Continual Learning: Quirks and Assumptions

Puneet K. Dokania
(University of Oxford & Five AI Ltd., UK)

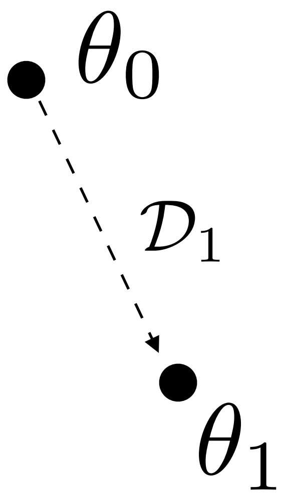
Presented at
RE.Work, Deep Learning 2.0 Virtual Summit
Deep Learning Landscape Stage
28th Jan 2021

Continual Learning: High-Level View

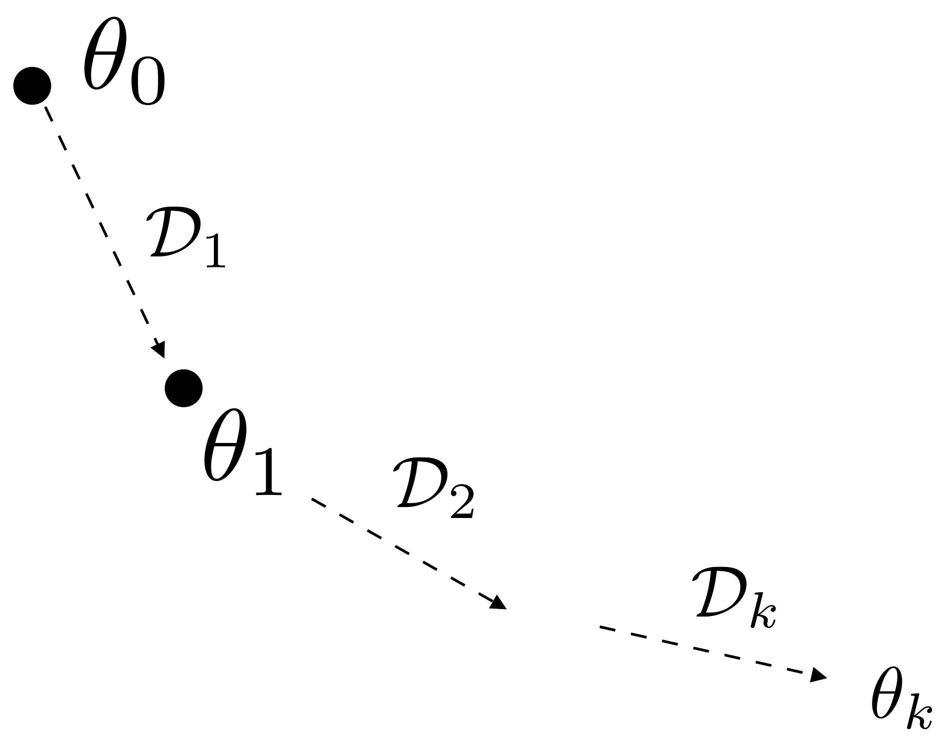
Continual Learning: High-Level View

- θ_0

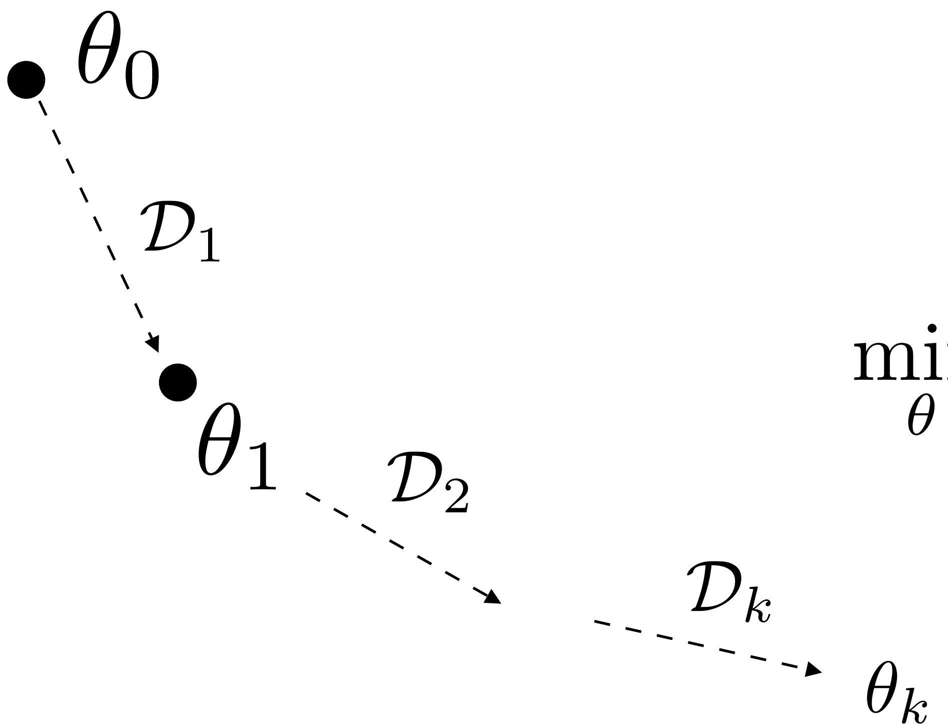
Continual Learning: High-Level View



Continual Learning: High-Level View



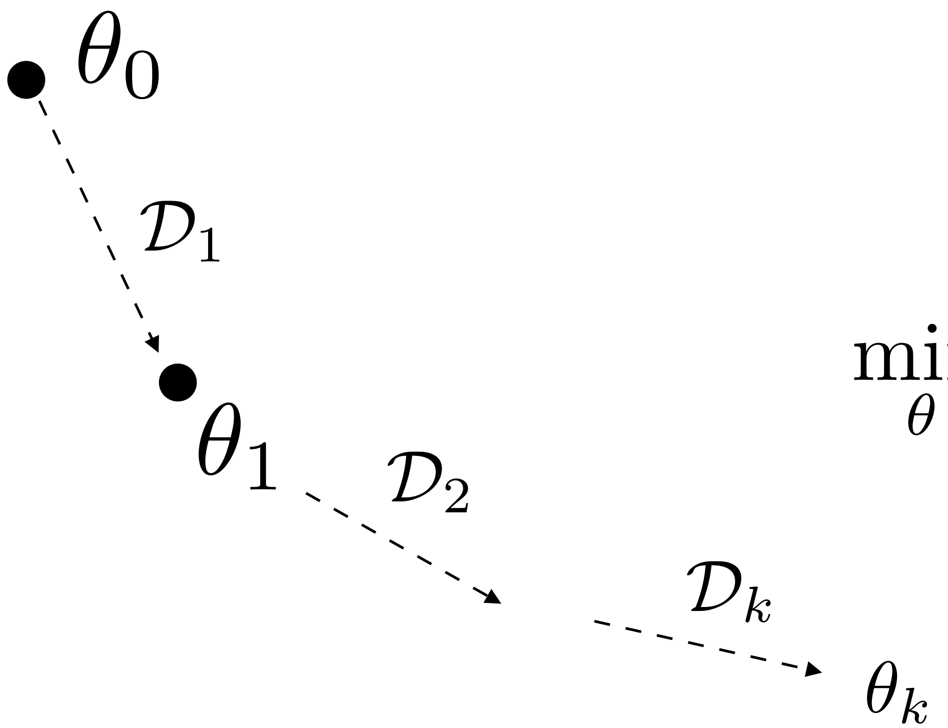
Continual Learning: High-Level View



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

(Previous knowledge)

