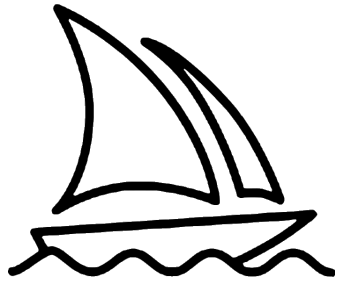
Scaling Trustless DNN Inference with zkml and its Applications

Daniel Kang

2023-Aug-02, ZKProof 5.5

ML is eating the world

Bard AI



Can we execute ML models *trustlessly*?

How can we verify computation?

MPC

HE

ZKP

Zero Knowledge Proofs (ZKP)

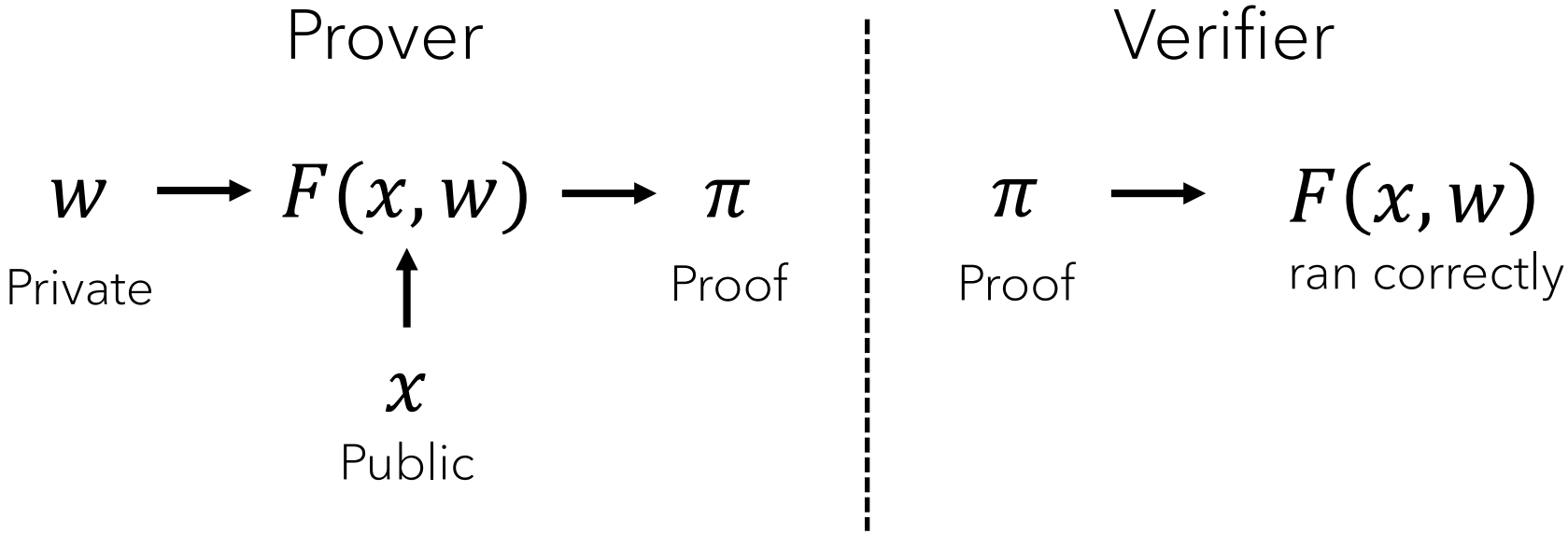
- Prove computation happened correctly
- Anyone can verify!

Performance considerations

- Can give privacy and validity
- No interaction required
- High compute overhead

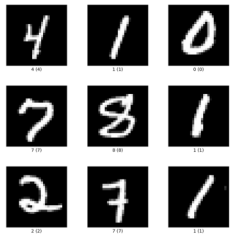
This talk: efficient ZKPs for ML, image edits

ZKPs: proofs of computation

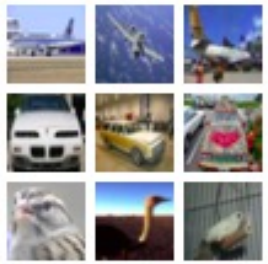


Proofs can be computed on-demand, after-the-fact

What can zkml do **today**?



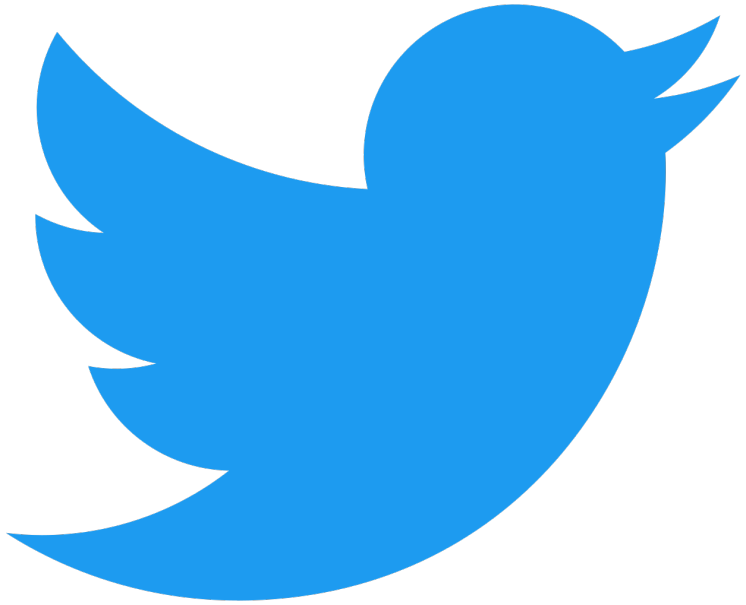
MNIST
(10 classes, 28x28)

