Meta-Learning Representations for Continual Learning

Khurram Javed

Work done in collaboration with Martha White

Continual Learning Prediction

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

Naive solution using a neural network:

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

Naive solution using a neural network:

1. Get a new sample

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

AAAAA BBBBBB CCCC DDDDD ...... ZZZZZZ

Naive solution using a neural network:

1. Get a new sample
2. Update parameters w.r.t latest sample

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

Naive solution using a neural network:

1. Get a new sample
2. Update parameters w.r.t latest sample
3. Repeat

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

AAAAA BBBBBB CCcC DDDDD ...... ZZZZZZ

Naive solution using a neural network:

1. Get a new sample
2. Update parameters w.r.t latest sample
3. Repeat

Continual Learning Prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Data could be highly correlated

AAAAA BBBBBB CCCC DDDDD ...... ZZZZZZ

Naive solution using a neural network:

1. Get a new sample
2. Update parameters w.r.t latest sample
3. Repeat

Traditional Neural Network Stack

Traditional Neural Network Stack

Traditional Neural Network Stack

Dataset : Omniglot

Traditional Neural Network Stack

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning

Traditional Neural Network Stack

Dataset : Omniglot

- ~~~950 characters from multiple alphabets for representation learning~~

Traditional Neural Network Stack

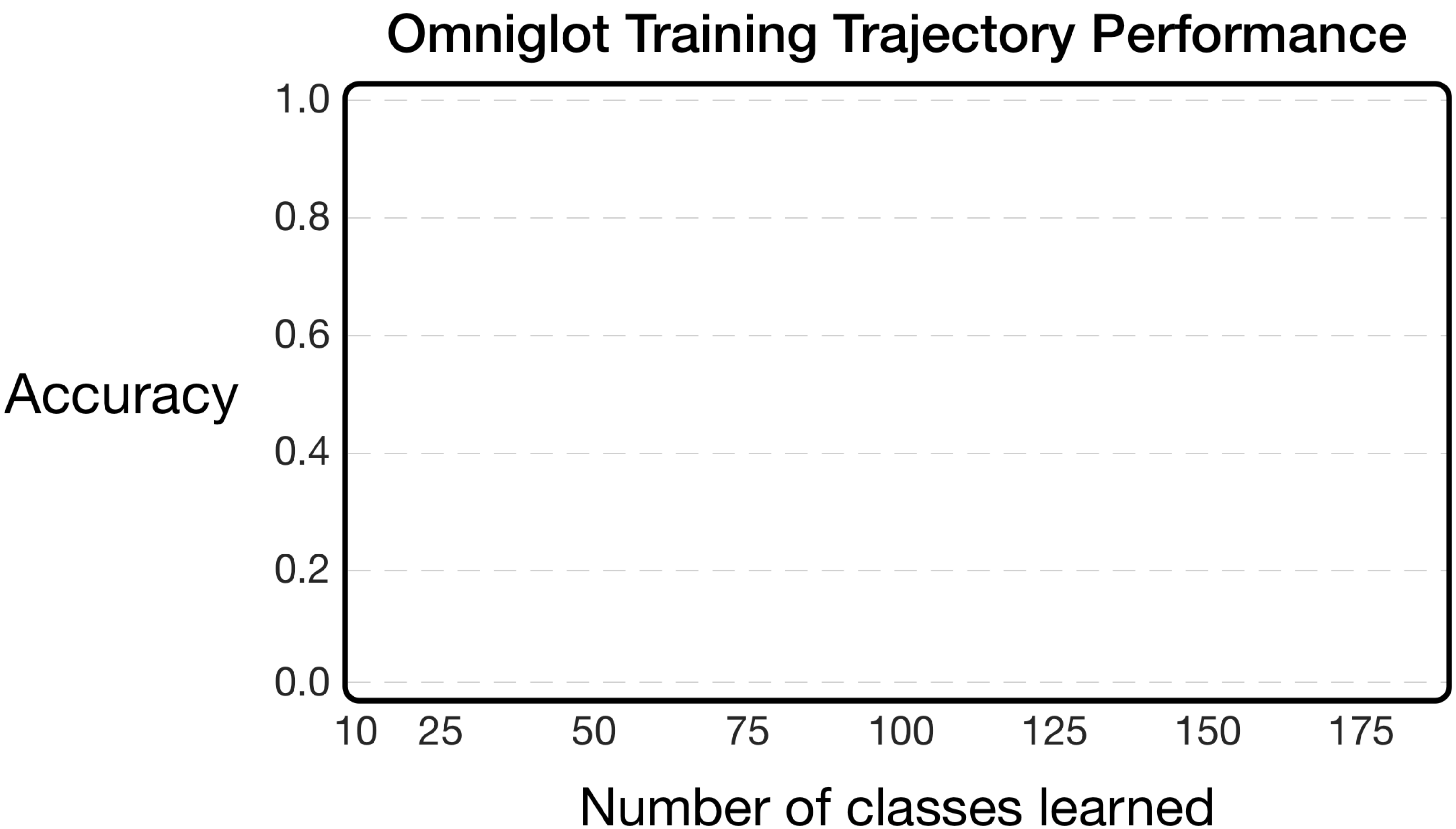
Dataset : Omniglot

- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for Continual Learning Prediction

Traditional Neural Network Stack

Dataset : Omniglot

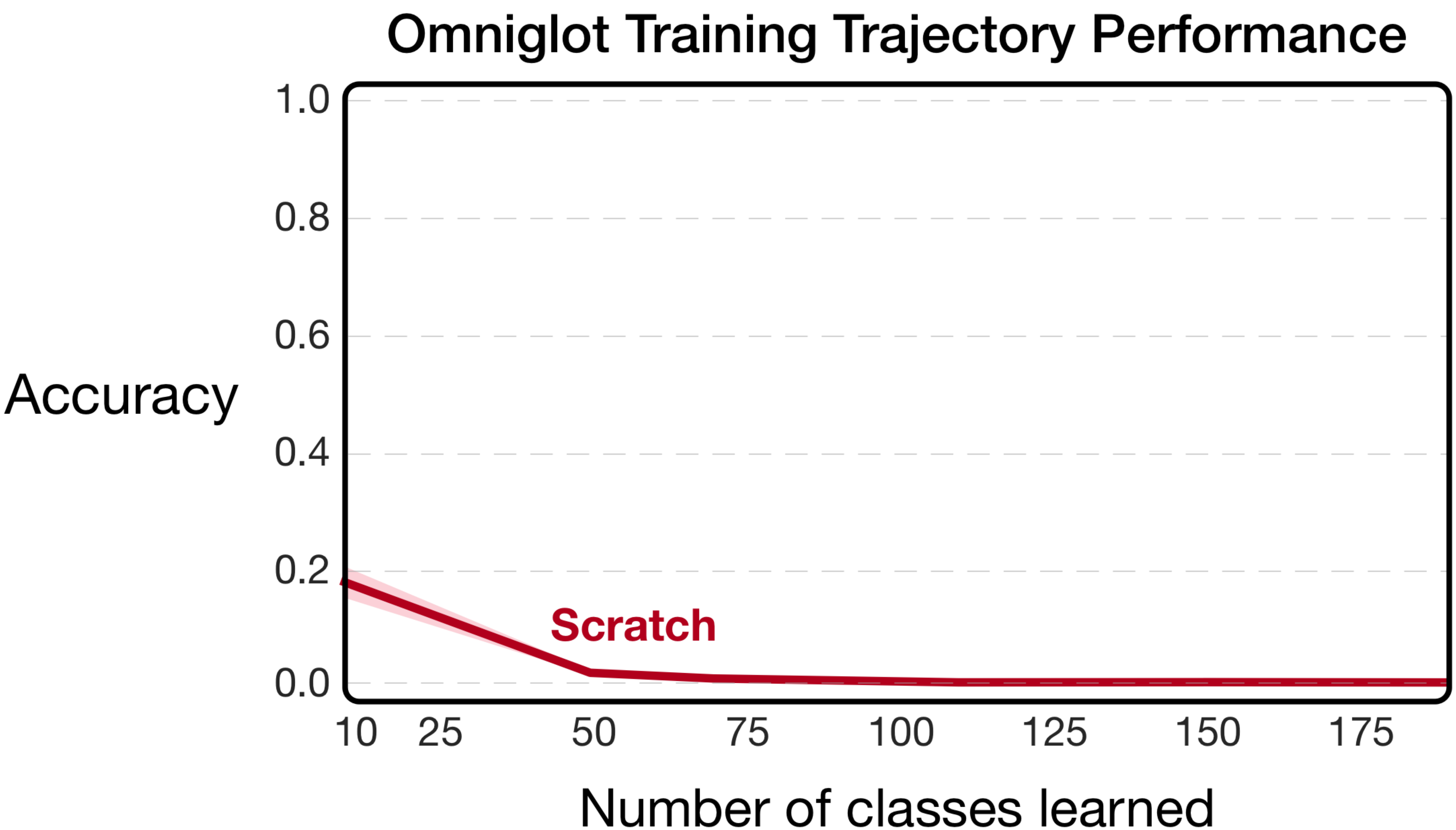
- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for Continual Learning Prediction



Traditional Neural Network Stack

Dataset : Omniglot

- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for Continual Learning Prediction



Catastrophic Interference

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling
2. Dense inputs

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling

2. Dense inputs

3. Global and greedy learning update

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

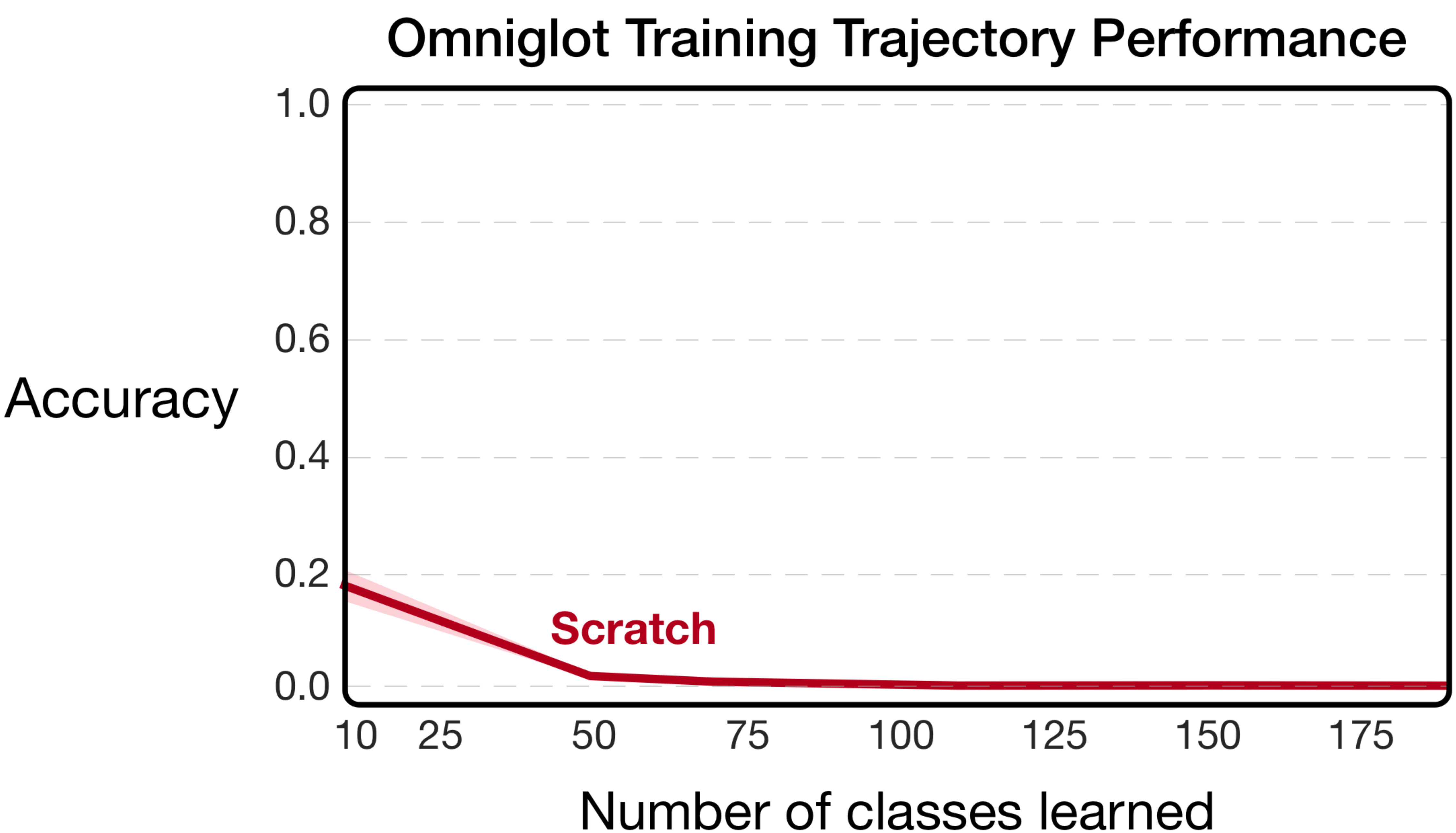
When and why?

1. $\boxed{\text{Non-IID sampling}}$
2. Dense inputs
3. Global and greedy learning update

Non-IID sampling

Dataset : Omniglot

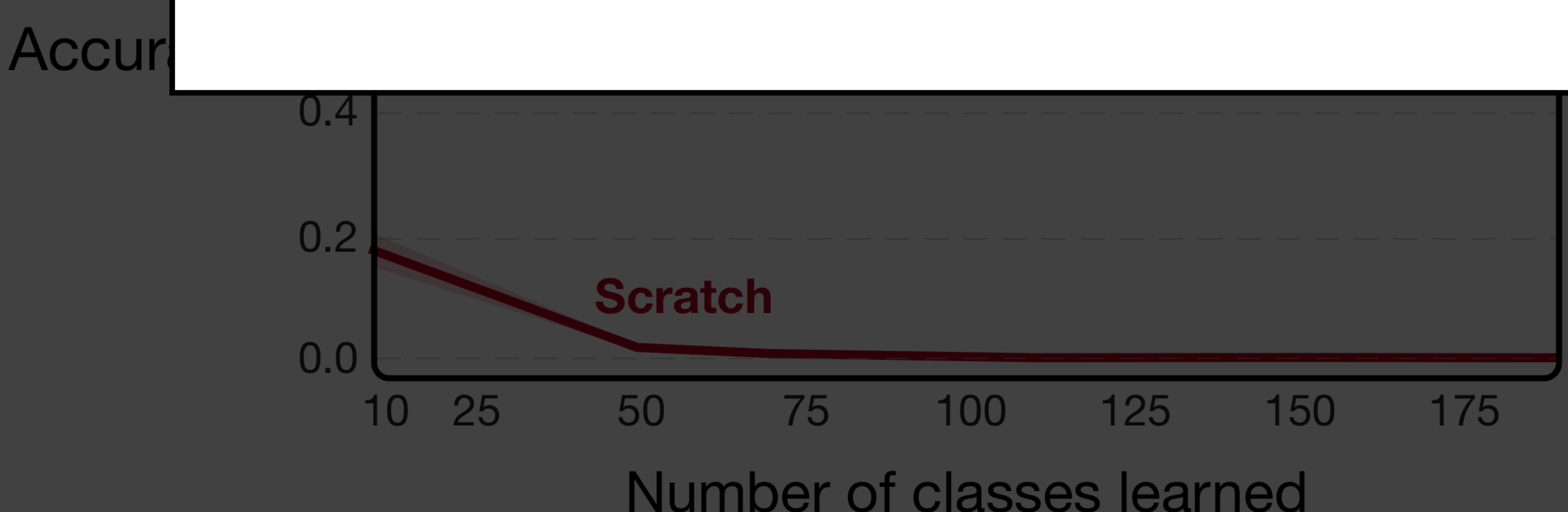
- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Non-IID sampling

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

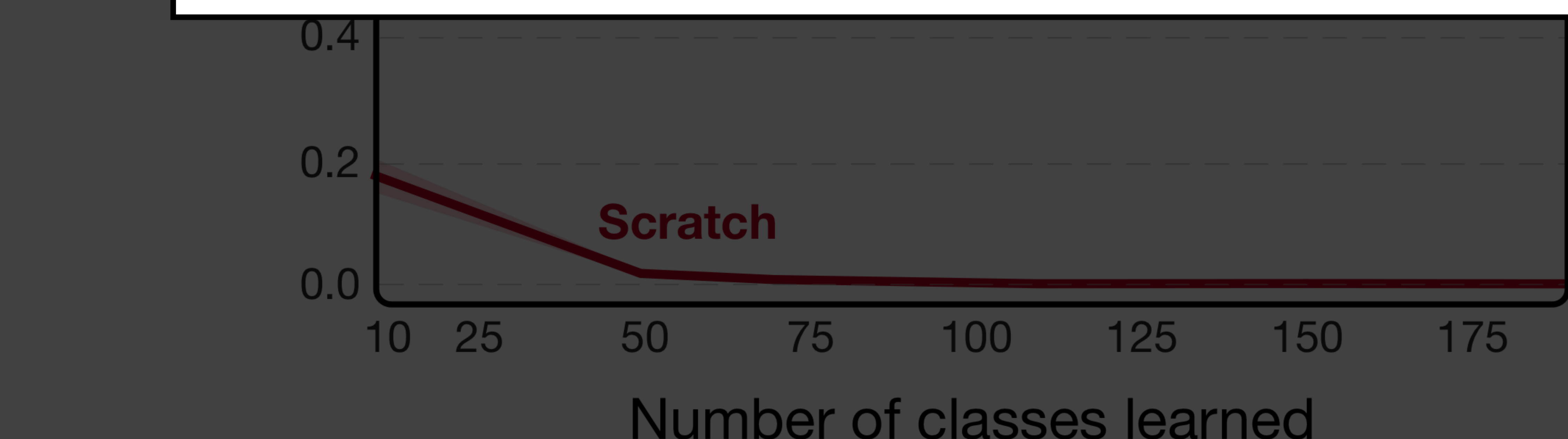


Non-IID sampling

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$



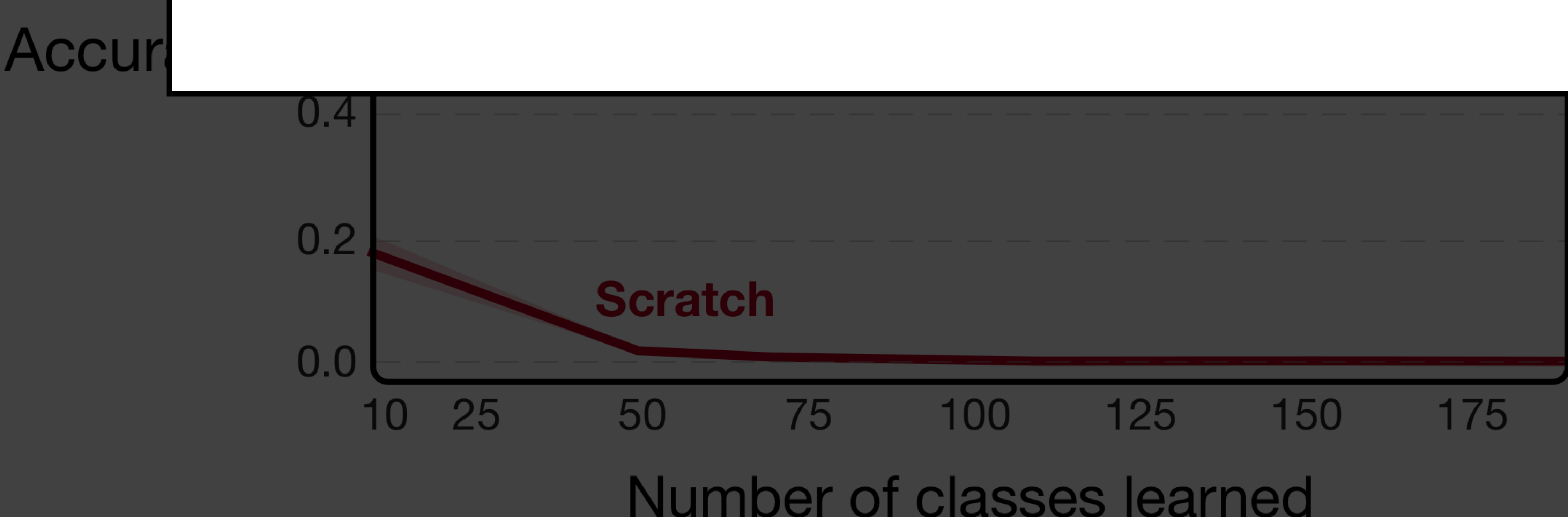
Non-IID sampling

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

Shuffle order



Non-IID sampling

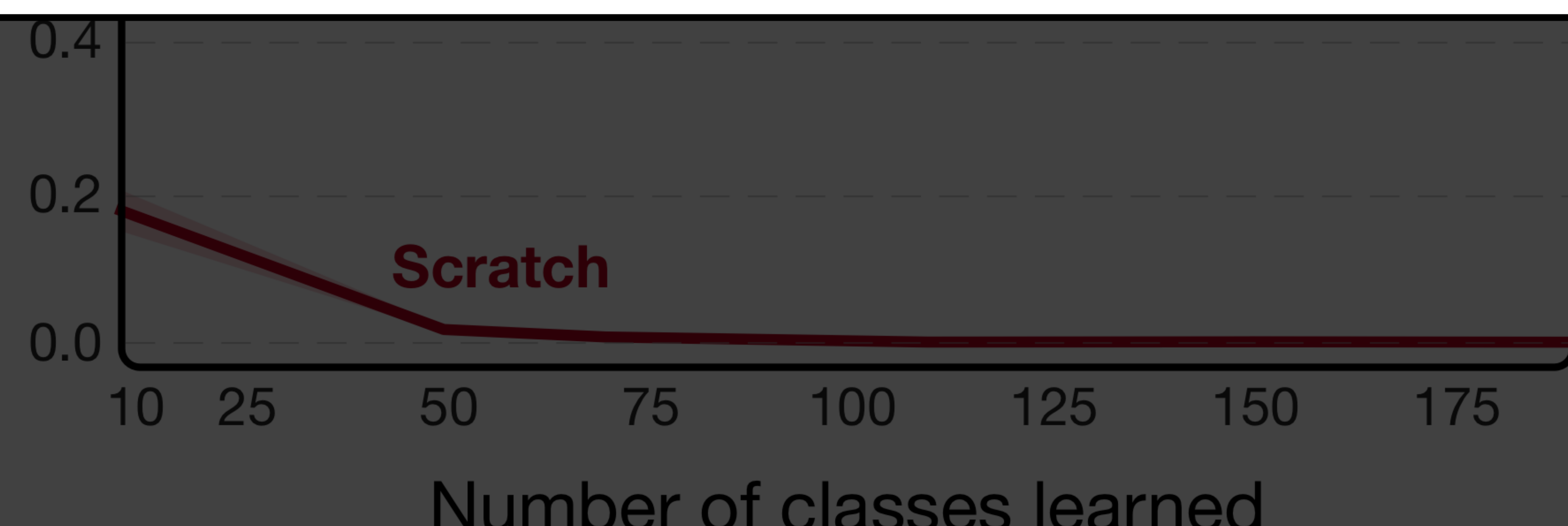
Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

$$D_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

$\Downarrow$ Shuffle order

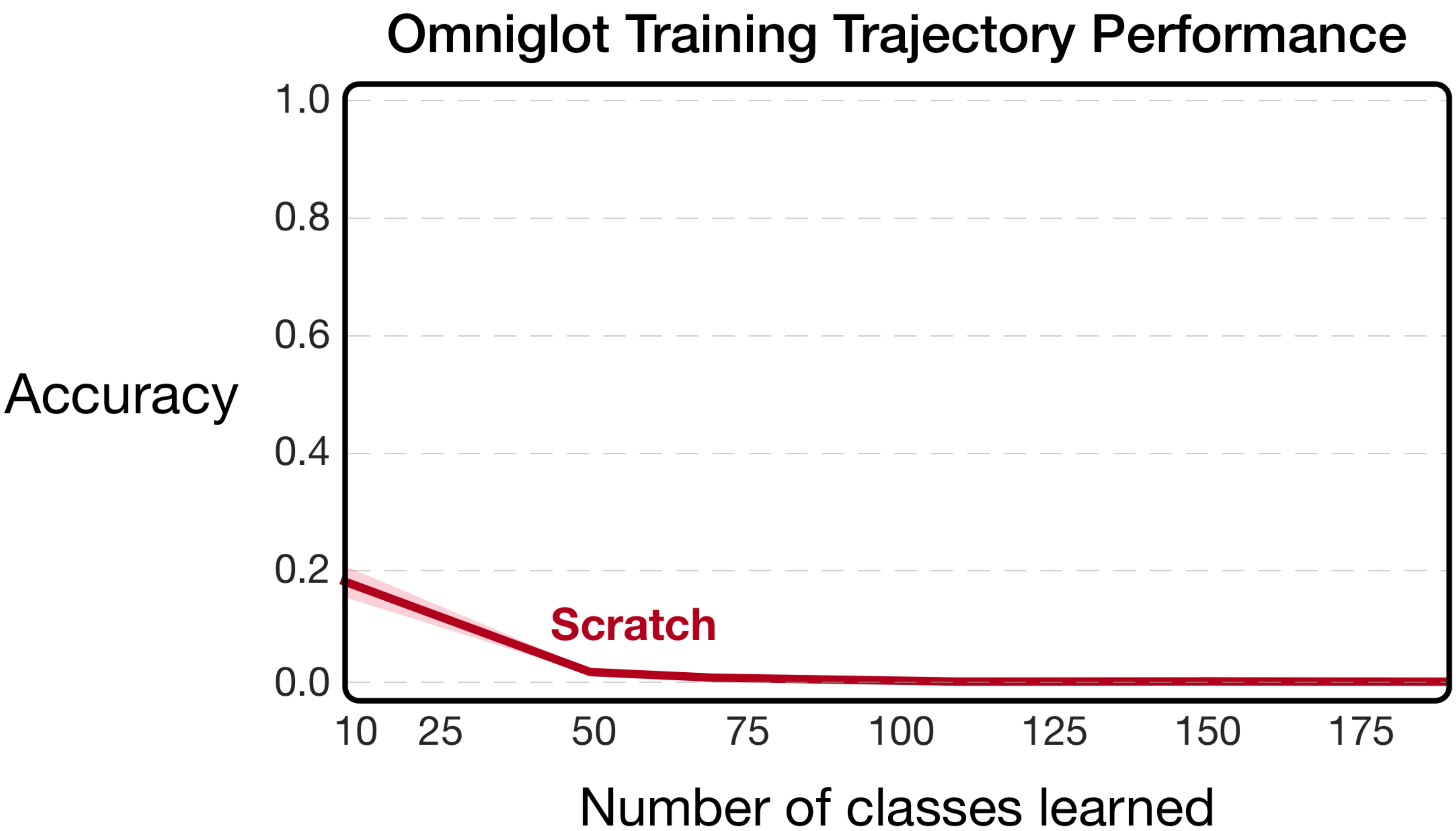
$$D_{rand} = (X_{10}, Y_{10}), (X_1, Y_1), \ldots, (X_m, Y_m), \ldots$$



Non-IID sampling

Dataset : Omniglot

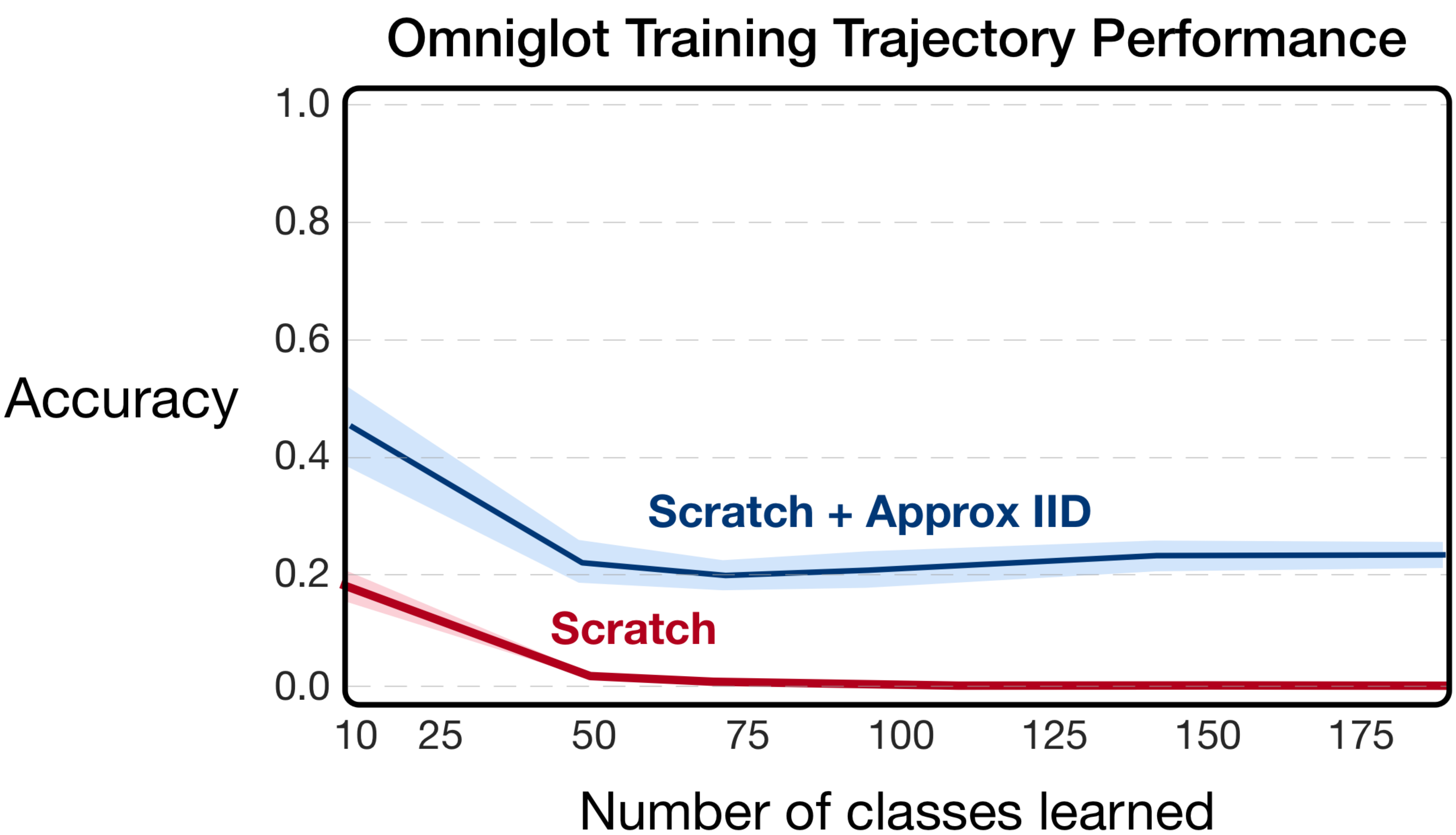
- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for continual learning prediction



Non-IID sampling

Dataset : Omniglot

- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for continual learning prediction



Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. $\boxed{\text{Non-IID sampling}}$
2. Dense inputs
3. Global and greedy update

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

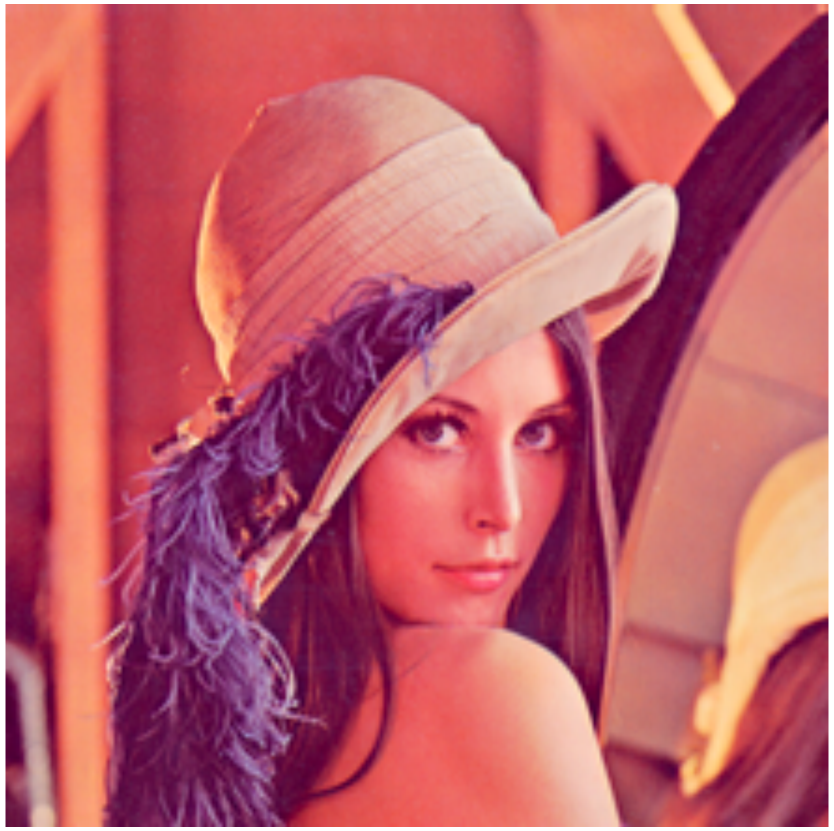
Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling
2. Dense inputs
3. Global and greedy update

Dense inputs

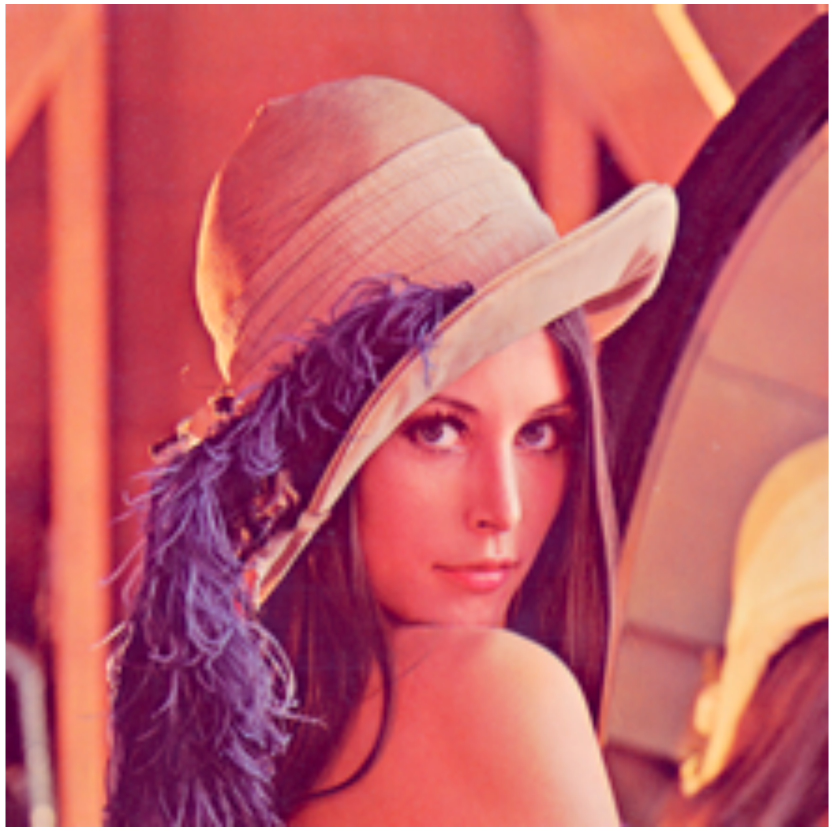
64



64

Dense inputs

64

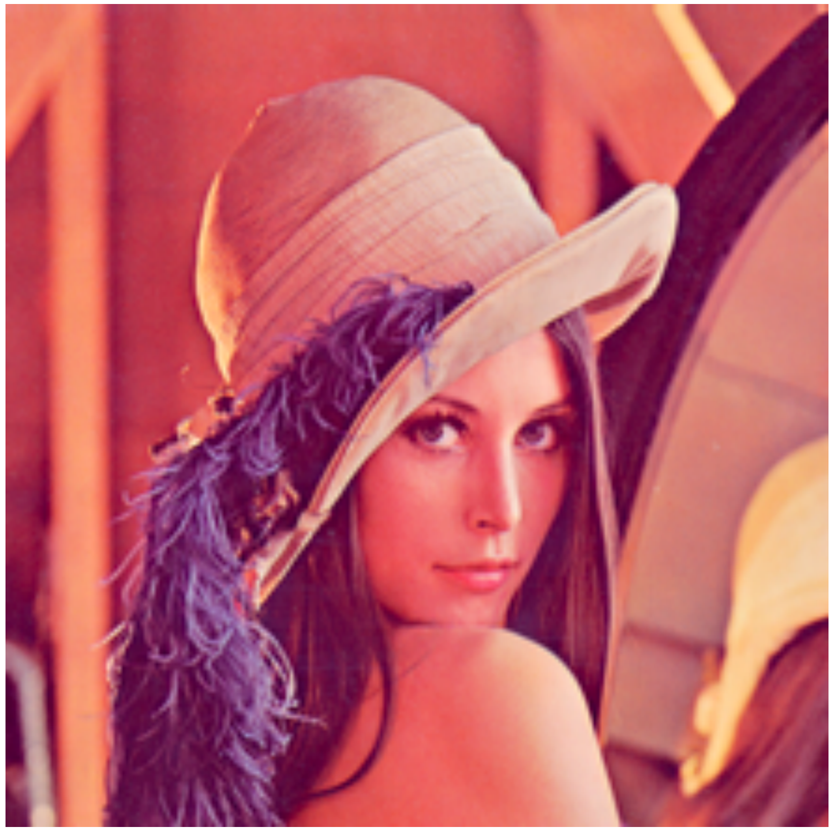


64

=> Input vector of length 64 x 64 = 4096

Dense inputs

64



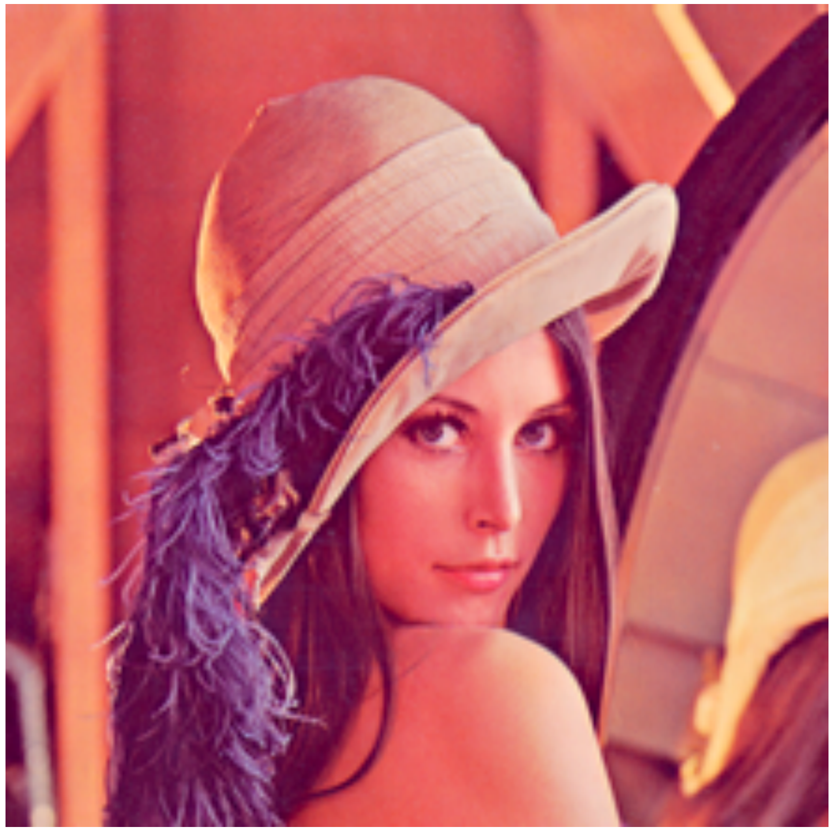
64

=> Input vector of length 64 x 64 = 4096

Most values are non-zero for most natural images

Dense inputs

64



64

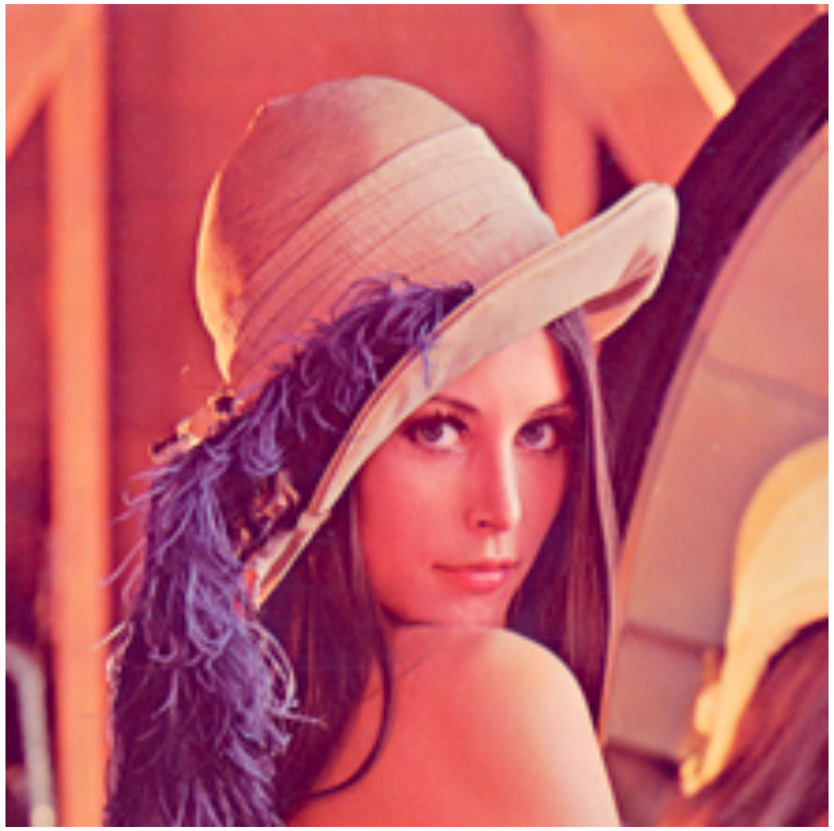
=> Input vector of length 64 x 64 = 4096

Most values are non-zero for most natural images

Alternative?

Dense inputs

64



64

=> Input vector of length 64 x 64 = 4096

Most values are non-zero for most natural images

Alternative?

Sparse input representations
Example: Tile coding

Dense Inputs

Dataset : Omniglot