Continual Learning: High-Level View

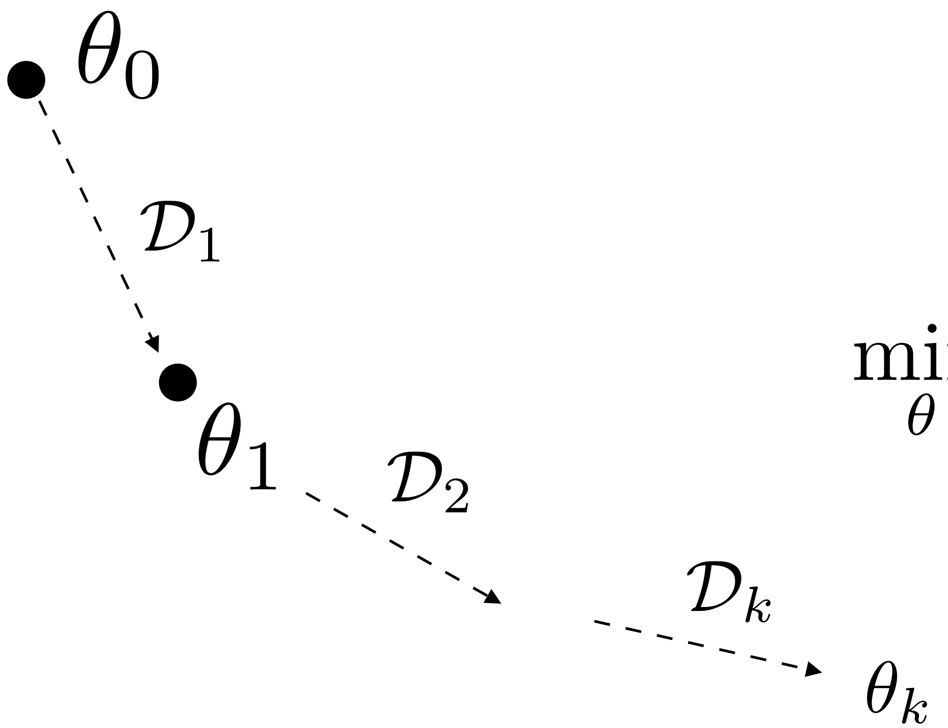


$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters

Continual Learning: High-Level View

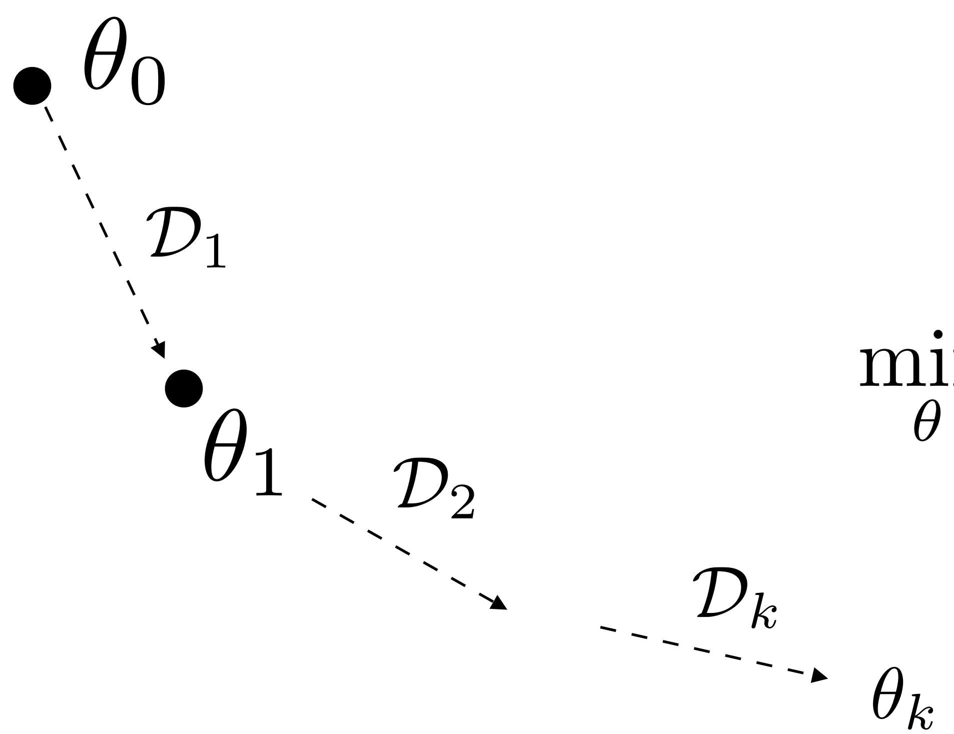


$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters
- Preserve Knowledge? Avoid Forgetting

Continual Learning: High-Level View



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

(Previous knowledge)

- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters
- Preserve Knowledge? Avoid Forgetting
- Update Knowledge? Avoid Intransigence (inability to learn new tasks)

Why Continual Learning?

Why Continual Learning?

- Efficiency — An example of extremely large scale classification (Mahajan et. al., ECCV2018)
  - Training images — 3.5B
  - GPUs — 336
  - Training — ~22 days
  - What if say one million new training data is available?

Why Continual Learning?

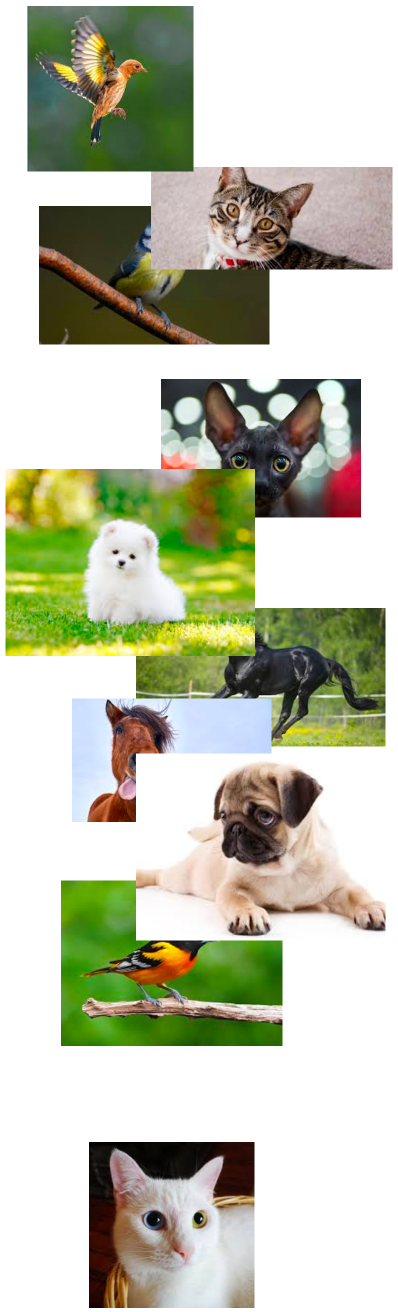
- Efficiency — An example of extremely large scale classification (Mahajan et. al., ECCV2018)
  - Training images — 3.5B
  - GPUs — 336
  - Training — ~22 days
  - What if say one million new training data is available?

