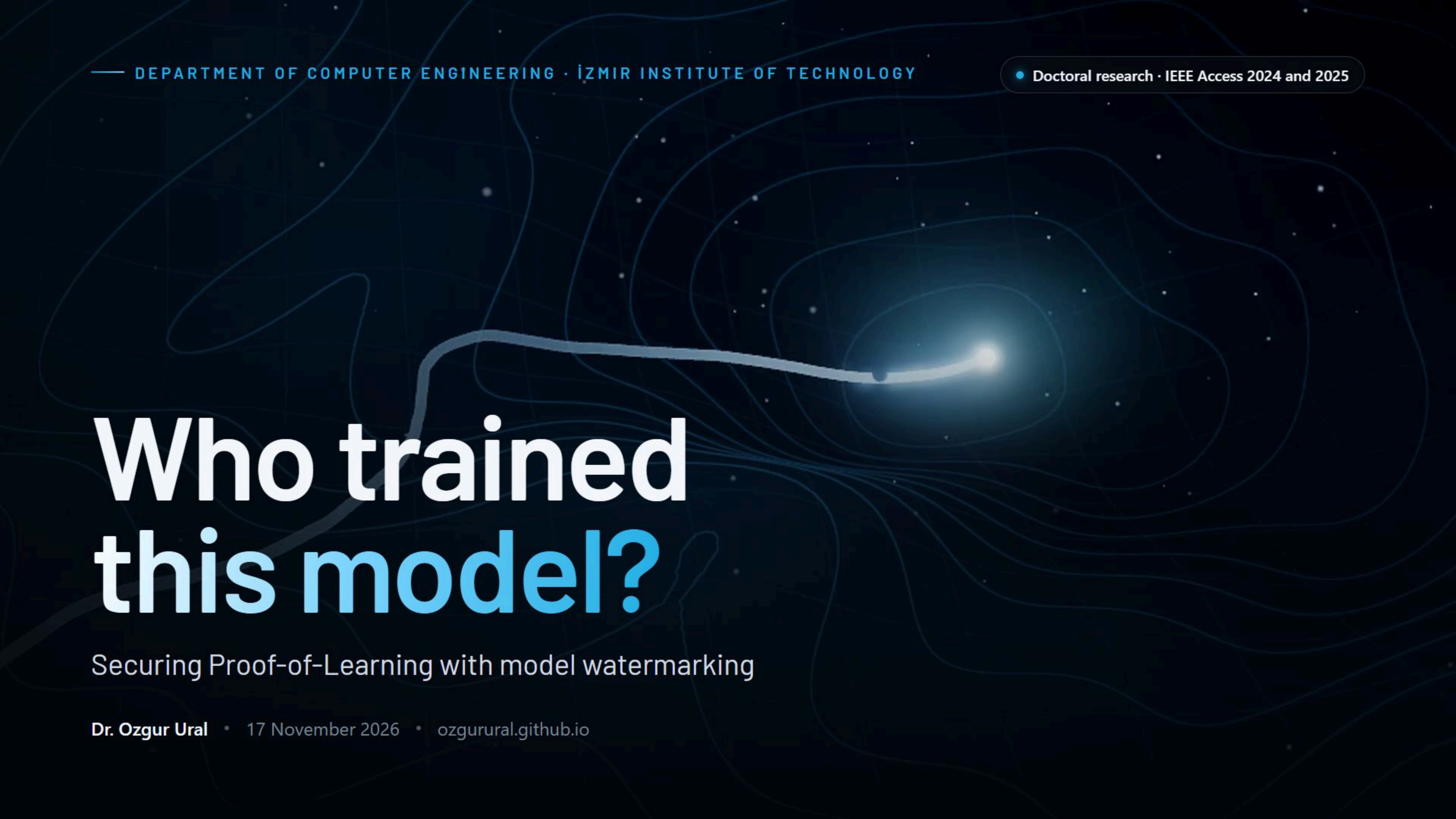
DEPARTMENT OF COMPUTER ENGINEERING · İZMIR INSTITUTE OF TECHNOLOGY

Doctoral research · IEEE Access 2024 and 2025



Who trained this model?

