Learning Representations for Continual Learning

Martha White
Assistant Professor
University of Alberta
Joint work with Khurram Javed

Goals for the talk

- Motivate the importance of learning an explicit Representation for online updating

- Introduce the Online-Aware Meta-Learning objective

- Show how the representations from this objective

  - improve prediction performance when learning online

  - complement previous strategies for continual learning, like EWC

On arXiv: Meta-Learning Representations for Continual Learning (https://arxiv.org/abs/1905.12588)

Problem Setting

- A Continual Learning Prediction (CLP) problem has an unending stream of samples

$$(X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

- Correlated sequence for inputs X

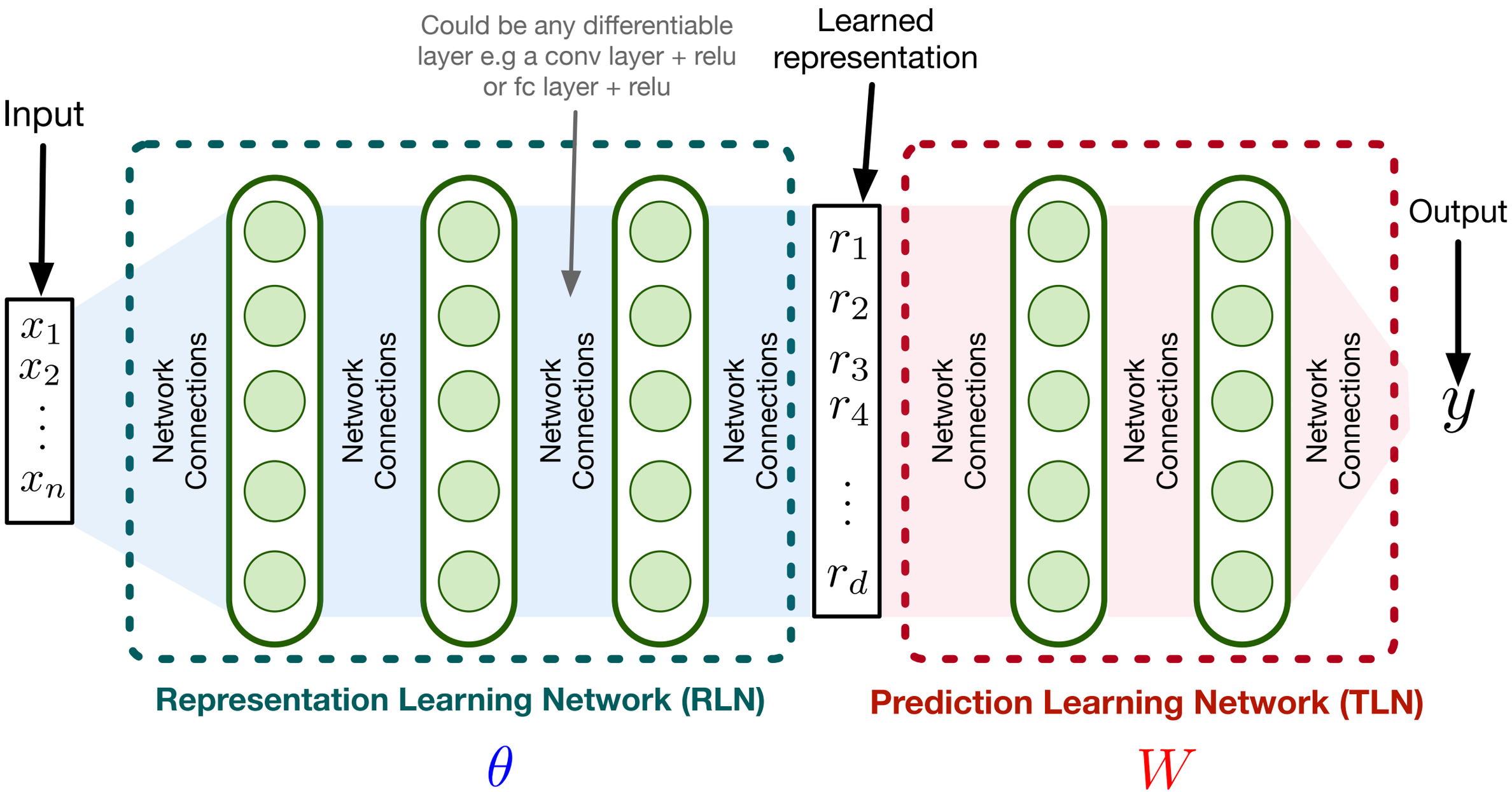
- Each Y dependent only on X: Y_t distributed according to $P(Y \mid X_t)$

CLP is General, it encompasses….