- Personalization
  - Imagine (say) Alexa trained on millions of diverse examples before deployment
    - How to efficiently update on some user-specific data without forgetting about the tasks it was trained on before deployment?
    - Privacy — what if the training or user-specific data can't be shared?

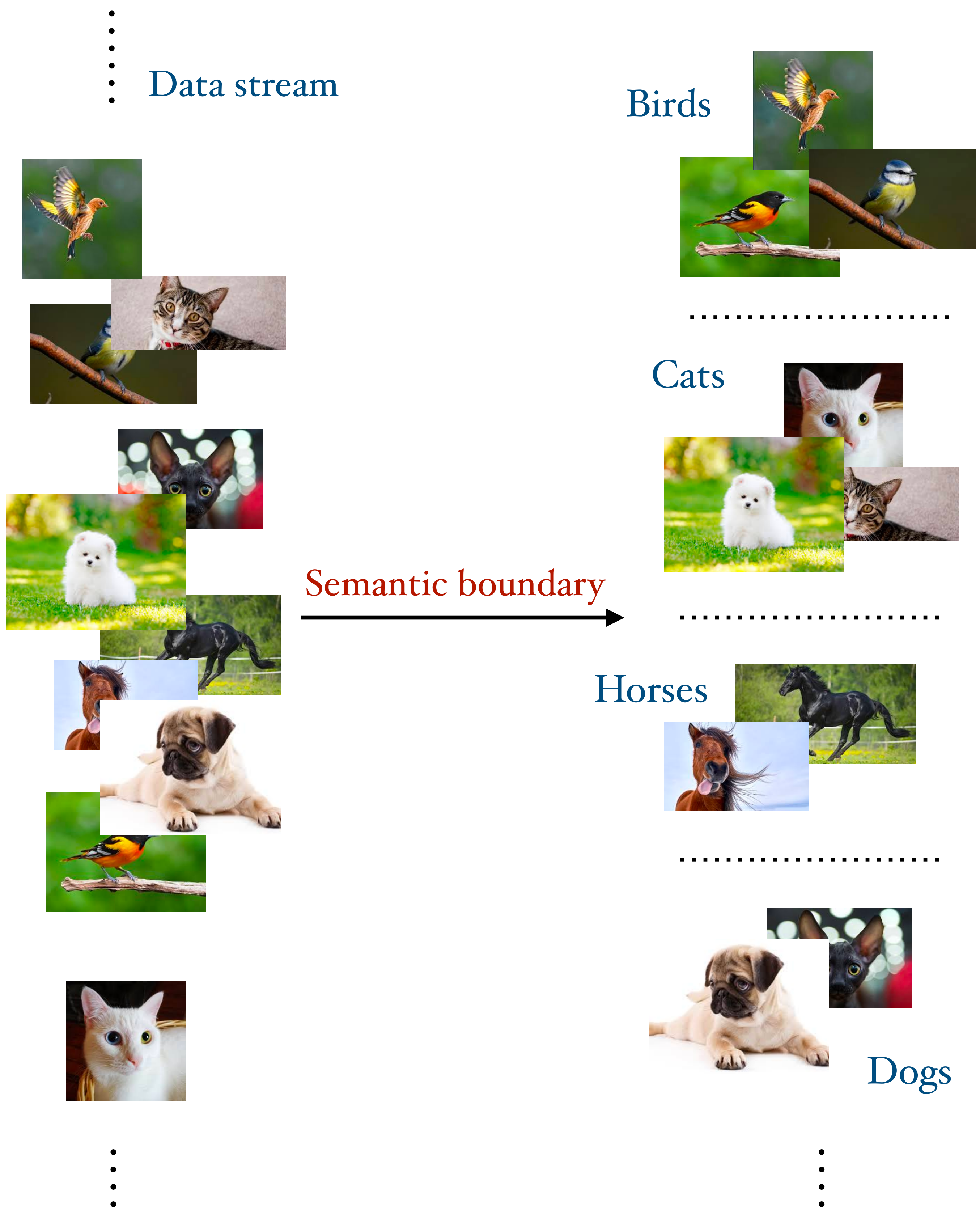
Continual Learning — Assumptions

Continual Learning — Assumptions

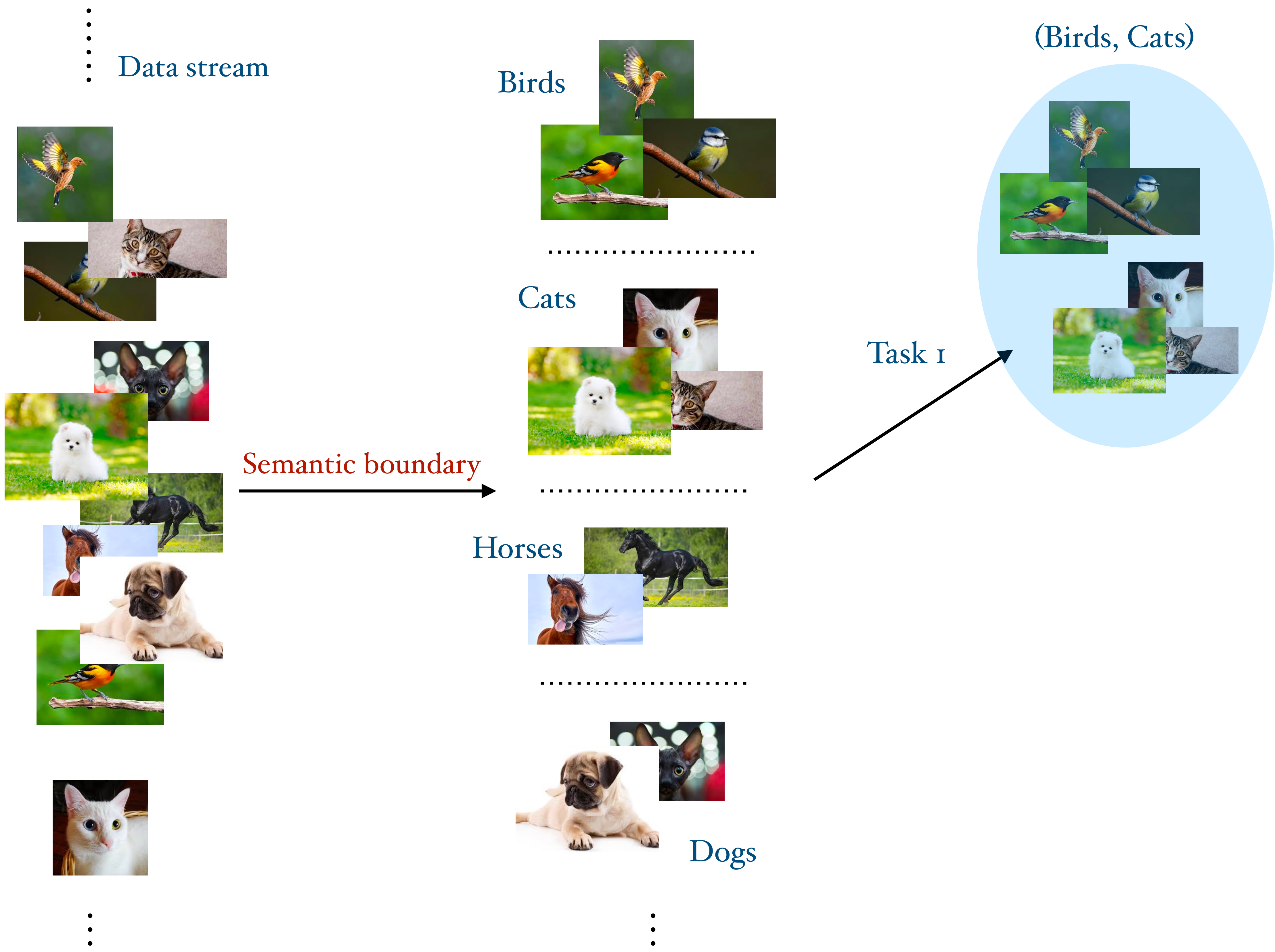
Data stream



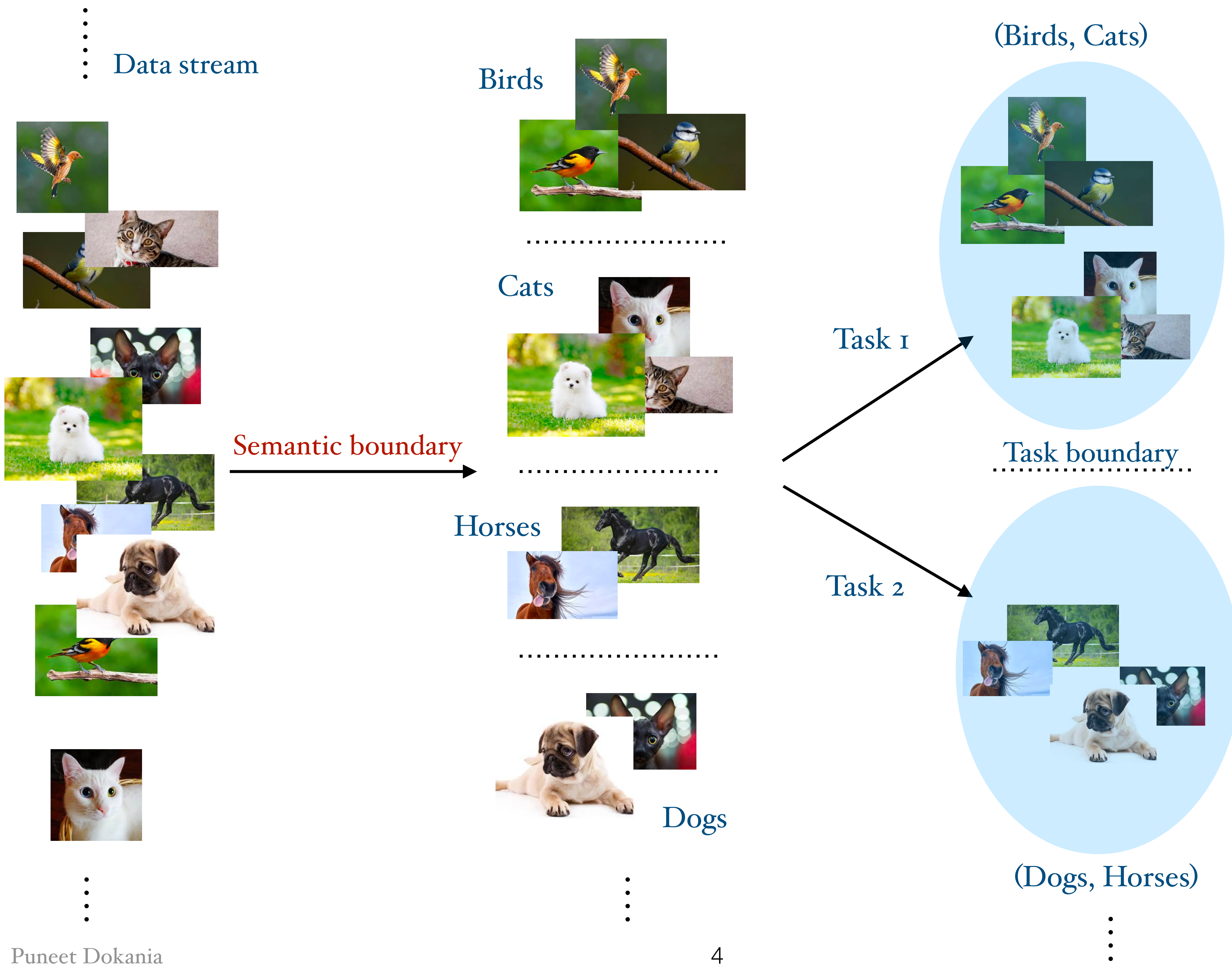
Continual Learning — Assumptions



