清華大學 智能产业研究院
Institute for AI Industry Research, Tsinghua University

VFLAIR: A RESEARCH LIBRARY AND BENCHMARK FOR VERTICAL FEDERATED LEARNING

Tianyuan Zou, Zixuan Gu, Yu He, Hideaki Takahashi, Yang Liu*, Ya-Qin Zhang

* Corresponding Author: Yang Liu (liuy03@air.tsinghua.edu.cn).

2024-4-18

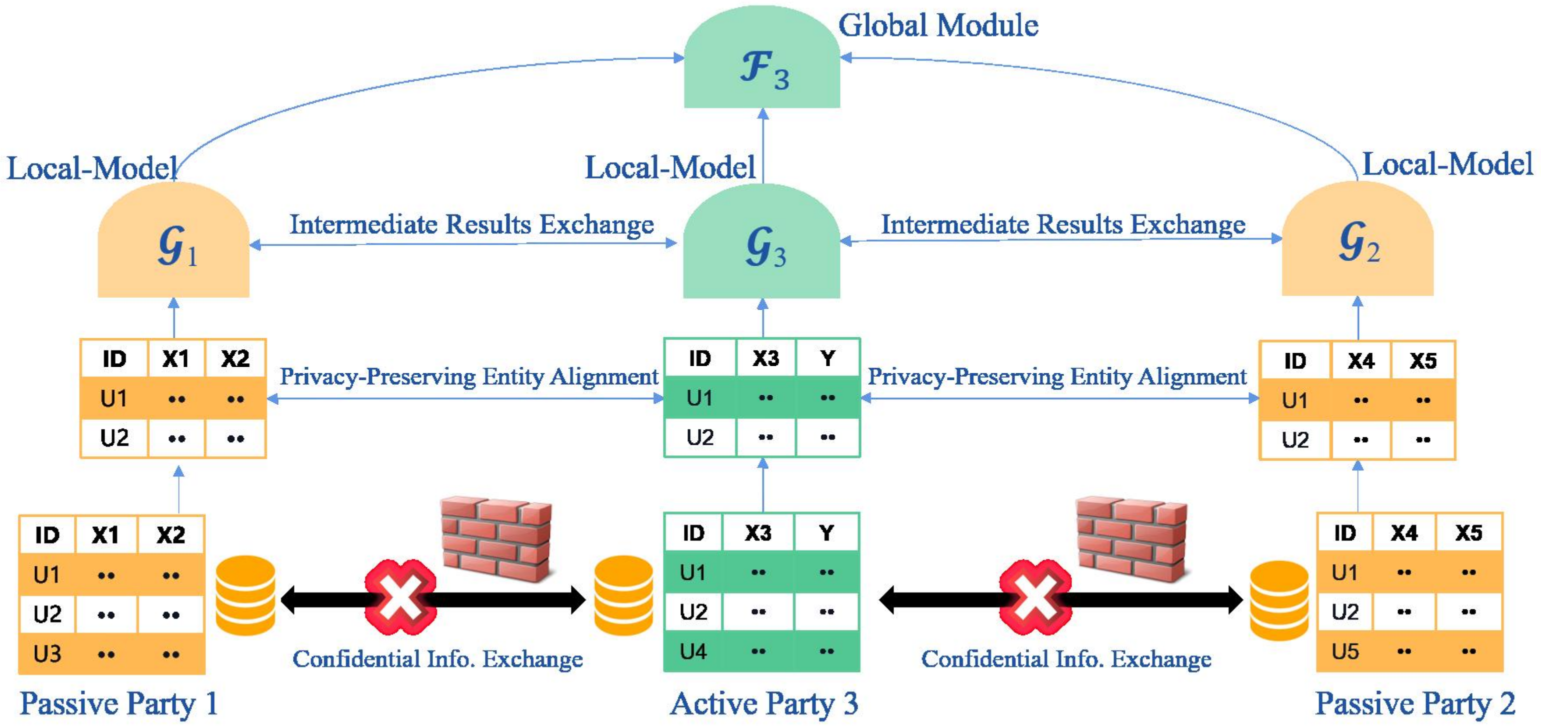
Outline

清華大學 智能产业研究院
Institute for AI Industry Research, Tsinghua University

- VFL Background
- VFLAIR Design
- VFLAIR Highlights
  - Comprehensive Evaluation of VFL Settings
  - Comprehensive Evaluation of 11 Attacks and 8 Defenses
  - Novel Evaluation Metric: Defense Capability Score
  - Additional Insights
- Comprehensive User Guidance and Documentation

Background: Vertical Federated Learning (VFL)

