IEEE TCCN SIG on AI

Brainstorming Generative Adversarial Networks (BGANs): Framework and Application to Wireless Networks

Walid Saad

Electrical and Computer Engineering Department,
Network sciEnce, Wireless, and Security (NEWS) Group

Virginia Tech

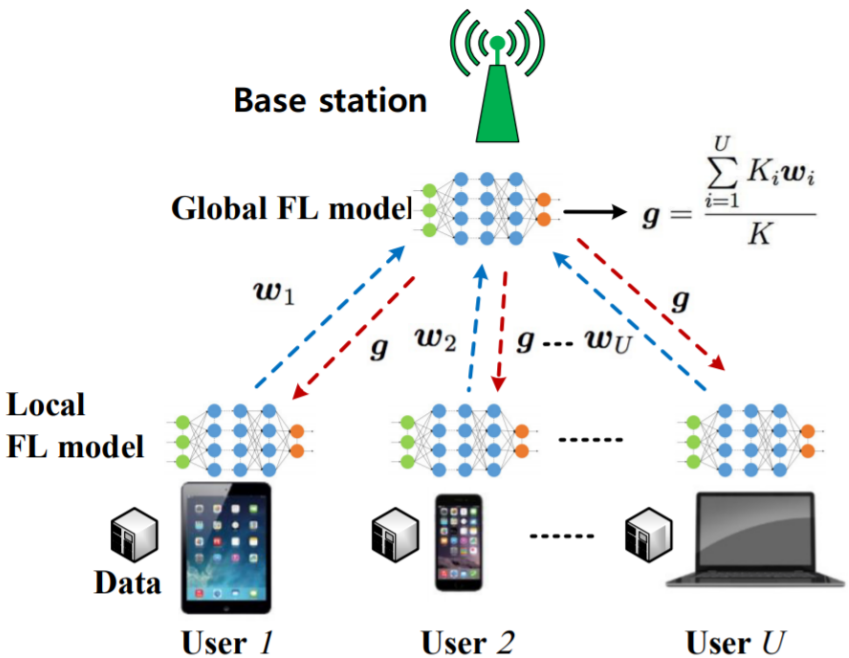
Email: walids@vt.edu
Group: http://www.netsciwis.com
Personal: http://resume.walid-saad.com

Outline

- Brief introduction on distributed learning
- Brainstorming generative adversarial networks
  - Why generative models?
  - Problem formulation
  - Key results
- BGAN in wireless networks
  - Application to channel modeling
- Glance at experienced deep RL with GANs
- Conclusions

Massive Small Data and Distributed Machine learning

- Cloud big data was the norm but data is distributed, local, and private
- Can users collaboratively learn a task of interest?



- Learning (at the edge) to communicate?
- Communication for learning (joint design)?
- Multi-agent learning and generative models

Generative Models vs Discriminative Models

- In statistics and machine learning two main areas are the **generative** approach and the **discriminative** approach.
- Let $\boldsymbol{X}$ be the observable variable and $\boldsymbol{y}$ be the target variable, then:
  - A generative model is a statistical model of the joint probability distribution on $\boldsymbol{X}\times\boldsymbol{y}$, $\boldsymbol{P}(\boldsymbol{X},\boldsymbol{y})$,
  - A discriminative model is a model of the conditional probability of the target $\boldsymbol{y}$, given an observation $\boldsymbol{X}$, $\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{X}=\boldsymbol{x})$.

What is a Generative Adversarial Network (GAN)?

- A **generative** model seeks to create data that is not seen before, but fits some input data distribution

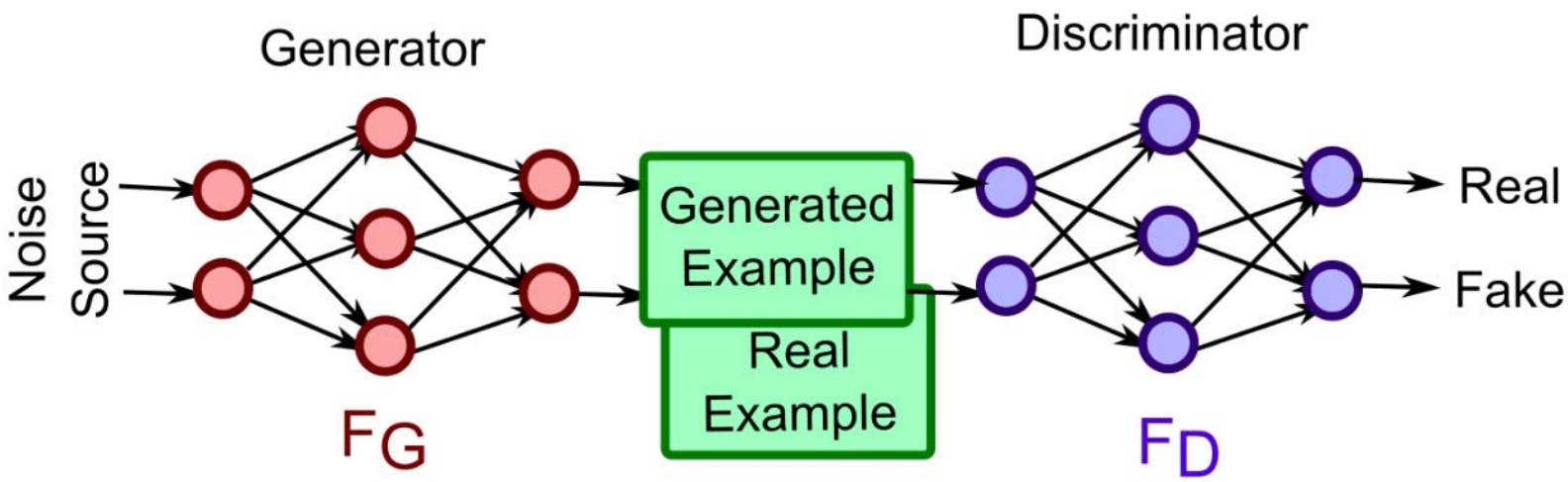


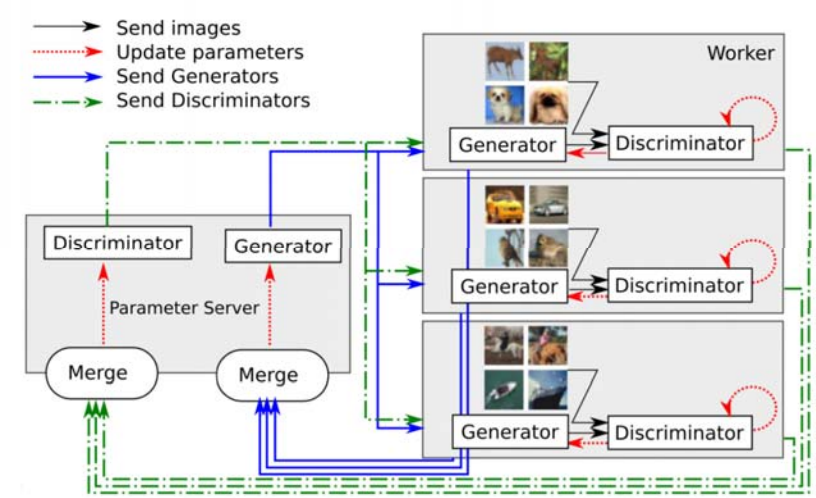
Figure source: http://hunterheidenreich.com/blog/what-is-a-gan/

- Generator: Tries to generate fake data
- Discriminator: Figure out whether data is fake or real
  - Adversarial interactions between the two (game theory)

Distributed GANs?

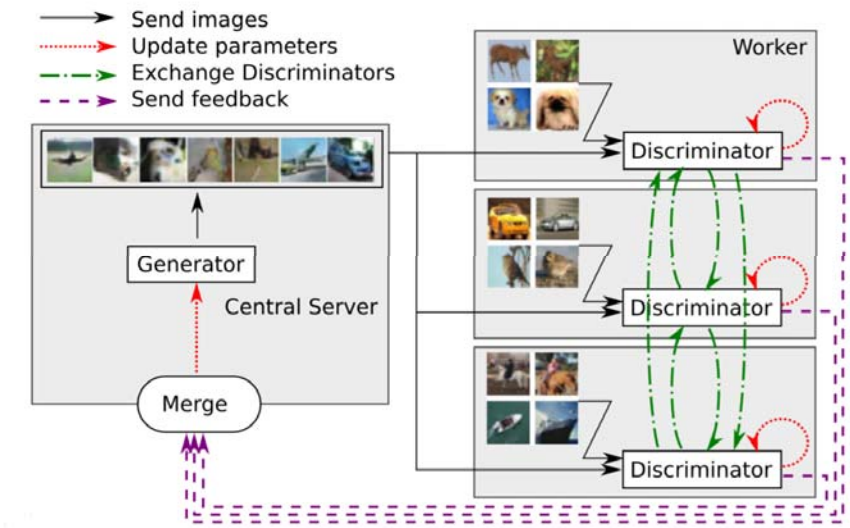
- Existing GAN models (including variants such as InfoGAN, conditional GAN, etc.) are ***centralized***
- What if the data of interest is:
  - Distributed among multiple agents
  - Scarce (each agent has partial data)
  - Private (agents do not want to share their data)
- Can we learn the distribution of the total data **without** sharing the raw data between the agents and **without** relying on a central server?

