

1. 자원의 효율성

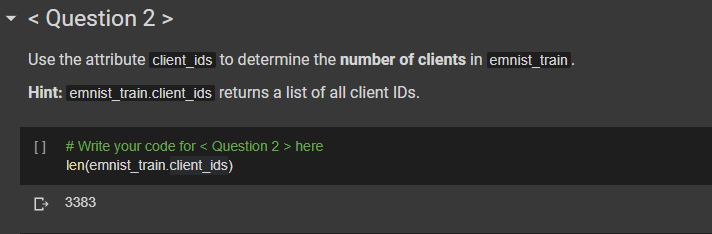
Local learning을 통한 1 server의 과부하 방지, local -> aggregation을 통한 model learning perfomance 향상 기대

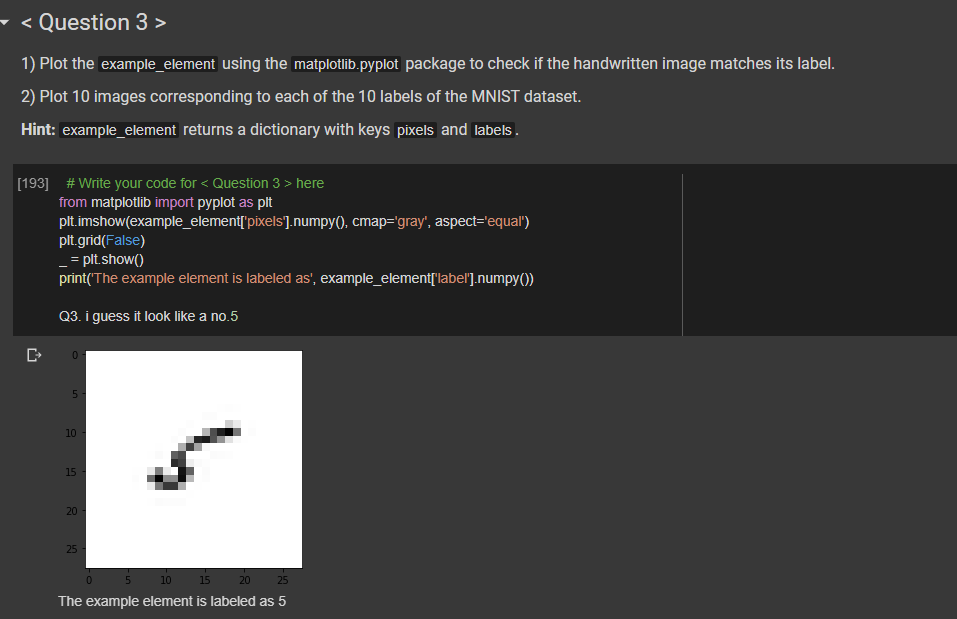
1. 데이터 priavacy보장

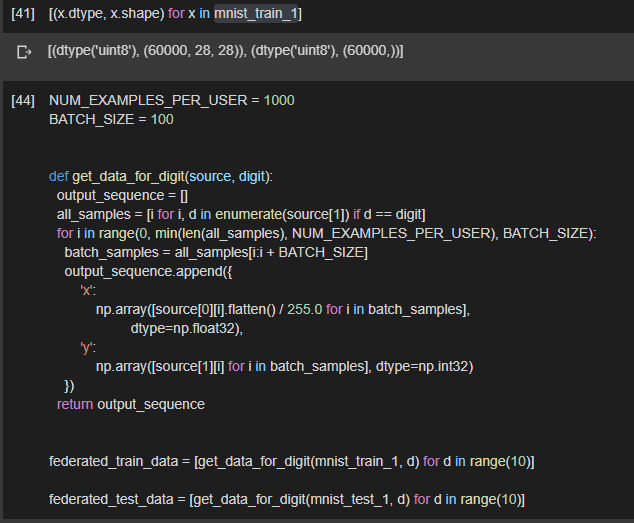
Tien-Dung Cao et al.[[1]](#footnote-1)에 따르면 딥러닝은 오늘날 눈부신 발전을 이뤘지만, 아직도 computational cost의 문제나 data privacy의 보장성에대한 문제를 해결하지 못했다고 이야기한다. 특히 medical data or healthcare data등 민감성 정보에 대해서는 개인정보보호에 취약성을 갖고있다.