- ~~~950 characters from multiple alphabets for representation learning~~
- ~600 characters from multiple alphabets for continual learning prediction

Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600

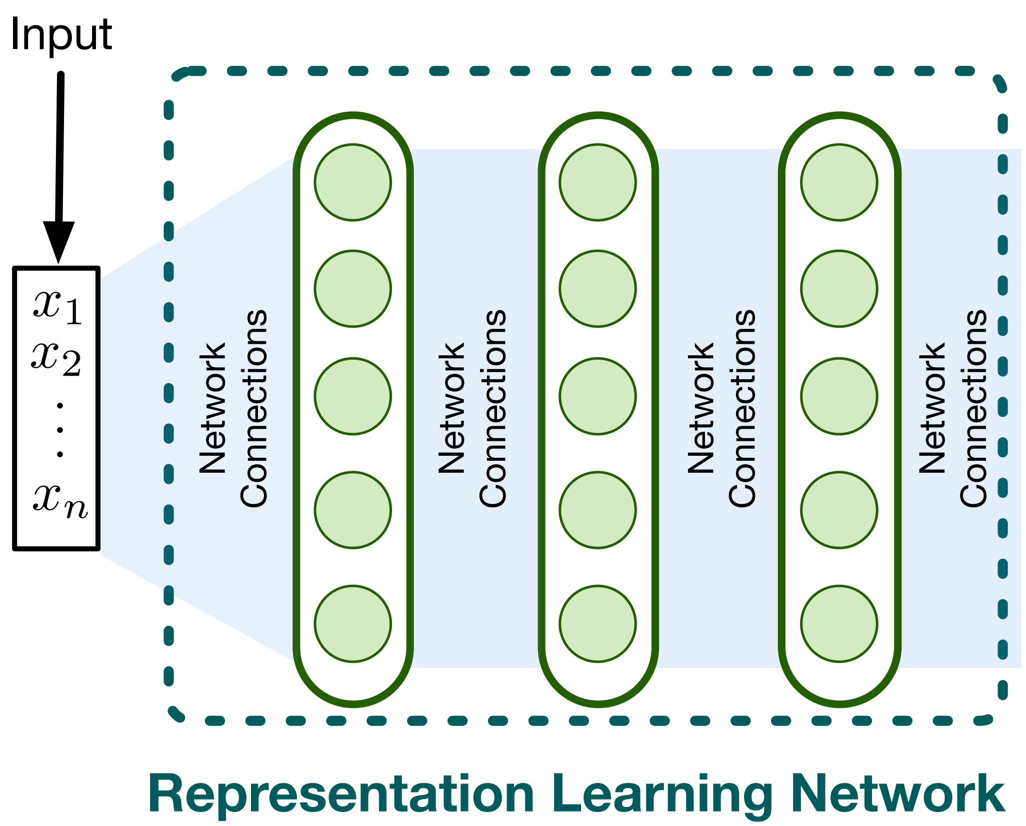
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600



Dense Inputs

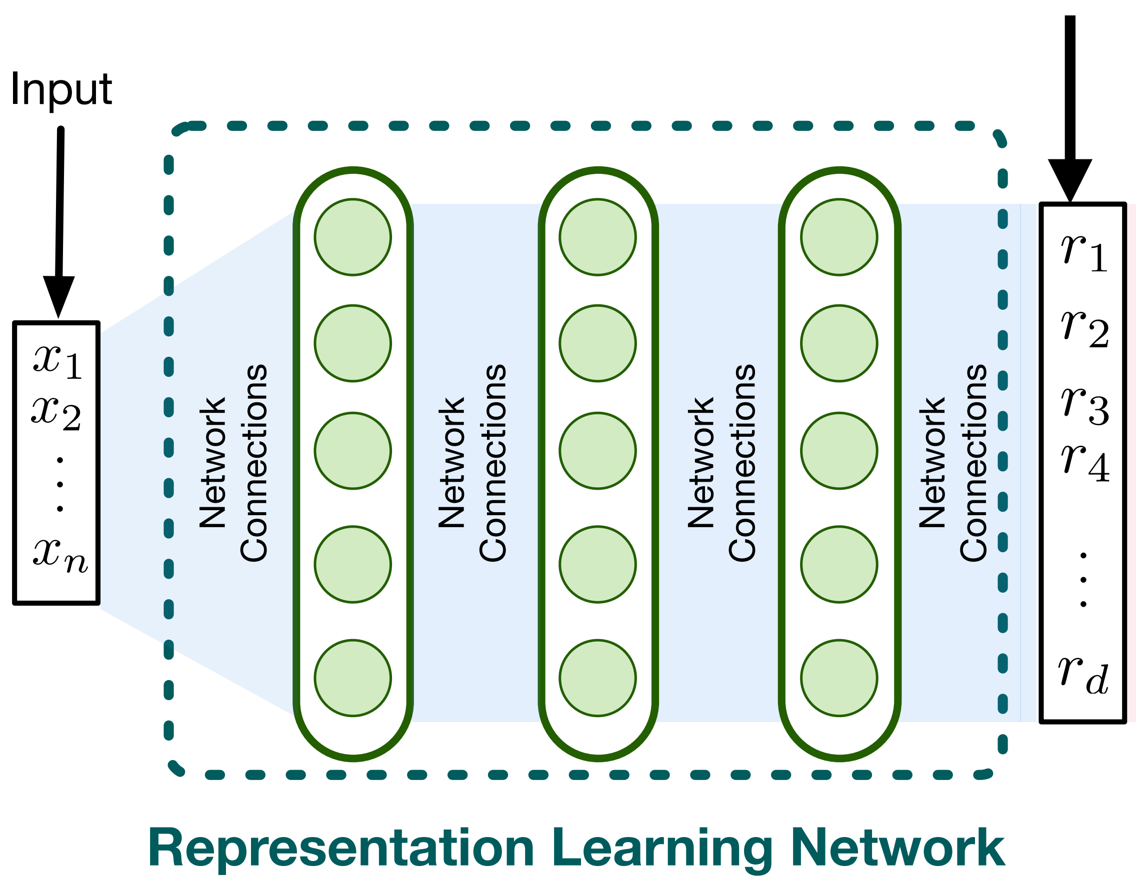
Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600

Learned representation



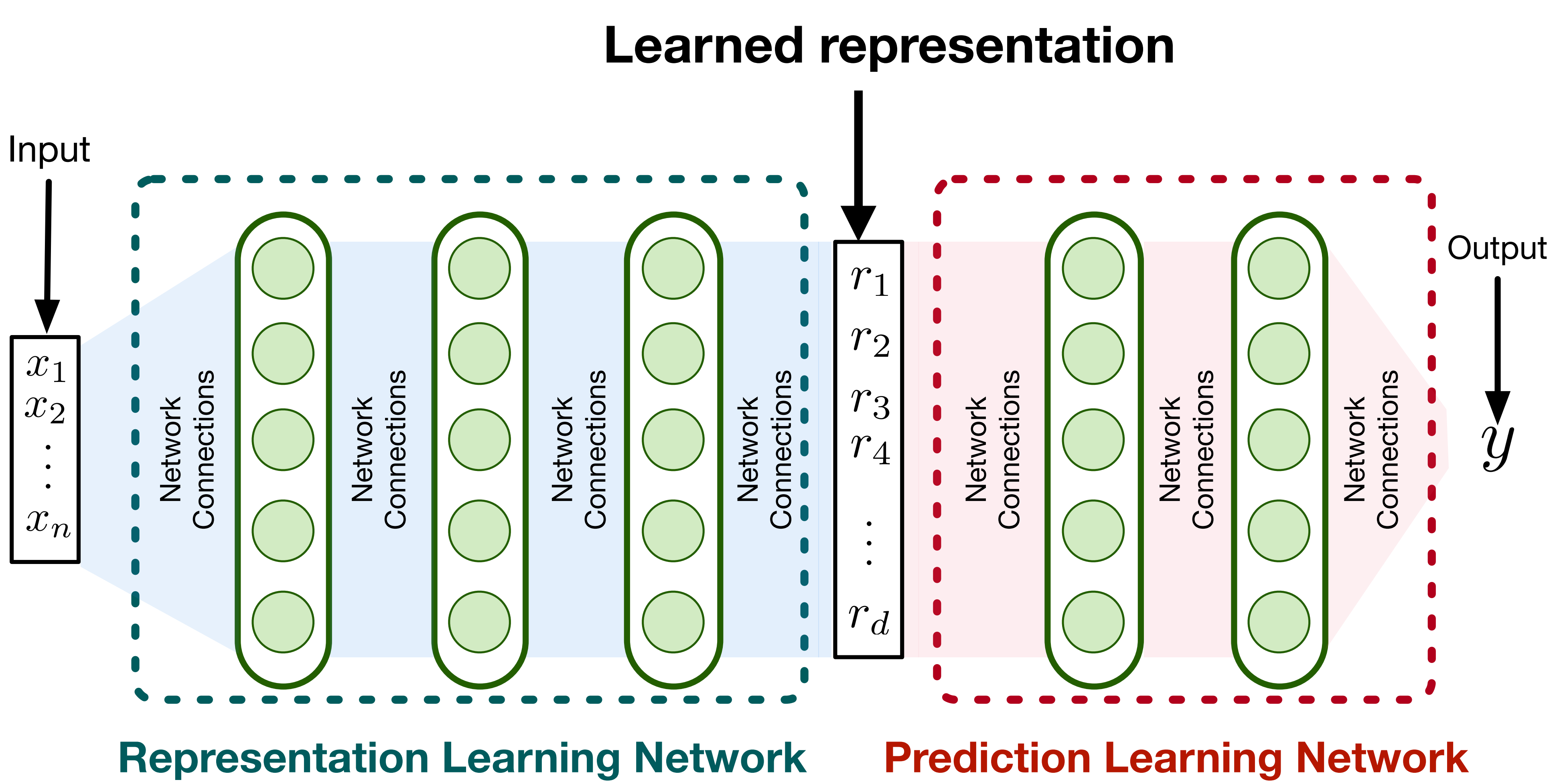
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600



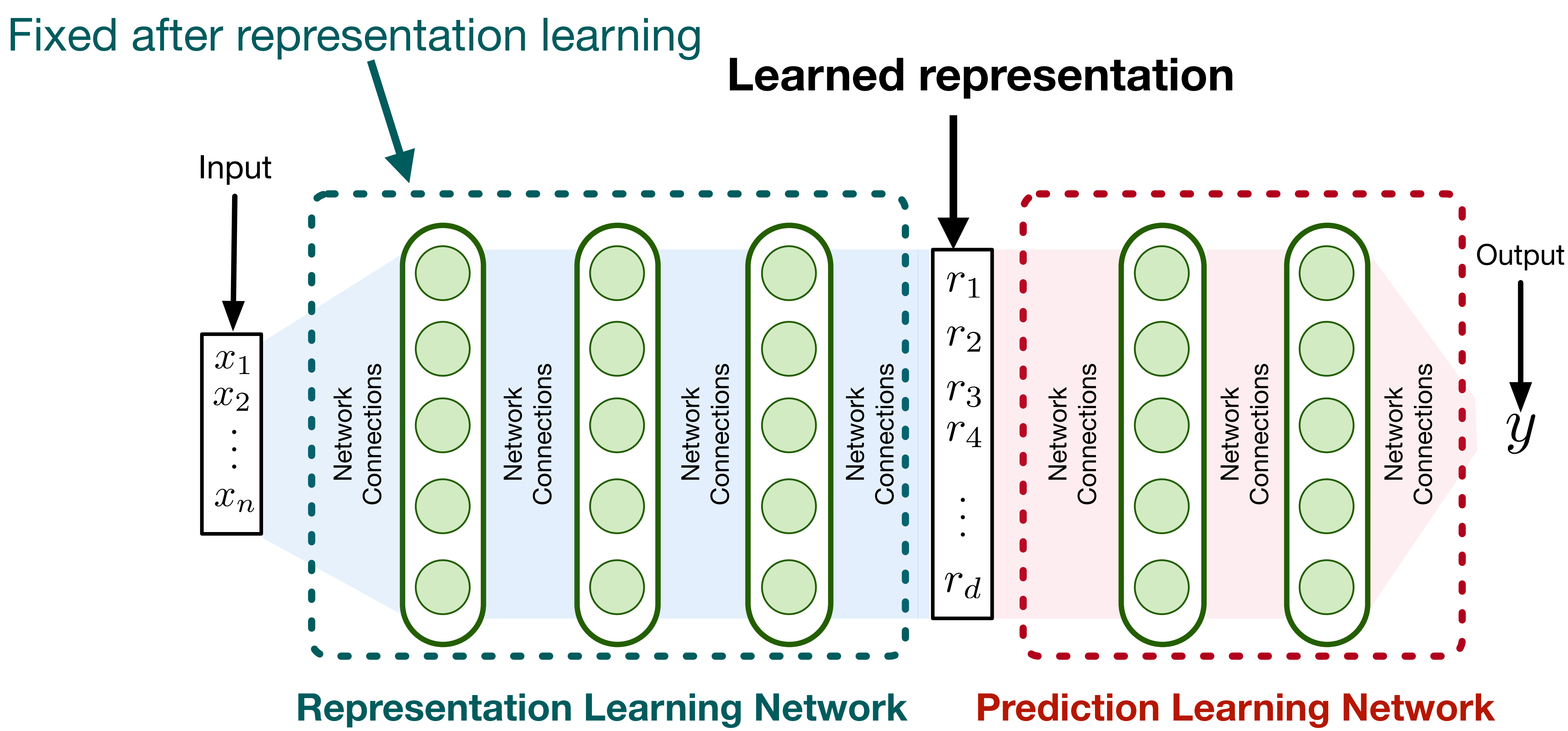
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600



