Diffusion-Driven Data Replay: A Novel Approach to Combat Forgetting in Federated Class Continual Learning

Jinglin Liang, Jin Zhong, Hanlin Gu, Zhongqi Lu, Xingxing Tang,

Gang Dai, Shuangping Huang, Lixin Fan, Qiang Yang

South China University of Technology, The Hong Kong University of Science and Technology,

China University of Petroleum, WeBank, Pazhou Laboratory

Contents

Federated Class Continual Learning

- **Federated Learning** enables decentralized learning while **protecting data privacy,** making it essential in fields like healthcare and finance.
- Real-world Federated Learning faces challenges like **new data classes from clients** and **varying participants**.

drives the development

Federated Class Continual Learning (FCCL)

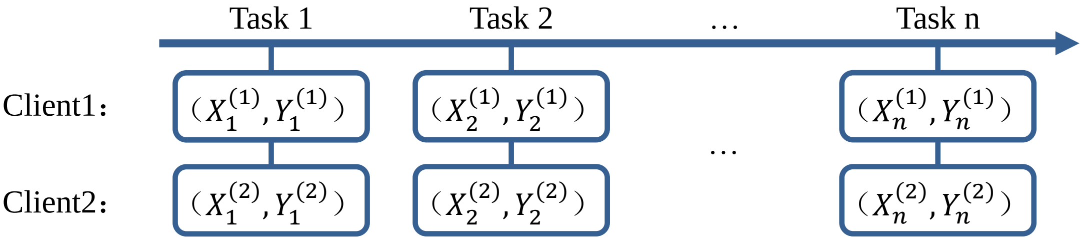
A novel concept requiring models to learn new classes in federated training without forgetting prior knowledge.

Federated Class Continual Learning

Problem definition:

- Multiple clients **collaboratively learn** a sequence of tasks with **data distributed locally**, and **no class overlap between tasks**.

$$Y_i^{(*)} \cap Y_j^{(*)} = \emptyset, \forall i, j$$



Challenges

Challengs

- **Catastrophic Forgetting**: forgetting old knowledge when learning new tasks.
- **Stricter privacy protection** in federated settings makes **some continual learning methods, like experience replay, unsuitable.**

Limitation of existing methods

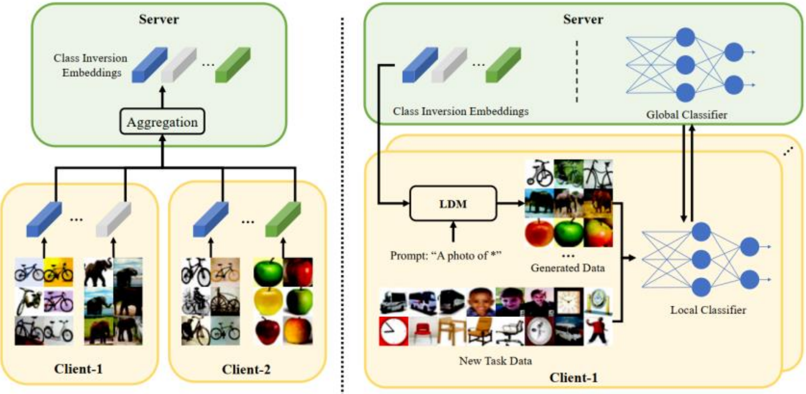
- Existing FCCL methods use GANs or Data-free knowledge distillation for data replay.
  - **Low generation quality** (GANs, Data-free knowledge distillation)
  - **Unstable training**, especially in federated settings (GANs)

Contents

Overview

To address these issues, we propose DDDR with two phases:

- **Federated Class Inversion:** Obtain class embeddings for subsequent generative replay.
- **Replay-Augmented Training**: Train the classifier with real and generated data.

