Regularizing Second-Order Influences for Continual Learning

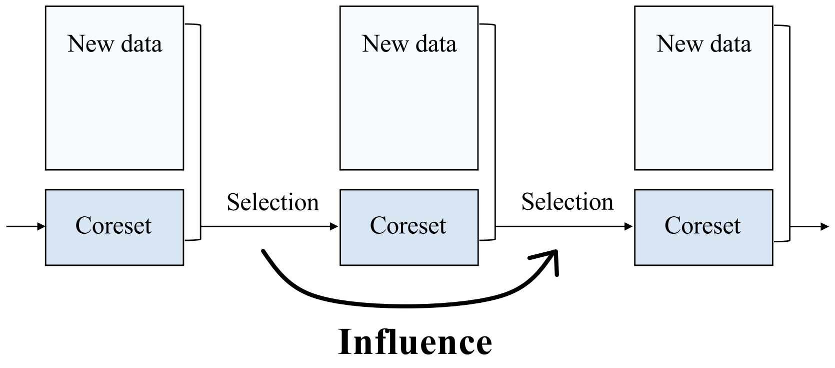
Zhicheng Sun[1], Yadong Mu[1,2*], Gang Hua[3]

[1]Peking University, [2]Peng Cheng Laboratory, [3]Wormpex AI Research

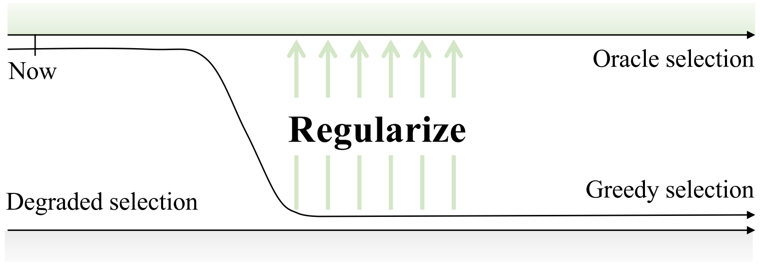
Quick Preview

- **Continual learning** aims to learn on long task sequences without catastrophic forgetting.
- **Replay-based methods** address this by rehearsing on a small replay buffer, which requires careful sample selection.
- However, existing strategies are designed for single-round selection, neglecting the **interactions** between selection steps.
- This work proposes to model the interactions with **influence functions** and address it via a regularized selection strategy.

(a)



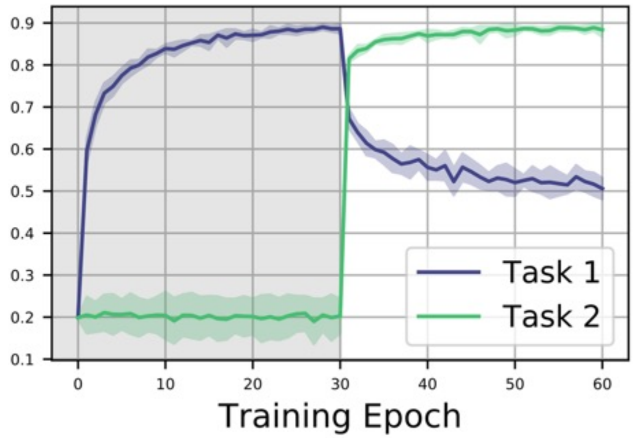
(b)



Introduction

Task description

- Continual learning[1] studies the training of models on long task sequences with potential data distribution shift.
- It is known for suffering from catastrophic forgetting[2], where the model abruptly forgets past knowledge after being updated on new tasks.