Securing Proof-of-Learning with model watermarking

Dr. Ozgur Ural • 17 November 2026 • ozgurural.github.io

PART I · THE STUDY

Proving who trained a model when copies are perfect

2024	Feature-based model watermarking inside Proof-of-Learning	IEEE Access
2025	SecurePoL: three watermark families, a threat model, statistical analysis	IEEE Access
2025	The doctoral dissertation that joins them	Embry-Riddle

Ozgur Ural and Kenji Yoshigoe (advisor) · code and configurations on GitHub

A copy of a model carries no trace of who trained it.

An AI model learns from examples, and what it learns is stored as a file of numbers. Training is costly. Copying the file is free, and the copy is identical.

QUESTION 1 · EFFORT

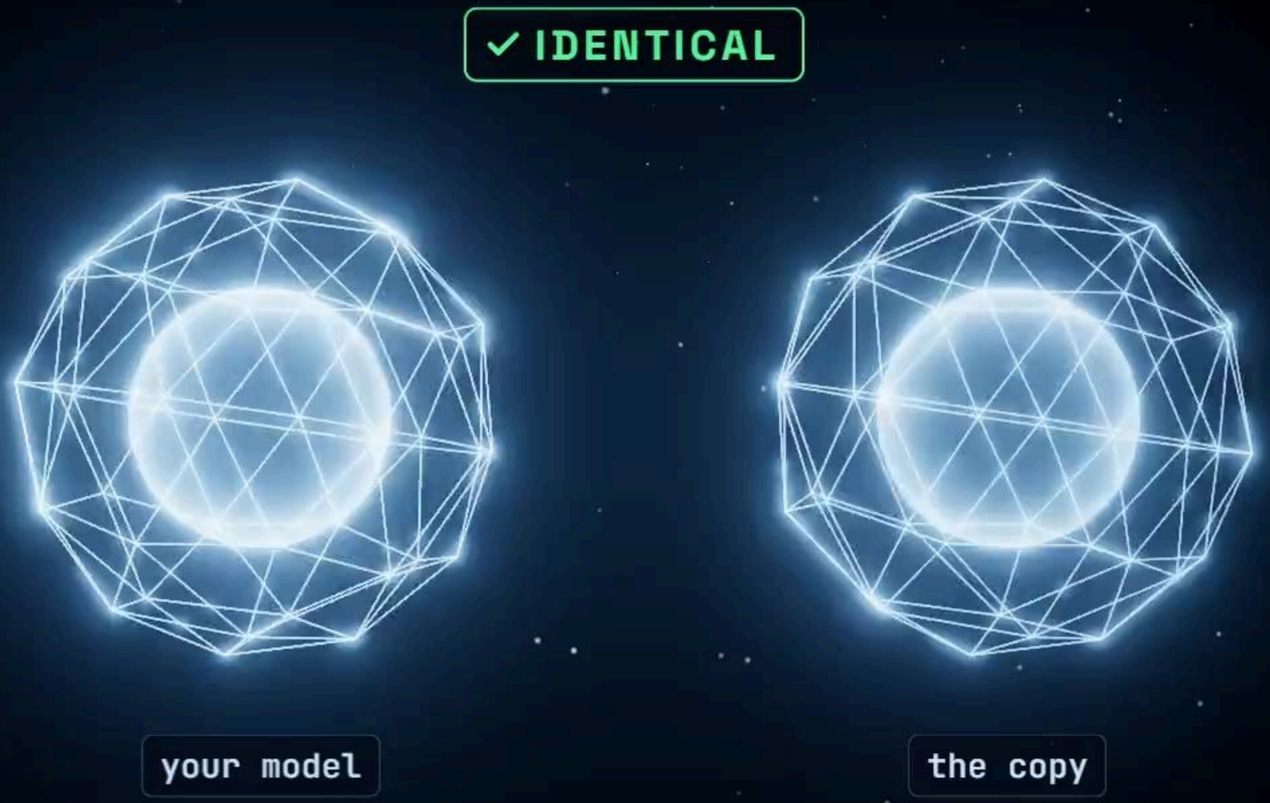
Did they do the training?