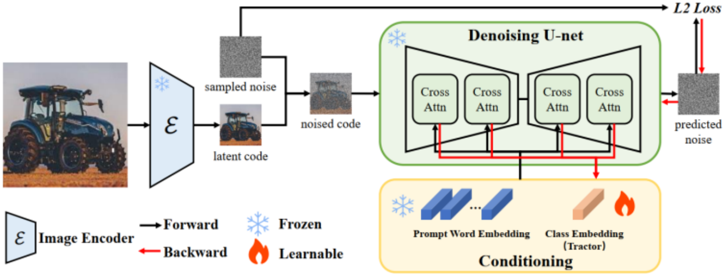


(a) Federated Class Inversion Phase (b) Replay-Augmented Training Phase

Federated Class Inversion Phase

A naive idea: **Train a diffusion model for each task for subsequent generative replay.** —but→ **costly**

- Inspired by personalized generation models, we propose **Federated Class Inversion**, searching the conditional space of a pretrained conditional diffusion model to find conditions that guide the model to generate specific classes as class embeddings.
- We use **FedAvg** to aggregate class embeddings from different clients.



Replay-Augmented Training Phase

$(\mathcal{X}_c, \mathcal{Y}_c)$
Real data
for the current task.

$(\widehat{\mathcal{X}}_c, \widehat{\mathcal{Y}}_c)$
Generated data
for the current task.

$(\widehat{\mathcal{X}}_p, \widehat{\mathcal{Y}}_p)$
Generated data
for the previous task.

$$\mathcal{L}_{CE} = \mathbb{E}_{x \sim \mathcal{X}_c \cup \widehat{\mathcal{X}}_c, y \sim \mathcal{Y}_c \cup \widehat{\mathcal{Y}}_c}[CE(\mathcal{F}(x), y)]$$

$$\mathcal{L}_{PCE} = \mathbb{E}_{x \sim \widehat{\mathcal{X}}_p, y \sim \widehat{\mathcal{Y}}_p}[CE(\mathcal{F}(x), y)]$$

$$\mathcal{L}_{KD} = \mathbb{E}_{x \sim \widehat{\mathcal{X}}_p}[KL(\mathcal{F}(x), \mathcal{F}'(x))]$$

Three common loss functions.

A domain gap exists between generated and real data, and only generated data is available for past tasks.

Contrastive learning
Enhance Generalizability in real and generated domains.

$$\mathcal{L}_{SCL} = \mathbb{E}_{e_i \sim \mathcal{F}_e(\mathcal{X}_c \cup \widehat{\mathcal{X}}_c), e_p \sim P(e_i)}\left[log \frac{\exp\left(sim\left(e_i, e_p\right)/\tau\right)}{\sum_{i \neq j} \exp\left(sim\left(e_i, e_j\right)/\tau\right)}\right]$$

Contents

Main Result

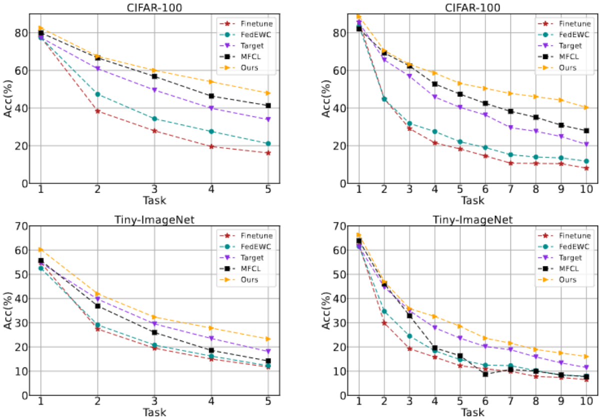
- DDDR significantly outperforms existing FCCL methods across all settings (task numbers and non-IID) for both datasets.