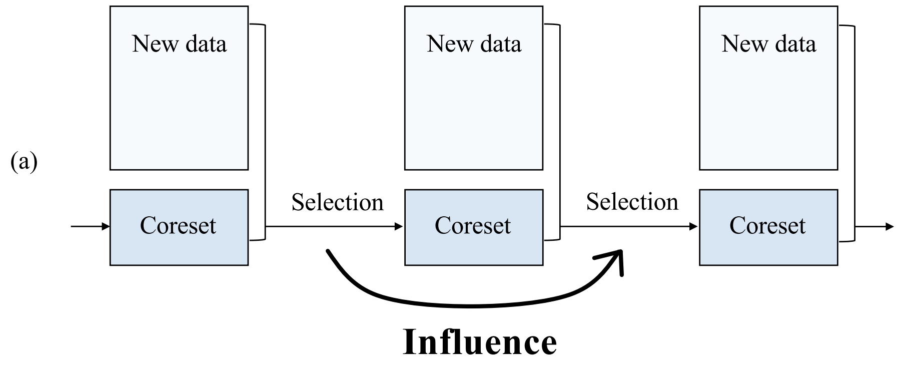


[1] Zhiyuan Chen and Bing Liu. Lifelong Machine Learning. Morgan & Claypool Publishers, 2018.
[2] Michael McCloskey and Neal J Cohen. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem". Psychology of Learning and Motivation, 1989, 24: 109–165.

Introduction

Motivation

- Replay-based approaches mitigate forgetting by rehearsing on a small replay buffer, which requires careful sample selection.



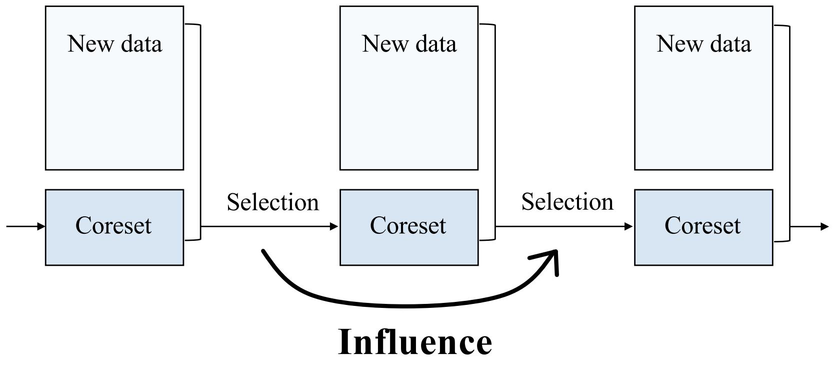
- However, existing selection strategies primarily focus on refining single-round performance, neglecting the interactions between consecutive selection steps through the data flow.

Introduction

Our contributions

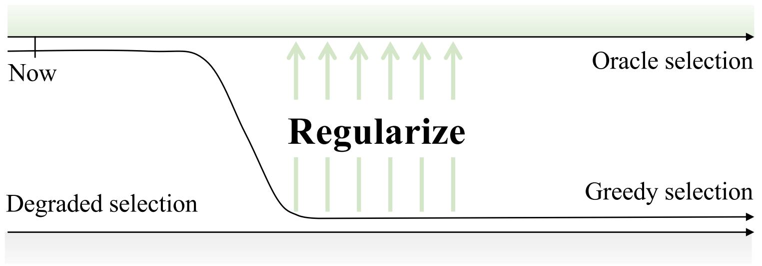
- We investigate the interaction between consecutive selection steps in continual learning and identify a new class of second-order influences.

(a)



- A novel regularizer is proposed to mitigate second-order influences, which also has clear connection to two other popular selection criteria.

(b)



Method

Problem formulation

- We consider learning on a data stream $\mathcal{Z}_{1:t} = \bigcup_{i=1}^{t} \mathcal{Z}_i$ with a small coreset $\mathcal{C}_t$. The sample selection goal is to preserve performance on $\mathcal{C}_{t-1} \cup \mathcal{Z}_t$ by replaying on $\mathcal{C}_t$:

$$\min_{\mathcal{C}_t \subset \mathcal{C}_{t-1} \cup \mathcal{Z}_t, |\mathcal{C}_t| \leq m} \sum_{z_i \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} L(z_i, \hat{\theta})$$

$$\text{s.t.} \quad \hat{\theta} = \arg\min_{\theta} \sum_{z_i \in \mathcal{C}_t} L(z_i, \theta).$$

- In the following, we will first present a greedy solution based on influence functions[1,2], then showcase its limitations and propose our improved version.

