Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference

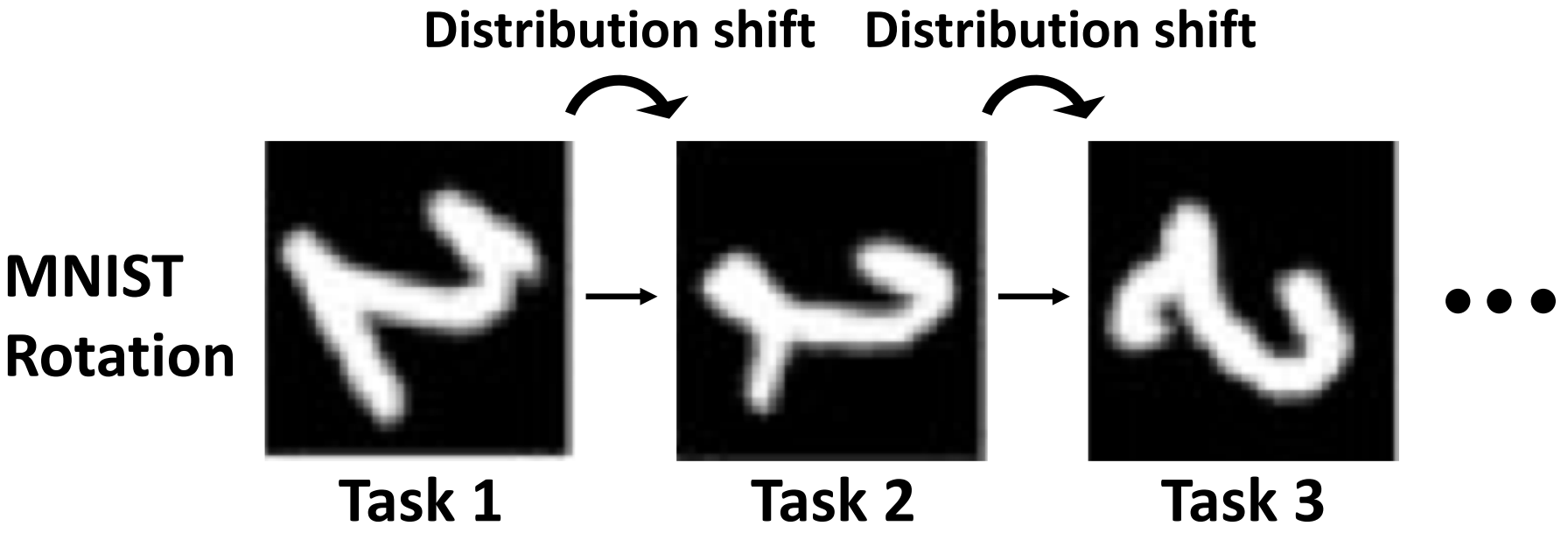
PRESENTOR: BYUNGSOO JEON

LLL READING GROUP @ CMU

* THIS PAPER WAS PUBLISHED IN ICLR19

Motivation: Challenge in CL

Discovering notions of tasks without supervision while learning incrementally is hard in CL setting:



Motivation: Challenge in CL

What makes incremental learning (or CL) so hard?

- Neural network forgets past tasks while learning new ones

Conceptualization of this challenge

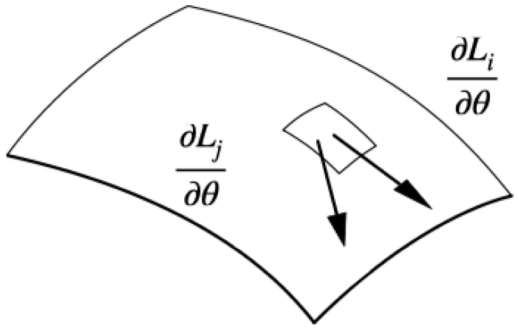
- Catastrophic forgetting (Lack of stability in NN)
  - Focus was on limiting weight sharing
- Stability-plasticity dilemma
  - Focus was on balancing limited weight sharing while ensuring fast learning
- **Transfer-interference trade-off**
  - **?**

Transfer and Interference

Operational measures of Transfer and Interference:

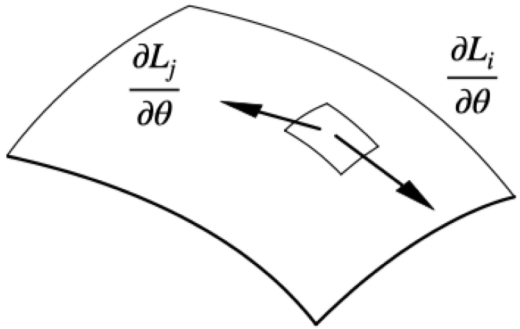
$$\frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta} > 0$$

B. Transfer

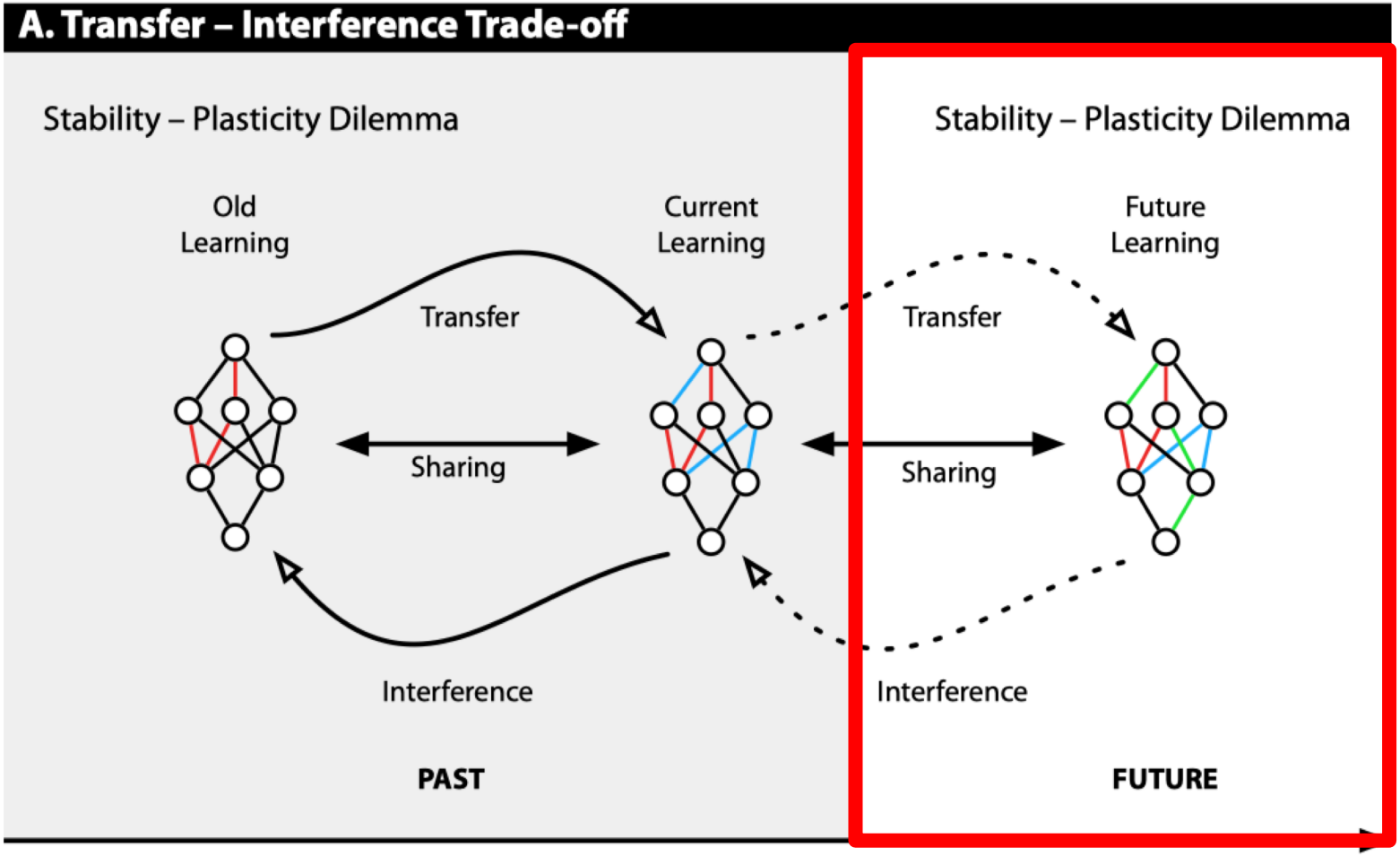


C. Interference

$$\frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta} < 0$$



Transfer-Interference Trade-off



Previous Approach: GEM

Gradient Episodic Memory for Continual Learning

- Goal: Minimize negative backward transfer (catastrophic forgetting) by the efficient use of episodic memory
- Key idea: Use the closest gradient to the original that has positive gradient dot products with gradients from previous tasks