A record of the training run could show it.

QUESTION 2 · IDENTITY

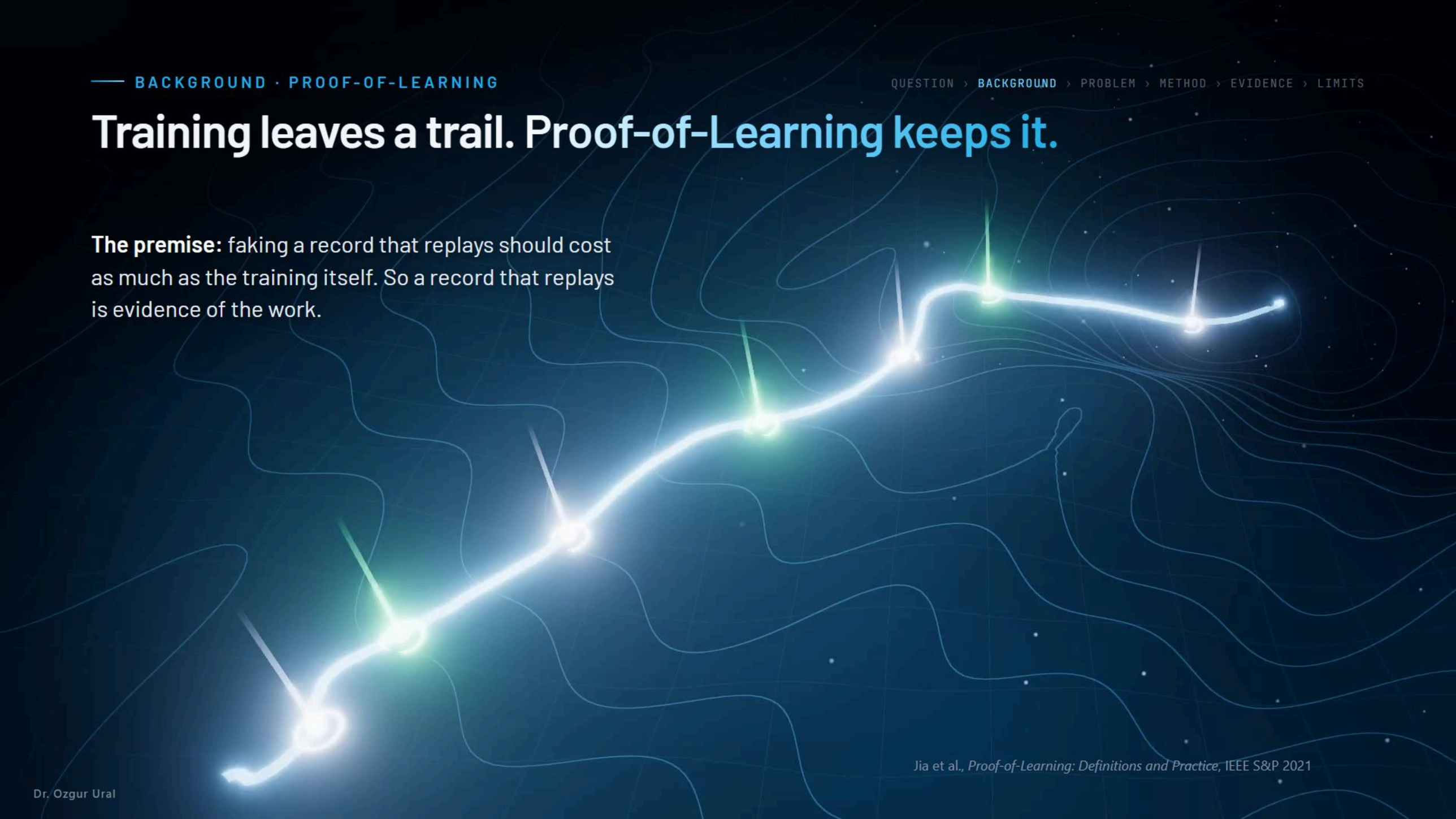
Is it their model?

