CLSAC 2020 Student Presentation

Privacy Preserving, Distributed, and Verifiable Machine Learning for COVID-19 Identification using Zero-Knowledge Proofs

Zachary DeStefano

Los Alamos National Laboratory

A-4: Advanced Research in Cyber Systems

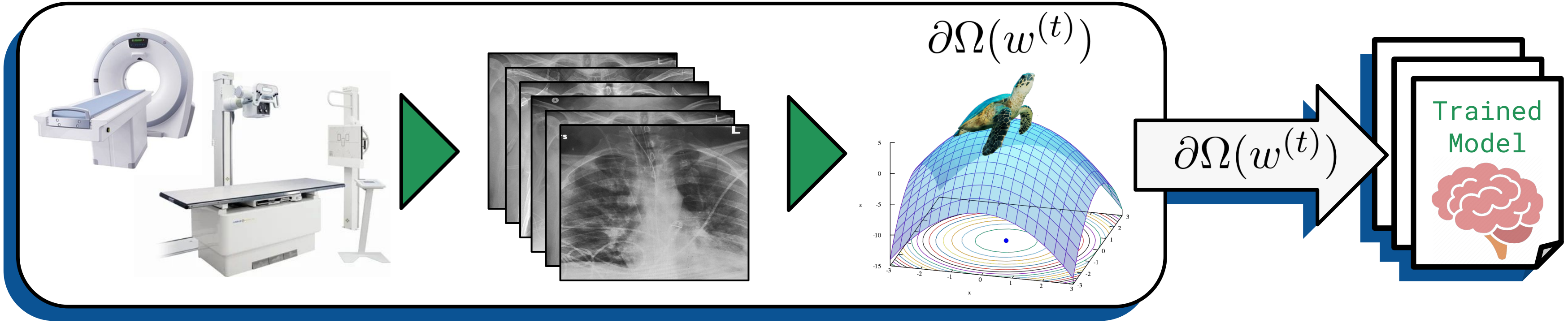
Mentor: Michael J. Dixon

October 6th, 2020

LA-UR-20-27590

Current Problem: Privacy-Preserving Medical ML

- There are several limits to current medical data science research:
  - Due to privacy concerns, it is difficult for researchers to gain access to patient data
  - When data is distributed to researchers, it needs to be explicitly anonymized and handled according to certain procedures to ensure HIPAA compliance
  - If researchers provide medical facilities with data analysis programs to run locally, they have no way of verifying the results



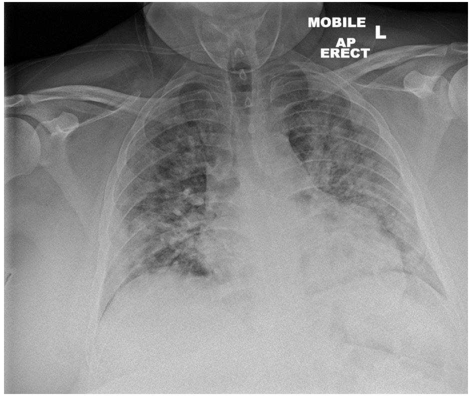
- How can we perform **distributed** medical machine learning in a **privacy-preserving** and **verifiable** way?

Application: Verifiable Edge Training for COVID-19

Public COVID datasets are hit or miss:

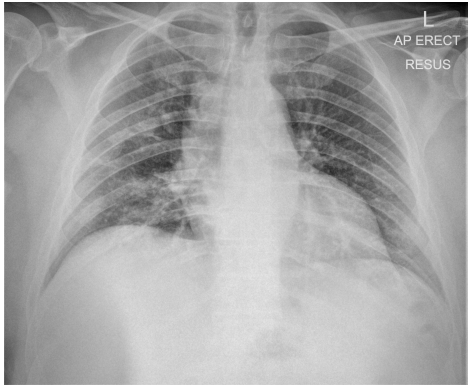
- Very limited, possibly due to health privacy concerns and infrastructure
- Inputs are not standardized
- There is a tradeoff between patient privacy and model accuracy

Edge training provides an agile approach for fast data science (particularly useful in crisis situations) by providing access to data without going through normal slow approval channels



Age / Gender:
33 / Male

COVID-19 Status:
positive



Age / Gender:
50+ / Male

COVID-19 Status:
negative

Solution: Zero-Knowledge Proofs and Verifiable Computation

Zero-knowledge proofs (ZKPs) allow us to prove that a claim **IS** true without revealing **WHY** it is true, even if the prover is untrusted and malicious.

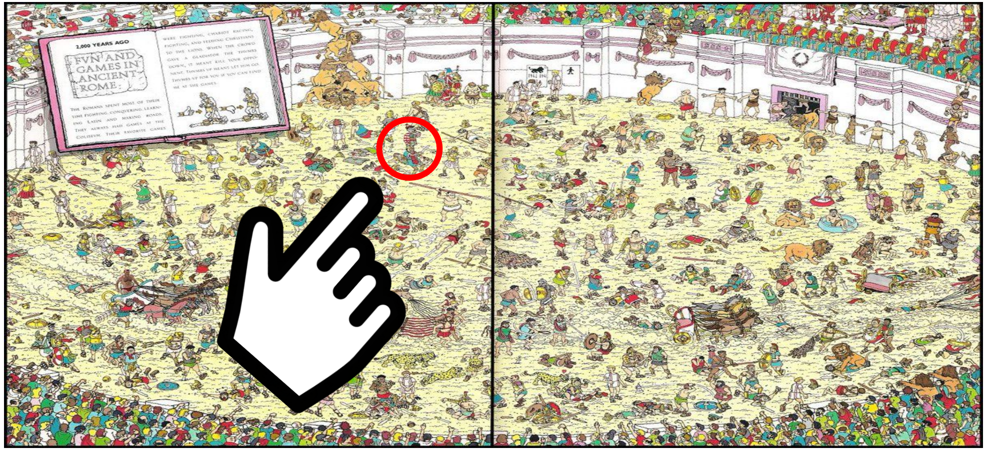
zk-SNARKs are special ZKPs that are ***tiny*** and ***non-interactive***