[1] Frank R Hampel. "The Influence Curve and Its Role in Robust Estimation". Journal of the American Statistical Association, 1974, 69(346): 383–393.
[2] Pang Wei Koh and Percy Liang. "Understanding Black-Box Predictions via Influence Functions". In: ICML. 2017: 1885–1894.

Method

Influence-based selection

- To solve the bilevel optimization problem, we linearly approximate the effect of selecting each sample z by perturbing its weight:

$$\hat{\theta}_{\epsilon,z} = \arg\min_{\theta} \sum_{z_i \in C_t} L(z_i, \theta) + \epsilon L(z, \theta).$$

- A classic result[1] gives the influence of upweighting z on the outer loss:

$$\begin{aligned}\mathcal{I}(z) &= \sum_{z_i \in C_{t-1} \cup \mathcal{Z}_t} \frac{dL(z_i, \hat{\theta}_{\epsilon,z})}{d\epsilon}\Big|_{\epsilon=0} \\ &= -\sum_{z_i \in C_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z_i, \hat{\theta}_t)^\top H_{\hat{\theta}_t}^{-1} \nabla_\theta L(z, \hat{\theta}_t).\end{aligned}$$

- It further yields an optimal solution that greedily select the most influential samples.

[1] R Dennis Cook and Sanford Weisberg. Residuals and Influence in Regression. New York: Chapman and Hall, 1982.

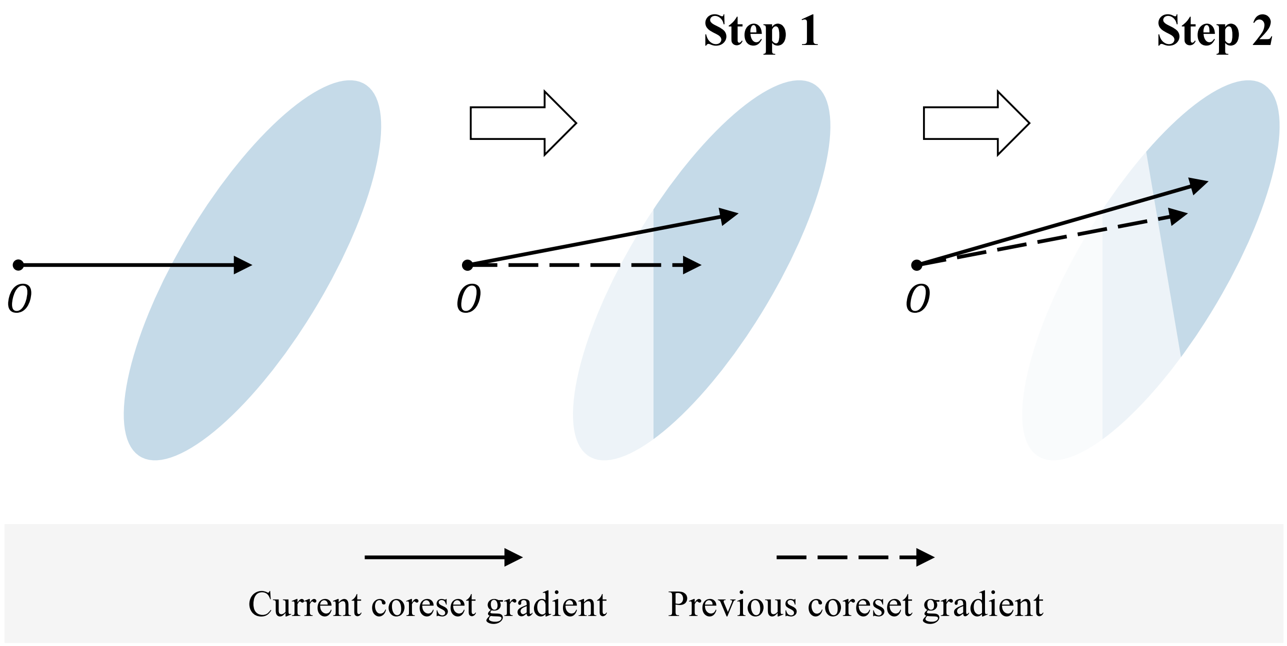
Method

Second-order influences

- This greedy selection strategy favors samples that are more similar to the existing ones.