Limitation

1) It does not consider positive forward transfer

2) Gradients are adjusted in an ad hoc manner

(Solving a quadratic equation separately for each example)

Proposed Method

Meta **E**xperience **R**eplay (a.k.a. **MER**)

Problem: CL setting without task ID

Goal: Learn the optimal weight sharing scheme to balance between new learning and forgetting

Advantages

- It learns a generalizable theory about weight sharing
- It enables transfer to improve future performance while not disrupting performance on previous tasks

Learning to learn w/o forgetting

In typical offline supervised learning with **stationary** data distribution,

$$\theta = arg\min_{\theta} \mathbb{E}_{(x,y)\sim D}[L(x,y)]$$

In the same setting with **non-stationary** data distribution, given two samples $(x_i, y_i), (x_j, y_j)$,

$$\theta = arg\min_{\theta} \mathbb{E}_{[(x_i,y_i),(x_j,y_j)]\sim D}[L(x_i,y_i) + L(x_j,y_j) - \alpha \frac{\partial L(x_i,y_i)}{\partial \theta} \cdot \frac{\partial L(x_j,y_j)}{\partial \theta}]$$

Challenges

In the same setting with **non-stationary** data distribution, given two samples $(x_i, y_i), (x_j, y_j)$,

$$\theta = arg\min_{\theta} \mathbb{E}_{[(x_i,y_i),(x_j,y_j)]\sim D}[L(x_i, y_i) + L(x_j, y_j) - \alpha \frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta}]$$

Challenge

1) We have non-stationary stream of data

2) The gradient of this loss depends on the second derivative of the loss function (EXPENSIVE to compute)

Learning to learn w/o forgetting

In the same setting with **non-stationary** data distribution, given two samples $(x_i, y_i), (x_j, y_j)$,

$$\theta = arg\min_{\theta} \mathbb{E}_{[(x_i,y_i),(x_j,y_j)]\sim D}[L(x_i, y_i) + L(x_j, y_j) - \alpha \frac{\partial L(x_i, y_i)}{\partial \theta} \cdot \frac{\partial L(x_j, y_j)}{\partial \theta}]$$

Optimize over the new meta-learning loss (Reptile):

$$\theta = arg\min_{\theta} \mathbb{E}_{B_1,\ldots,B_s\sim D}[2\sum_{i=1}^{s}[L(B_i) - \sum_{j=1}^{i-1} \alpha \frac{\partial L(B_i)}{\partial \theta} \cdot \frac{\partial L(B_j)}{\partial \theta}]]$$

Learning to learn w/o forgetting

Optimize over the new meta-learning loss (Reptile):

$$\theta = arg\min_{\theta} \mathbb{E}_{B_1,\ldots,B_s \sim D}[2\sum_{i=1}^{s}[L(B_i) - \sum_{j=1}^{i-1}\alpha\frac{\partial L(B_i)}{\partial\theta}\cdot\frac{\partial L(B_j)}{\partial\theta}]]$$

Optimize over the proposed loss (MER):

$$\theta = arg\min_{\theta} \mathbb{E}_{[(x_{11},y_{11}),\ldots,(x_{sk},y_{sk})]\sim M}[2\sum_{i=1}^{s}\sum_{j=1}^{k}[L(x_{ij},y_{ij}) - \sum_{q=1}^{i-1}\sum_{r=1}^{j-1}\alpha\frac{\partial L(x_{ij},y_{ij})}{\partial\theta}\cdot\frac{\partial L(x_{qr},y_{qr})}{\partial\theta}]]$$

