNIVERSITY OF OXFORD

## GDumb: A Simple Approach that Questions Our Progress in Continual Learning

Ameya Prabhu[1] Philip H.S. Torr[1] Puneet K. Dokania[1,2]
{ameya,phst,puneet}@robots.ox.ac.uk

### 3 Greedy Sampler and Dumb Learner (GDumb)

We now propose a simple approach that does not put any restrictions, as discussed above, over the growing nature of the label space, task boundaries, online vs offline, and the ordering of the samples in which the data-stream provides them. Thus, can easily be applied to all the CL formulations discussed in Table 1. The only requirement is to be allowed to store some episodic memories. We emphasize that we do not claim that our approach solves the general CL problem.

## Computationally Budgeted Continual Learning: What Does Matter?

Ameya Prabhu[1*] Hasan Abed Al Kader Hammoud[2*] Puneet Dokania[1] Philip H.S. Torr[1]
Ser-Nam Lim[3] Bernard Ghanem[2] Adel Bibi[1]

### 2.1. Problem Formulation

We start by first defining our proposed setting of *computationally budgeted* continual learning. Let $\mathcal{S}$ be a stream revealing data sequentially over time steps. At each time step $t \in \{1, 2, \ldots, \infty\}$, the stream $\mathcal{S}$ reveals $n_t$ image-label pairs $\{(x_i^t, y_i^t)\}_{i=1}^{n_t} \sim \mathcal{D}_j$ from distribution $\mathcal{D}_j$ where $j \in \{1, \ldots, t\}$. In this setting, we seek to learn a function $f_{\theta_t} : \mathcal{X} \to \mathcal{Y}_t$ parameterized by $\theta_t$ that maps images $x \in \mathcal{X}$ to class labels $y \in \mathcal{Y}_t$, where $\mathcal{Y}_t = \bigcup_{i=1}^{t} \mathcal{Y}_i$, which aims to correctly classify samples from any of the previous distributions $\mathcal{D}_{j \leq t}$. In general, there are no constraints on the incoming distribution $\mathcal{D}_j$, *e.g.*, the distribution might change after every time step or it may stay unchanged for all time steps. The size of the revealed stream data $n_t$ can generally

# Our Approach: Evaluate across Different Streams

UNIVERSITY OF OXFORD

**Computationally Budgeted Continual Learning: What Does Matter?**

Ameya Prabhu[1*] Hasan Abed Al Kader Hammoud[2*] Puneet Dokania[1] Philip H.S. Torr[1]
Ser-Nam Lim[3] Bernard Ghanem[2] Adel Bibi[1]

### 2.3. Constructing the Stream

We explore three stream settings in our proposed benchmark, which we now describe in detail.

(**1**) *Data Incremental Stream*: In this setting, there is no restriction on the incoming distribution $\mathcal{D}_j$ over time that has not been well-explored in prior works. We randomly shuffle all data and then reveal it sequentially over steps, which could lead to a varying distribution $\mathcal{D}_j$ over steps in which there are no clear distribution boundaries.

(**2**) *Time Incremental Stream*: In this setting, the stream data is ordered by the upload timestamp to a server, reflecting a natural distribution change $\mathcal{D}_j$ across the stream as it would in real scenarios. There is a recent shift toward studying this ordering as apparent in recent CL benchmarks, *e.g.*, CLEAR [26], Yearbook [51] and FMoW [51], Continual YFCC100M [11] and Continual Google Landmarks V2 [36].

(**3**) *Class Incremental Stream*: For completeness, we consider this classical setting in the CL literature. Each of the distributions $\mathcal{D}_j$ represents images belonging to a set of classes different from the classes of images in any other distribution $\mathcal{D}_{i \neq j}$. We benchmark these three settings using a large-scale dataset that will be detailed in the Experiments.

# Insight: Large Scale Datasets Behave Differently

| Dataset | Size | Memory | Timesteps | |
|---|---|---|---|---|
| MNIST | 40MB | 0.2-1.0 MB | 0.5-5K | Mainstream Papers |
| CIFAR10 | 160MB | 0.7-5 MB | 5K | |
| CIFAR100 | 160MB | 3-10 MB | 5K | |
| MiniImageNet | 700MB | 5-20MB | 6K | |
| CoRe50-NI | 27GB | 0.2-2GB | 12.5K | |
| GLDv2 | 400GB | **100GB** | **400K** | Us |
| YFCC-100M | 2 TB | **2 TB** | **38M** | |

[Cai et.al., ICCV 2021]

## Our Proposed Datasets

### YFCC100M-Geolocalization

- Train: 39M images
  #Classes: 712
- Challenges: Really Large

### Continual Google-Landmarks V2

- Train: 480K images
  #Classes: 10713
- Challenges: Long-tailed

### Continual- ImageNet2K

- Train: 2.5M images
  #Classes: 2000
- Challenges: Preserve pretrained knowledge

# Insight: Large Scale Datasets Behave Differently

| Dataset | Size | Memory | Timesteps |
|---|---|---|---|
| MNIST | 40MB | 0.2-1.0 MB | 0.5-5K |
| CIFAR10 | 160MB | 0.7-5 MB | 5K |
| CIFAR100 | 160MB | 3-10 MB | 5K |
| MiniImageNet | 700MB | 5-20MB | 6K |
| CoRe50-NI | 27GB | 0.2-2GB | 12.5K |
| GLDv2 | 400GB | **100GB** | **400K** |
| YFCC-100M | 2 TB | **2 TB** | **38M** |

Mainstream Papers

Us

## Changes in Constraints Apparent

- Don't constrain memory
- Constrain computational cost

Large-scale datasets not essential!
But ensure similar problems replicated

[Cai etal, 2021]

### YFCC100M-Geolocalization

- Train: 39M images
  #Classes: 712
- Challenges: Really Large

Our Datasets

### Continual Google-Landmarks V2

- Train: 580K images
  #Classes: 10713
- Challenges: Long-tailed

### Continual- ImageNet2K

- Train: 2.5M images
  #Classes: 2000
- Challenges: Preserve pretrained knowledge

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than we Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Promising Past Algorithms with No Storage Limitations

| Dir. | Reference | Applicability (our setup) | Distillation | MemUpdate | Components MemRetrieve | FC Correction | Others |
|---|---|---|---|---|---|---|---|
| | Naive | ✓ | - | Random | Random | - | - |
| Distillation | iCARL [40] | ✓ | **BCE** | **Herding** | Random | - | **NCM** |
| | LUCIR [22] | ✓ | **Cosine** | Herding | MargRank | **CosFC** | NCM |
| | PODNet [19] | ✓ | **POD** | Herding | Random | **LSC** | Imprint,NCM |
| | DarkER [10] | ✓ | **MSE** | Reservoir | Random | - | - |
| | $CO^2L$ [13] | × | **IRD** | Random | Random | **Asym.SupCon** | - |
| | SCR [32] | ✓ | - | Reservoir | Random | **SupCon** | NCM |
| Sampling | TinyER [15] | ✓ | - | **FIFO,KMeans,Reservoir** | - | - | - |
| | GSS [5] | × | - | **GSS** | Random | - | - |
| | MIR [3] | × | - | Reservoir | **MIR** | - | - |
| | GDumb [36] | ✓ | - | **Balanced** | Random | - | **MemOnly** |
| | Mnemonics [28] | × | - | **Mnemonics** | - | - | BalFineTune |
| | OCS [52] | × | - | **OCS** | Random | - | - |
| | InfoRS [44] | × | MSE | **InfoRS** | Random | - | - |
| | RMM [27] | × | - | **RMM** | - | - | - |
| | ASER [43] | × | - | **SV** | **ASV** | - | - |
| | RM [7] | ✓ | - | **Uncertainty** | Random | - | AutoDA |
| | CLIB [24] | × | - | **Max Loss** | Random | - | MemOnly,AdaLR |
| FC Layer | BiC [48] | × | CrossEnt | Random | Random | **BiC** | - |
| | WA [55] | × | CrossEnt | Random | Random | **WA** | - |
| | SS-IL [2] | × | TKD | Random | Balanced | **SS** | - |
| | CoPE [17] | ✓ | - | Balanced | Random | **PPPLoss** | - |
| | ACE [11] | ✓ | - | Reservoir | Random | **ACE** | - |

## Past Algorithms

Aim: Make DNNs robust to distribution shifts

Part of Work: Computationally Budgeted Continual Learning: What Does Matter? CVPR 2023

*Ameya Prabhu*, Hasan Abed Al Kader Hammoud*, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, Adel Bibi*

# Promising Past Algorithms with No Storage Limitations

| Dir. | Reference | Applicability (our setup) | Distillation | MemUpdate | Components MemRetrieve | FC Correction | Others |
|---|---|---|---|---|---|---|---|
| | Naive | ✓ | - | Random | Random | - | - |
| Distillation | iCARL [40] | ✓ | **BCE** | **Herding** | Random | - | **NCM** |
| | LUCIR [22] | ✓ | **Cosine** | Herding | MargRank | **CosFC** | NCM |
| | PODNet [19] | ✓ | **POD** | Herding | Random | **LSC** | Imprint,NCM |
| | DarkER [10] | ✓ | **MSE** | Reservoir | Random | - | - |
| | $CO^2L$ [13] | × | **IRD** | Random | Random | **Asym.SupCon** | - |
| | SCR [32] | ✓ | - | Reservoir | Random | **SupCon** | NCM |
| Sampling | TinyER [15] | ✓ | - | **FIFO,KMeans,Reservoir** | - | - | - |
| | GSS [5] | × | - | **GSS** | Random | - | - |
| | MIR [3] | × | - | Reservoir | **MIR** | - | - |
| | GDumb [36] | ✓ | - | **Balanced** | Random | - | **MemOnly** |
| | Mnemonics [28] | × | - | **Mnemonics** | - | - | BalFineTune |
| | OCS [52] | × | - | **OCS** | Random | - | - |
| | InfoRS [44] | × | MSE | **InfoRS** | Random | - | - |
| | RMM [27] | × | - | **RMM** | - | - | - |
| | ASER [43] | × | - | **SV** | **ASV** | - | - |
| | RM [7] | ✓ | - | **Uncertainty** | Random | - | AutoDA |
| | CLIB [24] | × | - | **Max Loss** | Random | - | MemOnly,AdaLR |
| FC Layer | BiC [48] | × | CrossEnt | Random | Random | **BiC** | - |
| | WA [55] | × | CrossEnt | Random | Random | **WA** | - |
| | SS-IL [2] | × | TKD | Random | Balanced | **SS** | - |
| | CoPE [17] | ✓ | - | Balanced | Random | **PPPLoss** | - |
| | ACE [11] | ✓ | - | Reservoir | Random | **ACE** | - |

## Past Algorithms

Aim: Make DNNs robust to distribution change

## Three Principle Directions

Reduce effect of distribution shift from past data by:

- Distillation
  > Enforce Output Similarity with Old Models
- Sampling Old Data
  > Create representative coreset of past knowledge
- Correcting FC Layer
  > Posits knowledge in representations far less affected, but classifier gets worse

Part of Work: Computationally Budgeted Continual Learning: What Does Matter? CVPR 2023

*Ameya Prabhu*, Hasan Abed Al Kader Hammoud*, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, Adel Bibi*

# Promising Past Algorithms with No Storage Limitations

UNIVERSITY OF OXFORD

(New) Primary Constraint

Reduce computational costs: Storage is (virtually) free but GPUs are expensive

## Takeaways

Extensive testing across streams which add *new classes, new data & across time* says:

- All three algorithmic directions fail when computational costs are equalized
- The baseline: Sample random subset of data and train using all budget is the best!

Conclusion was consistent across tested system constraints:
(a) varying computational budgets (b) varying stream sizes and timesteps

Three Principle Directions

Reduce effect of distribution shift from past data by:

- Distillation
  > Enforce Output Similarity with Old Models
- Sampling Old Data
  > Create representative coreset of past knowledge
- Correcting FC Layer
  > Posits knowledge in representations far less affected, but classifier gets worse

Part of Work: Computationally Budgeted Continual Learning: What Does Matter? CVPR 2023

*Ameya Prabhu*, Hasan Abed Al Kader Hammoud*, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, Adel Bibi*

# Promising Past Algorithms with No Storage Limitations

UNIVERSITY OF OXFORD

## (New) Primary Constraint

Reduce computational costs: Storage is (virtually) free but GPUs are expensive

## Three Principle Directions

Reduce effect of distribution shift from past data by:

- Distillation
  > Enforce Output Similarity with Old Models
- Sampling Old Data
  > Create representative coreset of past knowledge
- Correcting FC Layer
  > Posits knowledge in representations far less affected, but classifier gets worse

## Takeaways

Extensive testing across streams which add *new classes, new data & across time* says:

- All three algorithmic directions fail when computational costs are equalized
- The baseline: Sample random subset of data and train using all budget is the best!

Conclusion was consistent across tested system constraints:
(a) varying computational budgets (b) varying stream sizes and timesteps

Why: What's the core limiting phenomenon?
Addressing that could give better directions for design of continual algorithms

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Core Issue: Large-Scale Loss of Past Knowledge

In One Sentence: Continual Learning updates the model too much, too quickly!

# Core Issue: Large-Scale Loss of Past Knowledge

In One Sentence: Continual Learning updates the model too much, too quickly!

**Continual evaluation for lifelong learning: Identifying the stability gap**

Matthias De Lange, Gido van de Ven, Tinne Tuytelaars

My Hypothesis: Essentially

- Nice feature space, condensing a lot of knowledge (separation between lots of concepts)
- Small ad-hoc updates needed to destroy this separation (deep nets are highly non-linear)!
- Forgetting is catastrophic, as feature space changes dramatically with these updates

# Core Issue: Large-Scale Loss of Past Knowledge

In One Sentence: Continual Learning updates the model too much, too quickly!

**Continual evaluation for lifelong learning: Identifying the stability gap**

Matthias De Lange, Gido van de Ven, Tinne Tuytelaars

My Hypothesis: Essentially,

- Nice feature space, condensing a lot of knowledge (separation between lots of concepts)
- Small ad-hoc updates needed to destroy this separation (deep nets are highly non-linear)!
- Forgetting is catastrophic, as feature space changes dramatically with these updates

This is worse because,

- Recovering the clean separation requires lot more updating than it takes to disrupt it
- This process repeats every time we want to learn something new..

Important: *Not a last layer* problem, community vastly underestimate this discovery!

# Core Issue: Large-Scale Loss of Past Knowledge

Stability Gap:

- Small ad-hoc updates needed to destroy well-separated features!
- Forgetting is catastrophic, as feature space changes dramatically
- Recovering the clean separation is hard, requiring a lot more updates than it takes to disrupt it

Important: *Not a last layer* problem, vastly underestimated! Why?

Toy tests on Split-MNIST/Split-CIFAR style datasets with no pretrained model (train-from-scratch)!

*The lack of vast amount of forgetting creates the illusion of quick recovery*

# Core Issue: Large-Scale Loss of Past Knowledge

Stability Gap:

- Small ad-hoc updates needed to destroy well-separated features!
- Forgetting is catastrophic, as feature space changes dramatically
- Recovering the clean separation is hard, requiring a lot more updates than it takes to disrupt it

There is a vast amount of forgetting, overshadows any new knowledge learned

(#1) Regularization; preserve features

**Fine-tuning can cripple your foundation model; preserving features may be the solution**

Jishnu Mukhoti[1], Yarin Gal[1], Philip H. S. Torr[1], Puneet K. Dokania[1,2]
University of Oxford[1] & Five AI Limited[2]

(#2) Constrain plasticity; LoRA

OVERCOMING THE STABILITY GAP IN CONTINUAL LEARNING

**Md Yousuf Harun**
Rochester Institute of Technology
mh1023@rit.edu

**Christopher Kanan**
University of Rochester
ckanan@cs.rochester.edu

My Direction: Something more radical is needed! Don't touch the deep net (for now)!