Existing Distributed GAN Solutions



Federated learning (FLGAN)

- Not fully distributed



Multi-discriminator (MDGAN)

Forgiver-First Update (F2UGAN)

Drawbacks of the state-of-the art

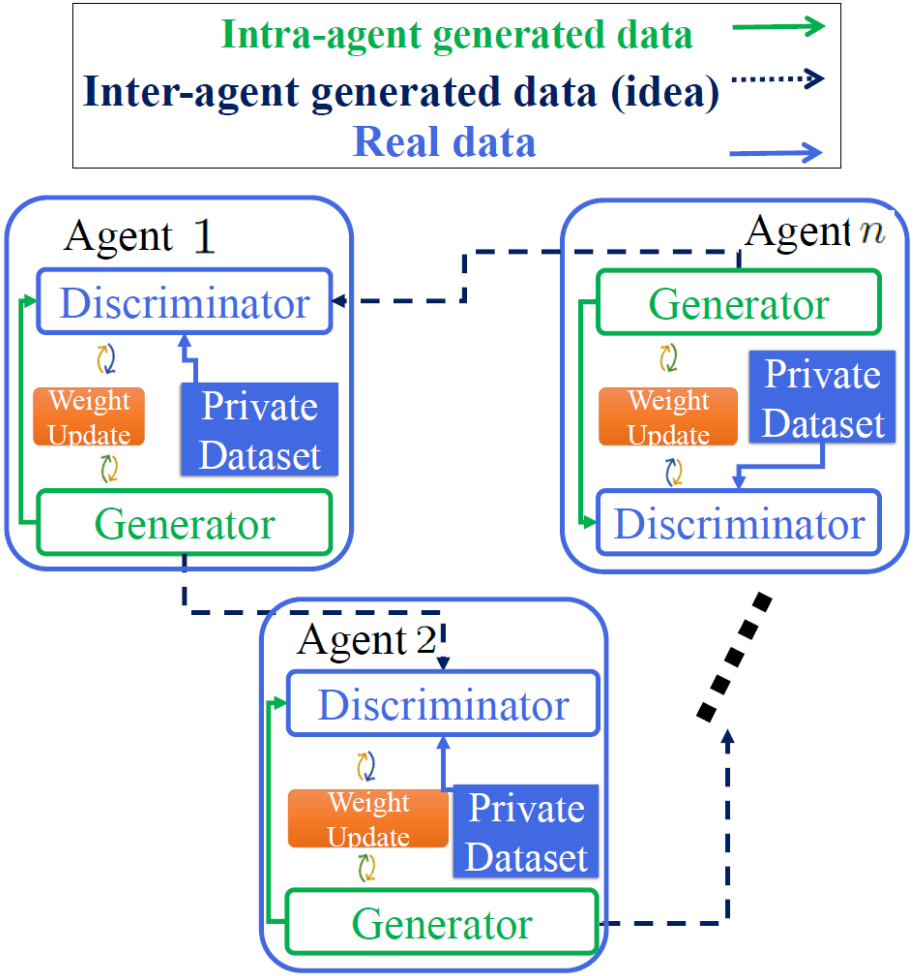
- Not fully distributed (need a central controller)
- Expensive communication requirements particularly for MDGAN and F2UGAN
- Agents cannot have different neural network architectures and, thus, they must be homogeneous
- Agents do not own their generators
- Can we create a **fully-distributed** solution with multiple, **heterogeneous** agents?

- A. Ferdowsi and W. Saad, "Brainstorming Generative Adversarial Networks (BGANs): Towards Multi-Agent Generative Models with Distributed Private Datasets", arXiv:2002.00306.

Brainstorming GANs

- Architecture allows each agent to have their own generator and discriminator

- **Brainstorming:** Share the output of the generator **(ideas)** with other agents at every training epoch



Benefits of BGANs

- Fully distributed, we do not require a central controller or aggregator
- Agents can have different neural network architectures
  - Capabilities-tailored neural networks
- Less communication overhead than most baselines (depends on the dimensions of the data points rather than the neural network parameters)
  - Will be shown to be more efficient in practical cases

Theoretical Formulation

- Each individual agent will have a *brainstorming* value function that extends standalone GAN

$$V_i(D_i, G_i, \{G_j\}_{j\in\mathcal{N}_i}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\mathcal{D}_i}}[\log D_i(x)] + \mathbb{E}_{\boldsymbol{z}\sim p_{z_i}}[\log(1 - D_i(G_i(z)))]$$

Discriminator

Generators of *i* and neighbors

Mixture of data

  - Generator aims to minimize this value, and discriminator seeks to maximize it
  - Interdependence across agents => **game theory!**
- Formally:

$$\{D_i^*\}_{i=1}^n, \{G_i^*\}_{i=1}^n = \arg\min_{G_1,\ldots,G_n} \arg\max_{D_1,\ldots,D_n} V$$

Total utility

- Can we find a Nash equilibrium for this game?

Theoretical Results

- **Intermediary Result:** The optimal discriminator:

$$D_i^* = \frac{p_{b_i}}{p_{b_i} + p_{g_i}}$$

Probability distribution of agent *i*'s generator

- **Theorem.** The game between the agents has a unique Nash equilibrium such that:

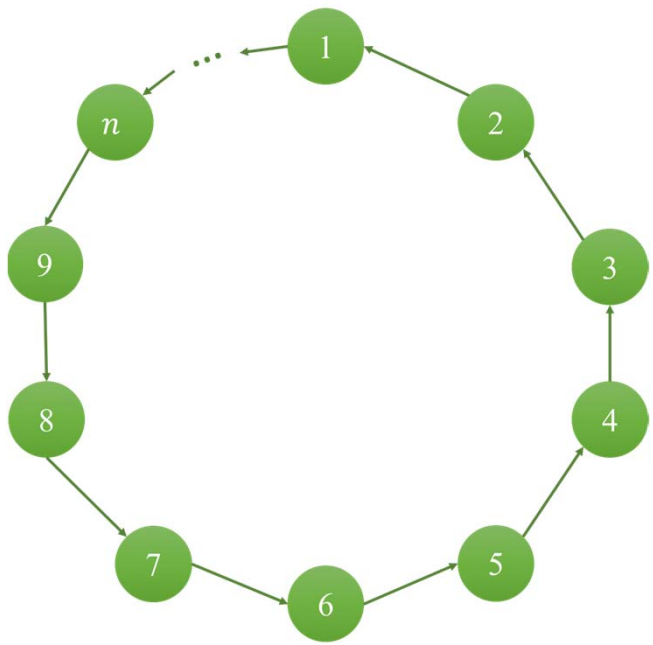
$$p_{g_i}^* = \sum_{j \in \mathcal{N}} \lambda_{ij} p_{\text{data}_j}$$

Depends on communication graph structure

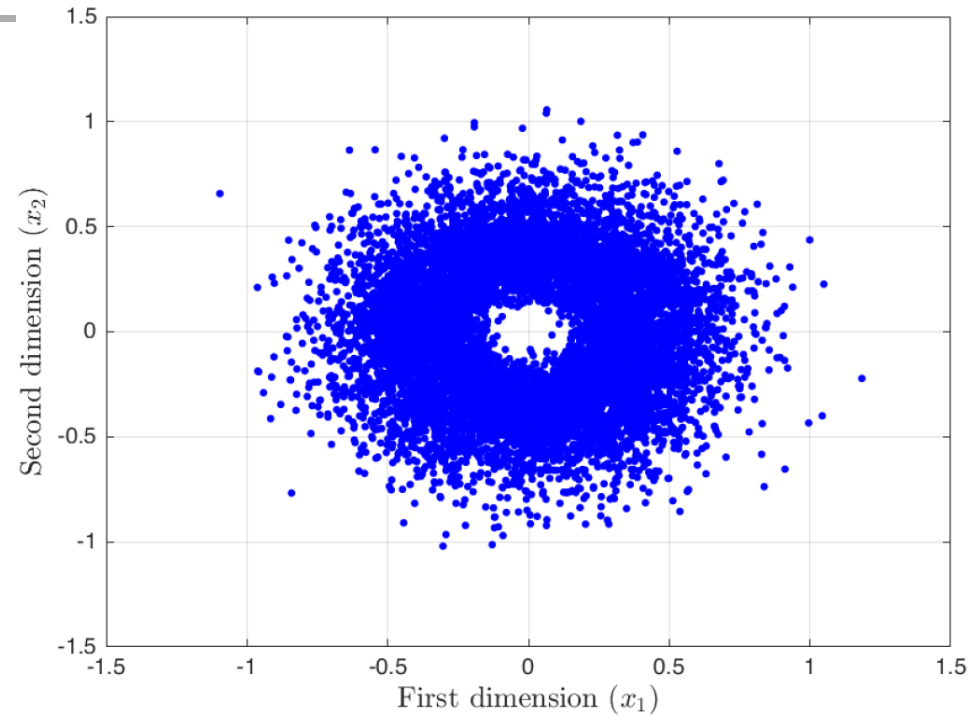
Probability distribution of agent *j*'s data

  - Proof is elaborate uses an additional intermediary result

Simulation Setup



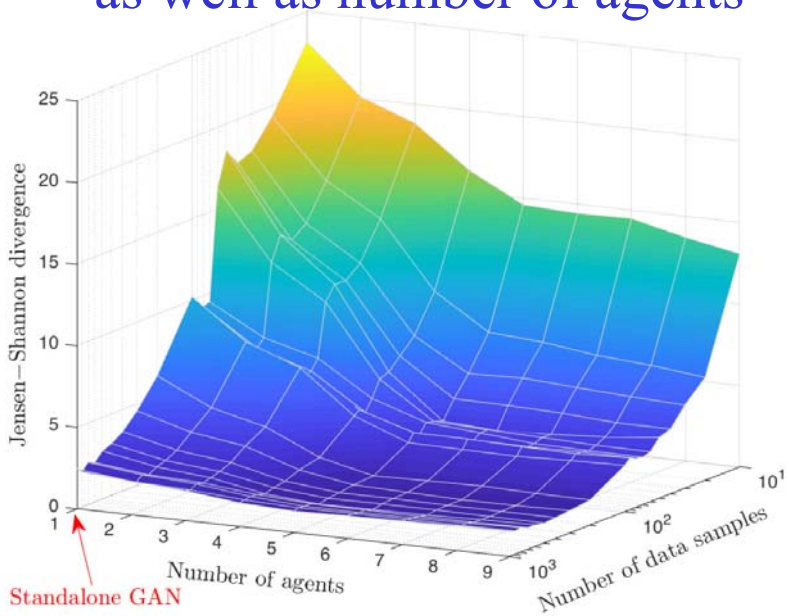
- Graph of connections with only one neighbor for each agent; this graph is strongly connected (some results use a different graph)