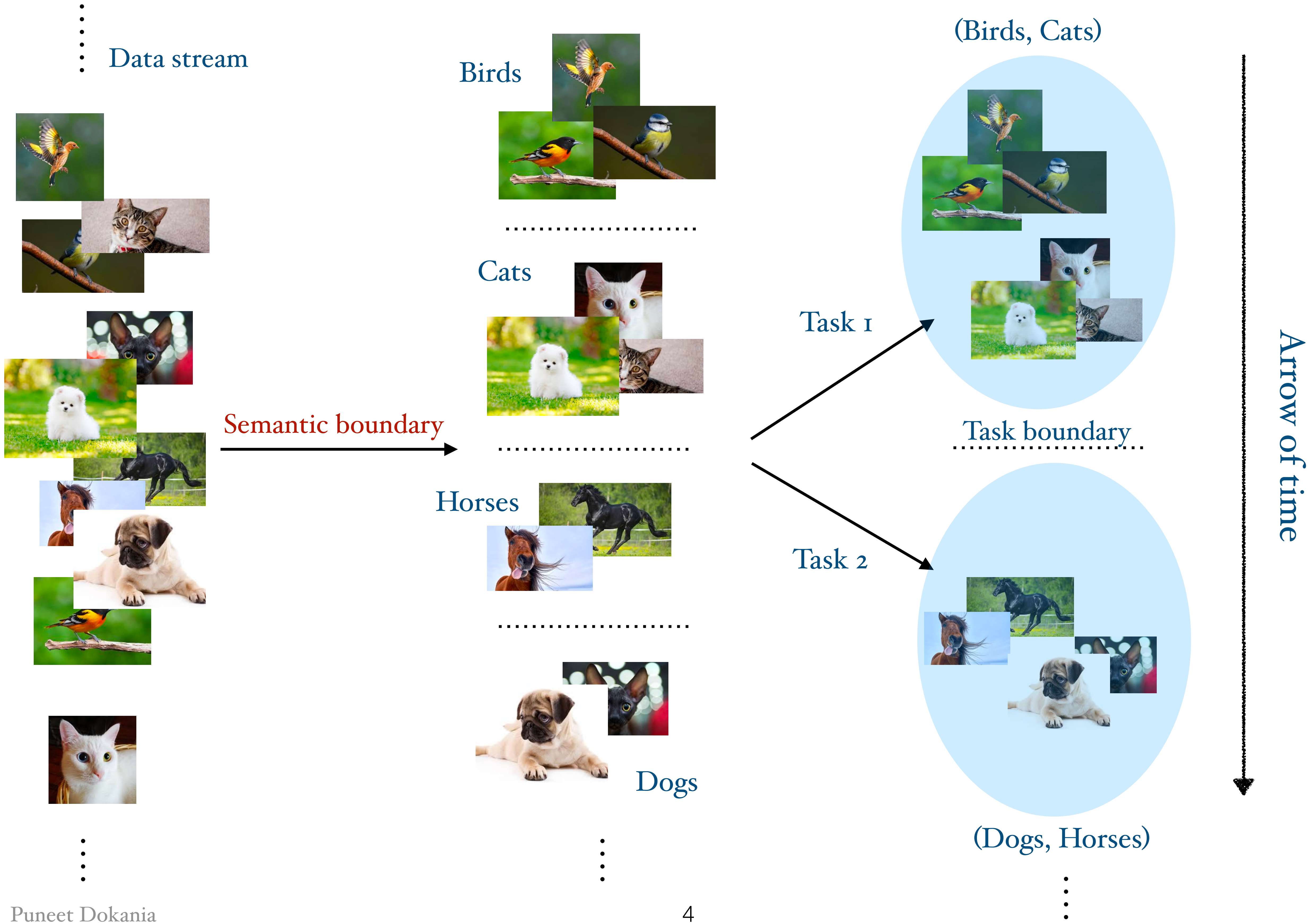
Continual Learning — Assumptions



Continual Learning — Assumptions

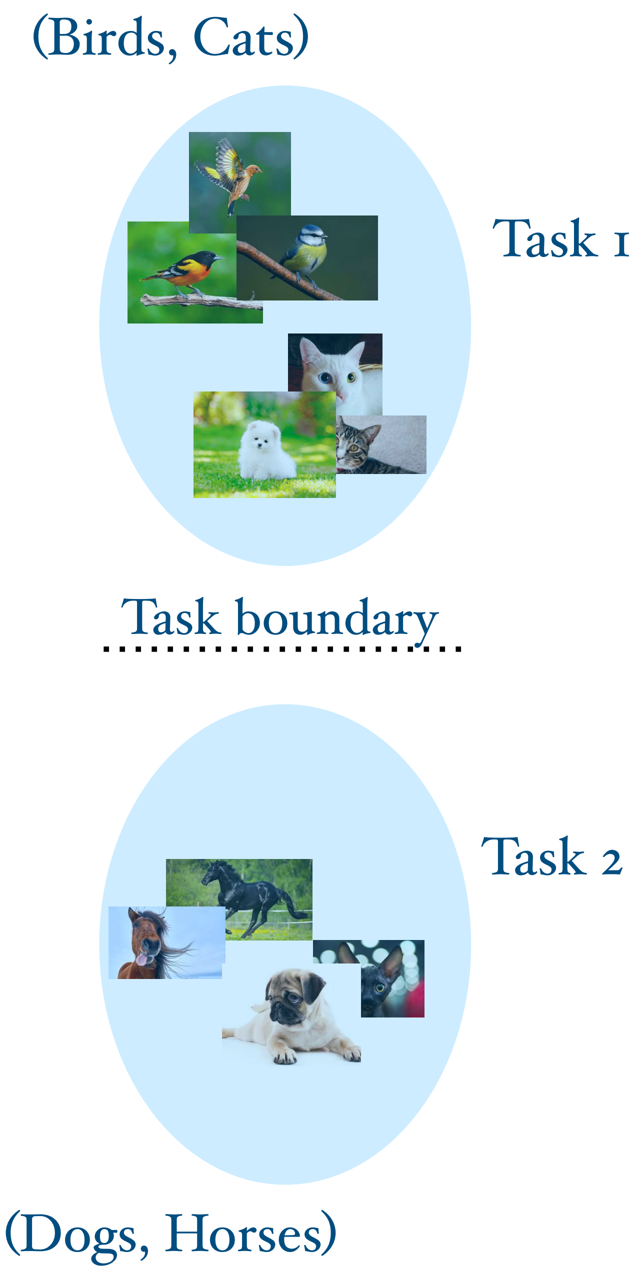


Continual Learning — Assumptions



Continual Learning — Assumptions

Continual Learning — Assumptions



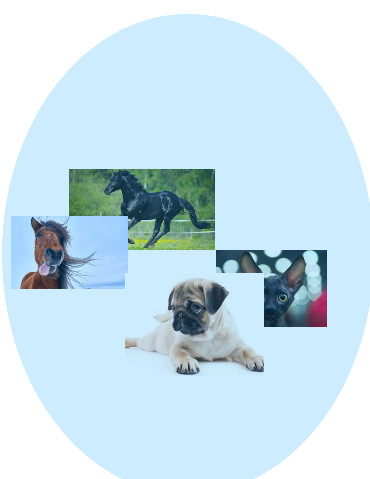
Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary

Task 2



(Dogs, Horses)

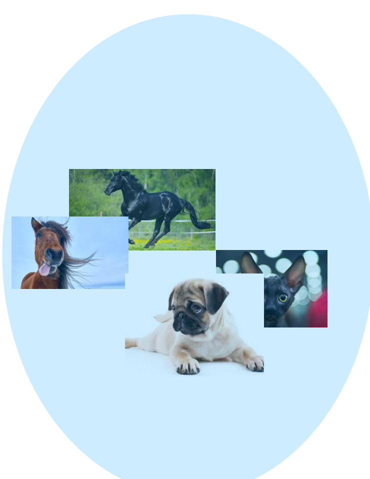
$\mathbf{x}$

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

$\mathbf{x}$

If allowed to use multiple times — **offline**