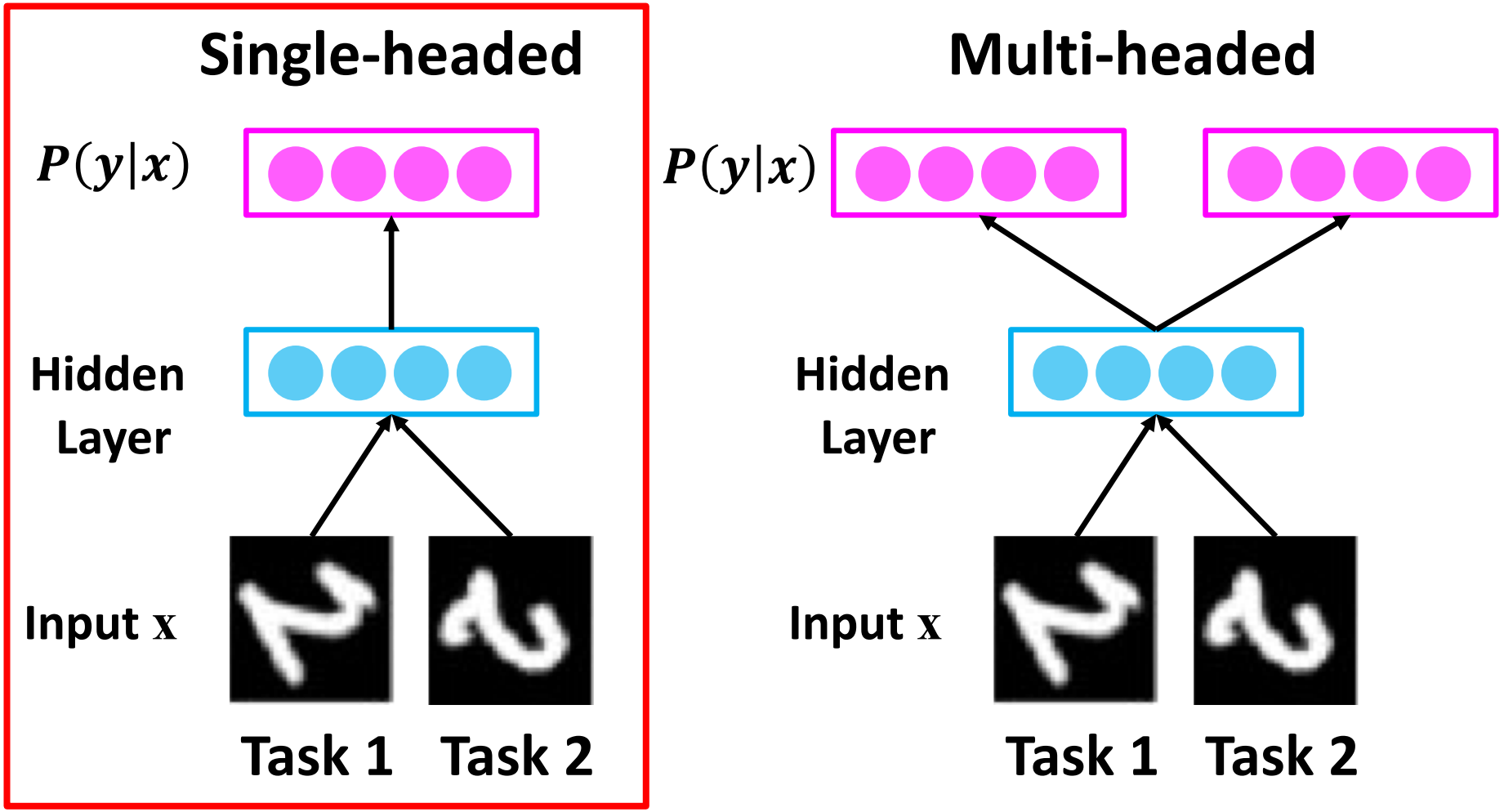
MER

1. Prioritizing the current example
2. Combining ER with optimization based meta-learning (Reptile)
3. Reservoir Sampling