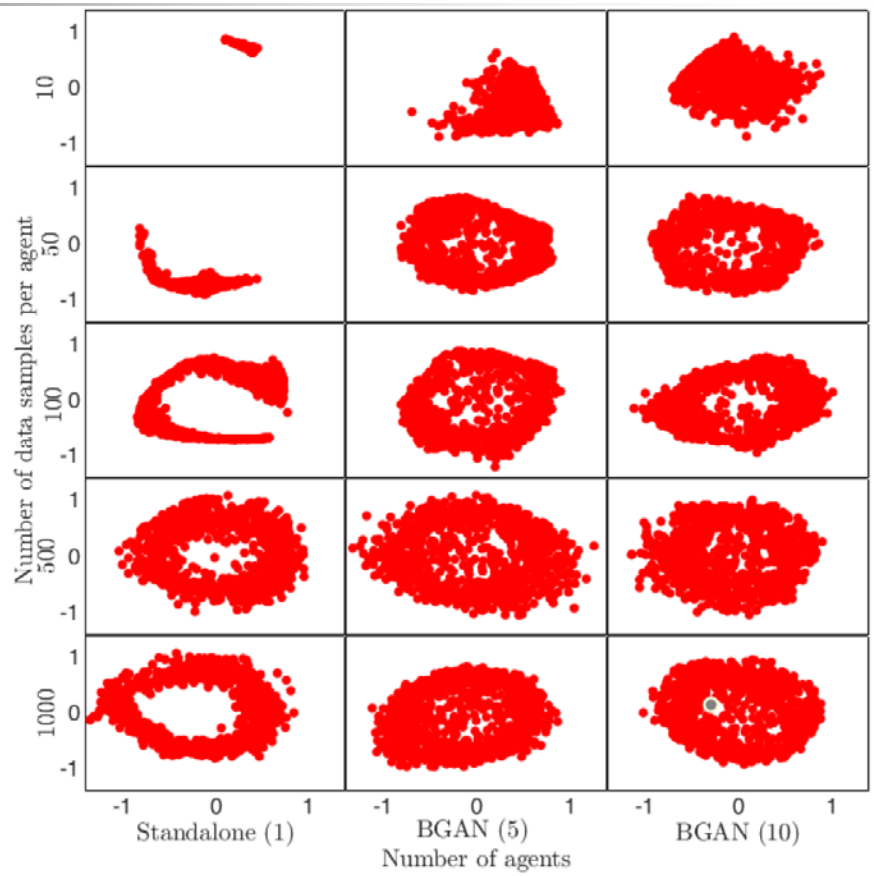


- Illustration of the used data samples drawn from a nonlinear combination of gamma and uniform distributions (ring dataset). Used for ease of exposition

Simulation Results

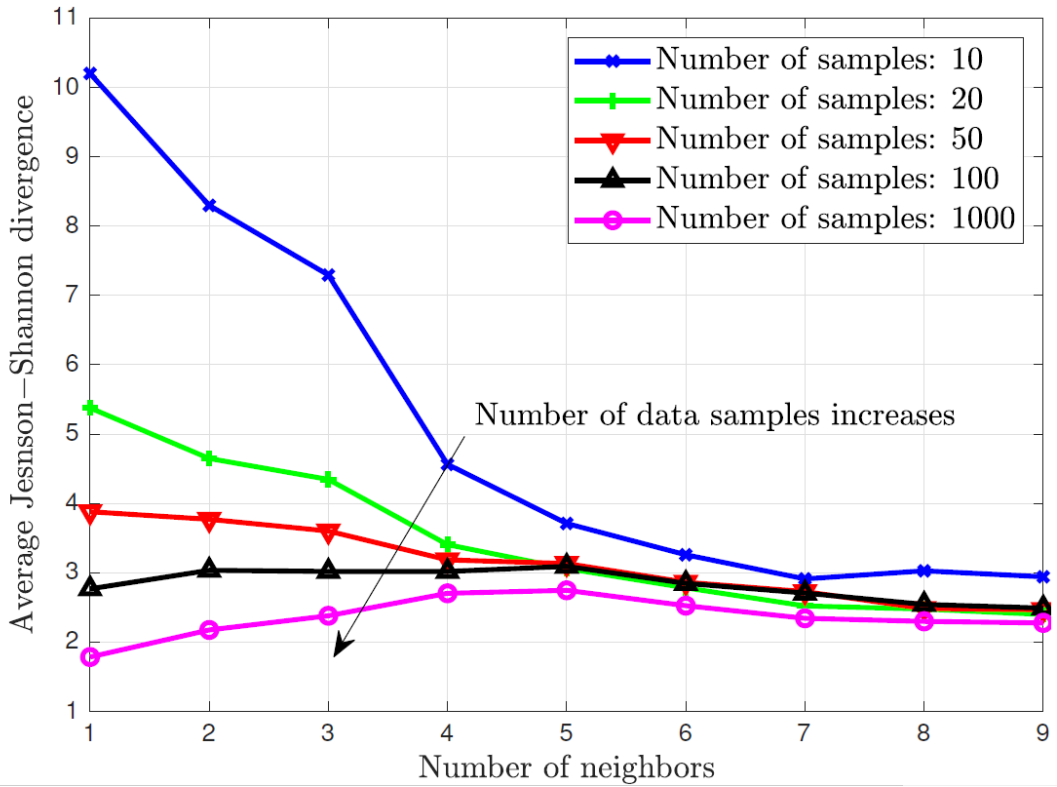
- JSD vs. number of available samples for each BGAN agent as well as number of agents

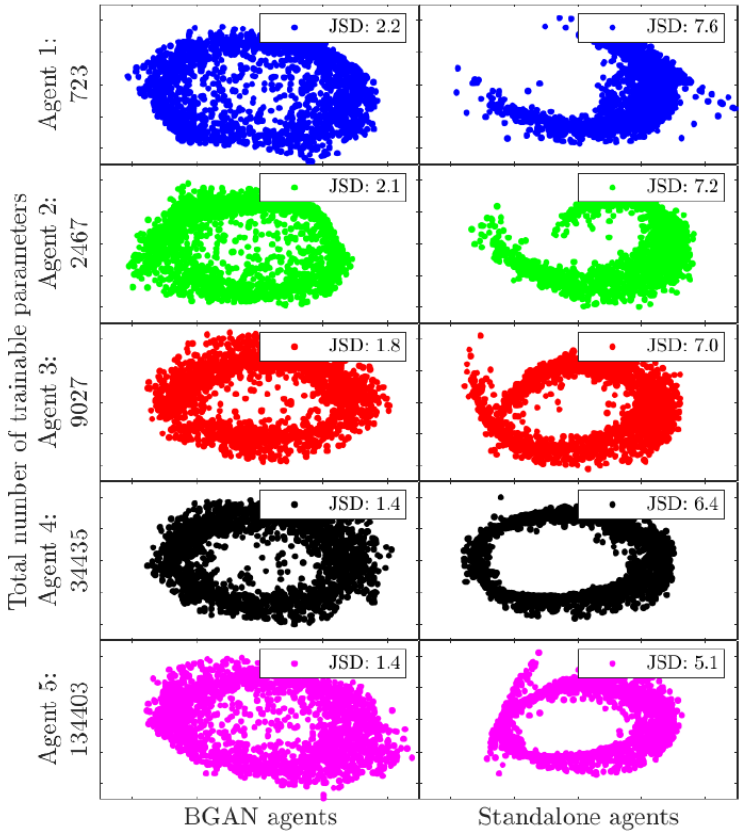




- BGAN can compensate lack of data through brainstorming
- GAN with 10 samples (JSD = 24) vs 10 agents (JSD = 13)
- Normalized generated samples of standalone GAN and BGAN with 5 and 10 agents where each agent has access to 10, 50, 100, 500, or 1000 data samples.

Simulation Results

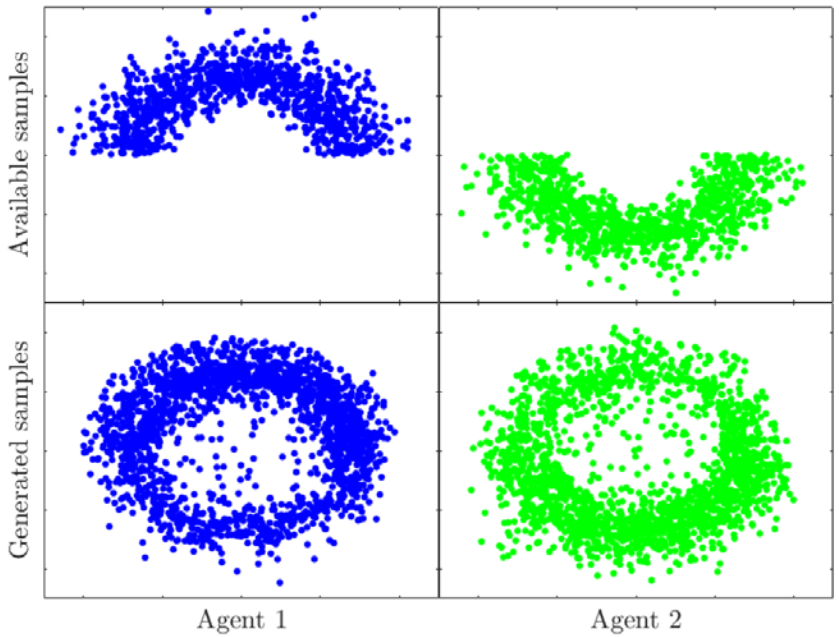




- If the agent owns enough samples, brainstorming does not give significant gains
- Each agent has a different number of neurons, BGAN improves upon standalone (this is not possible with baselines like FL GAN)

Simulation Results

- If the agents own partial, non-overlapping data, can they figure out the entire distribution?