Correlated X
Y_t ~ P(Y | X_t)

- Formulations with task descriptors (X_t, T_t, Y_t)
- Continual learning classification benchmarks
  - observe pairs (X,Y) for each class in order
- Online regression problems
  - even RL by considering Y_t to be a bootstrapped target
- Settings where Y_t depends on the last k observations
  - X_t equal to the last k observations, with overlap between X_t and X_{t-1}

Parameterized function



CLP Objective

Objective is still an empirical risk minimization objective

$$\mathrm{CLP}(\theta, W) \stackrel{\text{def}}{=} \mathbb{E}[\ell(f_{\theta,W}(X), Y)] = \int \left[\int \ell(f_{\theta,W}(x), y) p(y|x) dy \right] \mu(x) dx.$$

$\mu : \mathcal{X} \to [0, \infty)$ marginal distribution over X

Different training regime

- Do not get iid samples, $X_t \sim \mu \text{ and } Y_t \sim P(Y|X_t)$
- The sequence of samples is correlated
- This has a significant impact on training NNs online
  - Updates are not spread across the space
  - The NN begins to specialize to recent samples and forget what it learned before on other parts of the space

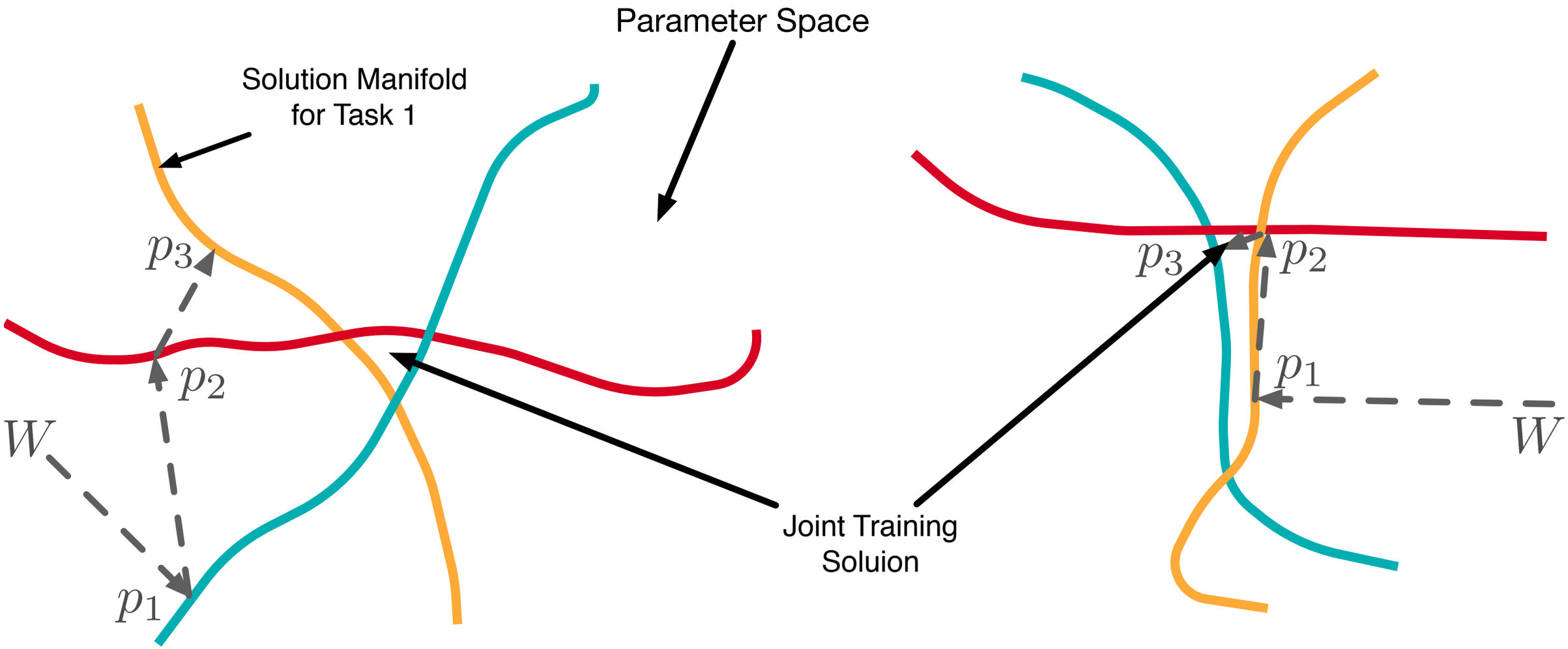
Strategies to Mitigate Interference

- Replay or generate samples for more updates
  - e.g., replay in RL, model-based RL, knowledge distillation
- Modify the online update to retain knowledge
  - e.g., Elastic Weight Consolidation
- Use sparse (or semi-distributed) representations
  - e.g., early work by French in 90s

Strategies to Mitigate Interference

- Replay or generate samples for more updates
  - e.g., replay in RL, model-based RL, knowledge distillation
- Modify the online update to retain knowledge
  - e.g., Elastic Weight Consolidation
- **Use sparse (or semi-distributed) representations**
  - e.g., early work by French in 90s

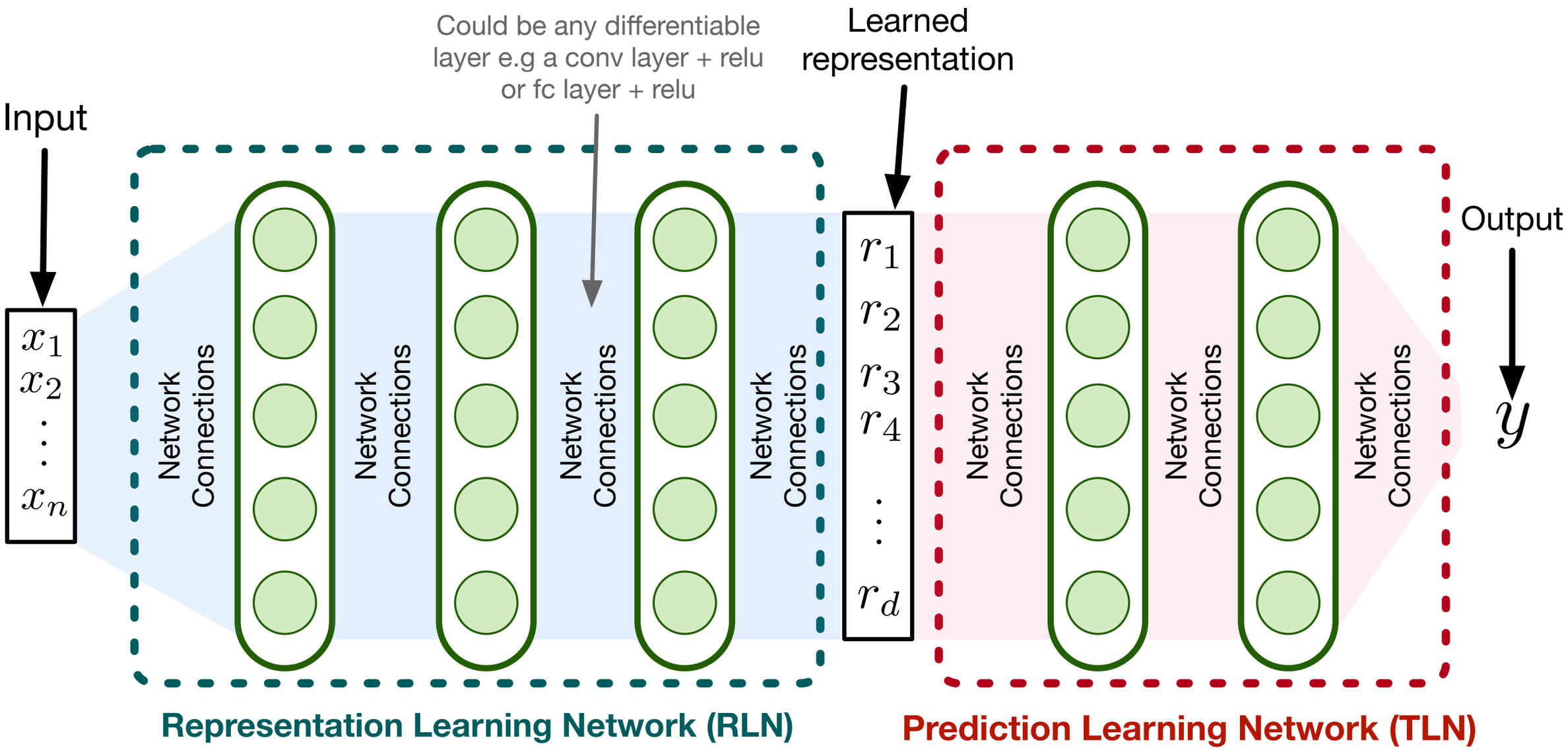
Importance of the Representation for Mitigating Interference



