CONTINUAL LEARNING IN NEURAL NETWORKS: ON CATASTROPHIC FORGETTING AND BEYOND

Polina Kirichenko

New York University

NYU

PLAN

- Continual learning & catastrophic forgetting
- Alleviating forgetting
  - Replay
  - Regularization
  - Expansion
- Learning continually with invertible models [if we have enough time]

CONTINUAL LEARNING (CL)

- A model is presented with a sequence of tasks $T_{t_1}, T_{t_2}, \ldots, T_{t_\tau}$ with task IDs $t_i \in \{1, \ldots, M\}$
- When training on task $T_i = \{(x^i_j, y^i_j)\}_{j=1}^{N_i}$ we don't have access to previous data
- The assumption is that the tasks will be revisited in the future

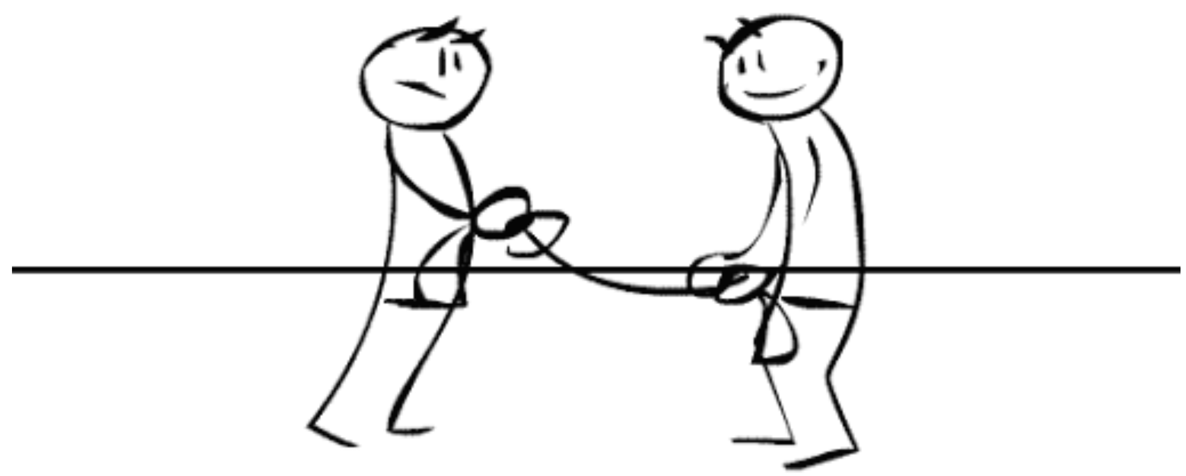
- **Catastrophic forgetting**: model's performance on previous task degrades when training on new one

GOALS

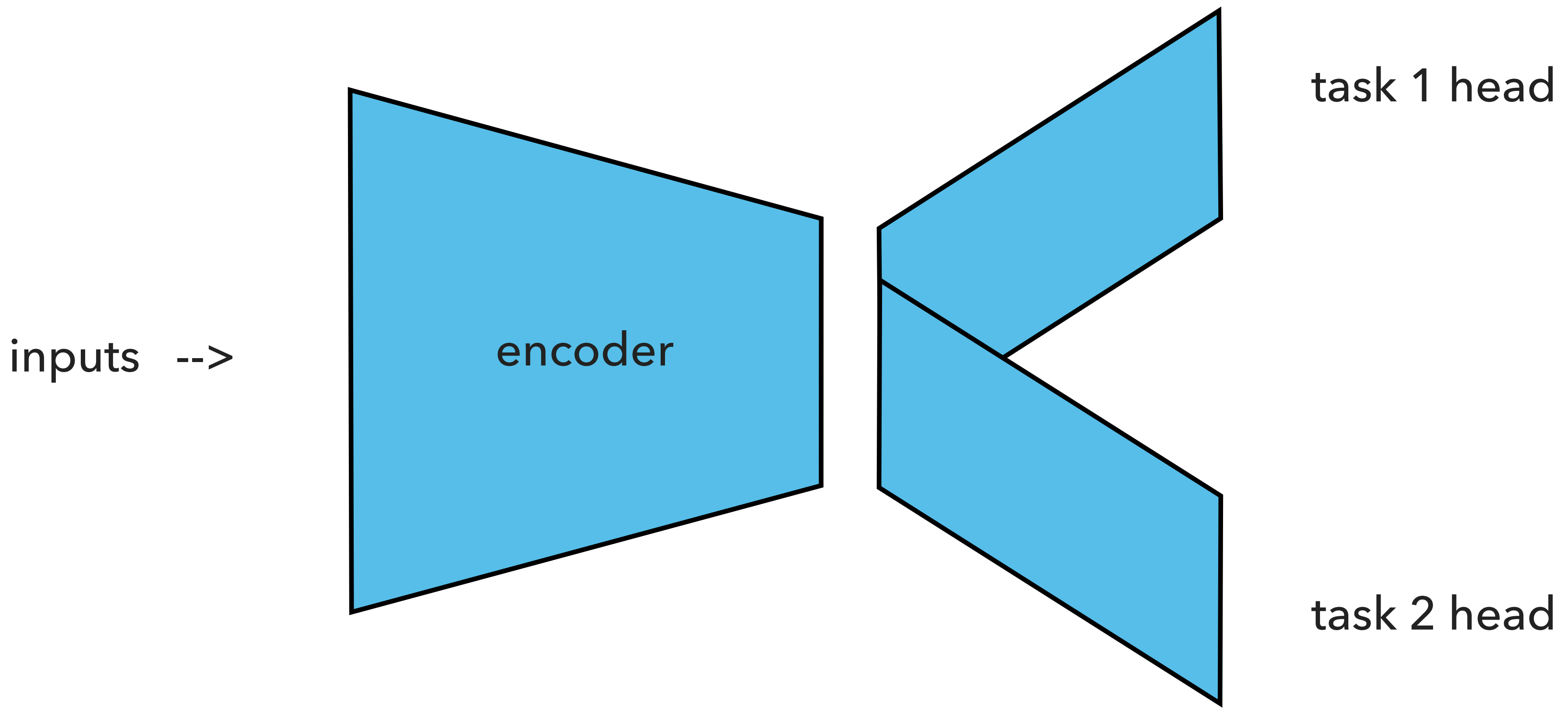
- **Mitigate forgetting**
- Transfer knowledge to new and old tasks
- Fixed or limited memory and computation (scalability)

CONTINUAL LEARNING: CATASTROPHIC FORGETTING

- Tug-of-war dynamics while learning on non iid distribution
- Problem of "locality" of learning and optimization
- Stability-plasticity tradeoff