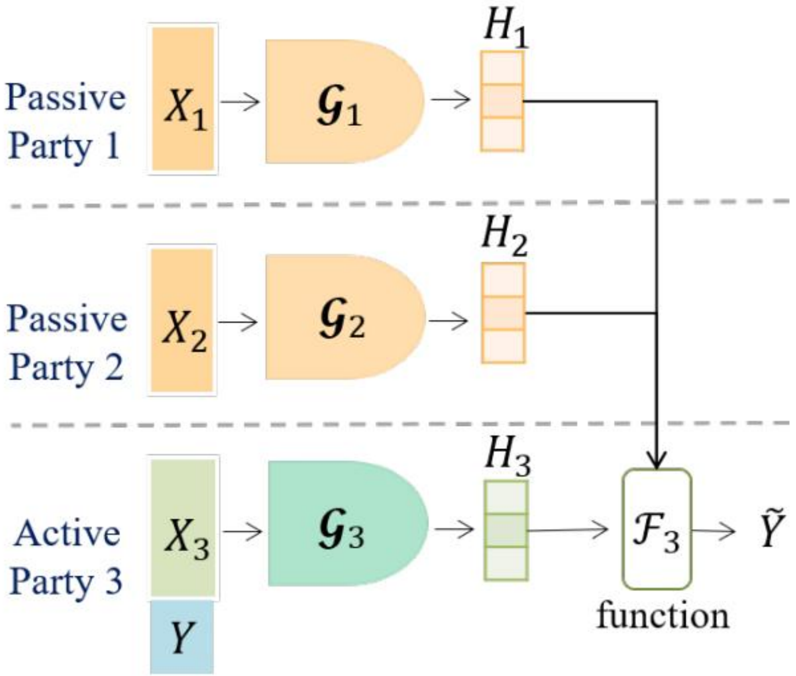
- In VFL, each of the K participating parties keeps its **private data** X_k and **private model** G_k local but exchanges intermediate computed results, including **local model outputs** H_k and their **gradients**. The only party that controls the private label information (active party) additionally controls the **global model** F_K. [1]
  - After training, each party in the VFL owns the separate **private local model** G_k.
  - During inference, parties in VFL collaborate to make inferences.



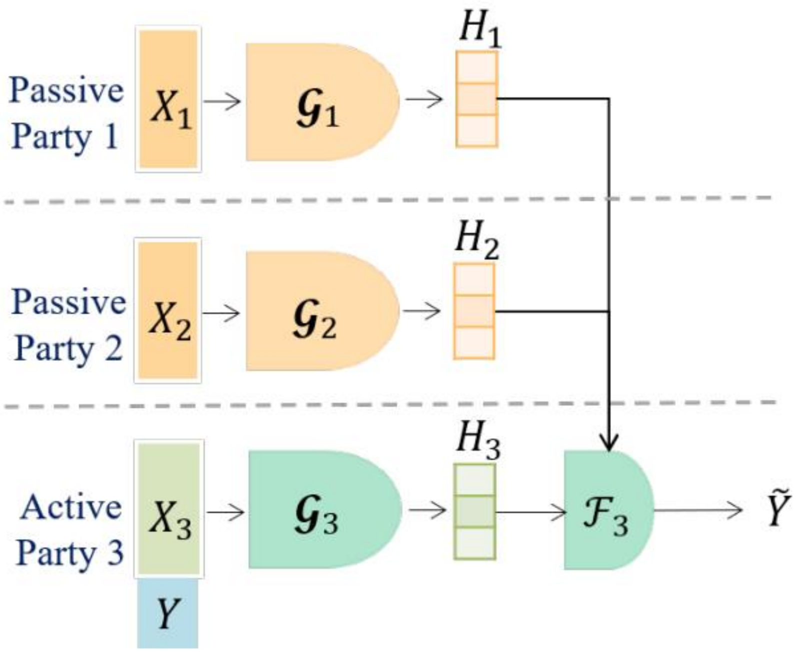
[1] Y. Liu et al. Vertical Federated Learning: Concepts, Advances, and Challenges. IEEE Transactions on Knowledge and Data Engineering, 2024.

Background: Vertical Federated Learning (VFL)

- Depending on how the model is partitioned among active and passive parties, VFL can be further divided into **aggVFL** and **splitVFL** in which a non-trainable global function or a trainable global model is used at the active party. [1]