Algorithm 1 Meta-Experience Replay (MER)

procedure TRAIN($D, \theta, \alpha, \beta, \gamma, s, k$)
  $M \leftarrow \{\}$
  for $t = 1, ..., T$ **do**
    for (x, y) in D_t **do**
      // Draw batches from buffer:
      $B_1, ..., B_s \leftarrow sample(x, y, s, k, M)$
      $\theta_0^A \leftarrow \theta$
      for $i = 1, ..., s$ **do**
        $\theta_{i,0}^W \leftarrow \theta$
        for $j = 1, ..., k$ **do**
          $x_c, y_c \leftarrow B_i[j]$
          $\theta_{i,j}^W \leftarrow SGD(x_c, y_c, \theta_{i,j-1}^W, \alpha)$
        end for
        // Within batch Reptile meta-update:
        $\theta \leftarrow \theta_{i,0}^W + \beta(\theta_{i,k}^W - \theta_{i,0}^W)$
        $\theta_i^A \leftarrow \theta$
      end for
      // Across batch Reptile meta-update:
      $\theta \leftarrow \theta_0^A + \gamma(\theta_s^A - \theta_0^A)$
      // Reservoir sampling memory update:
      $M \leftarrow M \cup \{(x, y)\}$ (algorithm 3)
    end for
  end for
  return θ, M
end procedure

Experiment Setting



Result

How does MER perform on supervised CL benchmarks?

Model	MNIST Rotations			MNIST Permutations		
	RA	LA	BTI	RA	LA	BTI
Online	53.38 ± 1.53	58.82 ± 2.17	-5.44 ± 1.70	55.42 ± 0.65	69.18 ± 0.99	-13.76 ± 1.19
Independent	60.74 ± 4.55	60.74 ± 4.55	-	55.80 ± 4.79	55.80 ± 4.79	-
Task Input	79.98 ± 0.66	81.04 ± 0.34	-1.06 ± 0.59	80.46 ± 0.80	81.20 ± 0.52	-0.74 ± 1.10
EWC	57.96 ± 1.33	78.38 ± 0.72	-20.42 ± 1.60	62.32 ± 1.34	75.64 ± 1.18	-13.32 ± 2.24
GEM	87.58 ± 0.32	86.46 ± 0.46	+1.12 ± 0.35	83.02 ± 0.23	80.46 ± 0.38	**+2.56** ± 0.42
MER	**89.56** ± 0.11	**87.62** ± 0.16	**+1.94** ± 0.18	**85.50** ± 0.16	**86.36** ± 0.21	-0.86 ± 0.21

- RA: Retained Accuracy
- LA: Learning Accuracy
- BTI: Backward Transfer and Interference

Result

How do the performance gains from MER vary as a function of the buffer size?

Model	Buffer	MNIST Rotations			MNIST Permutations		
		RA	LA	BTI	RA	LA	BTI
GEM	5120	87.58 ± 0.32	86.46 ± 0.46	+1.12 ± 0.35	83.02 ± 0.23	80.46 ± 0.38	**+2.56** ± 0.42
	500	74.88 ± 0.93	85.90 ± 0.53	-11.02 ± 1.22	69.26 ± 0.66	80.28 ± 0.52	-11.02 ± 0.71
	200	67.38 ± 1.75	**85.40** ± 0.53	-18.02 ± 1.99	55.42 ± 1.10	79.84 ± 1.01	-24.42 ± 1.10
MER	5120	**89.56** ± 0.11	**87.62** ± 0.16	**+1.94** ± 0.18	**85.50** ± 0.16	**86.36** ± 0.21	-0.86 ± 0.21
	500	**82.08** ± 0.31	**85.66** ± 0.20	**-3.58** ± 0.39	**77.50** ± 0.46	**83.90** ± 0.23	**-6.40** ± 0.35
	200	**77.42** ± 0.78	83.02 ± 0.23	**-5.60** ± 0.70	**73.46** ± 0.45	**84.42** ± 0.20	**-9.96** ± 0.45

- The benefits of MER seem to grow as the buffer becomes smaller.