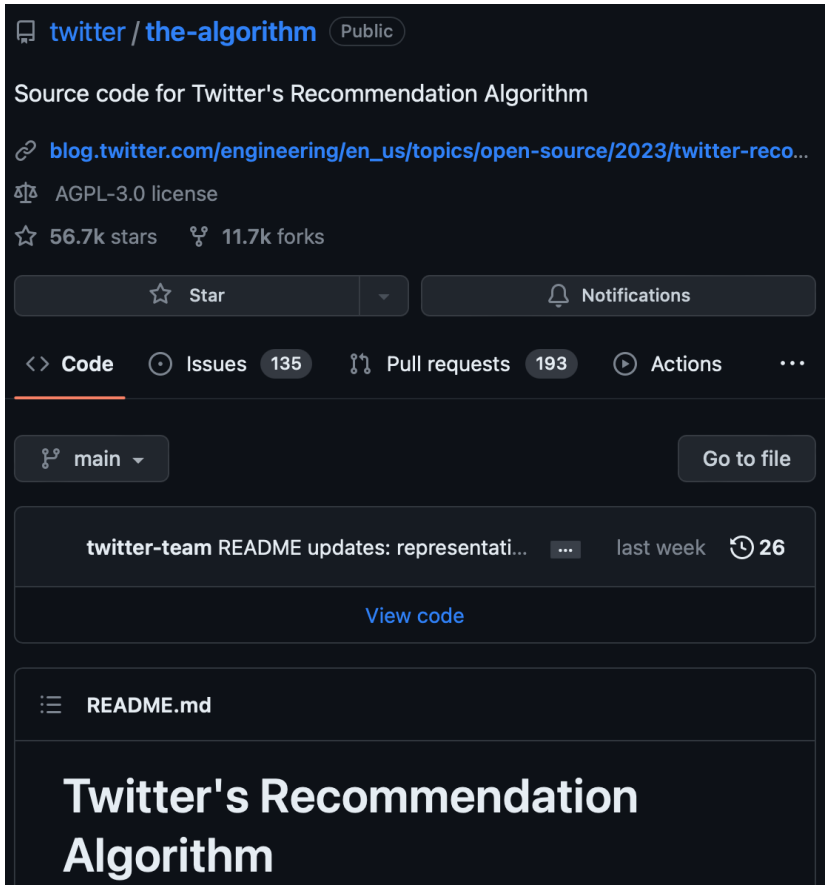


CIFAR 10
(10 classes, 32x32)

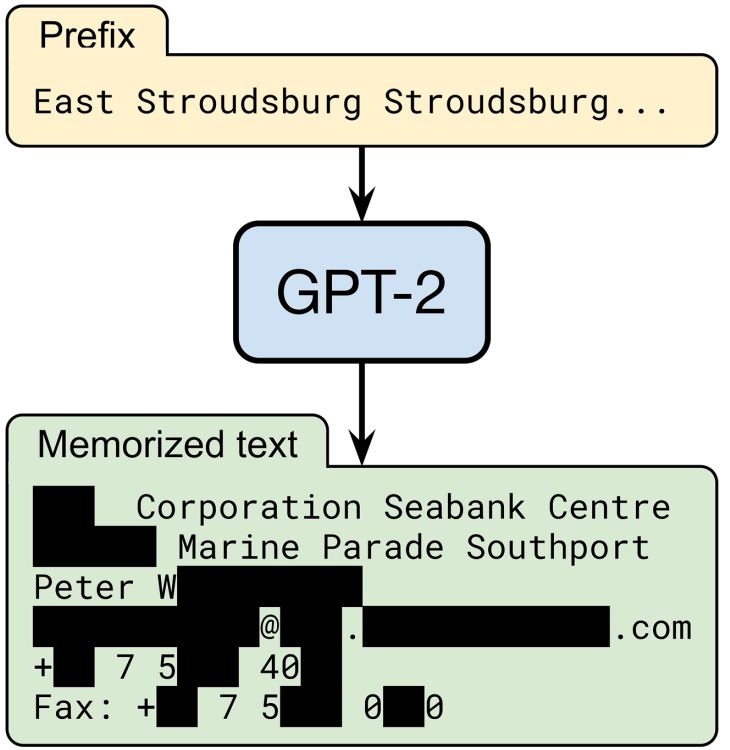
ImageNet
(1000 classes, 224x224)