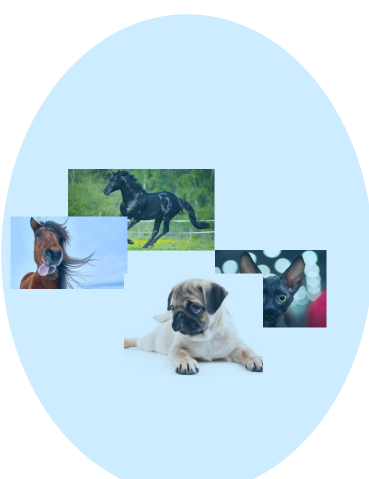
- Multiple epoch over the train of new task

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

x

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

If can be used only once — **online**

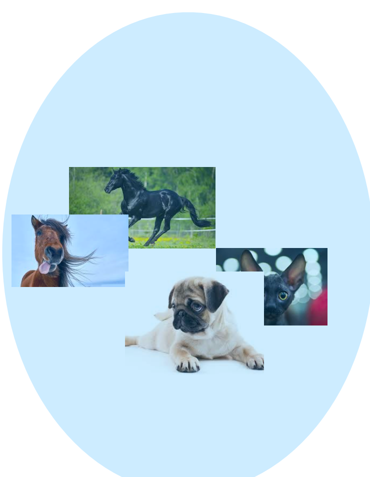
- Only one pass over the train data of new task

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

$\mathbf{x}$

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

If can be used only once — **online**

- Only one pass over the train data of new task

Store?