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Radical Idea: Develop Continual Data Structures

UNIVERSITY OF OXFORD

## Past Algorithms

Aim: Make DNNs robust to distribution change

## Proposed Algorithm

Aim: Make a learner with a *.add(), .retrieve()* property

Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Towards a Continual-by-Design System

## Past Algorithms

Aim: Make DNNs robust to distribution change

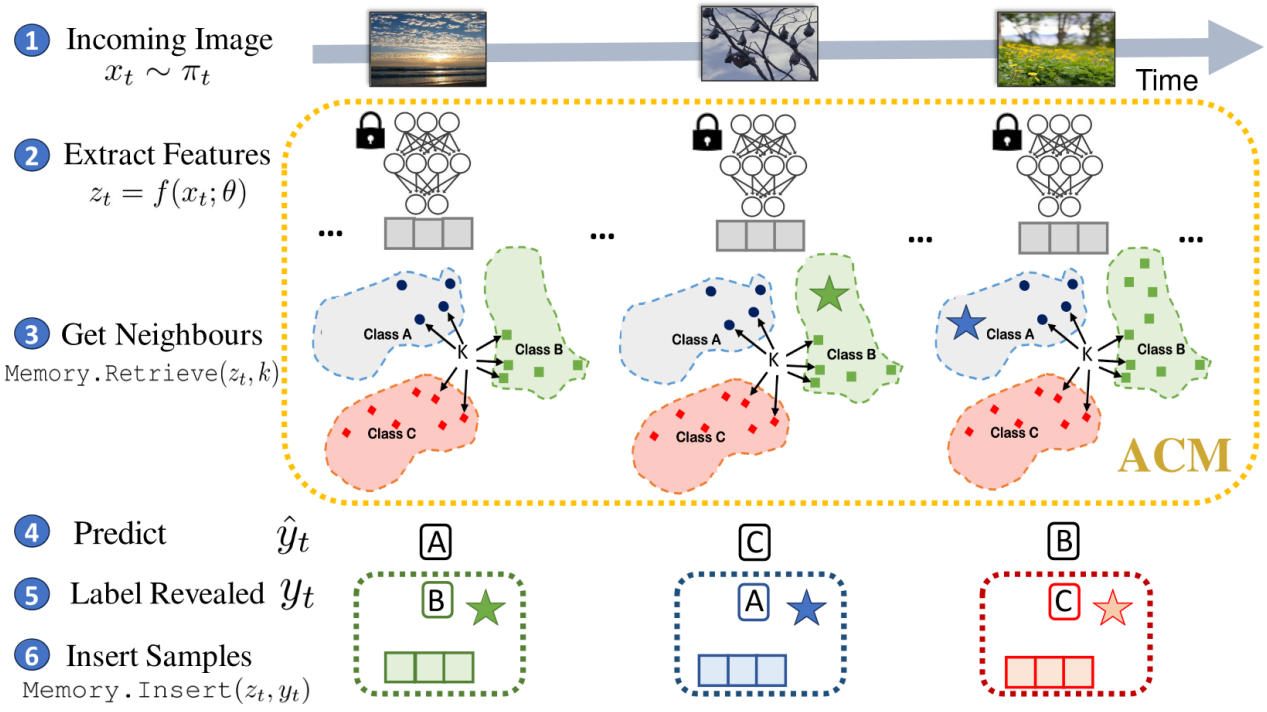
## Our Algorithm

Aim: Make a learner with a *.insert(), .retrieve()* property

## Proposed Method

A continual by-design system with: insert() & retrieve()

- Apply kNN on pretrained DNN features!




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Towards a Continual-by-Design System

## Past Algorithms

Aim: Make DNNs robust to distribution change

## Our Algorithm

Aim: Make a learner with a *.insert(), .retrieve()* property

## Proposed Method

A continual by-design system with: insert() & retrieve()

- Apply kNN on pretrained DNN features!

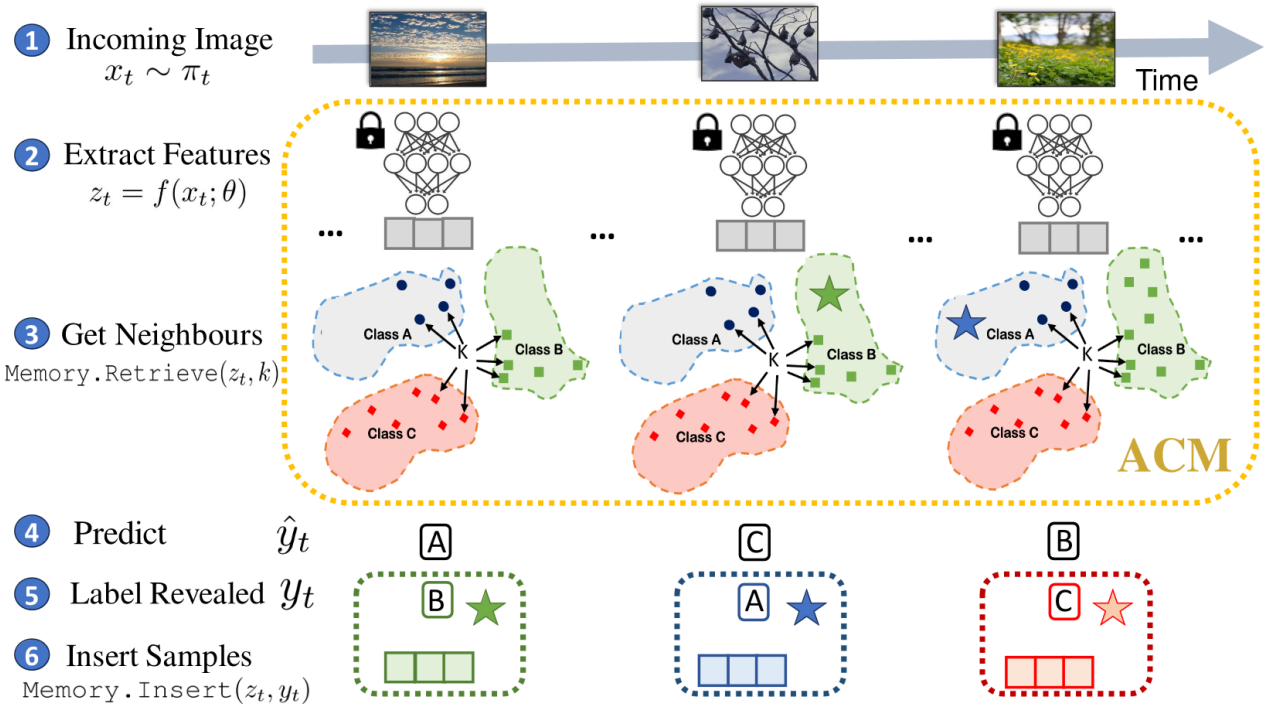
Properties

> Consistency: Guarantee of *zero* forgetting on seen data

> *Natural* insert() operation: The primary missing property

> Zero stability gap: Not updating the deep network

> Fast updates: HNSW is an extremely efficient structure




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

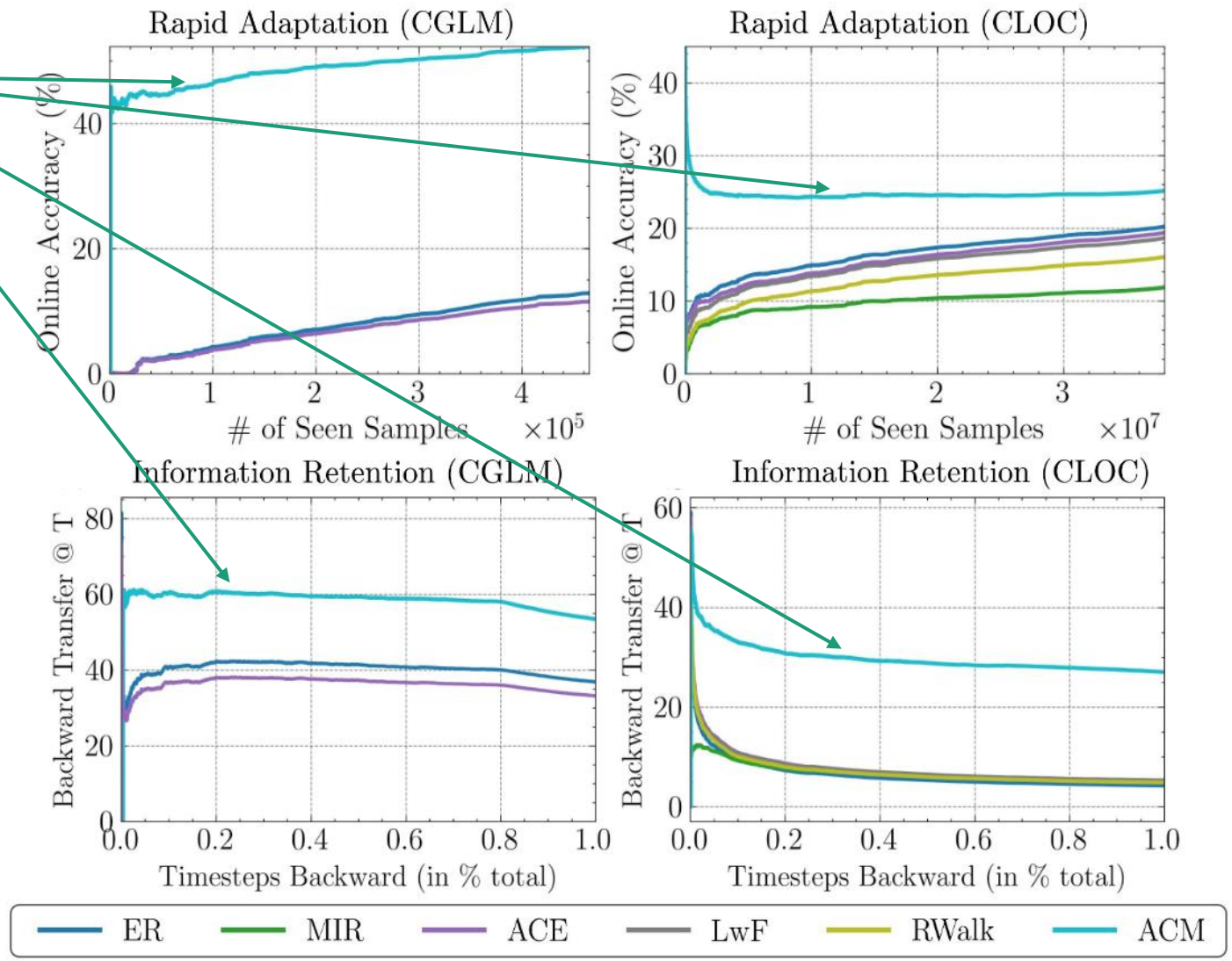
*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Proposed Method (ACM)

**Results**

- Outperforms SOTA systems by large margin
- Very little compute for insert() and retrieve()




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Proposed Method (ACM)

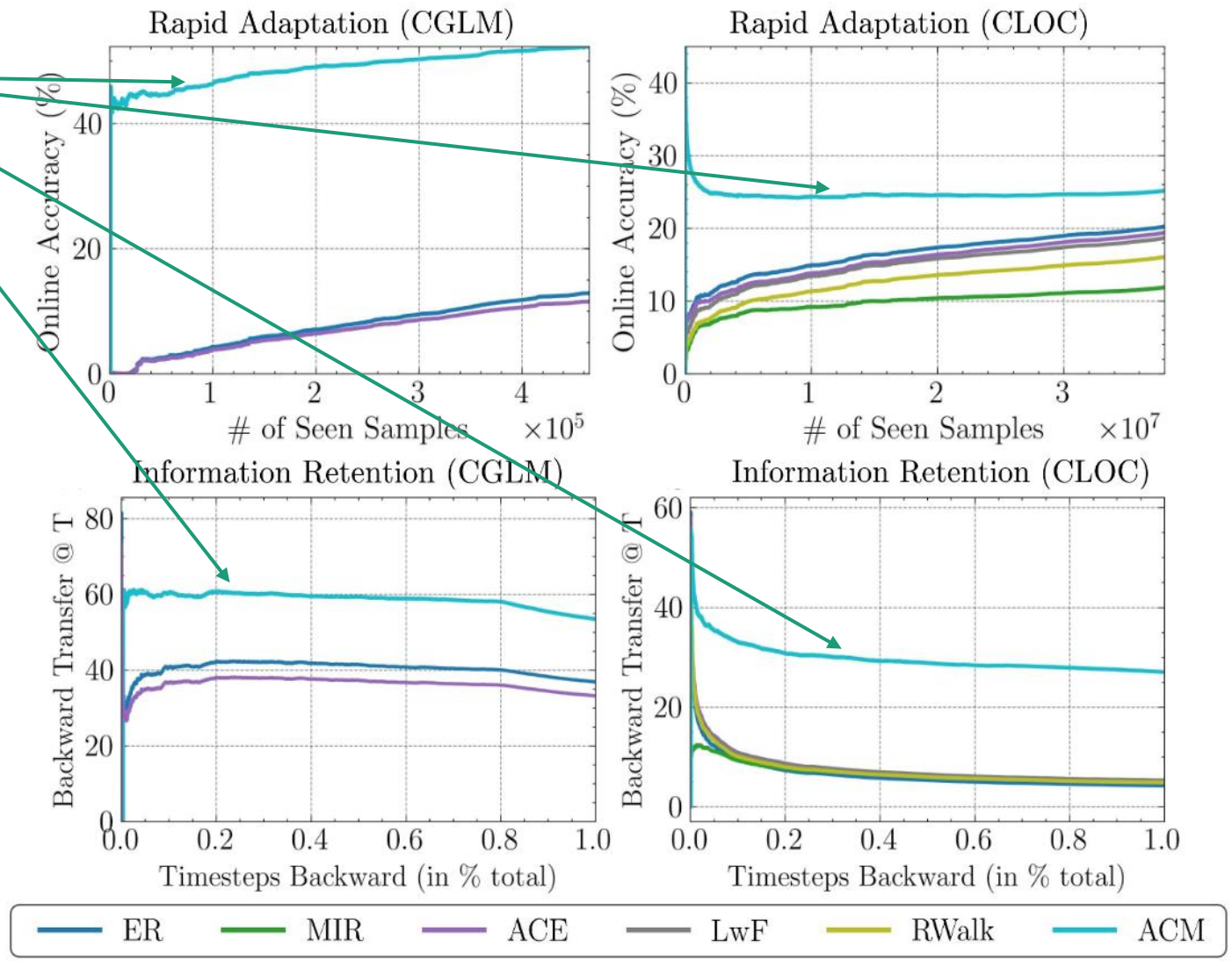
**Results**

- Outperforms SOTA systems by large margin
- Very little compute for insert() and retrieve()

Adaptation: Rapidly adapt to incoming data

> Per sample adaptability, unfeasible with DNNs

> Natural streams need this fast adaptability!




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Proposed Method (ACM)

**Results**

- Outperforms SOTA systems by large margin
- Very little compute for insert() and retrieve()

Adaptation: Rapidly adapt to incoming data

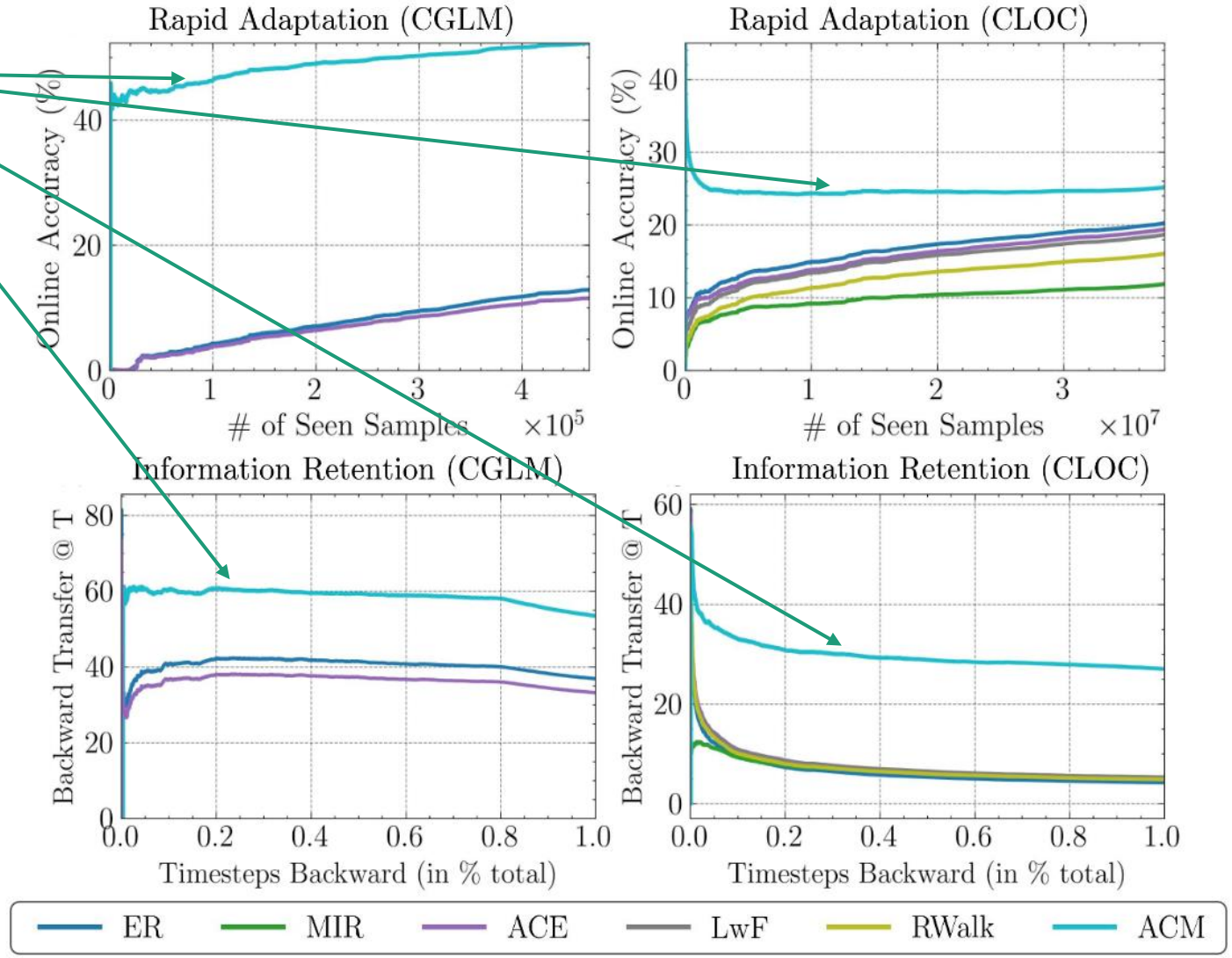
> Per sample adaptability, unfeasible with DNNs

> Natural streams need this fast adaptability!