***Z**ero-*
***K**nowledge*
***S**uccinct*
***N**on-Interactive*
***A**rgument of*
***K**nowledge*

- ***zk-SNARKs** can be used to remotely **verify** the execution of programs*
- ***zk-SNARKs** are **constant** size and **verify** in **milliseconds***
- ***zk-SNARKs** reveal **NO information** about the claim they are proving (except that it is true)*

Proofs and Zero-Knowledge Proofs (π)

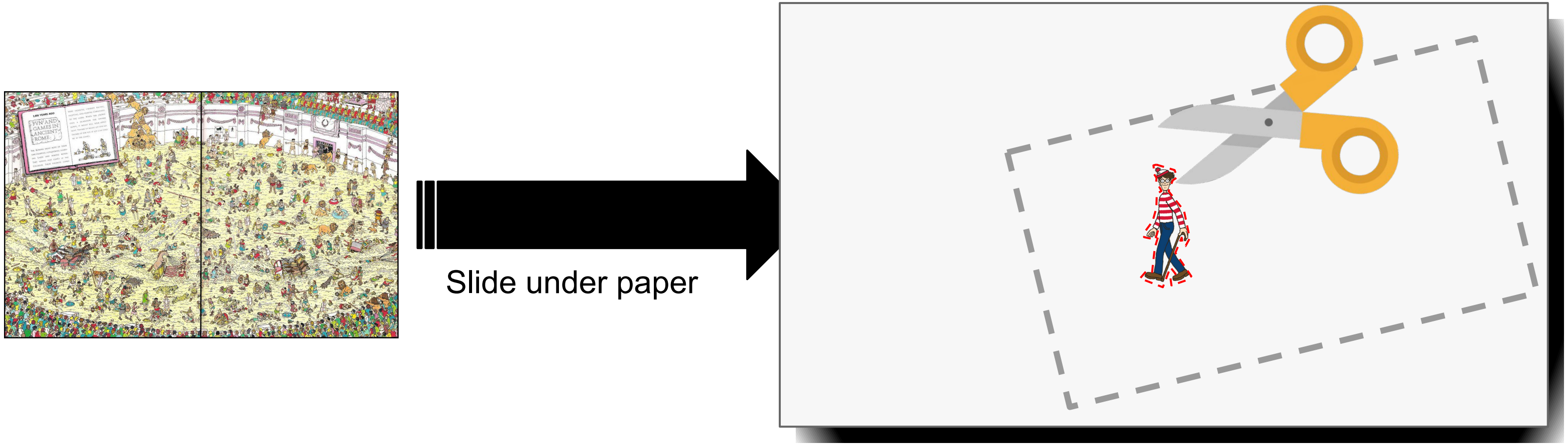
- A traditional proof for “Where’s Waldo?”
  - Point to Waldo to demonstrate you know where he is



  - This proves that you know where he is by showing someone else where he is

Proofs and Zero-Knowledge Proofs (π)

- A zero-knowledge proof (π) for “Where’s Waldo?”
  - Cut out a Waldo shaped hole in a much larger piece of paper
  - Place the hole over the location of Waldo