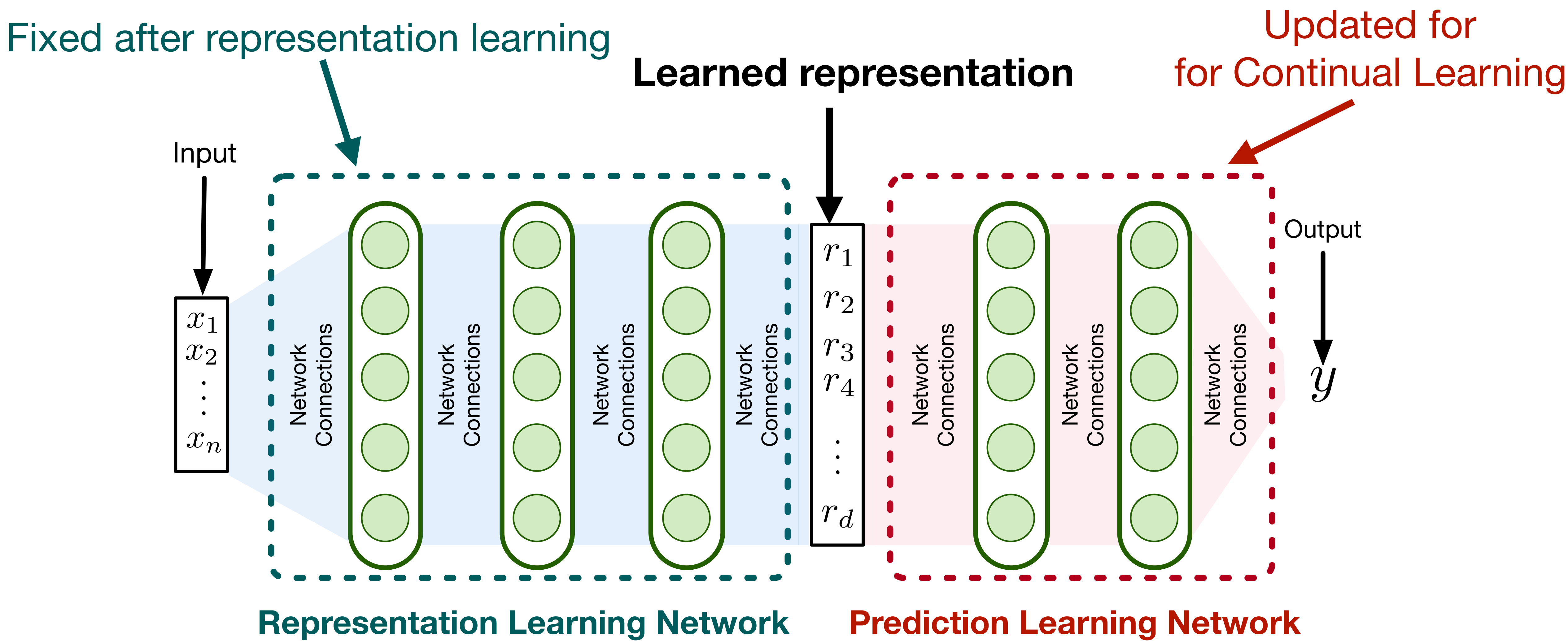
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Plan

Learn a representation on the ~950 characters, evaluate on the same ~600



Dense Inputs

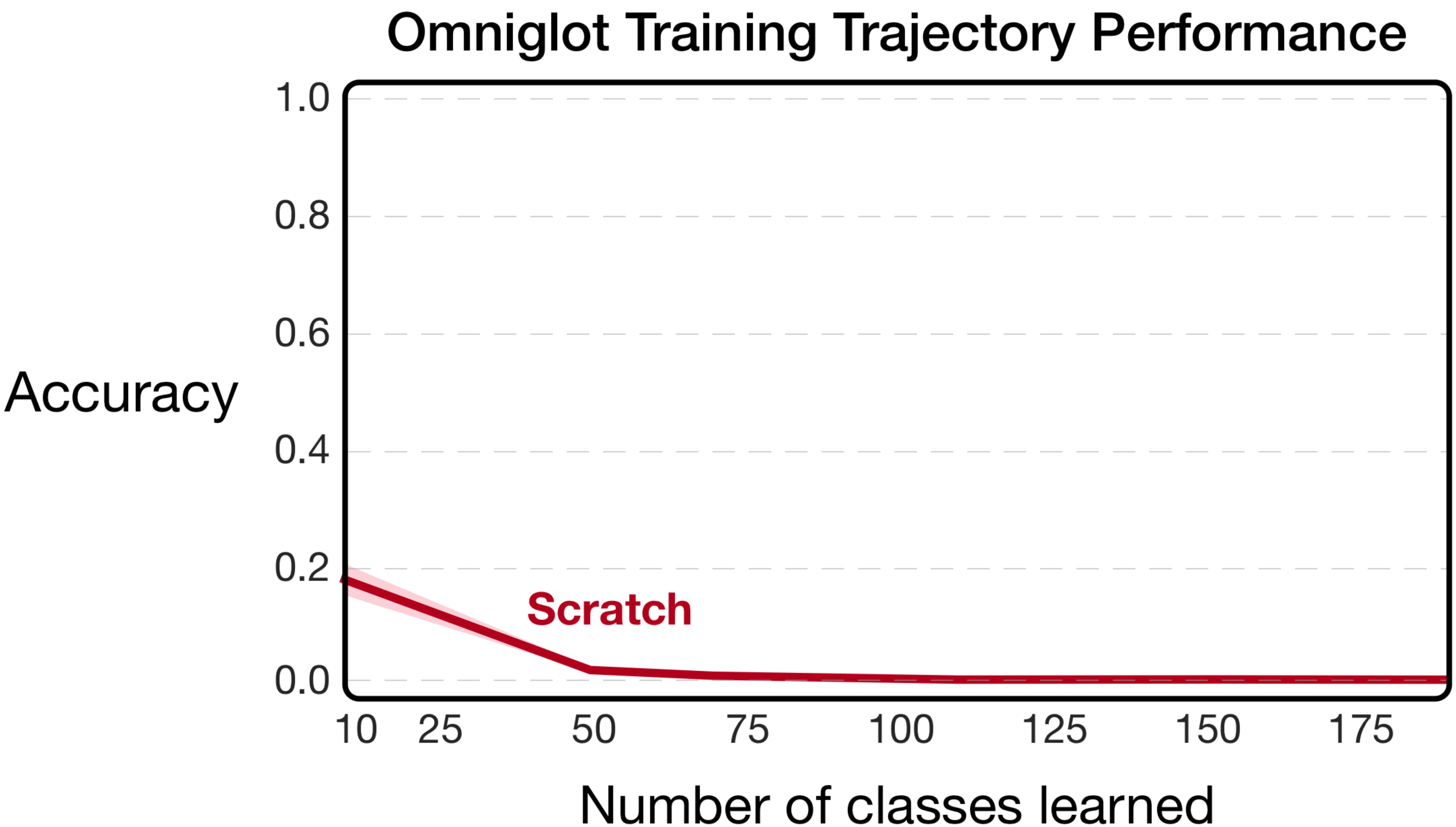
Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

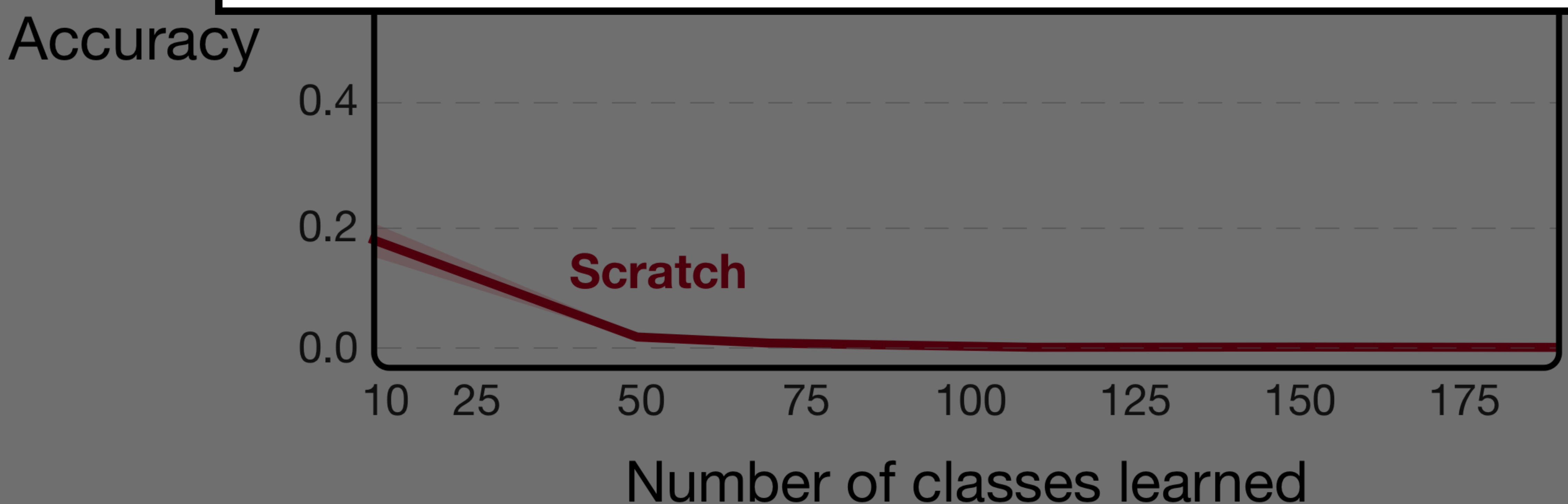


Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Is “Scratch” a fair baseline?



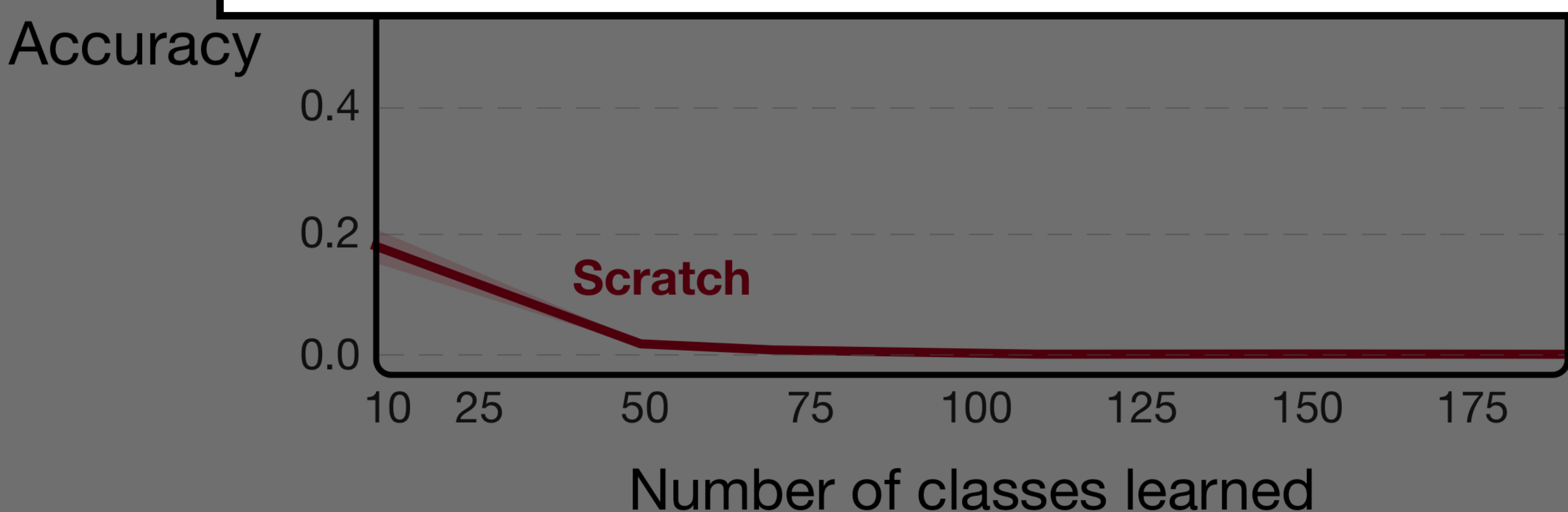
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Is "Scratch" a fair baseline?

- Scratch is not longer a fair baseline



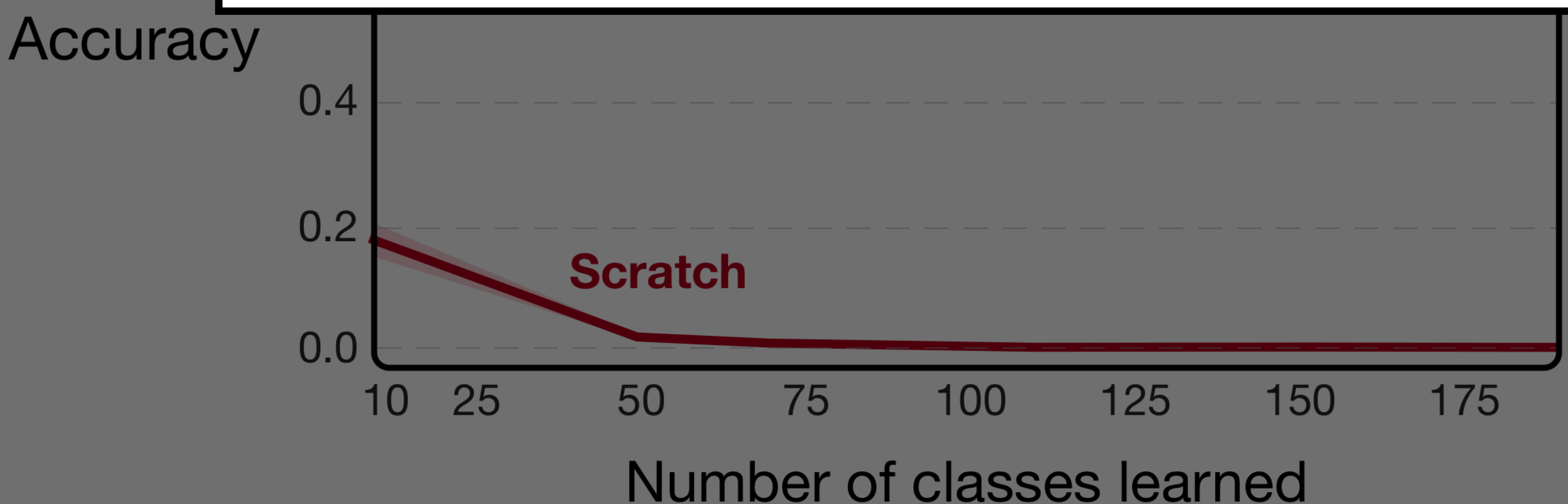
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Is "Scratch" a fair baseline?

- Scratch is not longer a fair baseline
- Need a baseline that uses the information from the representation learning dataset



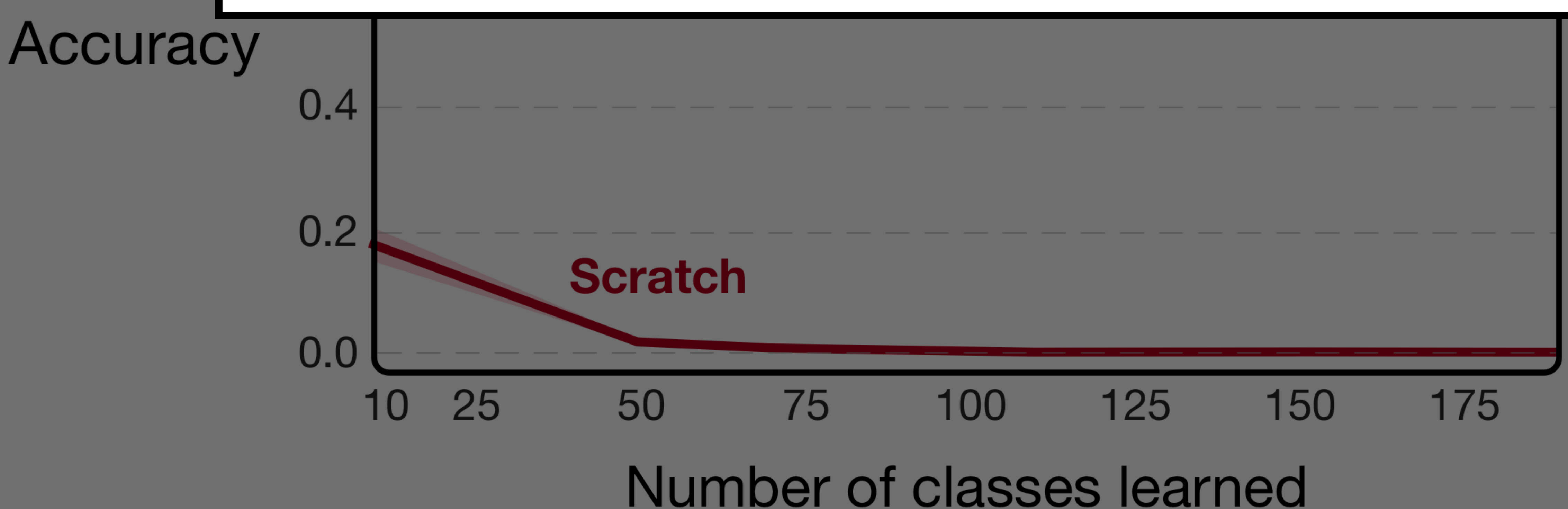
Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Is "Scratch" a fair baseline?

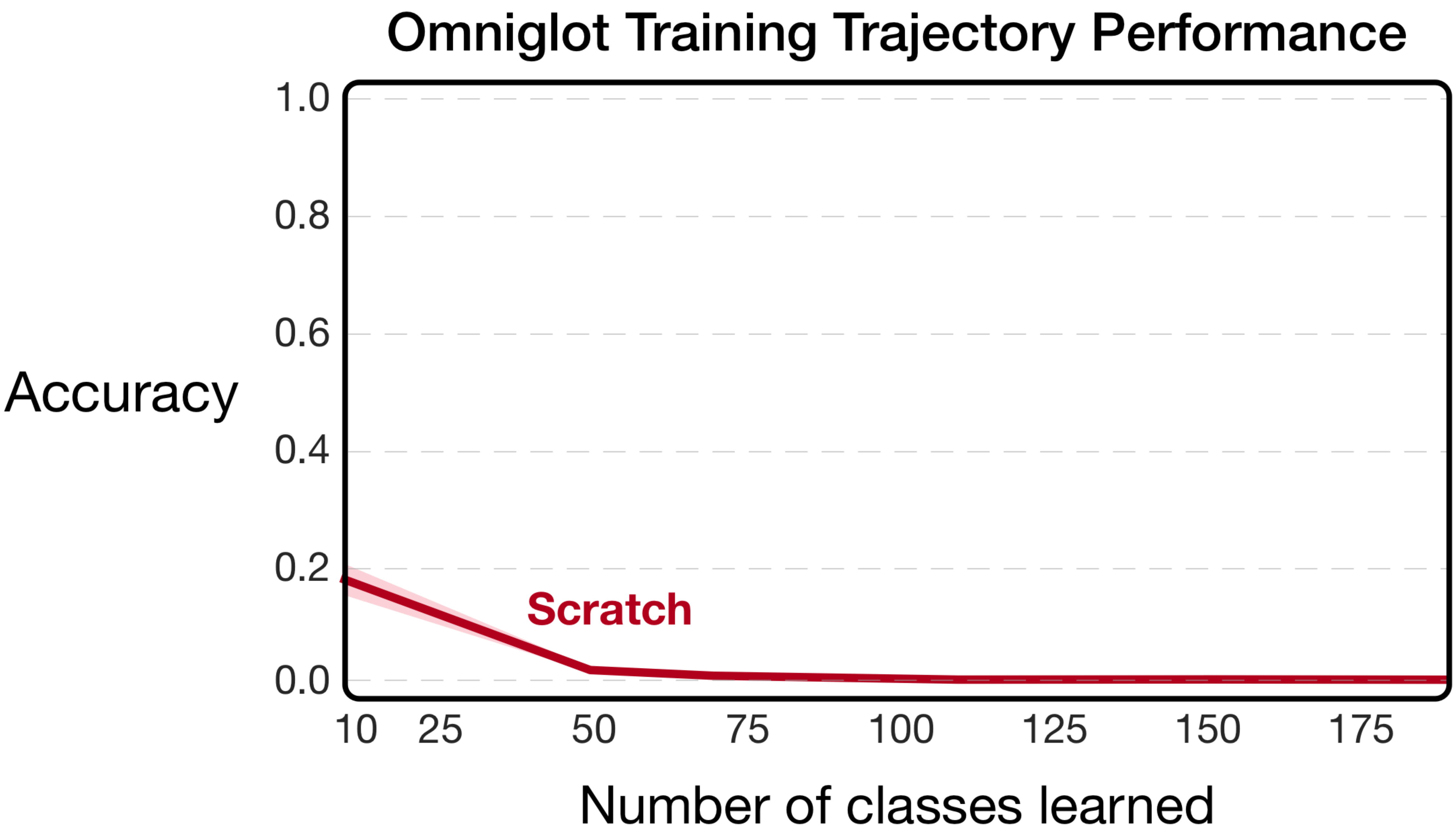
- Scratch is not longer a fair baseline
- Need a baseline that uses the information from the representation learning dataset
- **New baseline :** Learn a 950 way classifier using iid sampling and multiple epochs



Dense Inputs

Dataset : Omniglot

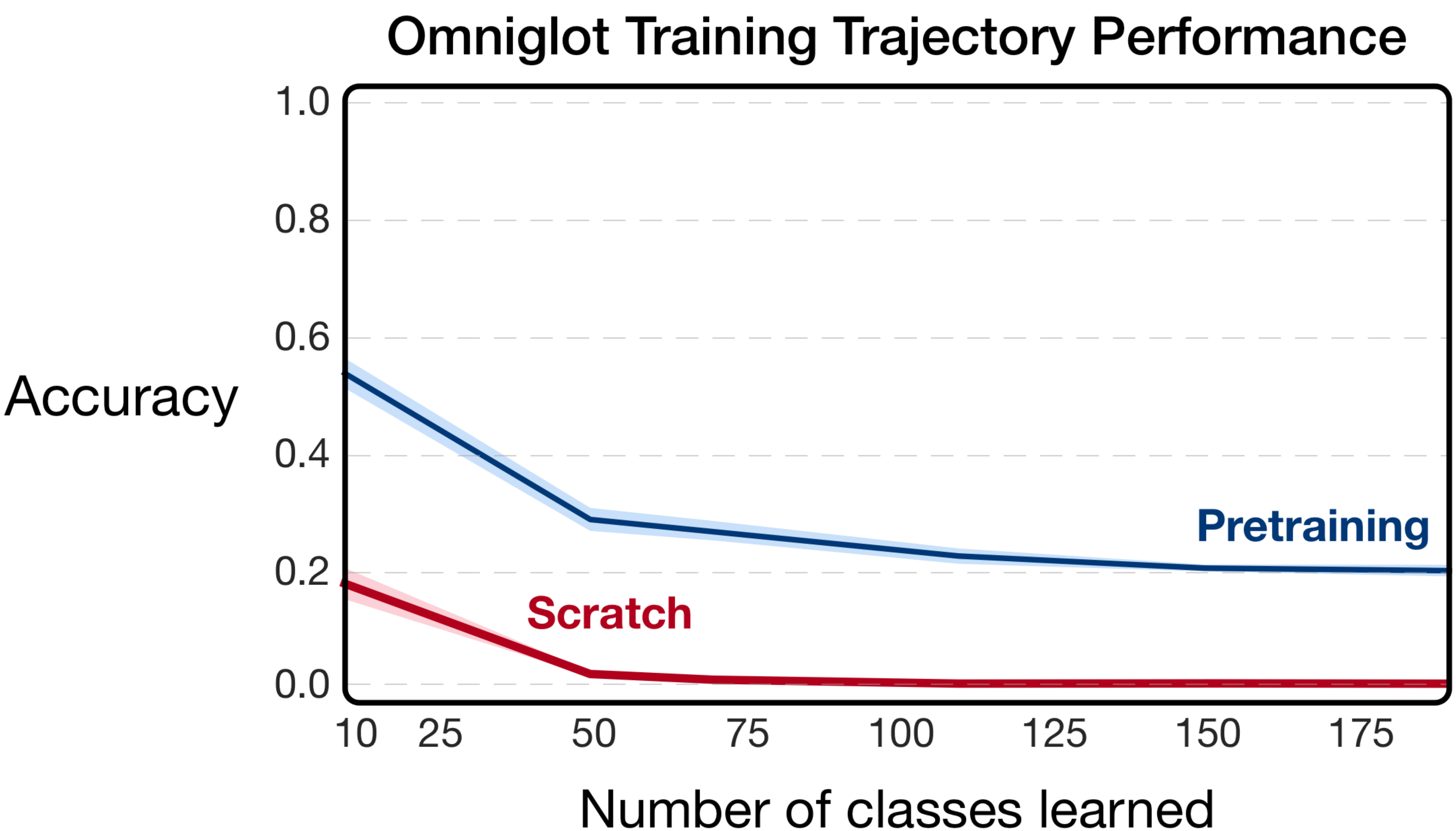
- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Dense Inputs

Dataset : Omniglot

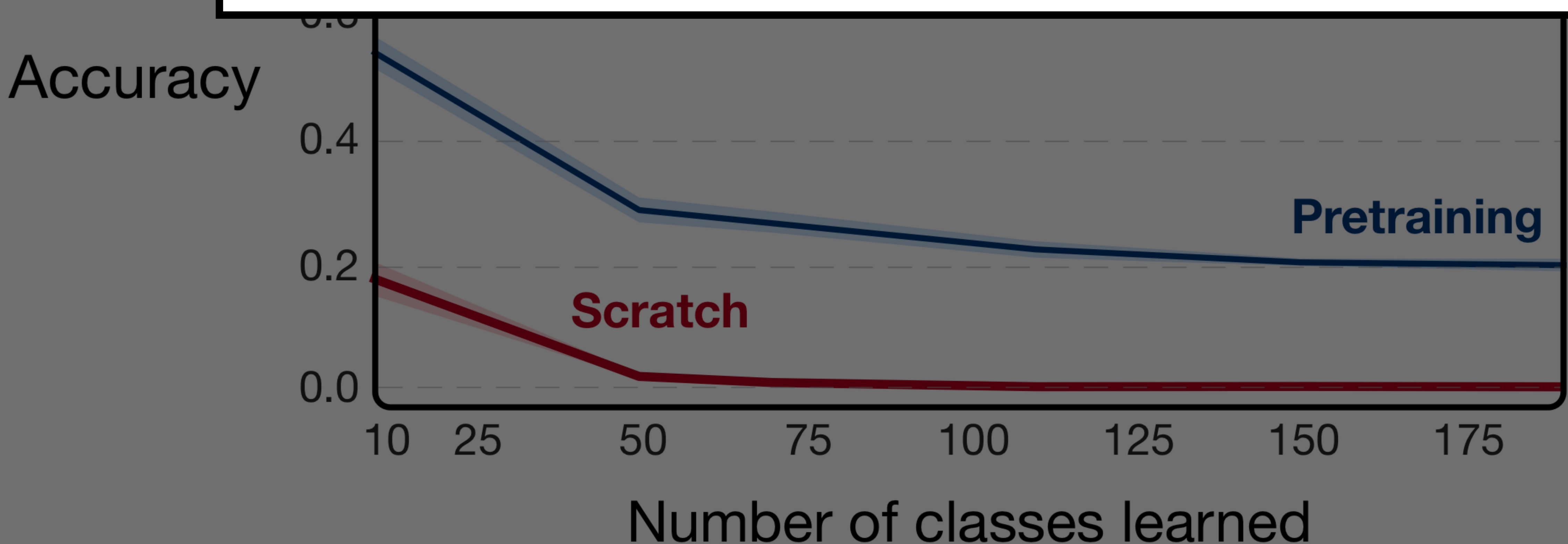
- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

