Federated Robustness Propagation: Sharing Adversarial Robustness in Federated Learning

Junyuan Hong[1], Haotao Wang[2], Zhangyang Wang[2], Jiayu Zhou[1]

[1] Michigan State University, [2] University of Texas at Austin

Adversarial Training (AT) on Heterogeneous Devices

$$\ell = (\ell_a + \ell_{\text{CE}})/2$$

$$\ell_{\text{CE}}(f(x), y) = -\sum_{t=1}^{c} y_t \log(f(x)_t) \qquad \ell_a(f; x, y) = \max_{\|\delta\| \le \epsilon} \ell(f(x+\delta), y)$$

- **High cost of adversarial training**:
  - Increased communication latency.
  - High energy cost for battery-powered edge devices.
- **Ubiquitous essence of robustness**:
  - Security in self-driving vehicles.
  - Generalization on mild perturbation.

Federated Robustness Propagation

- **Problem setup**
  - Resources
  - Features
- **Challenges**
  - Transferability of robustness;
  - Efficiency of robustness sharing.

$$\text{FRP}(\{f_k\}; \{D_k | D_k \sim \mathcal{D}_i\}) \triangleq \sum_{k \in T} L_{\text{ST}}(f_k, D_k) + \sum_{k \in S} L_{\text{AT}}(f_k, D_k). \quad (1)$$

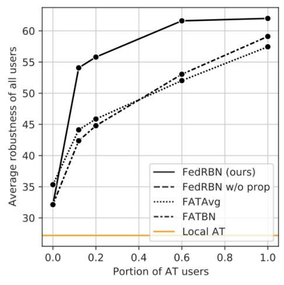
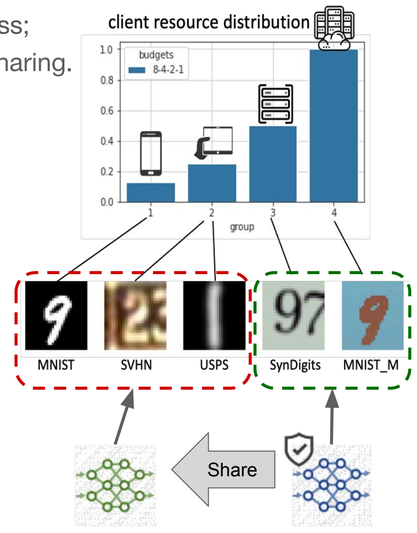


Figure 1: Comparison of robustness on a varying portion of AT users, where a 5-domain digit recognition dataset is distributed to 50 users in total and details are in Appendix C.7.



Federated Robust Batch-Normalization (FedRBN)

Revisit BN

$$\text{BN}(x; \mu, \sigma) \triangleq w \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon_0}} + b,$$

Known principles:

- **Dual BN (DBN)**: Use different BN for clean and adversarial samples, respectively.
  - Pros: Better trade-off between robust accuracy (RA) and standard accuracy (SA).
  - Cons: Cannot mitigate domain gap for using the same BN for all clients.
- **Local BN (LBN)**: Use local BN to mitigate feature heterogeneity.
  - Pros: Better SA on different domains.
  - Cons: Trade in more SA for higher RA.

Share robustness via BN

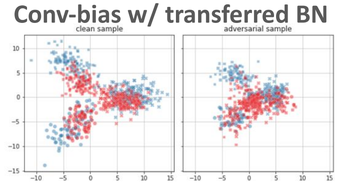
1) **LBN+DBN:** Joint use of LBN and DBN for better trade-off and domain personalization.
2) **Fill in missing BN statistics:** Estimate adversarial BN

$$\hat{\mu}_{a,k} = \frac{1}{|S|} \sum_{j \in S} \alpha_j \mu_{a,j}, \ \hat{\sigma}^2_{a,k} = \frac{1}{|S|} \sum_{j \in S} \alpha_j \sigma^2_{a,j},$$

$$\alpha_j = \text{Softmax}_T\left[\frac{1}{L} \sum_{l=1}^{L} \text{Sim}^l(D_k, D_j)\right],$$

3) **Reduce conv-bias:** Debias convolutional parameters by transferred adversarial BN,

$$(1-\lambda)\ell_{\text{CE}}\big(f_k(x; \text{BN}_c), y\big) + \lambda \ell_{\text{CE}}\big(f_k(x; \text{BN}_a), y\big),$$