Forgetting: No stability gap + consistency shows!

> Radically higher retention of information

> Unmatched by deep nets despite more training



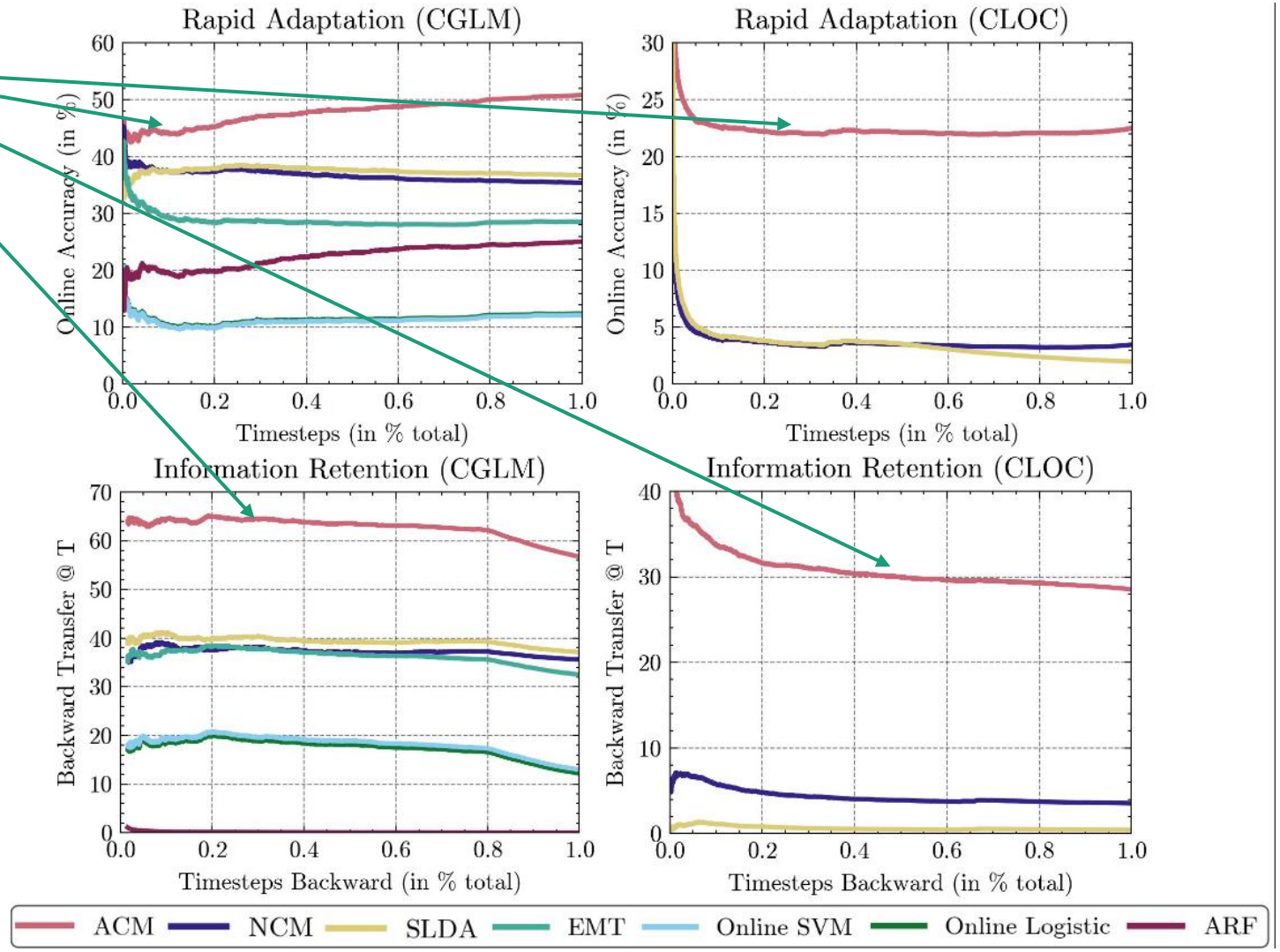

Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Proposed Method (ACM)

Other approaches which have a fixed deep net?
> Isolates the effect of consistency from stability




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

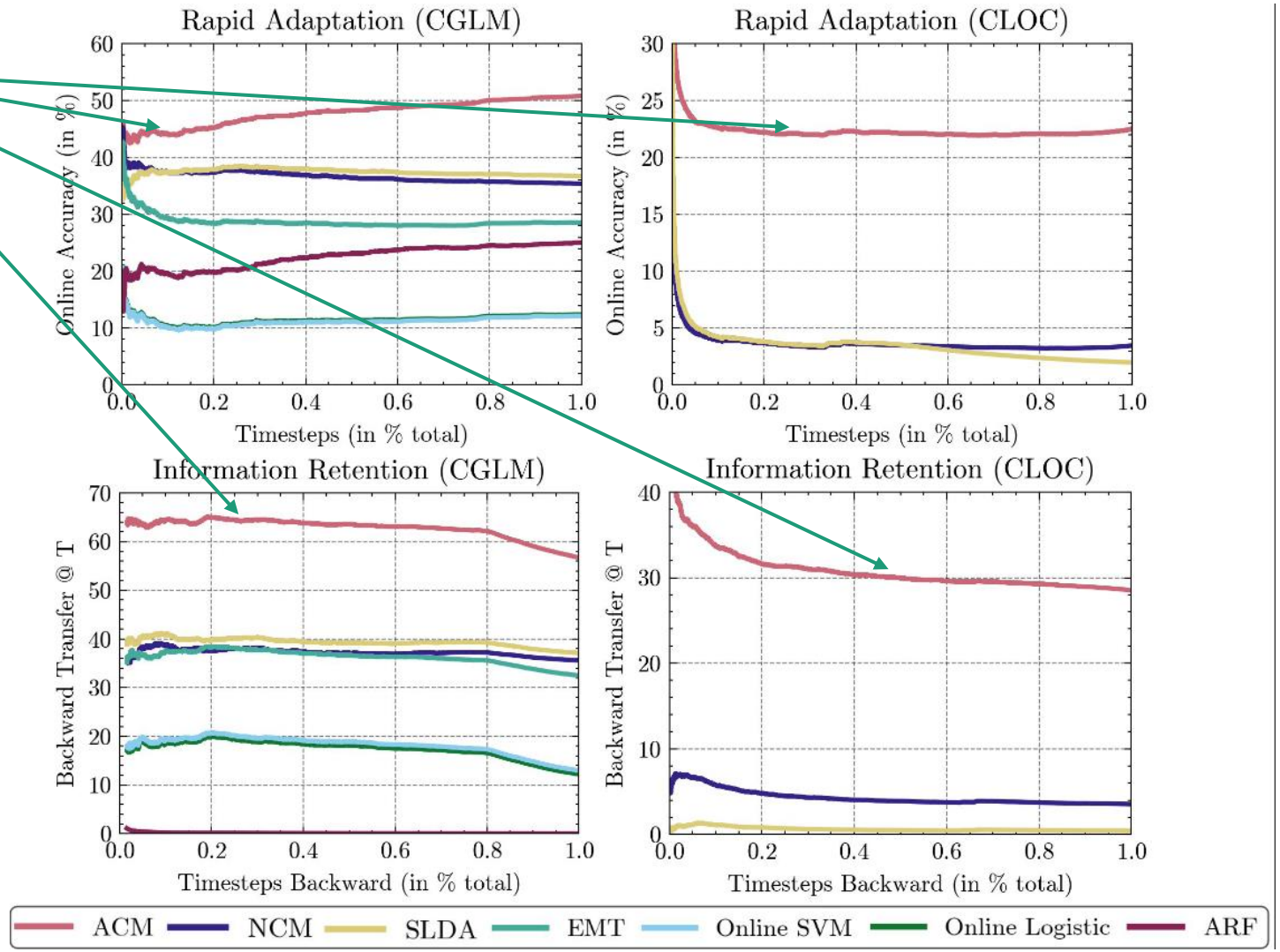
*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Proposed Method (ACM)

Other approaches which have a fixed deep net?
> Isolates the effect of consistency from stability

Observations
> All have high adaptability unlike deep nets
> Collapse of test accuracy (extreme forgetting)
> Cannot handle more complex CLOC dataset




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

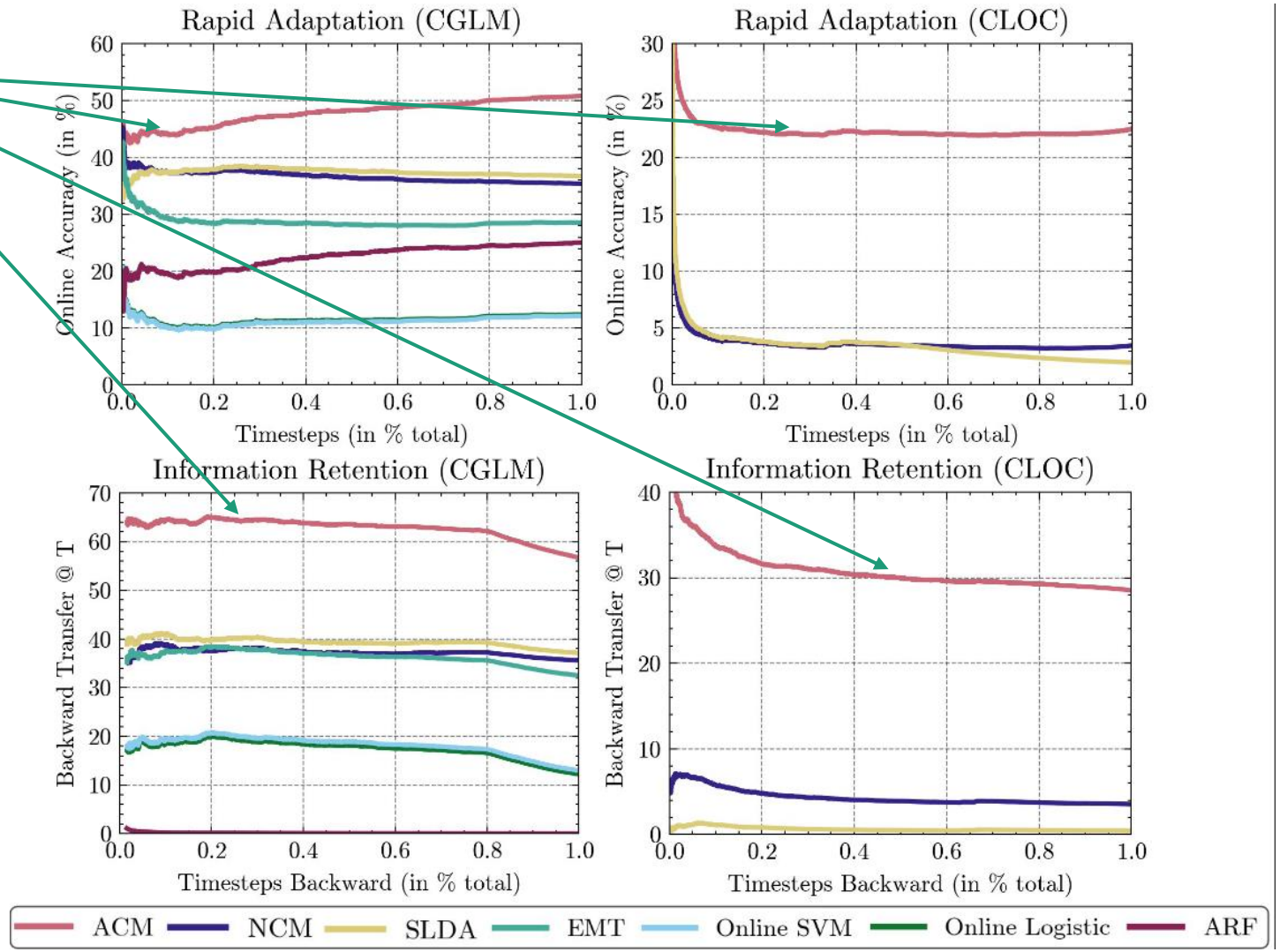
# Results: A Shocking Improvement!

## Proposed Method (ACM)

Other approaches which have a fixed deep net?
> Isolates the effect of consistency from stability

Observations
> All have high adaptability unlike deep nets
> Collapse of test accuracy (extreme forgetting)
> Cannot handle more complex CLOC dataset

Drawbacks of ACM
a) Helpful only in online scenarios:
> compute is constrained; need fast updates




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

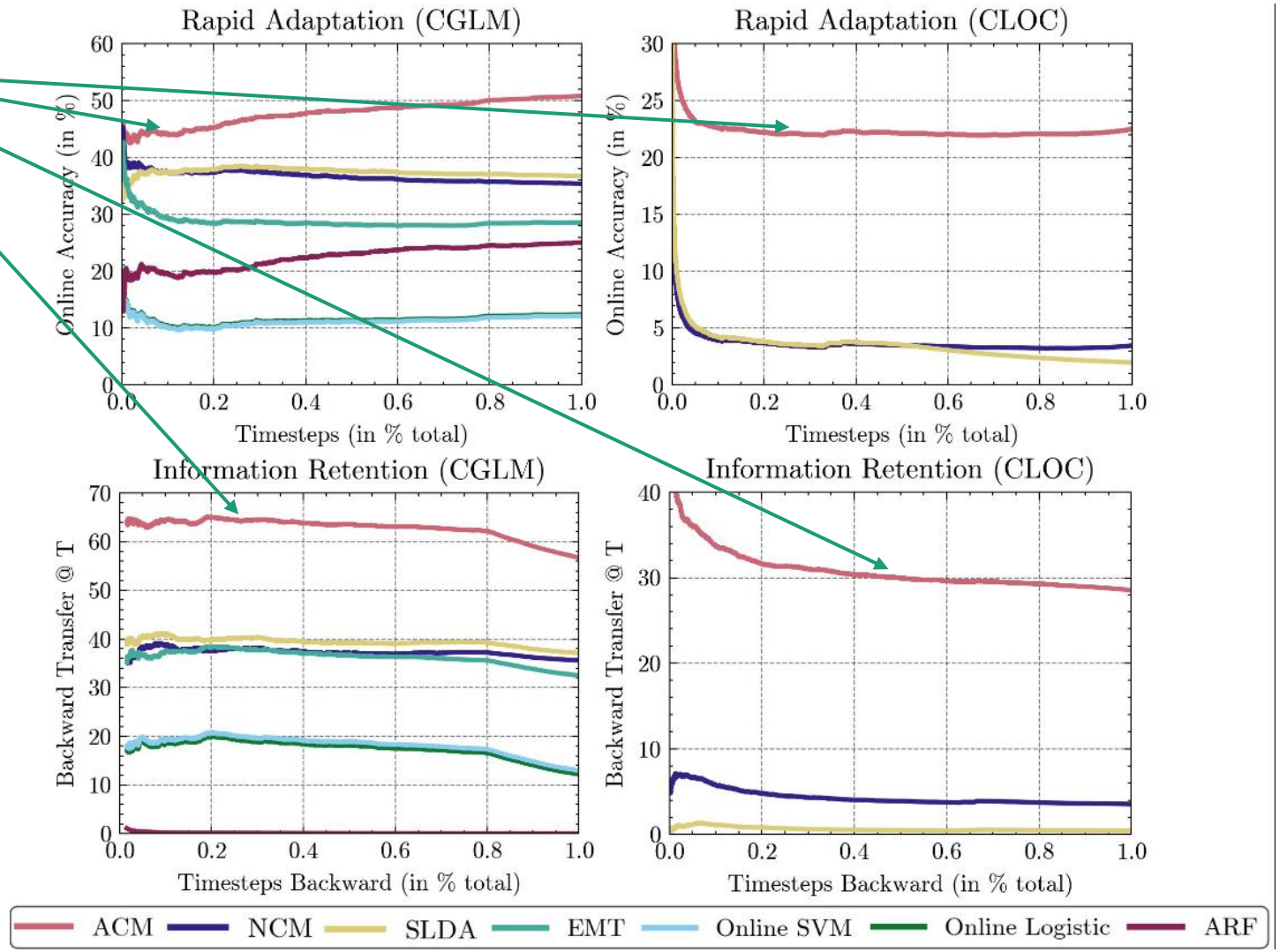
## Proposed Method (ACM)

