Improving Task-free Continual Learning by Distributionally Robust Memory Evolution

University at Buffalo

京东探索研究院
JD EXPLORE ACADEMY

Meta

Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, Mingchen Gao

Task-free Continual Learning

- Task-free continual learning aims to learn non-stationary data stream and not forget previous knowledge

- Data distribution shift could happen arbitrarily without clear task splits

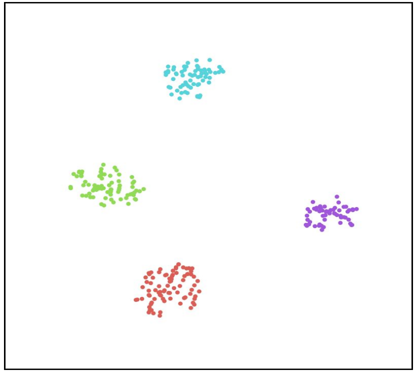
- Majority work of existing task-free CL methods are memory-replayed based methods

- Memory-replay methods optimize an objective under a known probability distribution for the memory buffer μ_0

$$\min_{\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}} [\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_k, y_k) + \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu_0} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y)],$$

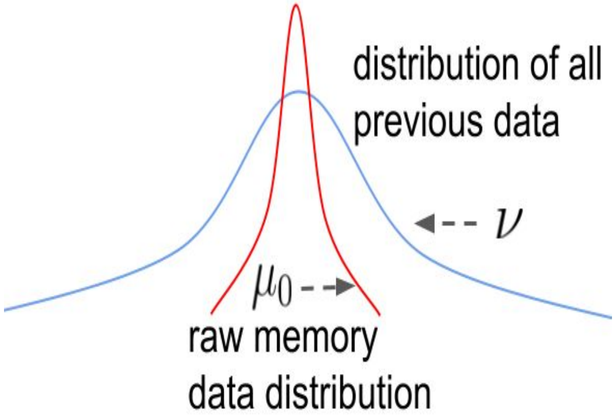
Motivation

- **Memory overfitting**: CL model would overfit the memory buffer, and memory buffer gradually less effective for mitigating forgetting as the model repeatedly learns the memory buffer



- a big gap between the memory data distribution and the distribution of all the previous data examples

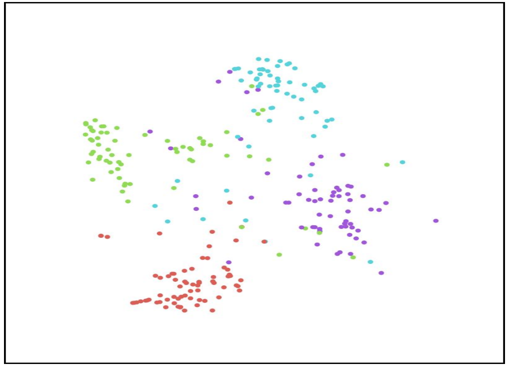
- high uncertainty in the memory data distribution since a limited memory buffer cannot accurately reflect the stationary distribution of all examples seen so far in the data stream



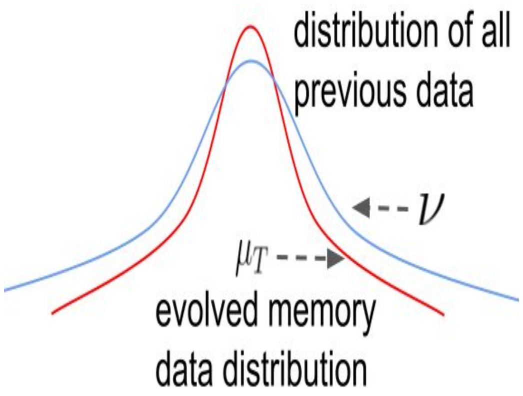
Task-free DRO

Solution: Evolve the memory data distribution by Distributionally Robust Optimization (DRO).

- Make the memory buffer data harder to classify and overfit



- Narrow the gap between the memory data distribution and the distribution of all the previous data examples.