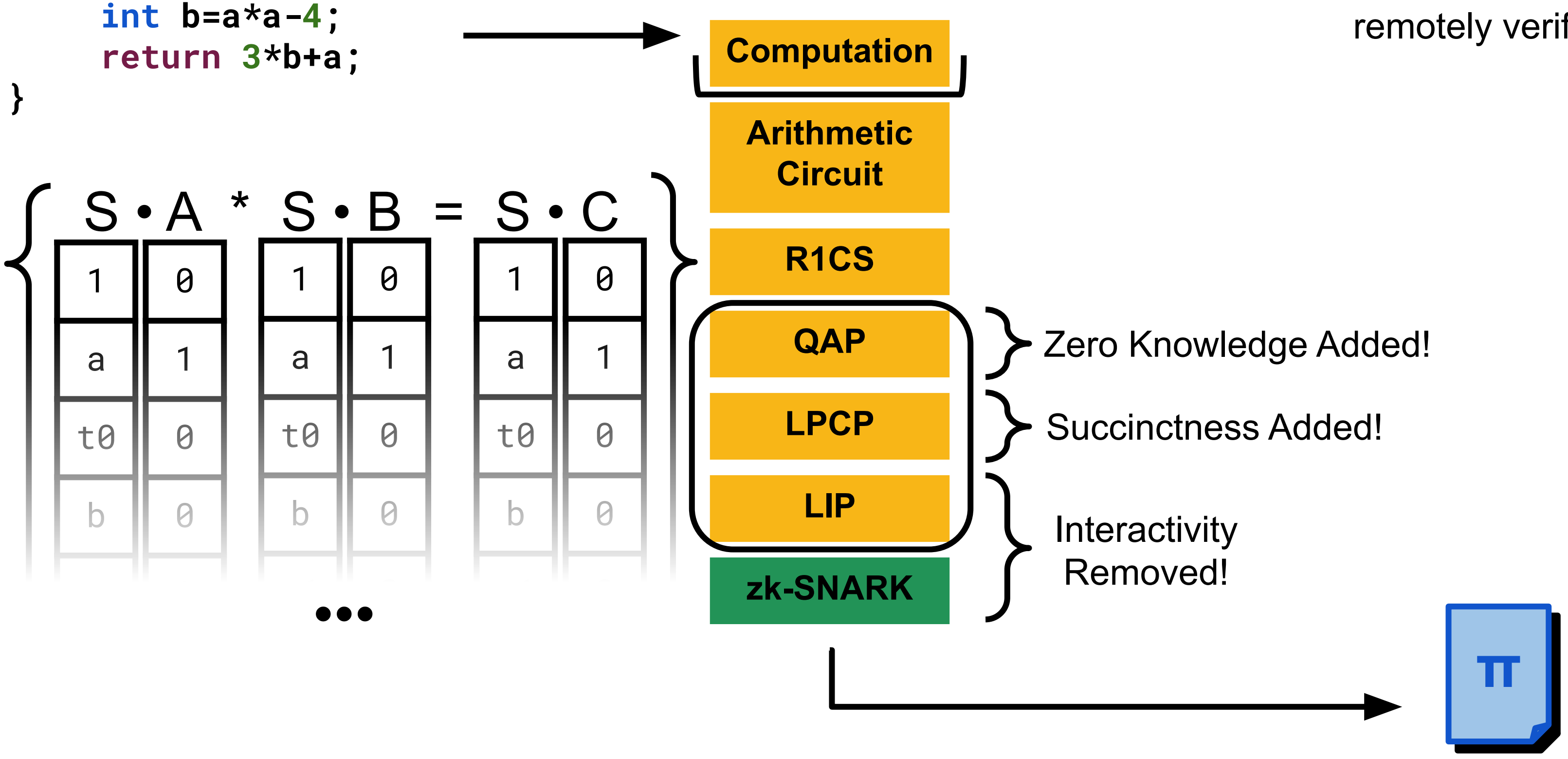


  - This proves that you know where Waldo is without giving any information to anyone else about where he is

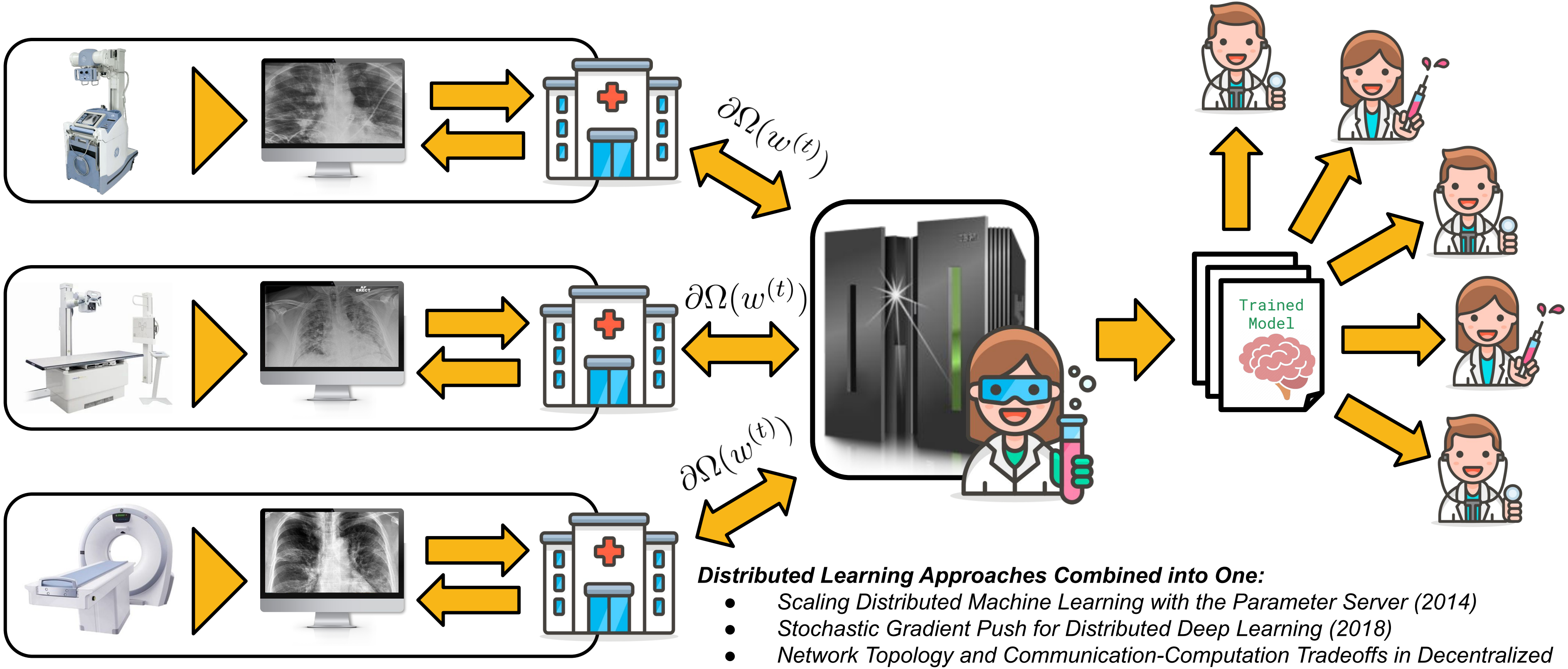
zk-SNARK Construction for Program Verification

```
int myFunction(int a) {
    int b=a*a-4;
    return 3*b+a;
}
```

zkSNARKs can be used to remotely verify program execution!