$$\mathcal{I}(z) = -\sum_{z_i \in \mathcal{C}_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z_i, \hat{\theta}_t)^\top H_{\hat{\theta}_t}^{-1} \nabla_\theta L(z, \hat{\theta}_t).$$

- Due to second-order effects, it would result in a biased and less diversified coreset:



Method

Second-order influences

- To model such an effect, we upweight two samples z and z' from consecutive selection steps. Upweighting the previous sample interferes with the subsequent selection:

- If z and z' are not jointly optimized in the next round:

$$\mathcal{I}_{\epsilon,z}(z') = -\left(\sum_{z_i \in \mathcal{C}_t \cup \mathcal{Z}_{t+1}} \nabla_\theta L(z_i, \hat{\theta}_{t+1}) + \epsilon \nabla_\theta L(z, \hat{\theta}_{t+1})\right)^\top H_{\hat{\theta}_{t+1}}^{-1} \nabla_\theta L(z', \hat{\theta}_{t+1}).$$

$$\mathcal{I}^{(2)}(z, z') = -\nabla_\theta L(z, \hat{\theta}_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1} \nabla_\theta L(z', \hat{\theta}_{t+1}).$$

- If z and z' are jointly optimized in the next round:

$$\mathcal{I}_{\epsilon,z}(z') = -\left(\sum_{z_i \in \mathcal{C}_t \cup \mathcal{Z}_{t+1}} \nabla_\theta L(z_i, \hat{\theta}_{t+1}) + \epsilon \nabla_\theta L(z, \hat{\theta}_{t+1})\right)^\top (H_{\hat{\theta}_{t+1}} + \epsilon H_{\hat{\theta}_{t+1},z})^{-1} \nabla_\theta L(z', \hat{\theta}_{t+1}).$$

$$\mathcal{I}^{(2)}(z, z') = -(\nabla_\theta L(z, \hat{\theta}_{t+1}) - H_{\hat{\theta}_{t+1},z} s_{t+1})^\top H_{\hat{\theta}_{t+1}}^{-1} \nabla_\theta L(z', \hat{\theta}_{t+1}).$$

Method

Regularizing influences

- The total interference is a weighted sum of the two second-order influences:

$$\Delta\mathcal{I}(z') \approx -\sum_{z\in\overline{C}_t} \mathcal{I}^{(2)}(z,z')\cdot 1$$
$$= \sum_{z\in\overline{C}_t} (\nabla_\theta L(z,\hat{\theta}_{t+1}) - \mu H_{\hat{\theta}_{t+1},z} s_{t+1})^T H^{-1}_{\hat{\theta}_{t+1}} \nabla_\theta L(z',\hat{\theta}_{t+1}).$$

- Its magnitude can be upper-bounded with the following regularizer:

$$|\Delta\mathcal{I}(z')| \le \left\| \sum_{z\in\overline{C}_t} (\nabla_\theta L(z,\hat{\theta}_{t+1}) - \mu H_{\hat{\theta}_{t+1},z} s_{t+1}) \right\| \times \left\| H^{-1}_{\hat{\theta}_{t+1}} \nabla_\theta L(z',\hat{\theta}_{t+1}) \right\|,$$

$$\mathcal{R}(C_t) = \left\| \sum_{z\in\overline{C}_t} (\nabla_\theta L(z,\hat{\theta}_t) - \mu H_{\hat{\theta}_t,z} s_t) \right\|$$

- This regularizer is used in the final selection criterion: minimize $\sum_{z\in C_t} \mathcal{I}(z) + \nu\mathcal{R}(C_t)$.