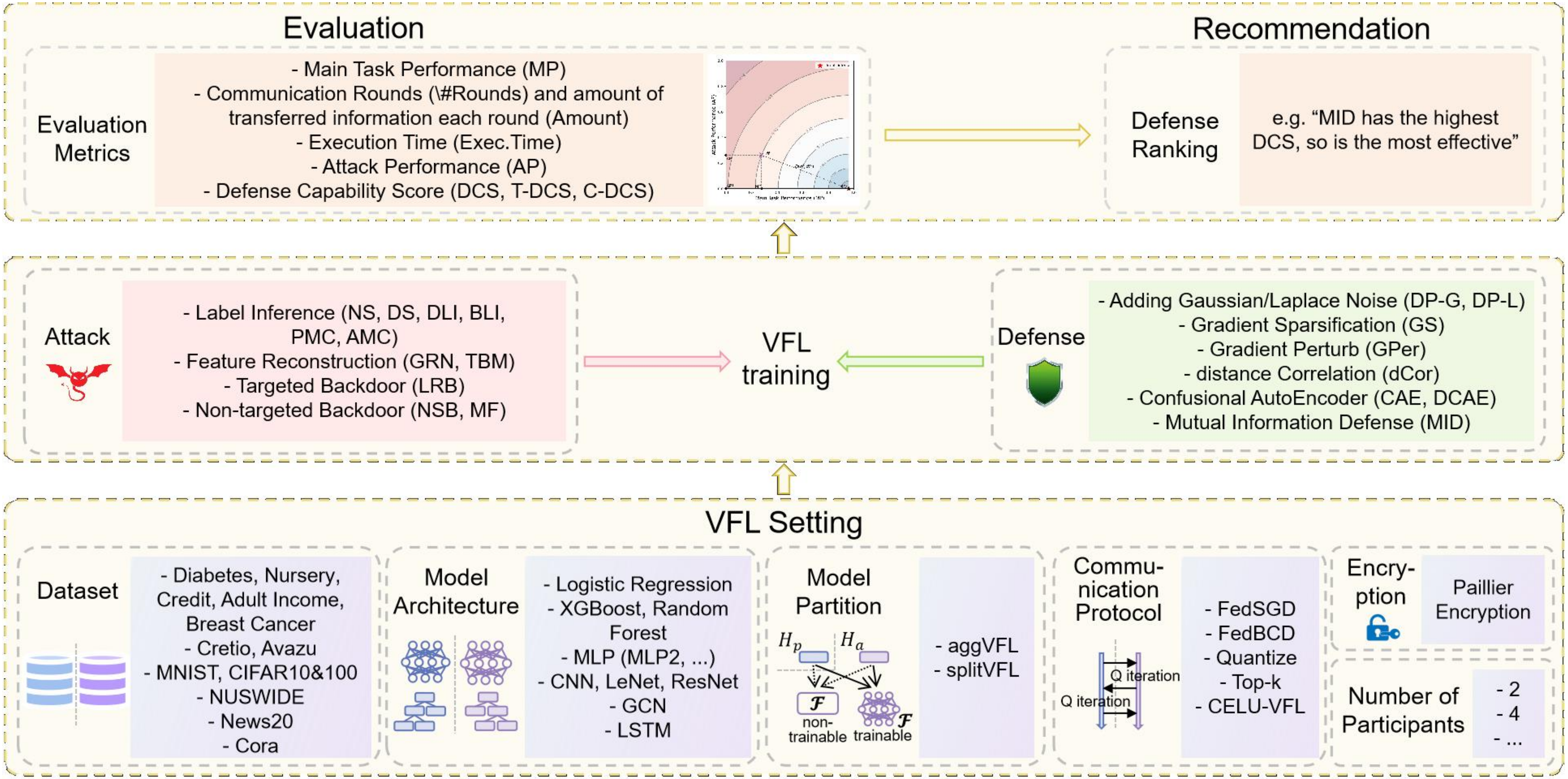
aggVFL: non-trainable global function



splitVFL: trainable global model

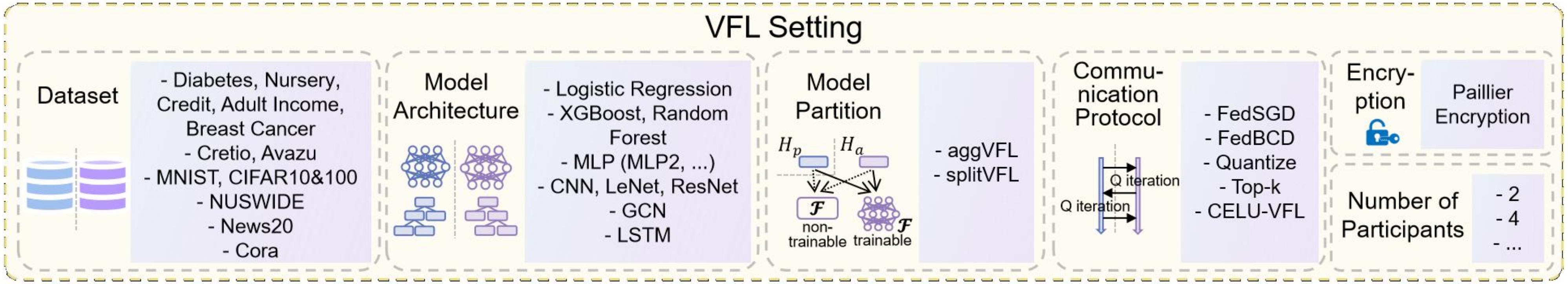
[1] Y. Liu et al. Vertical Federated Learning: Concepts, Advances, and Challenges. IEEE Transactions on Knowledge and Data Engineering, 2024.

VFLAIR:
An Extensible and Lightweight VFL Research Library



Highlight #1: Comprehensive Evaluation of VFL Settings

清華大学 智能产业研究院
Institute for AI Industry Research, Tsinghua University



Evaluted settings include (each can be user-defined):

- 13 datasets
  - including 4 real world dataset (Criteo, Avazu, Cora and News20-S5)
- 20+ model architectures
  - including LR, tree, random forest and NN
- 2 partition settings
  - aggVFL and splitVFL
- 5 communication protocols
  - FedBCD, FedSGD, Quantize, Top-k and CELU-VFL
- 1 encryption technique
  - Paillier Encryption
- 2 kinds of number of participants
  - 2-party and 4-party

Table 3: MP under 4 different settings of NN-based VFL. $Q = 5$ when FedBCD is applied. In "#Rounds" column, the first and second numbers are the communication rounds needed to reach the specified MP for FedSGD and FedBCD respectively.