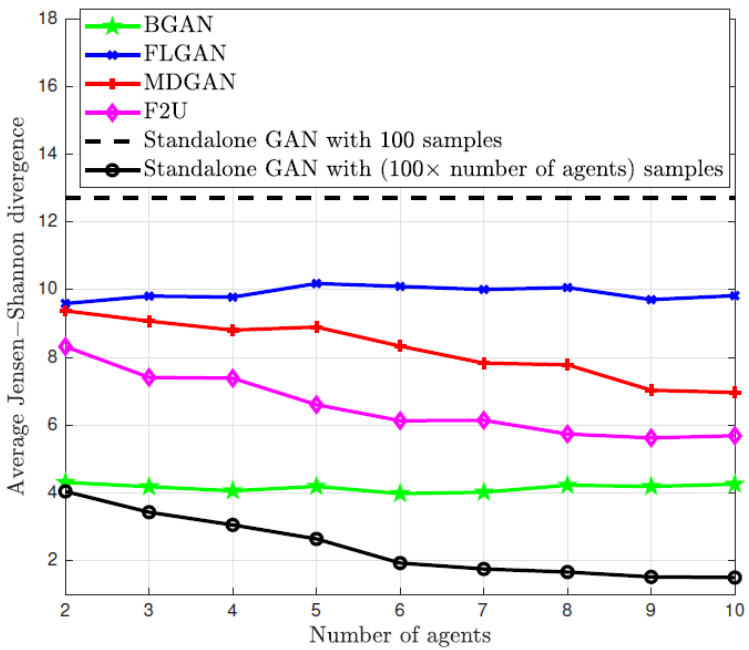
- Each agent owns part of the circle
- Each agent owns a single digit

Simulation Results

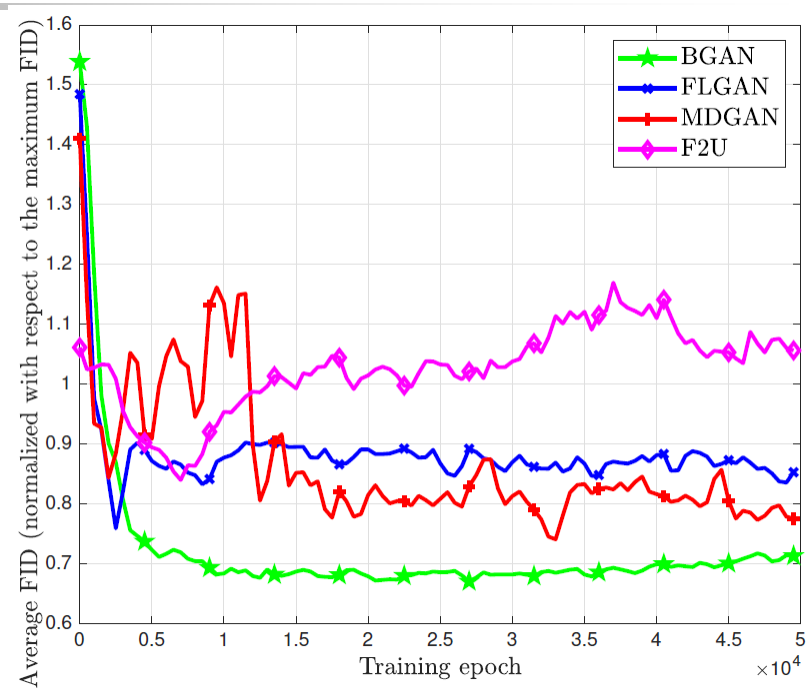
- If the agents own partial, non-overlapping data, can they figure out the entire distribution?

- CIFAR-10

Simulation Results



JSD comparison between BGAN, FLGAN, MDGAN and F2U on ring dataset



JSD comparison between BGAN, FLGAN, MDGAN, and F2U on the MNIST dataset

n	Number of agents
b	Batch size
$\lvert\boldsymbol{x}\rvert$	Data size
$\lvert\boldsymbol{\theta}_g\rvert$ $\lvert\boldsymbol{\theta}_d\rvert$	Neural network size

Architecture	Communication resources
BGAN	$\mathcal{O}(nb\lvert\boldsymbol{x}\rvert)$
MDGAN	$\mathcal{O}(n(b\lvert\boldsymbol{x}\rvert + \lvert\boldsymbol{\theta}_d\rvert))$
FLGAN	$\mathcal{O}(n(\lvert\boldsymbol{\theta}_g\rvert + \lvert\boldsymbol{\theta}_d\rvert))$
F2U	$\mathcal{O}(n(b\lvert\boldsymbol{x}\rvert + 1))$

Summary

- We showed that distributed GAN models with partial datasets that are distributed across devices can be devised with no centralized control
- What can we do with this next?
  - Enhance security/privacy
  - Distributed BGAN discriminator for inference
  - More sophisticated network connectivity and graphs
  - Applications to security (intrusion detection, GC'19)
  - Applications of BGAN to wireless networks (from channel modeling to resource allocation and vehicular networking)
- Let's see an example application

System Model

- Consider a set of I UAV base stations that are collecting channel information during mmWave wireless service and building a stochastic model to estimate the channel parameters
- We use a standard channel model:

$$\boldsymbol{H} = \sum_{l=1}^{L} \alpha_l \boldsymbol{a}_r(\phi_l^r)\boldsymbol{a}_t^H(\phi_l^t)$$

Number of paths | Channel gain | Antenna steering vectors

- Simple A2G mmW channel: $L = 1$
  - Massive MIMO array with narrow beam and perfect directional radiation

$$\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{y}, t, \phi^t, \phi^r) = \alpha(\boldsymbol{x}, \boldsymbol{y}, t, \phi^t, \phi^r)\boldsymbol{a}_r(\phi^r)\boldsymbol{a}_t^H(\phi^t)$$

Spatial-temporal parameters

Angle-of-arrival (AoA) and angle-of-departure (AoD)

System Model

- Based on a codebook of length *K*, the received pilot signal at a UE for the *k*-th beam training is:

$$r_k = \sqrt{P}\boldsymbol{q}_k^H \boldsymbol{H}_k \boldsymbol{w}_k + \boldsymbol{q}_k^H \boldsymbol{n}$$

Transmit power | Combining vector | Beamforming vector

- By separating the channel gain and other parameters:

$$\begin{aligned} r_k &= \sqrt{P}(\boldsymbol{w}_k^T \otimes \boldsymbol{q}_k^H)\mathrm{vec}(\boldsymbol{H}_k) + \boldsymbol{q}_k^H \boldsymbol{n} \\ &= \sqrt{P}(\boldsymbol{w}_k^T \otimes \boldsymbol{q}_k^H)[\boldsymbol{a}_t^*(\boldsymbol{w}_k) \otimes \boldsymbol{a}_r(\boldsymbol{q}_k)]\alpha(\boldsymbol{x}, \boldsymbol{y}, t, \boldsymbol{\phi}_k) + \boldsymbol{q}_k^H \boldsymbol{n} \\ &= \beta_k \alpha_k(\boldsymbol{x}, \boldsymbol{y}, t, \boldsymbol{\phi}_k) + \boldsymbol{q}_k^H \boldsymbol{n} \end{aligned}$$

- The estimated channel gain (data sample)

$$\tilde{\alpha}_k(\boldsymbol{x}, \boldsymbol{y}, t, \boldsymbol{\phi}_k) = r_k \beta_k^{-1} = \alpha_k(\boldsymbol{x}, \boldsymbol{y}, t, \boldsymbol{\phi}_k) + \tilde{n}_k$$

- Dataset: $\mathcal{S}_i = \{\boldsymbol{s}_n, \boldsymbol{\phi}_n\}_{n=1,\cdots,S_i} = \{\boldsymbol{x}_n, \boldsymbol{y}_n, t_n, \tilde{\alpha}_n, \boldsymbol{\phi}_n\}_{n=1,\cdots,S_i}$

Conditional GAN (CGAN) framework

- Local CGAN channel model
  - **Condition sampler** draws a AoA-AoD pair

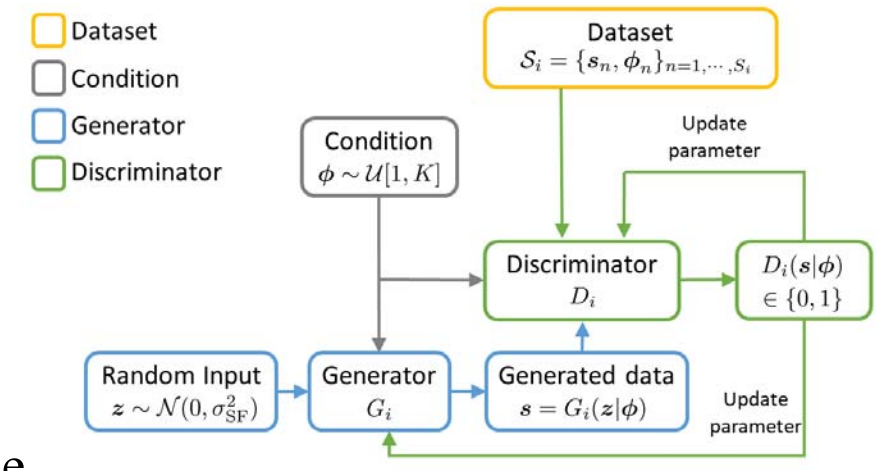
$$\phi \sim \mathcal{U}[1, K]$$

  - **Generator** creates samples given a random input and the condition

$$\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{y}, t, \alpha) \sim G_i(\boldsymbol{z}, \boldsymbol{\theta}_i^g | \boldsymbol{\phi})$$