Result

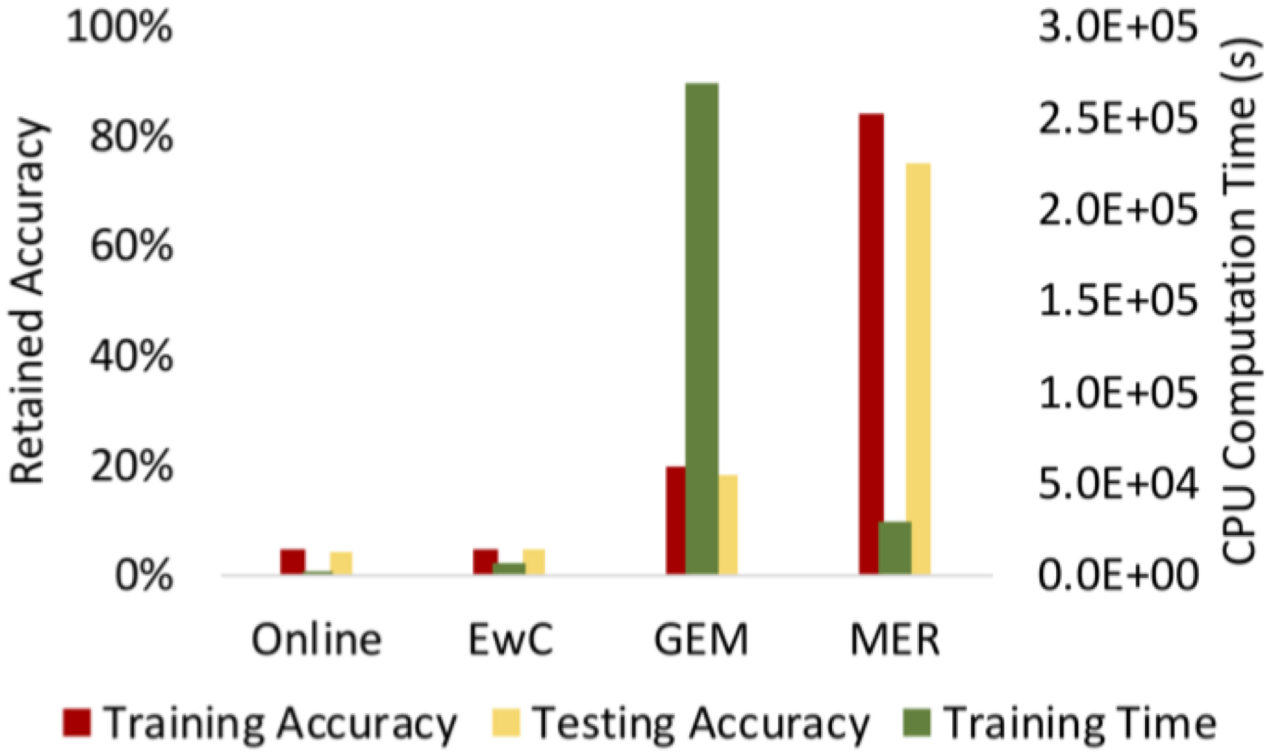
How effective is MER at dealing with increasingly non-stationary settings?

Model	Buffer	Many Permutations			Omniglot		
		RA	LA	BTI	RA	LA	BTI
Online	0	32.62 ± 0.43	51.68 ± 0.65	-19.06 ± 0.86	4.36 ± 0.37	5.38 ± 0.18	-1.02 ± 0.33
EWC	0	33.46 ± 0.46	51.30 ± 0.81	-17.84 ± 1.15	4.63 ± 0.14	9.43 ± 0.63	-4.80 ± 0.68
GEM	5120	56.76 ± 0.29	**59.66** ± 0.46	-2.92 ± 0.52	18.03 ± 0.15	3.86 ± 0.09	**+14.19** ± 0.19
	500	32.14 ± 0.50	55.66 ± 0.53	-23.52 ± 0.87	-	-	-
MER	5120	**62.52** ± 0.32	**59.44** ± 0.23	**+3.08** ± 0.31	**75.23** ± 0.52	**69.12** ± 0.83	+6.11 ± 0.62
	500	**47.40** ± 0.35	**65.18** ± 0.20	**-17.78** ± 0.39	**32.05** ± 0.69	**28.78** ± 0.91	+3.27 ± 1.04

- Many Permutations is a variant of MNIST Permutations that has 5 times more tasks (100 total) and 5 times less training examples per task (200 each)