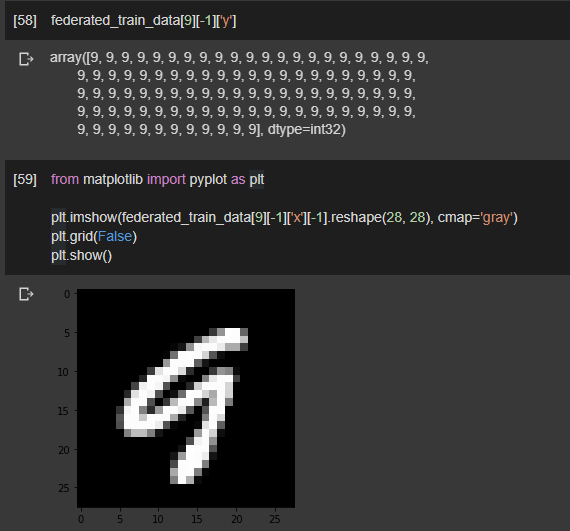
1. 결론: FTT learning (프레임워크 개발을 통해서)은 centeralized learning대비 4.8배 이상 빠르게 처리할 수 있다.

We formally prove the convergence of the learning model when training with the developed framework and its privacy-preserving property. We carry out extensive experiments to evaluate the performance of the framework in terms of speedup ratio, the approximation to the upper-bound performance (when training centrally) and communication overhead between the master and training workers. The results show that the **developed framework achieves a speedup of up to 4.8× compared to the centralized training approach and maintaining the performance approximation of the models within 4.5% of the centrally-trained models**[[2]](#footnote-2)