Replay buffer

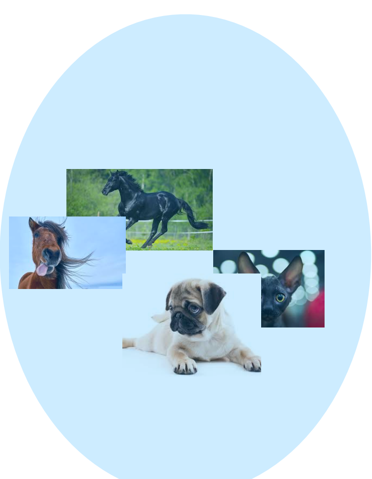
- **Memory based**

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

X

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

If can be used only once — **online**

- Only one pass over the train data of new task

Store?

Replay buffer

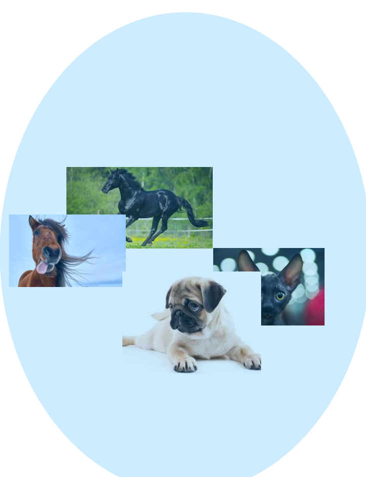
- **Memory based**

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

$\mathbf{x}$

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

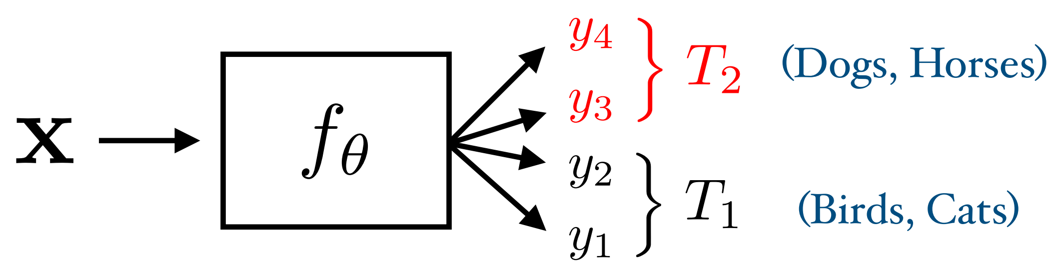
If can be used only once — **online**

- Only one pass over the train data of new task

Store?

Replay buffer

- **Memory based**

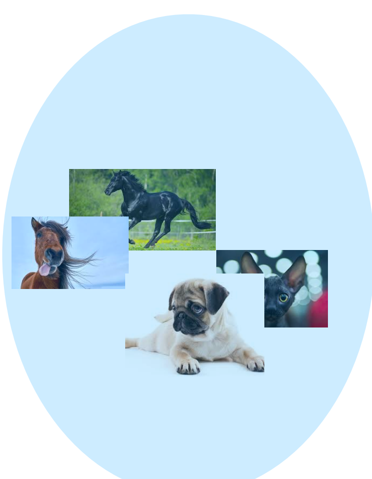


Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

$\mathbf{x}$

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

If can be used only once — **online**

- Only one pass over the train data of new task

Store?

Replay buffer

- **Memory based**

$$\mathbf{x} \longrightarrow \boxed{f_\theta} \longrightarrow \begin{cases} \left.\begin{matrix} y_4 \\ y_3 \end{matrix}\right\} T_2 & \text{(Dogs, Horses)} \\ \left.\begin{matrix} y_2 \\ y_1 \end{matrix}\right\} T_1 & \text{(Birds, Cats)} \end{cases}$$

Task id known — **Task Incremental (Multi-head)**

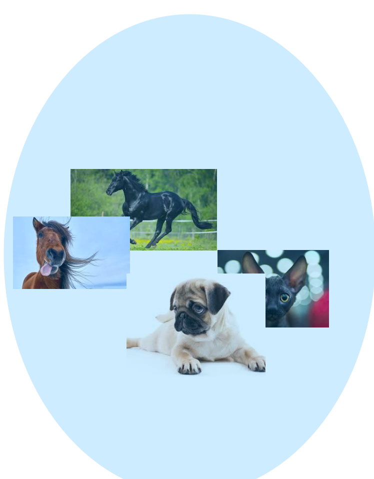
- If task = 1, then either bird or cats

Continual Learning — Assumptions

(Birds, Cats)

Task 1

Task boundary



Task 2

(Dogs, Horses)

$\mathbf{x}$

If allowed to use multiple times — **offline**

- Multiple epoch over the train of new task

If can be used only once — **online**

- Only one pass over the train data of new task

Store?

Replay buffer

- **Memory based**

$$\mathbf{x} \rightarrow \boxed{f_\theta} \rightarrow y_1, y_2, y_3, y_4$$

y_4, y_3 } T_2 (Dogs, Horses)

y_2, y_1 } T_1 (Birds, Cats)

Task id known — **Task Incremental (Multi-head)**

- If task = 1, then either bird or cats

Task id unknown — **Class Incremental (Single-head)**

- Can be {bird, cats, dog, horse}
- Much harder, more realistic

Continual Learning — Various formulations

Form.	CI-CL	Online	Disjoint	Papers	Regularize	Memory	Distill	Param iso
A	✓	✓	✓	MIR[11], GMED[12]	×	✓	×	×
B	✓	×	✓	LwM[13], DMC[14]	×	×	✓	×
				SDC [15]	✓	×	×	×
				BiC[16], iCARL[4], UCIR[17], EEIL[18], IL2M[19], WA[20], PODNet[21], MCIL[22]	×	✓	✓	×
				RPS-Net[23], iTAML[24]	×	✓	✓	✓
				CGATE[25]	×	✓	×	✓
				RWALK[8]	✓	✓	×	×
C	×	×	✓	PNN[26], DEN[27]	×	×	×	✓
				DGR [28]	×	✓	×	×
				LwF[3]	×	×	✓	×
				P&C[29]	×	×	✓	✓
				APD[30]	✓	×	×	✓
				VCL[31]	✓	✓	×	×
				MAS[32], IMM[33], SI[5], Online-EWC[29], EWC[6]	✓	×	×	×
D	×	✓	✓	TinyER[34], HAL[35]	×	✓	×	×
				GEM[7], AGEM[36]	✓	✓	×	×
E	✓	✓	×	GSS[37]	×	✓	×	×

Most algorithms
- Focus on one particular setting
- Most of these are oversimplified
- Often fail to generalize

Most algorithms
- Sensitive to hyperparameters
- Small scale experiments
  - No practical benefit

Hard to understand if the algorithms are actually capturing all the intricacies involved in the continual learning scenario

Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as it could)

Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as it could)

- No hyperparameter
- Not restricted to one of the formulations
  - Can be applied offline/online task/class incremental
- Nothing special to prevent forgetting
  - No regularization
  - No knowledge distillation
  - No bias correction
  - ...