Solution manifolds for an unconstrained representation

Solution manifolds for an ideal representation for continual learning

Separately considering the Representation



Claim: It is useful to delineate the representation for continual learning
(and learn it differently)

Hypothesized Advantages

- Learning this representation could be a slower, background process

- Slowly changing representation would suffer much less from interference

    - The Prediction Learning Network can still learning quickly online

    - Gain stability without losing reactivity

- Can consider alternative objectives and learning approaches for the Representation Learning Network

Online-aware Meta-Learning Objective

$$\mathrm{OML}(\theta, W) \stackrel{\text{def}}{=} \int \mathbb{E}\left[\mathrm{CLP}\Big(U(\theta, W, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})\Big) | X_t = x\right] \mu(x) dx.$$

where $U(\theta, W, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})$
is the online update for k steps starting from θ, W
$[\theta, W_{t+k}] = U(\theta, W_t, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})$

Recall:

$$\mathrm{CLP}(\theta, W) \stackrel{\text{def}}{=} \mathbb{E}[\ell(f_{\theta,W}(X), Y)] = \int \left[\int \ell(f_{\theta,W}(x), y) p(y|x) dy\right] \mu(x) dx.$$

Online-aware Meta-Learning Objective

$$\mathrm{OML}(\theta, W) \stackrel{\text{def}}{=} \int \mathbb{E}\left[\mathrm{CLP}\Big(U(\theta, W, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})\Big) | X_t = x\right] \mu(x) dx.$$

where $U(\theta, W, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})$
is the online update for k steps starting from θ, W
$[\theta, W_{t+k}] = U(\theta, W_t, \{(X_{t+i}, Y_{t+i})\}_{i=1}^{k})$