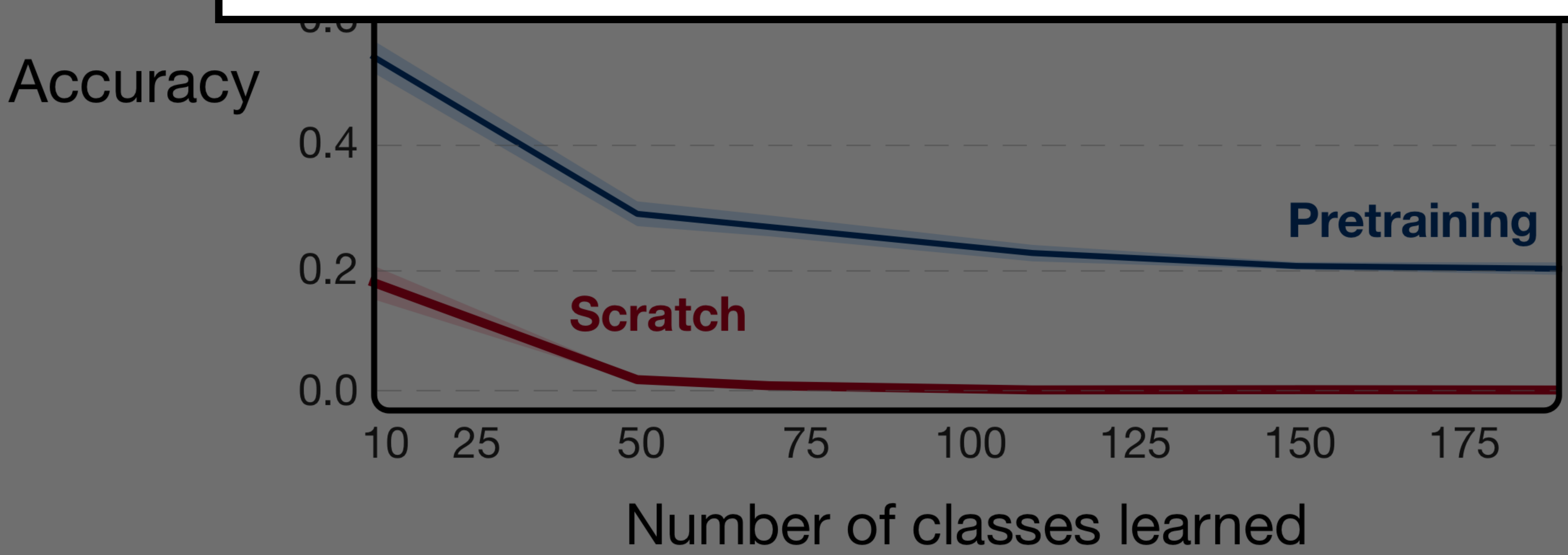


Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Sparse Representations



Dense Inputs

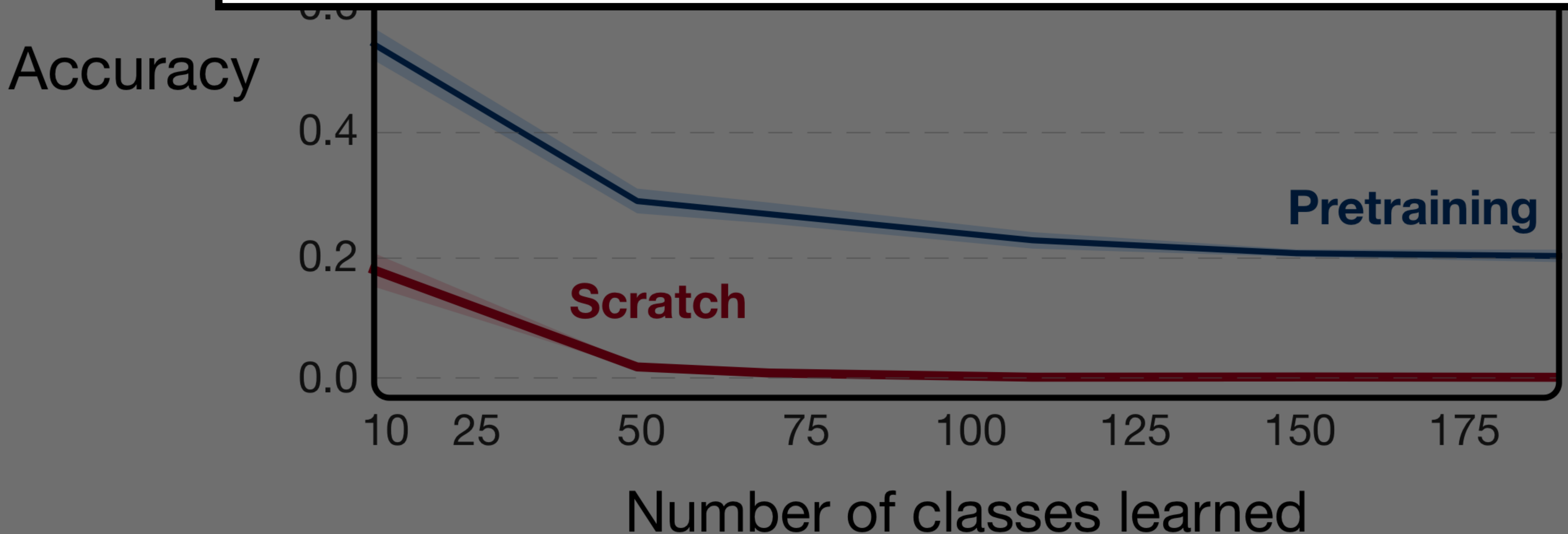
Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Sparse Representations

- We used SR-NN introduced by Vincent *et al.* [1]

[1] Liu, V., Kumaraswamy, R., Le, L., & White, M. (2018). The utility of sparse representations for control in reinforcement learning. AAAI 19



Dense Inputs

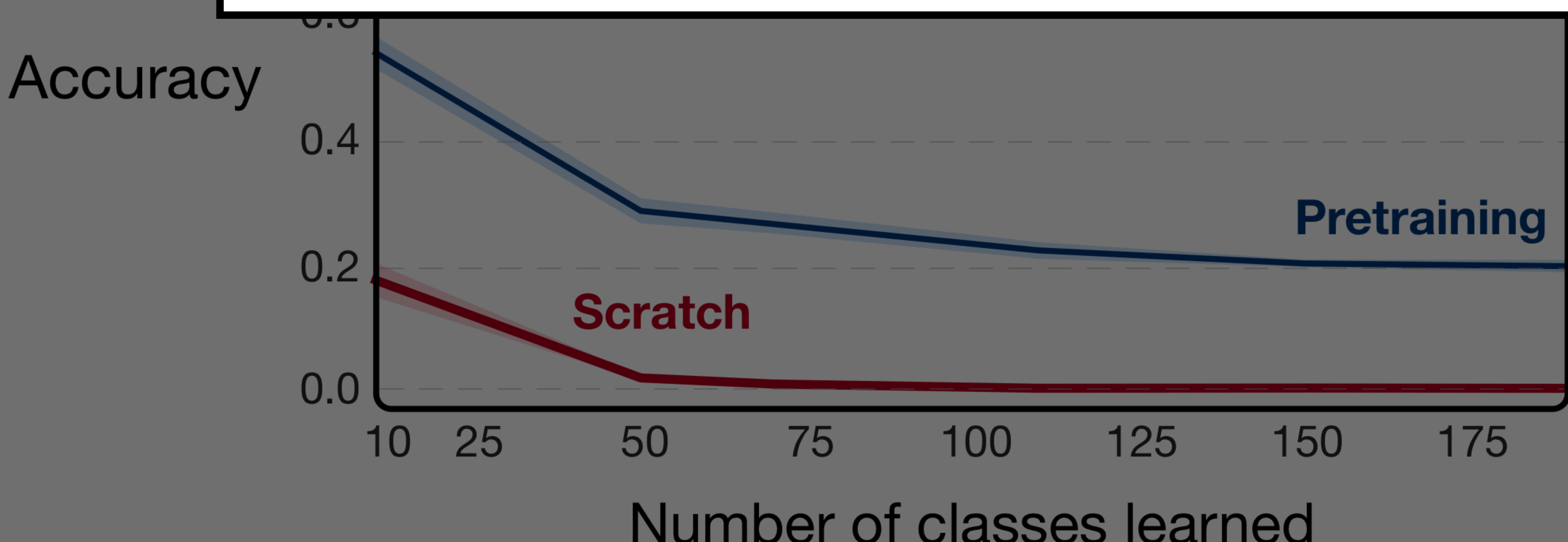
Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction

Sparse Representations

- We used SR-NN introduced by Vincent *et al.* [1]
- Results reported using the best performing SR-NN

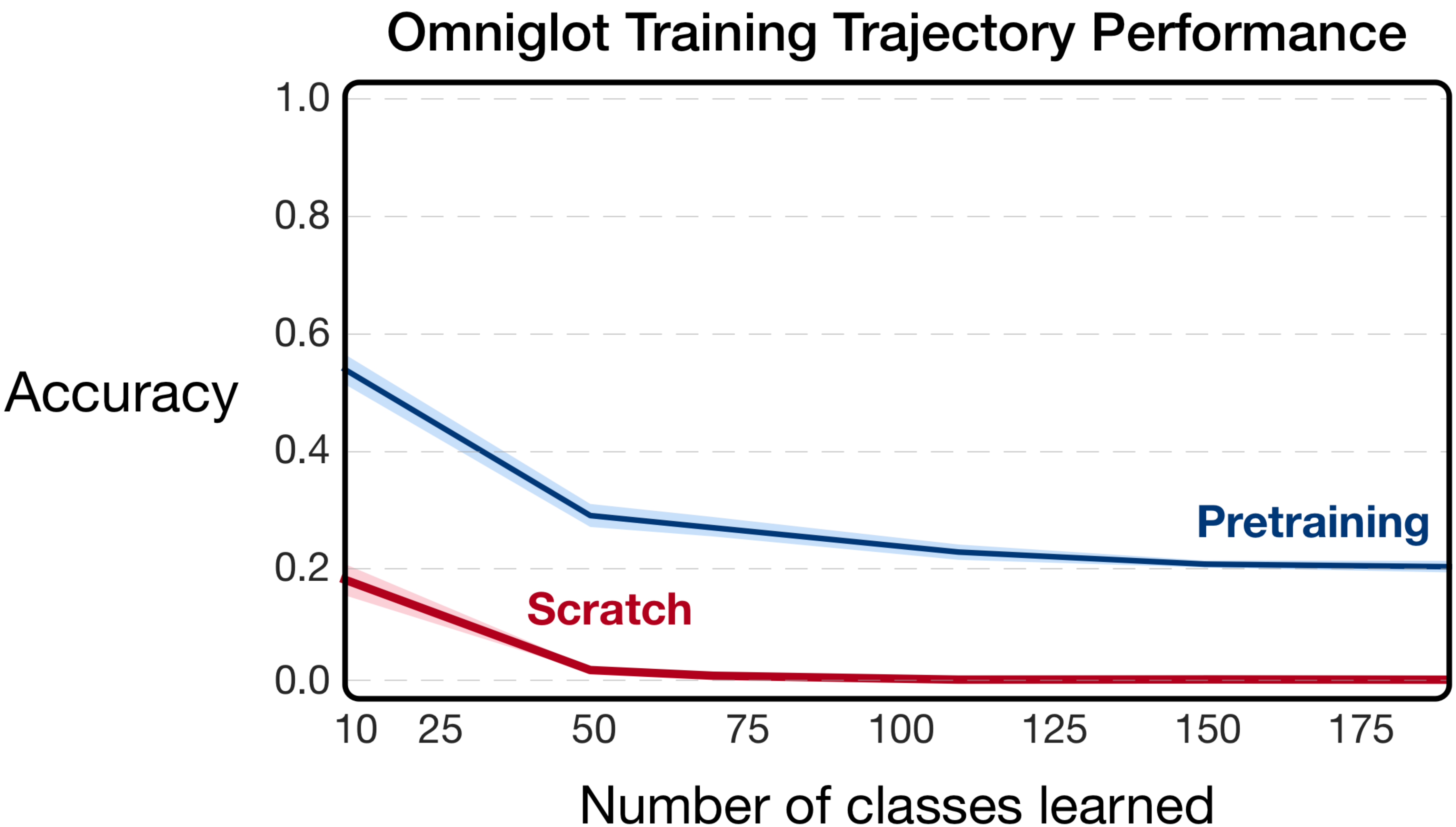
[1] Liu, V., Kumaraswamy, R., Le, L., & White, M. (2018). The utility of sparse representations for control in reinforcement learning. AAAI 19



Dense Inputs

Dataset : Omniglot

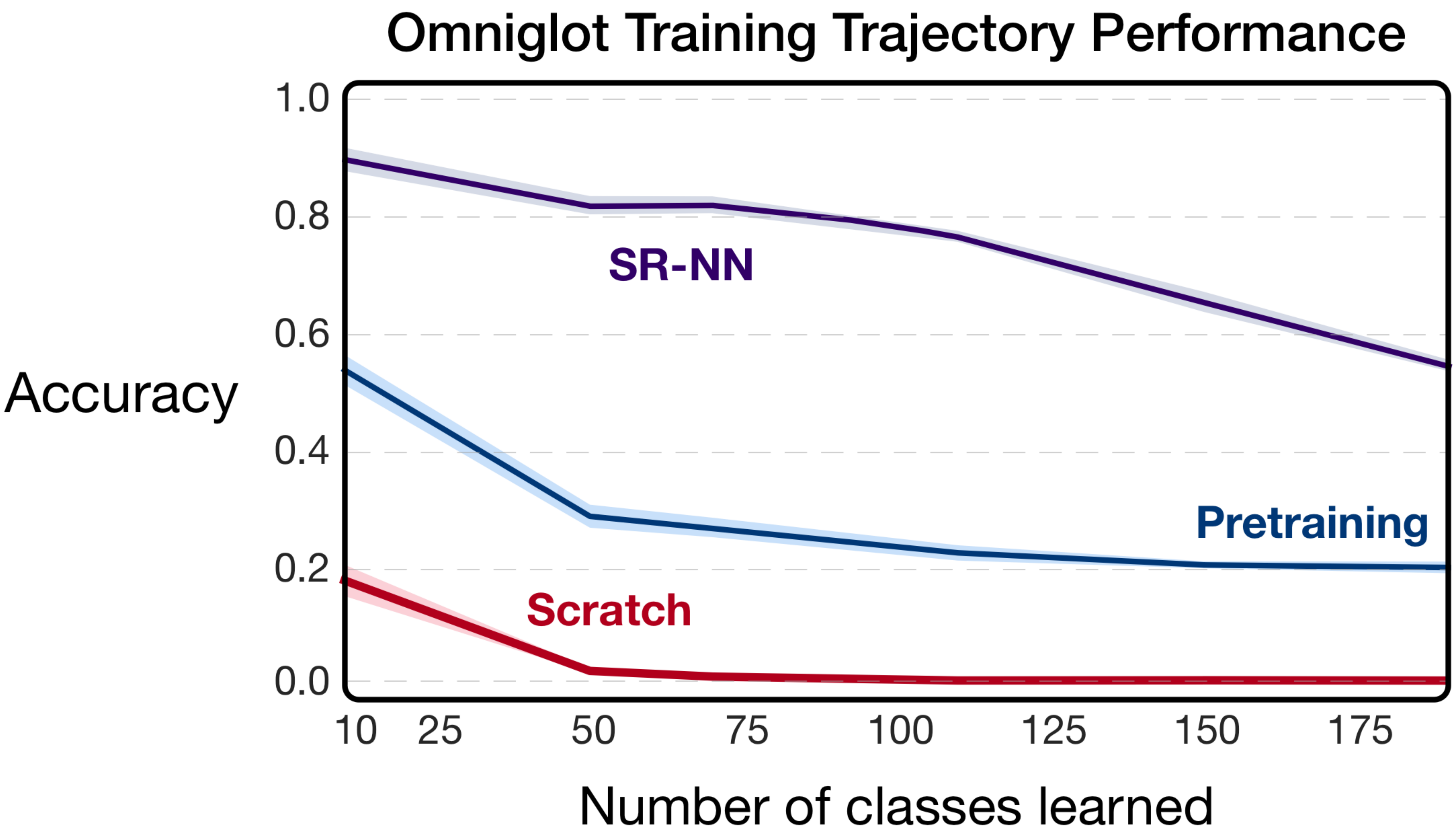
- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Dense Inputs

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling
2. $\boxed{\text{Dense inputs}}$
3. Global and greedy update

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling

2. Dense inputs

3. $\boxed{\text{Global and greedy update}}$

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling
2. Dense inputs
3. Global and greedy update

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling

2. Dense inputs

3. Global and greedy update

Experience Replay

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling — **Experience Replay**
2. Dense inputs — **Tile coding**
3. Global and greedy update

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

When and why?

1. Non-IID sampling — **Experience Replay**
2. Dense inputs — **Tile coding**
3. Global and greedy update — **Adam optimizer?**

Catastrophic Interference

Neural networks suffer from ***catastrophic*** interference

Consequences : ***Forgetting*** and ***slow learning***

Our work

When and why?

1. Non-IID sampling — **Experience Replay**
2. Dense inputs — **Tile coding**
3. Global and greedy update — **Adam optimizer?**

Meta-Learning Representations for Continual Learning

Meta-Learning Representations for Continual Learning

Core idea

Meta-Learning Representations for Continual Learning

Core idea

- Don’t use sparsity as a proxy to a good representation

Meta-Learning Representations for Continual Learning

Core idea

- Don't use sparsity as a proxy to a good representation
- Directly optimize for representations which allow for continual learning

Meta-Learning Representations for Continual Learning

Meta-Learning Representations for Continual Learning

Data stream

$$\boxed{\mathcal{D} = (X_0, Y_0), (X_1, Y_1), \ldots, (X_k, Y_k), \ldots, (X_n, Y_n), \ldots,}$$

Meta-Learning Representations for Continual Learning

Data stream

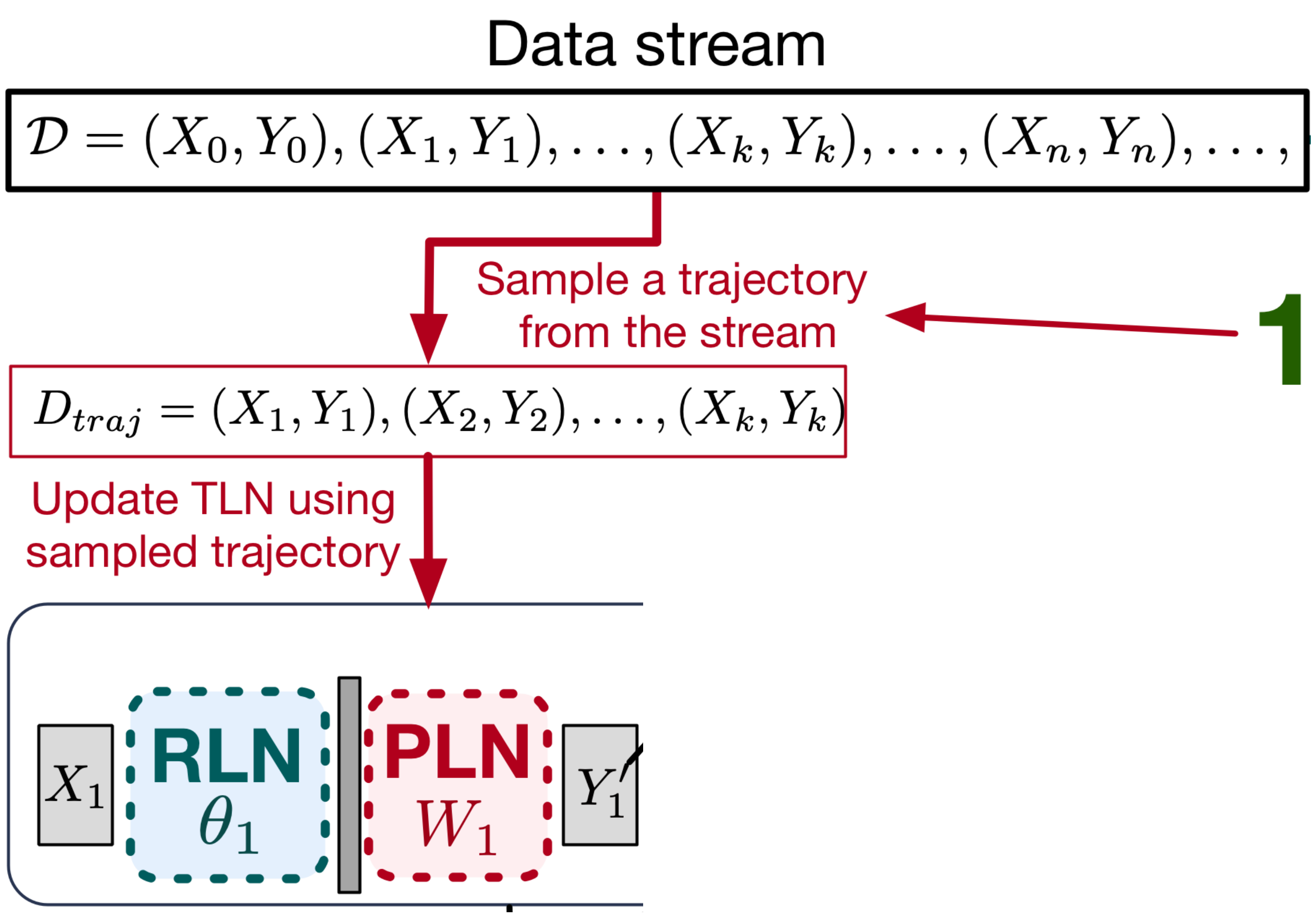
$$\mathcal{D} = (X_0, Y_0), (X_1, Y_1), \ldots, (X_k, Y_k), \ldots, (X_n, Y_n), \ldots,$$

Sample a trajectory from the stream

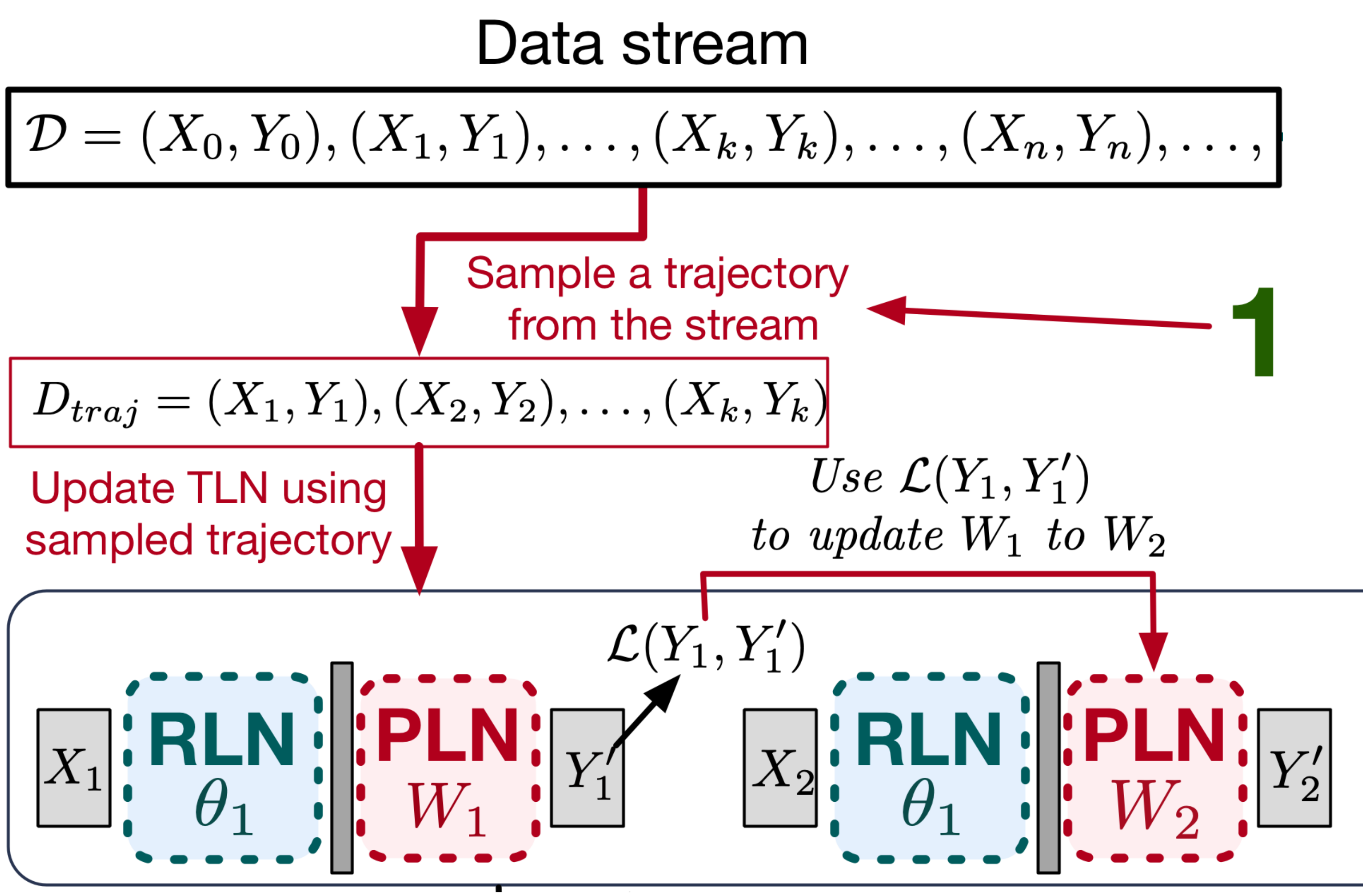
1

$$D_{traj} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_k, Y_k)$$

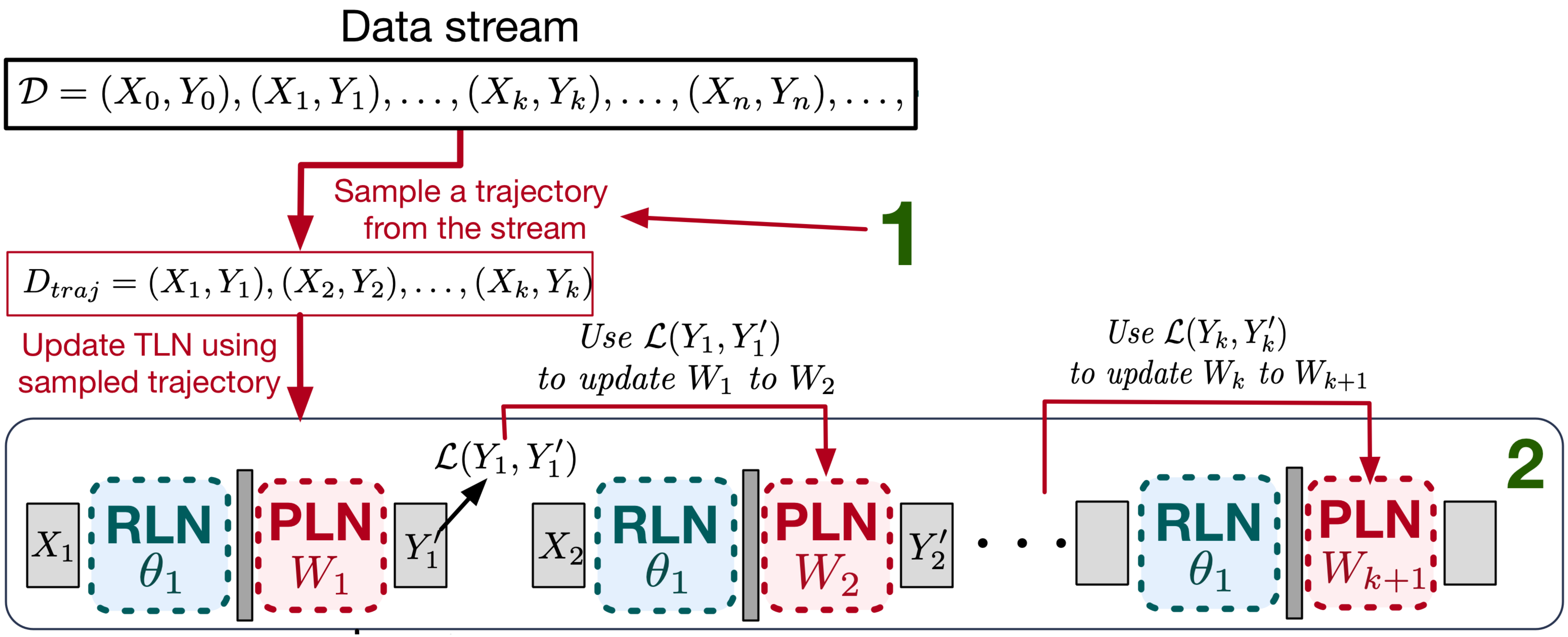
Meta-Learning Representations for Continual Learning