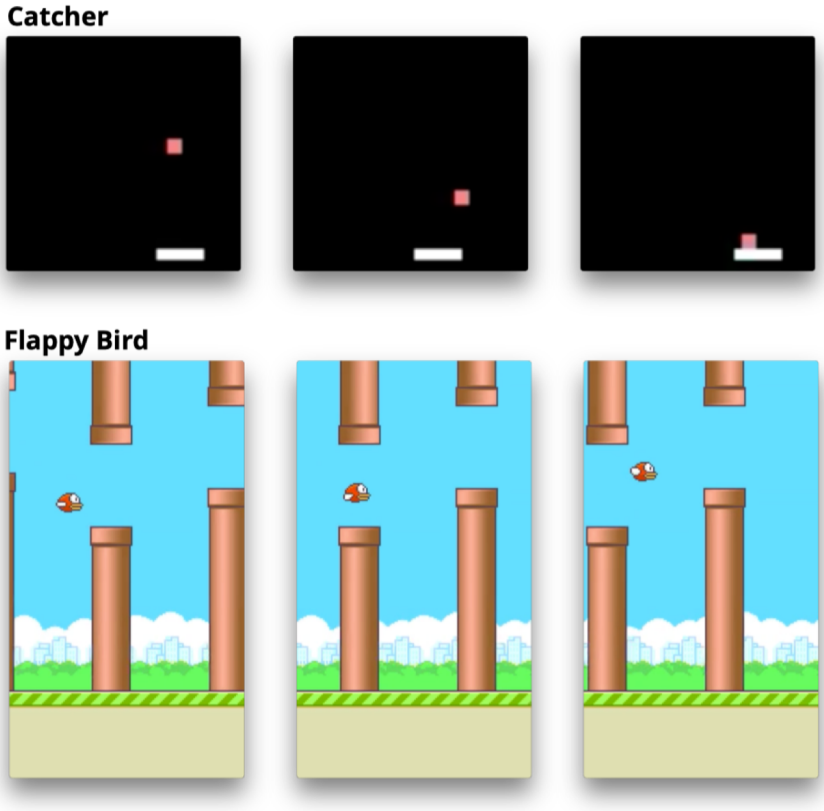
Result

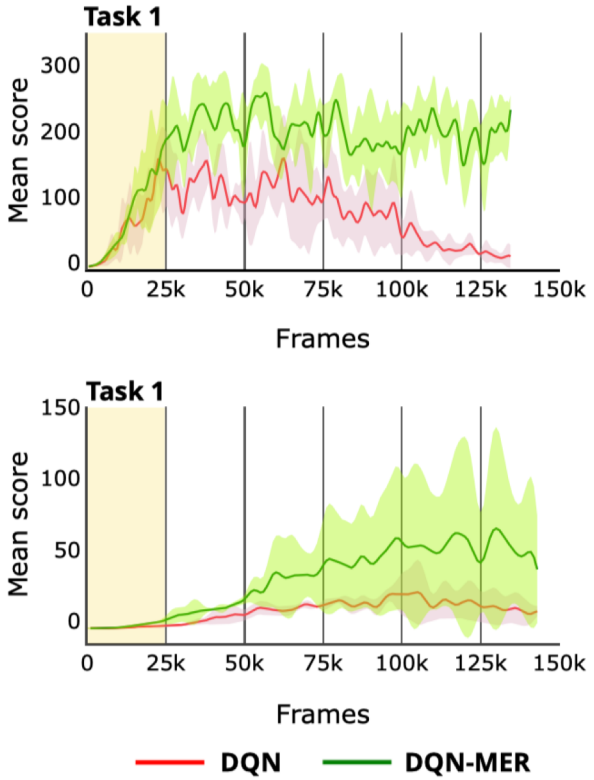
How effective is MER at dealing with increasingly non-stationary settings?