We SNARKed **ImageNet**



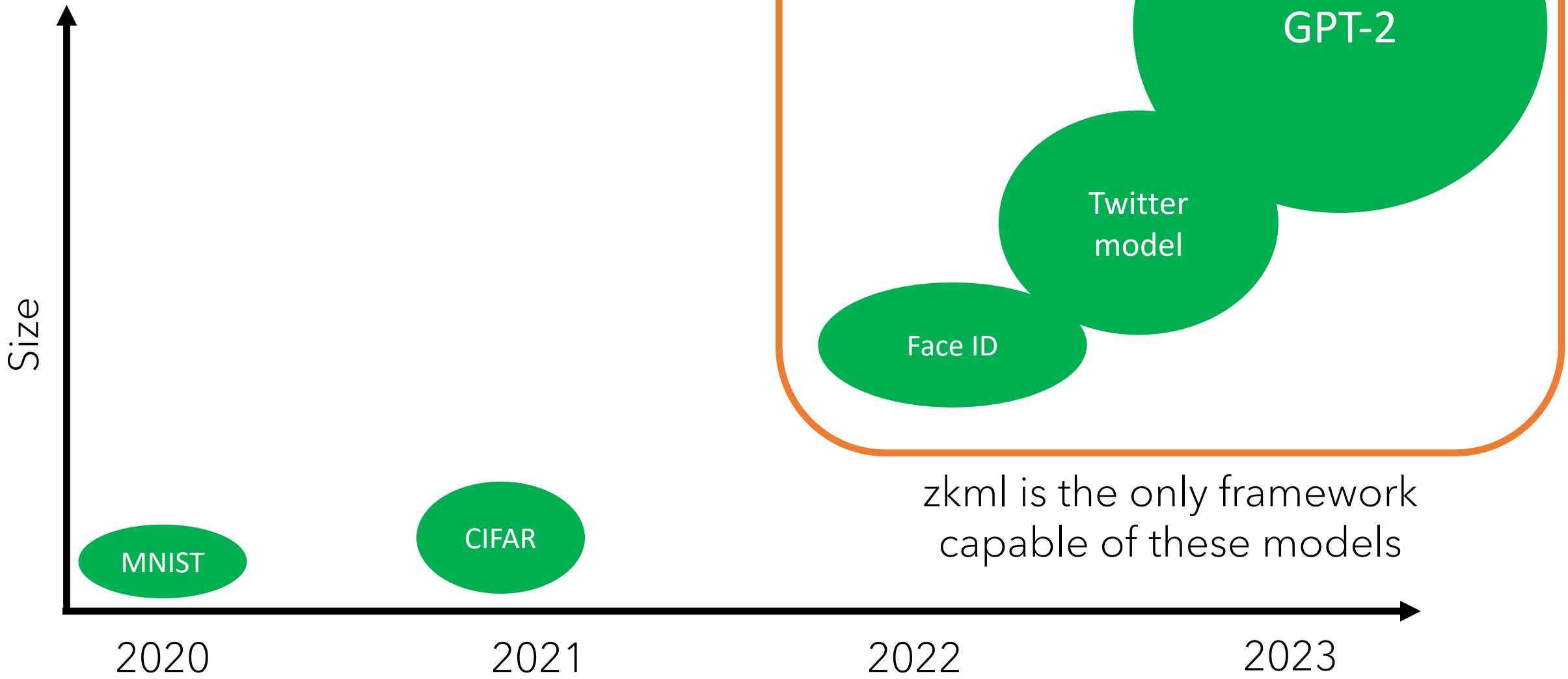


We SNARKed **The Algorithm**



We SNARKed **GPT-2**

ZKPs for ML

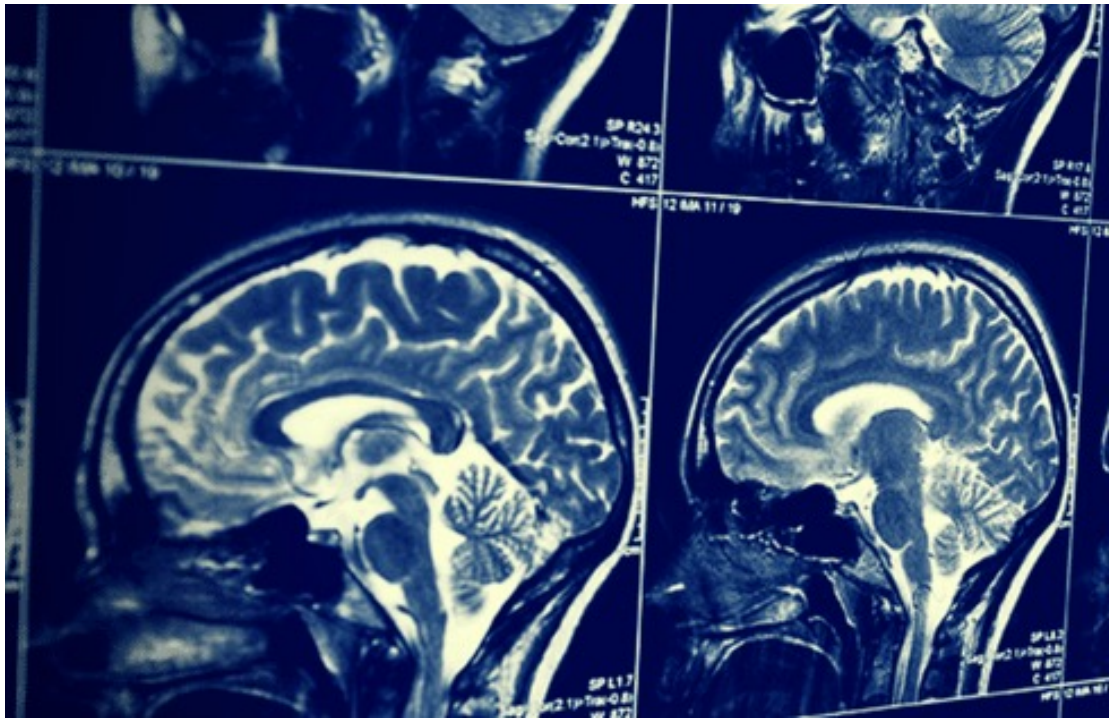


zkml is the only framework capable of these models

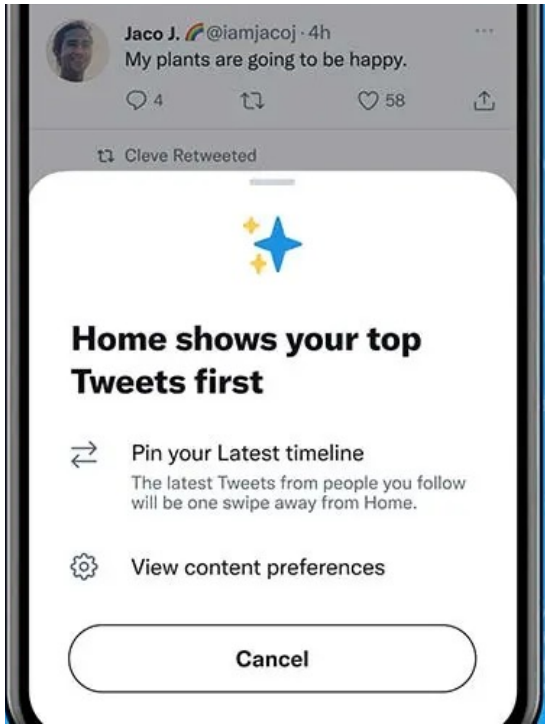
Outline

- Applications
  - Trustless audits
  - Decentralized prompt marketplaces
  - Trustless biometric identification / fighting deepfakes
- Open-sourcing zkml