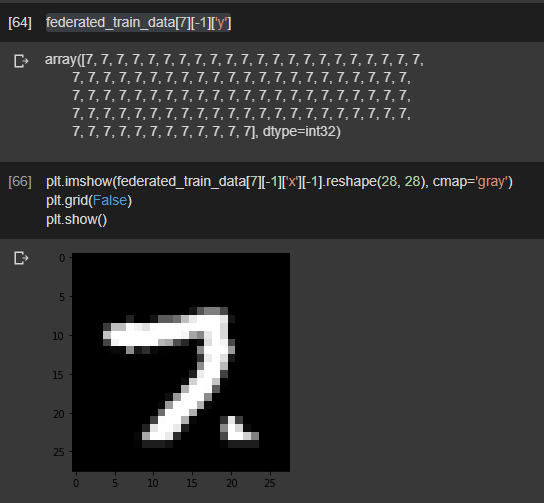


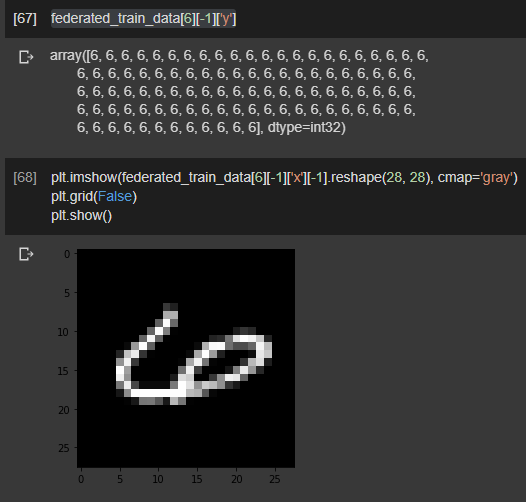






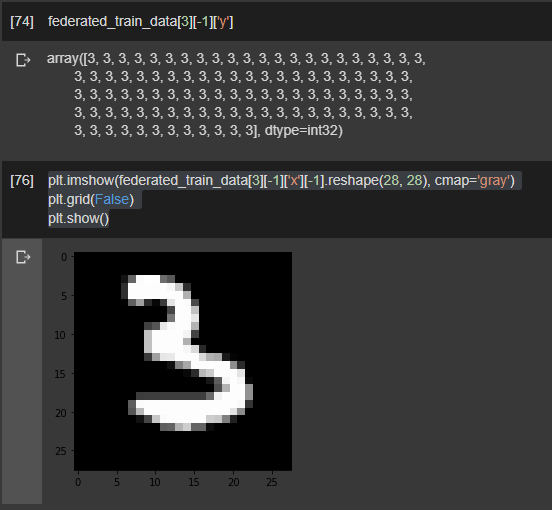


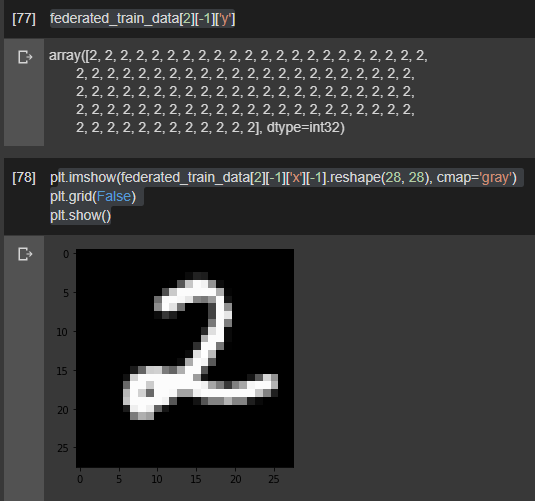


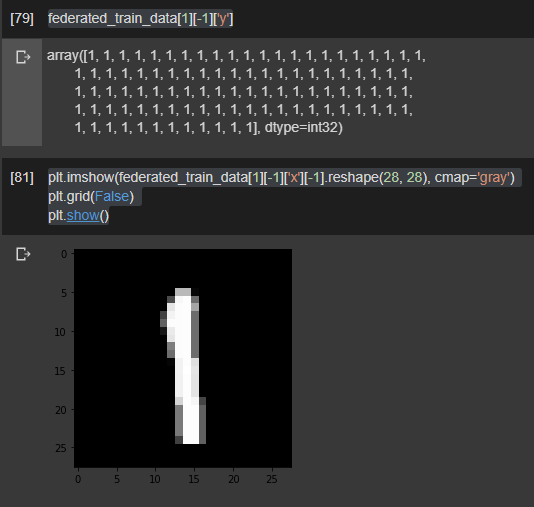




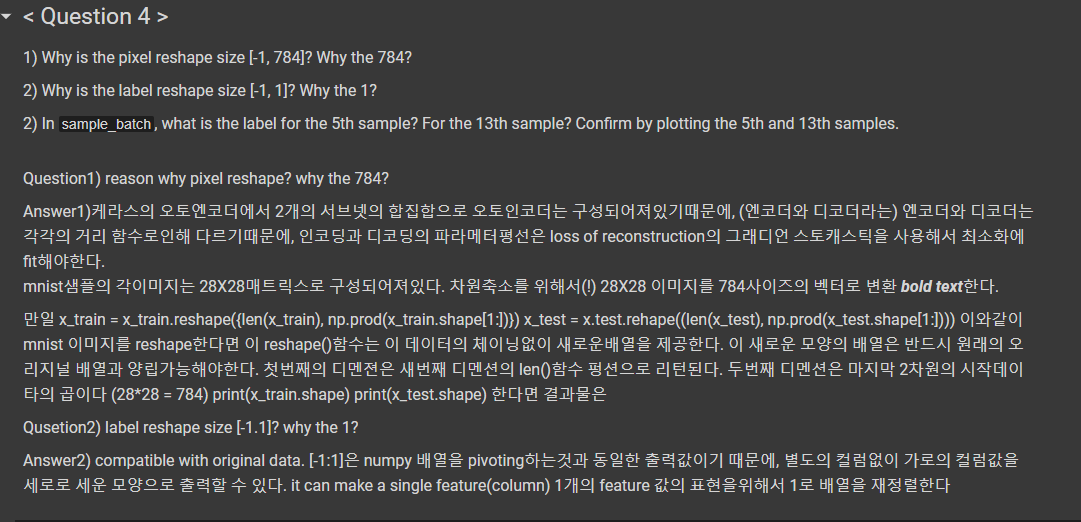


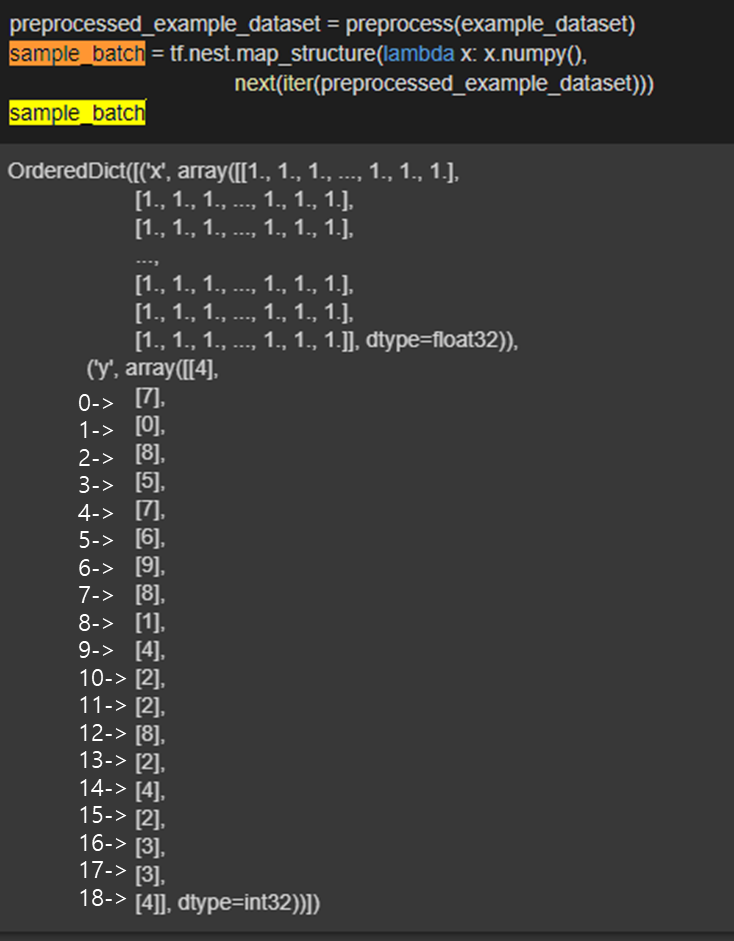


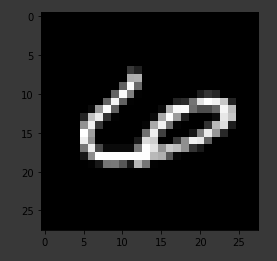


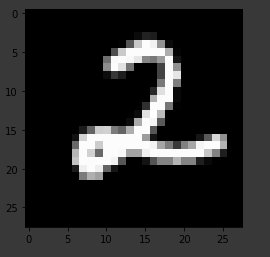