Other approaches which have a fixed deep net?
> Isolates the effect of consistency from stability

Observations
> All have high adaptability unlike deep nets
> Collapse of test accuracy (extreme forgetting)
> Cannot handle more complex CLOC dataset

Drawbacks of ACM
a) Helpful only in online scenarios:
> compute is constrained; need fast updates

b) Does need to update deep net at some point!
> Scales surprisingly well (Imagenet1K -> CLOC)
> However, is avoiding the issue (for now)




Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

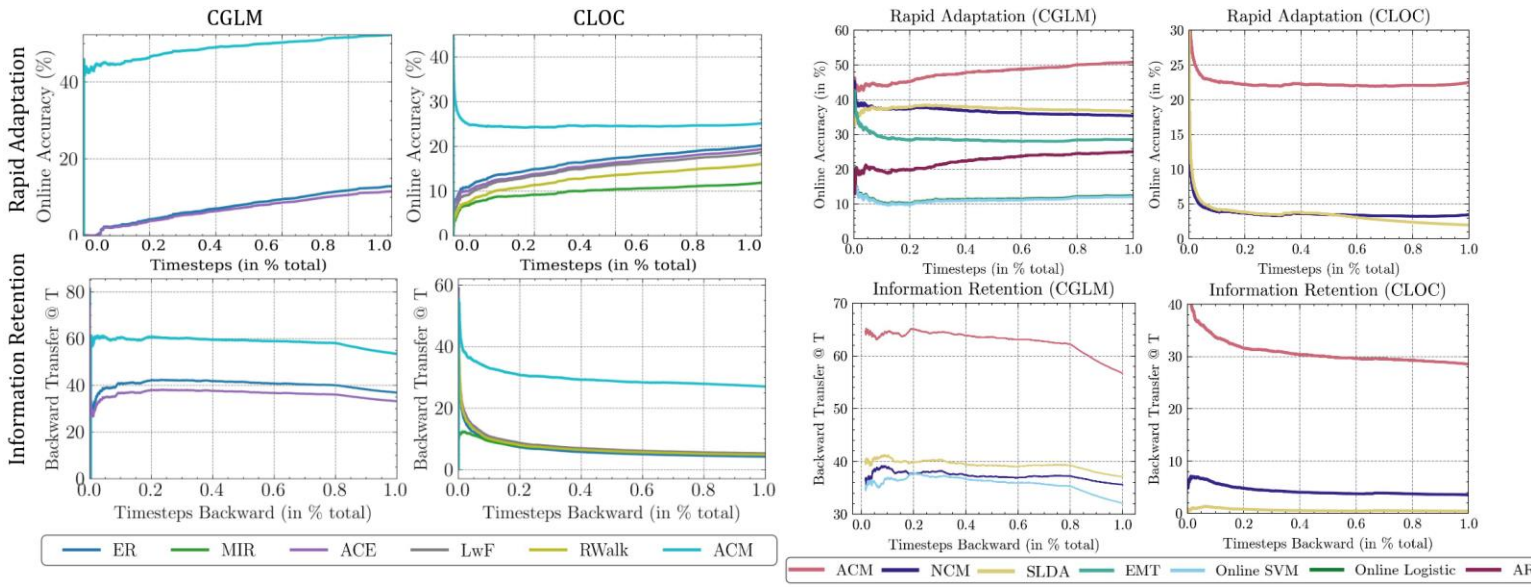
## Proposed Method (ACM)

A *memory* which is continual by-design: with addition, update & inference operations

- Use kNN on pretrained DNN features!

  > Consistency

  > Natural insert operation




## Conclusions

- Updating overwrites information in Deep Nets
- This causes forgetting
- Add Data to (Your!) Continual Data Structure
- ACM is just the first step:

A simple Continual Data Structure

ACM++ coming up soon, with learnable metric spaces

> *Vastly more* flexible!

> Can handle *far higher* representation drift

Part of Work: Online Continual Learning without the Storage Constraint? Arxiv

*Ameya Prabhu, Zhipeng Cai, Puneet Dokania, Philip Torr, Vladlen Koltun, Ozan Sener*

# Results: A Shocking Improvement!

## Aim

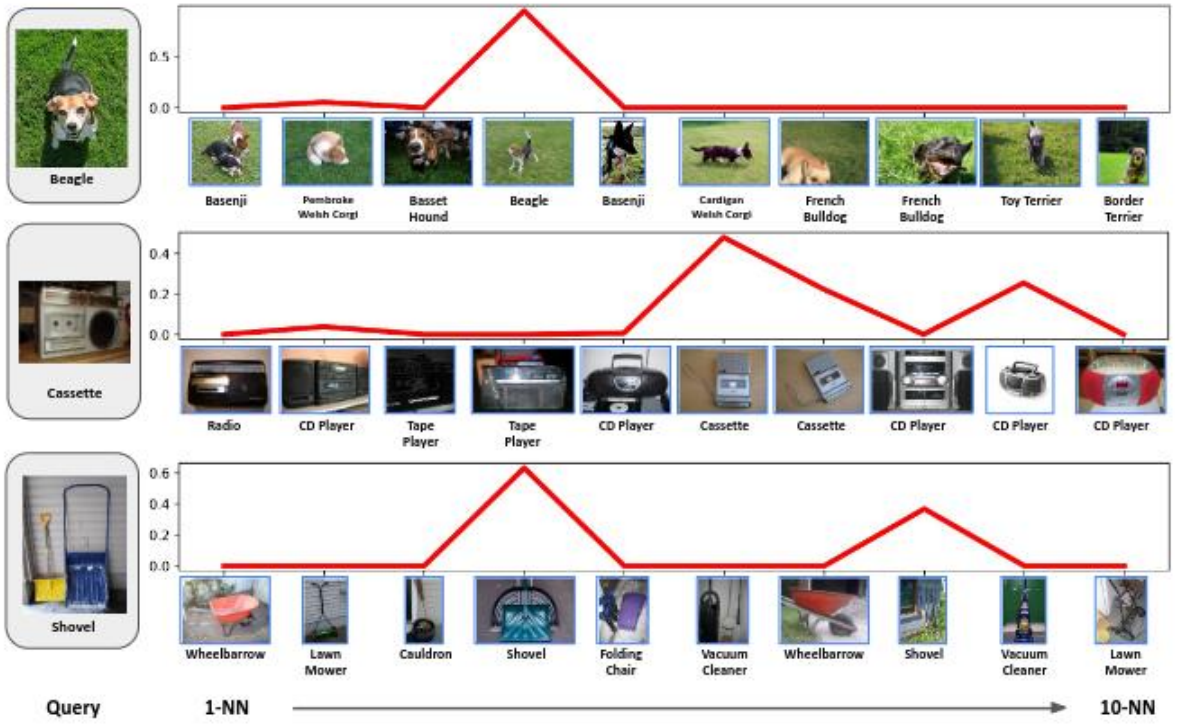
- Towards Retrieval Augmented Models

Does not really give the oomph off-the-shelf

Current Idea: kNN + transformer re-ranking

**Improving Image Recognition by Retrieving from Web-Scale Image-Text Data**

Ahmet Iscen Alireza Fathi Cordelia Schmid
Google Research



# Results: A Shocking Improvement!

## Aim

- Towards Retrieval Augmented Models

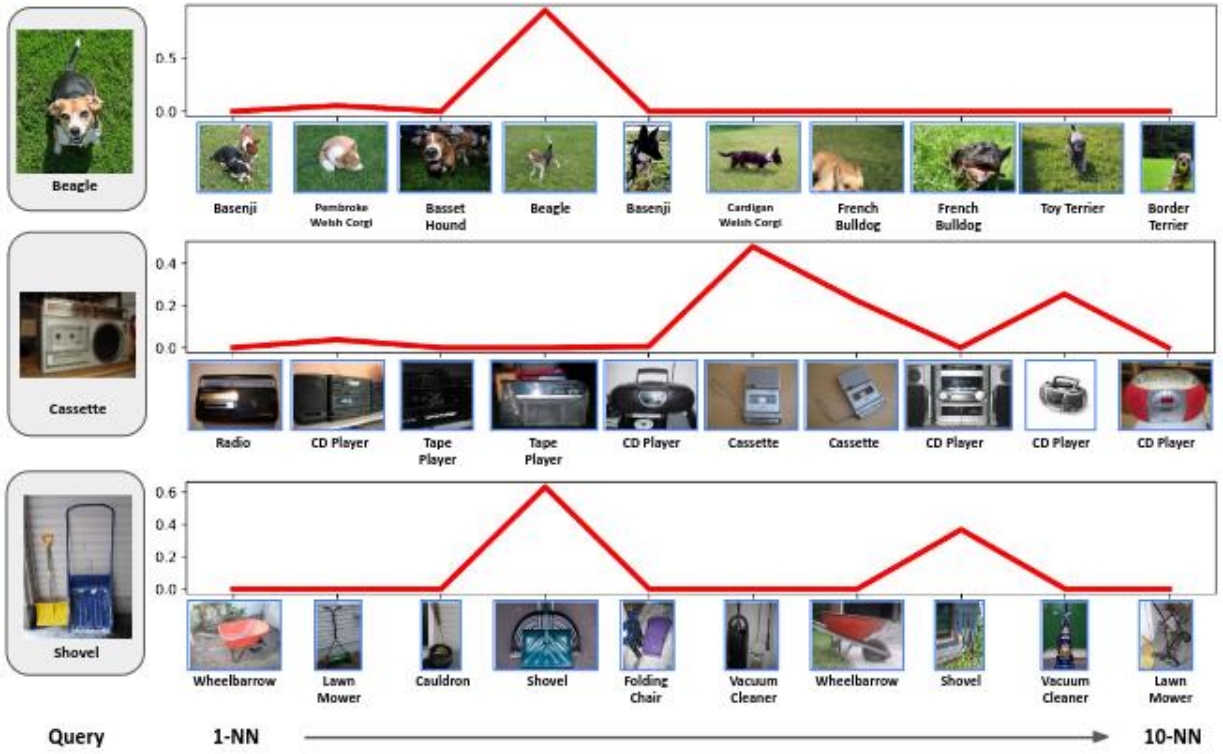
Current method: kNN + transformer re-ranking

Does not really give an impressive boost off-the-shelf
- Needs to get additional components right
- Extension to multiple layers in the transformer hard

| | | | ImageNet-LT | | | | Places-LT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Retrieval | Backbone | Many-shot | Mid-shot | Low-shot | All | Many-shot | Mid-shot | Low-shot | All |
| **BASELINES** | | | | | | | | | | |
| Linear Classifier | | ViT-B16 🔒 | 76.5 | 72.6 | 66.5 | 73.5 | 44.5 | 44.4 | 44.0 | 44.3 |
| MLP Classifier | | ViT-B16 🔒 | 80.1 | 74.1 | 66.9 | 75.2 | 48.6 | 46.1 | 41.3 | 46.0 |
| Mean $k$-NN | ✓ | ViT-B16 🔒 | 75.9 | 75.8 | **75.7** | 75.8 | 44.3 | 45.2 | 45.5 | 44.9 |
| **EXISTING METHODS** | | | | | | | | | | |
| PaCo [10] | | ResNext-101 | 68.2 | 58.7 | 41.0 | 60.0 | 36.1 | 47.9 | 35.3 | 41.2 |
| VL-LTR [43] | | ViT-B16 | 84.5 | 74.6 | 59.3 | 77.2 | **54.2** | 48.5 | 42.0 | 50.1 |
| RAC [29] | ✓ | ViT-B16 | - | - | - | - | 48.7 | 48.3 | 41.8 | 47.2 |
| RAC† [29] | ✓ | ViT-B16 🔒 | 80.9 | 76.0 | 67.5 | 76.7 | 50.3 | 48.3 | 42.5 | 47.9 |
| RAC† [29] | ✓ | ViT-B16 | **85.9** | 79.3 | 69.3 | 80.5 | 51.9 | 49.8 | 46.8 | 50.0 |
| **Ours** | ✓ | ViT-B16 🔒 | 80.6 | 77.5 | 74.5 | 78.3 | 50.9 | 49.9 | 47.5 | 49.9 |
| **Ours + FT** | ✓ | ViT-B16 | 85.4 | **81.5** | **76.4** | **82.3** | 52.4 | **52.0** | **48.5** | **51.4** |

**Improving Image Recognition by Retrieving from Web-Scale Image-Text Data**

Ahmet Iscen Alireza Fathi Cordelia Schmid
Google Research



# Results: A Shocking Improvement!

## Aim

- Towards Retrieval Augmented Models

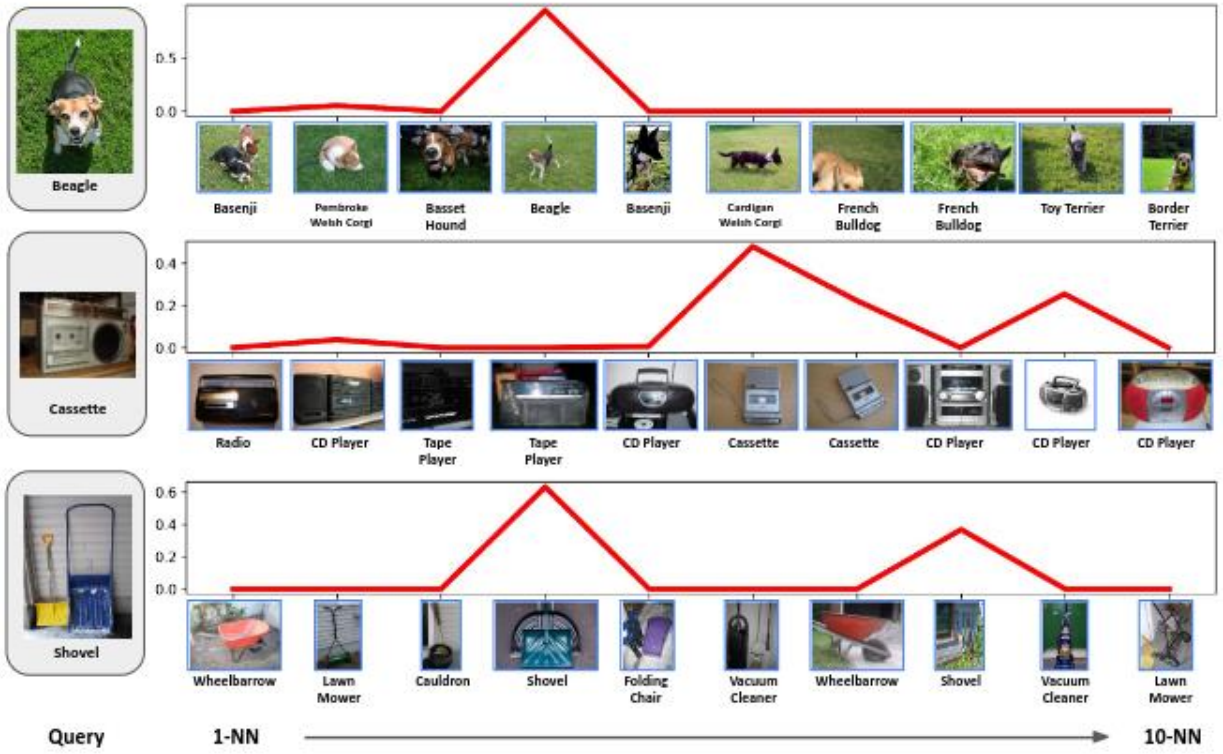
Current method: kNN + transformer re-ranking

Does not really give an impressive boost off-the-shelf
- Needs to get additional components right
- Extension to multiple layers in the transformer hard

| | | | ImageNet-LT | | | | Places-LT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Retrieval | Backbone | Many-shot | Mid-shot | Low-shot | All | Many-shot | Mid-shot | Low-shot | All |
| **BASELINES** | | | | | | | | | | |
| Linear Classifier | | ViT-B16 🔒 | 76.5 | 72.6 | 66.5 | 73.5 | 44.5 | 44.4 | 44.0 | 44.3 |
| MLP Classifier | | ViT-B16 🔒 | 80.1 | 74.1 | 66.9 | 75.2 | 48.6 | 46.1 | 41.3 | 46.0 |
| Mean $k$-NN | ✓ | ViT-B16 🔒 | 75.9 | 75.8 | **75.7** | 75.8 | 44.3 | 45.2 | 45.5 | 44.9 |
| **EXISTING METHODS** | | | | | | | | | | |
| PaCo [10] | | ResNext-101 | 68.2 | 58.7 | 41.0 | 60.0 | 36.1 | 47.9 | 35.3 | 41.2 |
| VL-LTR [43] | | ViT-B16 | 84.5 | 74.6 | 59.3 | 77.2 | **54.2** | 48.5 | 42.0 | 50.1 |
| RAC [29] | ✓ | ViT-B16 | - | - | - | - | 48.7 | 48.3 | 41.8 | 47.2 |
| RAC† [29] | ✓ | ViT-B16 🔒 | 80.9 | 76.0 | 67.5 | 76.7 | 50.3 | 48.3 | 42.5 | 47.9 |
| RAC† [29] | ✓ | ViT-B16 | **85.9** | 79.3 | 69.3 | 80.5 | 51.9 | 49.8 | 46.8 | 50.0 |
| **Ours** | ✓ | ViT-B16 🔒 | 80.6 | 77.5 | 74.5 | 78.3 | 50.9 | 49.9 | 47.5 | 49.9 |
| **Ours + FT** | ✓ | ViT-B16 | 85.4 | **81.5** | **76.4** | **82.3** | 52.4 | **52.0** | **48.5** | **51.4** |