A hidden mark inside the model could show it.



Training leaves a trail. Proof-of-Learning keeps it.

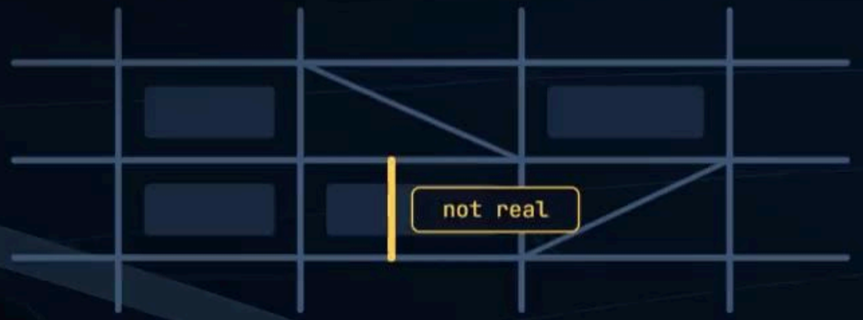
The premise: faking a record that replays should cost as much as the training itself. So a record that replays is evidence of the work.



Jia et al., *Proof-of-Learning: Definitions and Practice*, IEEE S&P 2021

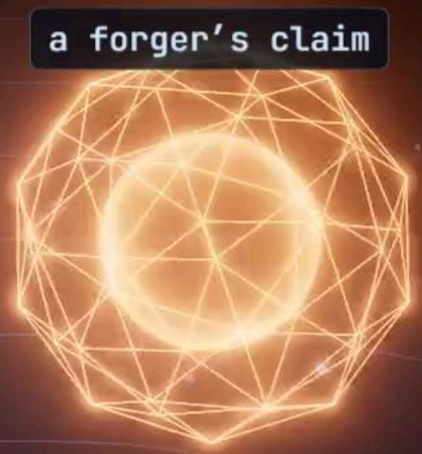
A model watermark is a secret only the owner can test.

Its limit: a mark shows whose model it is. It says nothing about how the model was trained.