Key Point: U maintains correlated ordering that the online learner will see

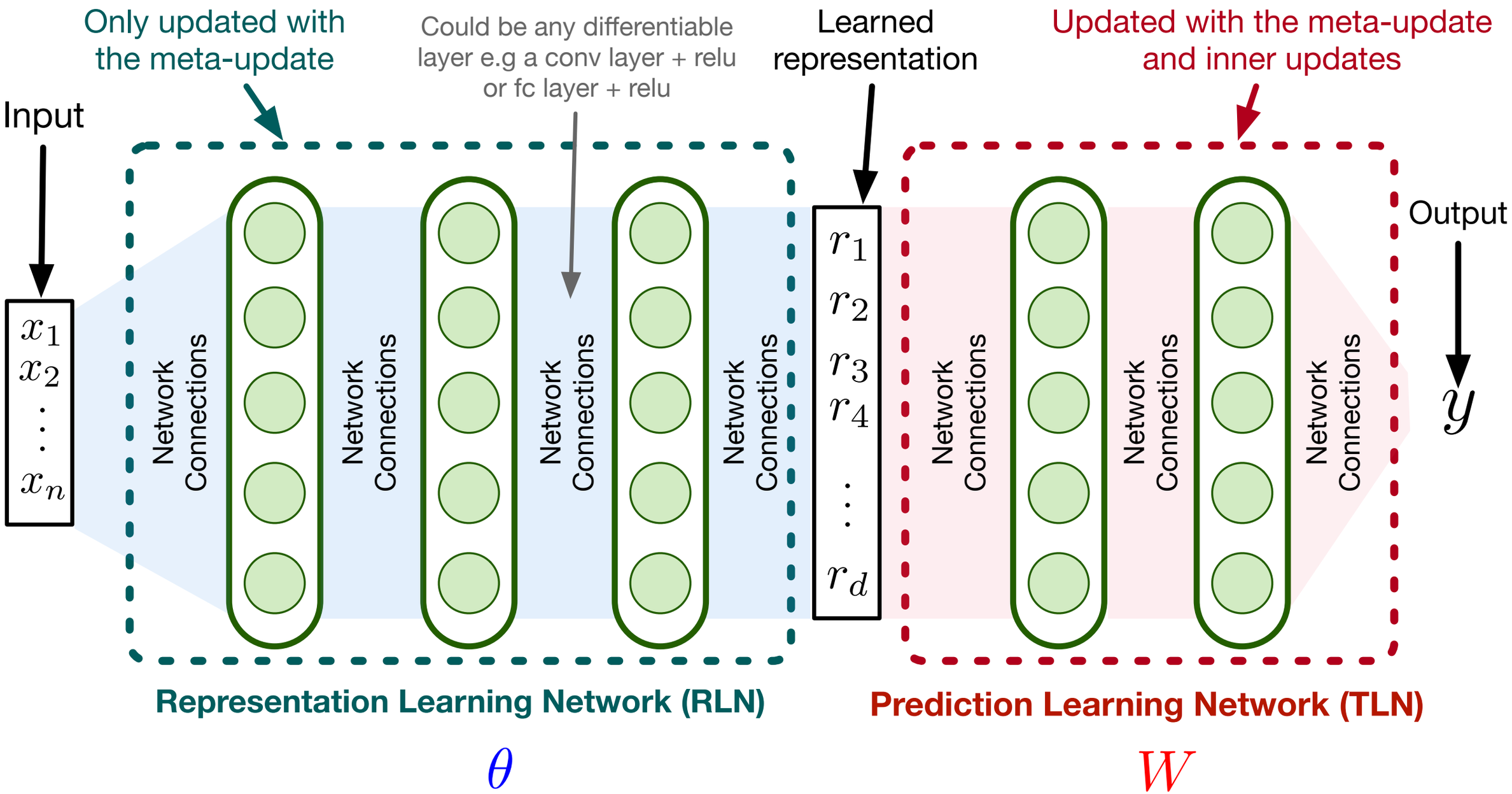
Recall:

$$\mathrm{CLP}(\theta, W) \stackrel{\text{def}}{=} \mathbb{E}[\ell(f_{\theta,W}(X), Y)] = \int \left[\int \ell(f_{\theta,W}(x), y) p(y|x) dy\right] \mu(x) dx.$$

How is this different from MAML? Similar idea but…

- The goal is to learn a **representation** for continual learning, rather than **few-shot learning** (fine-tuning the network)
  - Our goal is to learn fast and minimize interference across all data
    - Sample a sequence for the online update, and optimize error for all other data
- There are no **explicit tasks**, just **correlated data**
- Instead of updating the **whole network online**, only the **prediction layers are updated online**
  - Can be seen as setting stepsizes for representation params to zero

Separately considering the Representation



Q1: Can we learn representations that are effective for online updating?

- Pre-train the representation on a pre-training set
  - batch of data can be used however, including iid training
- Test performance when learning online on new data, with a fixed pre-trained representation
- Two settings:
  - One prediction, Y | x generated from 10 different functions
  - An increasing number of predictions (increasing classes)