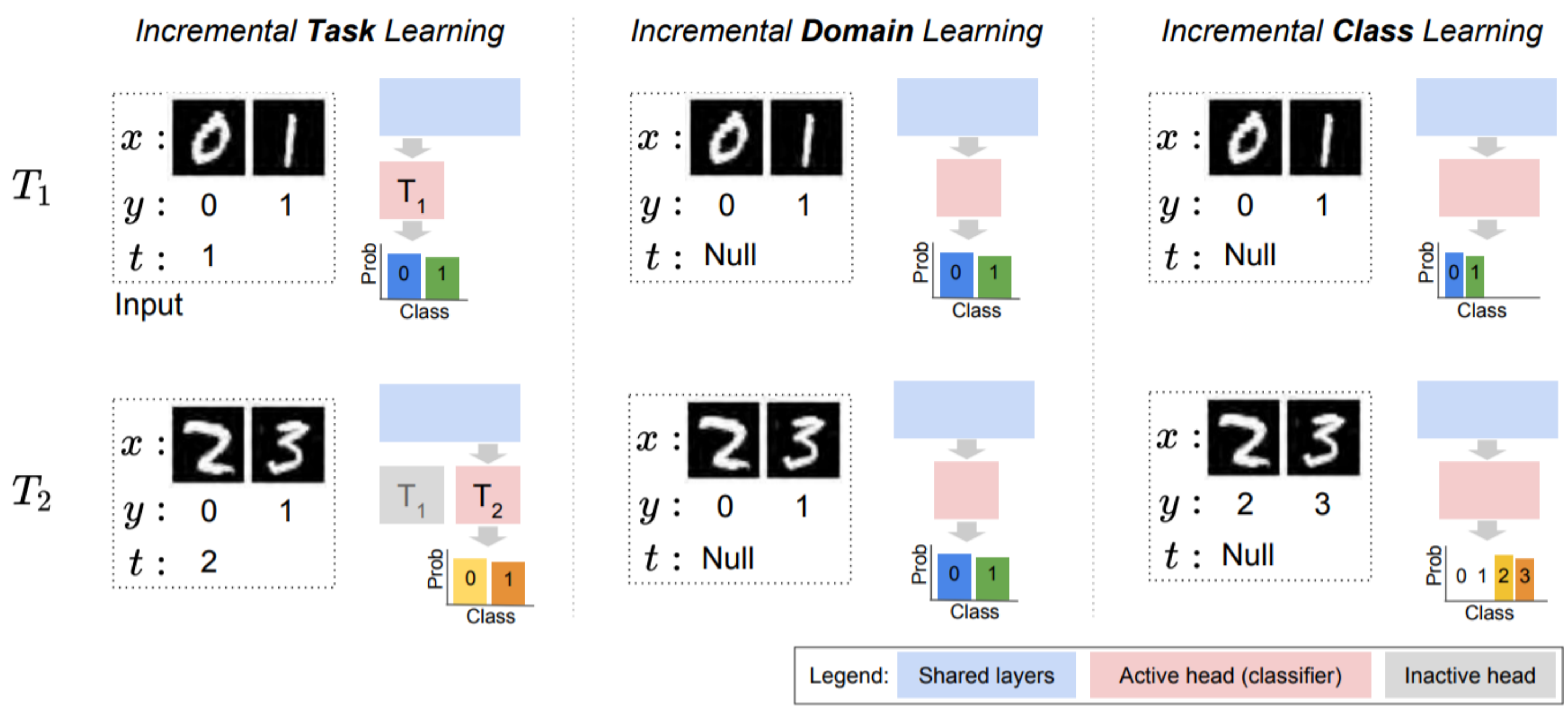


CONTINUAL LEARNING NEURAL NETWORK



CONTINUAL LEARNING SCENARIOS

- Split MNIST task: T1 is classes 0 and 1, T2 is classes 2 and 3, …



Hsu et al, Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines

CONTINUAL LEARNING BENCHMARKS

- ''Split'' datasets: MNIST, CIFAR, ImageNet, CUB [Wah et al 2011]
- Permuted or rotated MNIST, SVHN-MNIST
- Taskonomy [Zamir et al 2018]

METRICS

Let $a_{i,j}$ be the accuracy of the model on task i after training on task j

- Final test accuracy $a_{i,\tau}$ on each task i or average across tasks $\frac{1}{d}\sum_{i=1}^{d} a_{i,\tau}$

- Average forgetting $\frac{1}{\tau-1}\sum_{i=1}^{\tau-1}(a_{i,i} - a_{i,\tau})$ (or backward transfer!)

- Forward transfer $$\frac{1}{\tau - 1}\sum_{i=2}^{\tau-1}(a_{i,i-1} - r_i)$$ [Lopez-Paz & Ranzato 2017]

ALLEVIATING CATASTROPHIC FORGETTING

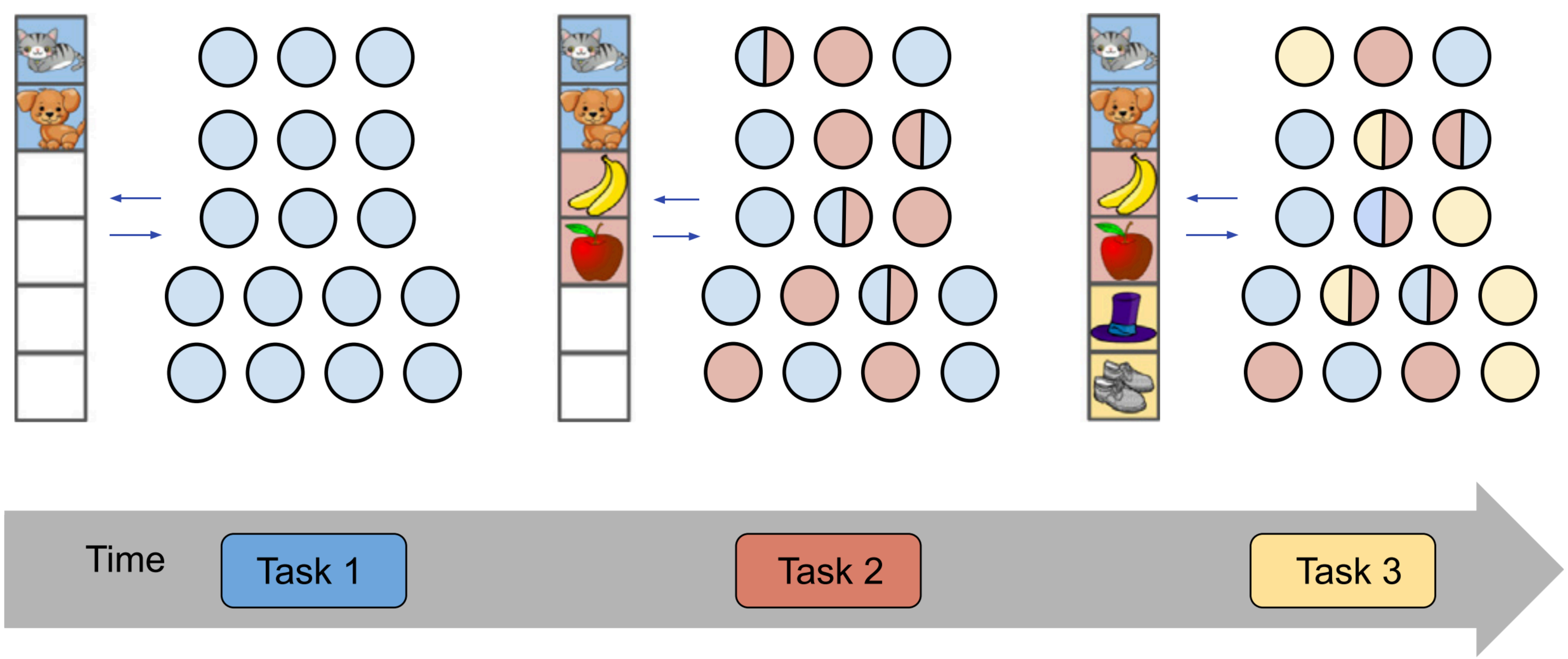
- Replay based
- Regularization based (parameter and function space)
- Expansion based (adding capacity)