Trustless audits

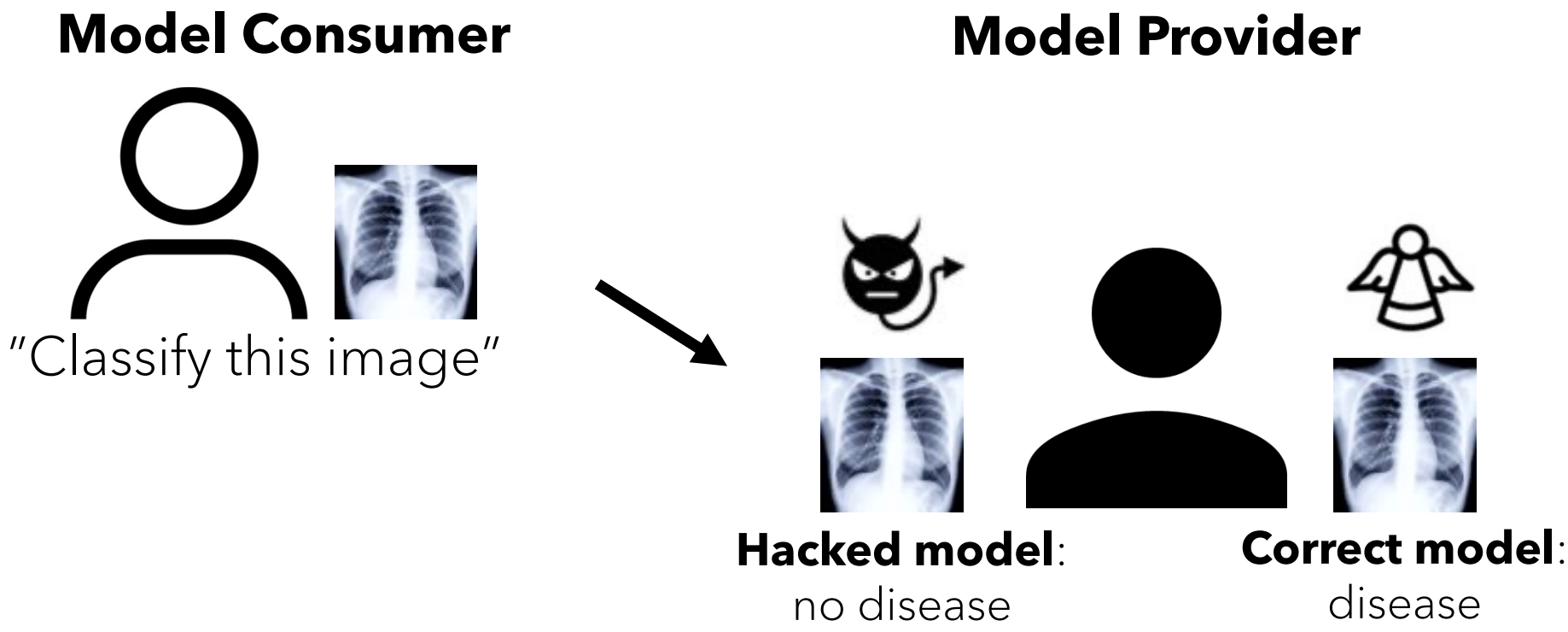


Prove that you ran the FDA regulated model

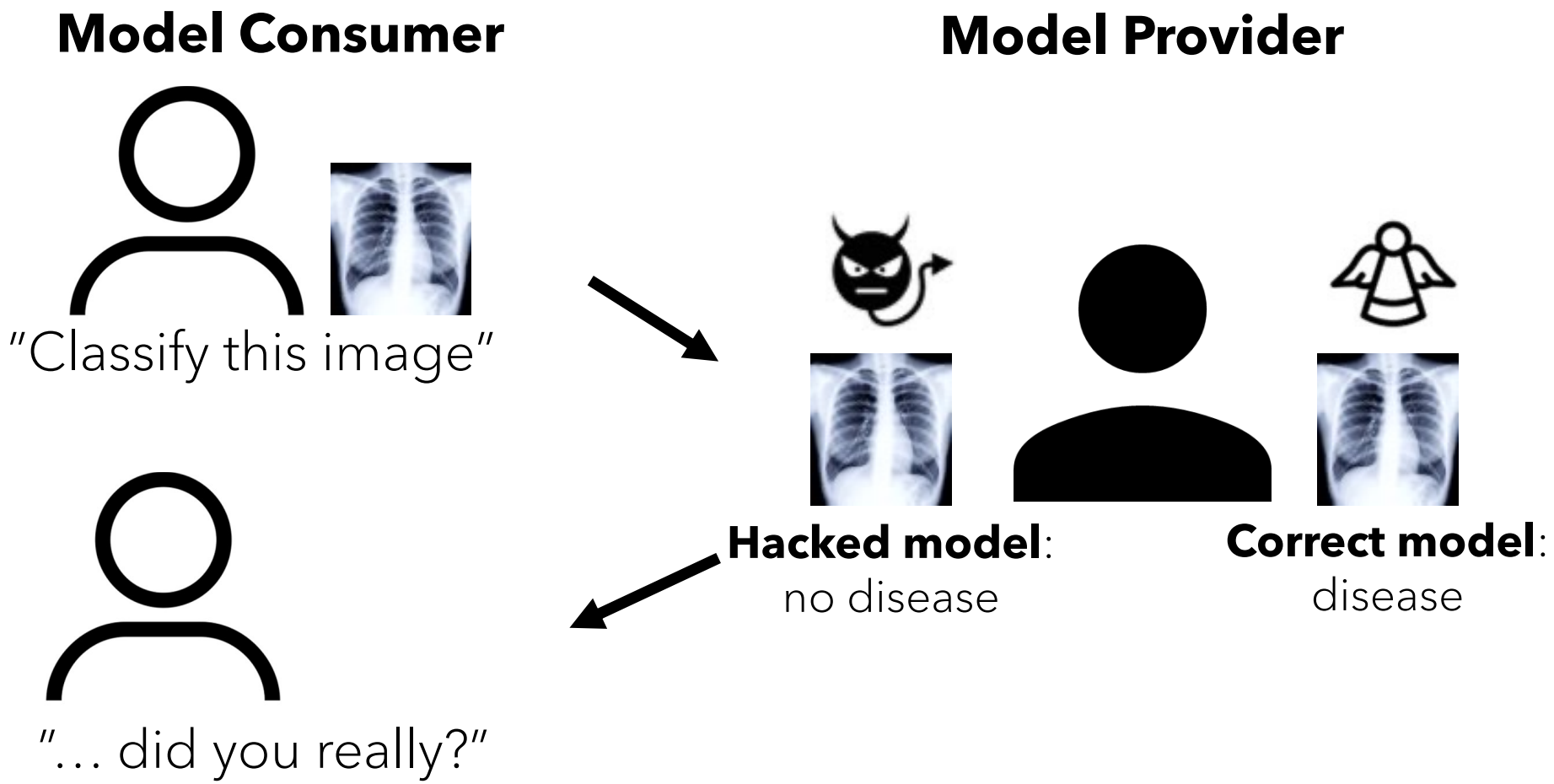


Prove that my timeline isn't biased

ML service providers are proliferating



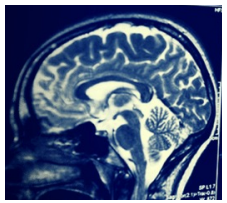
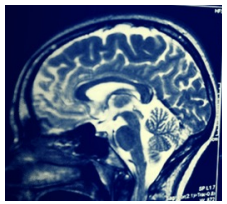
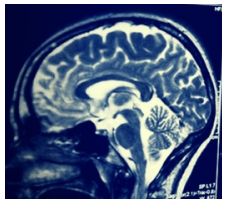
Core challenge: how can we *verify* that a machine learning model was run?



What is $F(x, w)$ for inference?

- Functional view of inference
  - Forward(weights, data) → output
- Inference:
  - x (public): commitments to weights, data, output
  - w (hidden): weights, data

Proof of training

H_1

H_2

H_3

Commit to training data

H_1

H_2

H_3

SGD → W

Produce ZK-SNARKs of training

What is $F(x, w)$ for training?

- Functional view of inference, SGD
  - Forward(weights, data) → output
  - SGD(weights, data) → new weights
- SGD:
  - x (public): commitments to weights, data
  - w (hidden): weights, data