  - **Discriminator** evaluates each sample

$$D_i(\boldsymbol{s}, \boldsymbol{\theta}_i^d | \boldsymbol{\phi}) \in \{0, 1\}$$



- Zero-sum game with conditions:

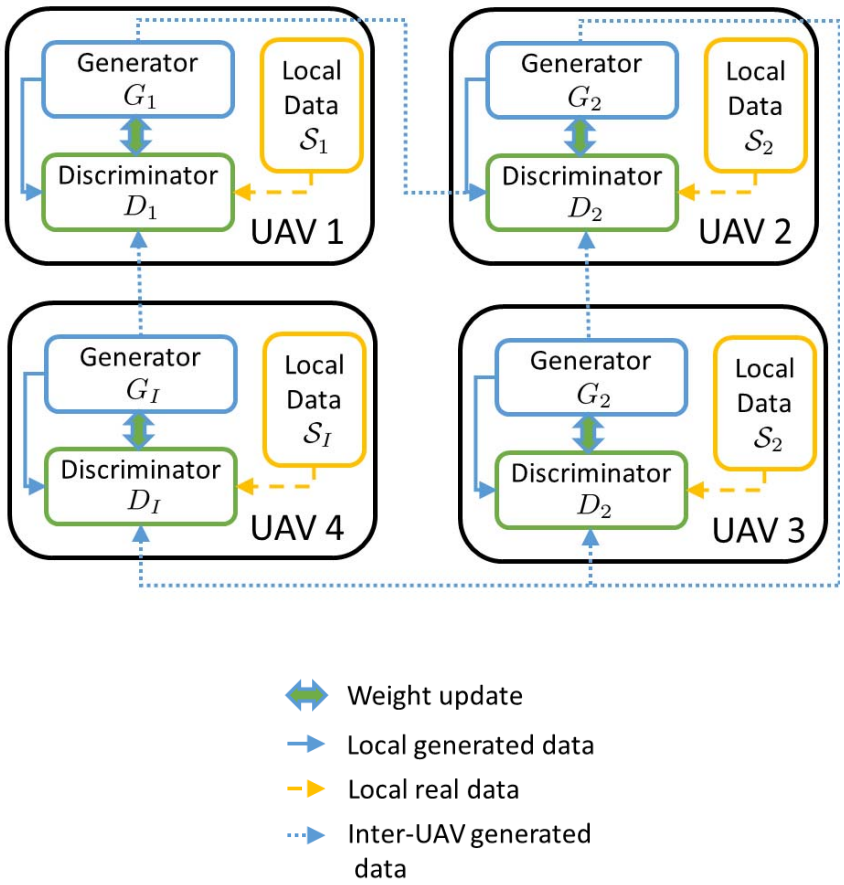
$$V_i(D_i, G_i) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{s} \sim f_i} \left[\log D_i(\boldsymbol{s}|\boldsymbol{\phi}_k) \right] + \mathbb{E}_{\boldsymbol{z} \sim f_i^z} \left[\log(1 - D_i(G_i(\boldsymbol{z}|\boldsymbol{\phi}_k))) \right]$$

Discriminator aims to maximize the value function

Generator aims to minimize this value

BGAN-based UAV Channel Modeling

- Can we create a spatio-temporal map through UAV collaboration?
  - Assume stable air-to-air links
- Each UAV has its local dataset, a generator, and a discriminator
- Each UAV shares its generated channel samples with other UAVs, by forming a distributed learning network
- All generators collaboratively generate channel samples to fool all of the discriminators
- Reformulated with BGAN



BGAN for UAV Communications

Q. Zhang, A. Ferdowsi, W. Saad, and M. Bennis, "Distributed Conditional Generative Adversarial Networks (GANs) for Data-Driven Millimeter Wave Communications in UAV Networks", *IEEE Transactions on Wireless Communications*, to appear, 2021.

UAV Communication Network

- The UAV network structure is a directed graph: $\mathcal{G} = (\mathcal{I}, \mathcal{E})$
  - $\mathcal{I}$ is the set of UAVs
  - Each edge $e_{ij} \in \mathcal{E}$ is an air-to-air communication link
- At each iteration, each UAV shares ηS_i generated samples
- **Theorem:** Probability that the learning converges after iteration T:

Training error of the local CGAN at each UAV

Maximum shortest path

Number of in-degree UAVs

Shortest loop-path

$$p_{\mathcal{G}}(T) = \begin{cases} 0 & 0 < T < l^{\max}, \\ \frac{[(1-\epsilon)\eta]^{l^{\max}}}{(1+N\eta)^{l^{\max}-1}} & T = l^{\max}, \\ p_{\mathcal{G}}(l^{\max}) + \sum_{i=l^{\max}+1}^{T} \left[\prod_{j=l^{\max}}^{i-1} \left(1 - \frac{[(1-\epsilon)\eta]^{l^{\max}}}{(1+N\eta)^{j-1}}\right)\right] \frac{[(1-\epsilon)\eta]^{l^{\max}}}{(1+N\eta)^{i-1}} & l^{\max} < T < l^{\max} + l^{\min}_{\text{loop}}, \end{cases}$$

and for $T \geq l^{\max} + l^{\min}_{\text{loop}}$, $p_{\mathcal{G}}(T) = p_{\mathcal{G}}(l^{\max} + l^{\min}_{\text{loop}} - 1) +$

Acceleration coefficient

$$\sum_{i=l^{\max}+l^{\min}_{\text{loop}}}^{T} \left[\prod_{j=l^{\max}+l^{\min}_{\text{loop}}-1}^{i-1} \left(1 - \frac{[(1-\epsilon)\eta]^{l^{\max}}}{(1+N\eta)^{j-1}} \prod_{k=l^{\max}+l^{\min}_{\text{loop}}-1}^{j} \gamma(k)\right)\right] \frac{[(1-\epsilon)\eta]^{l^{\max}}}{(1+N\eta)^{i-1}} \prod_{l=l^{\max}+l^{\min}_{\text{loop}}}^{i} \gamma(l),$$

Convergence time

- The probability of convergence decreases as the length of the maximum shortest-path $l^{\max}$ increases.

Minimize $l^{\max}$

- The number of iterations $T_{\mathcal{G}}$ required for convergence of the distributed CGAN with a confidence level: $p_{\tau} \in (0, 1)$

$$p_{\mathcal{G}}(T_{\mathcal{G}} - 1) < p_{\tau} \leq p_{\mathcal{G}}(T_{\mathcal{G}})$$

- Then we can quantify this convergence time:

$$C(\mathcal{G}) = (t_{\tau} + t_{\epsilon}) \cdot T_{\mathcal{G}}$$

Upper limit of A2A transmission time for the generated channel samples

Upper time limit for the local CGAN training at each UAV with an error rate ϵ

Problem Formulation

- The objective of each UAV is to choose the optimal UAV to whom it sends its generated channel samples, so as to minimize the convergence time over its maximum shortest-path, i.e.:

$$\min_{\mathcal{G}} \quad C(\mathcal{G})$$

$$\text{s.t.} \quad \sum_{e_{ij}\in\mathcal{E}} P_{ij} \leq P_{\max}, \quad \forall i \in \mathcal{I},$$