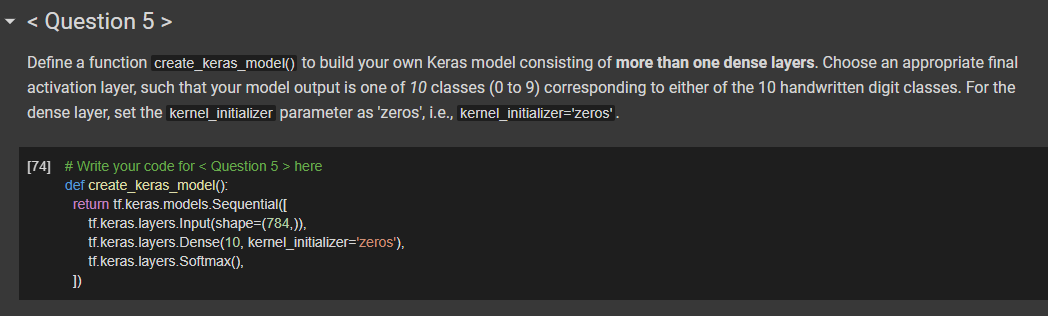


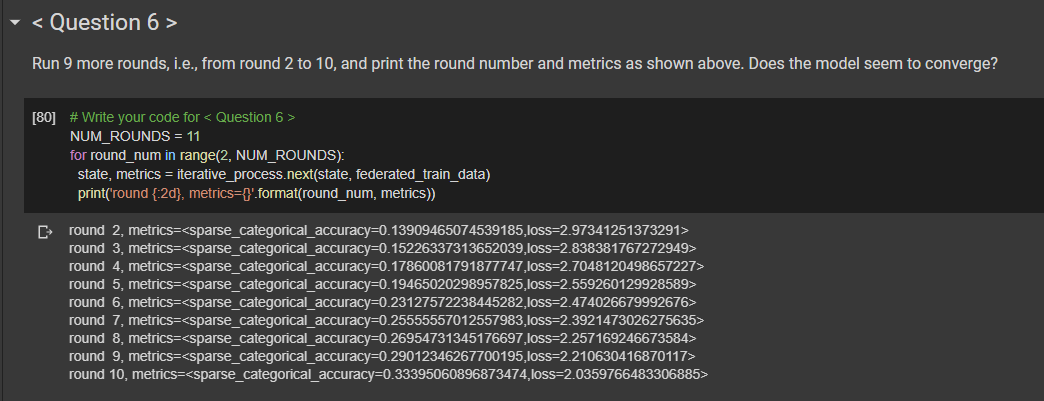




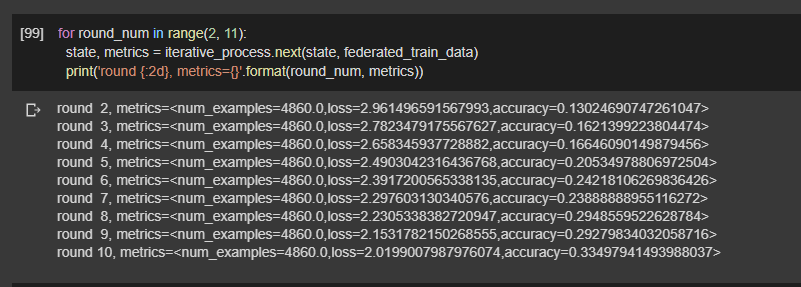


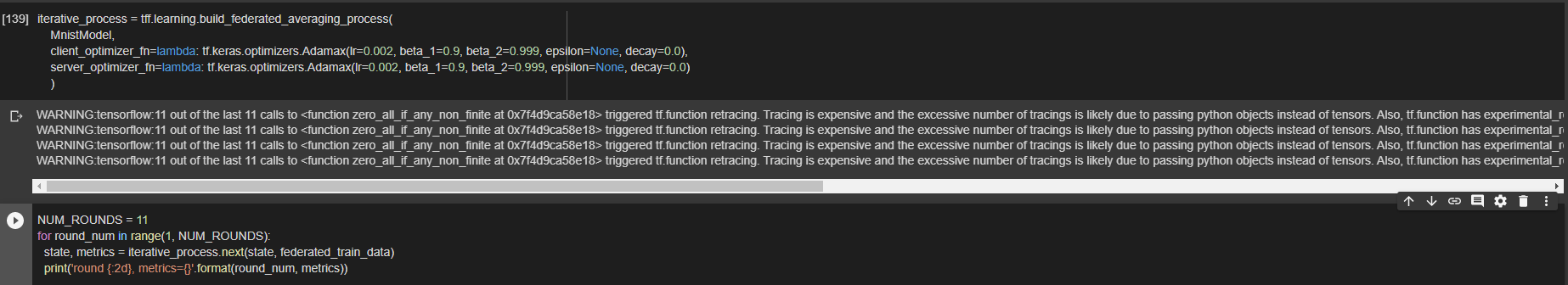






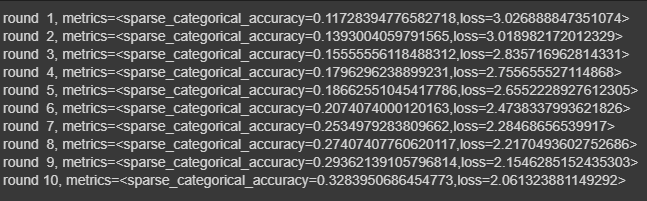




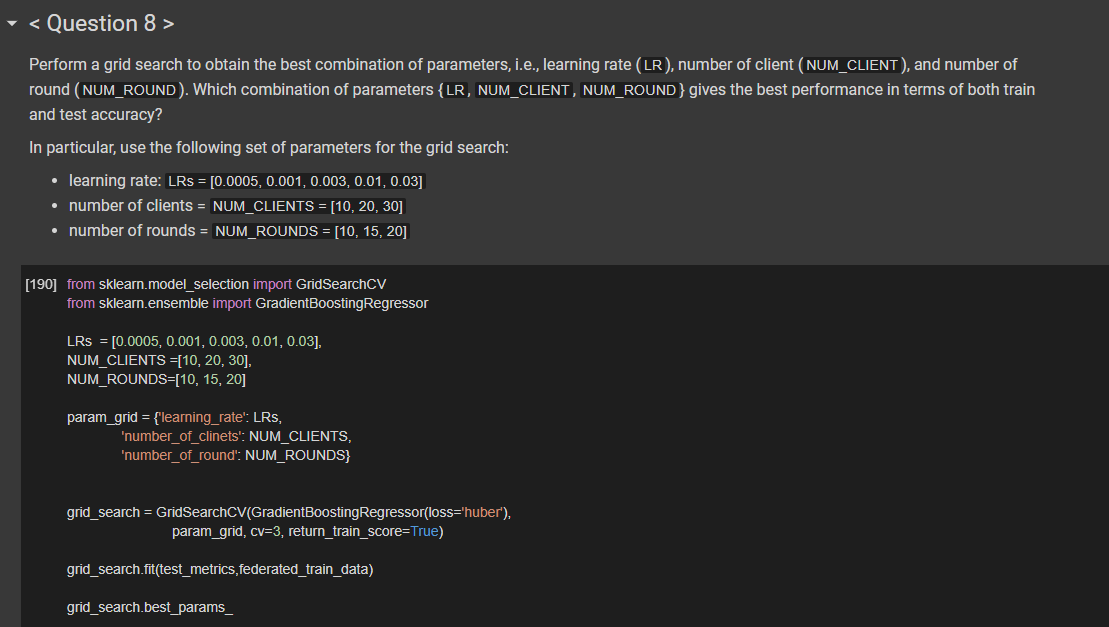


Keras.optimizer => Adam, Adamax 등으로 변경해봤는데 Warning tensorflow 오류발생..

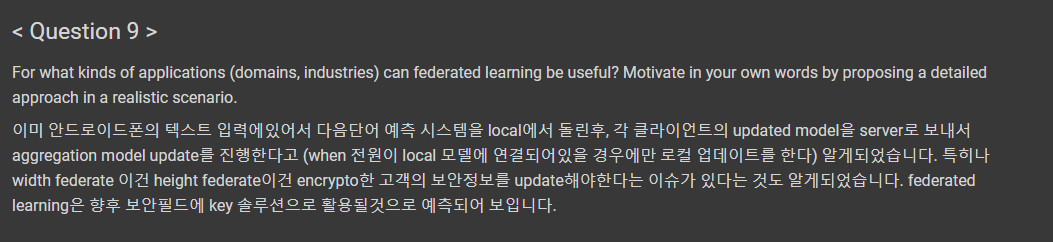