Dataset 2: Split-Omniglot

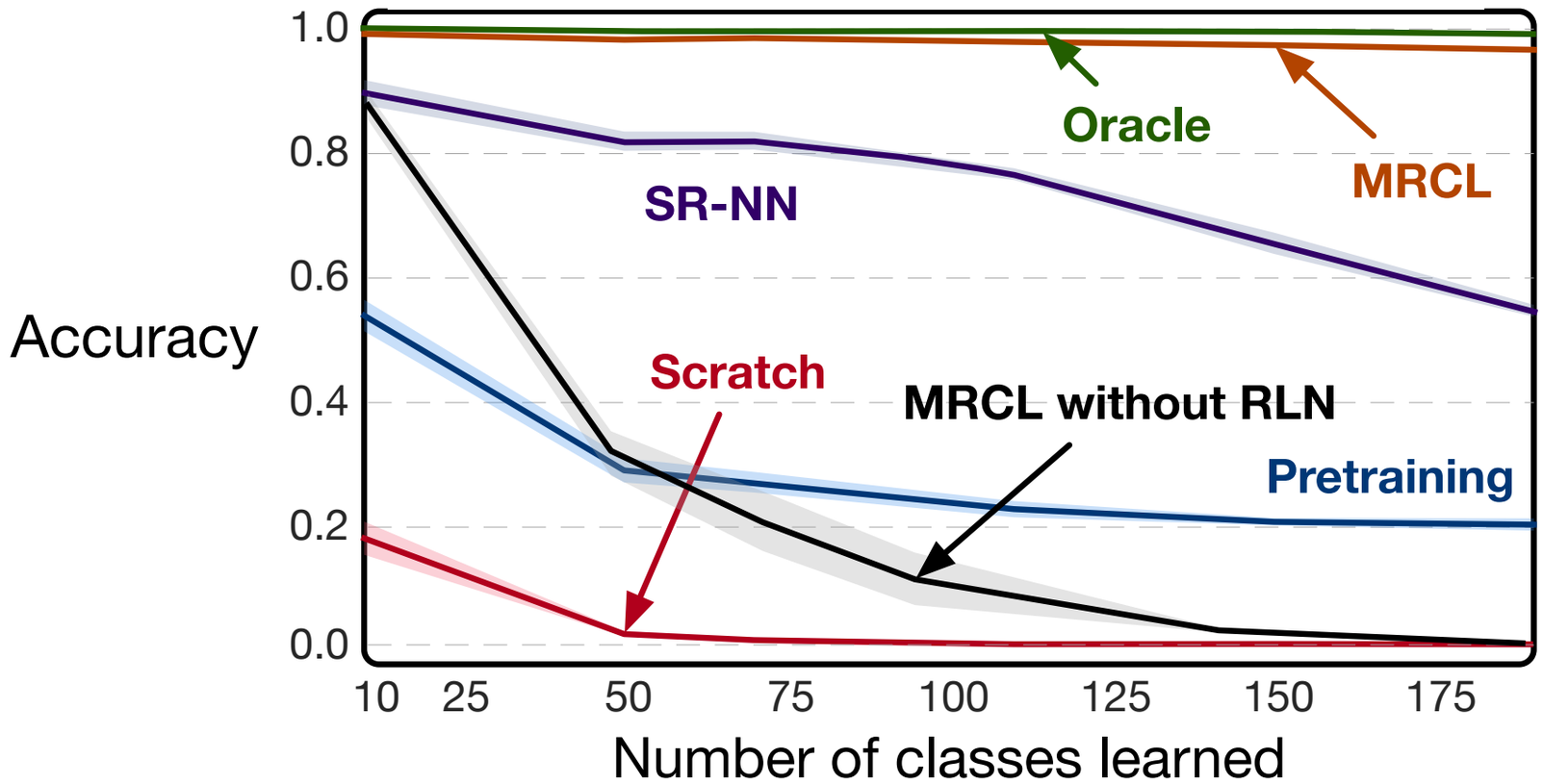
- Omniglot has 1623 characters, each with 20 hand-written images
- Pre-training data: the first 963 classes
- The remaining classes are Evaluation data (for online learning): all of one class is seen before going to the next
- Inputs are the image (we do not use task IDs)
- Chosen because (a) a hard problem with many predictions, (b) a given split between pre-training and evaluation classes

Algorithms

- 8 layers in NN, with 6 for RLN and 2 for TLN
- **Scratch**: learns online from a random initialization, no pre-training
- **Pretraining**: iid training on pre-training set
- **SR-NN**: a neural network with sparse activations, trained using the Set-KL method
- **MRCL**: our algorithm, Meta-learned Reps for Continual Learning
- **MRCL without RLN**: define online update on whole network