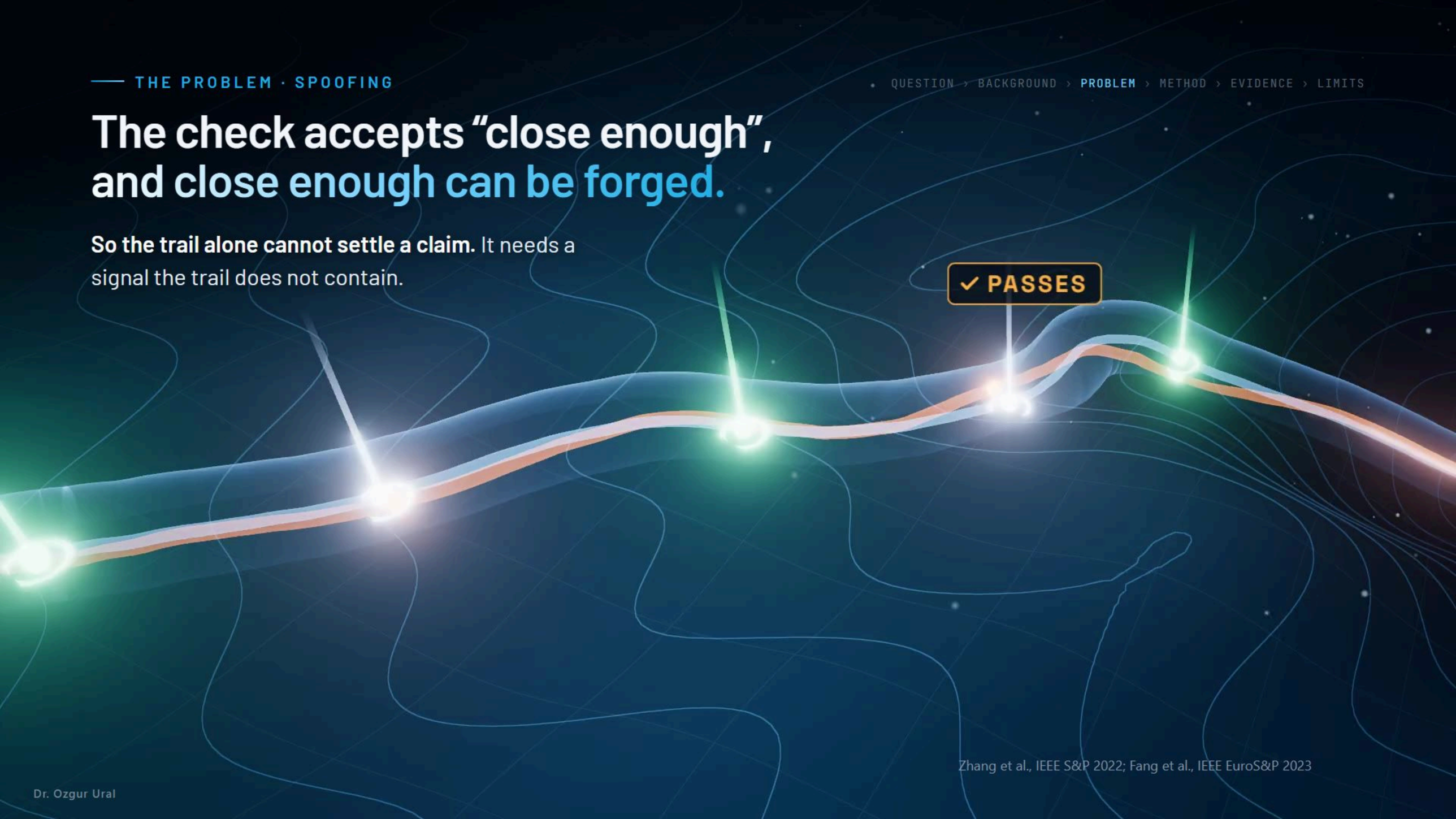
A street that **doesn't exist**, hidden in the map to catch anyone who copies it.





The check accepts “close enough”, and close enough can be forged.

So the trail alone cannot settle a claim. It needs a signal the trail does not contain.



Zhang et al., IEEE S&P 2022; Fang et al., IEEE EuroS&P 2023

Each tool answers one question. Can one claim carry both?

	Did they train it?	Is it theirs?	Weak point
Proof-of-Learning	✓	✗	the trail can be forged inside the tolerance
Model watermark	✗	✓	says nothing about the training
Both, in one recorded run	?	?	the question of this study

RQ1 · DOES IT WORK?

Joined in one run, does each check catch what the other misses?

H1: forged trails fail the mark; erased marks break the trail.

RQ2 · WHAT DOES IT COST?

Does it stay inside four budgets set in advance?

- accuracy −1 point at most
- training +20% at most
- the honest owner always passes
- proof under 12 MB

The attacker may know everything except the owner's secret

Attacker	Can query the model	Sees some snapshots	Has the weights and full trail	Knows the watermark method	Knows the owner's secret
Outsider	✓	✗	✗	✗	✗
Partner	✓	✓	✗	✗	✗
Insider	✓	✓	✓	✓	✗

THE CARELESS ATTACKER

Edits a copied model, for example by fine-tuning it, but does not rebuild the trail.