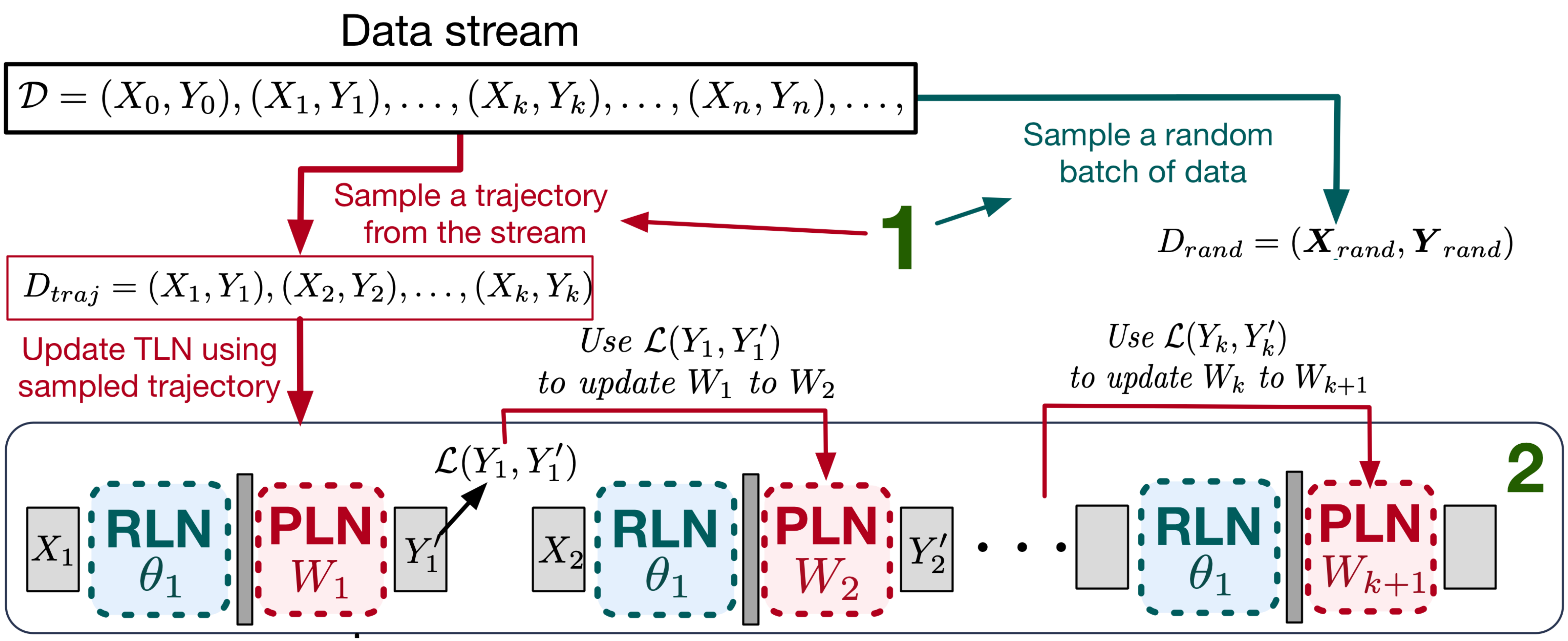
Meta-Learning Representations for Continual Learning



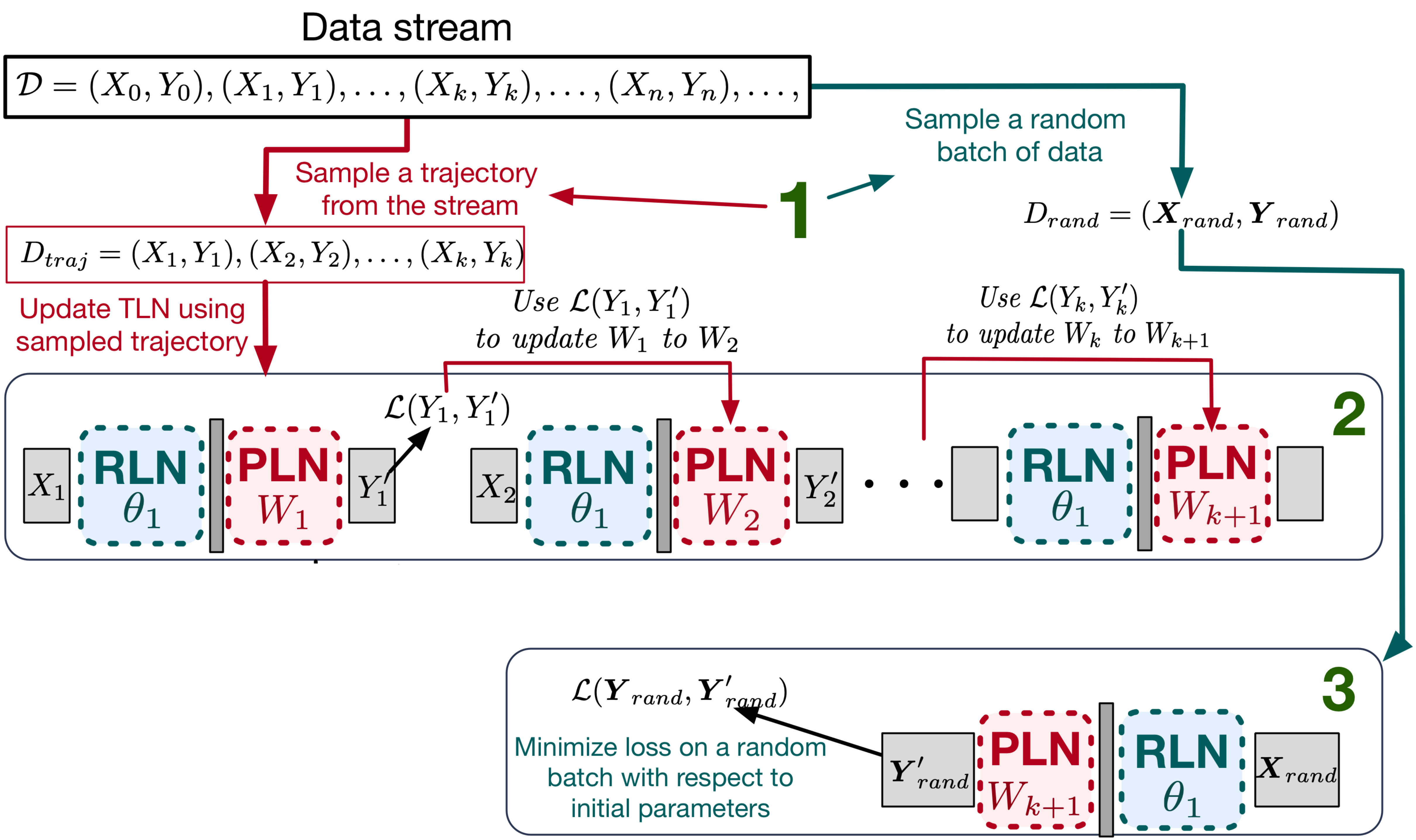
Meta-Learning Representations for Continual Learning



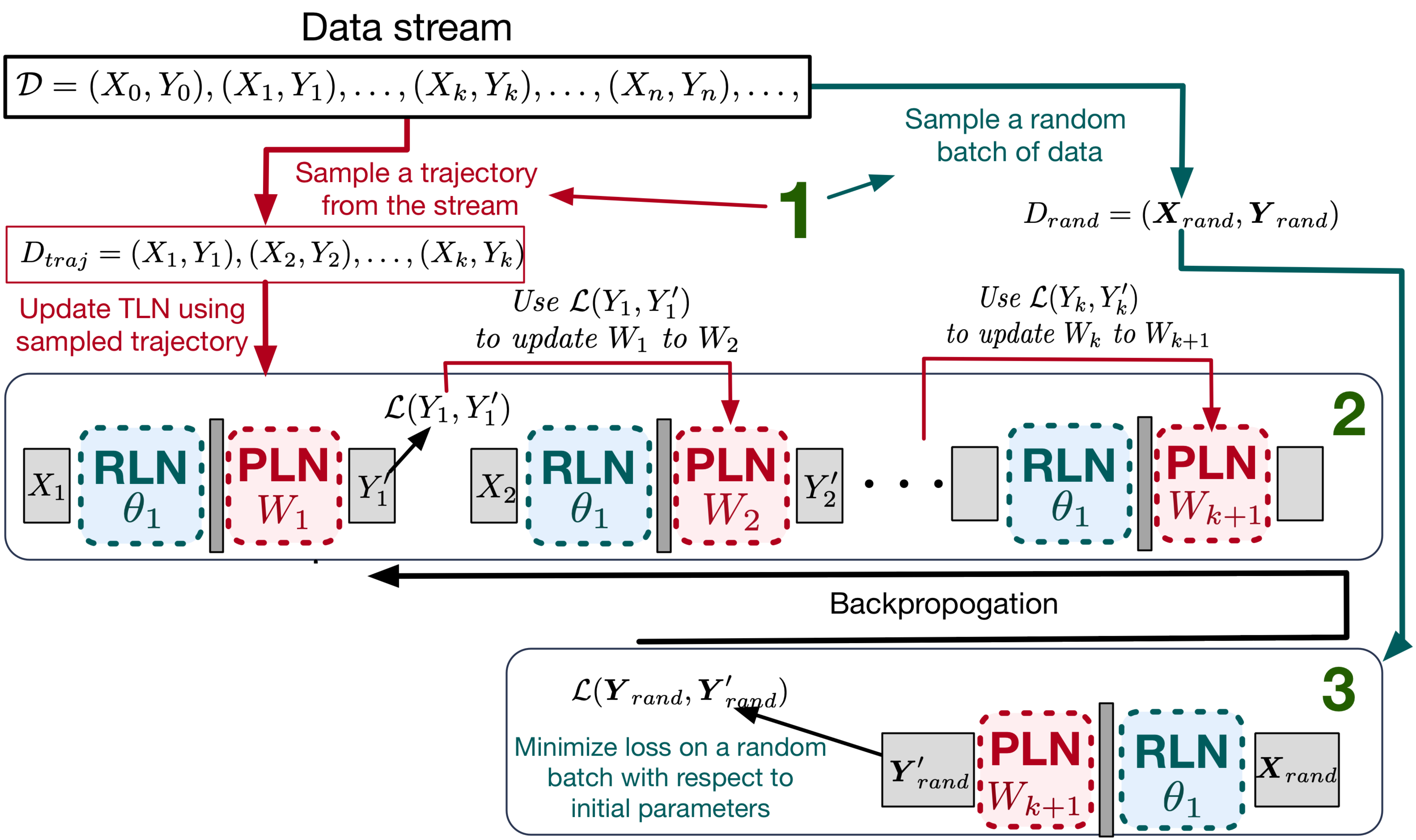
Meta-Learning Representations for Continual Learning



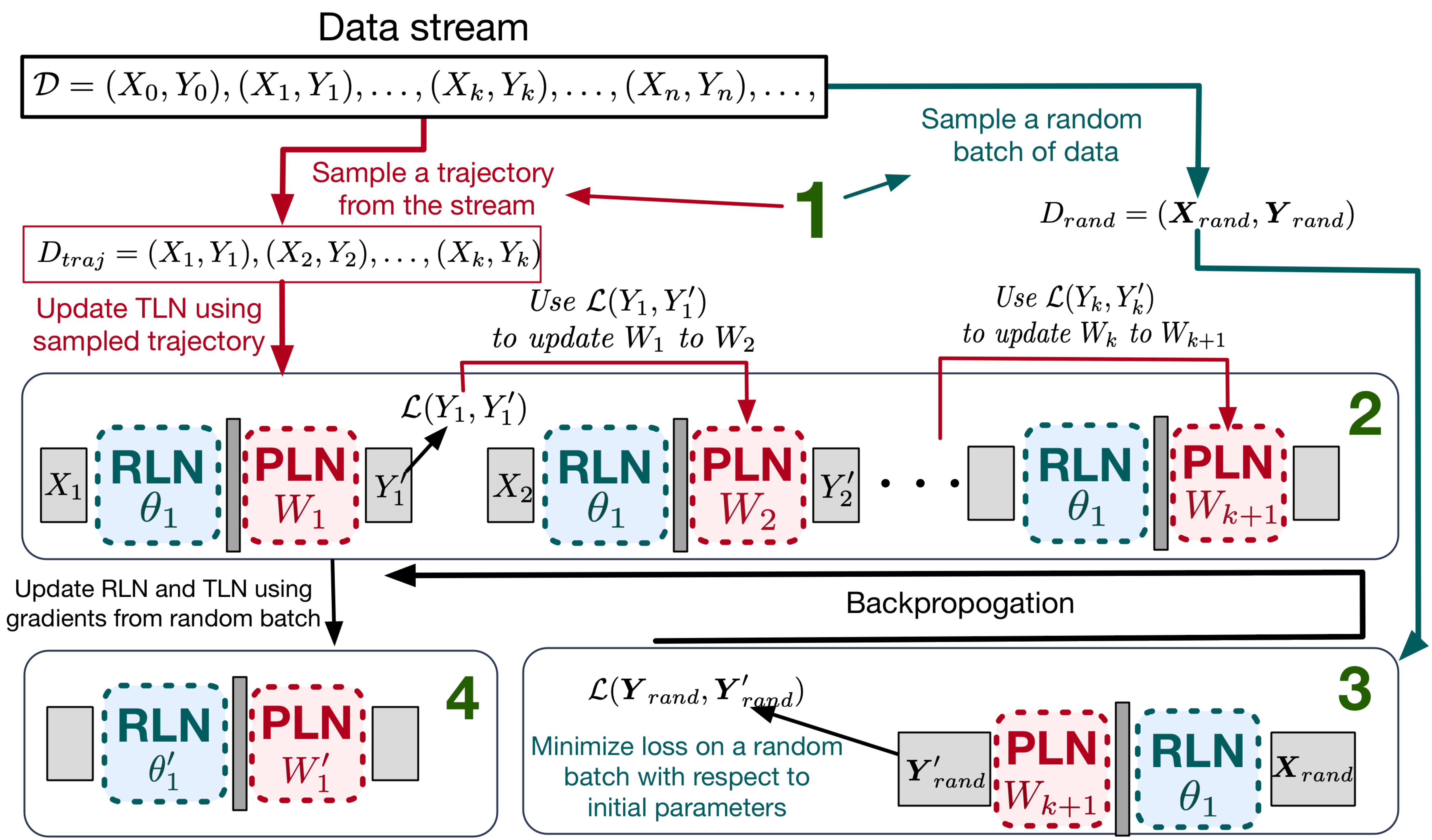
Meta-Learning Representations for Continual Learning



Meta-Learning Representations for Continual Learning

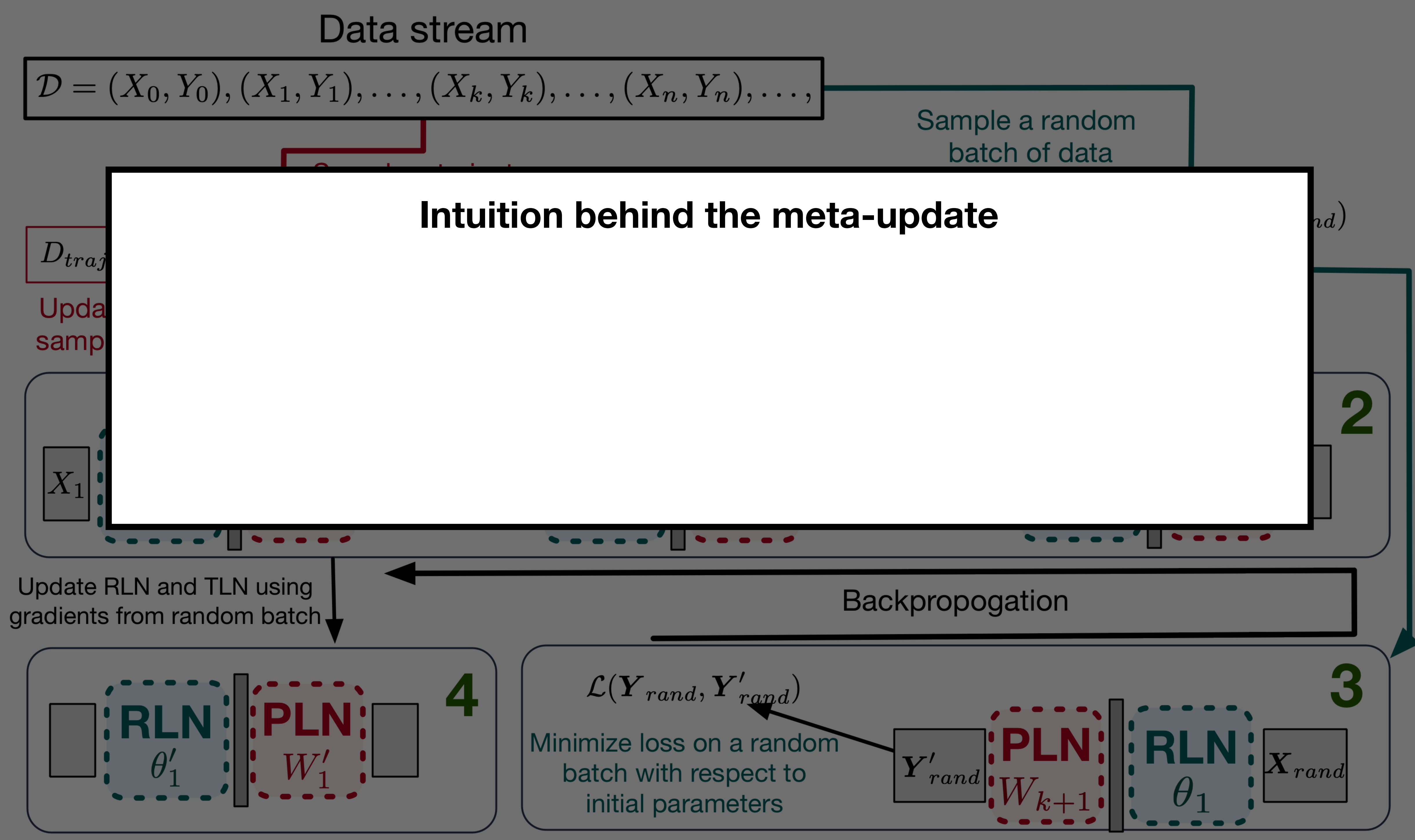


Meta-Learning Representations for Continual Learning

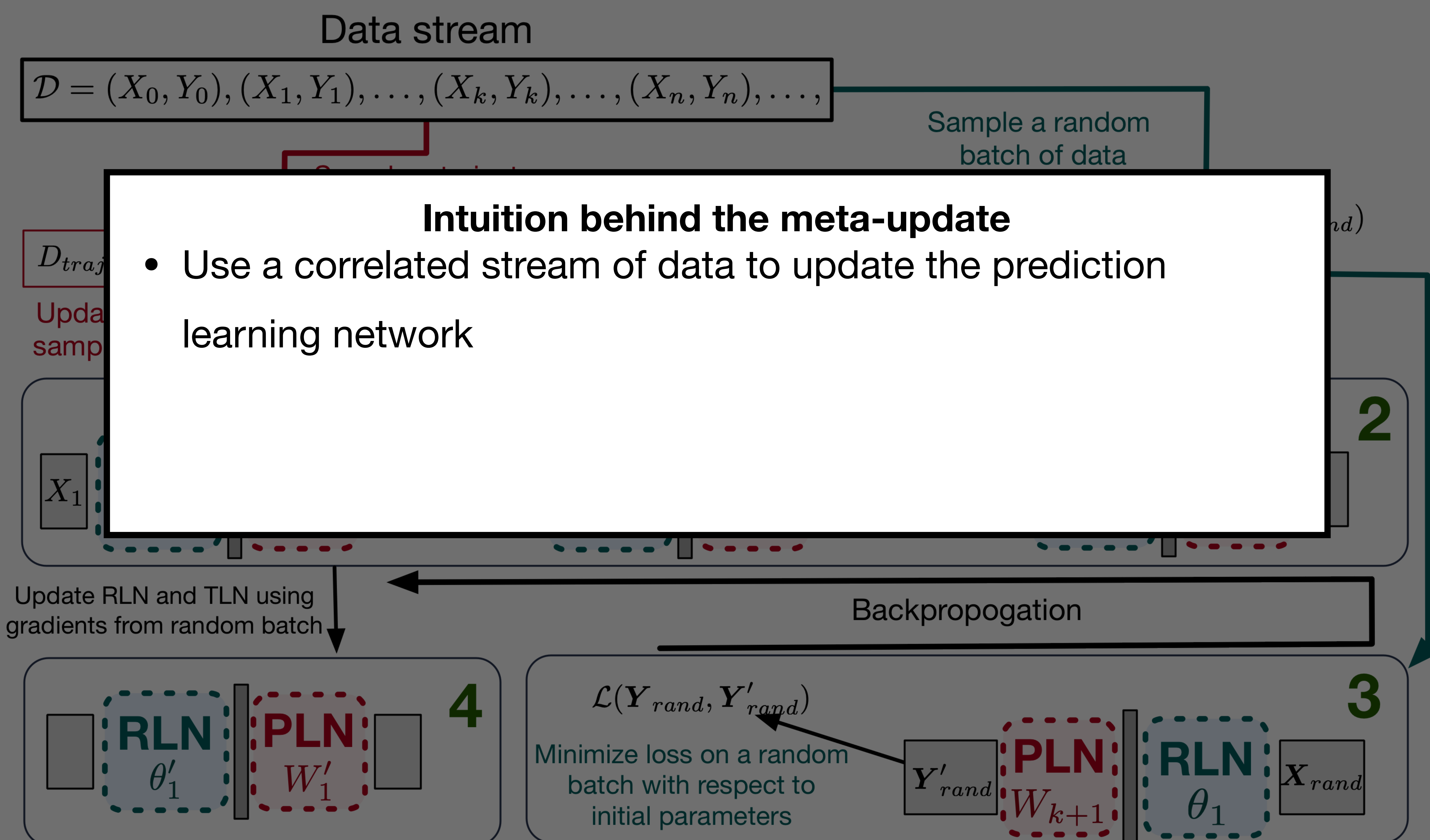


Meta-Learning Representations for Continual Learning

Intuition behind the meta-update



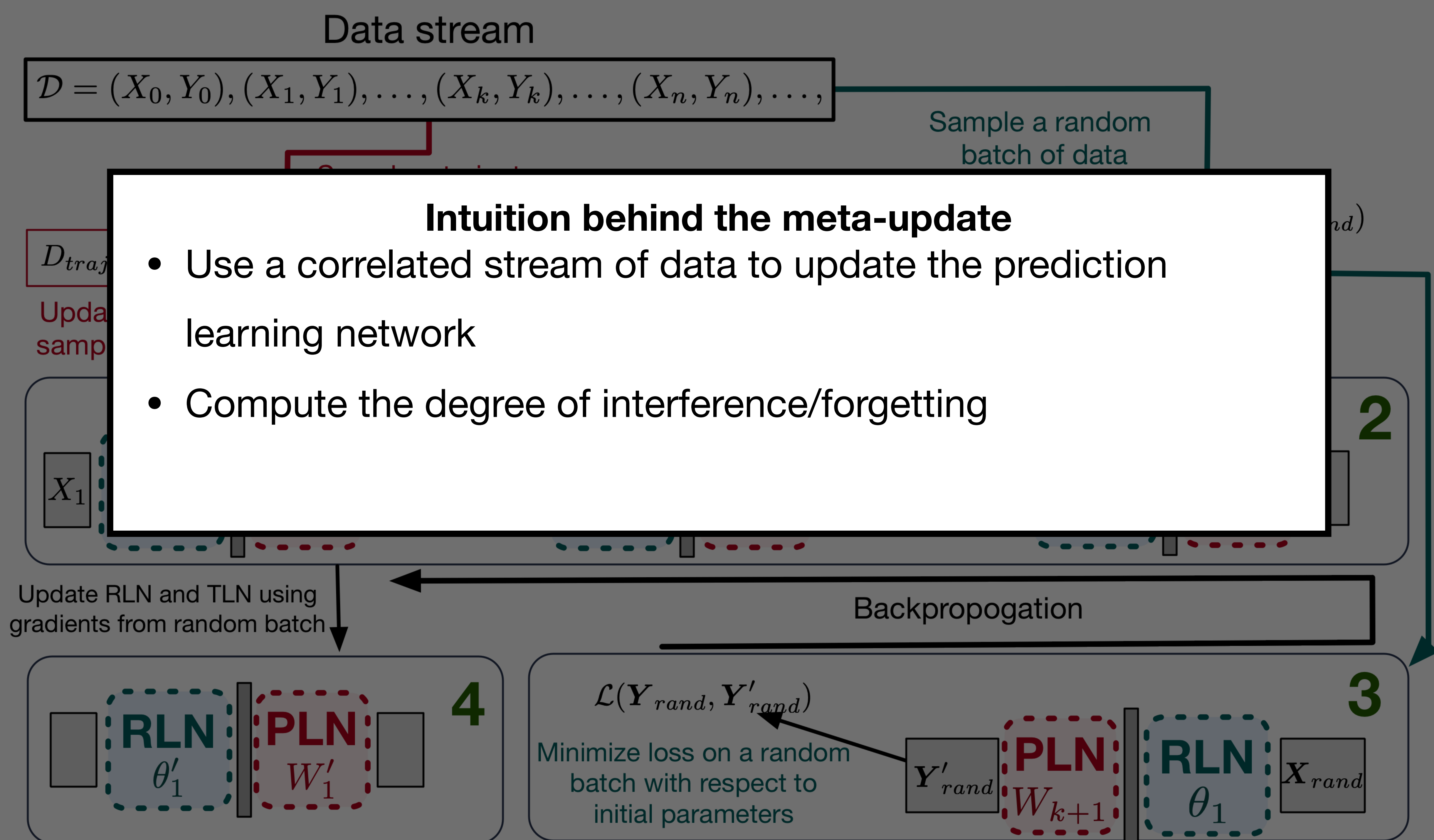
Meta-Learning Representations for Continual Learning



