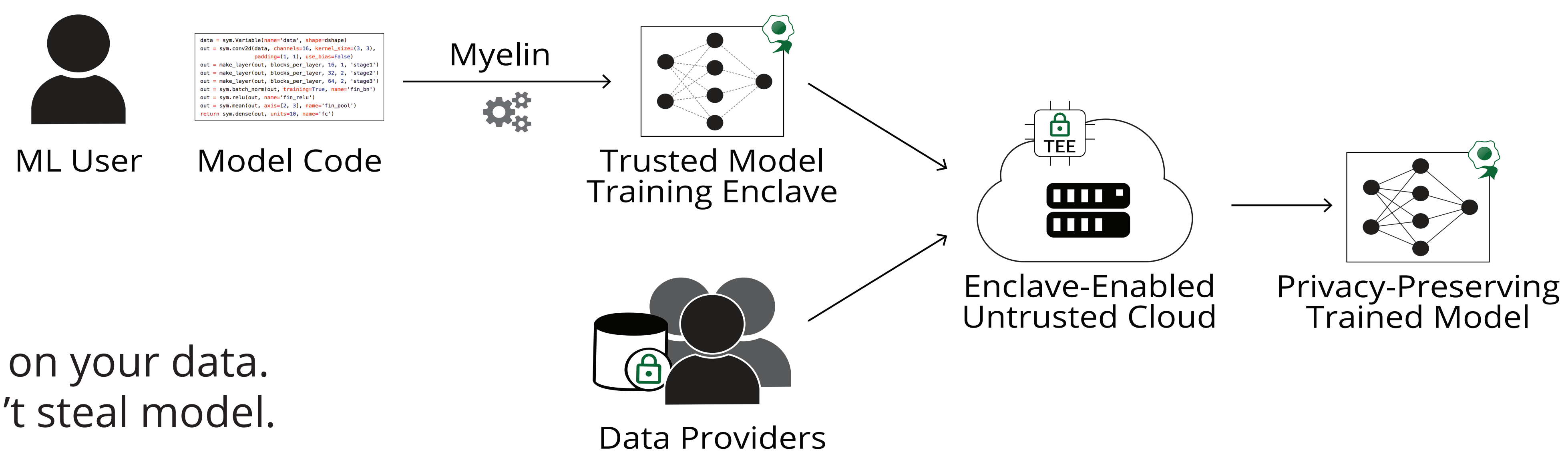
Privacy-Preserving Machine Learning in Trusted Hardware Enclaves
or How We're Actually Going to "Democratize AI"

Nick Hynes <nhynes@berkeley.edu>

The Goal

train a model on multiple users' private data without compromising their privacy



Use Cases

private MLaaS: a cloud provider runs their architecture on your data. You get model outputs, your data stays private, you can't steal model.

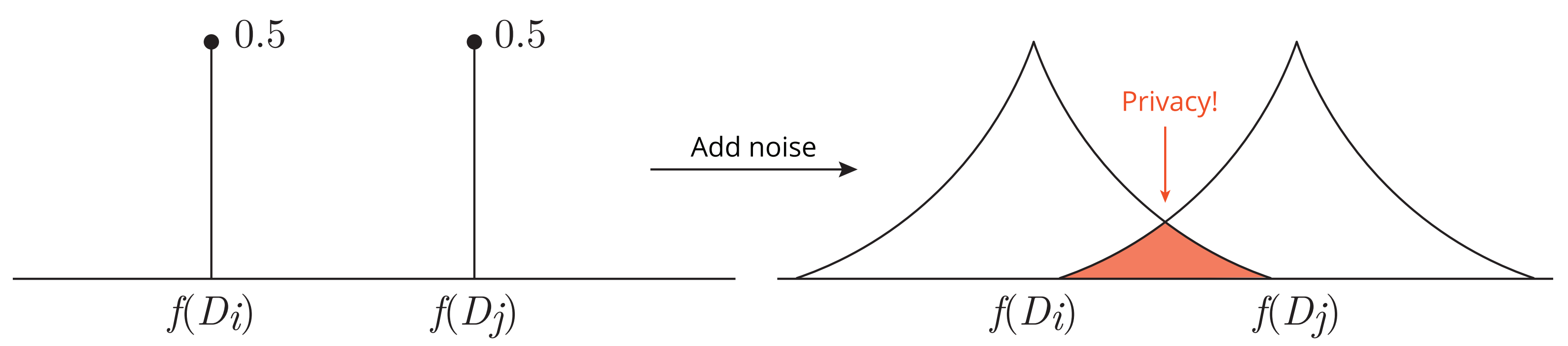
trustworthy ML competitions: you train a model on contest data. Organizer sends private test data to your model and gets verifiable report of accuracy. Your model stays safe until organizer decides to purchase it. Other participants can't cheat by training on test data.

training on shared private data: you (a researcher) want to train a model on several hospitals' data. Directly sharing is too complicated. Instead, have a "trusted third party" train a privacy-preserving model

Privacy Preservation Primitives

Differential Privacy (DP): formal guarantee that models trained on similar datasets are indistinguishable.