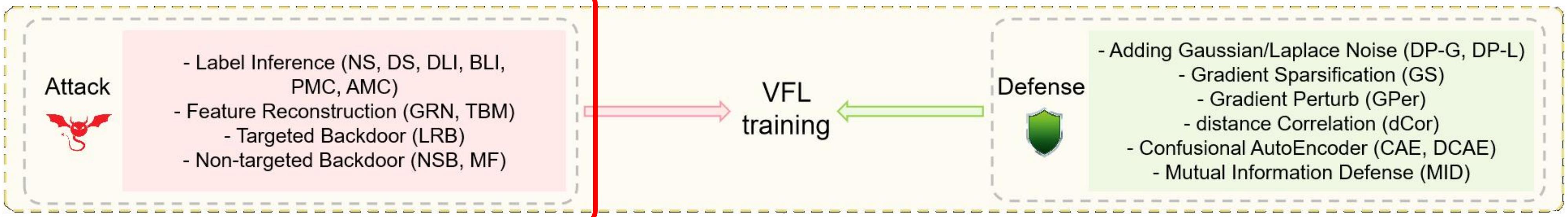
Dataset	aggVFL, FedSGD	aggVFL, FedBCD	#Rounds	splitVFL, FedSGD	splitVFL, FedBCD	#Rounds
MNIST	0.972±0.001	0.971 ±0.001	150 / 113	0.973±0.001	**0.974±0.001**	180 / 143
NUSWIDE	0.887±0.001	0.882±0.001	60 / 26	**0.888±0.001**	0.884±0.001	60 / 29
Breast Cancer	0.914±0.033	0.919±0.029	5 / 3	**0.925±0.028**	0.907±0.045	5 / 4
Diabetes	0.755±0.043	0.736±0.021	15 / 13	**0.766±0.024**	0.746±0.039	15 / 11
Adult Income	0.839±0.006	0.841±0.005	17 / 15	0.842±0.004	**0.842±0.005**	30 / 13

Table 5: MP and execution time under 2 different types of tree-based VFL.

Dataset		Random Forest w/o Encryption	XGBoost w/o Encryption	Random Forest w/ Encryption	XGBoost w/ Encryption (a.k.a. SecureBoost)
Credit	MP	0.816±0.005	0.816±0.004	0.816±0.005	0.816±0.004
	Exec.Time [s]	138±4	366±16	410±10	881±6
Nursery	MP	0.884±0.010	0.890±0.011	0.884±0.010	0.890±0.011
	Exec.Time [s]	29±2	69±4	243±5	1194±21

Highlight #2:
Comprehensive Evaluation of 11 Attacks and 8 Defenses

清華大学 智能产业研究院
Institute for AI Industry Research, Tsinghua University



Label Inference Attack:

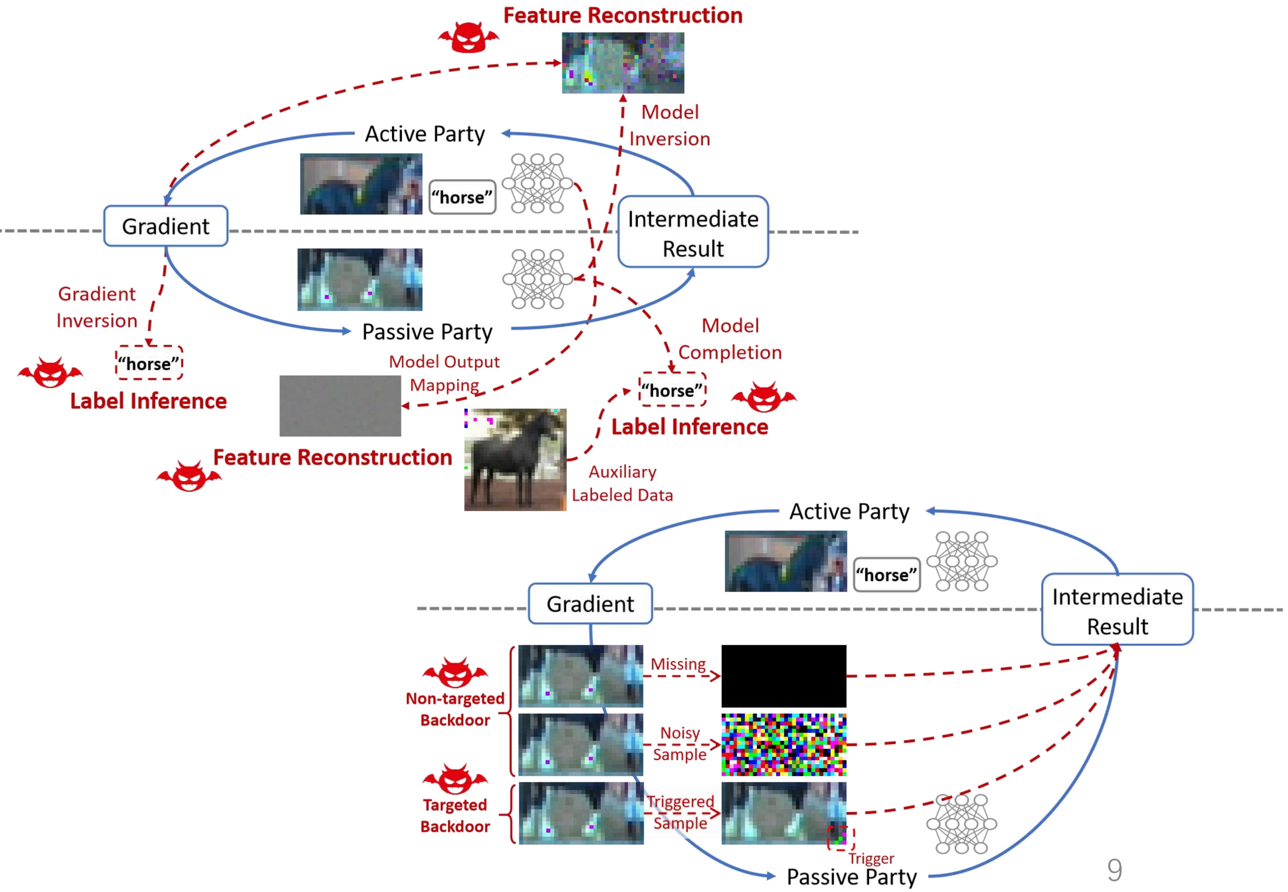
- Use sample or batch level gradient inversion or auxiliary labeled data to infer sensitive label information