Intuition behind the meta-update

- Use a correlated stream of data to update the prediction learning network

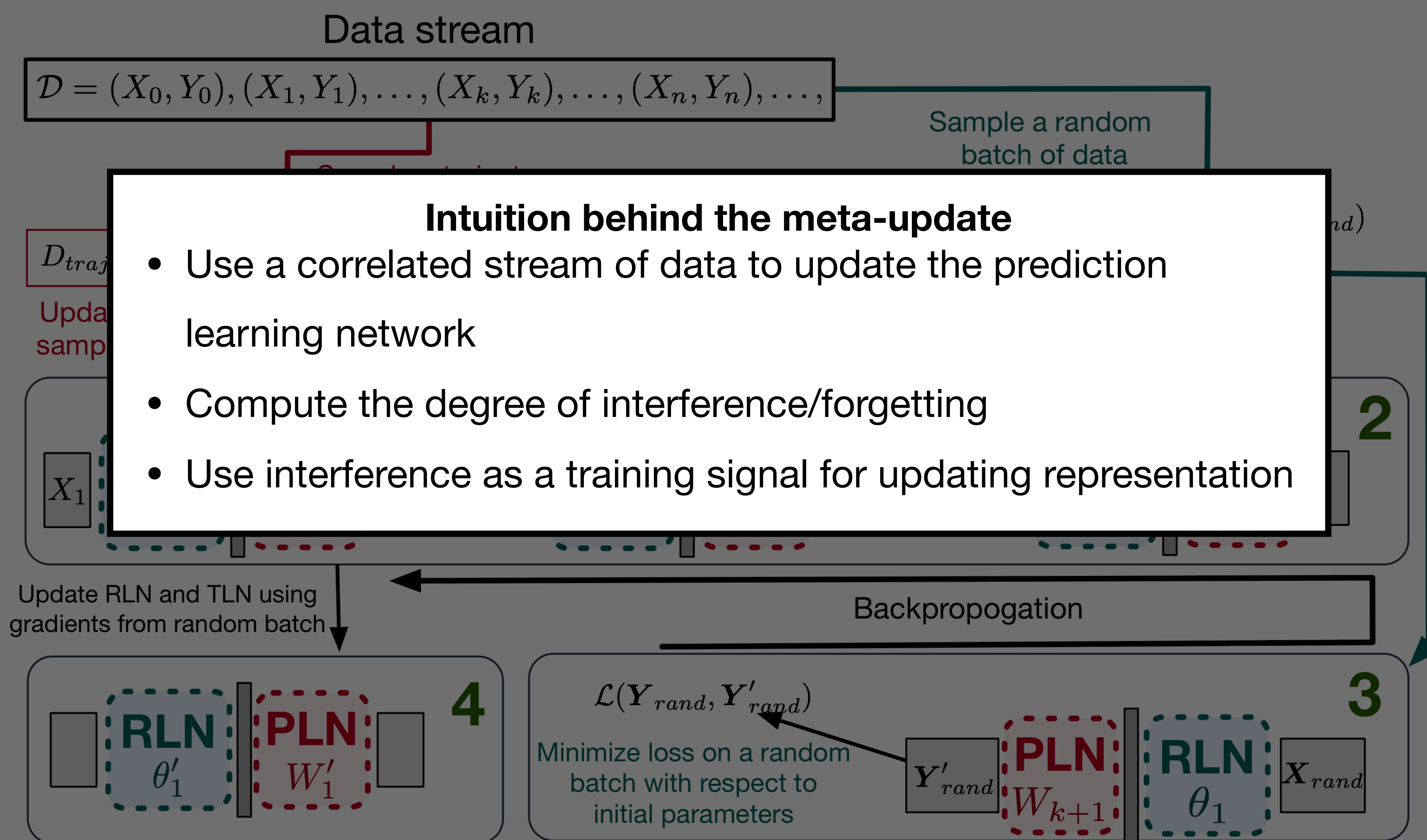
Meta-Learning Representations for Continual Learning



Intuition behind the meta-update

- Use a correlated stream of data to update the prediction learning network
- Compute the degree of interference/forgetting

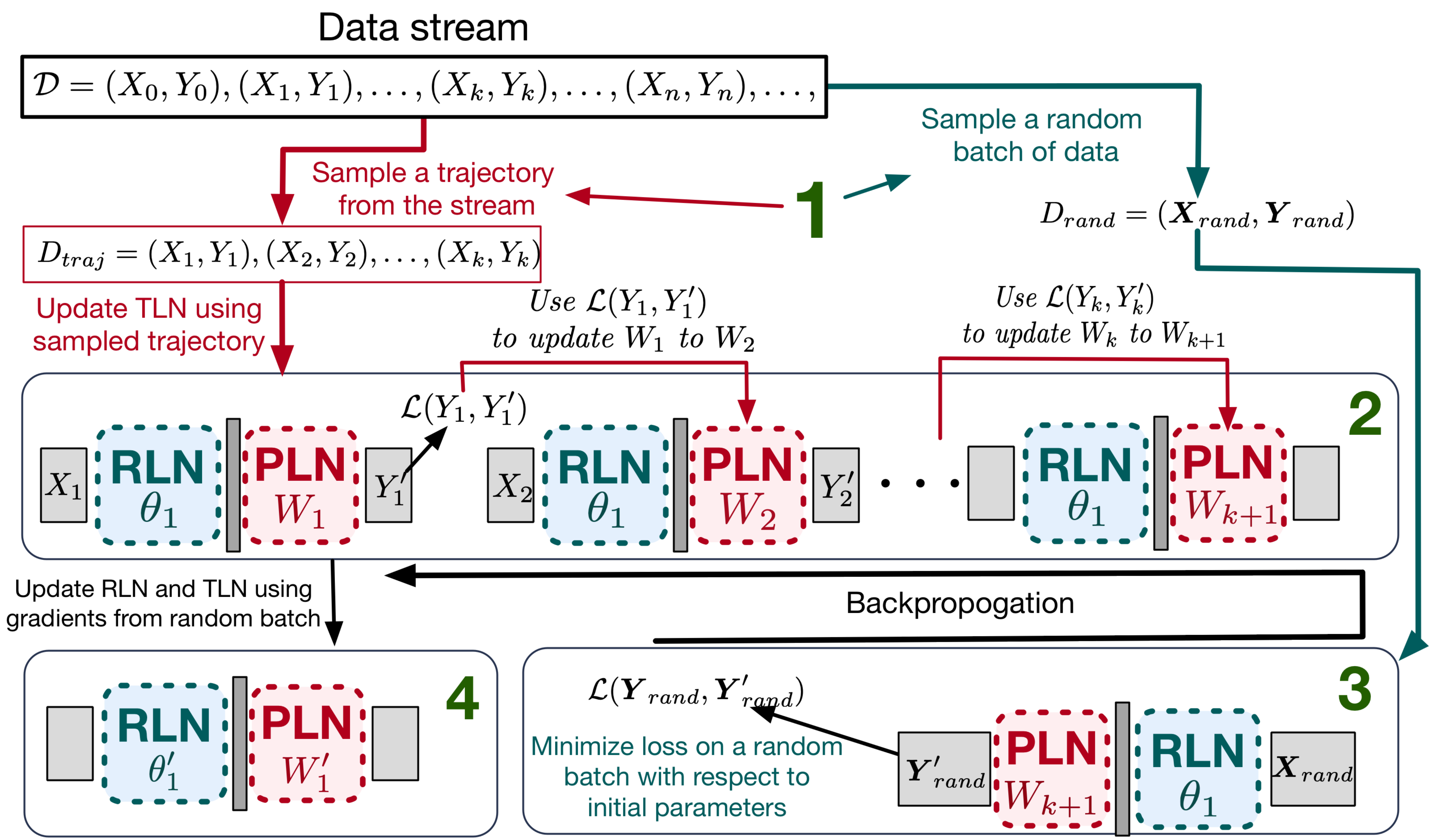
Meta-Learning Representations for Continual Learning



Intuition behind the meta-update

- Use a correlated stream of data to update the prediction learning network
- Compute the degree of interference/forgetting
- Use interference as a training signal for updating representation

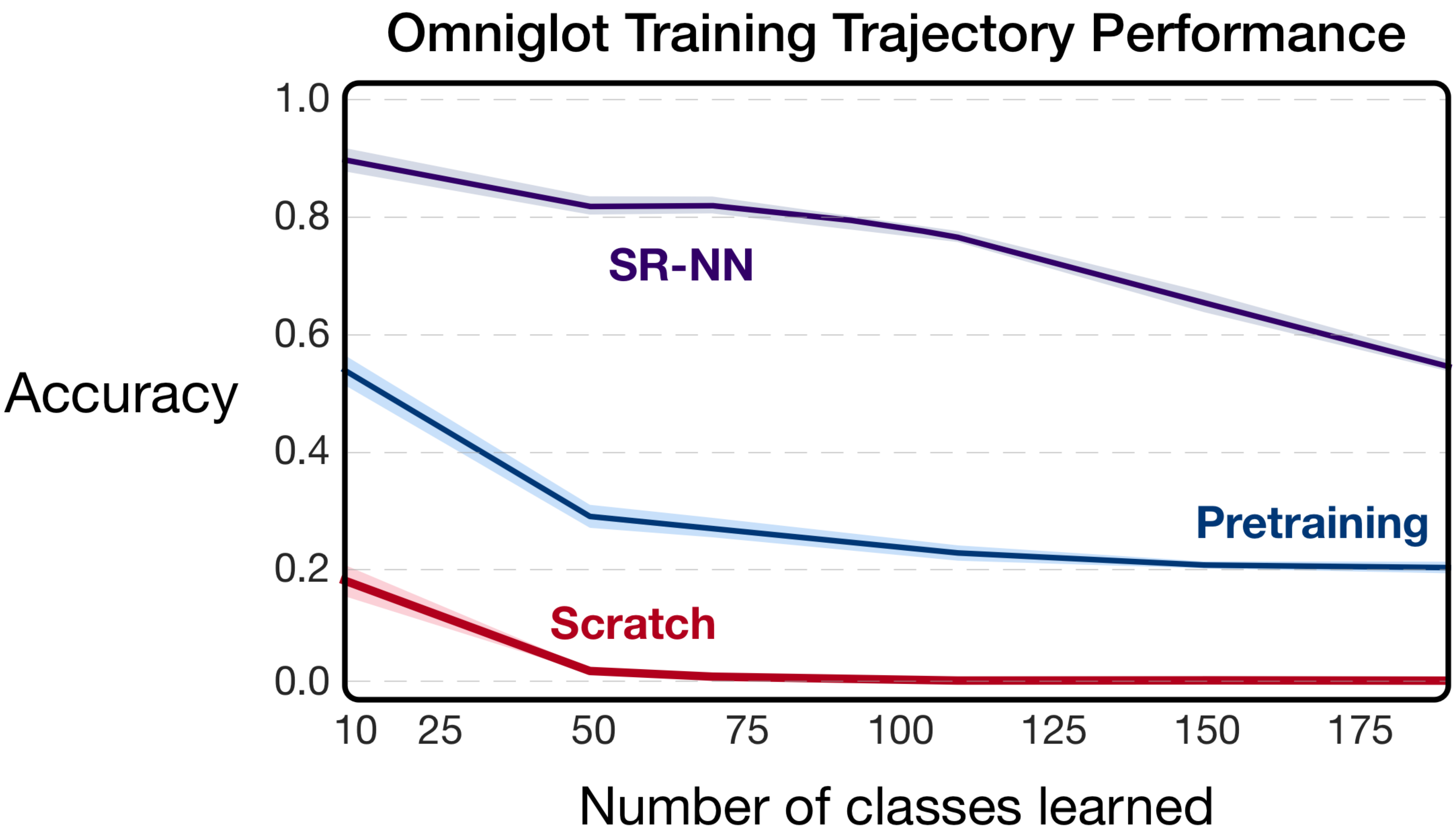
Meta-Learning Representations for Continual Learning



Meta-Learning Representations for Continual Learning

Dataset : Omniglot

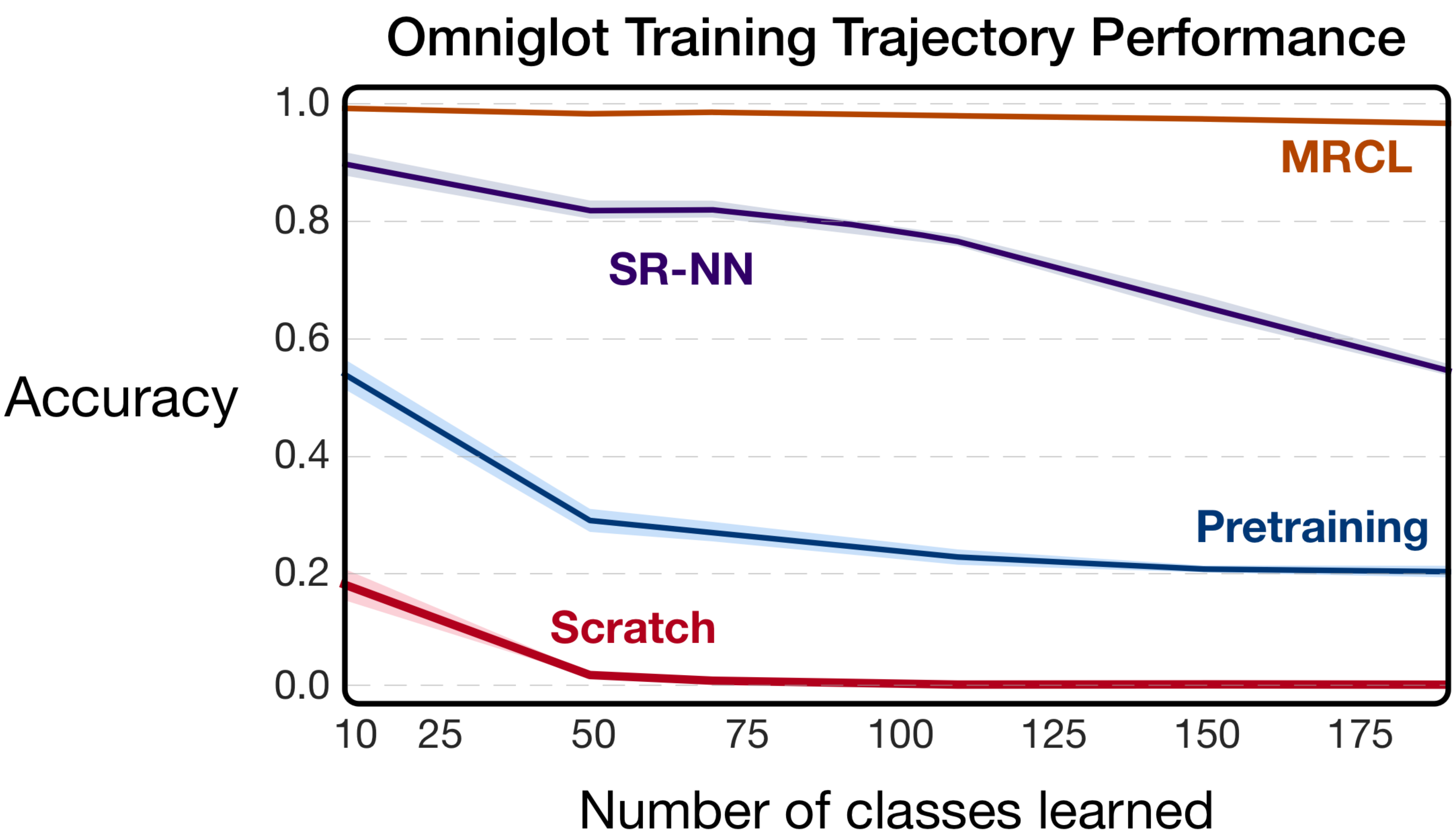
- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Meta-Learning Representations for Continual Learning

Dataset : Omniglot

- ~950 characters from multiple alphabets for representation learning
- ~600 characters from multiple alphabets for continual learning prediction