Method

Interpreting the regularizer

$$\mathcal{R}(C_t) = \left\| \sum_{z \in \overline{C}_t} (\nabla_\theta L(z, \hat{\theta}_t) - \mu H_{\hat{\theta}_t, z} s_t) \right\|$$

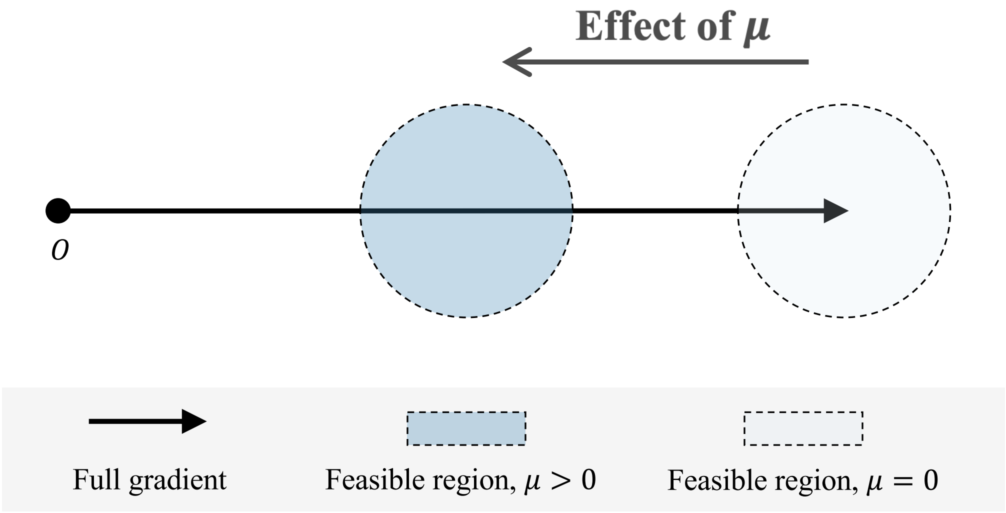
- $\mu = 0 \Rightarrow$ gradient matching[1]:

$$\mathcal{R}(C_t) = \left\| \sum_{z \in C_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in C_t} \nabla_\theta L(z, \hat{\theta}_t) \right\|.$$

- $\mu > 0$, identical Hessian $\Rightarrow$ diversity[2]:

$$\mathcal{R}(C_t) = \left\| (1 - \alpha\mu) \sum_{z \in C_{t-1} \cup \mathcal{Z}_t} \nabla_\theta L(z, \hat{\theta}_t) - \sum_{z \in C_t} \nabla_\theta L(z, \hat{\theta}_t) \right\|,$$

- additional Hessian-related information



[1] Bo Zhao, Konda Reddy Mopuri and Hakan Bilen. "Dataset Condensation with Gradient Matching". In: ICLR. 2021.
[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud et al. "Gradient based Sample Selection for Online Continual Learning". In: NeurIPS. 2019: 11817–11826.

Experiments

- Comparison to state-of-the-art methods on Split CIFAR-10

Method		Class-incremental				Task-incremental			
		m = 300		m = 500		m = 300		m = 500	
		ACC (%)	BWT (%)	ACC (%)	BWT (%)	ACC (%)	BWT (%)	ACC (%)	BWT (%)
Non-IF	GEM[12]	37.51	-70.48	36.95	-69.76	89.34	-9.09	90.42	-7.88
	A-GEM[93]	20.02	-95.68	20.01	-95.69	85.52	-14.07	86.45	-12.83
	ER[89]	34.19	-78.18	40.45	-70.36	88.97	-9.95	90.60	-7.74
	GSS[22]	35.89	-75.80	41.96	-68.24	88.05	-10.63	90.38	-7.73
	ER-MIR[94]	38.53	-72.72	42.65	-67.50	88.50	-10.33	90.63	-7.62
	GDUMB[21]	36.92	-	44.27	-	73.22	-	78.06	-
	HAL[95]	24.45	-83.56	27.94	-80.01	79.90	-14.39	81.84	-12.73
	GMED[96]	38.12	-73.16	43.68	-66.21	88.91	-9.76	89.72	-8.75
IF	Vanilla IF	41.76	-68.59	47.14	-62.20	90.67	-7.65	91.06	-7.36
	MetaSP[36]	43.76	-66.37	50.10	-58.39	89.91	-9.00	91.41	-7.36
	Ours	**48.62**	**-60.24**	**53.07**	**-54.44**	**91.52**	**-6.94**	**92.53**	**-5.46**