Result

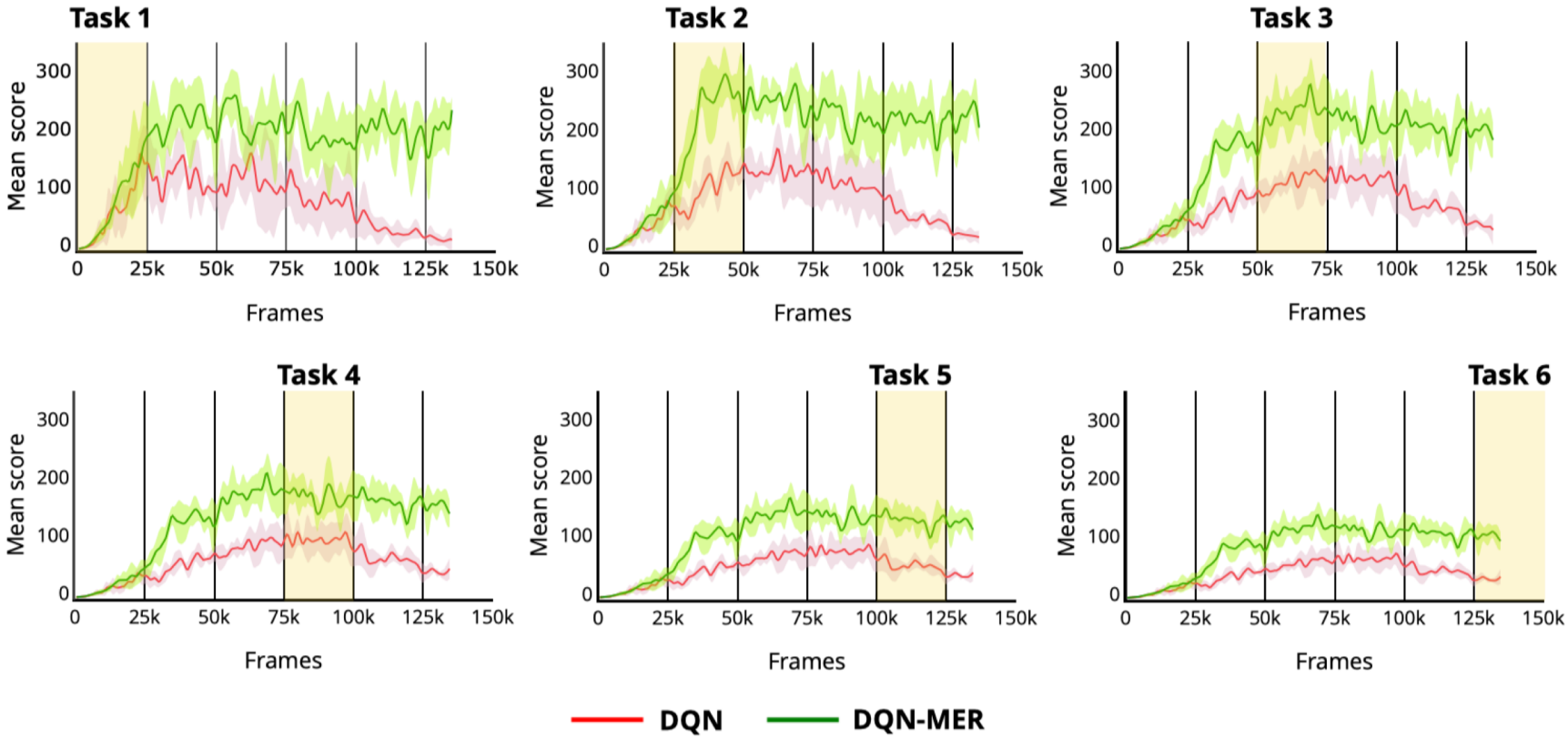
Can MER improve a DQN with ER in CRL settings?





Result

Can MER improve a DQN with ER in CRL settings?



Result

Does MER lead to a shift in the distribution of gradient dot products?

Model	MNIST Permutations	MNIST Rotations	Many Permutations
ER	-0.569 $\pm$ 0.077	-1.652 $\pm$ 0.082	-1.280 $\pm$ 0.078
MER	+0.042 $\pm$ 0.017	+0.017 $\pm$ 0.007	+0.131 $\pm$ 0.027

- It verifies that a very significant and reproducible difference in the mean gradient encountered is seen for MER in comparison to ER alone

Conclusion

This work cast a new perspective on CL problem: trade-off btw transfer and interference.

They also propose a new algorithm MER that

- regularizes the objective of experience replay so that gradients on incoming examples are more likely to have transfer and less likely to have interference with respect to past examples

The experimental results show that it outperforms baselines in non-stationary supervised and reinforcement learning setting

Future Work

Combine MER with mixtures of experts model to amplify the ability to control weight sharing dynamics

Combine MER with option frameworks allowing for orthogonalization of weights relating to different skills

References

[MIR19] Matthew Riemer et al., “Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference”, ICLR 2019

Question or Discussion