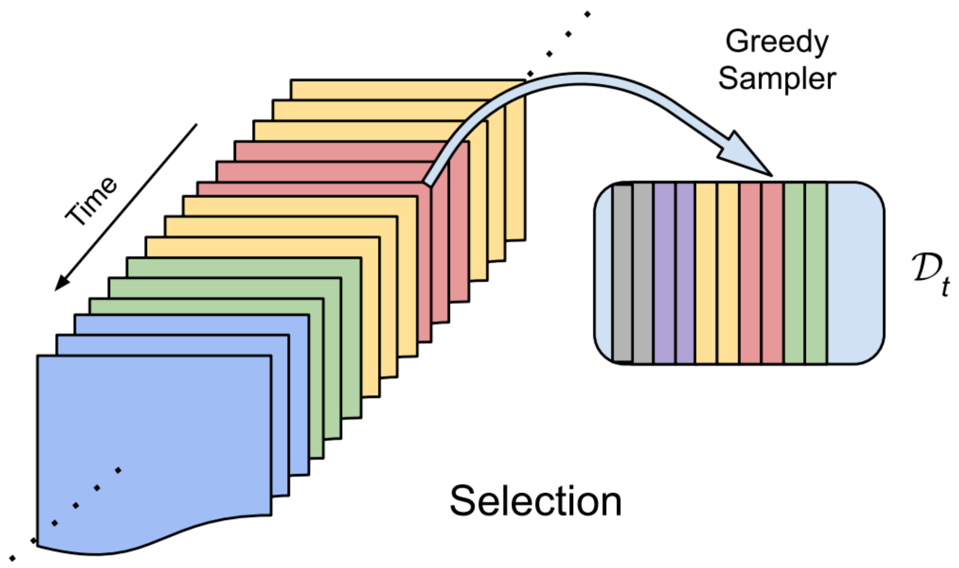
Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as it could)

- No hyperparameter
- Not restricted to one of the formulations
  - Can be applied offline/online task/class incremental
- Nothing special to prevent forgetting
  - No regularization
  - No knowledge distillation
  - No bias correction
  - …

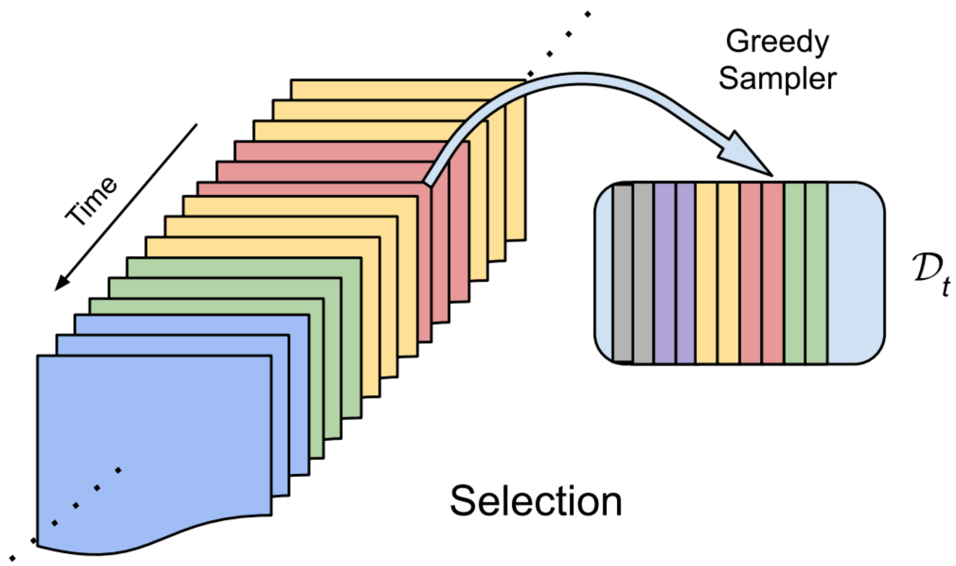
- Use memory — greedily store data given memory budget
- Train only on the memory when asked

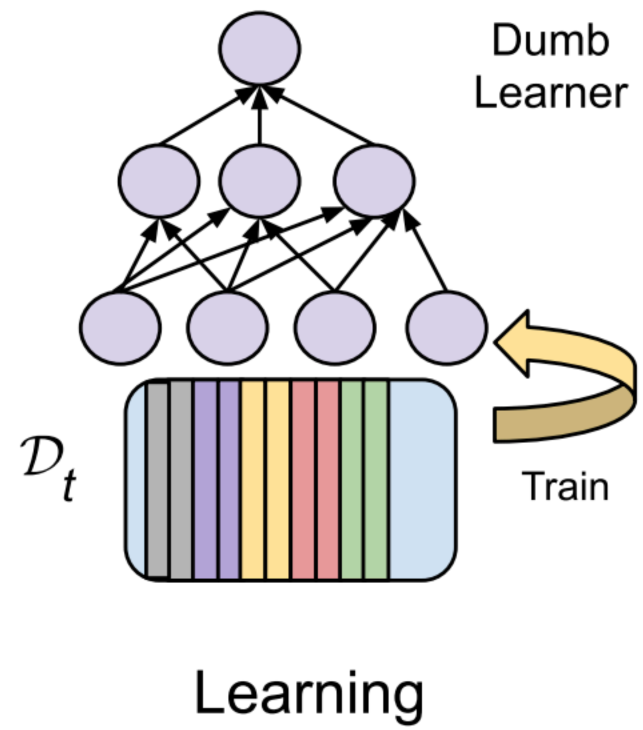
Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as we could)

Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as we could)