+ their combination

- Baselines:
  - Upper bound: Multi-Task Learning (T1, T1+T2, …) or iid training
  - Lower bound: SGD (more common than adaptive optimizers in non iid tasks)

REPLAY

- Write task data to fixed size memory and use it later to prevent forgetting
- Need to choose: update rule using replay samples, sampling strategy to fill the replay buffer



REPLAY

- Reservoir sampling / choosing samples uniformly at random

Algorithm 1 Experience Replay for Continual Learning.

procedure ER($\mathcal{D}$, mem_sz, batch_sz, lr)
    $\mathcal{M} \leftarrow \{\} * \text{mem\_sz}$ ▷ Allocate memory buffer of size mem_sz
    $n \leftarrow 0$ ▷ Number of training examples seen in the continuum
    for $t \in \{1, \cdots, T\}$ **do**
        for $B_n \overset{K}{\sim} \mathcal{D}_t$ **do** ▷ Sample without replacement a mini-batch of size K from task t
            $B_{\mathcal{M}} \overset{K}{\sim} \mathcal{M}$ ▷ Sample a mini-batch from $\mathcal{M}$
            $\theta \leftarrow SGD(B_n \cup B_{\mathcal{M}}, \theta, \text{lr})$ ▷ Single gradient step to update the parameters by stacking current minibatch with minibatch from memory
            $\mathcal{M} \leftarrow \texttt{UpdateMemory}(\text{mem\_sz}, t, n, B_n)$ ▷ Memory update, see §4
            $n \leftarrow n + \text{batch\_sz}$ ▷ Counter update
    return $\theta, \mathcal{M}$

REPLAY

- Gradient Episodic Memory (GEM): we want the loss on memory samples to not increase

$$\begin{aligned} \text{minimize}_\theta \quad & \ell(f_\theta(x,t),y) \\ \text{subject to} \quad & \ell(f_\theta,\mathcal{M}_k) \le \ell(f_\theta^{t-1},\mathcal{M}_k) \text{ for all } k<t \end{aligned}$$

- Project gradients:

$$\langle g, g_k \rangle := \left\langle \frac{\partial \ell(f_\theta(x,t),y)}{\partial \theta}, \frac{\partial \ell(f_\theta,\mathcal{M}_k)}{\partial \theta} \right\rangle \ge 0, \text{ for all } k<t.$$

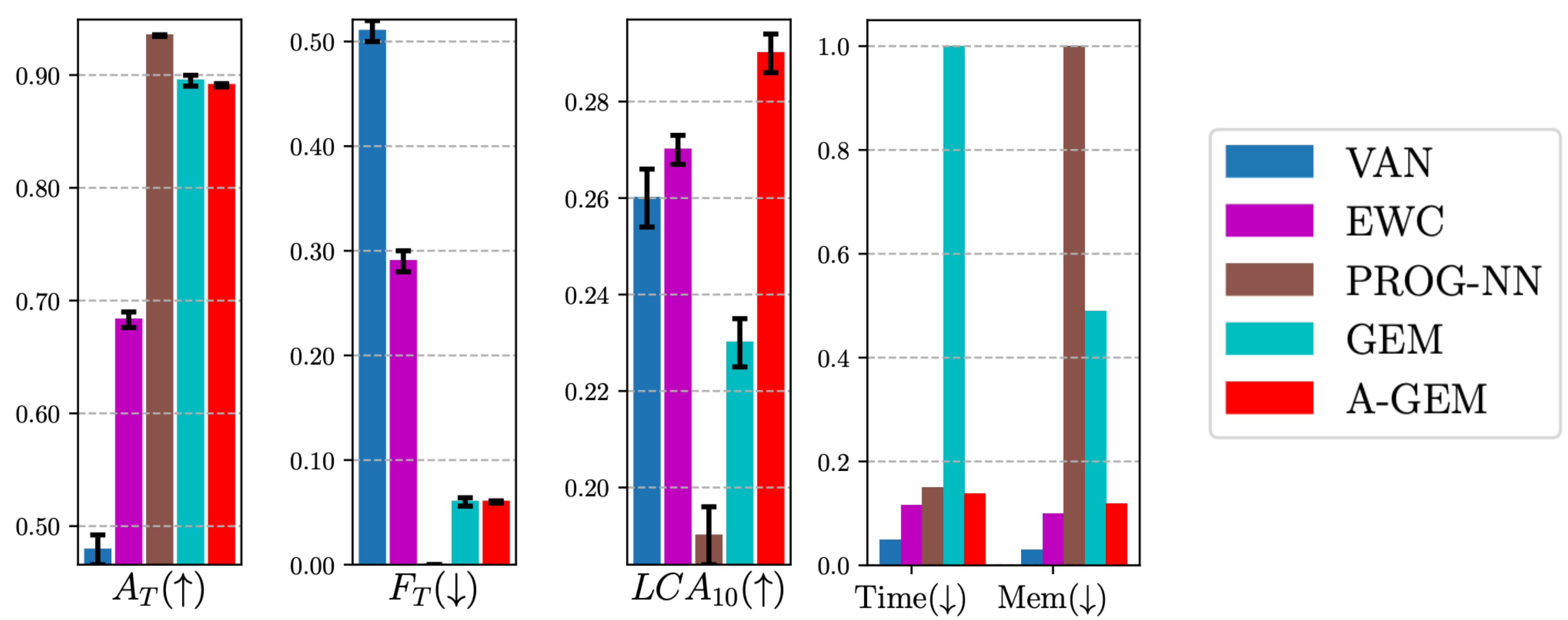
$$\begin{aligned} \text{minimize}_{\tilde{g}} \frac{1}{2} \quad & \|g-\tilde{g}\|_2^2 \\ \text{subject to} \quad & \langle \tilde{g}, g_k \rangle \ge 0 \text{ for all } k<t \end{aligned}$$

REPLAY

- Averaged GEM: more memory efficient

$$\text{minimize}_{\tilde{g}} \quad \frac{1}{2}||g - \tilde{g}||_2^2 \quad \text{s.t.} \quad \tilde{g}^\top g_{ref} \geq 0 \qquad\qquad (\mathbf{x}_{ref}, y_{ref}) \sim \mathcal{M}$$