Continual Classification Results

Up is good

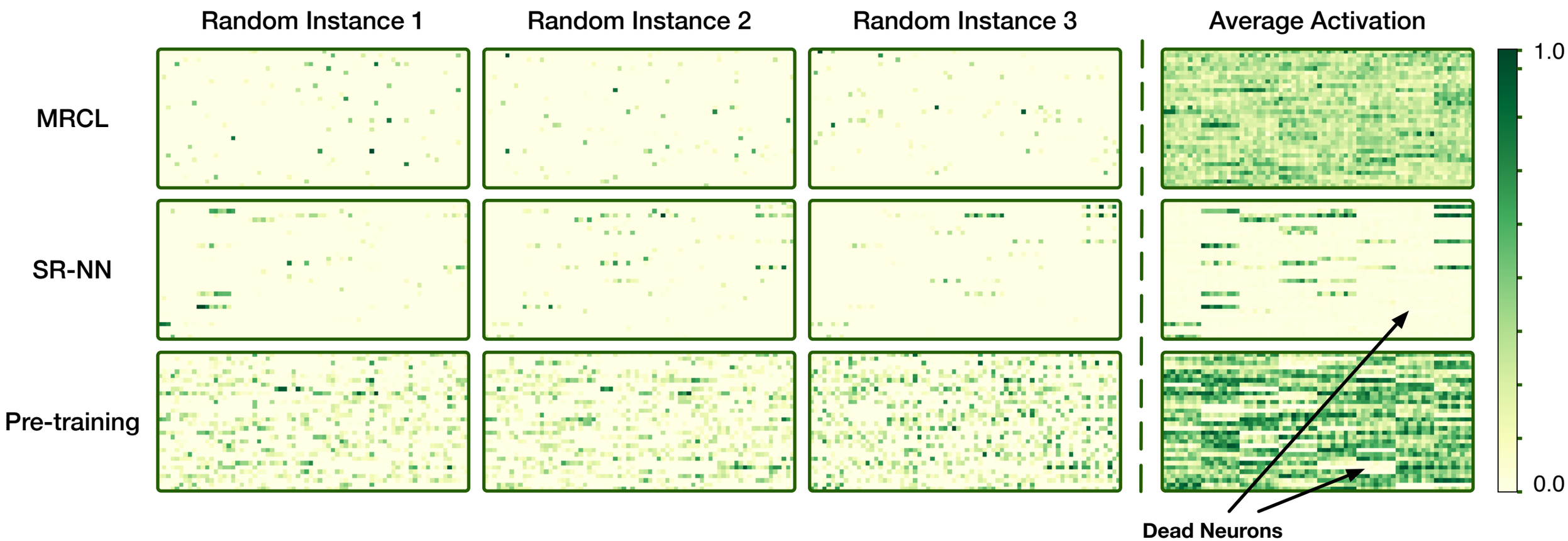


MRCL is our method

1. Separating out the RLN is very important
2. Just fixing the representation does not prevent interference
3. Sparse Activation NN helps quite a bit, but not as much as MRCL

***Oracle: learned using IID training on the trajectory with multiple epochs**

Sparsity naturally emerges when using OML



MRCL has 4% activation and no dead neurons
Pre-training has 40% activation 3% dead neurons
Best SR-NN has 15% activation with 1% dead neurons
SR-NN trained to be more sparse, with 4% activation, has 14% dead neurons

*Reshaped the 2304 length representation vectors into 32x72 for visualization

Q2: Do the learned representations complement other strategies for forgetting?

	Split-Omniglot		
Method	Standard	MRCL	Pretraining
Online	04.64 ± 2.61	**64.72** ± 2.57	21.16 ± 2.71
Approx IID	53.95 ± 5.50	**75.12** ± 3.24	54.29 ± 3.48
ER-Reservoir	52.56 ± 2.12	**68.16** ± 3.12	36.72 ± 3.06
MER	54.88 ± 4.12	**76.00** ± 2.07	62.76 ± 2.16
EWC	05.08 ± 2.47	**64.44** ± 3.13	18.72 ± 3.97

1. MRCL improves all the algorithms.
2. The results are not due to just fixing the representation.
3. MRCL with a basic Online updating strategy is already competitive.
4. MRCL improves even approximate IID sampling (suggesting it is not only mitigating interference but making learning faster on new data).

Key take-away

We should be explicitly learning representations that are well-suited for online updating

Open Questions

- **How** can we learn these representations online?

- **When** can we expect to learn representations amenable to online updating?
    - e.g., how related does pre-training data have to be to future data?

- What are the **disadvantages** of delineating a representation and training it differently?

- Can we learn representations **tailored** to more complex continual updates?

Open Questions

- **How** can we learn these representations online?

- **When** can we expect to learn representations amenable to online updating?
  - e.g., how related does pre-training data have to be to future data?

- What are the **disadvantages** of delineating a representation and training it differently?

- Can we learn representations **tailored** to more complex continual updates?

Thank you!