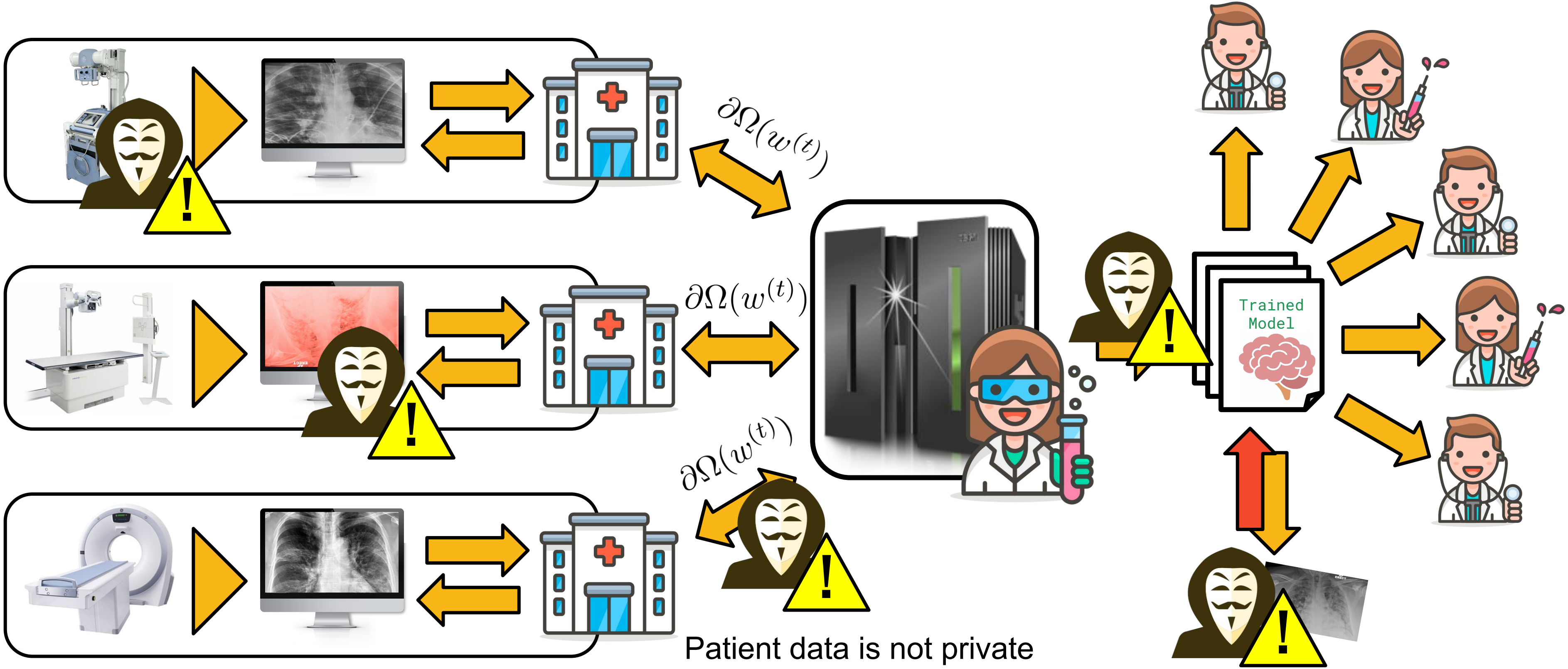
Distributed Learning with Subgradients



Distributed Learning Approaches Combined into One:

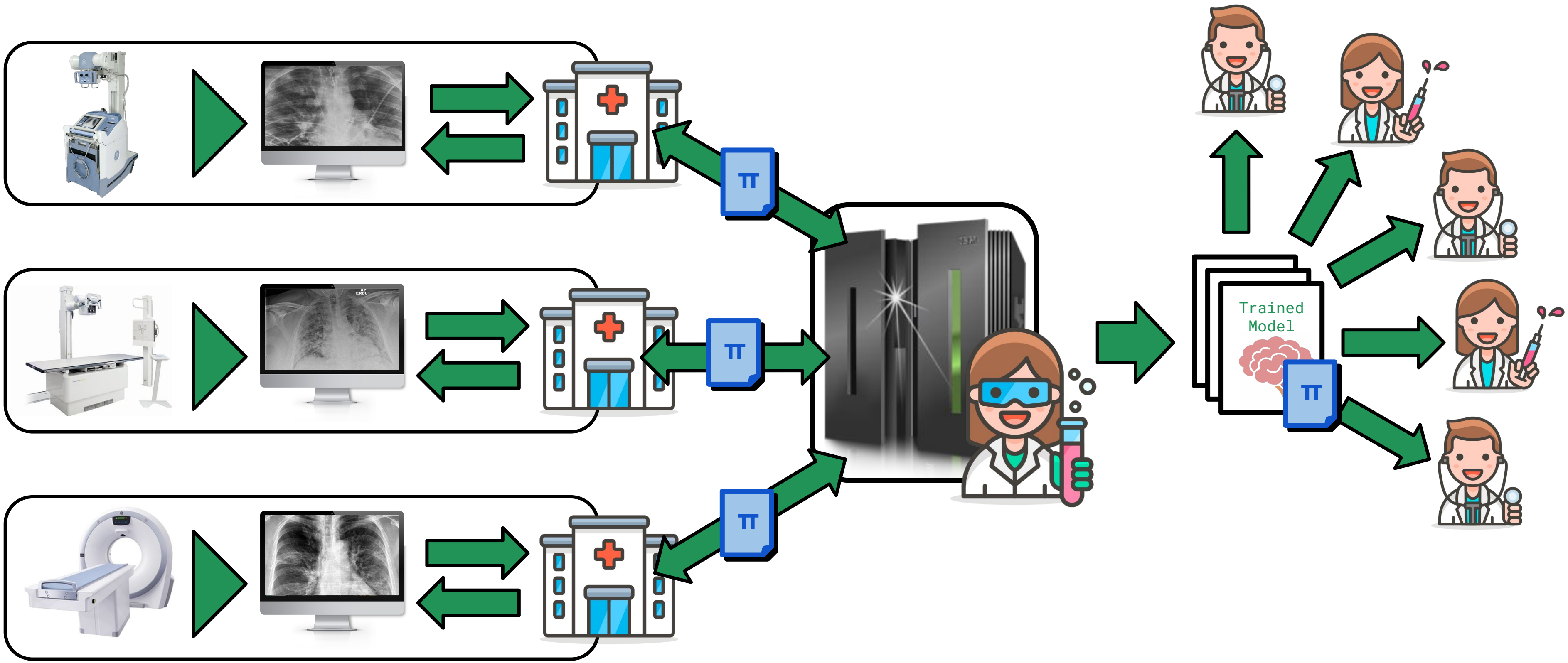
- *Scaling Distributed Machine Learning with the Parameter Server (2014)*
- *Stochastic Gradient Push for Distributed Deep Learning (2018)*
- *Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization (2018)*