Loss, accuracy 구하는 def 정의해서 (create\_tff\_model) Num\_rounds = 11로 돌려서 계속 돌릴수록 loss = 2.06까지 떨구고, sparse\_categorical\_accuracy = 0.32까지올리는것으로 대체함.



FTT는 경사하강법 optimizer밖에 적용아 안되는지? Tensorflow 튜토리얼 예제를 보아도 이외의 keras optimizer를 사용한 예를 확인하지 못해서 궁금합니다..



Number of rounds = 10, number of clients = 30, LRs = 0.01 일때 best perfomance를 산출해낸다.



Any busineees ‘per-user’ domain, has got importance of each user’s privacy (like medical, healthcare) with high privacy. [[3]](#footnote-3) 가장 실용적인 예는 역시 핸드폰의 다음 텍스트를 예측하는 알고리즘으로 각각 local model leaning을 수행한후 최종적인 하나의 서버에서 각 로컬의 업데이트 모델을 수용하고 재 배포하는 시나리오를 예로 들 수 있을 것입니다.

1. Tien-Dung Cao et al. A Dederated Learning Framework for Privacy-preserving and Parallel Training(2020) [↑](#footnote-ref-1)
2. 1번 동일 [↑](#footnote-ref-2)
3. Daniel W.perterson et al.(2019) Private Federated Learning with Domain Adaptaion [↑](#footnote-ref-3)