Caught by the trail check

THE CAREFUL ATTACKER

Keeps the trail consistent and also tries to fake the watermark.

The attacker this work is designed for

Learn the mark inside the recorded training. Accept only if both checks pass.

A forger's trail can pass, but the forged model cannot answer. Rejected.

	TRAIL	SECRET	VERDICT
The owner	✓	✓	ACCEPTED
A forger	✓	✗	REJECTED



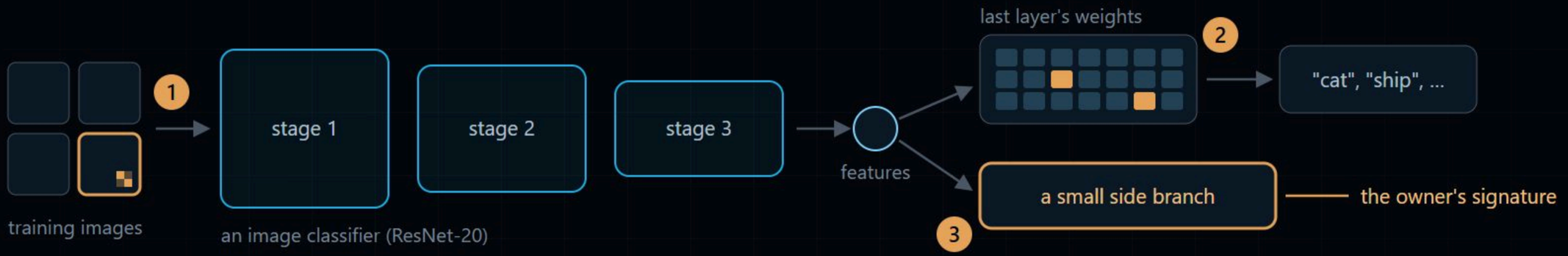
THE SAME, IN SYMBOLS

TRAIN minimise $L_{\text{task}} + \lambda \cdot L_{\text{mark}}$

ACCEPT re-run error $\leq \delta$ **and** mark score $\geq \tau$

λ: how strongly the mark is learned · δ, τ: the thresholds of the two checks

Where the secret lives decides how it is checked and what it costs



1 · TRIGGER IMAGES

Secret inputs, planted answers

The cat that answers "ship". Checked just by asking the model, so it works even when the model is only reachable online.

Costs the most training time

2 · WEIGHT SIGNATURE

A pattern in two weights

A signature is written into two of the model's numbers. Checked by reading those numbers directly.

Almost free in time; costs a little accuracy

3 · SIDE BRANCH

A signature beside the task

A small extra branch learns the mark, beside the classifier's own task.

Almost no accuracy cost; moderate time

Three established designs, one from each family, compared under the same training. The contribution is joining them to Proof-of-Learning, not new watermarks.

Each check covers the other's blind spot.

Erase the secret, and the model changes: it no longer matches its own trail.



AND THE OTHER WAY ROUND

✓ X X ✓ X X X X X

Fake the trail instead, and the secret remains. With trigger images, a forger who never saw the questions can only guess. One lucky answer is easy; twenty questions with ten possible answers each is a chance of **one in 10^{20}**.

Forged snapshots fooled the trail check. The secret check caught every one.



Check 1 alone: each forged snapshot sits just inside the tolerance, so Proof-of-Learning accepts the forgery.

Check 2, at the same moments: the forged model does not carry the owner's secret, so the claim is rejected.

Drawn after Figure 20 of the paper: a forgery attack on Proof-of-Learning ("blindfold Top-Q"), image classifier on CIFAR-10.