UAV transmit power

$$P_{ij}h_{ij}/\sigma^2 \geq \tau, \quad \forall e_{ij} \in \mathcal{E},$$

SIR threshold

$$\eta S_i \rho / R_{ij} \leq t_\tau, \quad \forall e_{ij} \in \mathcal{E},$$

Transmission time threshold

$$\exists E_{i,j} \subset \mathcal{E}, \quad \forall i, j \in \mathcal{I},$$

Strongly connected network

$$I \leq |\mathcal{E}| \leq B,$$

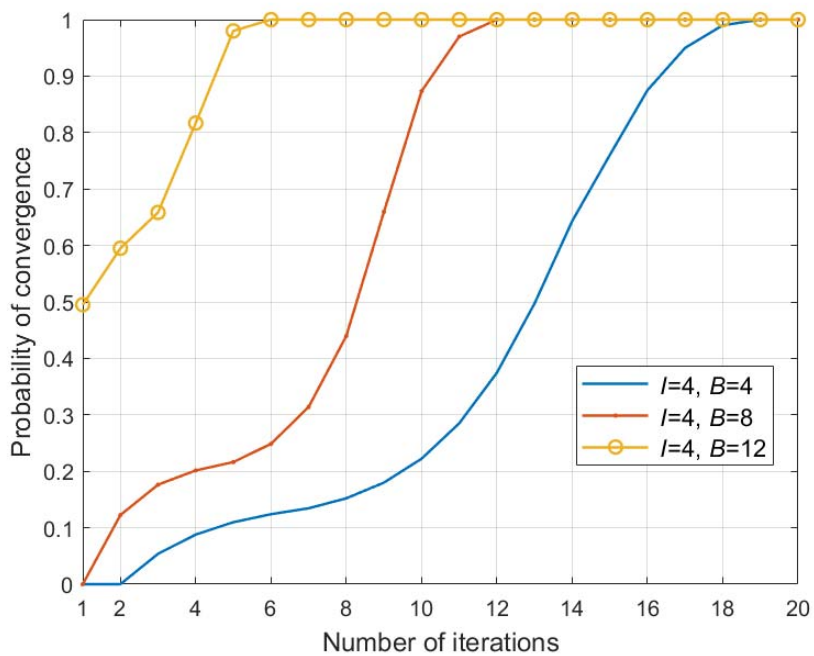
No interference

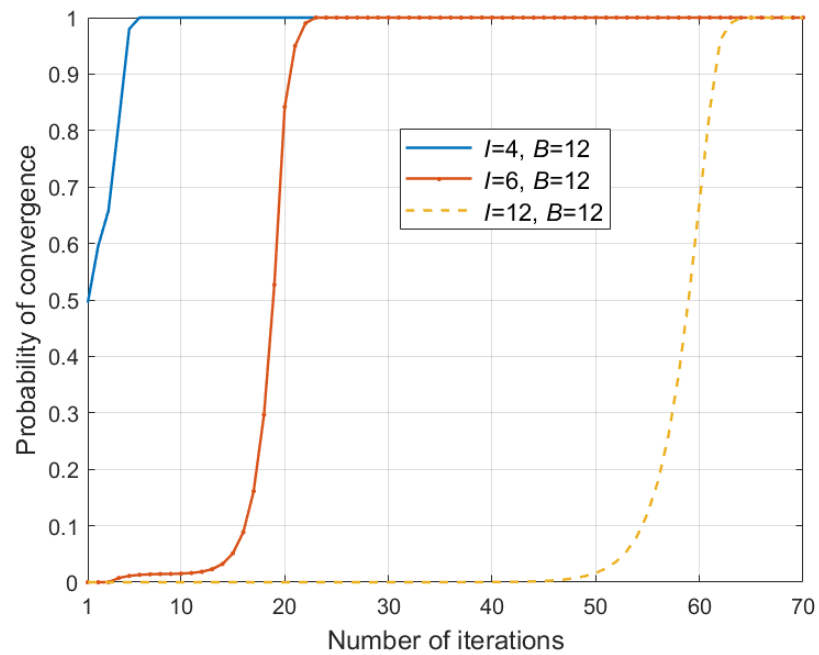
Centralized Problem ! We propose a distributed solution

$$I = |\mathcal{E}| \leq B$$

- We show that the distributed solution can be characterized with reasonable complexity

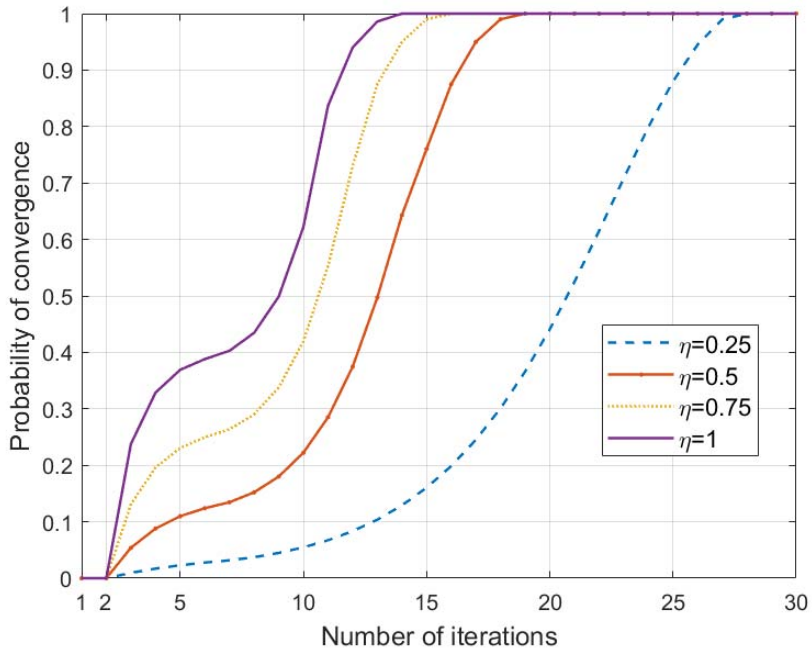
Simulation Results

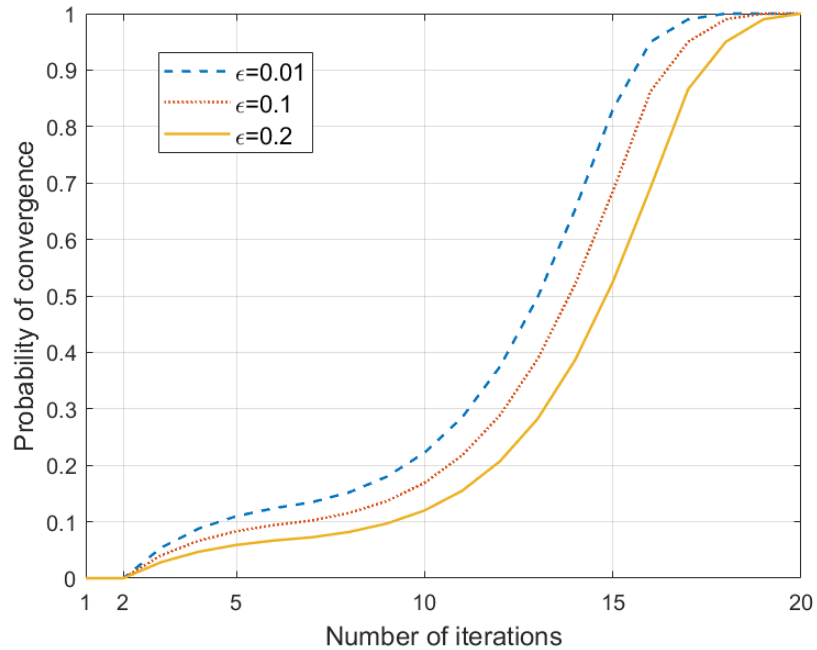




- The learning rate increases given more communication resources for A2A data transmissions, but it decreases with more UAVs.
- A larger network size requires more communication resources per UAV to guarantee an efficient learning rate.

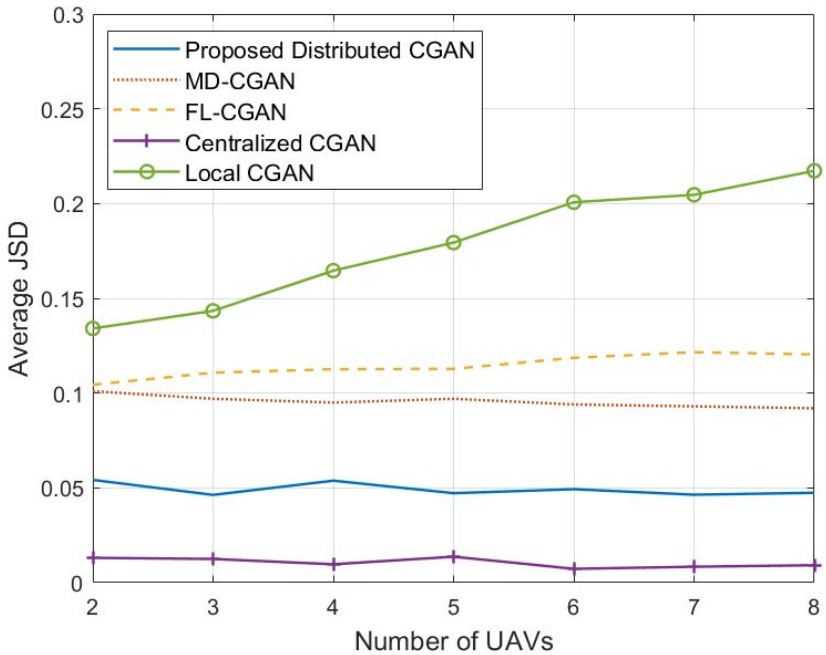
Simulation Results

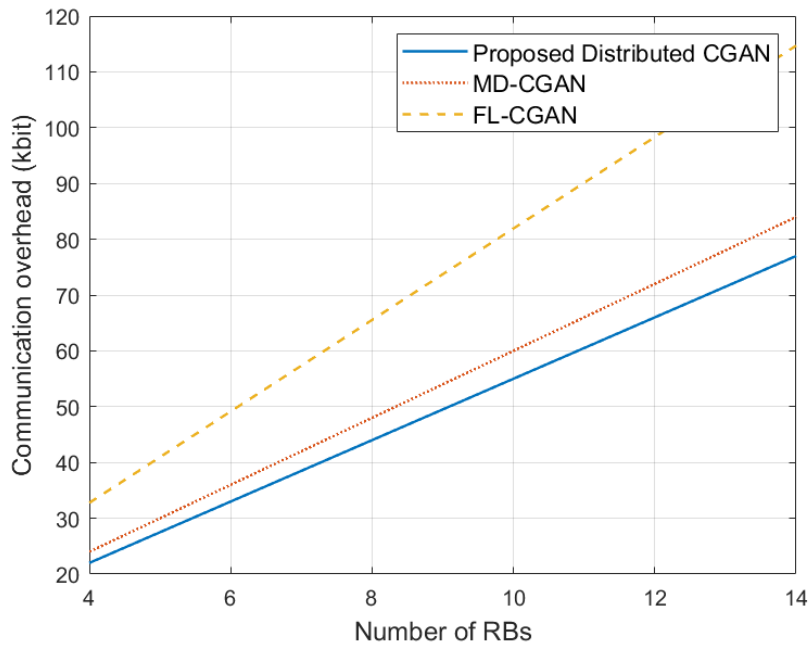




- By sharing more generated samples in each iteration, and given a smaller training error at each UAV, the proposed distributed CGAN approach yields a higher learning rate
- The proposed distributed CGAN method is robust to the local training error at each UAV (effect of training error is not significant)