Distributed Learning with Subgradients

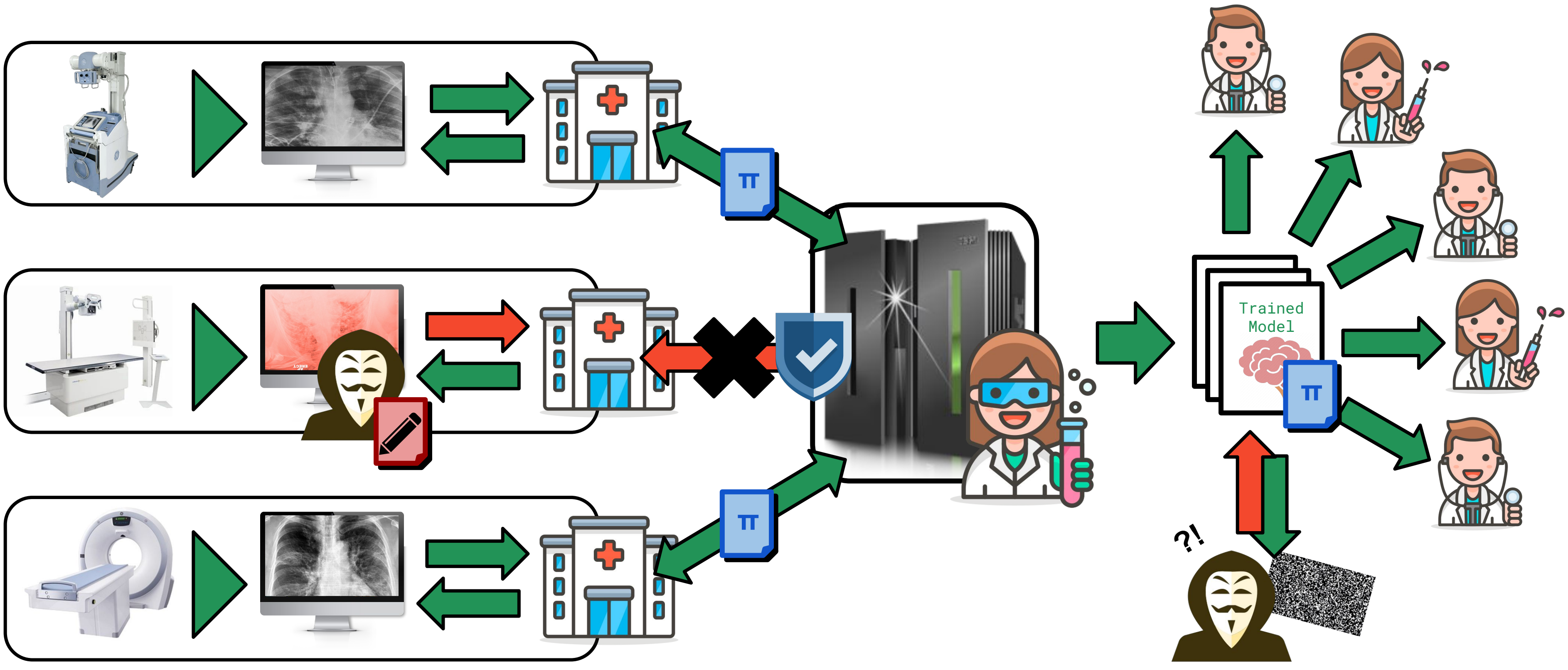


Patient data is not private
Tampering can occur anywhere unnoticed

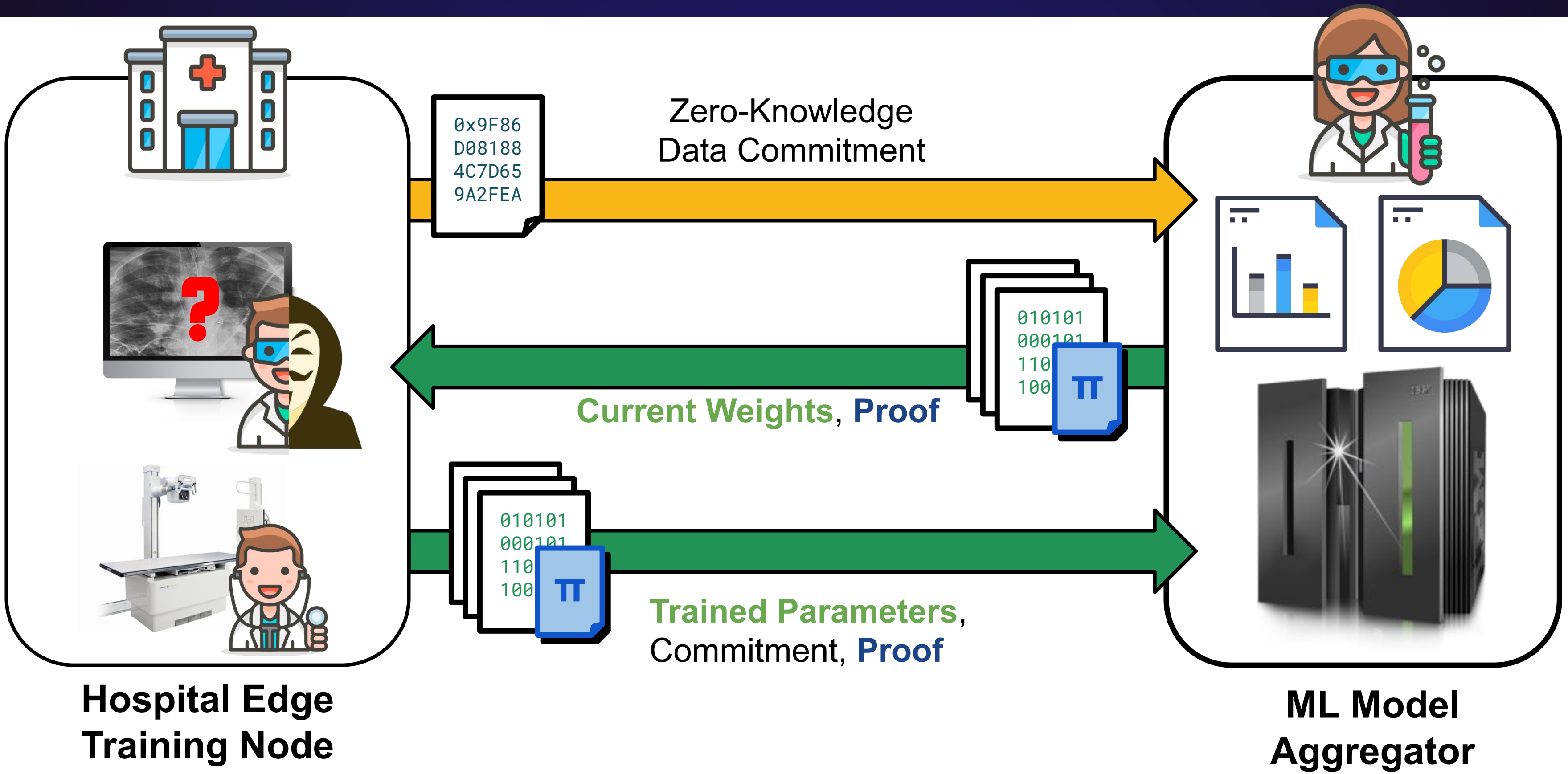
Verifiable, Distributed Learning with zkSNARKs