Test-time trustless audit

» Data isn't biased

» Model performs well

» …

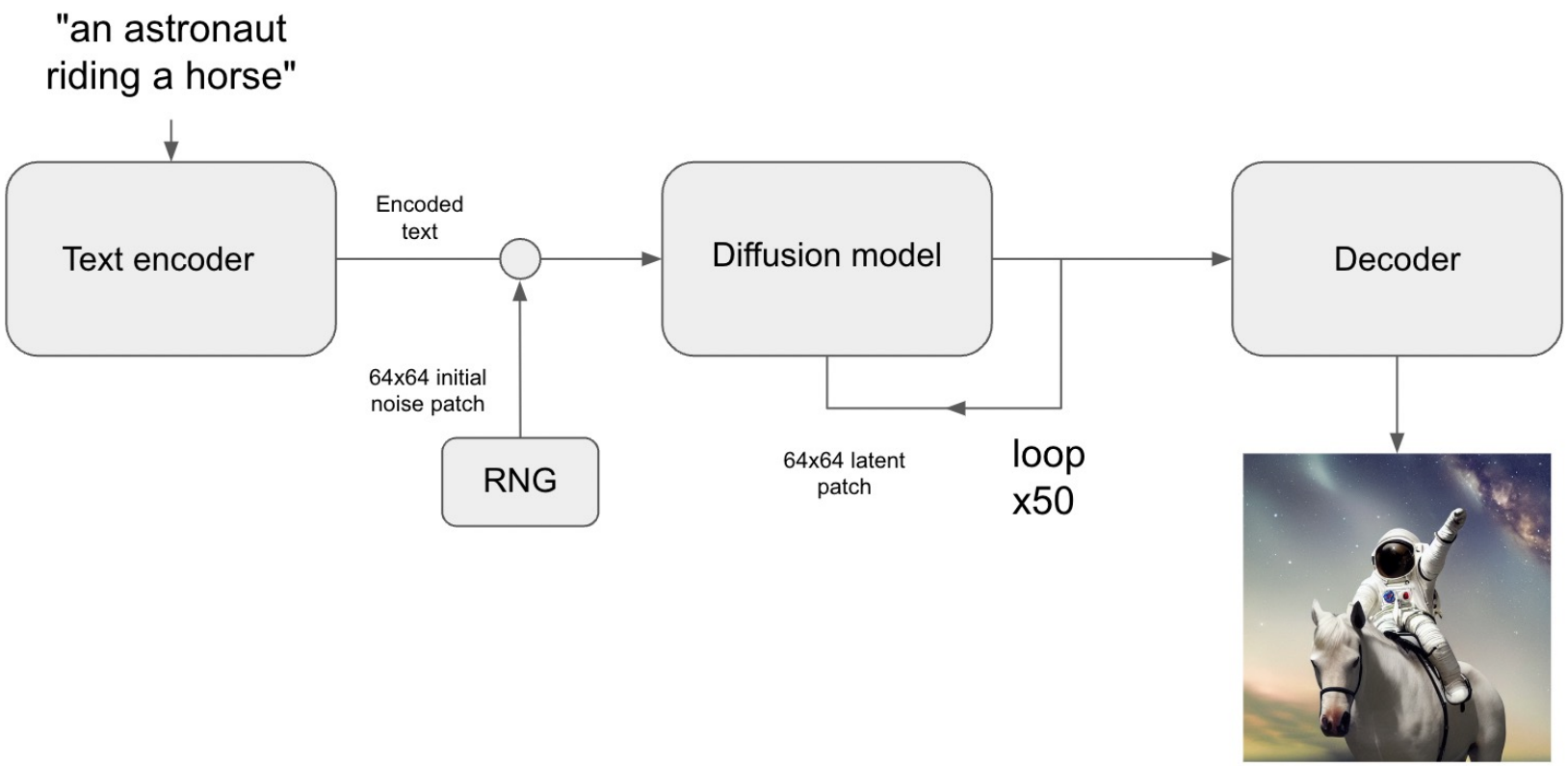
$$F(H_1, H_2, \ldots, H_n; W)$$

Compute audit functions over data, weights

Outline

- Applications
  - Trustless audits
  - Decentralized prompt marketplaces
  - Trustless biometric identification / fighting deepfakes
- Open-sourcing zkml

Generative AI requires prompts



Prompts are valuable and difficult!

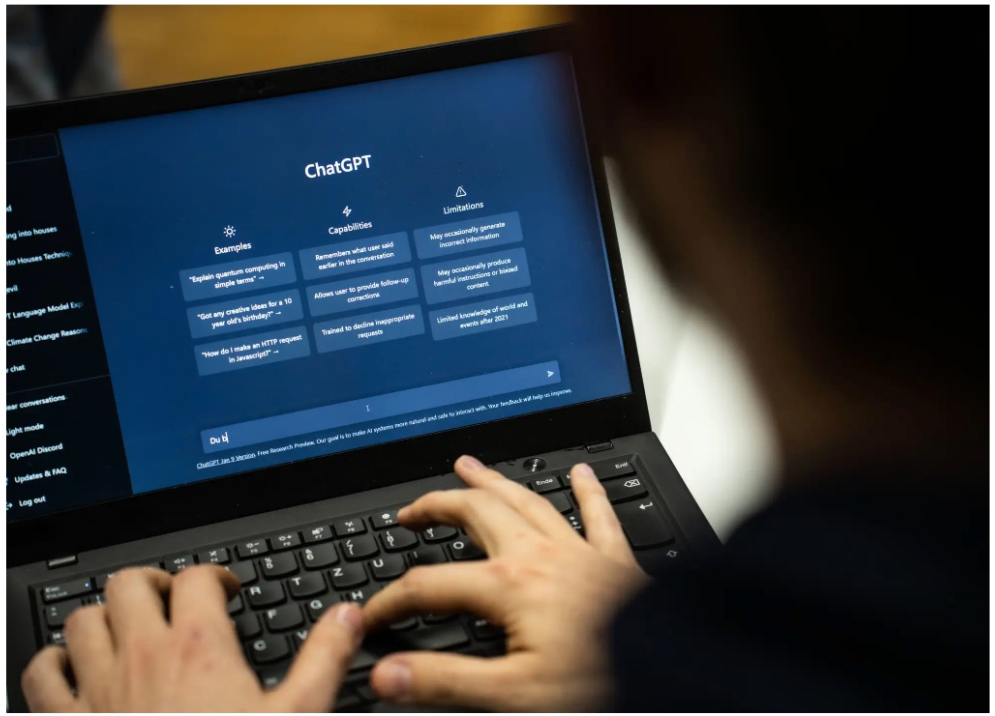
Cinematic, off-center, two-shot, 35mm film still of a 30-year-old french man, curly brown hair and a stained beige polo sweater, reading a book to his adorable 5-year-old daughter, wearing fuzzy pink pajamas, sitting in a cozy corner nook, sunny natural lighting, sun shining through the glass of the window, warm morning glow, sharp focus, heavenly illumination, unconditional love --ar 16:9
Credit:
https://twitter.com/nickfloats/status/1640827136809345024

Drip pope???

AI 'prompt engineer' jobs can pay up to $335,000 a year and don't always require a background in tech

Britney Nguyen Mar 31, 2023, 2:33 PM

Read in app



The rise of generative AI tools like ChatGPT is creating a hot market for "prompt engineers" who test and improve chatbot answers. Getty Images

Proof of prompt

Cinematic, off-center, two-shot, 35mm film still of a 30-year-old french man, curly brown hair and a stained beige polo sweater, reading a book to his adorable 5-year-old daughter, wearing fuzzy pink pajamas, sitting in a cozy corner nook, sunny natural lighting, sun shining through the glass of the window, warm morning glow, sharp focus, heavenly illumination, unconditional love --ar 16:9

ZK-SNARK

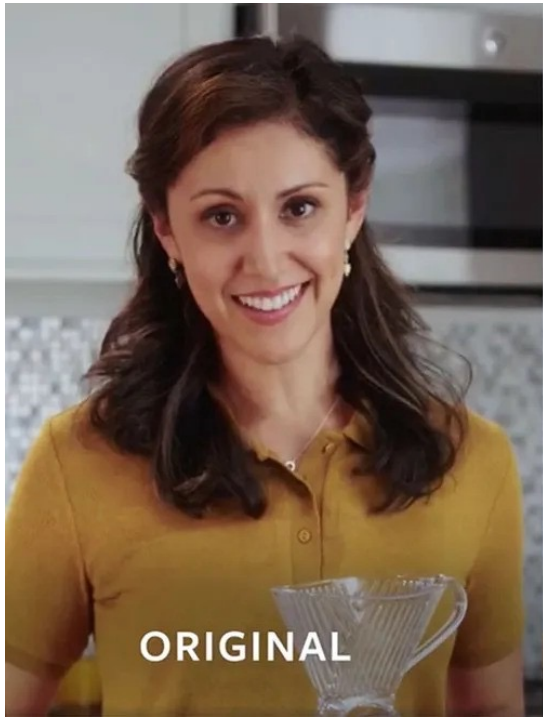
What is $F(x, w)$ for prompt marketplaces?