Feature Reconstruction Attack:

- Use model inversion or model output mapping to infer other parties private local data

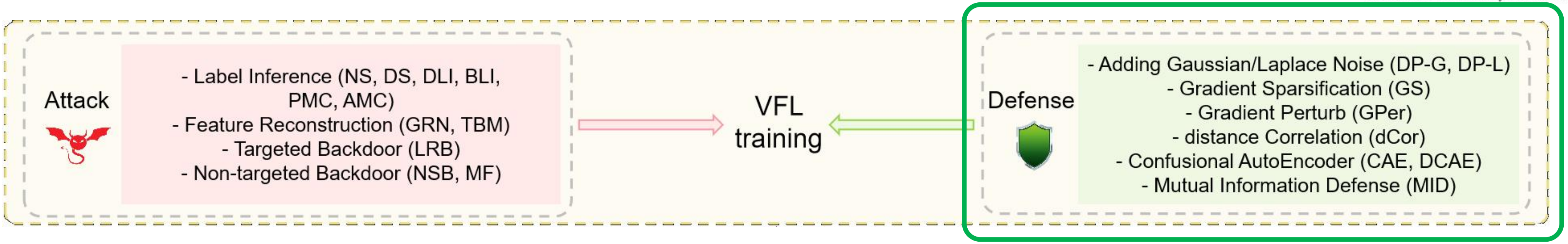
Backdoor Attack:

- **Targeted**: Inject backdoor through transmitted information to mislabel samples marked with attacker selected trigger into target class during training
- **Non-targeted**: Harm model performance



Highlight #2: Comprehensive Evaluation of 11 Attacks and 8 Defenses

清華大学 智能产业研究院
Institute for AI Industry Research, Tsinghua University



8 kinds of non-cryptography defense techniques:

1. Defend by reduce information:
   - Add random noise [1]
     - Gaussian noise (DP-G)
     - Laplace noise (DP-L)
   - Gradient Sparsification (GS) [2]

2. Emerging defense methods:
   - Achieve label-DP by Gradient Perturb (GPer) [3]
   - Disguise label (CAE, DCAE) [4]
   - Distance Correlation Regularization (dCor) [5]
   - Mutual Information Regularization (MID) [6]

[1] C. Dwork. Differential privacy. In Proceedings of the 33rd International Conference on Automata, Languages and Programming, 2006.
[2] A. F. Aji et al. Sparse communication for distributed gradient descent. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.
[3] X. Yang et al. Differentially private label protection in split learning. arXiv preprint, 2022.
[4] T. Zou et al. Defending batch-level label inference and replacement attacks in vertical federated learning. IEEE Transactions on Big Data, 2022.
[5] J. Sun et al. Label leakage and protection from forward embedding in vertical federated learning. arXiv preprint, 2022.
[6] T. Zou et al. Mutual information regularization for vertical federated learning. arXiv preprint, 2023.

Highlight #2:
Comprehensive Evaluation of 11 Attacks and 8 Defenses

清華大学 智能产业研究院
Institute for AI Industry Research, Tsinghua University

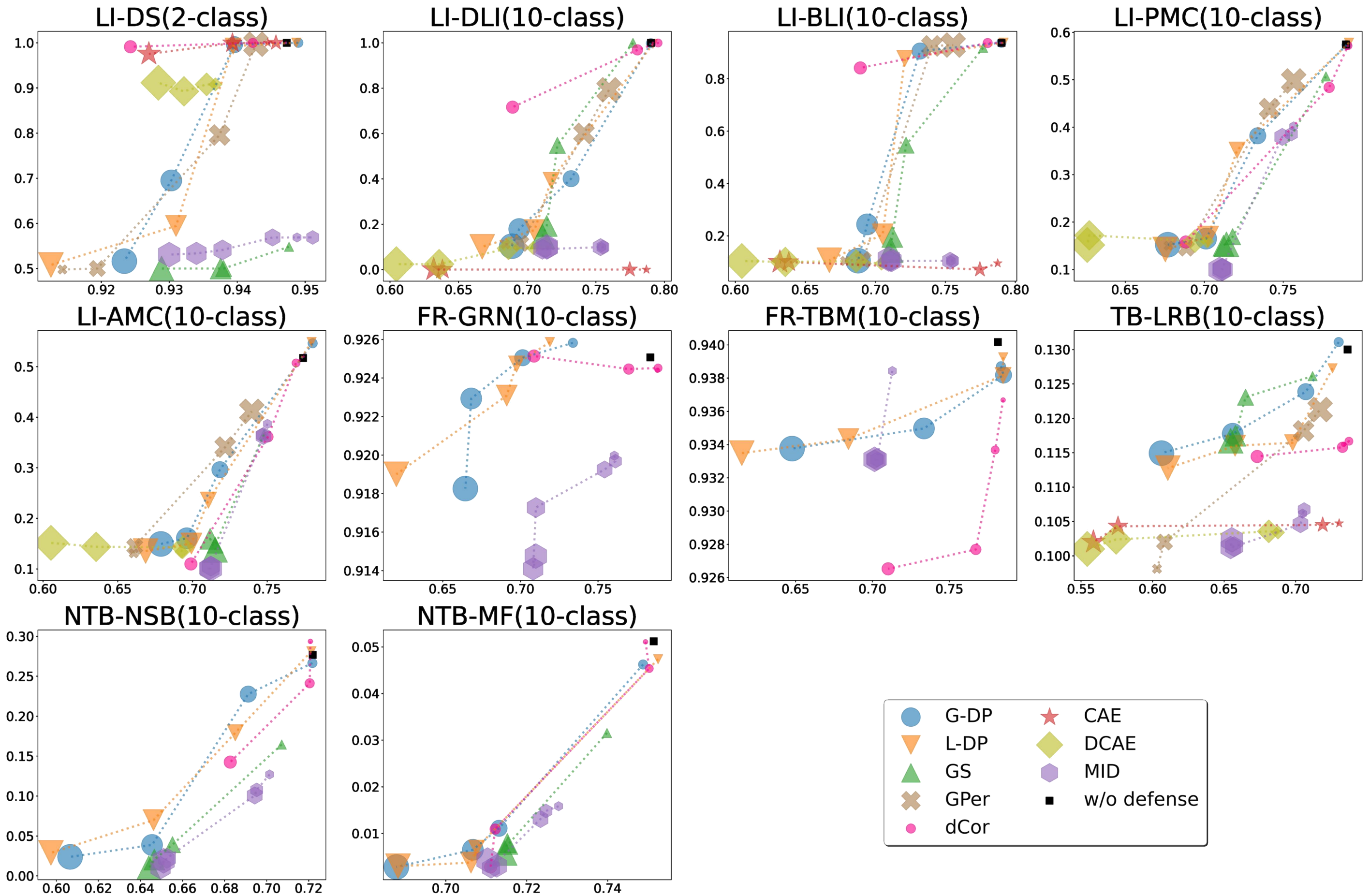
- Attacks pose great threat to VFL.