Attacks that passed one check were stopped by the other

WHAT THE ATTACKER TRIED	STOPPED BY	WHAT HAPPENED
A forged trail that stays inside the tolerance	the secret	It passed the trail check, then failed the secret check
Retraining a copy without a new trail	the trail	The trail check failed at once
Retraining it, or training a copycat, to erase the secret	its cost	Not blocked, but costly: typically more than 5% of accuracy lost
Attacking both checks at once	both	Either the secret held, or the trail check broke first

FORGED TRAILS ACCEPTED, IN OUR TESTS

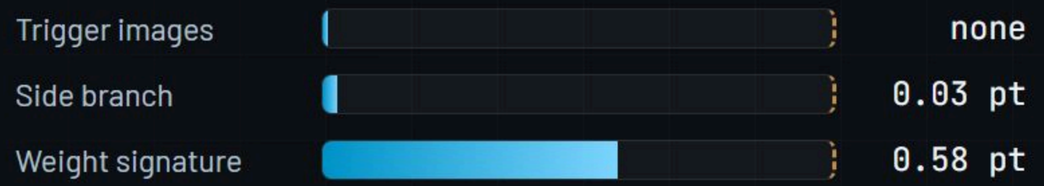
some → none

With the trail check alone, some forgeries were accepted. With both checks, none were.

All three stayed well inside the budgets set in advance

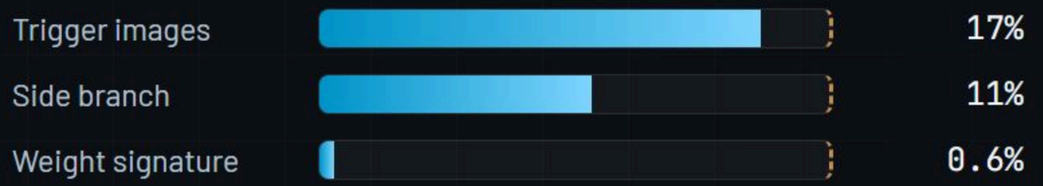
Accuracy lost

in percentage points; the right edge is the budget: 1 point



Extra training time

the right edge is the budget: 20%



The honest owner passed both checks every time

The whole proof fits in under 12 MB

No design wins on everything: time, accuracy or flexibility

What did not work: writing the signature into 20 weights instead of two wrecked the model (54% accuracy).

Joined in one recorded run, each check covers the other.

	Did they train it?	Is it theirs?	What we found
Proof-of-Learning	✓	✗	the trail can be forged inside the tolerance
Model watermark	✗	✓	says nothing about the training
SecurePoL	✓*	✓	forged trails failed the mark; erased marks broke the trail

✓ H1 supported

Each check caught what the other missed, and an attack on both broke the trail first.

✓ H2 supported

Worst case 0.58 points of accuracy and 17% more training time; the honest owner always passed; the proof stayed under 12 MB.

* Against an attacker without the owner's key. On one benchmark (CIFAR-10, ResNet-20), for the attacks tested. Five seeds per configuration: the mark did not change where training ends up.

What the study does not show, and the research it opens

One small benchmark. A compact image classifier, not a language model.	→ Test it on large models and on fine-tuning
The trail grows with the model. For a billion-parameter model, terabytes of snapshots.	→ Store only what changes; record selected layers
Known attacks only. It makes them expensive; it does not prove every attack fails.	→ Tie the secret into the trail itself, so neither can change alone
Cryptographic proofs are stronger, but far more expensive, and they do not show whose model it is.	→ Combine with zero-knowledge proofs and trusted hardware

1. Why Proof-of-Learning's tolerance can be exploited, and why a second, independent signal is needed
2. SecurePoL: a secret planted in the recorded training, and a claim accepted only if both checks pass
3. A controlled comparison of three watermark designs under the same training, with budgets set in advance and repeated runs
4. Open code and configurations that reproduce the results:
   `github.com/ozgurural/SecurePoL-with-Watermarking`

A forged training record must now also forge a secret it never saw.