- Functional view of inference
  - Forward(weights, data) → output
- Prompt marketplace:
  - x (public): weights, commitment to data
  - w (hidden): data

Outline

- Applications
  - Trustless audits
  - Decentralized prompt marketplaces
  - Trustless biometric identification / fighting deepfakes
- Open-sourcing zkml

Deepfakes



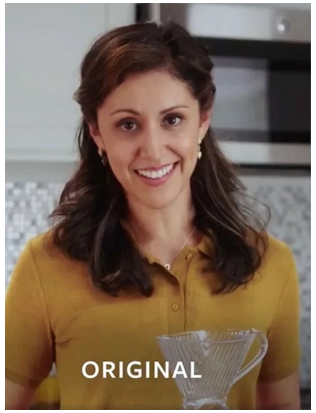
AI method

Deepfakes are on the rise

- Used by state actors to spread misinformation
- Trick businesses into miswiring funds
- …

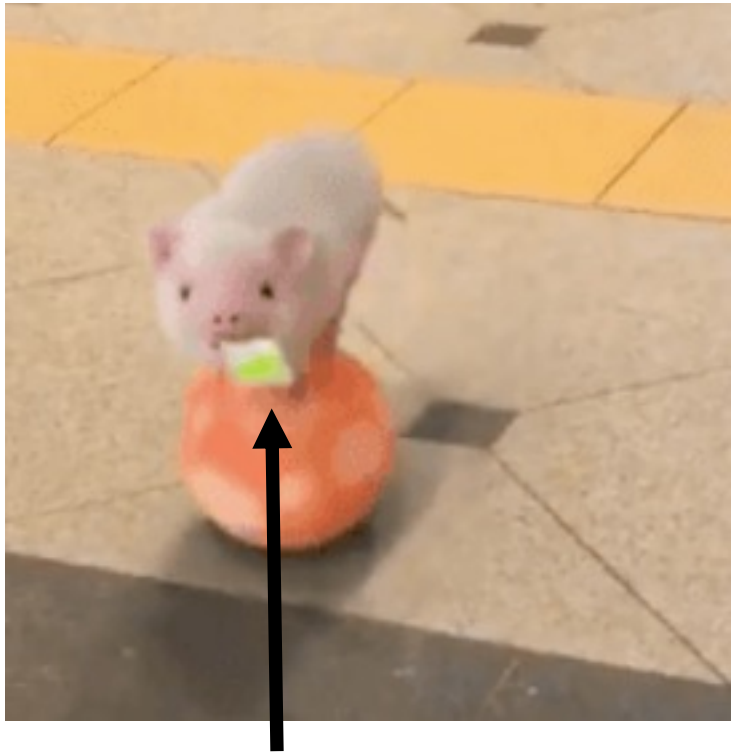
Attested cameras can help

I took this photo

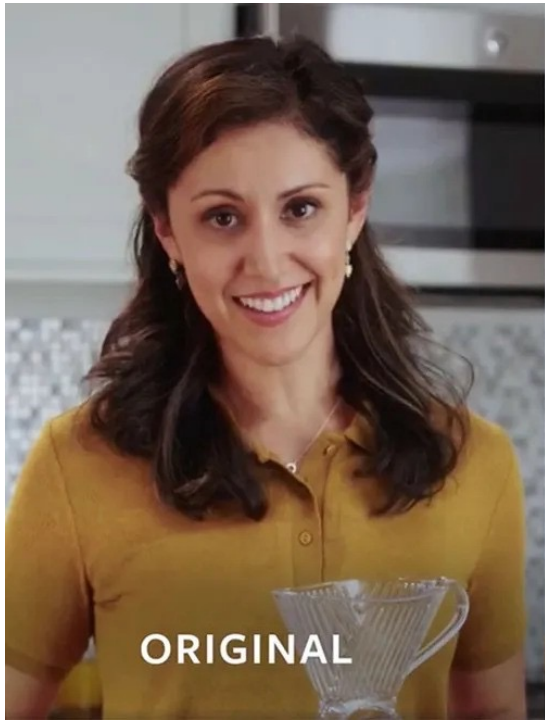


But not this one!

Photo takers want to edit images privately



Private credit
card information!



Crop picture
privately for face ID

zk-img: attesting to image edits securely and privately

Hidden

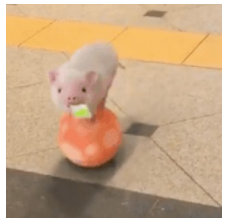
zk-img: the edit
was done properly

Revealed

What is $F(x, w)$ for zk-img?

- Functional view of image edits
  - Edit(image) → output
- zk-img:
  - x (public): commitment to image
  - w (hidden): output