- Project gradients: $\tilde{g} = g - \frac{g^\top g_{ref}}{g_{ref}^\top g_{ref}} g_{ref}$



(a) Permuted MNIST

INCREMENTAL LEARNING

- Incremental Classifier and Representation Learning (iCaRL): class-incremental learning setting
- Features extractor network $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$
- For representation learning $g_y(x) = 1/(1 + \exp(-w_y^T \phi(x)))$
- Exemplar set for each class P_t , classification is done via nearest mean of exemplar (class prototype)

outputs before updating

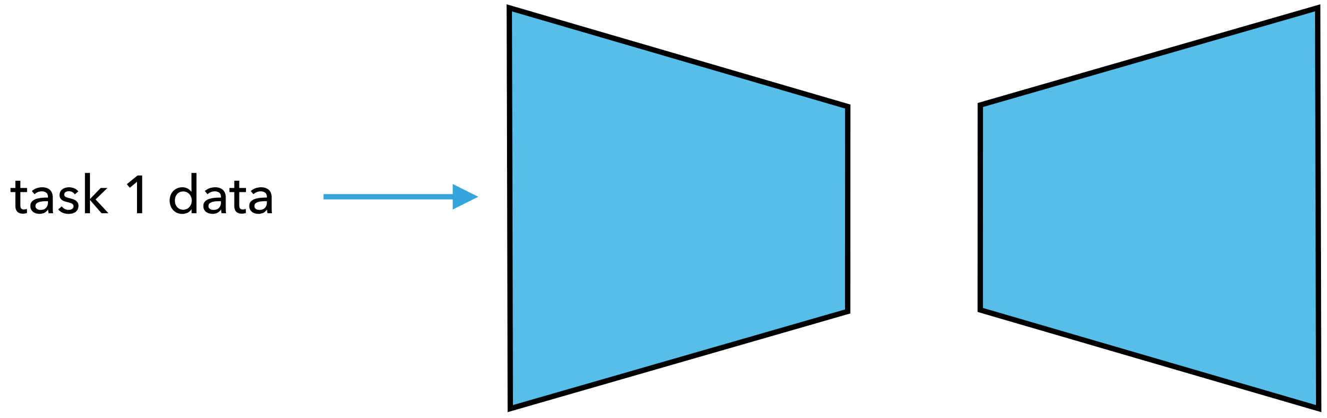
$$\ell(\Theta) = -\sum_{(x_i, y_i) \in \mathcal{D}} \Big[\sum_{y=s}^{t} \delta_{y=y_i} \log g_y(x_i) + \delta_{y \neq y_i} \log(1 - g_y(x_i)) + \sum_{y=1}^{s-1} q_i^y \log g_y(x_i) + (1 - q_i^y) \log(1 - g_y(x_i)) \Big]$$

classification loss

distillation loss

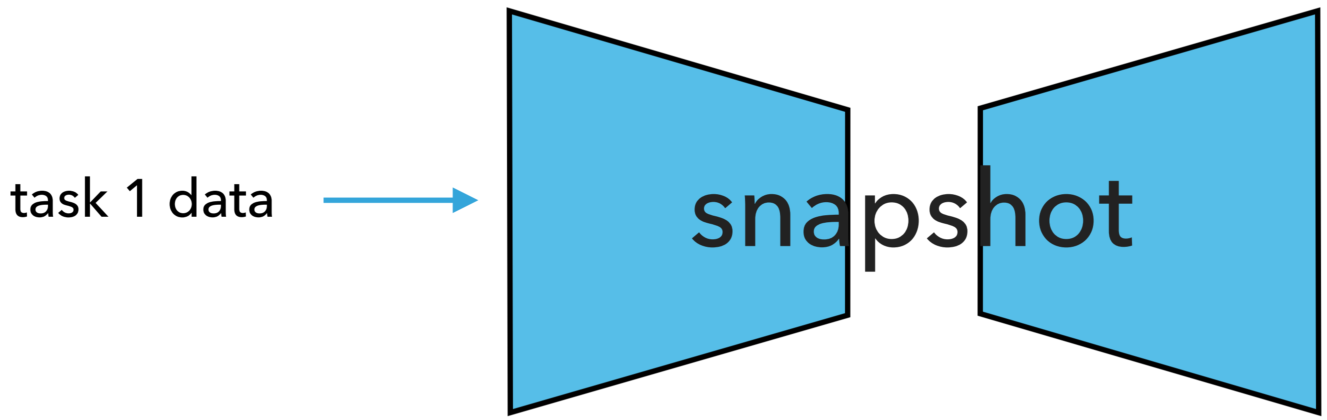
GENERATIVE REPLAY

- Continual Unsupervised Representation Learning [Rao et al 2019]