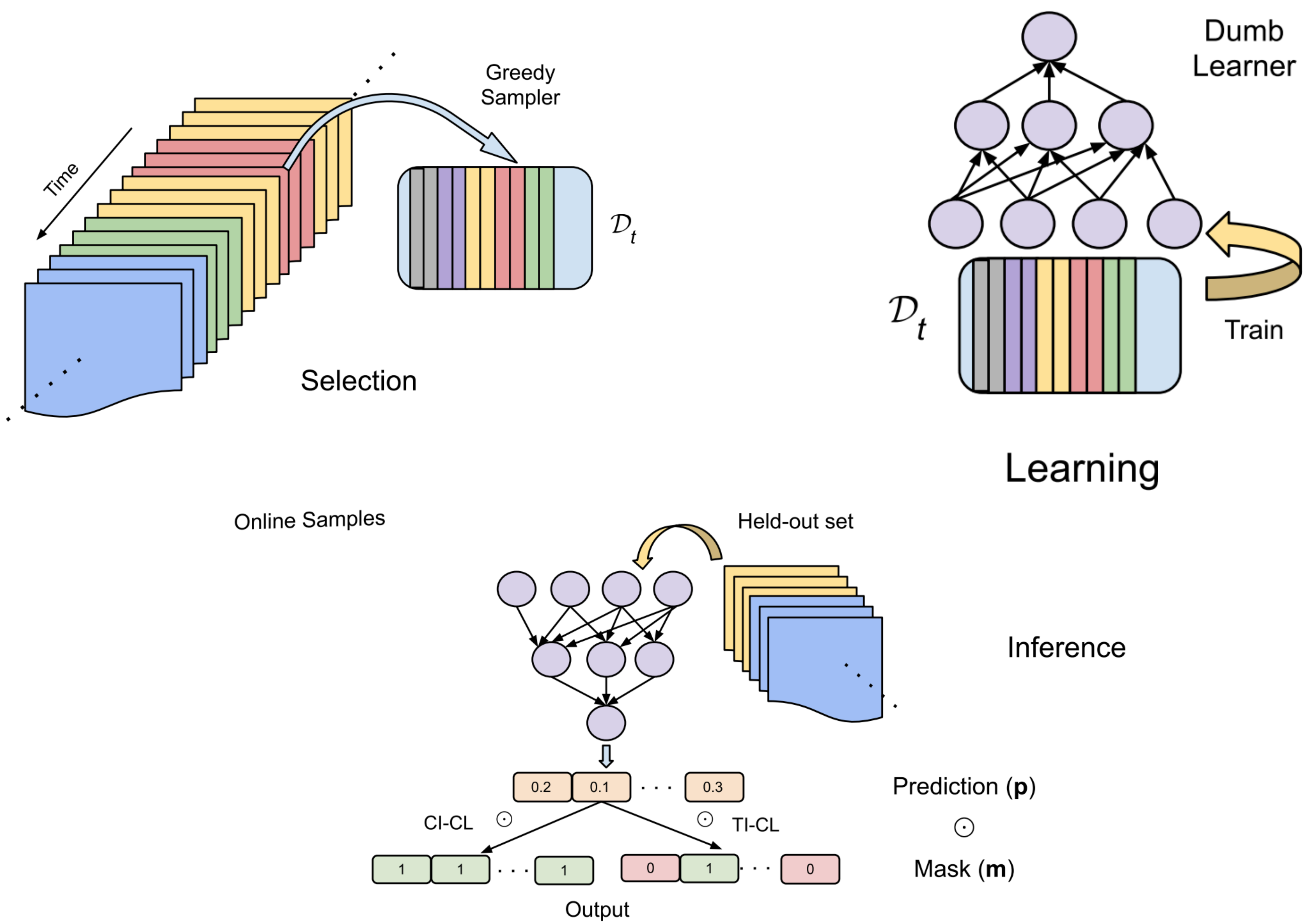


Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as we could)





Continual Learning — GDumb (ECCV2020, Oral)
(As dumb as we could)



Continual Learning — GDumb (ECCV2020, Oral) (evaluation)

Form.	Designed in	Model (Dataset)	memory (k)	Metric	CI-CL	Online	Disjoint
A1	[11]	MLP-400 (MNIST); ResNet18 (CIFAR10)	300, 500; 200, 500, 1000	Acc. (at end)			
A2	[12]	MLP-400 (MNIST); ResNet18 (CIFAR10)	500; 500	Acc. (at end)	✓	✓	✓
A3	[41]	MLP-400 (MNIST); ResNet18 (CIFAR10)	500; 1000	Acc. (at end)			
B1	[42]; [23]	MLP-400 (MNIST); ResNet18 (SVHN)	4400	Acc. (at end)			
B2	[4]	ResNet32 (CIFAR100)	2000	Acc. (avg in t)	✓	×	✓
B3	[21]	ResNet32 (CIFAR100); ResNet18 (ImageNet100)	1000-2000 (+20) x50	Acc. (avg in t)			
C1	[42]	MLP-400 (MNIST)	4400	Acc. (at end)	×	×	✓
C2	[9]	Many (TinyImageNet)	4500,9000	Acc. (at end)			
D	[36]	ResNet-18-S (CIFAR10)	0-1105 (+65) x17	Acc. (at end)	×	✓	✓
E	[37]	MLP-100 (MNIST); ResNet-18 (CIFAR10)	300; 500	Acc. (at end)	✓	✓	×

Continual Learning — GDumb (ECCV2020, Oral)
(Evaluation — Task-incremental, offline)

Method (k)	MNIST (4400)
GEM [7]	98.42 ± 0.10
EWC [6]	98.64 ± 0.22
SI [5]	99.09 ± 0.15
Online EWC [29]	99.12 ± 0.11
MAS [32]	99.22 ± 0.21
DGR [28]	99.50 ± 0.03
LwF [3]	99.60 ± 0.03
DGR+Distil [28]	99.61 ± 0.02
RtF	99.66 ± 0.03
GDumb	**99.77 ± 0.03**

(C1)

Method	Parameters	Regularization	Accuracy
No stored samples			
mean-IMM [33]	3.5M	none	32.42
mode-IMM [33]	9.0M	dropout	42.41
SI [5]	3.5M/9.0M	L2/dropout	43.74
HAT [51]	3.5M/9.0M	L2	44.19
EWC [6]	613K	none	45.13
LwF [3]	9.0M	L2	48.11
EBLL [52]	9.0M	L2	48.17
MAS [32]	3.5M/9.0M	none	48.98
PackNet [53]	613K/3.5M	L2/dropout	55.96
k=4500			
GEM [7]	613K/3.5M	none/dropout	44.23
GDumb	834K	cutmix	45.50
iCARL [4]	613K/3.5M	dropout	48.55
k=9000			
GEM [7]	613K/3.5M	none/dropout	45.27
iCARL [4]	613K/3.5M	dropout	49.94
GDumb	834K	cutmix	**57.27**

(C2)

Continual Learning — GDumb (ECCV2020, Oral) (Conclusions)

- GDumb performed extremely well on almost all the simplified forms of continual learning
- This is **alarming** as the methods we compared against were
  - specifically designed for the evaluation setting
  - had hyperameters to tune
  - etc. etc.

Continual Learning — GDumb (ECCV2020, Oral)
(Conclusions)

- GDumb performed extremely well on almost all the simplified forms of continual learning
- This is **alarming** as the methods we compared against were
  - specifically designed for the evaluation setting
  - had hyperameters to tune
  - etc. etc.

- Perhaps, the assumptions are too simplified
- These assumptions should be motivated from practical usefulness point of view
  - For example, there is no need to restrict on memory budget if we can afford to train
  - The online assumption, etc.

- It is important to try these algorithms on large scale problems to verify their usefulness
- Proper benchmarking is necessary

Thank You
(stay safe)