zk-img in context



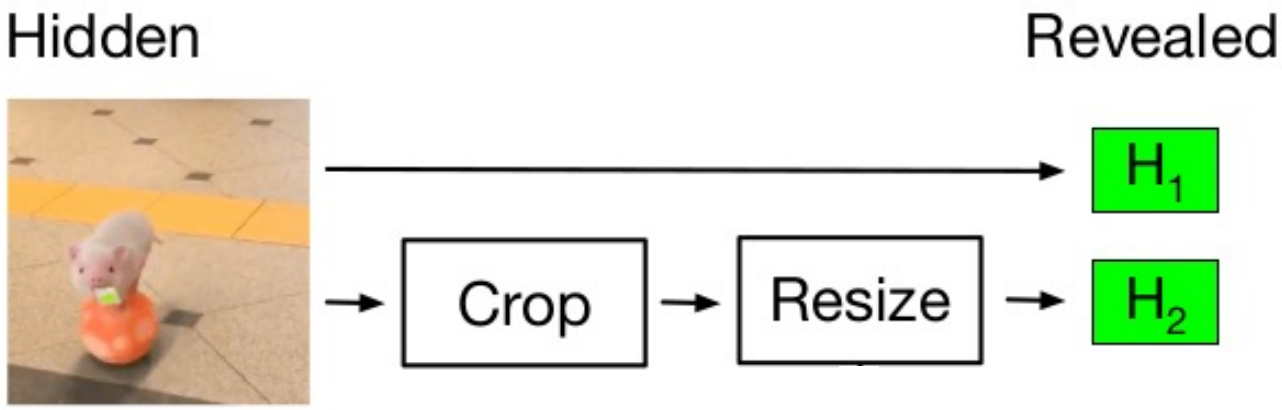
Original image

π
Proof

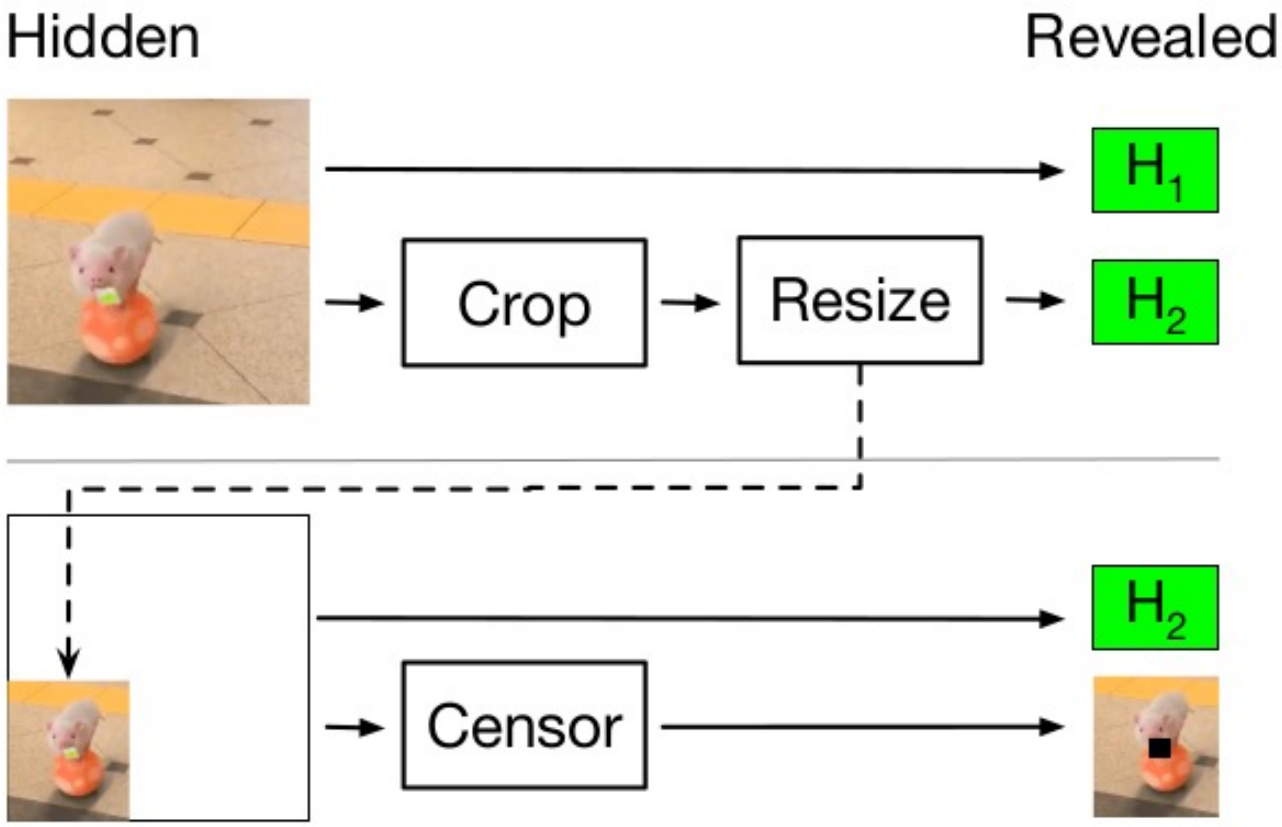
Registry

+ π → Valid

Attesting to multiple edits



Attesting to multiple edits



Trustless face ID

Private

Public

Evaluation setup

» HD images

» Instance type: `r6i.16xlarge` (64 threads, 512GB RAM)

» Metrics: key generation, proving/verification times

Benchmarks

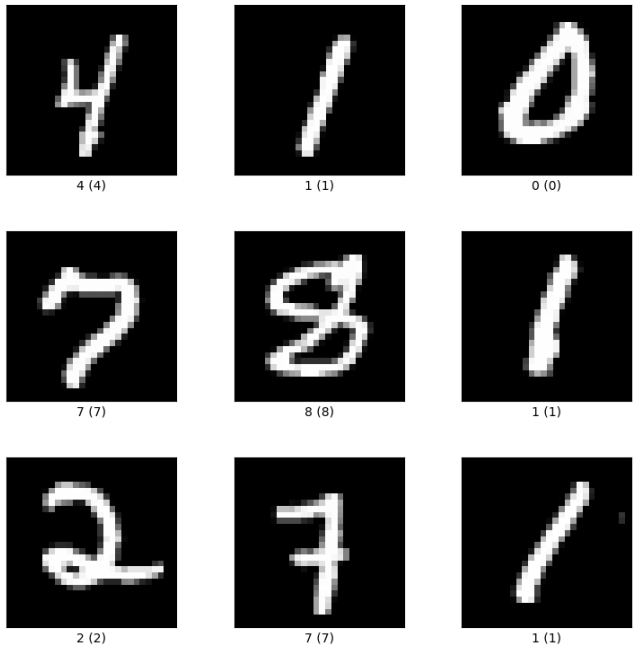
Transformation	Key generation	Proving	Verification	Proof size	Peak memory usage
Crop (HD → SD)	243.7s	330.7s	5.41ms	3040 bytes	70.7 GB
Resize (HD → SD)	247.7s	333.3s	5.95ms	3040 bytes	70.7 GB
Contrast	281.5s	363.9s	4.88ms	3712 bytes	85.3 GB
White balance	294.5s	362.7s	4.91ms	3776 bytes	87.9 GB
RGB2YCbCr	283.0s	607.5s	7.30ms	6592 bytes	91.6 GB
YCbCr2RGB	282.3s	608.8s	6.70ms	6592 bytes	91.6 GB
Convolution	339.7s	427.9s	5.89ms	4672 bytes	102.2 GB

Verification times as low as 4.9ms!