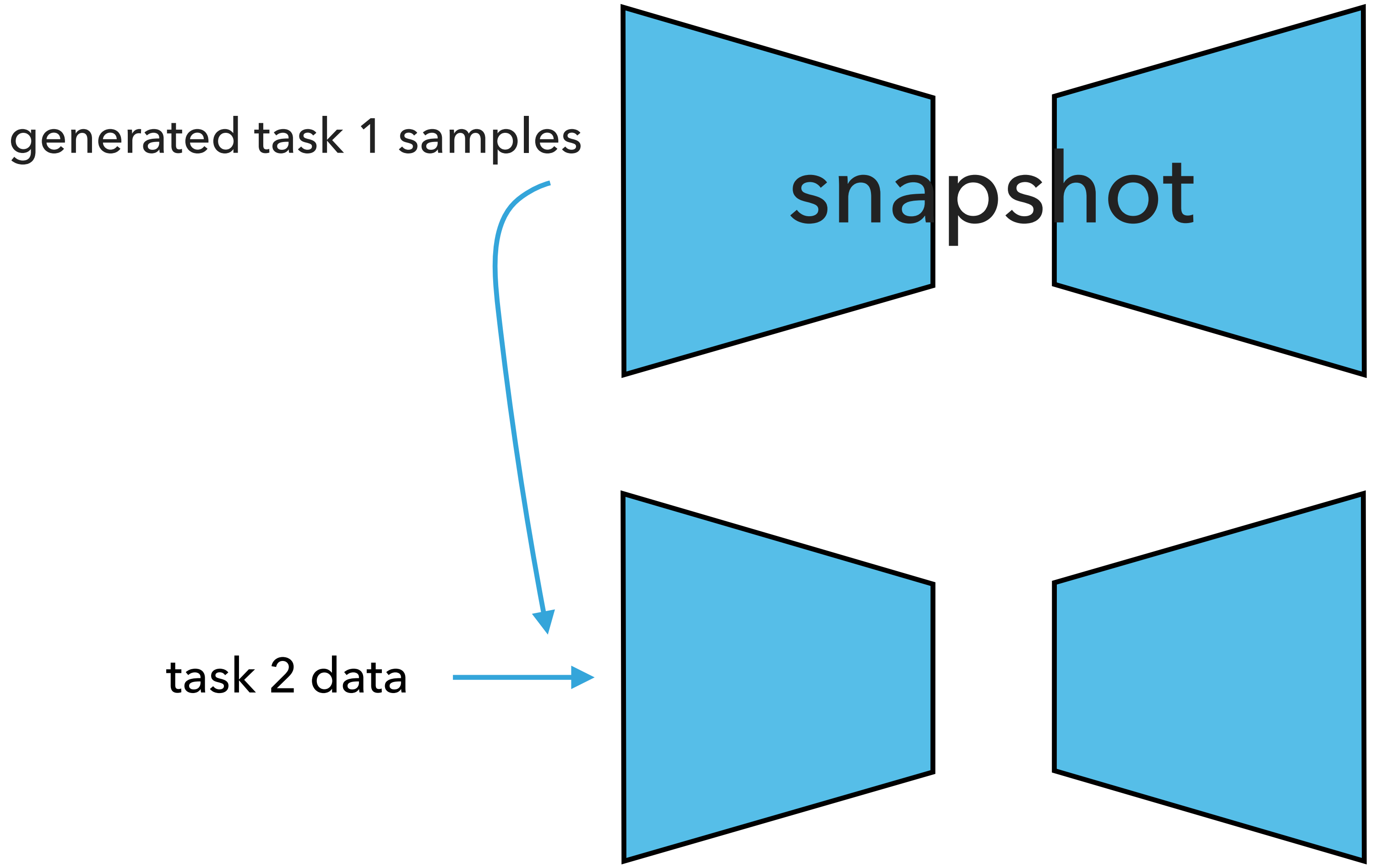
GENERATIVE REPLAY

- Continual Unsupervised Representation Learning



GENERATIVE REPLAY

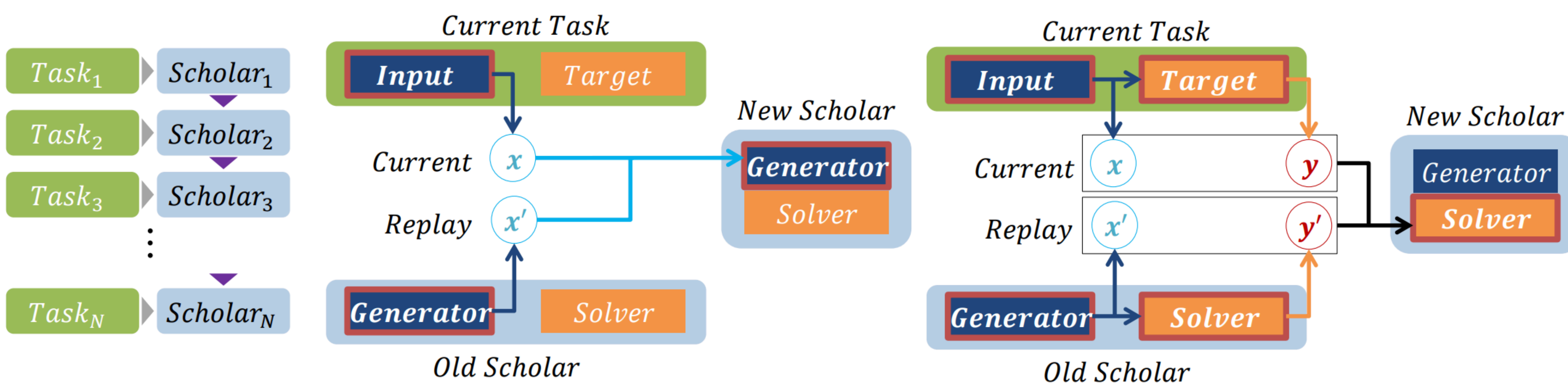
- Continual Unsupervised Representation Learning



GENERATIVE REPLAY

- Generative Adversarial Networks can be used to approximate evolving data distribution
- "Scholar" is a generator + task solver

$$L_{train}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{\boldsymbol{x}'\sim G_{i-1}}[L(S(\boldsymbol{x}';\theta_i),S(\boldsymbol{x}';\theta_{i-1}))]$$

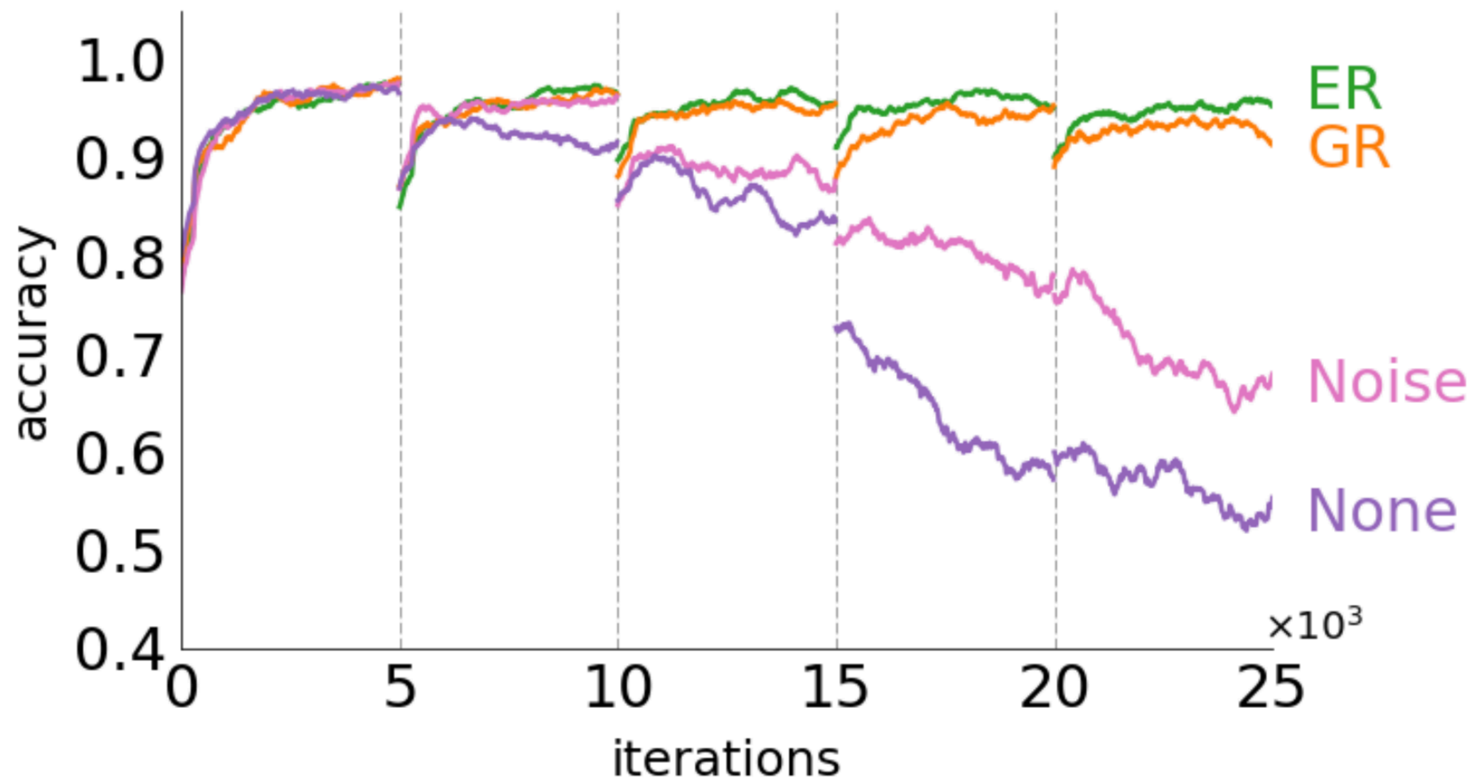


(a) Sequential Training (b) Training Generator (c) Training Solver

GENERATIVE REPLAY

- Generative Adversarial Networks can be used to approximate evolving data distribution
- ''Scholar'' is a generator + task solver