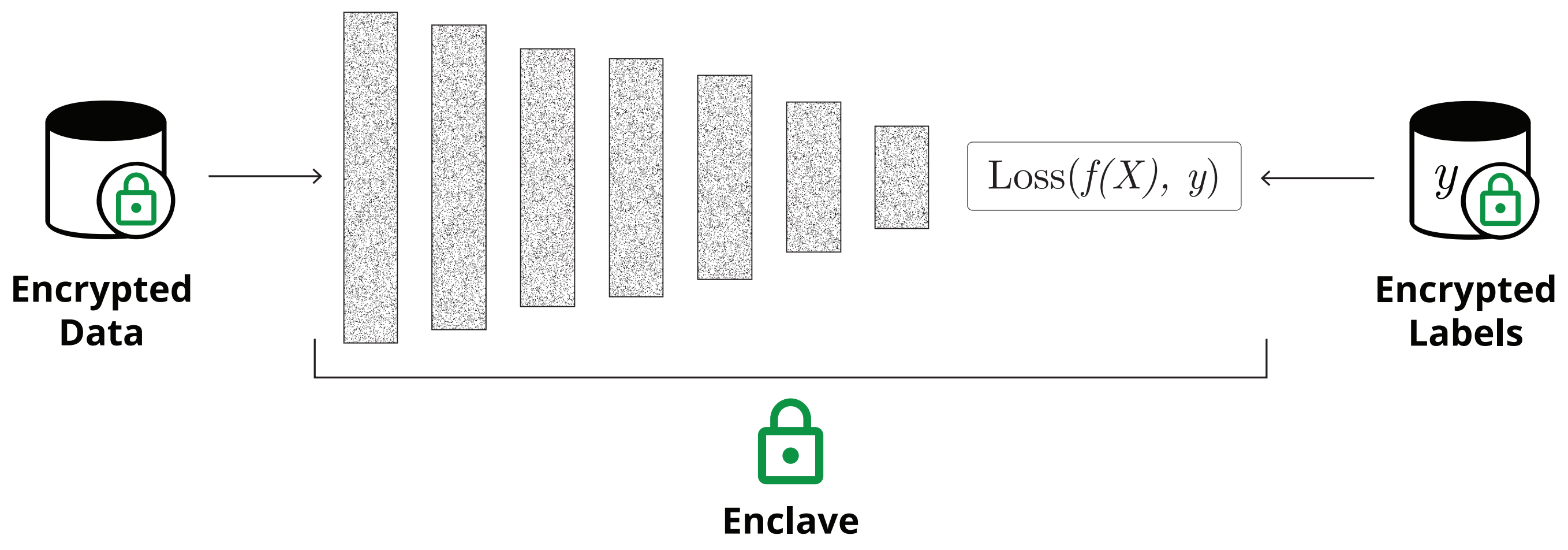
- Informally, *a user's privacy is not compromised by choosing to contribute data to a model*



Trusted Execution Environments (TEEs): hardware enforced isolation of computation+memory (enclave) & remote attestation of loaded code

Remote Attestation: a signed proof that a trusted processor has loaded the user's code. Used to establish secure link into enclave.

- Examples of TEEs: Intel SGX, Komodo, Keystone/Eyrie *(your laptop probably already has SGX!)*



Privacy-Preserving Machine Learning

High level idea: train differentially private models in TEEs

- TEE provides confidentiality and ensures that DP is correctly applied => privacy during and after training
- TEE-based training is ~1000x as fast as training using Homomorphic Encryption or Garbled Circuits and is actually comparable to native performance

Main challenge: code in TEE can't trust the rest of the system. This means: limited memory, no filesystem, no GPU, and no JIT compilation. *How do we do it?*

Key observation: TEE programming model is similar to hardware accelerators like FPGAs and TPUs. We already have tools for this! Specifically, compilers like TVM, JAX, and Glow.