**Improving Image Recognition by Retrieving from Web-Scale Image-Text Data**

Ahmet Iscen Alireza Fathi Cordelia Schmid
Google Research

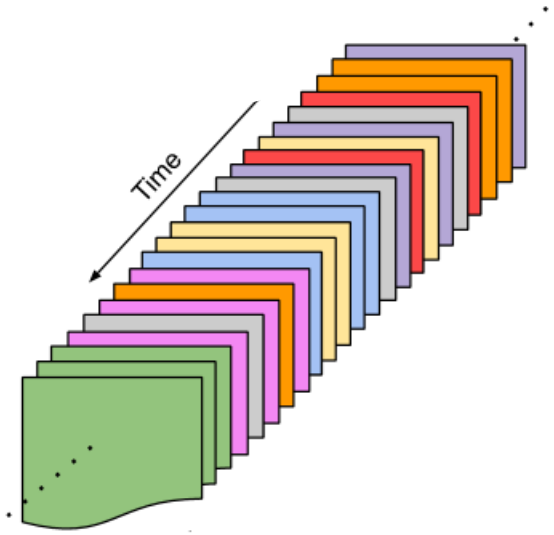


# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Studying Data Streams: Measuring Adaptation

Online CL

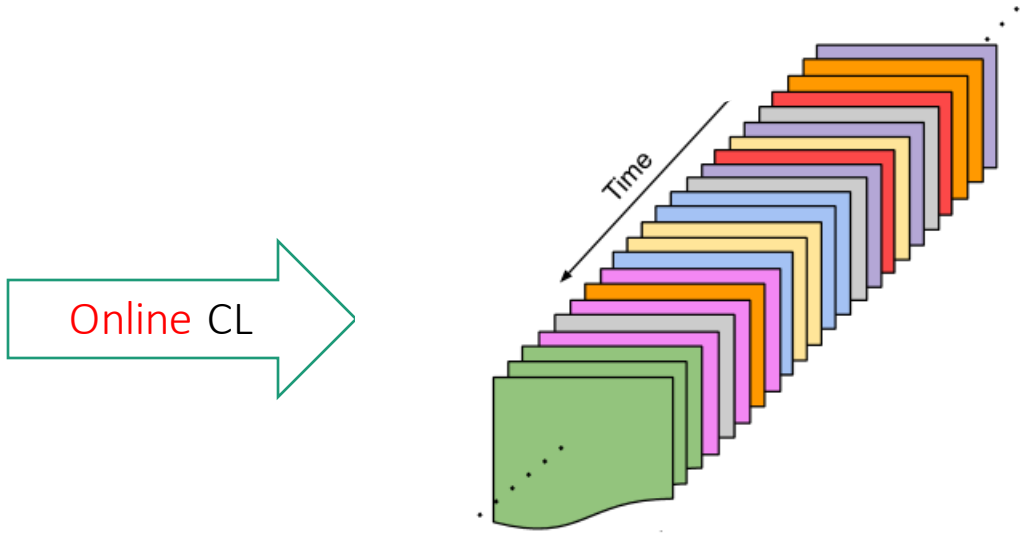



Does it Require Rapid Adaptation?

Yes => Time-Sensitive Model Updates
(Online CL)

- Must Measure Rapid Adaptation!

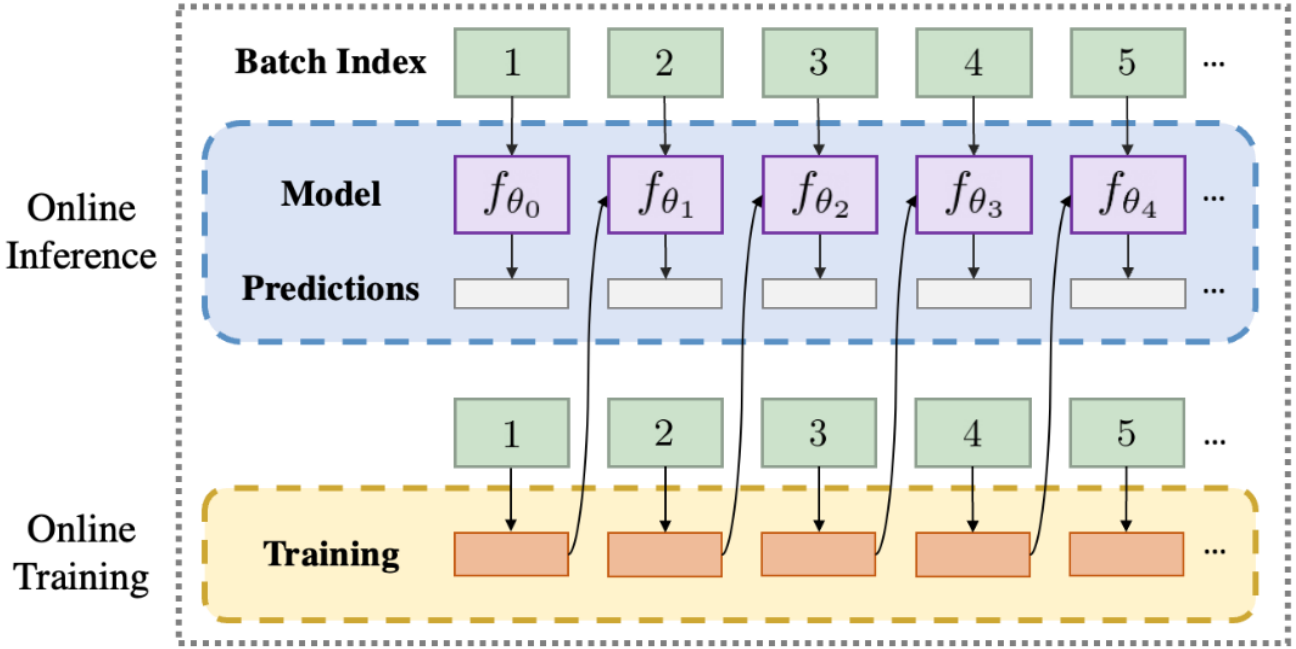
# Studying Data Streams: Measuring Adaptation




Does it Require Rapid Adaptation?

Yes => Time-Sensitive Model Updates (Online CL)

- Must Measure Rapid Adaptation!




Online accuracy

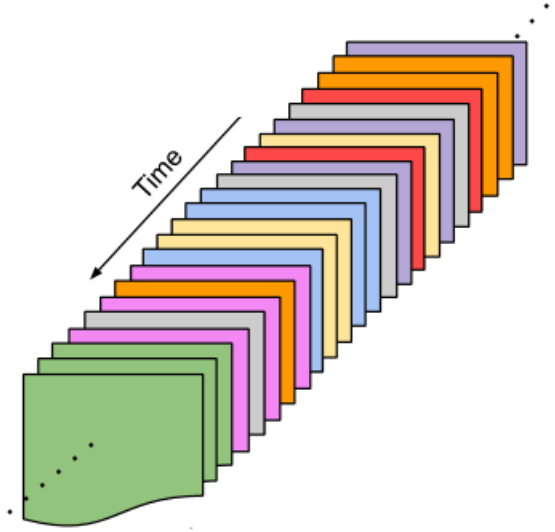
Measure of the model's performance on the next unseen sample/batch.

Hard to do in class-incremental setups

- Next samples are by design same class

# Studying Data Streams: Measuring Adaptation

Online CL




Online accuracy

Measure of the model's performance on the next unseen sample/batch.

Hard to do in class-incremental setups

- Next samples are by design same class

CLOC

- *Geolocation at scale*
- *713 classes, 39M images*
- *Simulates images arriving on a Flickr server.*

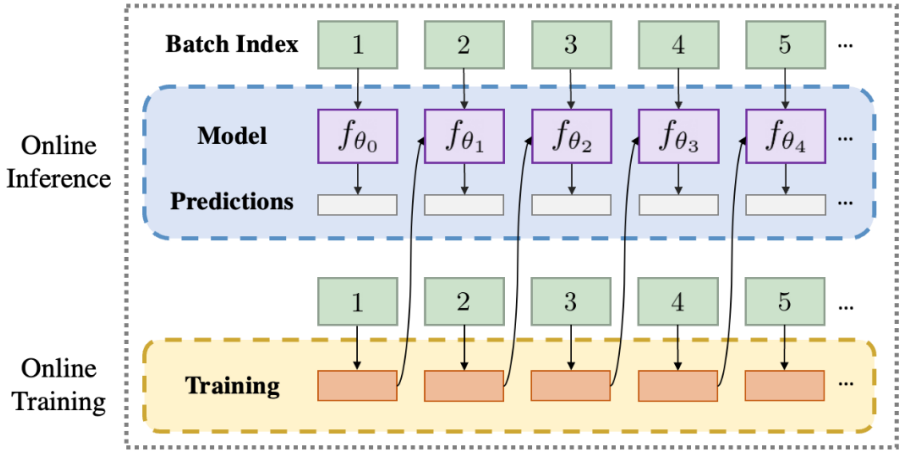
CGLM

- *Long-tailed landmark classification*
- *10,788 classes, 450K images*
- *Simulates arrival on a Wikimedia Commons server.*

# Insight: Incorrect Evaluation of Adaptation

Online accuracy

Measures model's performance on the next unseen sample/batch.



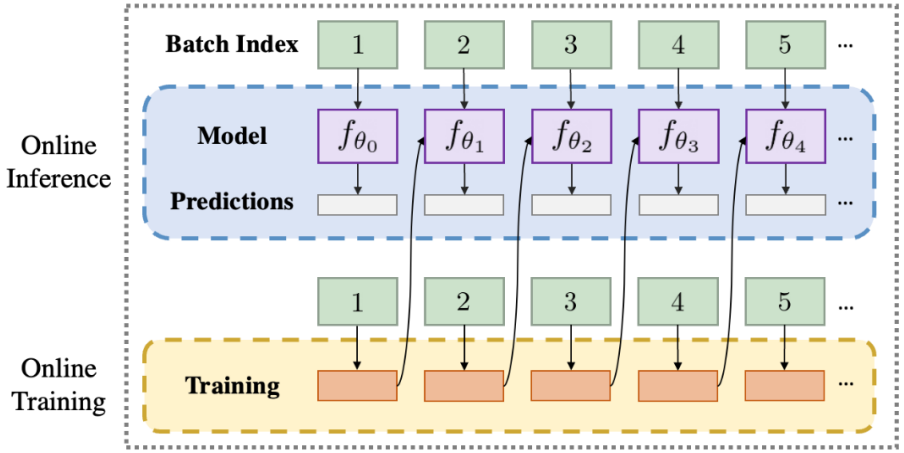

Finding: The stream labels are correlated in natural streams!

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

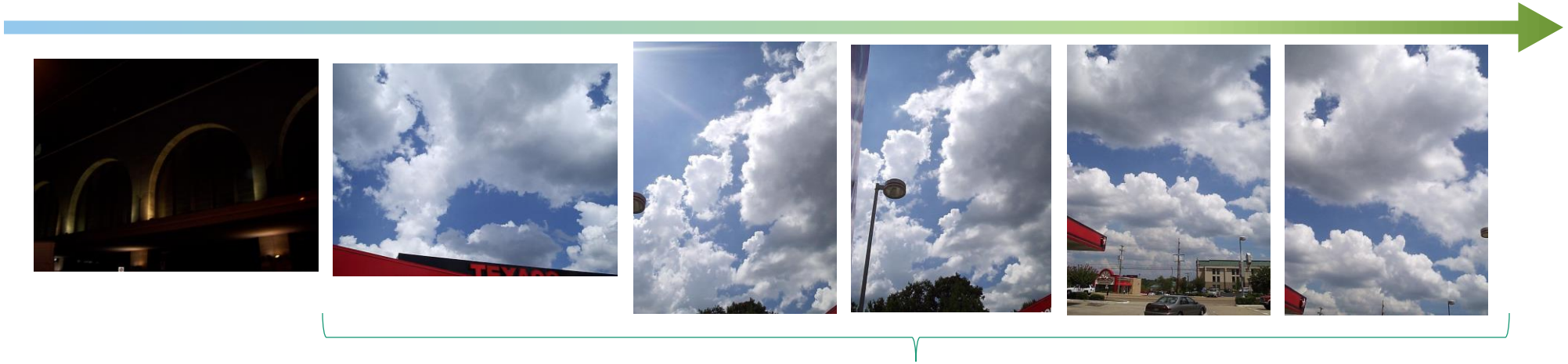
# Insight: Incorrect Evaluation of Adaptation

Online accuracy

Measures model's performance on the next unseen sample/batch.




Let us look at Real CLOC Samples!



Same label !

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Insight+: Why are Correlated Samples Important?

**Finding:** The stream labels are correlated in natural streams!

Why is it important?

## A Blind Classifier

*Blind Classifier: A model that predicts the mode of the last **K** samples seen without access to the input images.*

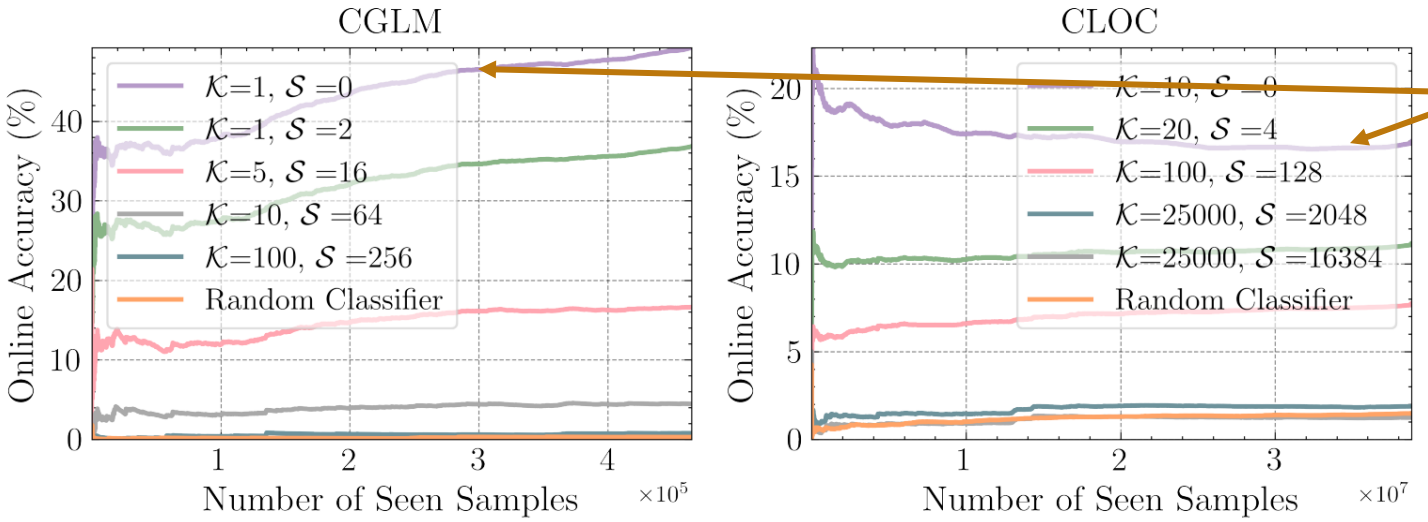
Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Insight+: Why are Correlated Samples Important?

**Finding:** The stream labels are correlated in natural streams!

Why is it important?

## A Blind Classifier has Great Performance




*Competitive accuracies to OCL methods!*

*It's unambiguously a bad model!*

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# OverAdapt: Hacking the Label Correlations

**Finding:** The stream labels are correlated in natural streams!

A Blind Classifier has Great Performance

OverAdapt: A simple and clearly *wrong* baseline!