$$L_{train}(\theta_i) = r\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D_i}[L(S(\boldsymbol{x};\theta_i),\boldsymbol{y})] + (1-r)\,\mathbb{E}_{\boldsymbol{x}'\sim G_{i-1}}[L(S(\boldsymbol{x}';\theta_i),S(\boldsymbol{x}';\theta_{i-1}))]$$

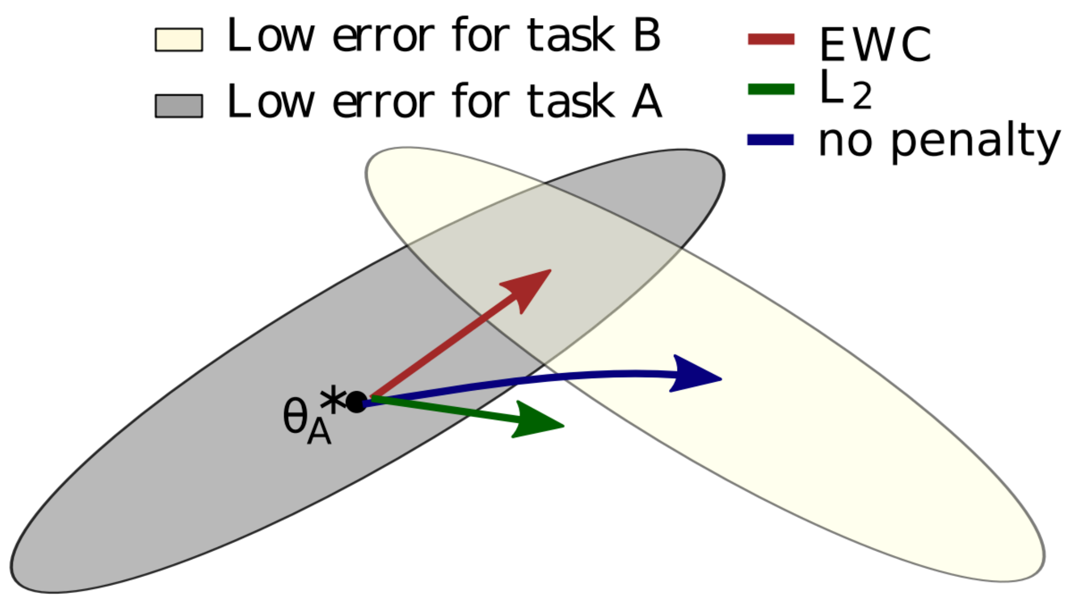


permuted MNIST

REGULARIZATION IN PARAMETER SPACE

- L2 regularization $\sum_{i=1}^{t} \alpha \|\theta - \theta_i^*\|^2$

- Estimate the importance of each parameter for previous tasks and penalize changes to each parameter proportional to this measure

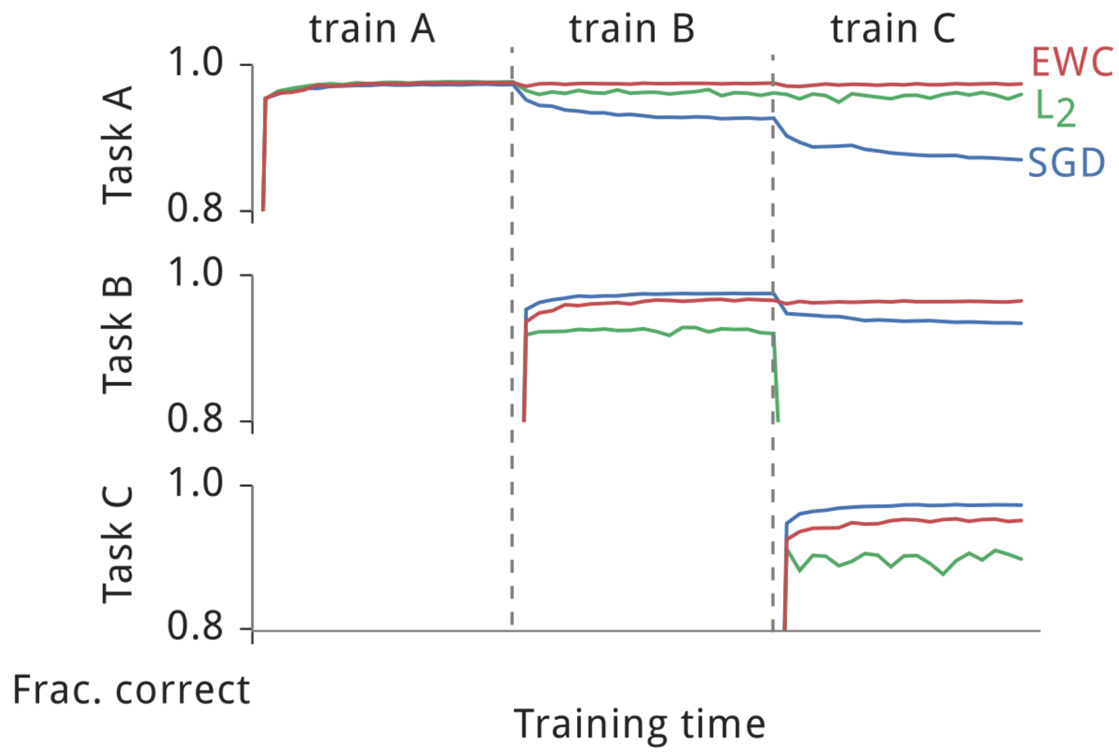


REGULARIZATION IN PARAMETER SPACE

- Elastic Weight Consolidation (EWC)