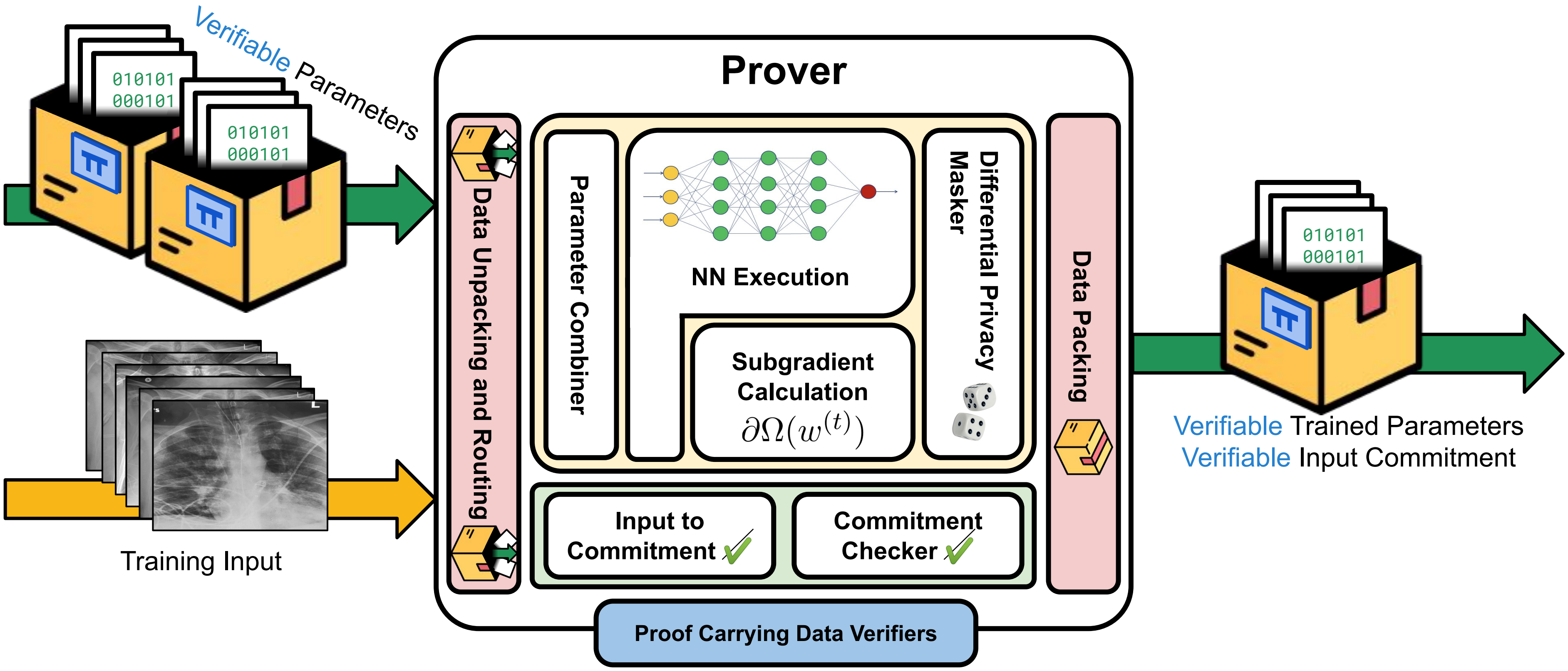
Verifiable, Distributed Learning with zkSNARKs



Distributed Learning Individual Interaction



Multi-Mode Distributed Learning Prover



Framework Implementation

- Necessary gadgets written in `C++` and are used in zkSNARK construction using the `libsnark` compiler toolchain
- Over a dozen reusable, modular gadgets were written and optimized by hand.
- Multiple proofs were combined and telescoped using recursive proof composition

Gadget Examples:

- Fixed Point Maximum and Vector Maximum
- Fast Dot Product
- Fully Connected Layer Execution/Gradient Calculation/Backpropagation
- RELU Execution/Gradient Calculation
- MAXPooling Execution/Gradient Calculation
- Full Neural Network Training
- Differential Privacy Masking

In Progress: Gadgets for Convolutions

```
size_t dex=0;
temp_input_weights=std::vector<std::vector<pb_variable<Fielc
temp_input_signs  =std::vector<std::vector<pb_variable<Fielc
for(size_t i=0;i<output_height;i++){
  for(size_t j=0;j<output_width;j++){
    size_t idex=output_width*i+j;
    temp_input_weights[idex]=std::vector<pb_variable<FieldT>
    temp_input_signs  [idex]=std::vector<pb_variable<FieldT>
    for(size_t k=0;k<k_size;k++){
      for(size_t l=0;l<k_size;l++){
        size_t temp_dex=l+j*k_size+(k+(i*k_size))*width;
        temp_input_weights[idex][l+k*k_size]=input_weights[t
        temp_input_signs  [idex][l+k*k_size]=input_signs  [t
      }
    }
    for(size_t k=0;k<depth;k++){
      DOT_PROD_Gadgets[dex].reset(new NN_Fast_Dot_Product_Ga
      BIT_SELE_Gadgets[dex].reset(new NN_BIT_SELECT_Gadget<F
      dex++;
    }
  }
```

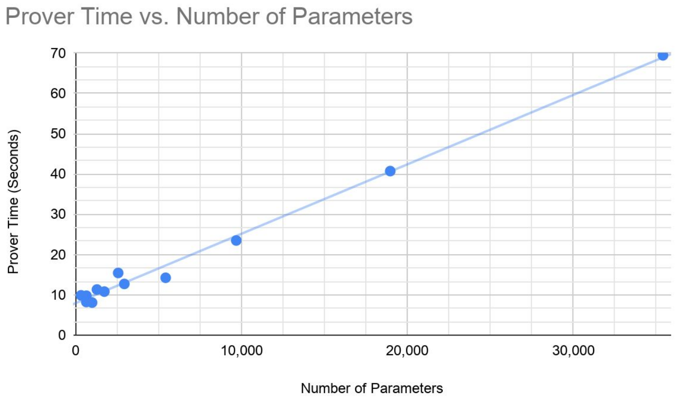
Figure. Relations enforcing properties that must be true are written in C++ and fed into libsnark

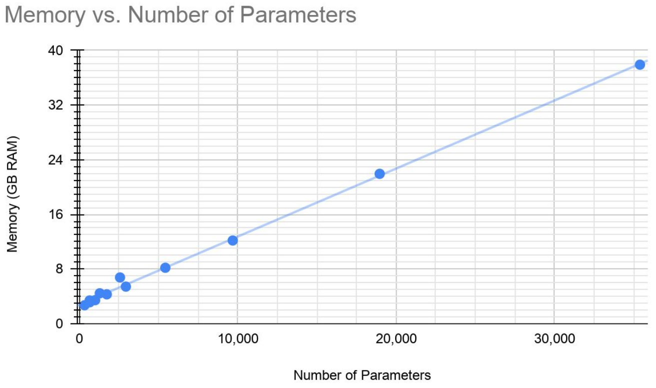
Benchmarking: Single Node Training Time and Memory

zk-SNARK Size: 2988 bits **Predicate Size:** 420 bits **Verifier Time:** < 0.1 seconds

We used default curves (MNT4/MNT6) that gave us ~80bits of security. Non-published curves with similar performance exist for 128 bits of security. DARPA SIEVE program should yield a backend with 100-1000x performance increase and post-quantum security.