Task-free DRO

- We optimize the worst-case evolved memory data distribution since we cannot access the actual data distribution of all the previous data examples, named task-free DRO.

$$\min_{\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}} \sup_{\mu \in \mathcal{P}} \mathbb{E}_{\mu} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y)$$

$$\text{s.t. } \mathcal{P} = \{\mu : \mathcal{D}(\mu || \pi) \leq \mathcal{D}(\mu_0 || \pi) \leq \epsilon\},$$

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu, \boldsymbol{x}' \sim \mu_0} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}', y) \geq \lambda,$$

- By Lagrange duality, convert into the following unconstrained optimization problem, still intractable to solve

$$\min_{\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}} \sup_{\mu} [\mathbb{E}_{\mu} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) - \gamma \mathcal{D}(\mu || \pi) +$$

$$\beta \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu, \boldsymbol{x}' \sim \mu_0} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y) \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}', y)],$$

Dynamic DRO

- Convert task-free DRO into a gradient flow system, named dynamic DRO
- Memory buffer evolves as Wasserstein Gradient Flow (WGF) in probability measure space of memory data.
- Model parameters follows gradient flow in Euclidean space.

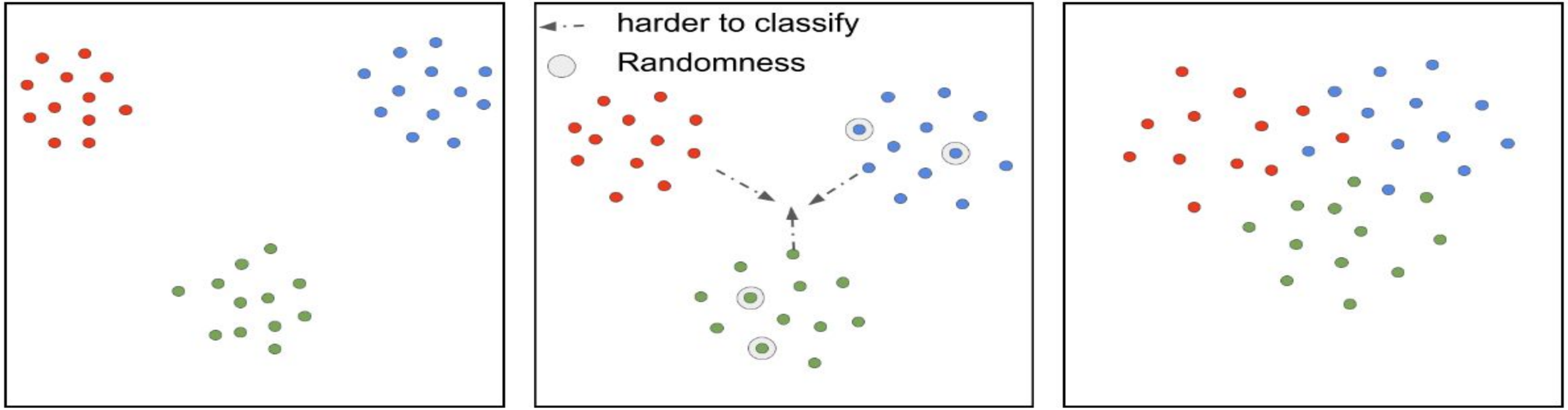
$$\begin{cases} \partial_t \mu_t &= div\left(\mu_t \nabla \frac{\delta F}{\delta \mu}(\mu_t)\right); \\ \frac{d\boldsymbol{\theta}}{dt} &= -\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mu_t} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}, y), \end{cases}$$

memory buffer evolves as WGF

model parameters follows gradient flow in Euclidean space

A family of Memory Evolution Methods for Dynamic DRO

Langevin Dynamics for Dynamic DRO (WGF-LD)



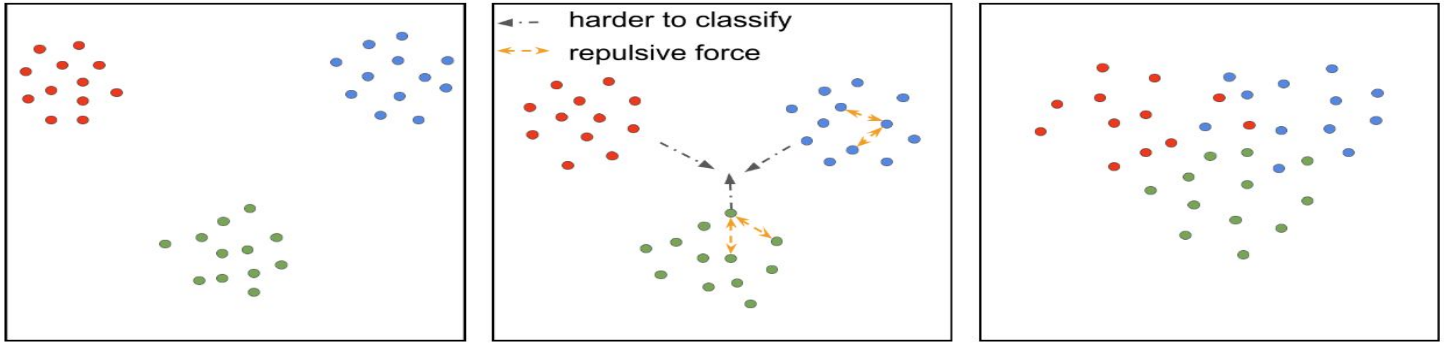
(a) $t = t_0$ (b) $t = t_1$ (c) $t = t_2$