Cifar-100

Data partition	IID				non-IID			
Tasks	T=5		T=10		T=5		T=10	
Method	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)
Finetune	17.41	0.82	8.76	0.87	16.15	0.79	8.08	0.85
FedEWC	21.02	0.69	10.61	0.75	21.16	0.69	11.79	0.75
Target	35.08	0.47	23.69	0.48	33.89	0.47	20.67	0.54
MFCL	40.39	0.40	32.47	0.45	41.34	0.32	27.97	0.43
Ours	**51.15**	**0.29**	**43.76**	**0.29**	**47.93**	**0.25**	**40.33**	**0.25**

Tiny-ImageNet

Data partition	IID				non-IID			
Tasks	T=5		T=10		T=5		T=10	
Method	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)	Acc(↑)	FM(↓)
Finetune	12.12	0.61	6.77	0.68	11.73	0.57	6.51	0.63
FedEWC	13.00	0.50	8.42	0.56	12.25	0.48	7.95	0.51
Target	18.03	0.45	11.59	0.50	18.11	0.41	11.49	0.35
MFCL	14.79	0.53	10.31	0.53	14.23	0.50	7.72	0.51
Ours	**25.77**	**0.35**	**19.23**	**0.35**	**23.30**	**0.34**	**16.05**	**0.27**



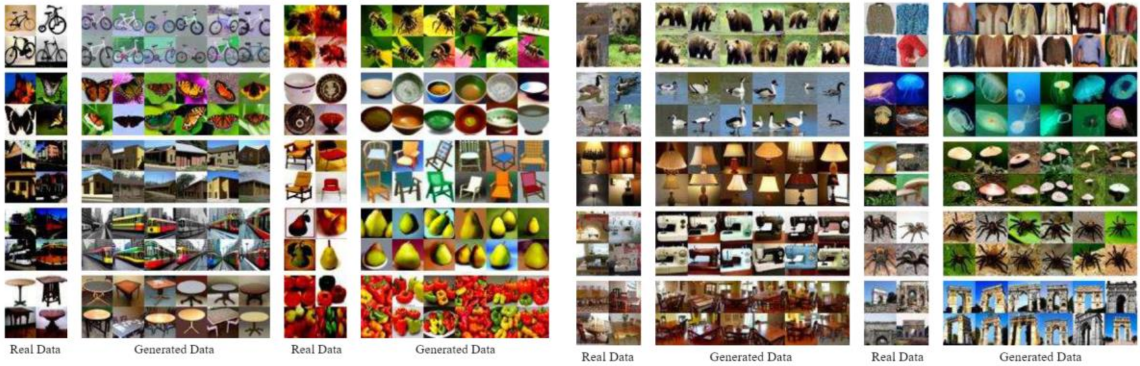
Ablation Study

- All modules contributed significant improvements to DDDR.
- $\mathcal{L}_{SCL}$ only brings improvement when used with $(\widehat{\mathcal{X}}_c, \widehat{\mathcal{Y}}_c)$ (generated data for the current task), otherwise, it can have a negative effect.

	$(\hat{\mathcal{X}}_p, \hat{\mathcal{Y}}_p)$	$(\hat{\mathcal{X}}_c, \hat{\mathcal{Y}}_c)$	$\mathcal{L}_{SCL}$	Acc(↑)	FM(↓)
1	✓	✓	✓	47.93	0.25
2	✗	✓	✓	17.13	0.84
3	✓	✗	✓	43.96	0.34
4	✓	✓	✗	44.96	0.27
5	✗	✗	✓	15.99	0.82
6	✓	✗	✗	44.77	0.28
7	✗	✓	✗	17.12	0.82
8	✗	✗	✗	16.15	0.79

Visualization of Generated Results

- Federated Class Inversion generates images that are close to the distribution of real images.



Contents

Conclusion

Contributions

- We propose DDDR, an innovative FCCL framework. This marks **the first application of employing the diffusion model to reproduce data in FCCL**, effectively mitigating catastrophic forgetting.
- We propose **Federated Class Inversion**, achieving high-quality data generation in federated settings without consuming excessive additional resources.
- By incorporating **contrastive learning**, we enhance the generalization ability of classifiers across generated and real domains, further strengthening the representational capacity of generated data towards real data.
- Comprehensive experiments across various datasets demonstrate that our approach significantly outperforms existing methods, **establishing a new state-of-the-art (SOTA) benchmark for FCCL.**

The End

Code is available at: https://github.com/jinglin-liang/DDDR