- Black squares in each sub-figure
- DS, DLI, BLI and TBM attacks are strong attacks.

- Defenses exhibit trade-offs between main task performance (MP) and attack performance (AP).

- Trade-off can be controlled by adjusting defense hyper-parameters.

Figure 3: MPs and APs for different attacks under defenses [CIFAR10 dataset, **aggVFL**, **FedSGD**]

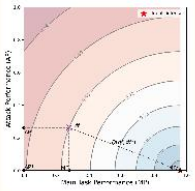


Highlight #3: Novel Evaluation Metric: Defense Capability Score

Evaluation

Evaluation Metrics

- Main Task Performance (MP)
- Communication Rounds (\#Rounds) and amount of transferred information each round (Amount)
- Execution Time (Exec.Time)
- Attack Performance (AP)
- Defense Capability Score (DCS, T-DCS, C-DCS)



Recommendation

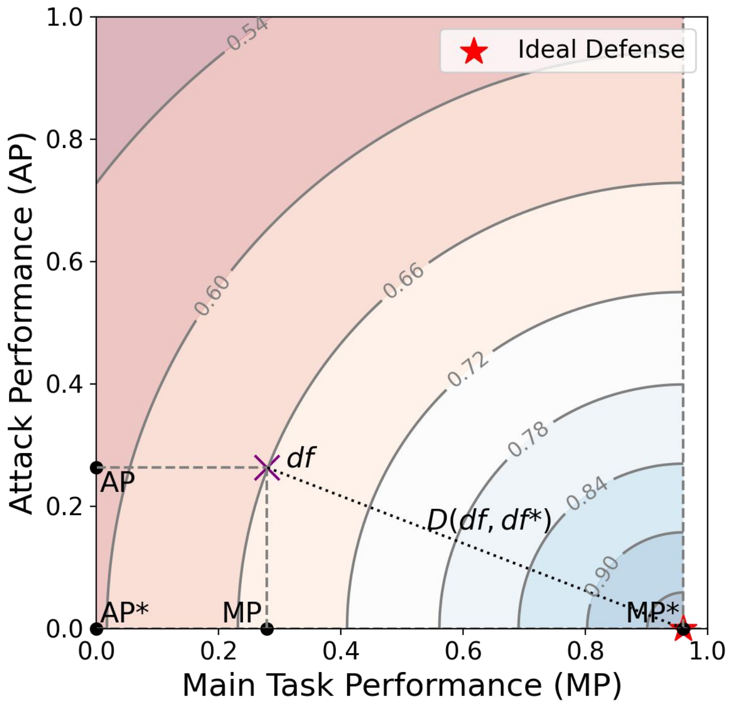
Defense Ranking

e.g. "MID has the highest DCS, so is the most effective"

Evaluation Metrics

1. Main Task Performance (MP)
2. Communication efficiency (till reaching the target training MP)
    - Required communication rounds (#Rounds)
    - Amount of transferred information each round (Amount)
3. Computation efficiency (till reaching the target training MP)
    - Execution Time (Exec. Time)
4. Attack Performance (AP)
    - Label Inference: ratio of corretly inferred label
    - Feature Reconstruction: negative MSE of real and inferred feature
    - Targeted Backdoor: successful rate of backdoor
    - Non-targeted Backdoor: decrease of MP
5. **Defense Capability Score (DCS)**: considering both MP and AP

$$\mathrm{DCS} = \frac{1}{1 + D(df, df^*)} = \frac{1}{1 + \sqrt{(1-\beta)(\mathrm{AP} - \mathrm{AP}^*)^2 + \beta(\mathrm{MP} - \mathrm{MP}^*)^2}}$$



Highlight #3:
Novel Evaluation Metric: Defense Capability Score

清華大学 智能产业研究院
Institute for AI Industry Research, Tsinghua University

- DCS rankings are consistent across various datasets and settings.

- Change in β does not significantly impact the C-DCS ranking.

- This demonstrates the stableness of the comparison results among various defenses.