A model made to overfit to the latest data by:

1. Adopting FIFO sampling to select training samples.
2. Fix a pretrained ResNet50 backbone and train linear layer only.

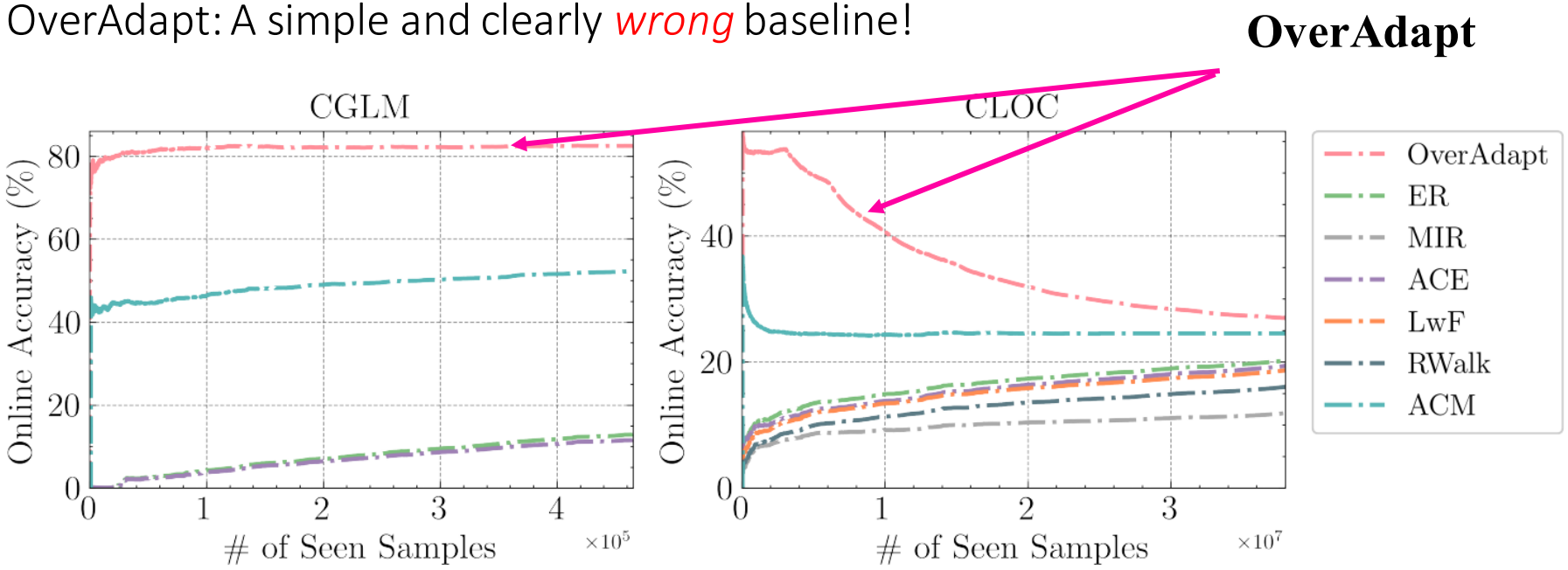
Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# OverAdapt: Hacking the Label Correlations

**Finding:** The stream labels are correlated in natural streams!

A Blind Classifier has Great Performance

- OverAdapt: A simple and clearly *wrong* baseline!




- Online Accuracy favours methods which overfit to incoming data!
- Incentives bad algorithms!

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Recent Methods which Purely Overfit

**Finding:** The stream labels are correlated in natural streams!

A Blind Classifier has Great Performance

**Kalman Filter for Online Classification of Non-Stationary Data**


**Michalis K. Titsias*** Google DeepMind mtitsias@google.com

**Alexandre Galashov*** Google DeepMind agalashov@google.com

**Amal Rannen-Triki** Google DeepMind arannen@google.com

**Razvan Pascanu** Google DeepMind razp@google.com

**Yee Whye Teh** Google DeepMind ywteh@google.com

**Jörg Bornschein** Google DeepMind bornschein@google.com


CoLLAs

Conference Papers Journal Track Papers

Low-rank extended Kalman filtering for online learning of neural networks from streaming data


Peter G. Chang, Gerardo Durán-Martín, Alex Shestopaloff, Matt Jones, Kevin Patrick Murphy



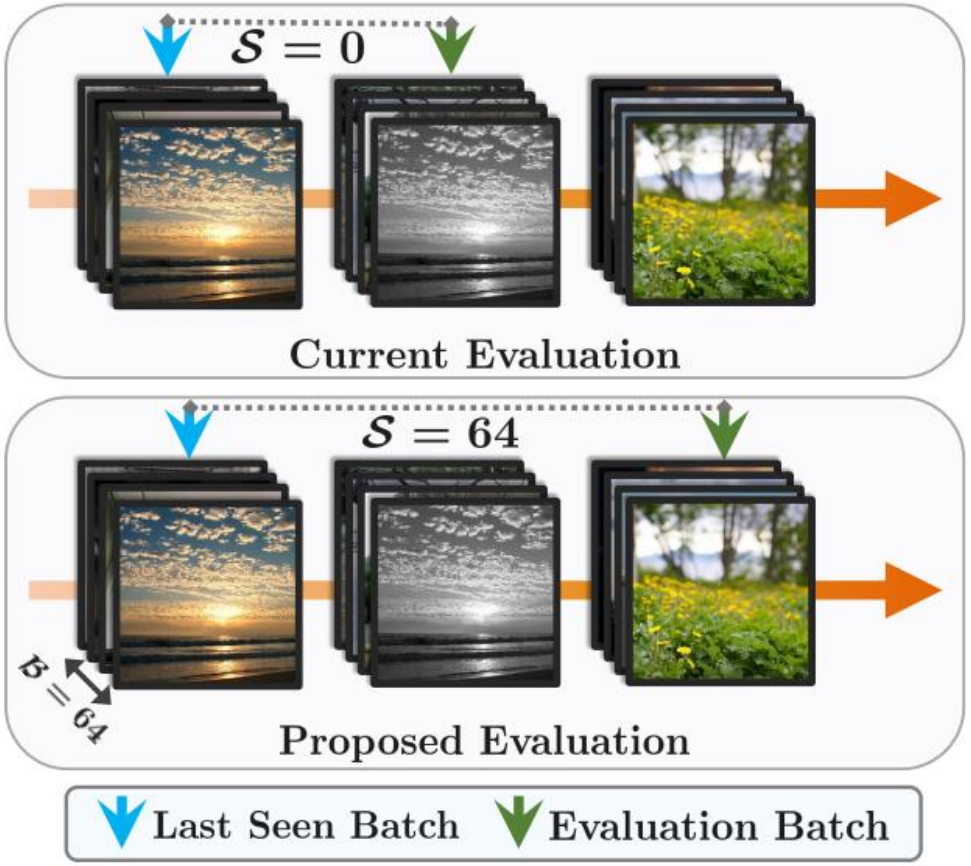
Keywords: Bayesian inference, online learning, extended Kalman filter, deep neural networks, non-stationary distributions, continual learning


Abstract Paper

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Proposed Solution: Near-Future Accuracy

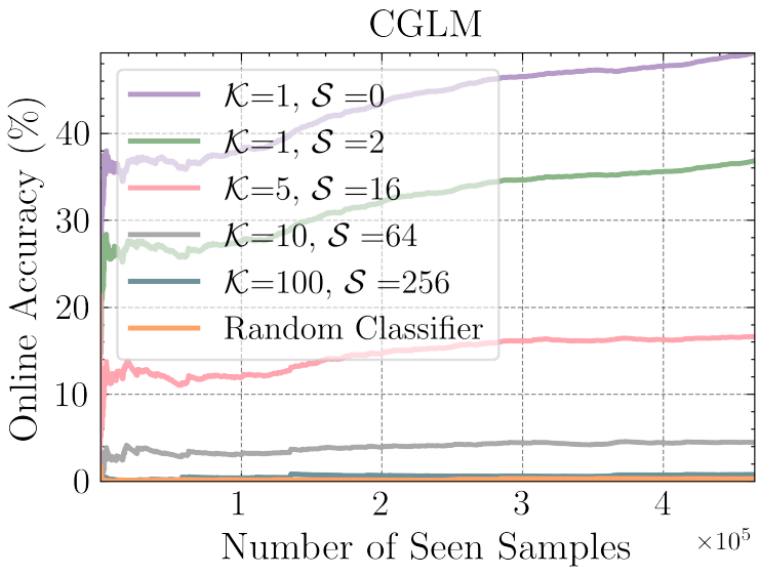
- Instead of measuring the accuracy on the immediate next batch how about we measure the accuracy on the next uncorrelated batch?
- **Question:** How do we estimate the batches to delay our evaluation with?

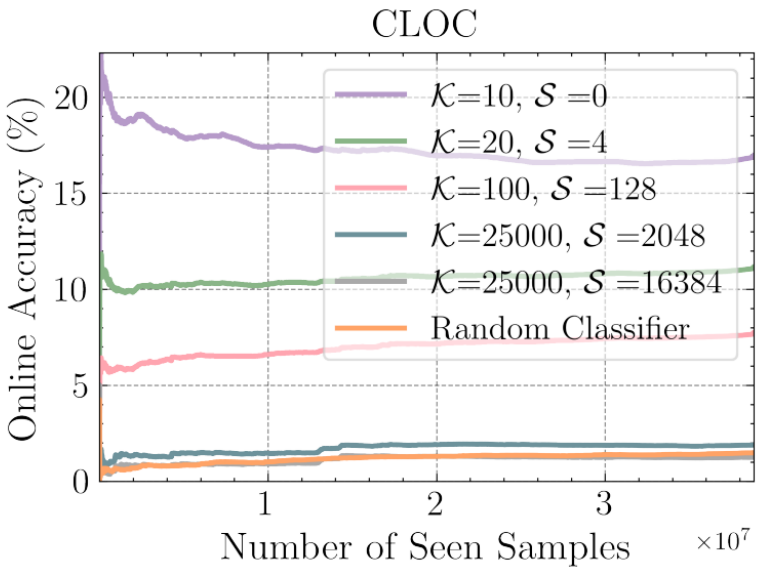



Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Near-Future Accuracy: Blind Classifier to Rescue

- **Question:** How do we estimate the batches to delay our evaluation with?

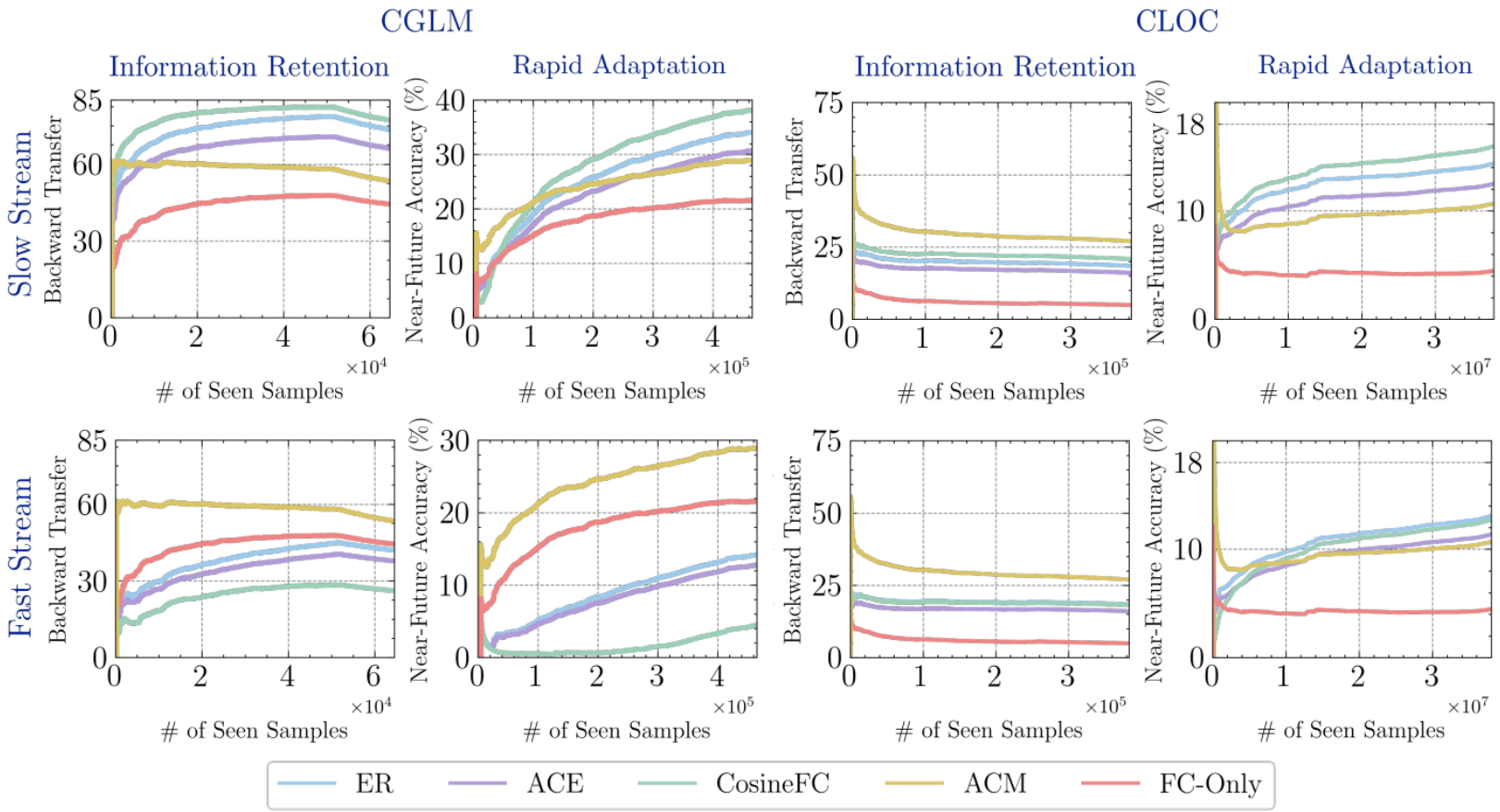



*Delay the Blind Classifier evaluation just until it converges to a random classifier!*

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Near-Future Accuracy: Benchmark




***Fast Stream (1 step):*** *The best thing to do is use* ***ACM****, freeze the backbone and use a kNN as your classifier.*

***Slow Stream (5/10 steps):*** *The best thing to do is to use* ***offline continual learning*** *approaches (randomly sample past data ignoring the latest with ER with Cosine FC layer).*

Hammoud et. al., Rapid Adaptation in Online Continual Learning: Are We Evaluating It Right?, ICCV' 23

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Studying Data Streams: Where is Training Data?

Continual Manual Annotation is costly and time-consuming—huge problem!

- CLEAR10 required $4500 for 30K

# Alternate Task: Going from Categories to Classifier

1. The data stream, $\mathcal{S}$, presents a set of categories, $\mathcal{Y}_t$, to be learned.

# Alternate Task: Going from Categories to Classifier

1. The data stream, $\mathcal{S}$, presents a set of categories, $\mathcal{Y}_t$, to be learned.
2. Under a given computational budget, $\mathcal{C}_t$, the classifier $f_{\theta_{t-1}}$ is updated to $f_{\theta_t}$.

Can use any public data/model assistance to do this! E.g. GPT3, DALL-E, LAION5B

Task: Go from Category List to Trained Classifier within minutes *continually*

# Alternate Task: Going from Categories to Classifier

1. The data stream, $\mathcal{S}$, presents a set of categories, $\mathcal{Y}_t$, to be learned.
2. Under a given computational budget, $\mathcal{C}_t$, the classifier $f_{\theta_{t-1}}$ is updated to $f_{\theta_t}$.
3. To evaluate the learner, the stream $\mathcal{S}$ presents test samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ with $y_i$ belonging to the collective set $\bigcup_{i=1}^{t} \mathcal{Y}_i$.