$$dX = -\nabla_X U(X, \boldsymbol{\theta})dt + \sqrt{2}dW_t,$$

$$\boldsymbol{x}_{t+1}^i - \boldsymbol{x}_t^i = -\alpha(\nabla_{\boldsymbol{x}} U(\boldsymbol{x}_t^i, \boldsymbol{\theta})) + \sqrt{2\alpha}\xi_t$$

A family of Memory Evolution Methods

Kernelized Method for Dynamic DRO (WGF-SVGD)



(a) $t = t_0$ (b) $t = t_1$ (c) $t = t_2$

$$\frac{dX}{dt} = -[\mathcal{K}_\mu \nabla \frac{\delta F}{\delta \mu}(\mu_t)](X) \qquad \mathcal{K}_\mu f(\boldsymbol{x}) = \int K(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') d\mu(\boldsymbol{x}')$$

$$\boldsymbol{x}_{t+1}^i - \boldsymbol{x}_t^i = -\frac{\alpha}{N} \sum_{j=1}^{j=N} [\underbrace{k(\boldsymbol{x}_t^i, \boldsymbol{x}_t^j) \nabla_{\boldsymbol{x}_t^j} U(\boldsymbol{x}_t^j, \boldsymbol{\theta})}_{\text{smoothed gradient}} + \underbrace{\nabla_{\boldsymbol{x}_t^j} k(\boldsymbol{x}_t^i, \boldsymbol{x}_t^j)}_{\text{repulsive term}}]$$