- MID, L-DP and G-DP are effective on a wide spectrum of attacks.

- MID ranks the highest, followed by DP for all datasets.

Table 8: T-DCS and C-DCS for All Defenses [NUSWIDE dataset, **aggVFL**, **FedSGD**]

Defense Name	Defense Parameter	$T\text{-}DCS_{LI_2}$	$T\text{-}DCS_{LI_5}$	$T\text{-}DCS_{LI}$	$T\text{-}DCS_{FR}$	$T\text{-}DCS_{TB}$	$T\text{-}DCS_{NTB}$	$C\text{-}DCS$
MID	10000	0.7358	0.8559	**0.8159**	0.5833	**0.7333**	0.8707	0.7508
MID	1.0	0.7476	0.8472	0.8140	0.5833	0.7331	0.8700	0.7501
MID	100	0.7320	0.8536	0.8130	0.5833	0.7326	**0.8711**	0.7500
G-DP	0.1	0.7375	0.8262	0.7966	0.5863	0.7282	0.8675	0.7447
L-DP	0.1	0.7389	0.8177	0.7915	0.5863	0.7258	0.8603	0.7410
MID	0.1	0.7516	0.8259	0.8011	0.5833	0.7172	0.8563	0.7395
MID	0.01	0.7280	0.8092	0.7822	0.5844	0.7151	0.8627	0.7361
dCor	0.3	**0.7641**	0.8411	0.8155	0.5834	0.7289	0.8051	0.7332
dCor	0.0001	0.6496	0.6340	0.6392	**0.5864**	0.6307	0.8287	0.6712
GS	99.0	0.7404	0.8060	0.7841	-	0.6415	0.8408	-
CAE	1.0	0.6863	0.7822	0.7502	-	0.6830	-	
DCAE	0.0	0.6669	**0.8660**	0.7996	-	0.6816	-	-
GPer	0.01	0.7386	0.8412	0.8070	-	0.7193	-	-

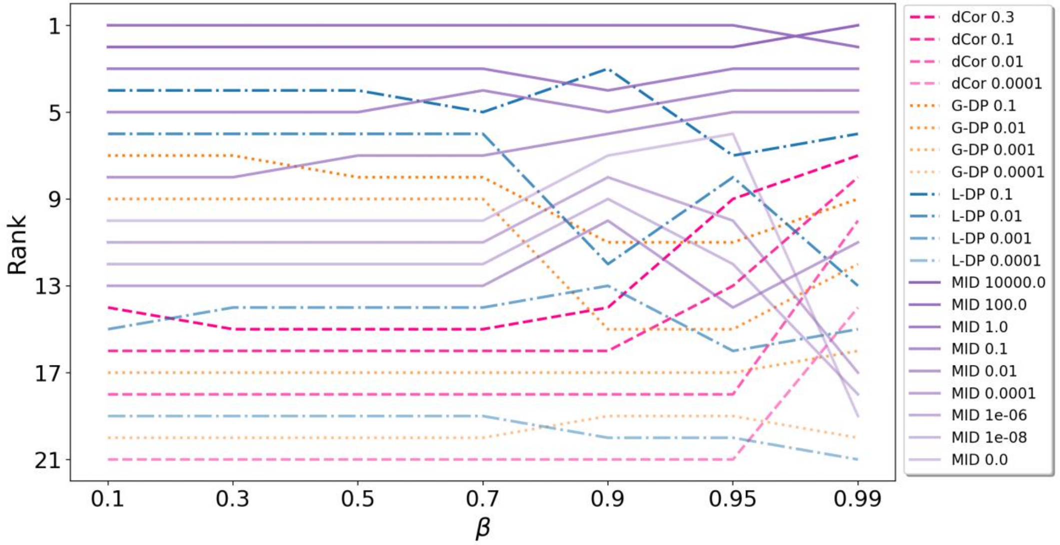


Figure 4: Change of C-DCS ranking with the change of β. [MNIST dataset, aggVFL, FedSGD]

Highlight #4: Additional Insights

- splitVFL is less vulnerable to attacks than aggVFL.

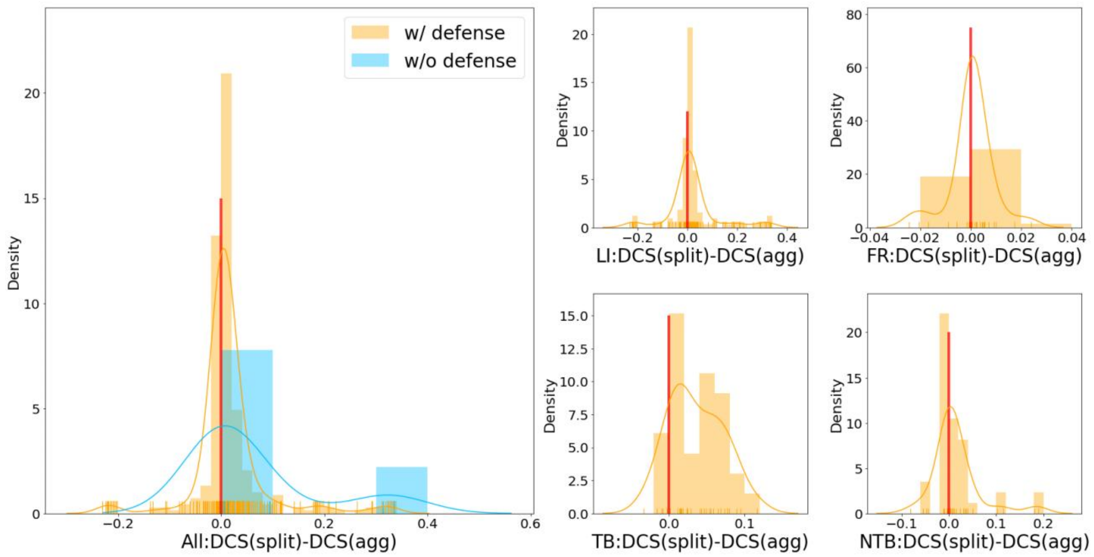


Figure 5: DCS gap Distribution, y-axis represents density [MNIST dataset, splitVFL/aggVFL, FedSGD]

- FedBCD is less vulnerable to attacks than FedSGD.