Tests were performed on a variety of fully connected neural network topologies



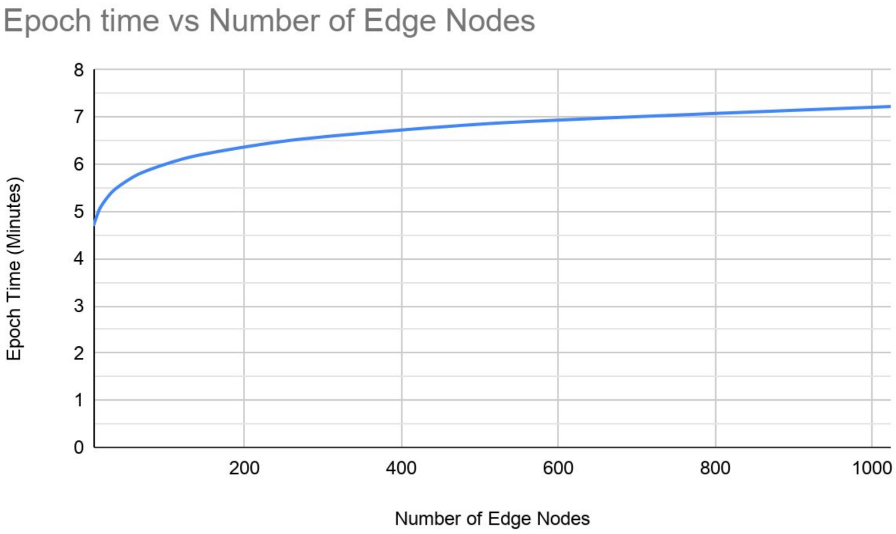


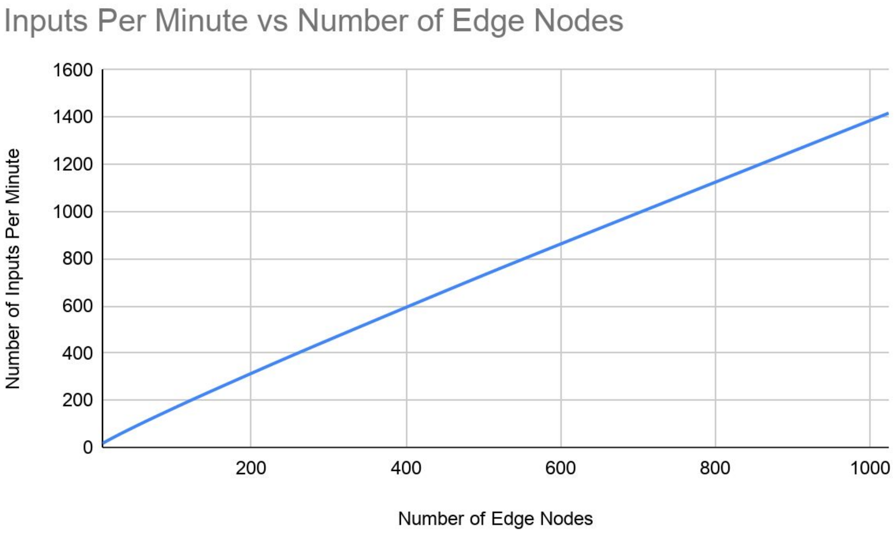
When the number of parameters is low and the input is large, the primary cost is the input hashing

When the number of parameters is high, the cost of compliance predicate checks dominates and scales linearly

Benchmarking: Training Speed Estimates

*We performed tests on a 4 layer neural network with an input size of 1024 and an output size of 10.
**These statistics assume the dataset is divided into batches of 8-10 inputs per edge node





As the number of edge nodes increases, there is a linear increase in the number of inputs per minute that this model can handle with exponentially decreasing additional time overhead per communication round

Other Potential Application Areas

- **Privacy-Preserving AI/ML**
  - Distributed data collection with dynamic network topologies
  - Verifiable and compliant decision making
- **Nuclear Security Science**
  - Remote facility modelling, assessment, and auditing without exposing protected information
- **IoT Data Synthesis**
  - Collaborative remote data science
  - Fully private and tamper proof data collection and analysis

Conclusion And Future Work

- **Conclusion**
  - It is possible to use zero knowledge proofs to do data science research on datasets which were not previously accessible due to privacy concerns or lack of trust
  - Tamper-proof, verifiable, distributed medical machine learning is currently a possibility and will soon be highly practical
  - It is possible do train a machine learning model so anyone can quickly verify that resulting model was trained correctly on a specific dataset
- **Future Plans**
  - We plan to release the code that we have developed publically for researchers to learn from
  - We plan to generalize this setup to a larger range of ML architectures