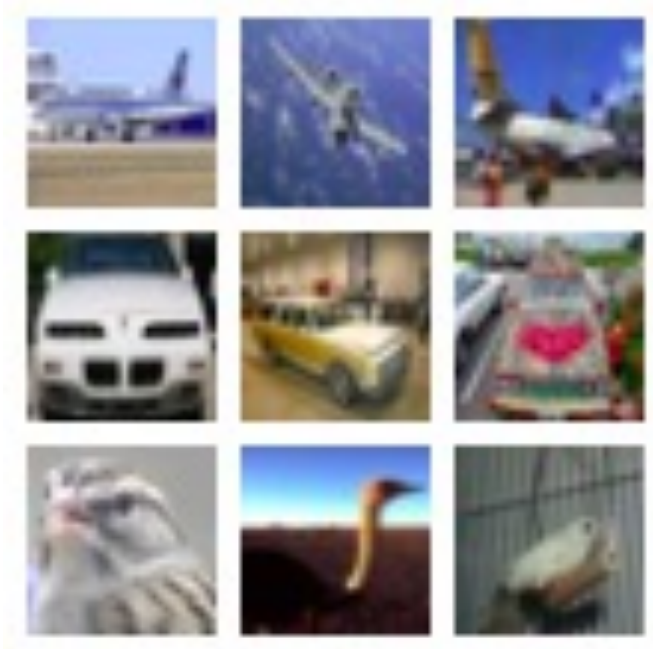
Outline

- Applications
  - Trustless audits
  - Decentralized prompt marketplaces
  - Trustless biometric identification / fighting deepfakes
- Open-sourcing zkml

ZKPs for DNN inference



MNIST
(10 classes, 28x28)



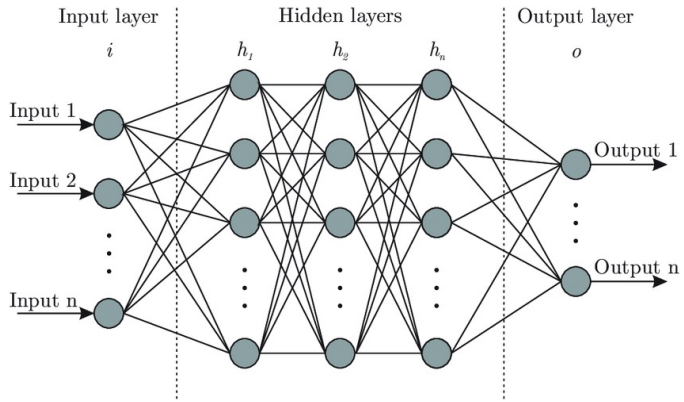
CIFAR 10
(10 classes, 32x32)

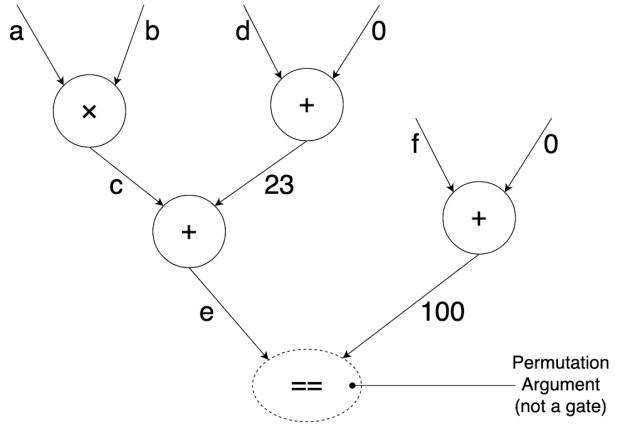
ImageNet
(1000 classes, 224x224)

All prior work

This work

ZKML: verified DNN inference





Architecture

Arithmetization

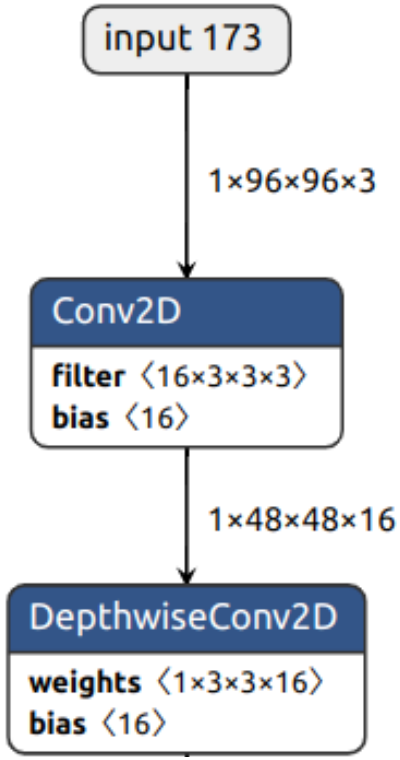
Proving system

Optimized DNNs for ZK

» Quantized models

» Optimize layout in our framework

» Can achieve high accuracy!



Optimized arithmetization

» Linear layers via **custom gates** (polynomial constraints)

» Non-linearities via **lookups**

» **Fused** fixed-point arithmetic

Using zkml

```
# Installs rust, skip if you already have rust installed
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

git clone https://github.com/ddkang/zkml.git
cd zkml
rustup override set nightly
cargo build --release
mkdir params_kzg
mkdir params_ipa

# This should take ~8s to run the first time
# and ~4s to run the second time
./target/release/time_circuit examples/mnist/model.msgpack examples/mnist/inp.msgpack kzg
```

Evaluation

» Models: MobileNet v2

» Metrics: accuracy, key generation, proving/verification times

Performance benchmarks

Model	Accuracy (top-5)	Setup time	Proving time	Verification time
MobileNet, 0.35, 96	59.1%	93.9s	163.2s	0.74s
MobileNet, 0.5, 224	75.7%	937.7s	1530.7s	6.32s
MobileNet, 0.75, 192	79.2%	1341.2s	2457.5s	10.27s

Open-source release: 10% higher accuracy, 6x cheaper proving, 500x cheaper verification

Cost of verifying private model accuracy

ϵ	Sample size	Total cost	
5%	600	~~$99.93~~	$16.65
2.5%	2,396	~~$399.08~~	$66.51
1%	14,979	~~$2494.90~~	$415.81

Compared to $85,000 for a moderate size dataset

Linear layers

» Common operation: dot products

» Split dot products across rows, accumulate

» Two gates:

$$c_i = \sum_{j=1}^{N} (x_i^j - z) \cdot w_i^j \qquad c_i = \sum_{j=1}^{N} x_i^j .$$

Non-linearities

» Examples: ReLU, reciprocal square root, square root, …

» Lookup arguments

Shape operations

» Examples: padding, split, concatenation

» Copy constraints

Tricky operations

» Max

» Softmax

» Variable division

Other considerations

» How to represent floating point arithmetic?

» How to hide weights?

» Optimal circuit layout?

What’s next?

Advances in hardware acceleration

ZK GPT2 Circuit

64-core CPU

- Degree = 25
- End-to-end time = 9591.2s

Our FPGA server + 64-core CPU

- Degree = 25
- End-to-end time = 190.32s
- End-to-end speedup = 50.396
- MSM + NTT percentage ~ 18.2%

50x faster proving with FPGAs!

Advances in proving systems

» Faster lookups: log derivative arguments, cq, …

» Faster matrix multiplication: cqlin, …

Advances in arithmetization

» Improved circuit layout

» Improved softmax, etc.

Conclusion

» We are increasingly interacting with digital systems

» ZKPs provide trust in the face of adversaries

  » For ML models

  » Against deepfakes

» Much to come!

ddkang@g.illinois.edu @daniel_d_kang

https://github.com/ddkang/zkml