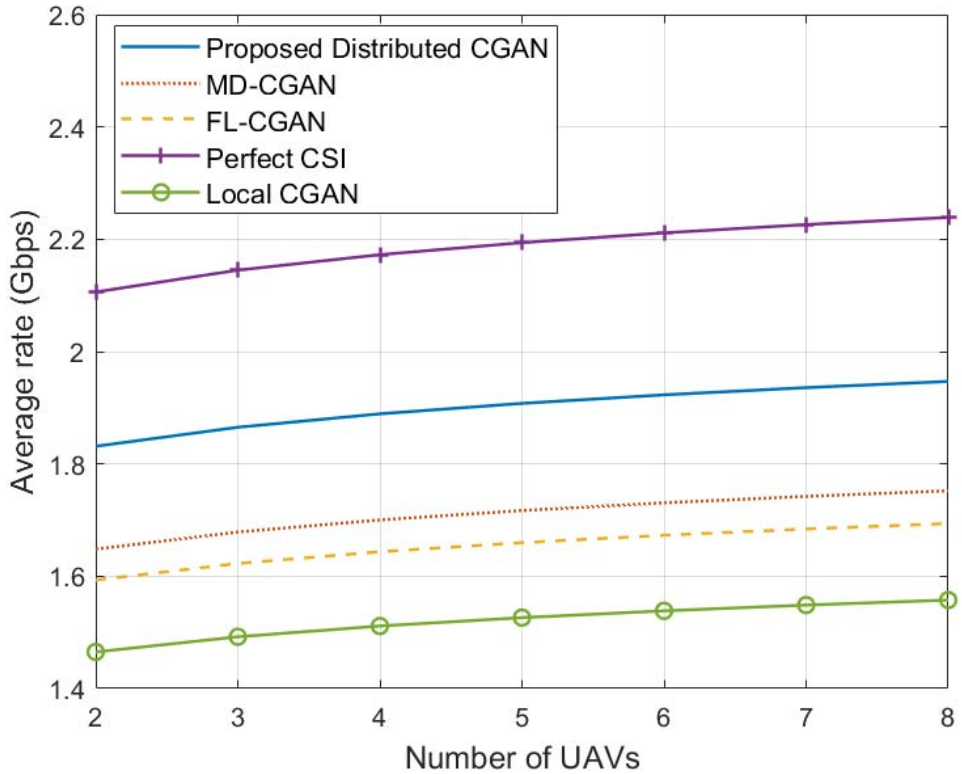
Simulation Results





- Our proposed approach (built on BGAN) yields the highest modeling accuracy and the lowest communication overhead compared with a local CGAN, a federated CGAN, and a multi-discriminator distributed CGAN
- Local GAN is worse for larger networks (smaller data per UAV)

Simulation Results



- Our approach also clearly improves the average data rate compared to the baselines

Quick glance at:

Experienced Deep Reinforcement Learning for Reliable Communications

A. T. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced Deep Reinforcement Learning with Generative Adversarial Networks (GANs) for Model-Free Ultra Reliable Low Latency Communications", *IEEE Transactions on Communications, 2020.*

Reliable Low Latency

- URLLC has been around for a while but prior art…
  - Focused on *IoT sensors* (uplink) – **autonomous vehicles/drones are different (downlink?)!**
  - Assumes *known models* for traffic (M/M/1 etc.)– **latency has many components, hard to model!**
  - Considers slow deep reinforcement learning (DRL) – **learning in URLLC must handle extreme, rare conditions!**
  - Assumes *rate* can be *ignored* – **autonomous systems may need some form of rate guarantees!**
  - Comes up with *arbitrary latency numbers* – **latency is driven by the autonomous vehicles control!**
- ***Problem 1: Experienced DRL for Model-free URLLC***
- ***Problem 2: Control meets Communication (not in this talk)***

Brief System Model

- Consider the downlink of a single-cell wireless network whose base station is sending latency-sensitive control message to autonomous vehicles
- We consider a downlink OFDMA system with resource blocks that must be allocated
- We define reliability as the probability of end-to-end packet delay exceeding a threshold
- We do not make any assumptions for a delay model
  - Delay is intrinsically hard to model, most models are often unrealistic and have some hidden drawbacks
  - Delay has many components, hard to model their combination precisely

Problem Formulation

- Our goal is to solve the following classical problem

$$\min_{p_{ij},\rho_{ij}} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{i=1}^{N} \sum_{j=1}^{K} p_{ij}(\tau),$$

$$\text{s.t} \quad \Pr\{D_i > D_i^{\max}\} < 1 - \gamma_i^*, \quad \forall i \in \mathcal{N},$$

$$r_i(t) > \lambda_i(t)\beta_i(t), \quad \forall i \in \mathcal{N}, \quad \forall t$$

$$p_{ij}(t) \geq 0, \quad \rho_{ij}(t) \in \{0,1\},$$

$$\forall i \in \mathcal{N}, \quad \forall j \in \mathcal{K}, \quad \forall t,$$

$$\sum_i \rho_{ij}(t) = 1, \quad \forall j \in \mathcal{K}, \quad \forall t.$$

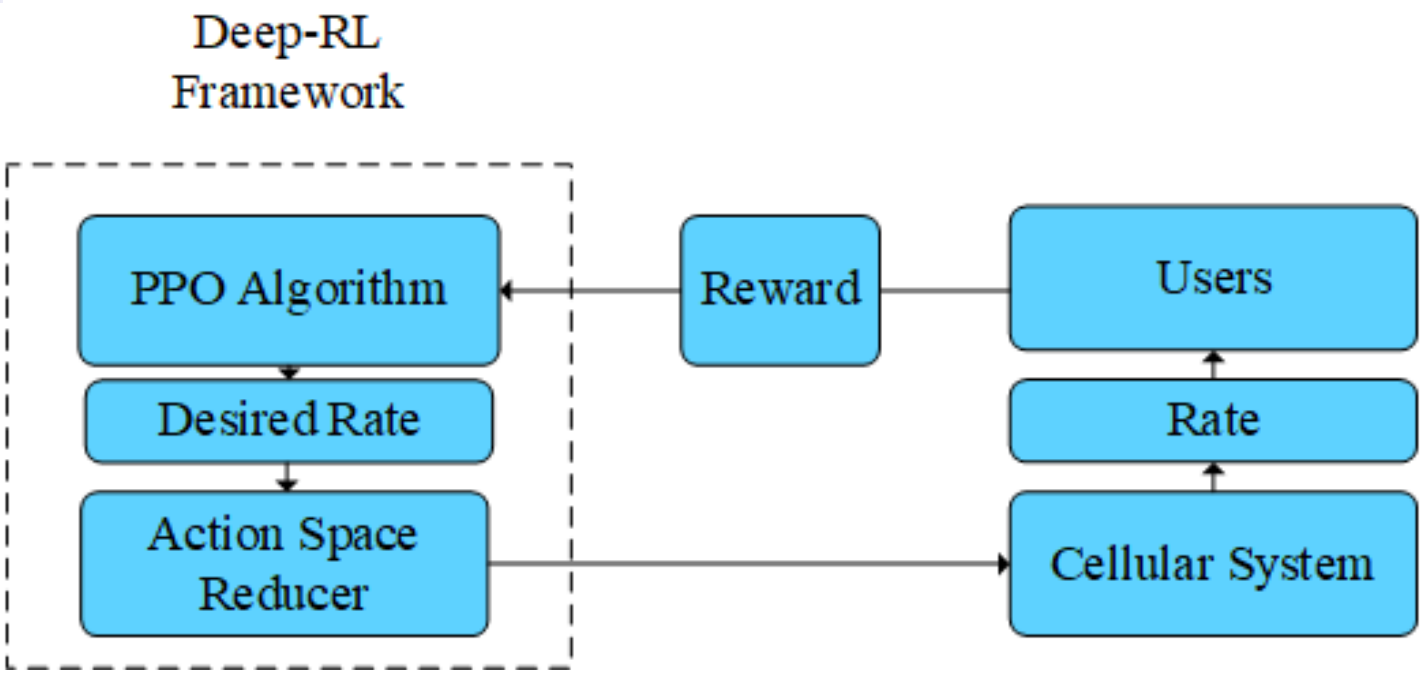
Reliability constraint

Rate constraint

Feasibility constraints

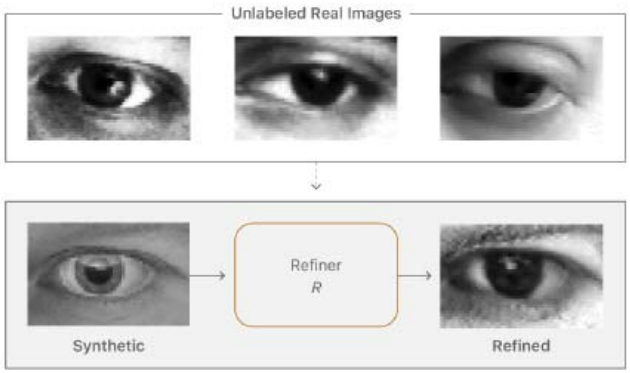
- Explicit rate guarantees imposed
- Challenging to solve because of our model-free assumption