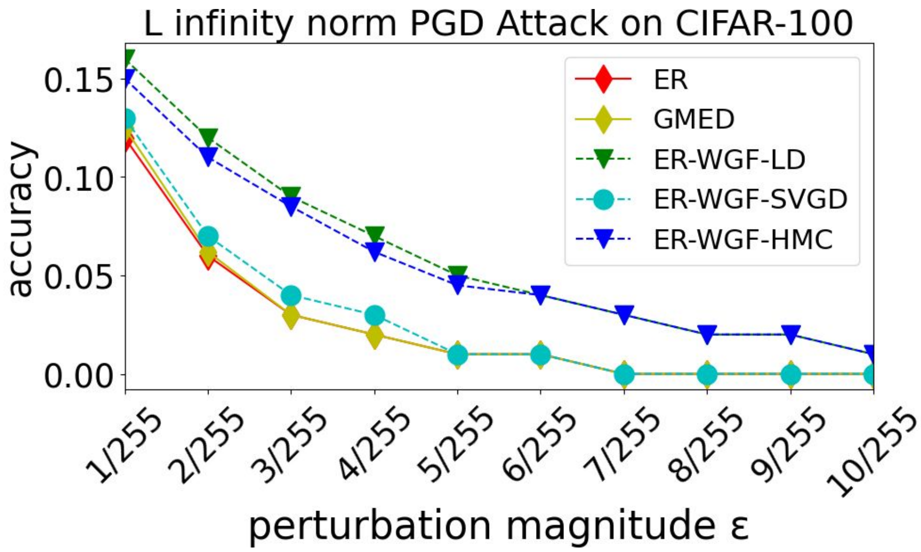
Experiment

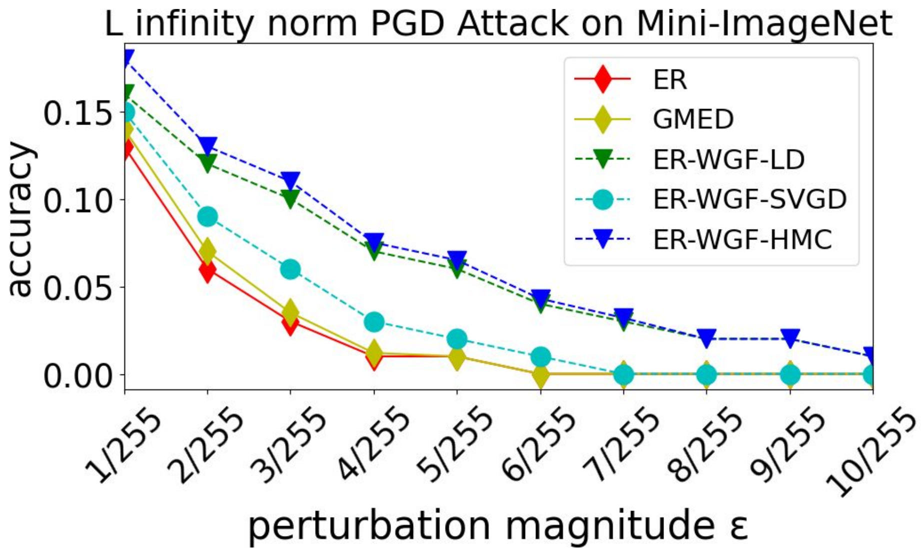
CIFAR10, CIFAR100, MiniImageNet
Split CIFAR10 into 5 tasks, each one consists of 2 classes
Split CIFAR100 and MiniImageNet into 20 tasks, each one consists of 5 classes

Algorithm	CIFAR10	CIFAR-100	MiniImagenet
fine-tuning	18.9 ± 0.1	3.1 ± 0.2	2.9 ± 0.5
A-GEM	19.0 ± 0.3	2.4 ± 0.2	3.0 ± 0.4
GSS-Greedy	29.9 ± 1.5	19.5 ± 1.3	17.4 ± 0.9
ER	33.3 ± 2.8	20.1 ± 1.2	24.8 ± 1.0
ER + WGF-LD	37.6 ± 1.5	$\mathbf{21.5 \pm 1.3}$	27.3 ± 1.0
ER + WGF-SVGD	36.5 ± 1.4	21.3 ± 1.5	$\mathbf{27.6 \pm 1.3}$
ER + WGF-HMC	$\mathbf{37.8 \pm 1.3}$	21.2 ± 1.4	27.2 ± 1.1
MIR	34.4 ± 2.5	20.0 ± 1.7	25.3 ± 1.7
MIR + WGF-LD	$\mathbf{38.2 \pm 1.2}$	$\mathbf{21.6 \pm 1.2}$	26.9 ± 1.0
MIR + WGF-SVGD	37.0 ± 1.4	21.2 ± 1.5	$\mathbf{27.4 \pm 1.2}$
MIR + WGF-HMC	37.9 ± 1.5	21.3 ± 1.4	27.1 ± 1.3
GMED (ER)	34.8 ± 2.2	20.9 ± 1.6	27.3 ± 1.8
GMED + WGF-LD	$\mathbf{38.4 \pm 1.6}$	21.7 ± 1.7	28.3 ± 1.9
GMED + WGF-SVGD	37.6 ± 1.7	$\mathbf{21.8 \pm 1.5}$	$\mathbf{28.7 \pm 1.5}$
GMED + WGF-HMC	37.8 ± 1.2	21.5 ± 1.9	28.4 ± 1.3
ER_{aug} + ER	46.3 ± 2.7	18.3 ± 1.9	30.8 ± 2.2
ER_{aug} + WGF-LD	47.6 ± 2.4	19.8 ± 2.2	31.9 ± 1.8
ER_{aug} + WGF-SVGD	$\mathbf{47.9 \pm 2.5}$	19.9 ± 2.3	$\mathbf{32.2 \pm 1.5}$
ER_{aug} + WGF-HMC	47.8 ± 2.6	$\mathbf{20.3 \pm 2.1}$	31.7 ± 2.0
iid online	60.3 ± 1.4	18.7 ± 1.2	17.7 ± 1.5
iid offline	78.7 ± 1.1	44.9 ± 1.5	39.8 ± 1.4

Experiment

As a by-product of the proposed framework, the methods are more robust to adversarial examples.





Thank you