Task: Go from Category List to Trained Classifier within minutes *continually*

# How to Get Around Manual Data Annotation?

Manual Annotation

Stable Diffusion

Retrieval from Source

OpenAI

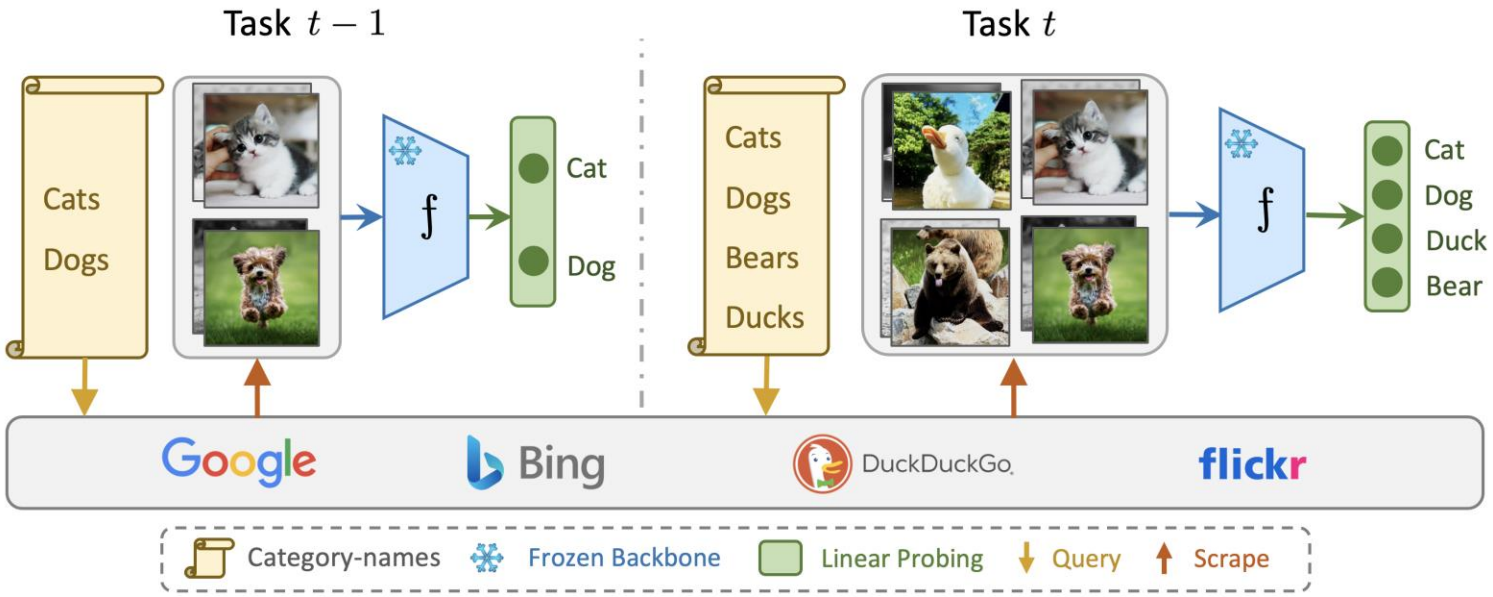
DALL-E

Internet: Billions of images uploaded **each day**

*Static* datasets are miniscule and out-of-date in comparison to the Internet!

# Internet as a Huge, Continually Evolving Train Set

Idea: Download data from searching the Internet! [Fergus, et.al. 2005]




We can go from Category List to Trained Classifier within minutes *continually!*

# Internet as a Huge, Continually Evolving Train Set

Is it better than using Stable Diffusion or Retrieval From LAION5B?

Table 3: **Comparison with Name-Only Classification Techniques with ResNet50:** When comparing with existing state-of-the-art name-only classification techniques, we show that our method outperforms those methods by margins ranging from 2% to 25%.

| Type | Method | Model | Birdsnap | Aircraft | Flowers | Pets | Cars | DTD |
|---|---|---|---|---|---|---|---|---|
| Data-Free | CLIP-ZS (Radford et al., 2021) | CLIP | 32.6 | 19.3 | 65.9 | 85.4 | 55.8 | 41.7 |
| | CaFo-ZS (Zhang et al., 2023) | CLIP | - | 17.3 | 66.1 | 85.8 | 55.6 | 50.3 |
| | CALIP (Guo et al., 2023) | CLIP | - | 17.8 | 66.4 | 86.2 | 56.3 | 42.4 |
| | CLIP-DN (Zhou et al., 2023) | CLIP | 31.2 | 17.4 | 63.3 | 81.9 | 56.6 | 41.2 |
| | CuPL (Pratt et al., 2023) | CLIP | 35.8 | 19.3 | 65.9 | 85.1 | 57.2 | 47.5 |
| | VisDesc (Menon & Vondrick, 2022) | CLIP | 35.7 | 16.3 | 65.4 | 82.4 | 54.8 | 42.0 |
| | SD-Clf (Li et al., 2023a) | SD-2.0 | - | 26.4 | 66.3 | 87.3 | - | - |
| Use-Data | GLIDE-Syn (He et al., 2022) | CLIP | 38.1 | 22.0 | 67.1 | 86.8 | 56.9 | 43.2 |
| | CaFo (Zhang et al., 2023) | CLIP | - | 21.1 | 66.5 | 87.5 | 58.5 | 50.2 |
| | SuS-X-LC (Udandarao et al., 2023) | CLIP | 38.5 | 21.1 | 67.1 | 86.6 | 57.3 | 50.6 |
| | SuS-X-SD (Udandarao et al., 2023) | CLIP | 37.1 | 19.5 | 67.7 | 85.3 | 57.2 | 49.2 |
| | C2C (Ours-Linear Probe) | CLIP | 48.1 (+9.6) | 44.0 (+22.0) | 82.0 (+14.3) | 88.1 (+0.6) | 71.3 (+12.8) | 57.1 (+6.5) |
| | C2C (Ours-MLP Adapter) | CLIP | 46.6 (+8.1) | 48.9 (+26.9) | 84.8 (+17.1) | 89.4 (+1.9) | 72.6 (+14.1) | 57.6 (+7.0) |
| | C2C (Ours-Linear Probe) | MocoV3 | 56.1 (+17.6) | 57.5 (+35.5) | 85.7 (+18.0) | 91.7 (+4.2) | 62.1 (+3.6) | 54.6 (+4.0) |
| | C2C (Ours-MLP Adapter) | MocoV3 | 53.7 (+15.2) | 65.5 (+43.5) | 87.1 (+19.4) | 92.8 (+5.3) | 66.8 (+8.3) | 55.8 (+5.2) |

Better Prompts

SD/ LAION

Using Internet is Vastly Better!

# Future: Never Ending Language Learning

## Never-Ending Learning


T. Mitchell*, W. Cohen*, E. Hruschka†¶, P. Talukdar‡¶, J. Betteridge*, A. Carlson§¶, B. Dalvi*, M. Gardner*, B. Kisiel*, J. Krishnamurthy*, N. Lao§¶, K. Mazaitis*, T. Mohammad¶, N. Nakashole*, E. Platanios*, A. Ritter‖¶, M. Samadi*, B. Settles**¶, R. Wang¶, D. Wijaya*, A. Gupta*, X. Chen*, A. Saparov*, M. Greaves††¶, J. Welling‡‡

tom.mitchell@cs.cmu.edu

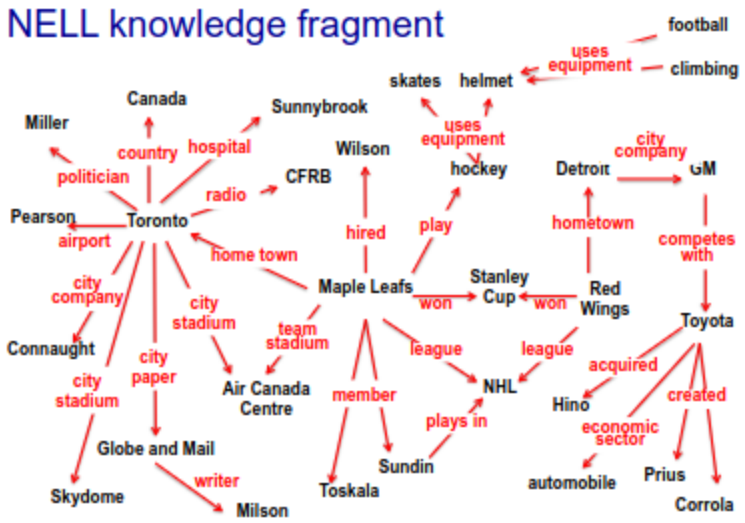




Figure 1: **Fragment of the 80 million beliefs NELL has read from the web.** Each edge represents a belief triple (e.g., play(MapleLeafs, hockey), with an associated confidence and provenance not shown here. This figure contains only correct beliefs from NELL's KB – it has many incorrect beliefs as well since NELL is still learning.

# Outline

- My Notion of Continual Learning: Learning from a Firehose
- Gaps of the Past: Continual Learning is an Efficiency Problem!
- Gaps of the Past: Why Continual Learning Methods Fail?
- Solution: Rethinking Assumptions in Continual Learning
- My Problem Setup: Online vs Offline Continual Learning
- My Problem Setup: Scenarios, Distributional Assumptions, Datasets
- Insight: Efficiently Updating the Model on New Data is Harder than We Think!
- Core Issue: Stability Gap
- A Radical Solution: Continual Data Structures
- Studying Data Streams: Idiosyncrasies of Data
- Studying Data Streams: Where is the Training Data?
- Other Open Problems!

# Thank You!

Questions?

# Extra Details: Why Scenarios as Constraints is Bad

- My Notion of Continual Learning: Learning from a Firehose
- Aligning Research with Practice: Useable Continual Learning
- Assumptions: The Good and Bad
  - Good: Online vs Offline Learning
  - Good: Compute Budget
  - Bad: "Scenarios" as Constraints and Small Datasets
  - Bad: Memory Constraints
  - Bad: No/Incorrect Evaluation of Online Continual Learning
  - Bad: Assumptions of Labeled Training Datasets
- Continual Learning: Distributional Generalization to Data Structures
- From Distributional Generalization to Continual Data Structures

# The Bad: "Scenarios" as Constraints

A useful way to categorize continual learning problems is based on:

How the aspect of data that changes over time relates to the mapping to be learned

-Tinne, CoLLAs 2022

# The Bad: "Scenarios" as Constraints

A useful way to categorize continual learning problems is based on:

How the aspect of data that changes over time relates to the mapping to be learned

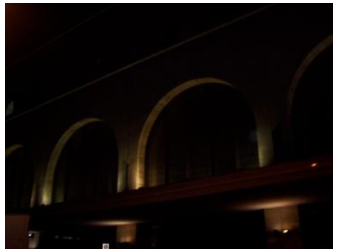
“Class Incremental Learning” and "Domain Incremental Learning" being the two scenarios.

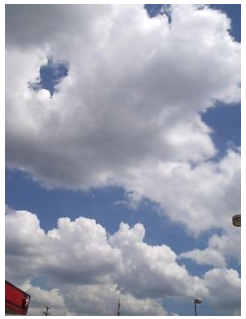
# The Bad: "Scenarios" as Constraints

A useful way to categorize continual learning problems is based on:
How the aspect of data that changes over time relates to the mapping to be learned

"Class Incremental Learning" and "Domain Incremental Learning" being the two scenarios.

Eg: CLOC Dataset (Geolocation at scale), 39 Million Samples, Images Ordered by Flickr Timestamps



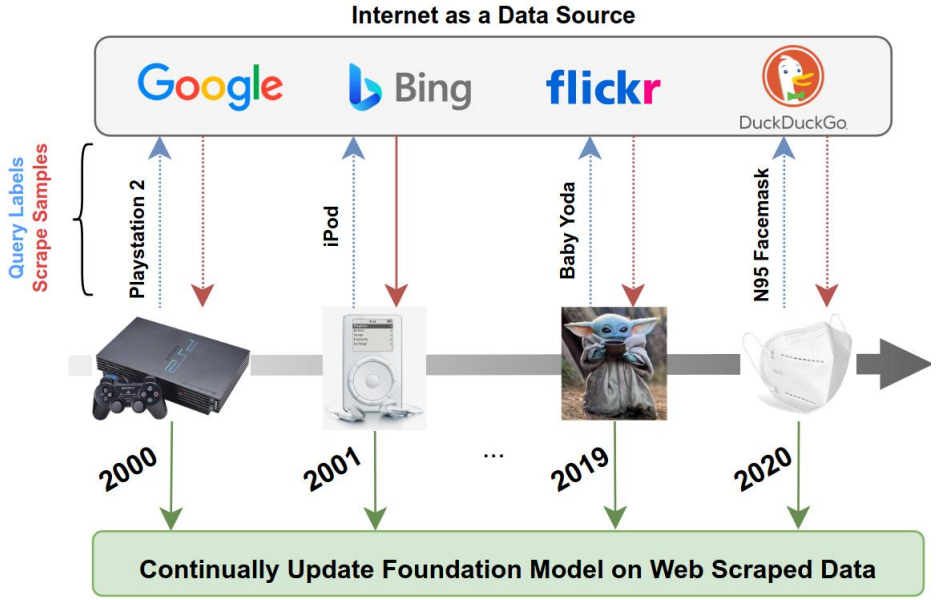


...

# The Bad: "Scenarios" as Constraints

A useful way to categorize continual learning problems is based on:
How the aspect of data that changes over time relates to the mapping to be learned

"Class Incremental Learning" and "Domain Incremental Learning" being the two scenarios.




FROM CATEGORIES TO CLASSIFIER: NAME-ONLY CONTINUAL LEARNING BY EXPLORING THE WEB

Ameya Prabhu[1*] Hasan Abed Al Kader Hammoud[2*] Ser-Nam Lim[3] Bernard Ghanem[2]
Philip H.S. Torr[1] Adel Bibi[1]
[1]University of Oxford [2]KAUST [3]Meta AI

Available on my github page!

# The Bad: "Scenarios" as Constraints

A useful way to categorize continual learning problems is based on:
How the aspect of data that changes over time relates to the mapping to be learned

Assumption of “Class Incremental Learning” or "Domain Incremental Learning" as the two scenarios.

Goal is always to learn a $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$ over a growing label space $\mathbf{y}$ which minimizes:

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y} \sim \cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y})$$

Question (For Discussion): Correct/Refine this!

# The Bad: "Scenarios" as Constraints

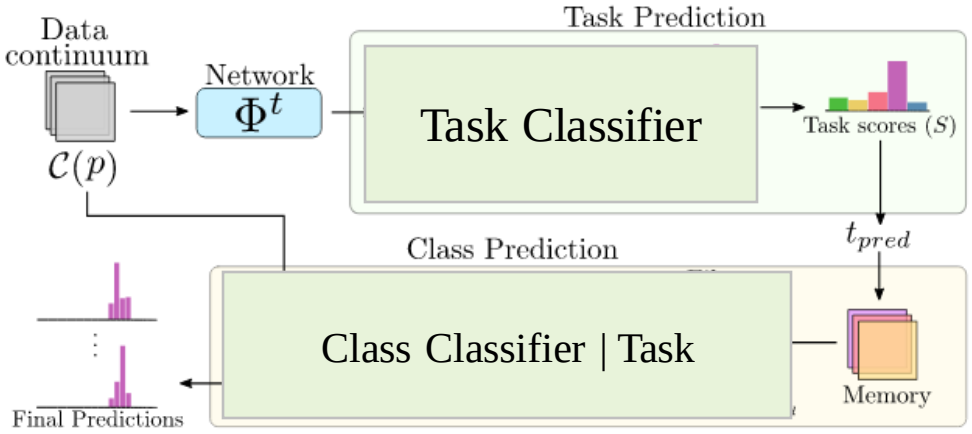
Assumption of “Class/Domain Incremental Learning” as a Scenario.

Why is this bad and important?

Example: Continual CIFAR10

Two-Tier Continual Algorithms: E.g. Architecture growing, meta-learning, prompt tuning, etc.

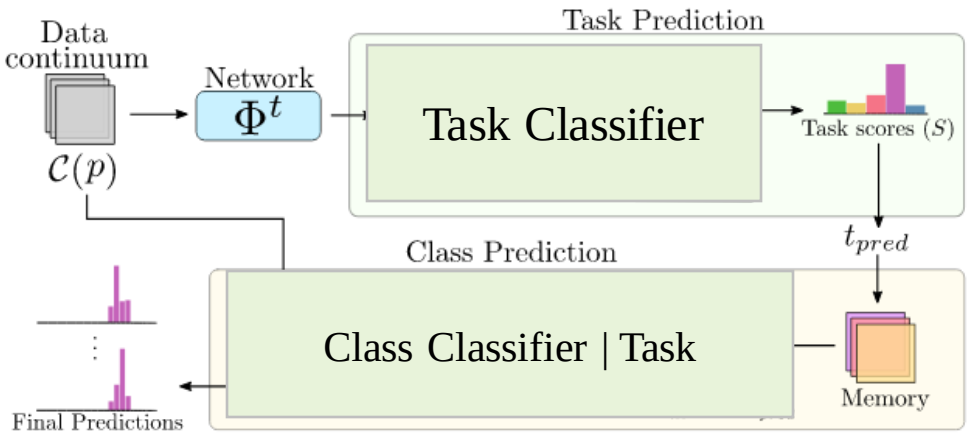
1) Predict the task-id
2) Then predict the class



# The Bad: "Scenarios" as Constraints

Two-Tier Continual Algorithms! E.g. Architecture growing, meta-learning, prompt tuning, etc.