Analyzing representations

Analyzing representations

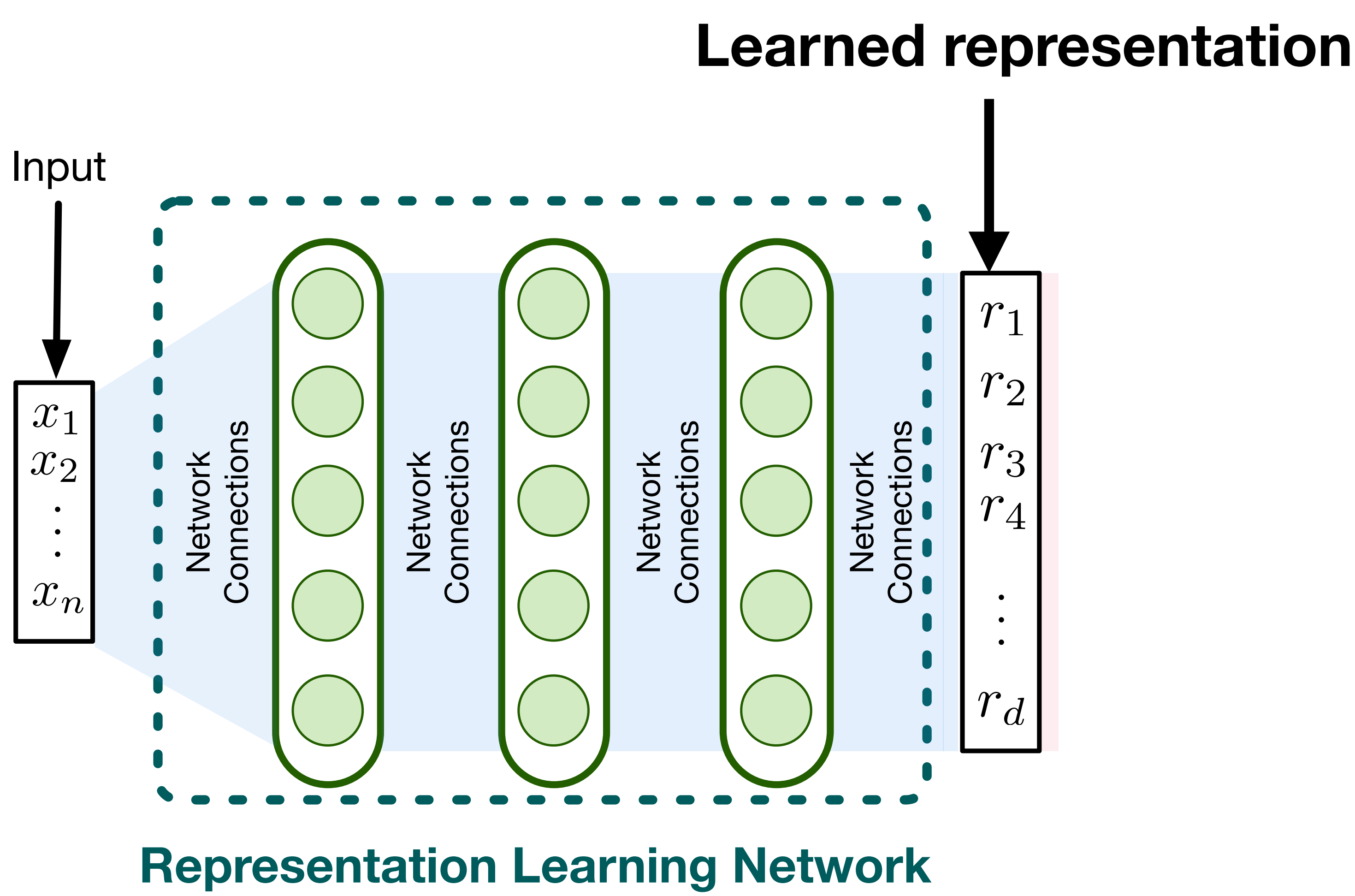
Instance sparsity

Percentage of non-zero entries used to represent an input on average

Analyzing representations

Instance sparsity

Percentage of non-zero entries used to represent an input on average

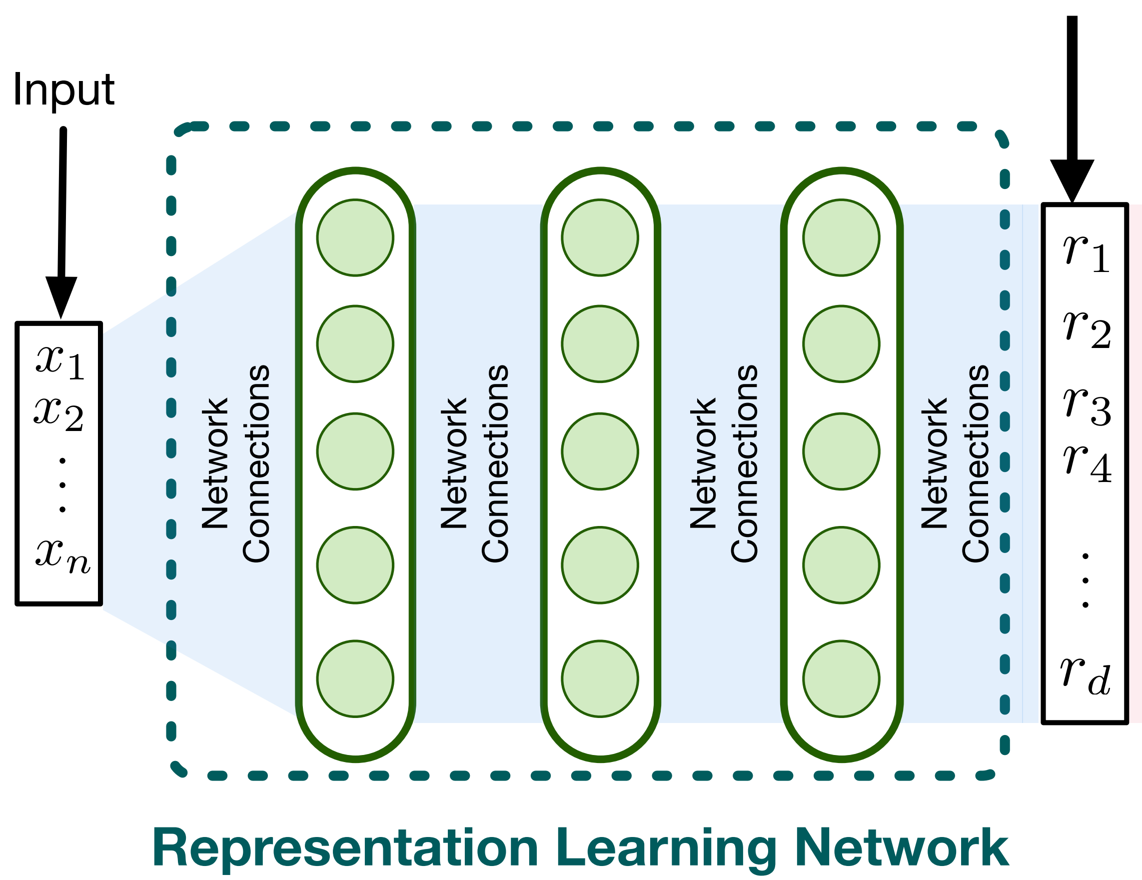


Analyzing representations

Instance sparsity

Percentage of non-zero entries used to represent an input on average

Learned representation

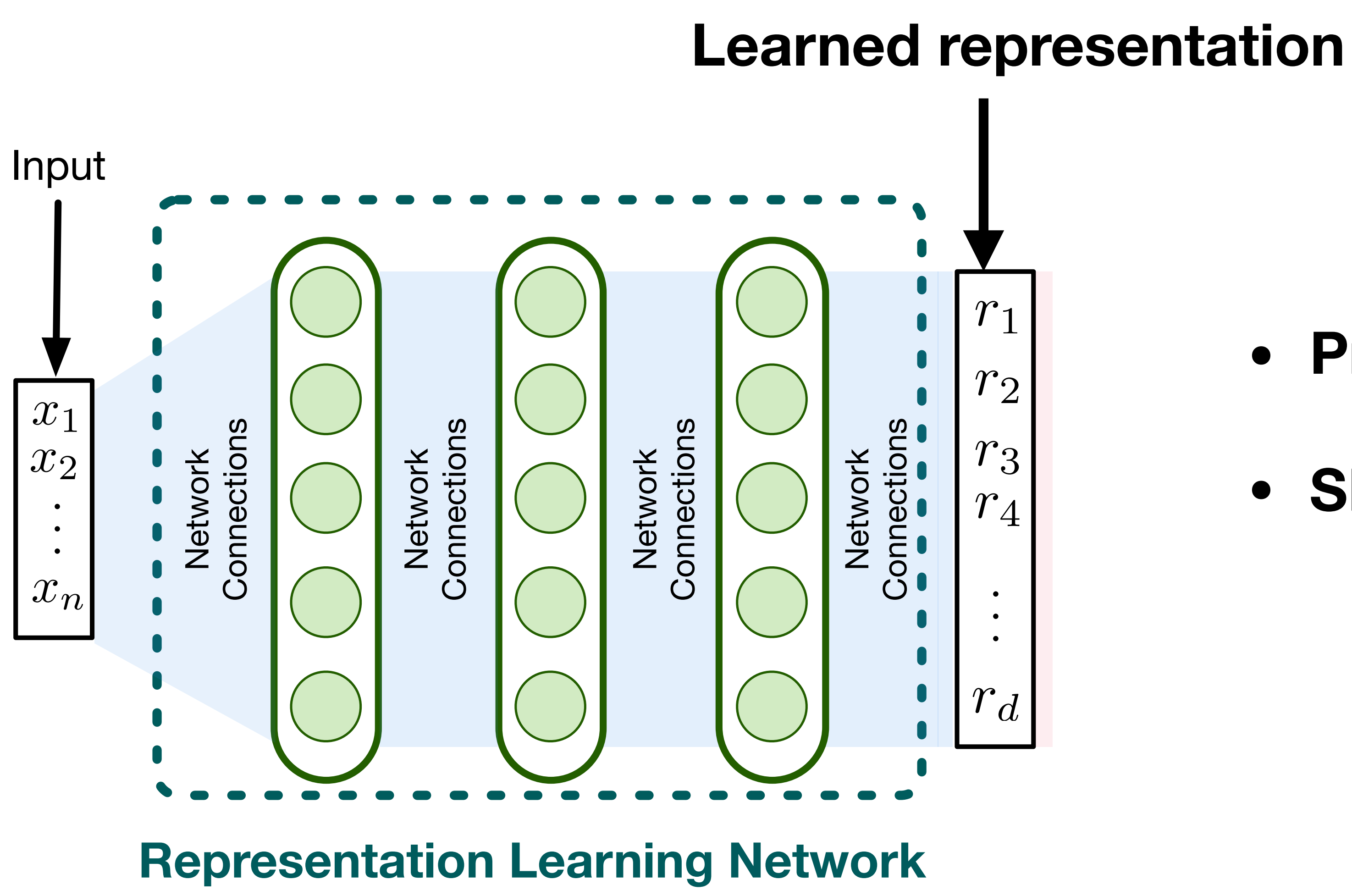


- **Pretraining:** 38%

Analyzing representations

Instance sparsity

Percentage of non-zero entries used to represent an input on average

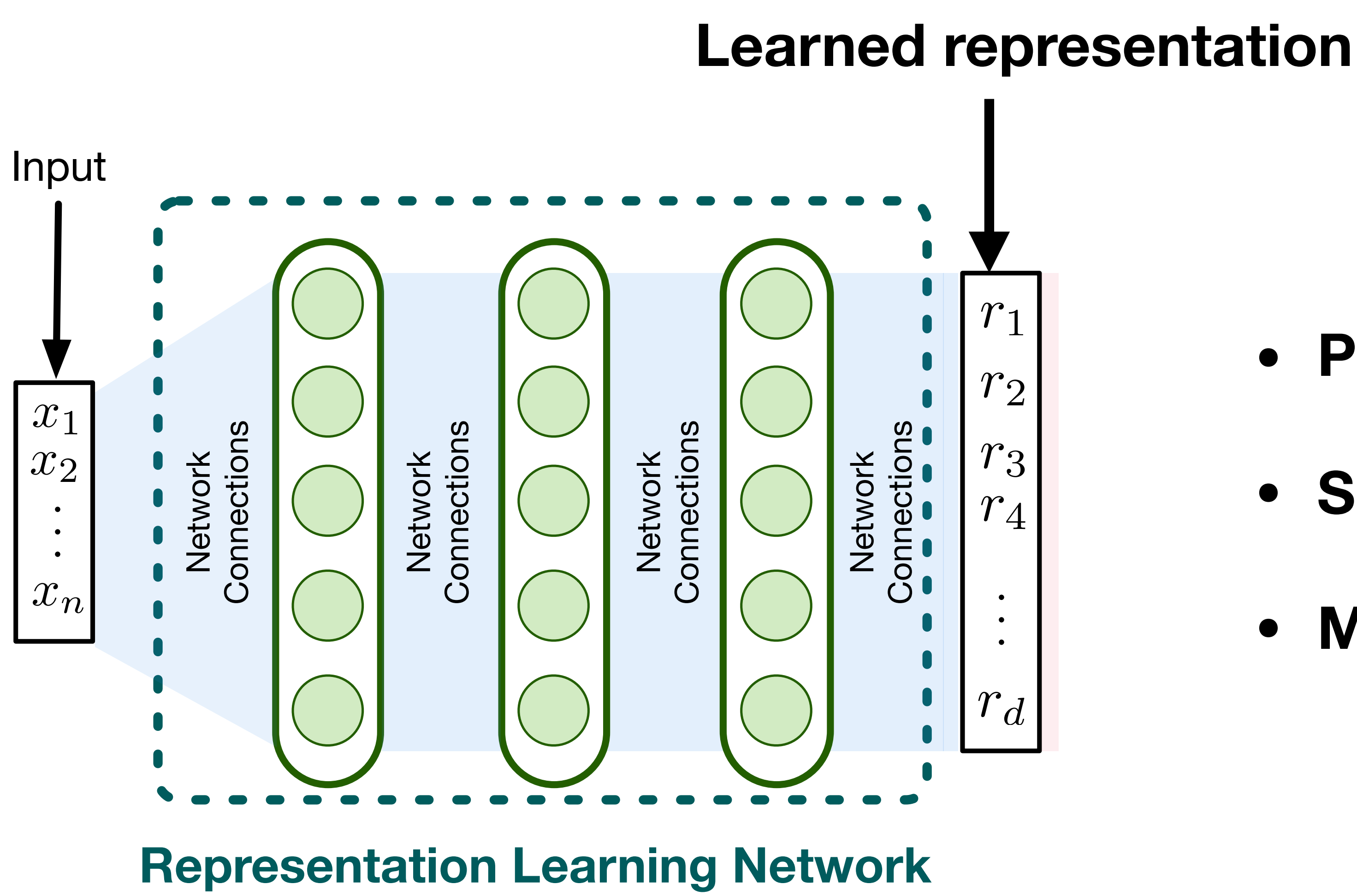


- **Pretraining:** 38%
- **SR-NN**: 15%

Analyzing representations

Instance sparsity

Percentage of non-zero entries used to represent an input on average

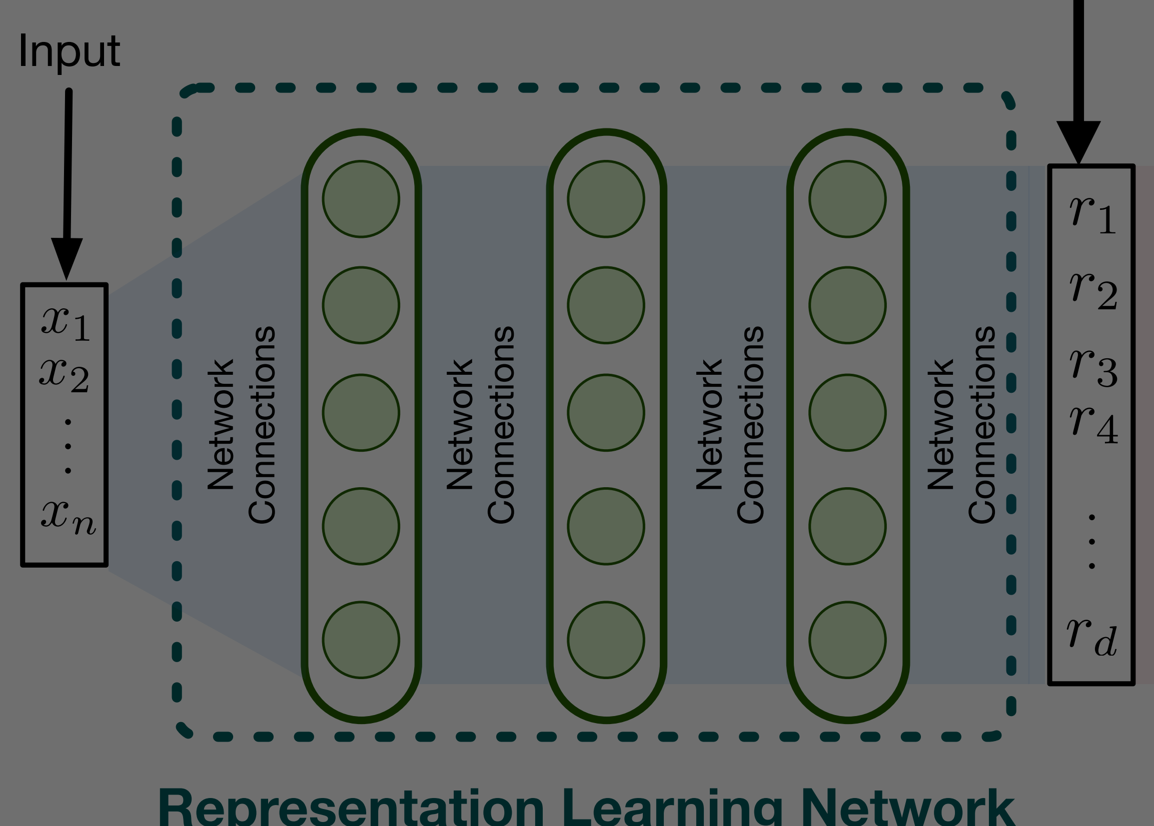


- **Pretraining:** 38%
- **SR-NN**: 15%
- **MRCL:** 3.8%

Analyzing representations

Instance sparsity

Per age



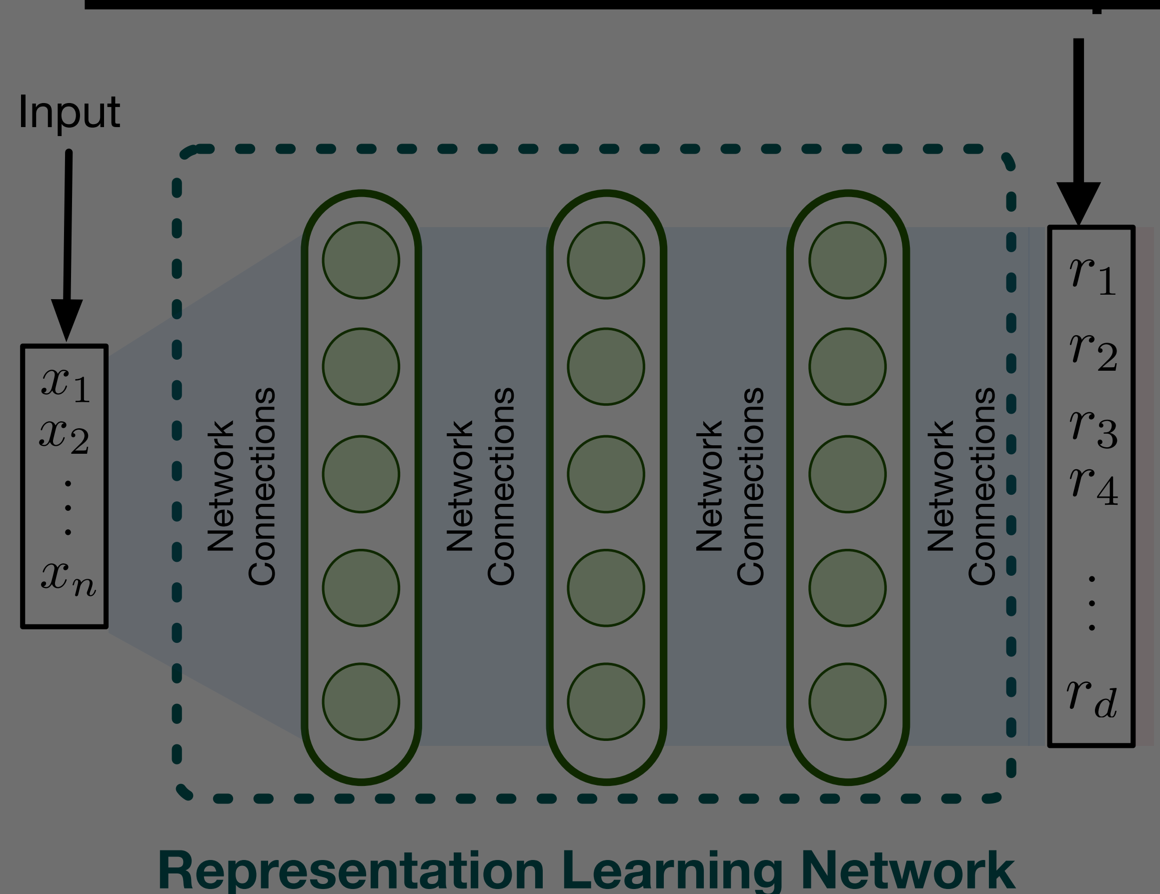
- **Pretraining:** 38%
- **SR-NN**: 15%
- **MRCL:** 3.8%

Analyzing representations

Instance sparsity

Per[...]age

We can train SR-NN with different hyper-parameter to achieve same degree of sparsity



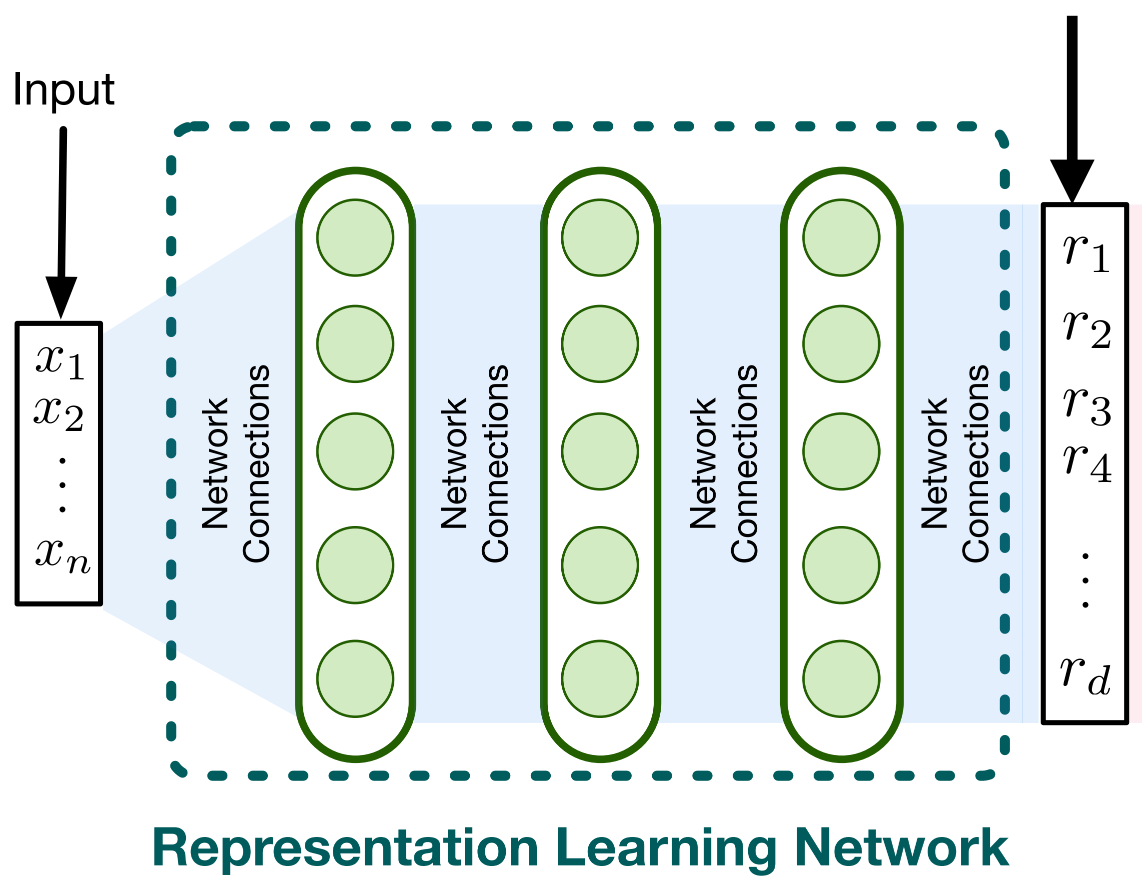
- **Pretraining:** 38%
- **SR-NN**: 15%
- **MRCL:** 3.8%

Analyzing representations

Instance sparsity

Percentage of non-zero entries used to represent an input on average

Learned representation

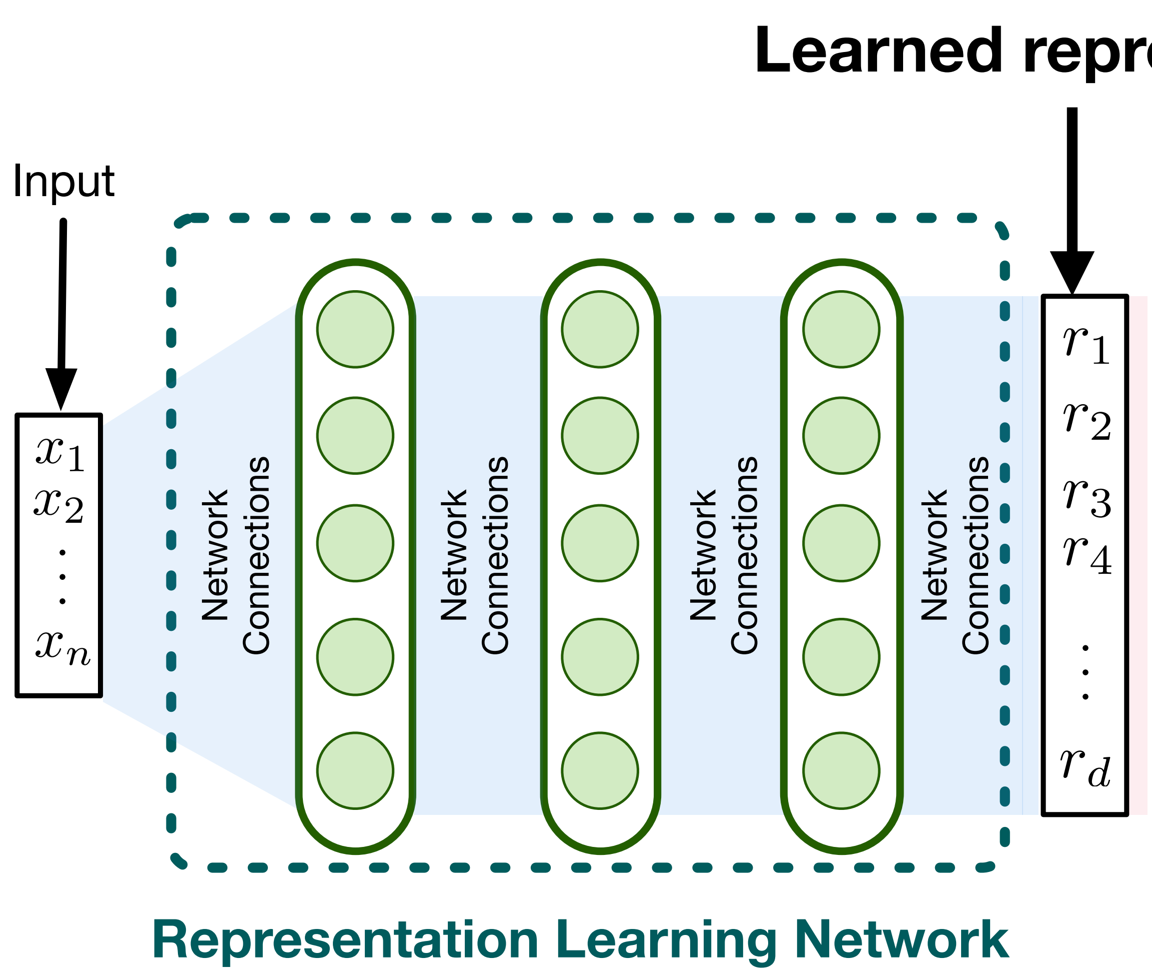


- **Pretraining:** 38%
- **SR-NN**: 15%
- **MRCL:** 3.8%

Analyzing representations

Instance sparsity

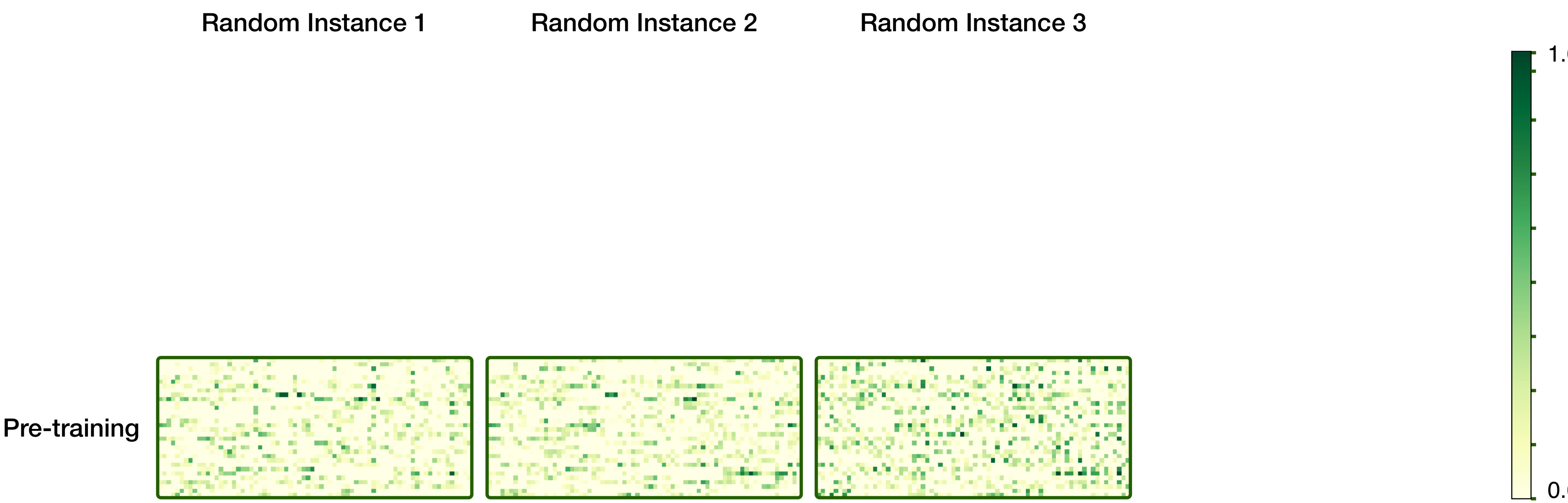
Percentage of non-zero entries used to represent an input on average



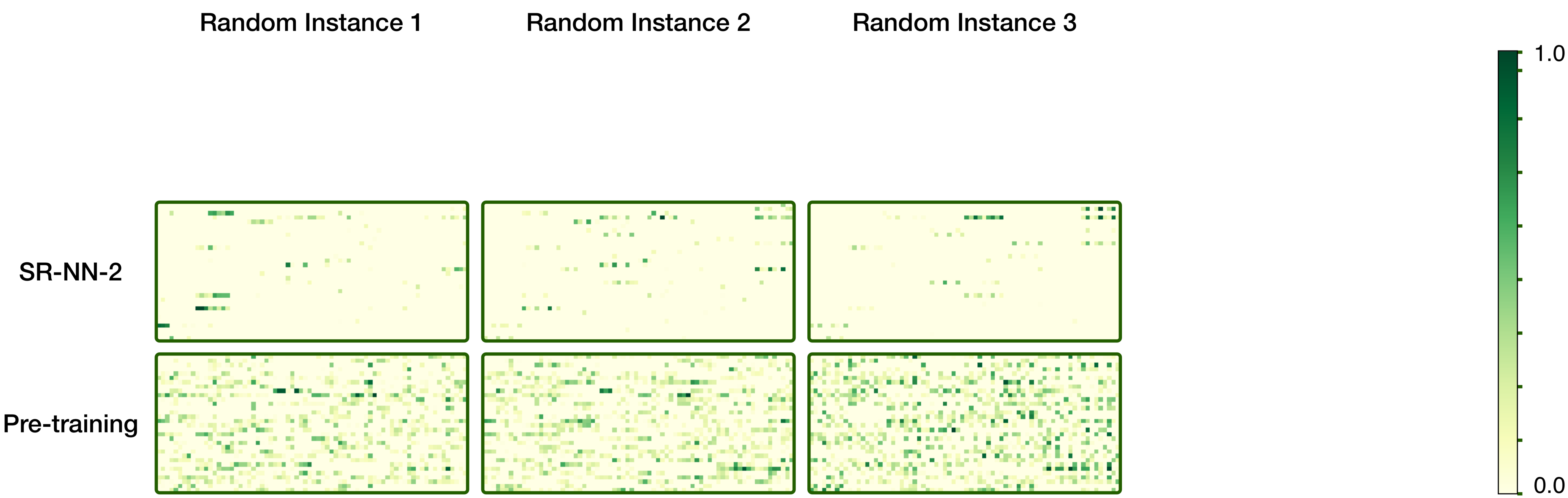
- **Pretraining:** 38%
- **SR-NN**: 15%
- **MRCL:** 3.8%
- **SR-NN-2**: 4.9%