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2$$

where F_i is a diagonal of the Fisher information matrix F



permuted MNIST

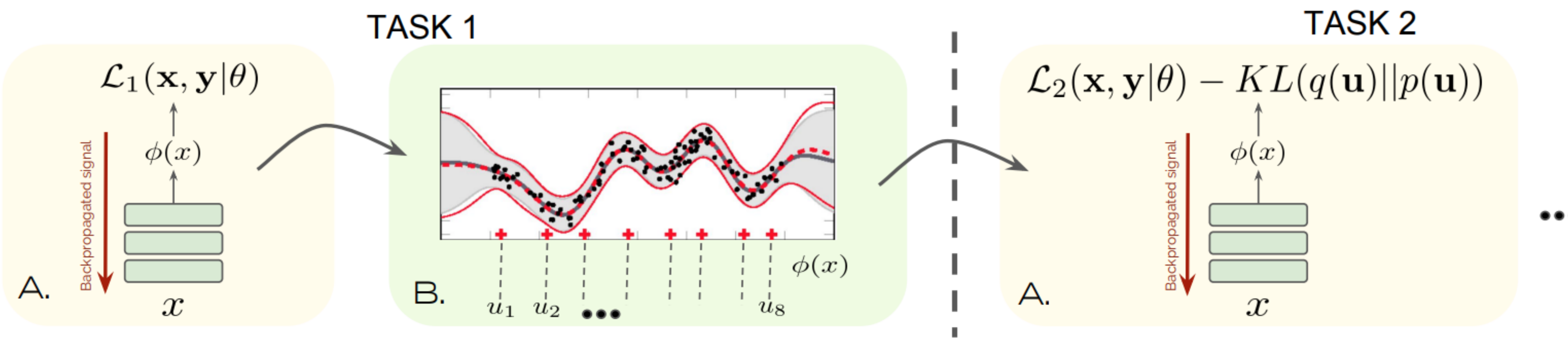
REGULARIZATION IN PARAMETER SPACE

- EWC Bayesian interpretation:

$$\log p(\theta|\mathcal{D}) = \log p(\mathcal{D}_B|\theta) + \log p(\theta|\mathcal{D}_A) - \log p(\mathcal{D}_B)$$

- Approximate posterior as a Gaussian distribution with mean θ_A^* and diagonal precision F

FUNCTIONAL REGULARIZATION FOR CONTINUAL LEARNING WITH GPS



- Replace the last layer of a neural network with a GP

$$f_i(x; w_i) \equiv f_i(x; w_i, \theta) = w_i^\top \phi(x; \theta),$$

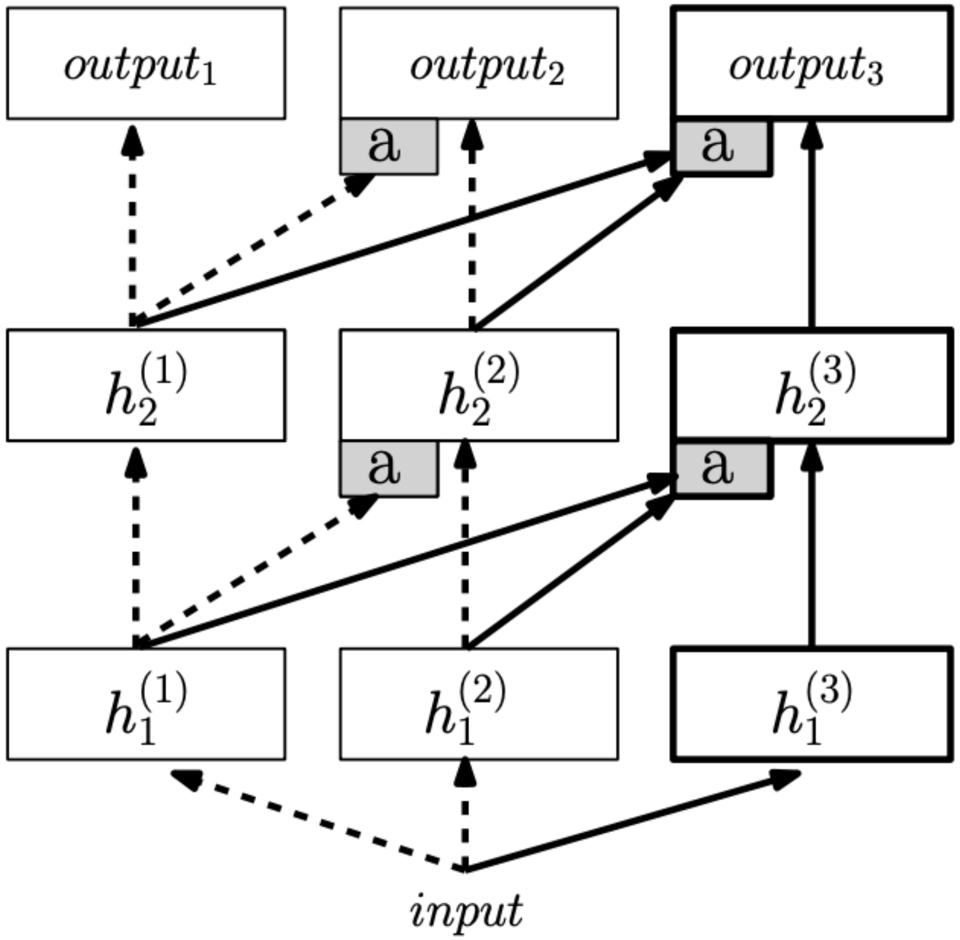
$$f_i(x) \sim \mathcal{GP}(0, k(x, x')),\ k(x, x') = \sigma_w^2 \phi(x; \theta)^\top \phi(x'; \theta),$$