paper code

Algorithm 1: FedRBN: user-end training

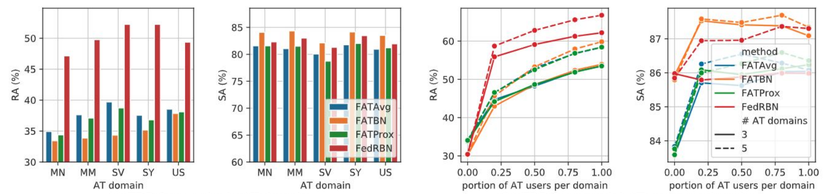
Input: User budget type (AT or ST), initial parameters θ (AT) or $(\theta, \hat{\mu}_a, \hat{\sigma}^2_a)$ (ST) of the model f from the server, adversary $A_\epsilon(\cdot)$, dataset D
for mini-batch $\{(x, y)\}$ in D **do**
  $\ell_c \leftarrow \mathbb{E}_{(x,y)}[\ell_{\text{CE}}(f(x; \text{BN}_c), y)]$
  Update (μ, σ^2) of BN_c
  if user budget type is AT **then**
    Perturb data $\tilde{x} \leftarrow A_\epsilon(f(x; \text{BN}_a))$
    $L \leftarrow \frac{1}{2}\{\ell_c + \mathbb{E}_{(\tilde{x},y)}[\ell_{\text{CE}}(f(\tilde{x}; \text{BN}_a), y)]\}$
    Update (μ_a, σ^2_a) of BN_a
  else
    Replace BN_a parameters with $(\hat{\mu}_a, \hat{\sigma}^2_a)$
    $L \leftarrow (1-\lambda)\ell_c + \lambda \mathbb{E}_{(x,y)}[\ell_{\text{CE}}(f(x; \text{BN}_a), y)]$
  end if
  Optimize L to update θ by gradient descent
end for
Upload $(\theta, \mu, \sigma^2, \mu_a, \sigma^2_a)$ (AT) or (θ, μ, σ^2) (ST)

Empirical Results

Ablation Study

Table 1: Ablation of different test-time BNs.

λ	test BN	weight	Digits						DomainNet					
			All		20%		MNIST		All		20%		Real	
			RA	SA	RA	SA	RA	SA	RA	SA	RA	SA	RA	SA
0	BN_c		52.8	**86.7**	41.9	86.6	34.6	84.7	35.5	61.4	**22.1**	**65.0**	15.4	**65.9**
0	tran. BN_a	uni	**62.0**	84.9	50.6	83.2	**41.5**	80.2	**35.7**	**61.6**	19.8	60.5	13.2	56.1
0	tran. BN_a	cos	**62.0**	84.9	**51.0**	83.5	**41.5**	80.2	**35.7**	**61.6**	21.4	62.5	12.8	56.1
0.5	BN_c		52.8	**86.7**	50.0	87.0	42.2	84.1	35.5	61.4	26.5	61.2	21.0	62.0
0.5	tran. BN_a	uni	**62.0**	84.9	55.4	86.9	51.5	87.2	**35.7**	**61.6**	27.5	61.3	**26.4**	**64.0**
0.5	tran. BN_a	cos	**62.0**	84.9	**55.8**	**87.3**	**58.5**	**86.5**	**35.7**	**61.6**	**28.1**	**62.5**	**26.4**	63.9



(a) FRP from a single AT domain (b) FRP from partial AT users per domain

Figure 3: Evaluating FRP performance with different FRP settings.

Benchmark Results

- Benchmarks of robustness propagation, where we measure the per-epoch computation time (T) by counting $\times 10^{12}$ times of multiplication-or-add operations (MACs) to evaluate the efficiency.

	LBN	DBN	Digits									DomainNet								
AT users			All			20%			MNIST			All			20%			Real		
Metrics			RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T
FedRBN $\lambda = 1$	✓	✓	**62.0**	84.9	7.4	**60.6**	86.5	2.5	**60.8**	83.9	2.5	**35.7**	**61.6**	127.9	27.6	56.0	42.6	**28.2**	58.3	39.1
FedRBN $\lambda = 0.5$	✓	✓	**62.0**	84.9	7.4	55.8	**87.3**	2.9	58.5	**86.5**	2.9	**35.7**	**61.6**	127.9	**28.1**	62.5	51.2	26.4	63.9	48.0
FedRBN $\lambda = 0$	✓	✓	**62.0**	84.9	7.4	51.0	83.5	2.2	41.5	80.2	2.2	**35.7**	**61.6**	127.9	21.4	62.5	38.4	12.8	56.1	34.6
FATAvg+DBN		✓	60.0	83.8	7.4	48.8	82.8	2.2	40.2	79.9	2.2	27.6	52.8	127.9	16.6	58.9	38.4	13.0	54.8	34.6
FATBN	✓		60.0	**87.3**	7.4	41.2	86.4	2.2	36.5	86.4	2.2	35.2	60.2	127.9	20.3	**63.2**	38.4	15.7	**64.7**	34.6
FATAvg			58.3	86.1	7.4	42.6	84.6	2.2	38.4	84.1	2.2	24.6	47.4	127.9	15.4	57.8	38.4	10.7	57.9	34.6
FATProx			58.5	86.3	7.4	42.8	84.5	2.2	38.1	84.1	2.2	24.8	47.1	127.9	14.5	57.3	38.4	10.4	57.1	34.6
FedRob			13.1	13.1	7.4	20.6	59.3	1032	17.7	48.9	645	-	-	-	-	-	-	-	-	-

Acknowledgement

This material is based in part upon work supported by the National Science Foundation (IIS-1749940, IIS-2212174, ECCS-2024270), NIH/National Institute on Aging (1RF1AG072449) and Office of Naval Research (N00014-20-1-2382).