Deep-RL for Model-Free URLLC

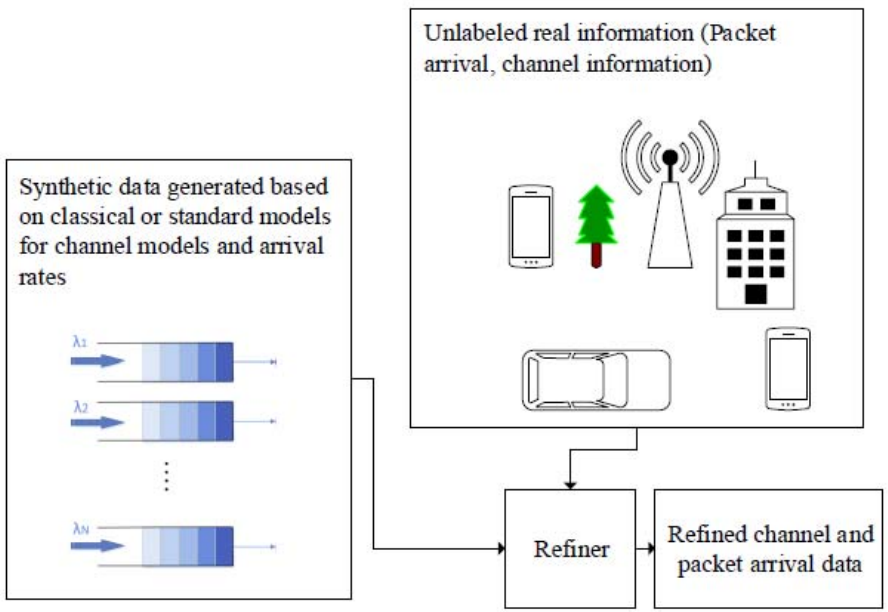


- **State space:** number of packets, packet size, and channel gains
- **PPO:** Proximal policy optimization determines target rates
- **Action space reducer:** Deep-RL made tractable
- **But deep RL is inherently unreliable in face of "rare but extreme events" => Create experience**

Experienced Deep RL



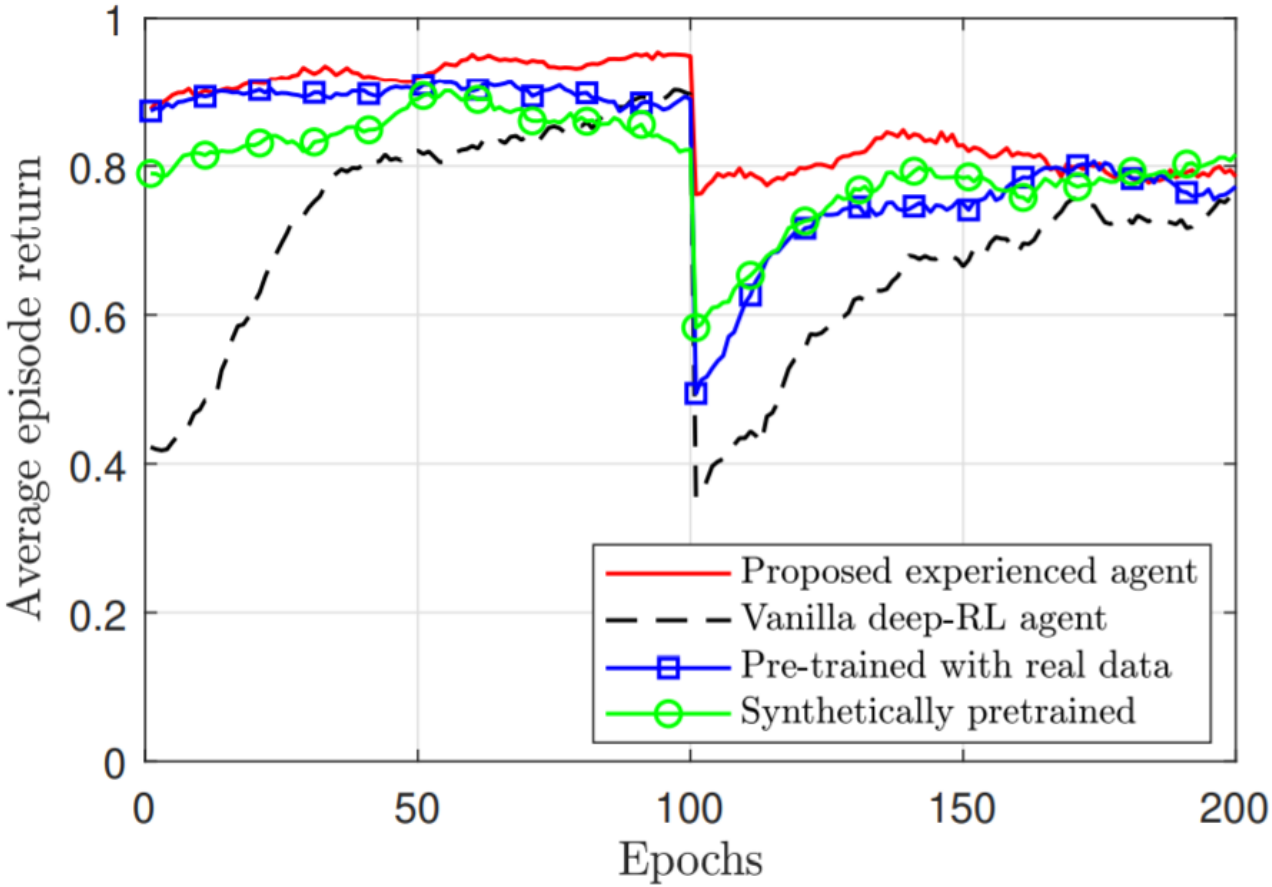
a)



b)

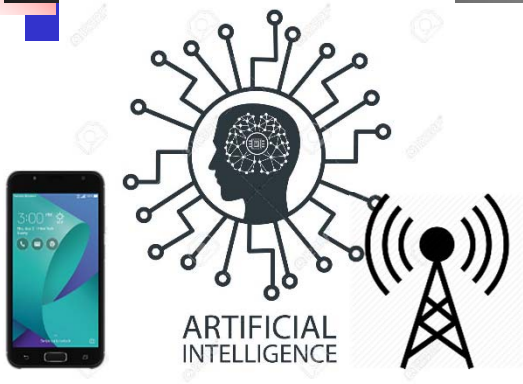
- GAN-based refiner
  - Proposed by Apple for computer vision
- Inputs
  - Unlabeled real data
  - Synthetic model data
- Output
  - Refined (and larger) dataset that includes new network conditions (extreme events) that can train your deep RL

Simulation Results



- Experience allows a very smooth handling of extreme events compared to vanilla deep-RL

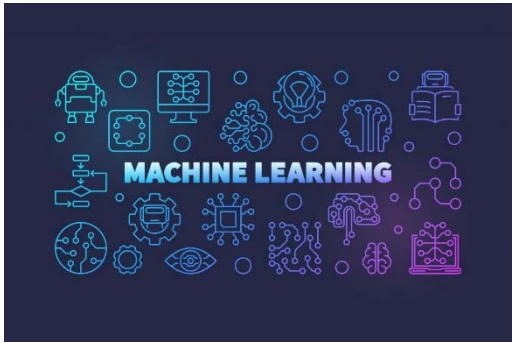
Other research areas



- 5G/6G/IoT systems
  - Reliable, low latency comm. with ML/GAN
  - Terahertz/RIS
  - Semantic communications

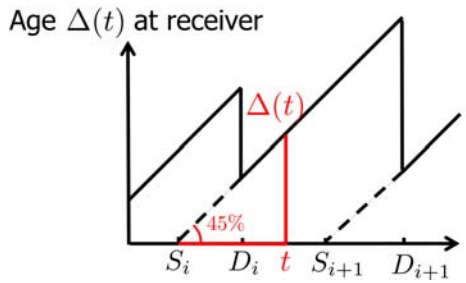
- AI-enabled XR
  - User experience
  - AI for wireless AR/VR
  - Holography

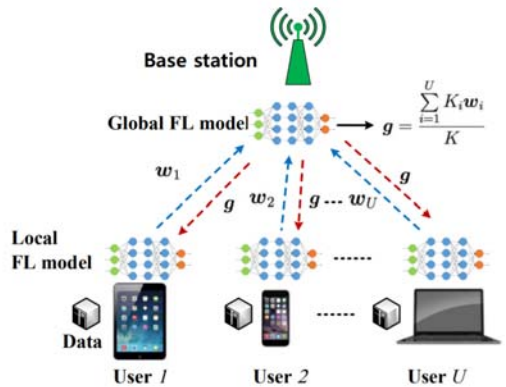
- Connected drones and autonomous vehicles
  - Distributed learning and control
  - Wireless connectivity
  - CPS Security



- Reliable, generalizable, distributed learning
  - Reliable machine learning
  - Meta-learning and training-free learning
  - Distributed and multi-agent learning
  - Beyond federated learning

Other research areas





- Age of information
  - Performance analysis of Internet of Things systems with age of information considerations
  - Information management
- Communications for learning
  - Impact of wireless factors on federated learning
  - Computing, energy, & comm
- Smart cities
  - Big data for smart city optimization
  - Air pollution
  - Security

- Game theory
  - Foundations
  - Applications to CPS, security, policy, wireless

- CPS security
- Blockchains
- Quantum comm/ML

Conclusions

- Distributed learning is an exciting area, particularly when merged with wireless
- Distributed learning is not just federated learning
  - Distributed generative models is but one example
- Generative models will play an important role in future machine learning algorithms
- BGAN is the first fully distributed GAN
- Abundant field of applications
  - Most networks are inherently multi-agent systems => we need more fundamentals that are tailored towards networking problems

Finally….

Thank You, Questions?