- Use inducing points to avoid forgetting with the GP

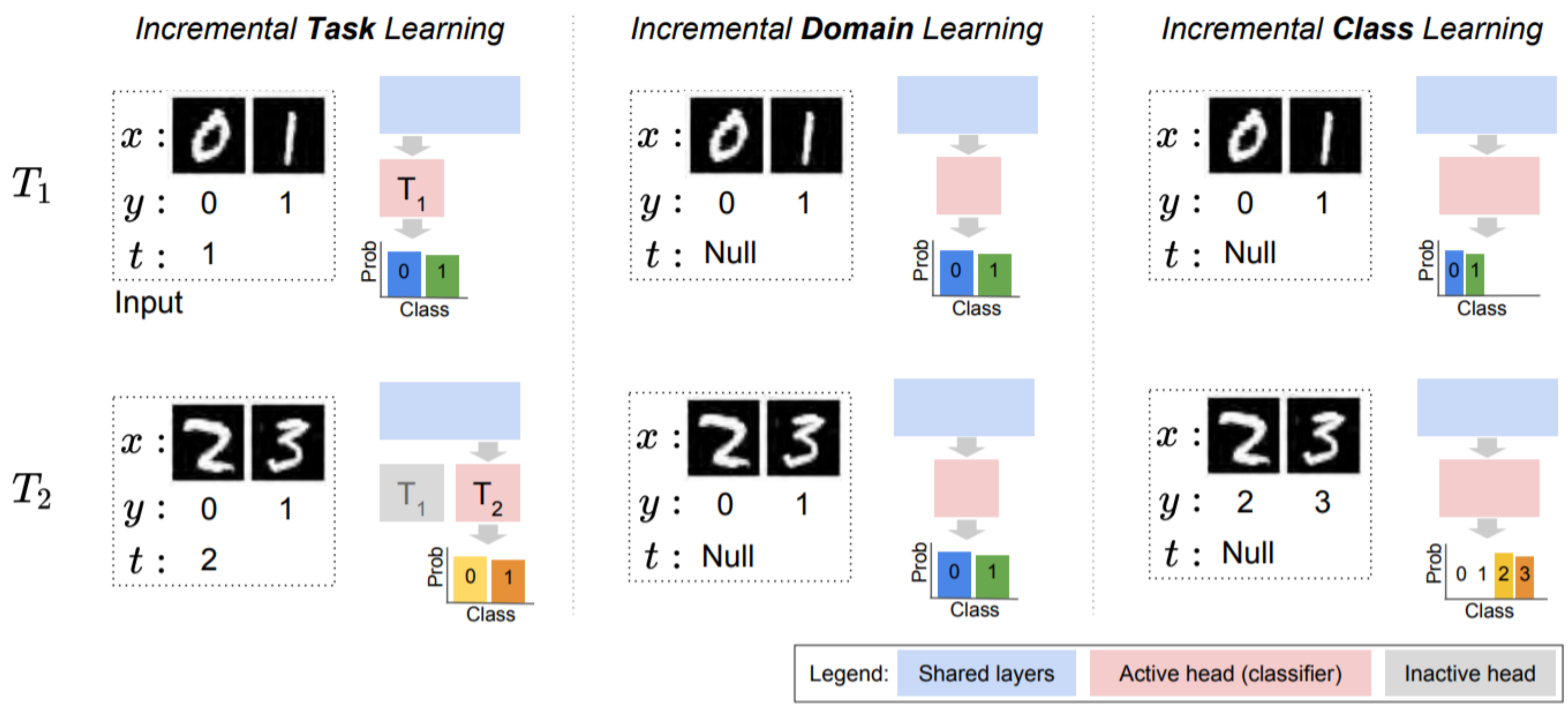
EXPANSION

- Progressive Neural Networks

$$h_i^{(k)} = f\left(W_i^{(k)} h_{i-1}^{(k)} + \sum_{j<k} U_i^{(k:j)} h_{i-1}^{(j)}\right)$$



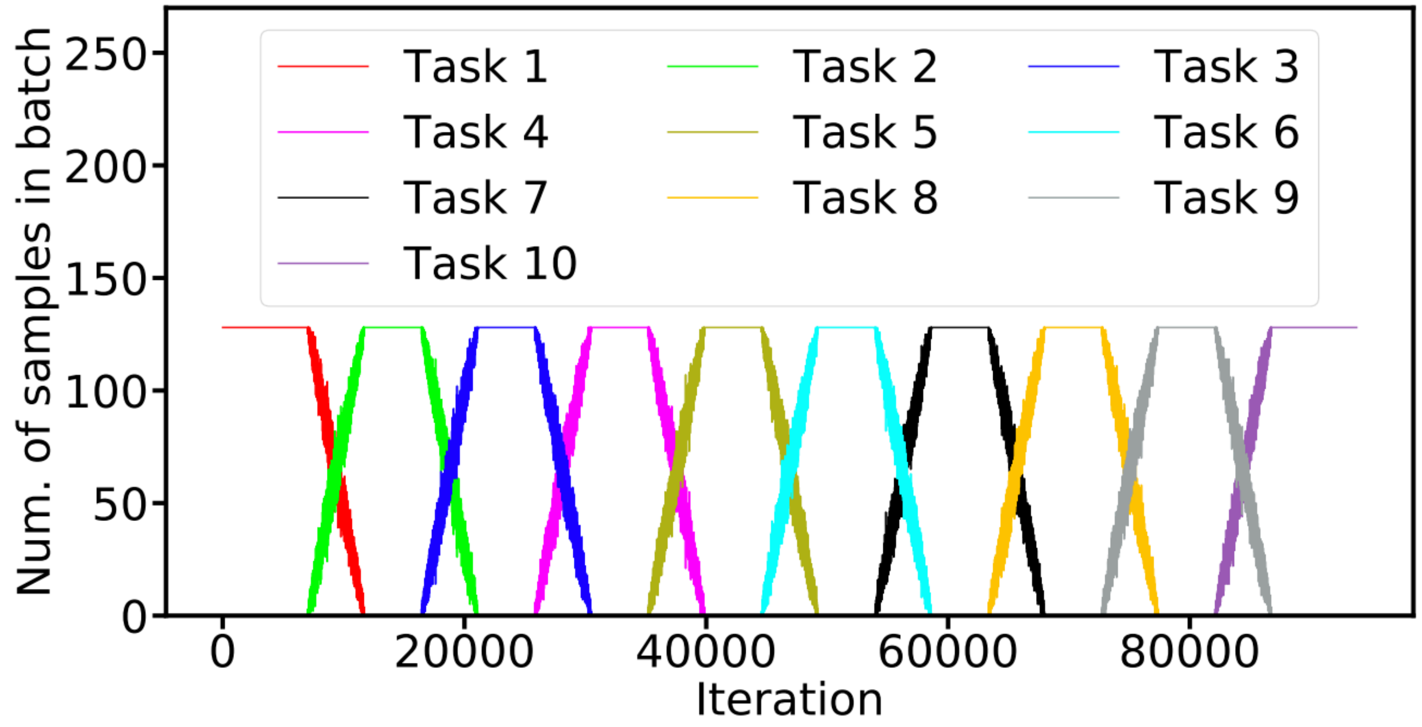
CONTINUAL LEARNING SCENARIOS



Hsu et al, Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines

CONTINUAL LEARNING SCENARIOS

- Typically the input is (x_i, y_i, t_i) but the model may not have access to task ID and only receive (x_i, y_i)
- Task agnostic domain incremental learning or unsupervised learning
- Task free: continuously drifting distribution (e.g. CIFAR-C with increasing noise intensity or mixed tasks)



BEYOND CATASTROPHIC FORGETTING

- Forward and backward transfer
- Sample efficiency: the minimum possible number of examples to replay for remembering
- Understanding continual learning and forgetting [Ramasesh et al 2020, Mirzadeh et al 2020]

REFERENCES

Kirkpatrick et al, Overcoming catastrophic forgetting in neural networks.
Hadsell et al, Embracing Change: Continual Learning in Deep Neural Networks.
Hsu et al, Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines.
Van de Ven and Tolias, Three scenarios for continual learning.
Zamir et al, Taskonomy: Disentangling task transfer learning.
Lopez-Paz and Ranzato, Gradient episodic memory for continual learning.
Chaudhry et al, Efficient lifelong learning with A-GEM.
Rebuffi et al, iCaRL: Incremental classifier and representation learning.
Rao et al, Continual unsupervised representation learning.
Shin et al, Continual learning with deep generative replay.
Titsias et al, Functional regularisation for continual learning with Gaussian Processes.
Rusu et al, Progressive neural networks.
Aljundi et al, Task-free continual learning.
Zeno et al, Task agnostic continual learning using online variational Bayes.

Questions?