1) Predict the task-id
2) Then predict the class

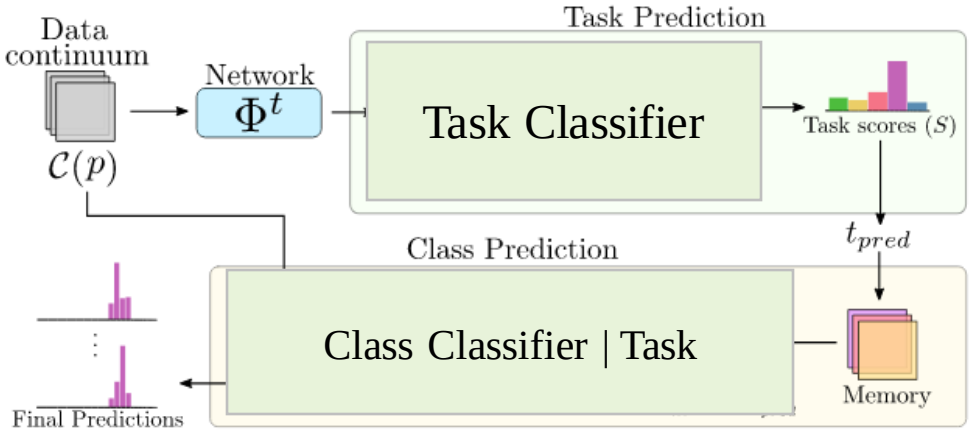



Works in class-incremental edge case!

10 classes -> 5 task classification {Task1, Task2, Task3, Task4, Task5} -> 2 class per task {cls1, cls2}

# The Bad: "Scenarios" as Constraints

Two-Tier Continual Algorithms: 1) Predict the task-id and 2) Then predict the class




Another Edge Case: Continual IID classification

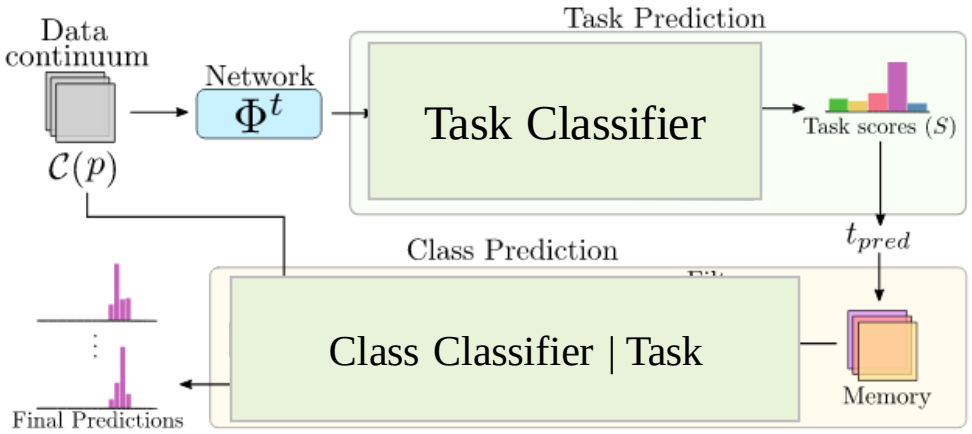
Idea: More data over time, higher #samples per class, better performance continually!

# The Bad: "Tasks" as Constraints

Two-Tier Continual Algorithms: 1) Predict the task-id and 2) Then predict the class

Another Edge Case: Continual IID Classification

Idea: More data over time, higher #samples per class, better performance continually!




5 tasks all have same classes!

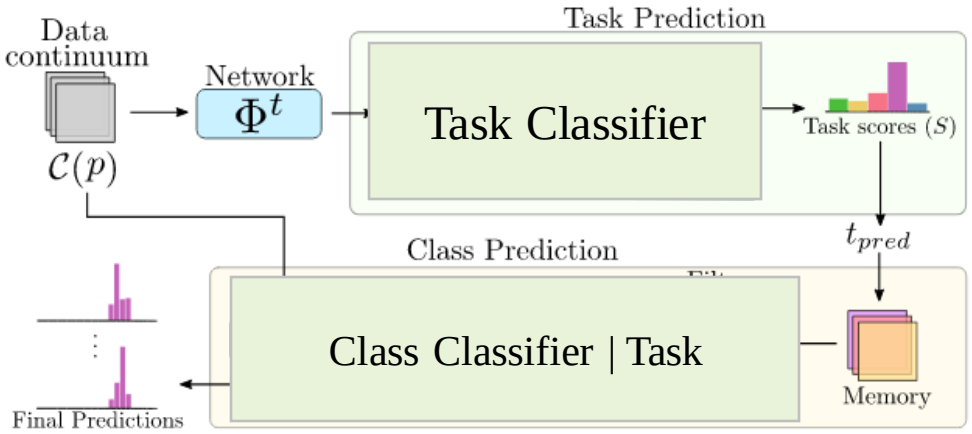
Task-classifier provably achieves random performance!

# The Bad: "Tasks" as Constraints

Two-Tier Continual Algorithms: 1) Predict the task-id and 2) Then predict the class

Another Edge Case: Continual IID Classification

Idea: More data over time, higher #samples per class, better performance continually!




5 tasks all have same classes!

Task-classifier provably achieves random performance!

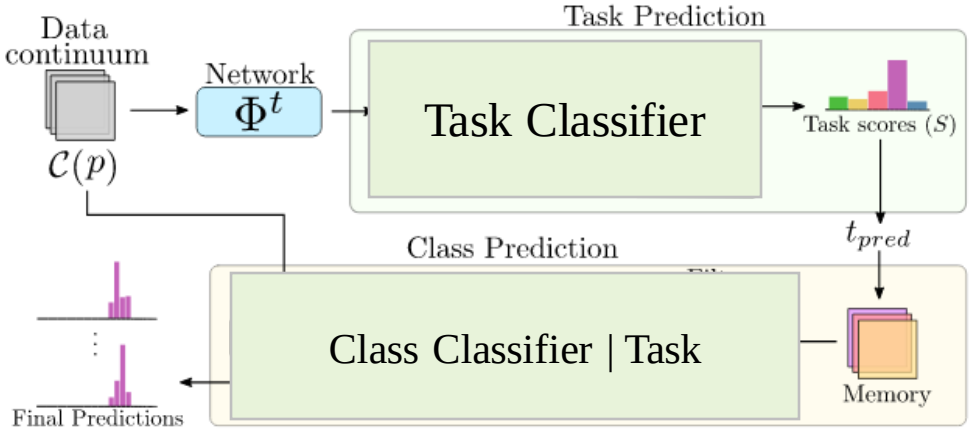
Far smaller network doing the complete 10-class classification at each step

# The Bad: "Tasks" as Constraints

Two-Tier Continual Algorithms: 1) Predict the task-id and 2) Then predict the class

Another Edge Case: Continual IID Classification

Idea: More data over time, higher #samples per class, better performance continually!




5 tasks all have same classes!

Task-classifier provably achieves random performance!

Far smaller network doing the complete 10-class classification at each step

<u>Bad Assumption</u>: Better to not have hard distributional constraints on data stream

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Question: Have we seen *any* evidence of positive backward transfer?

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Question: Have we seen *any* evidence of positive backward transfer?

Why? We need to be better than a Mixture of Experts baseline!

1) Finetune the last trained network/pretrained model
2) Save a model after every task
3) At inference time, use the model corresponding to task

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Question: Have we seen *any* evidence of positive backward transfer?

Why? We need to be better than a Mixture of Experts baseline!

1) Finetune the last trained network/pretrained model
2) Save a model after every task
3) At inference time, use the model corresponding to task

- Inference not expensive: Samples go through only one model
- Mixture of Experts used extensively in large-scale systems like GPT4

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Question: Have we seen *any* evidence of positive backward transfer?

Why? We need to be better than a Mixture of Experts baseline!

- Inference not expensive: Samples go through only one model
- Mixture of Experts used extensively in large-scale systems like GPT4

Note: I do not believe/say that Task-Incremental Learning is not useful!

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Note: I do not believe/say that Task-Incremental Learning is not useful! Why?

# Question: What About Task-Incremental?

- Task-incremental learning *(Task-IL)*
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks

Note: I do not believe/say that Task-Incremental Learning is not useful! Why?

**Drinking from a Firehose: Continual Learning with Web-scale Natural Language**


**Hexiang Hu** HEXIANGH@USC.EDU
*University of Southern California*

**Ozan Sener** OZAN.SENER@INTEL.COM
*Intel Labs*

**Fei Sha** FEISHA@USC.EDU
*University of Southern California*

**Vladlen Koltun** VLADLEN.KOLTUN@INTEL.COM
*Intel Labs*


Task: User Personalization

Goal: Personalized Language Modelling

✓ Positive Transfer between Users

Why? What people speak depends on time!
Common-backbone + conditioning useful

# Outline: Why Memory Constraint is Bad

- My Notion of Continual Learning: Learning from a Firehose
- Aligning Research with Practice: Useable Continual Learning
- Assumptions: The Good and Bad
  - Good: Online vs Offline Learning
  - Good: Compute Budget
  - Bad: "Scenarios" Constraints and Small Datasets
  - Bad: Memory Constraints
  - Bad: No/Incorrect Evaluation of Online Continual Learning
  - Bad: Assumptions of Labelled Training Datasets
- Continual Learning: Distributional Generalization to Data Structures
- From Distributional Generalization to Continual Data Structures

# The Bad: Memory Constraints

No valid justification for “tiny memory”

Prior art limits memory in continual learning because:

- Cost: Storage is Expensive
- Privacy: Users can Delete Data

# Argument #1: Cost (Memory Constraint)

Table 1: The cost of storing data has decreased rapidly, allowing the storage of a large dataset for a negligible cost compared to the cost of computation for both cloud and mobile systems.

| Cloud Storage | 1987 | 1997 | 2007 | 2017 |
|---|---|---|---|---|
| $/MB | 83.33 | 0.22 | 0.0003 | 0.00002 |
| Cost of storing CLOC ($) | 350M | 920K | 1250 | 83 |
| Training Cost (ER) | >2000$ | | | |

| Mobile NAND | 2000 | 2005 | 2010 | 2017 | 2022 |
|---|---|---|---|---|---|
| $/MB | 1100 | 85 | 1.83 | 0.25 | 0.03 |
| Cost of storing CLOC ($) | 4M | 300K | 6350 | 850 | 70 |
| Training Cost (ER) | Out of Memory (O.O.M.) | | | | |

Storage is cheap even for datasets 39x Imagenet, most costs come from training!

Prabhu et. al., Online Continual Learning Without the Storage Constraint, Arxiv

# Argument #1: Cost (Memory Constraint)

Table 1: The cost of storing data has decreased rapidly, allowing the storage of a large dataset for a negligible cost compared to the cost of computation for both cloud and mobile systems.

| Cloud Storage | 1987 | 1997 | 2007 | 2017 |
|---|---|---|---|---|
| $/MB | 83.33 | 0.22 | 0.0003 | 0.00002 |
| Cost of storing CLOC ($) | 350M | 920K | 1250 | 83 |
| Training Cost (ER) | >2000$ | | | |

| Mobile NAND | 2000 | 2005 | 2010 | 2017 | 2022 |
|---|---|---|---|---|---|
| $/MB | 1100 | 85 | 1.83 | 0.25 | 0.03 |
| Cost of storing CLOC ($) | 4M | 300K | 6350 | 850 | 70 |
| Training Cost (ER) | Out of Memory (O.O.M.) | | | | |

| Ref. | Dataset | Ordering | Memory | Cost | Iters | Cost |
|---|---|---|---|---|---|---|
| [2] | CIFAR10 | Cls Inc | 1-25MB | 0.05¢ | 250K-375K | 20$ |
| [12] | CIFAR100 | Cls Inc | 10 MB | 0.02¢ | 50K | 8$ |
| [5,9] | | | | | 125K | 15$ |
| [2] | TinyImageNet | Cls Inc | 5-20 MB | 0.04¢ | 350K-500K | 25$ |
| [5,9] | ImageNet100 | Cls Inc | 0.3-1 GB | 2¢ | 100K | 50$ |
| [5,9] | ImageNet1K | Cls Inc | 33GB | 66¢ | 1M | 500$ |
| [10] | CLEAR | Dist Shift | 0.4-1.2GB | 2¢ | 300K | 100$ |
| [8] | ResNet50 (bs=256) | | 22GB | | | |
| Ours | GLDv2-CL | Dist Shift | **90GB** | 2$ | 2K | 10$ |
| | ImageNet21K-CL | ClsInc, DataInc | **400GB** | 10$ | 8K | 35$ |

Table 1. **Cost of Memory vs Computation.** Google Cloud Standard Storage (2¢per GB per month for 1 month) and Compute Cost measured as running cost of an A2 instance (3$ per hour for 1 GPU). Number of iterations (forward+backward passes) for training a CL model on that dataset listed for comparison invariant to input image and model sizes. We observe that computational costs for running an experiment far outweigh the costs for storing replay samples.

- It costs 750x more to train models as opposed to store the datasets

Prabhu et. al., Computationally Budgeted Continual Learning: What Does Matter?, CVPR23

# Argument #2: Privacy (Memory Constraint)

Privacy: GDPR, Users can request data deletion
Due to privacy policies, storage only permitted for a fixed duration

Prabhu et. al., Computationally Budgeted Continual Learning: What Does Matter?, CVPR23

# Argument #2: Privacy (Memory Constraint)

Privacy: GDPR, Users can request data deletion
Due to privacy policies, storage only permitted for a fixed duration

Concern is valid!
However, does limiting the memory to few samples address this concern?

I argue: NO!

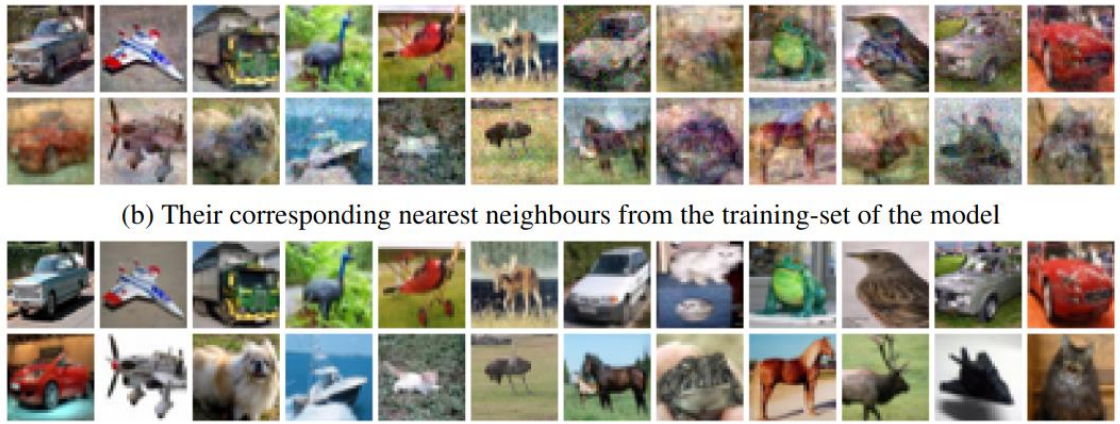
Prabhu et. al., Computationally Budgeted Continual Learning: What Does Matter?, CVPR23

# Argument #2: Privacy (Memory Constraint)

Privacy: GDPR, Users can request data deletion
Due to privacy policies, storage only permitted for a fixed duration

Does limiting the memory to few samples address this concern? We argue: NO!

(a) Top 24 images reconstructed from a binary classifier trained on 50 CIFAR10 images



(b) Their corresponding nearest neighbours from the training-set of the model

Figure 1: Reconstruction of training images from a pretrained binary classifier, trained on 50 CIFAR10 images. The two classes are "animals" and "vehicles". We calculate the nearest neighbor using the SSIM metric.

Model Inversion  Original Image

Images not from our work!
Original links in CLVision'23 Talk by Adel Bibi linked here

Not storing past data not enough to preserve privacy! Deep networks extensively memorize information.

Prabhu et. al., Computationally Budgeted Continual Learning: What Does Matter?, CVPR23

# Argument #2: Privacy (Memory Constraint)

Privacy: GDPR, Users can request data deletion
Due to privacy policies, storage only permitted for a fixed duration

Does limiting the memory to few samples address this concern? NO!

Towards Adversarial Evaluations for Inexact Machine Unlearning

Shashwat Goel*[1], Ameya Prabhu*[2], Amartya Sanyal[3,4], Ser-Nam Lim[5], Philip Torr[2], and Ponnurangam Kumaraguru[1]

- Catastrophic forgetting as a good baseline for data deletion!
- Continual learning seeks to preserve knowledge
- Fundamentally incompatible with goal of data deletion!

Prabhu et. al., Computationally Budgeted Continual Learning: What Does Matter?, CVPR23

# The Bad: Memory Constraints

No valid justification for “tiny memory”

Why is Eliminating "Tiny Memory" Essential?

- Removing this assumption leads to completely different results!
- Previous work has focused on getting around this limitation
- Very simple algorithms like finetuning on all past data performs best!

(This is essentially what industry does when they get more data)

Part of Work: Computationally Budgeted Continual Learning: What Does Matter? CVPR 2023

*Ameya Prabhu*, Hasan Abed Al Kader Hammoud*, Puneet Dokania, Philip HS Torr, Ser-Nam Lim, Bernard Ghanem, Adel Bibi*