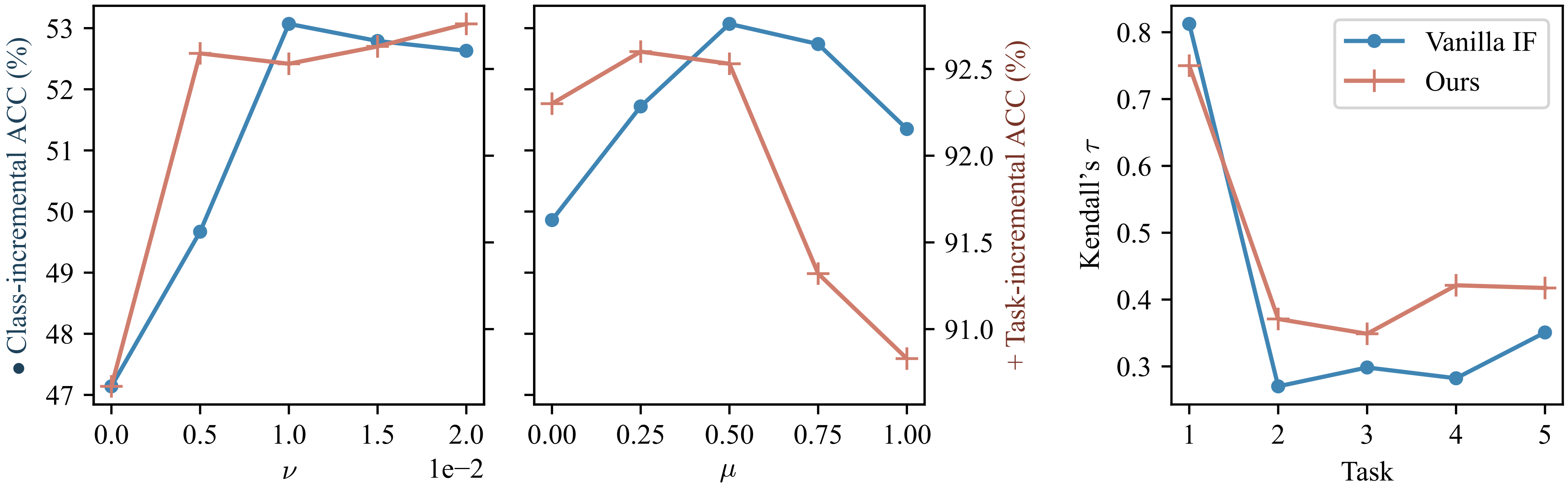
Experiments

- Comparison to state-of-the-art methods on Split *mini*ImageNet

Method		Class-incremental				Task-incremental			
		m = 500		m = 1000		m = 500		m = 1000	
		ACC (%)	BWT (%)	ACC (%)	BWT (%)	ACC (%)	BWT (%)	ACC (%)	BWT (%)
Non-IF	A-GEM[93]	10.69	-49.22	10.69	-49.16	18.34	-39.65	18.78	-39.05
	ER[89]	11.00	-50.84	11.35	-50.08	28.97	-28.40	31.59	-24.95
	GSS[22]	11.09	-50.66	11.42	-49.91	28.67	-28.71	31.75	-24.56
	ER-MIR[94]	11.07	-50.46	11.32	-49.92	29.10	-27.95	31.39	-24.89
	GDUMB[21]	6.22	-	7.15	-	16.37	-	17.69	-
	GMED[96]	11.03	-50.23	11.73	-48.93	30.47	-26.02	32.85	-22.69
IF	Vanilla IF	12.08	-48.55	14.64	-47.15	33.74	-21.71	37.55	-19.28
	MetaSP[36]	12.74	-48.84	14.54	-45.52	34.36	-21.70	37.20	-17.83
	Ours	**13.63**	**-47.94**	**16.15**	**-43.78**	**36.46**	**-19.48**	**39.61**	**-16.01**

Experiments

- Ablation studies of hyperparameter sensitivity and influence estimation accuracy



Thanks for listening

Code is available at https://github.com/feifeiobama/InfluenceCL