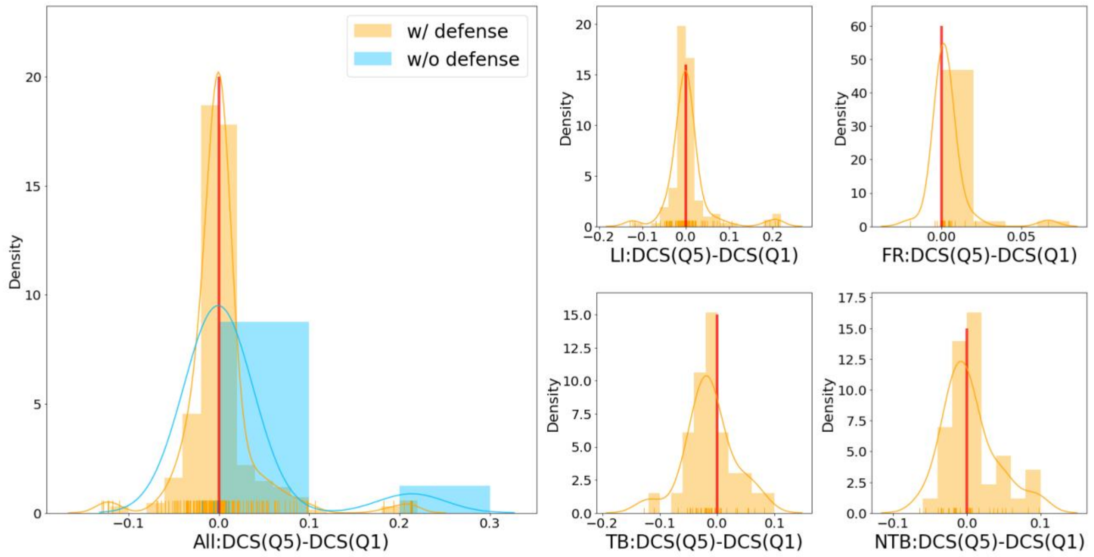


Figure 6: DCS gap Distribution, y-axis represents density [MNIST dataset, aggVFL, FedBCD/FedSGD]

Comprehensive User Guidance and Documentation

- User guidance is included in the appendix of the paper.

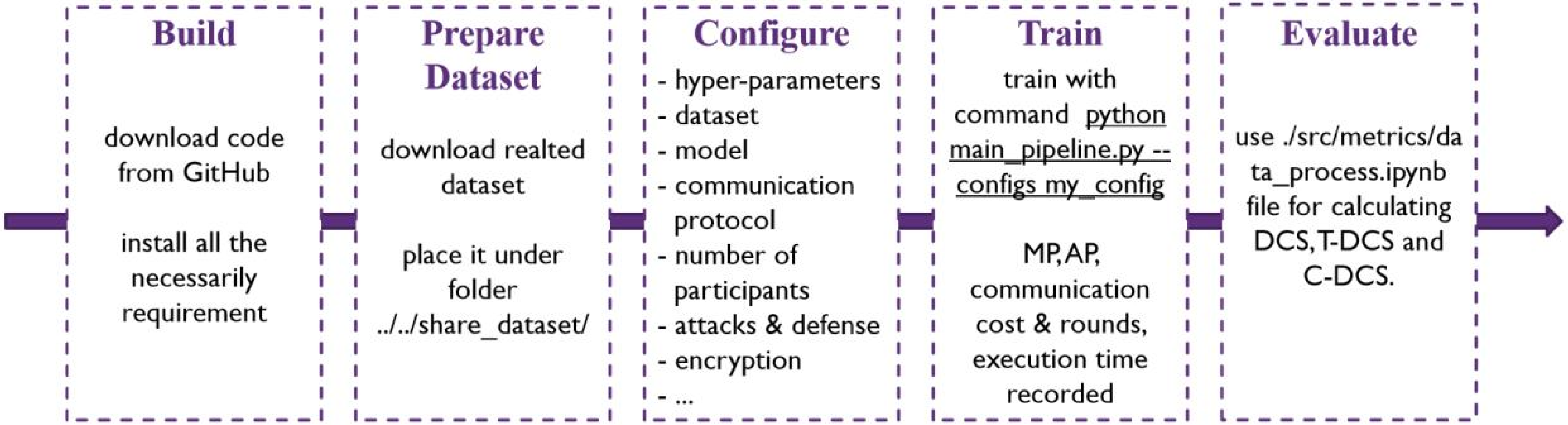


Figure 8: Step-by-step user guidance for using VFLAIR.

- Documentation is provided in the README.md file in our github **https://github.com/FLAIR-THU/VFLAIR**.

清华大学智能产业研究院
Institute for AI Industry Research, Tsinghua University

Thanks !