Visualizing representations

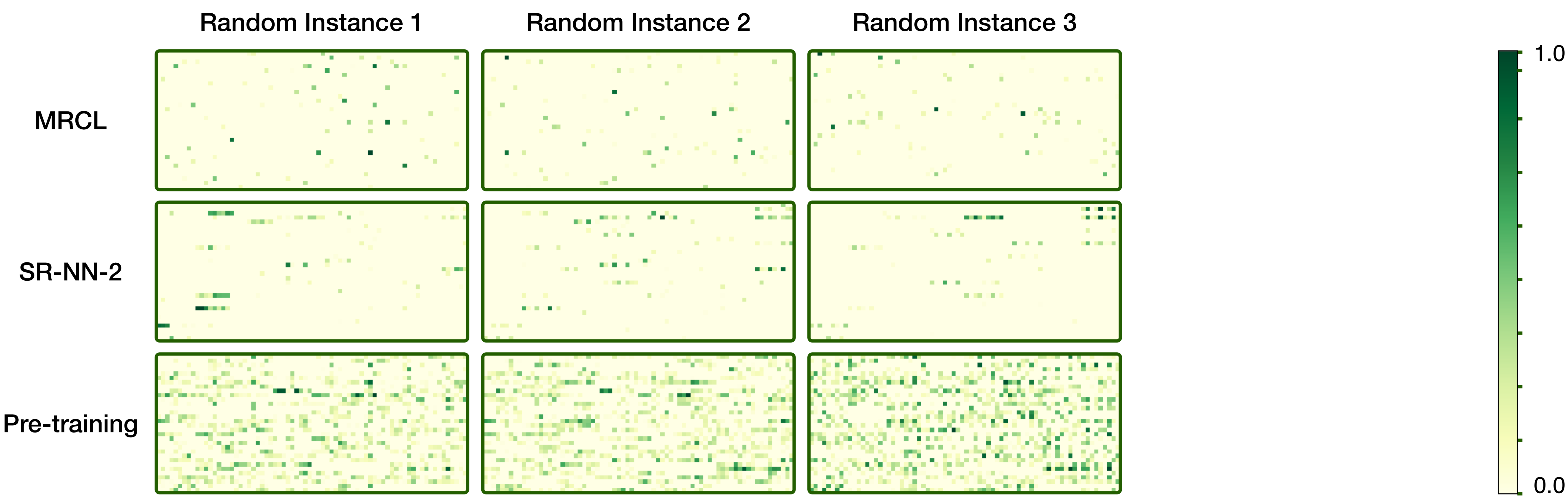
Visualizing representations



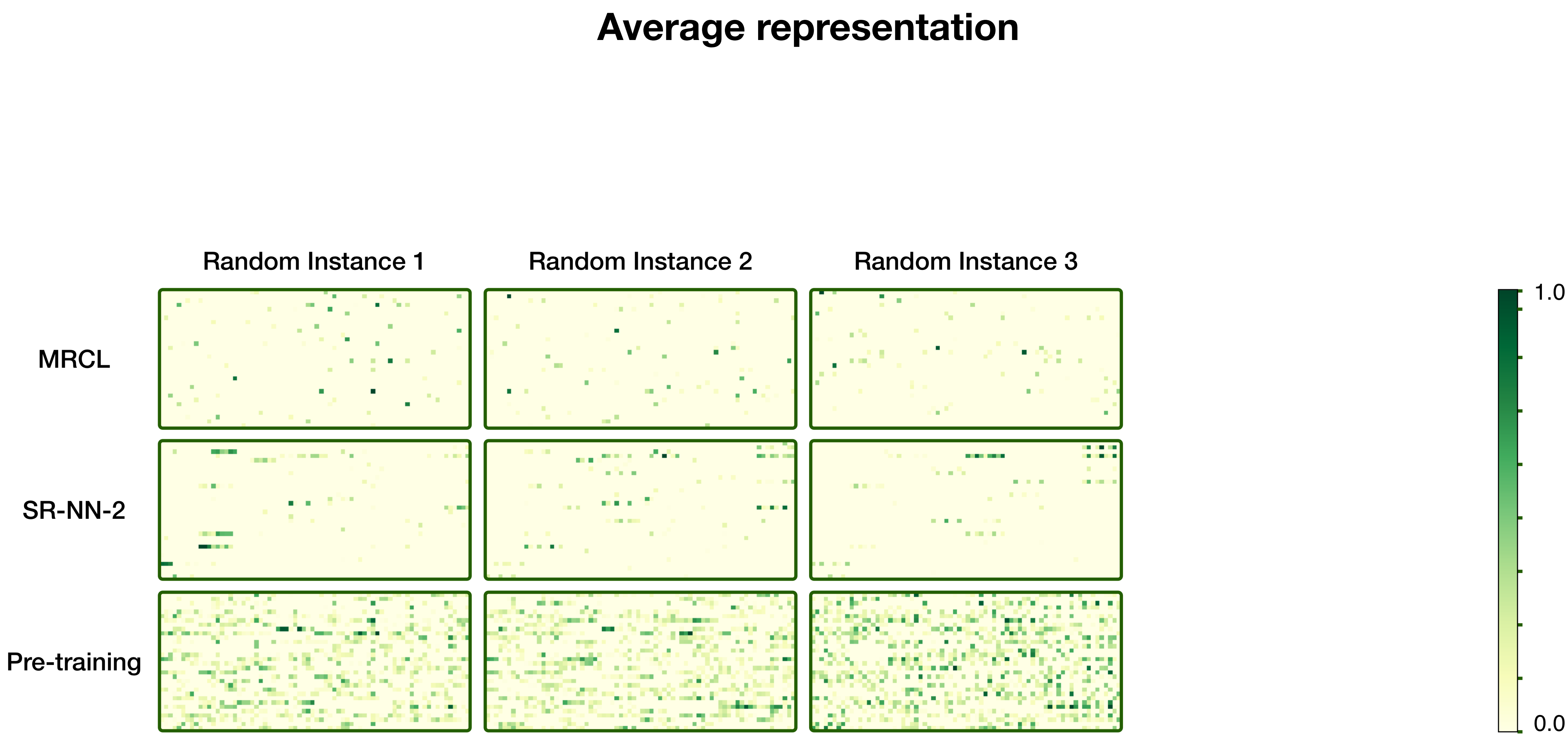
Visualizing representations



Visualizing representations



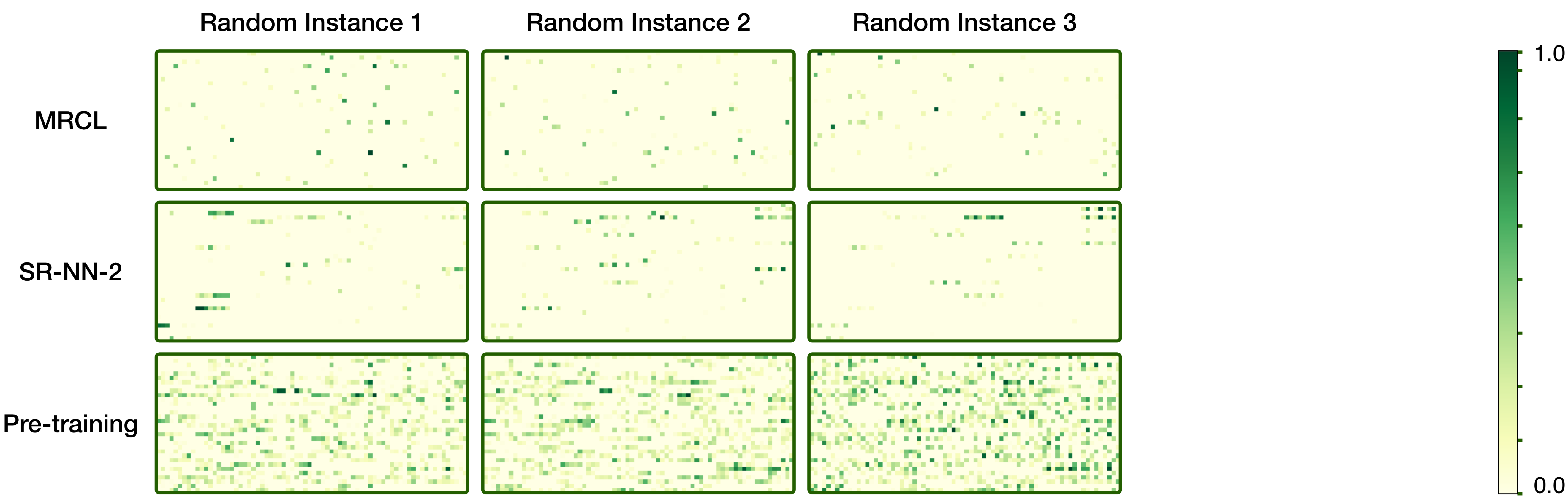
Visualizing representations



Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

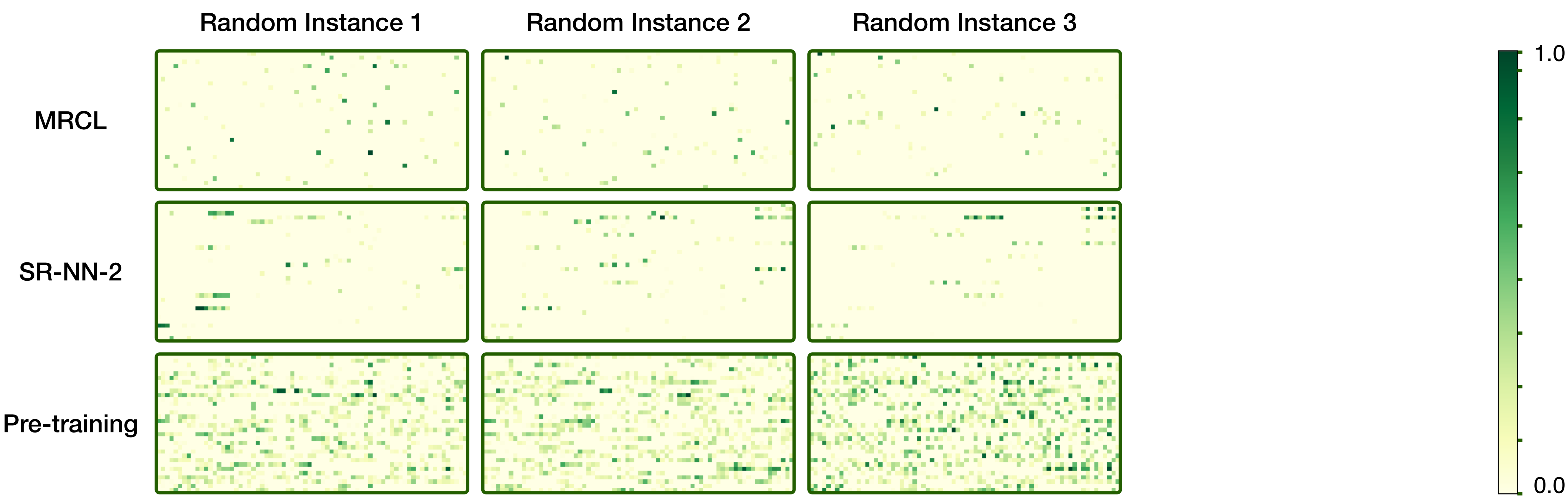


Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

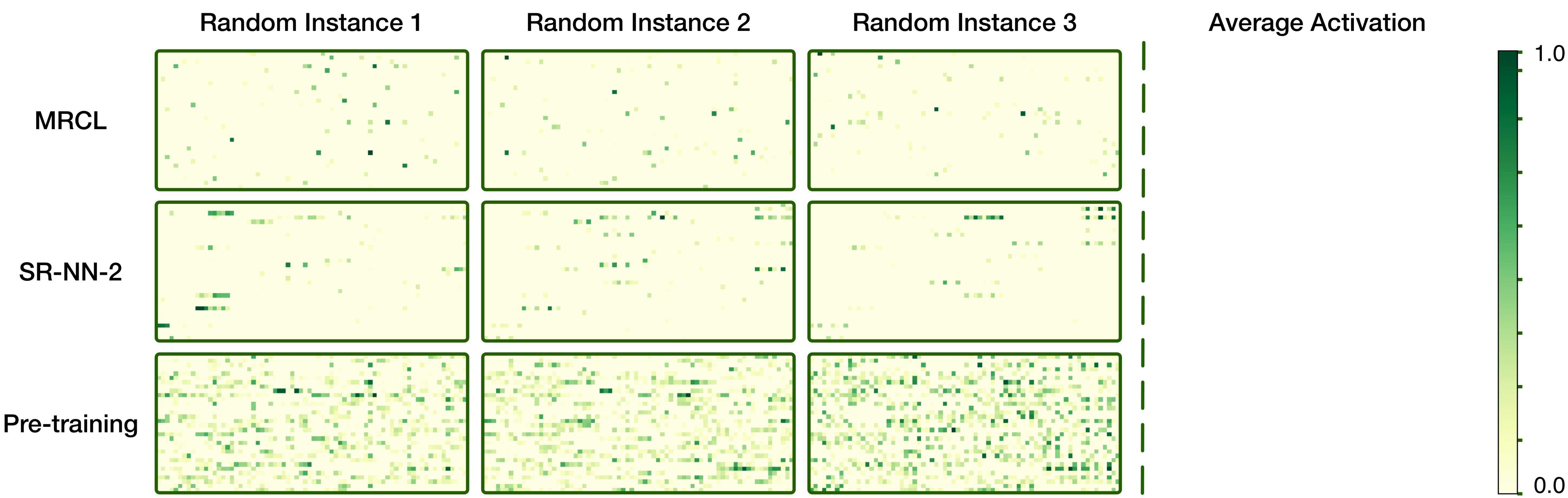


Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

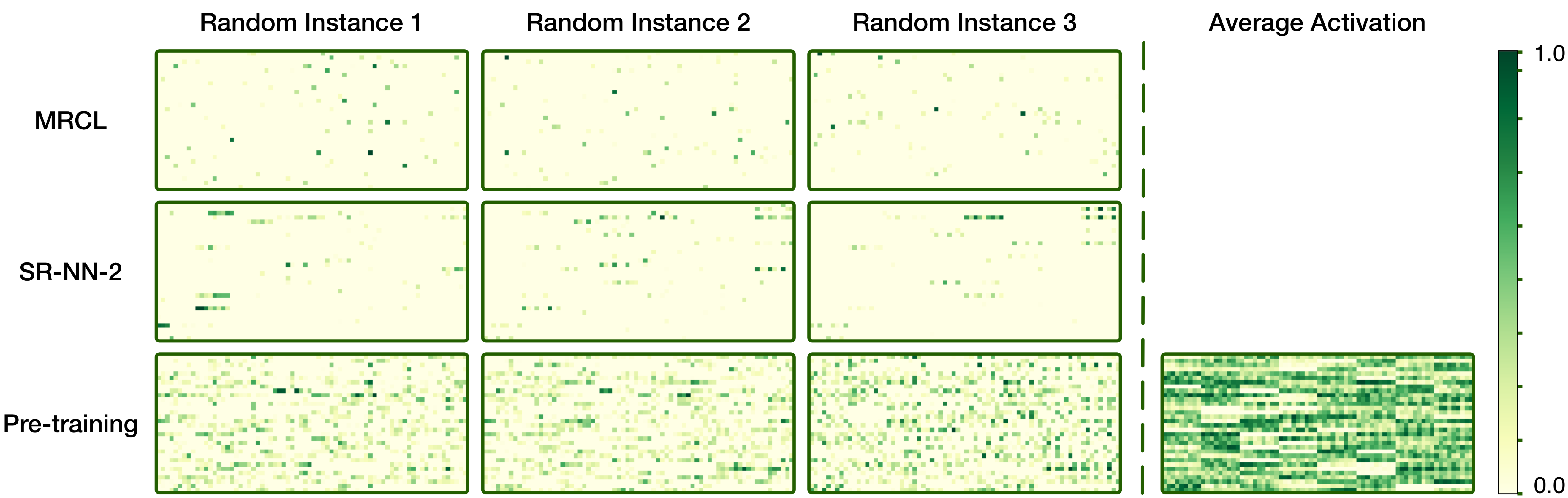


Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

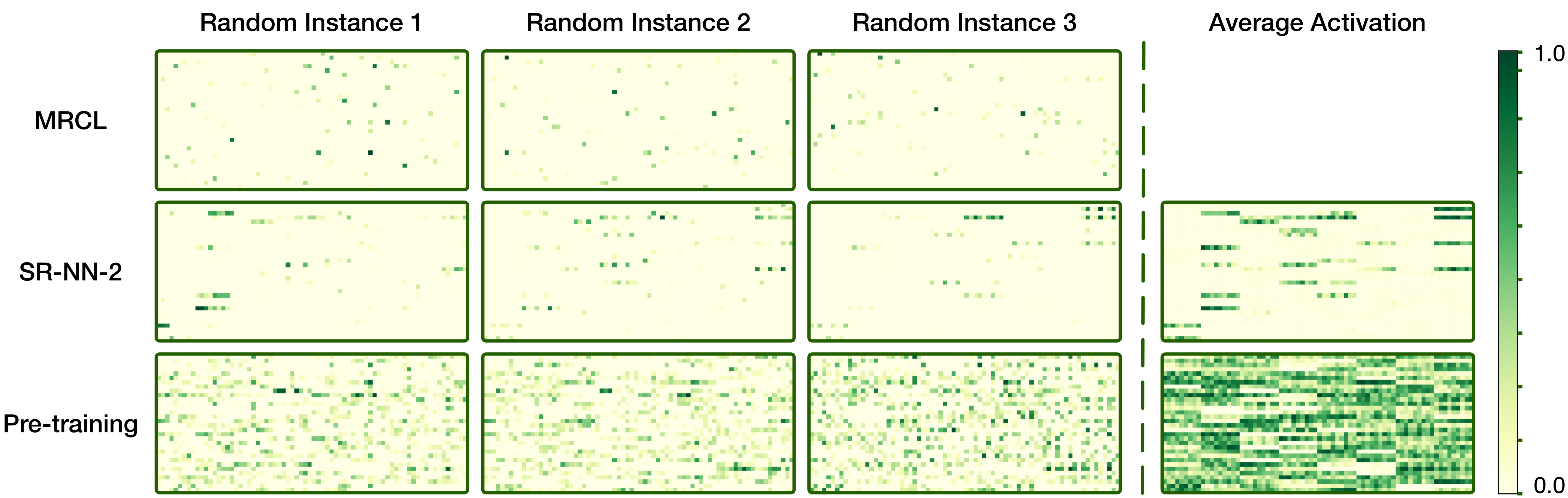


Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

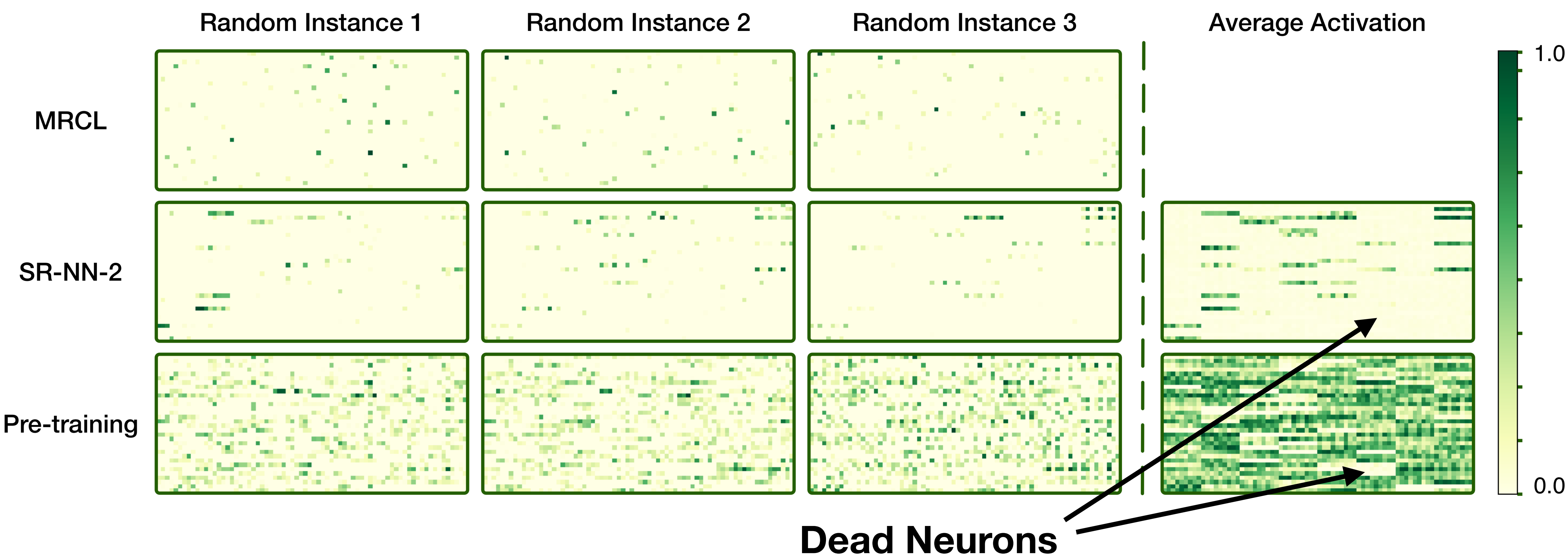


Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector



Visualizing representations

Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

Visualizing representations

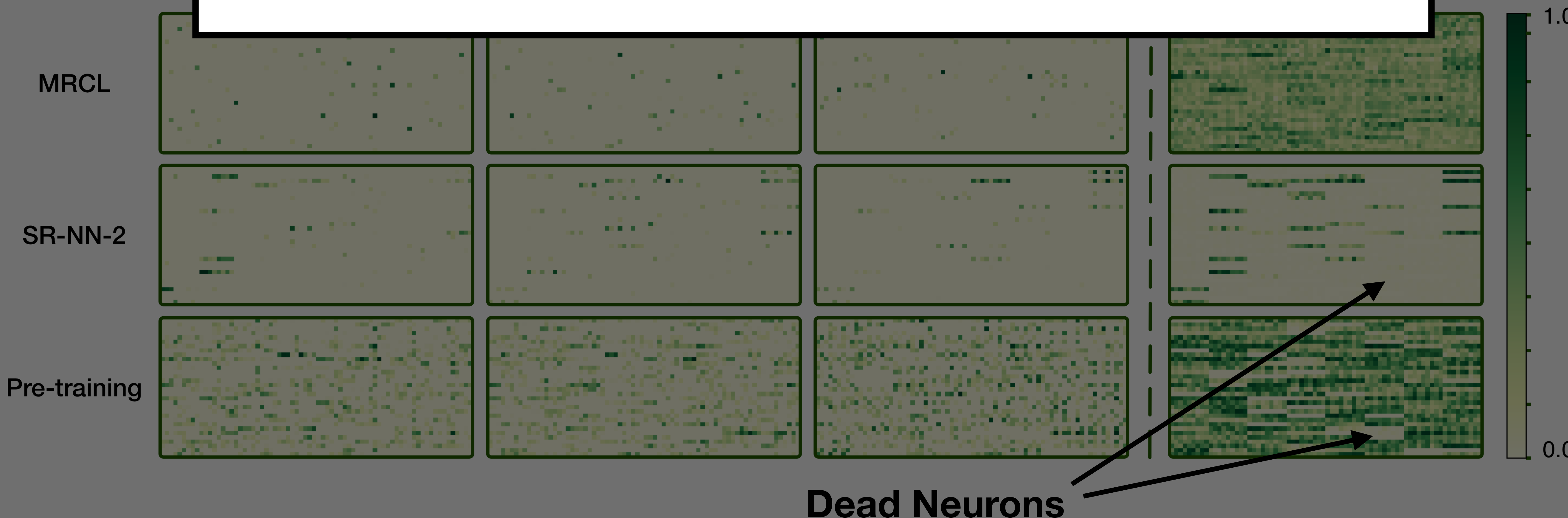
Average representation

Compute representation of the complete representation learning dataset

Compute the average representation vector

Possible explanation

MRCL does better because it is learning the right kind of sparsity!



Is catastrophic interference solved?

Is catastrophic interference solved?

- No! We assumed we have access to a dataset for learning representations

Is catastrophic interference solved?

- No! We assumed we have access to a dataset for learning representations

- Representation learning is not online

Is catastrophic interference solved?

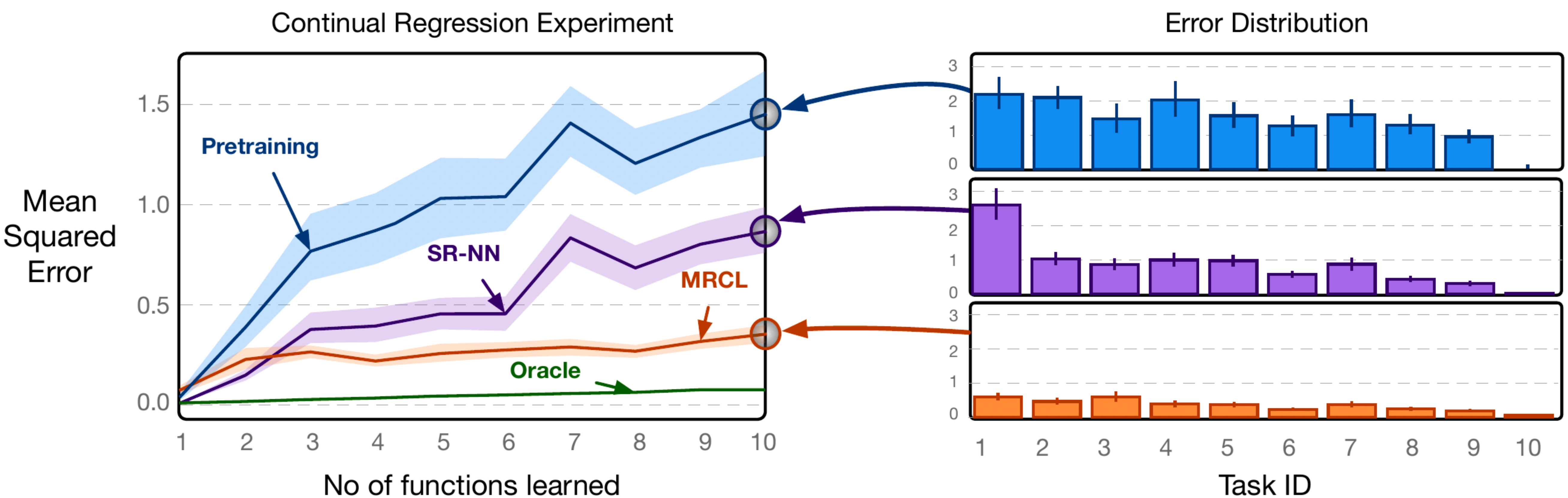
- No! We assumed we have access to a dataset for learning representations
- Representation learning is not online
- However, positive preliminary results for online representation learning

Questions?

Paper: https://arxiv.org/pdf/1905.12588.pdf

Code: https://github.com/khurramjaved96/mrcl

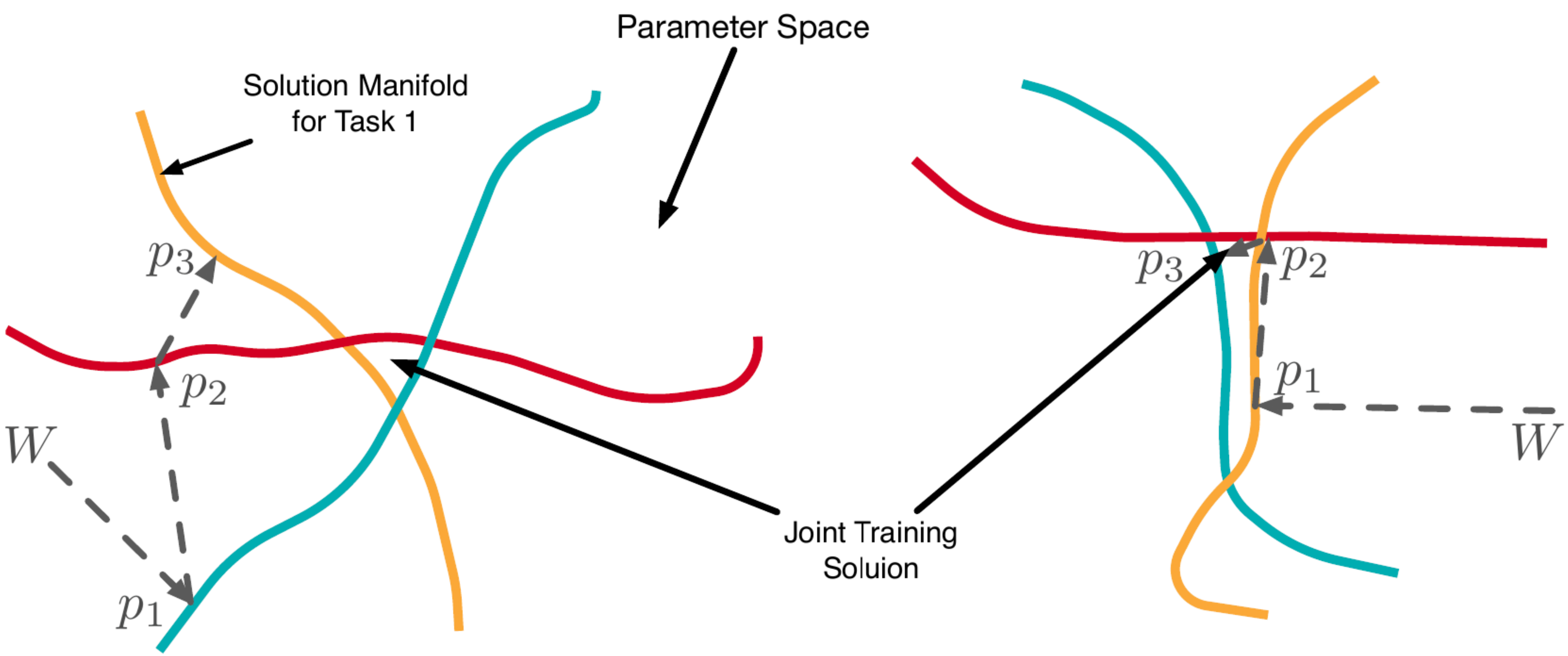
Continual Regression Tasks



Combining with existing methods

	Split-Omniglot					
	One class per task, 50 tasks			Five classes per task, 20 tasks		
Method	Standard	MRCL	Pretraining	Standard	MRCL	Pretraining
Online	04.64 ± 2.61	**64.72** ± 2.57	21.16 ± 2.71	01.40 ± 0.43	**55.32** ± 2.25	11.80 ± 1.92
Approx IID	53.95 ± 5.50	**75.12** ± 3.24	54.29 ± 3.48	48.02 ± 5.67	**67.03** ± 2.10	46.02 ± 2.83
ER-Reservoir	52.56 ± 2.12	**68.16** ± 3.12	36.72 ± 3.06	24.32 ± 5.37	**60.92** ± 2.41	37.44 ± 1.67
MER	54.88 ± 4.12	**76.00** ± 2.07	62.76 ± 2.16	29.02 ± 4.01	**62.05** ± 2.19	42.05 ± 3.71
EWC	05.08 ± 2.47	**64.44** ± 3.13	18.72 ± 3.97	02.04 ± 0.35	**56.03** ± 3.20	10.03 ± 1.53

Solution Manifolds



Solution manifolds for an unconstrained representation

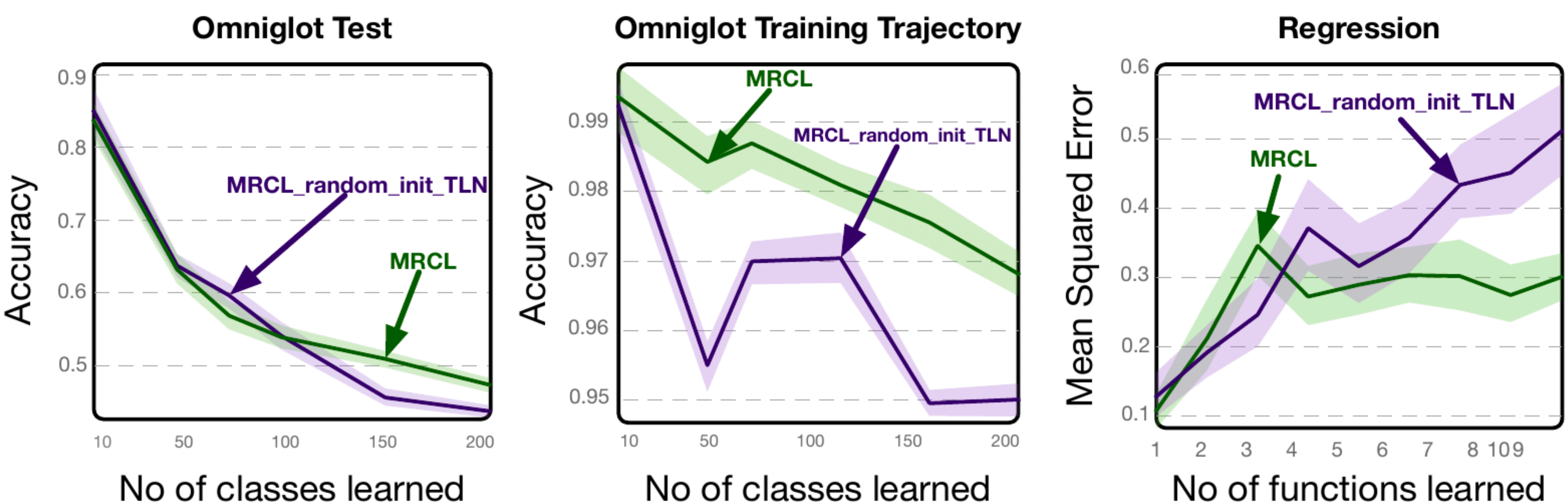
Solution manifolds for an ideal representation for continual learning

Pseudo-code

Algorithm 1: MRCL for optimizing objective in (3)

Require: $\mathcal{D}_{stream} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots;$
Require: $g_W(\phi_\theta(x))$ as a parametrized function;
Require: $\alpha, \beta, \mathcal{L}, n$ as meta learning rate, inner learning rate, loss metric over (X_i, Y_i) and total gradient updates;
Initialize RLN and TLN to θ and W;
for *iterations* $1, 2, 3, \ldots, n$ **do**
    Sample trajectory $(\mathbf{X_{traj}}, \mathbf{Y_{traj}}) = (X_{i+1}, Y_{i+1}) \ldots (X_{i+k}, Y_{i+k}) \sim \mathcal{D}_{stream}$;
    $W_0 = W$;
    for *j in* $1, 2, 3, \ldots, k$ **do**
        $W_j = W_{j-1} - \beta \nabla_{W_{j-1}} (\mathcal{L}(g_{W_{j-1}}(\phi_\theta(X_{i+j})), Y_{i+j});$
    end
    $(\mathbf{X_{rand}}, \mathbf{Y_{rand}}) \sim \mathcal{D}_{stream}$;   Sample a random batch of data
    $(\mathbf{X_{meta}}, \mathbf{Y_{meta}}) = (\mathbf{X_{rand}} + \mathbf{X_{traj}}, \mathbf{Y_{rand}} + \mathbf{Y_{traj}});$
    $W, \theta = (W, \theta) - \alpha \nabla_{W,\theta} \mathcal{L}(g_{W_k}(\phi_\theta(\mathbf{X_{meta}})), \mathbf{Y_{meta}});$
end

PLN Initialization



Generalization Error