Approach: Use tensor compiler to generate dependency-free ML code which runs in TEE and requests resources from OS

Where we are now

- currently using TVM
- performance is good but challenging to deploy
- limited support for autograd, training

Model	Framework	Speed Train (min/epoch)	 Test (img/s)	Test Acc.
(Liu et al., 2017)	Gazelle (HE + GC)	–	0.08	93.1
	Myelin	2.71	1035	93.1
VGG9	Chiron (4 enclaves)	6.74	–	88.1
	Myelin (1 enclave)	7.82	501	89.5
ResNet-32	Myelin	13.4	453	92.4
MobileNet	Slalom (1 enclave + GPU)	–	35.7	71.0
	Myelin (1 enclave)	–	32.4	71.0

PyTorch Blazing the Way

Imagine going from prototype to production, all within the same framework and without compromising privacy

Who cares about privacy? Several major cloud providers

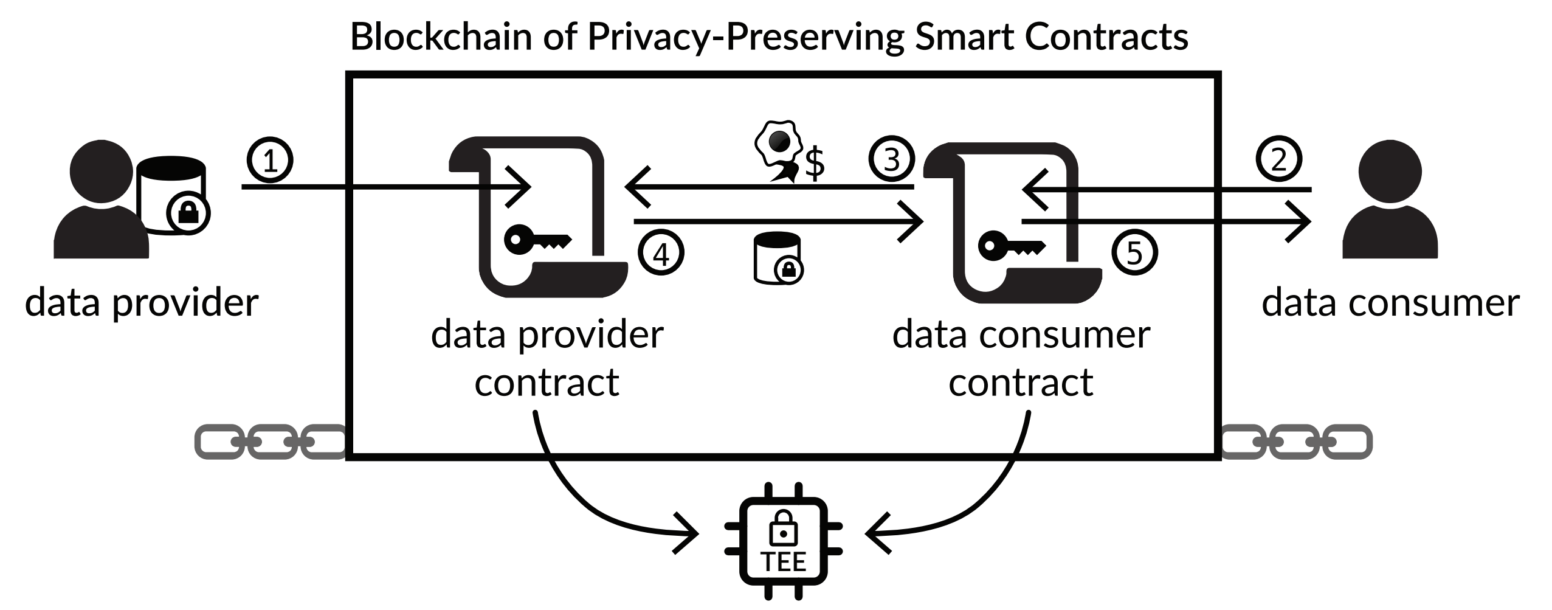
PyTorch is well poised to take the lead:

- Glow: already has support for autograd, has easy-to-use programming model
- active developer community
- emphasis on research to production

What we need:

- multi-threading for Glow CPUBackend: for non-private, splitting model across cores maximizes throughput, but switching between TEEs is incredibly expensive
- tiny tweaks to the runtime to support loading enclave libs
- otherwise, just code reviews! :)

Killer App: ML on the Blockchain



- Sterling, a privacy-preserving data marketplace, enables uncoordinate sharing and use of private data

1. *data providers* upload encrypted data to privacy-preserving smart contracts which only yield data to *data consumer* smart contracts which satisfy certain constraints (e.g., price, privacy)
2. data consumers create smart contracts which train models on data of providers whose constraints are satisfied
3. consumer smart contract acts as a virtual trusted third party, trains model on data, and returns privacy-preserving outputs

- **Use cases:** secure credit scoring, medical research, user-owned advertising profiles, unlocking value of IoT data

OASIS LABS
Berkeley
UNIVERSITY OF CALIFORNIA