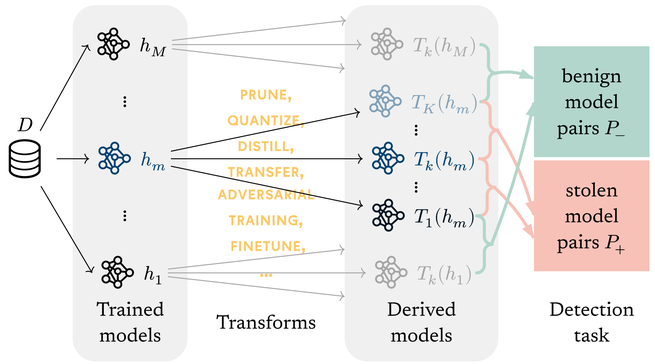
Queries, Representation & Detection: The Next 100 Model Fingerprinting Schemes

IRISA · Université de Rennes · Inria · PEReN Pôle d'Expertise de la Régulation Numérique · HI! PARIS HEC

The lemons: faithful benchmarks

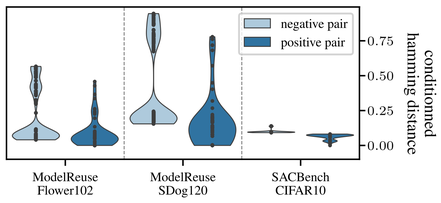


True Positive Rate $\mathrm{TPR}(\mathcal{T})$ **easy** estimation.

False Positive Rate $\mathrm{FPR}(\mathcal{T})$ **intractable!**

Existing benchmarks

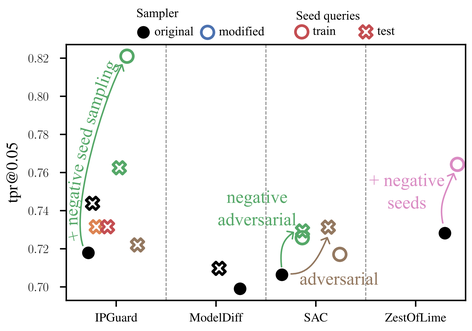
	t_j	ModelReuse Flower102	ModelReuse SDog120	SACBench CIFAR10
model leak	same	✓	✓	✓
	quantize	✓	✓	×
	finetune	×	×	✓
	transfer	×	×	✓
	prune	✓	✓	✓
extraction	probits	✓	✓	✓
	label	✓	✓	✓
	adversarial	×	×	✓



(Conditionned) Hamming distance

$$d_H(h,h') = \mathbb{P}(h(x) \neq h'(x))$$
$$d_C(h,h') = \mathbb{P}(h(x) \neq h'(x) \mid h(x) \neq y)$$

Improving existing fingerprints



How far can we go ?

- Negative Sampling greatly improve existing methods
- Representation doesn't matter much
- Using train samples helps adversarial-based methods
- For the rest, test samples are better.

Model Stealing detection

$h : \mathcal{X} \to \mathcal{Y}$ = platform's model $h' : \mathcal{X} \to \mathcal{Y}$ = suspected model

Objective: Design a test $\mathcal{T}$ such that

Effectiveness: if $h = h'$,
$\mathbb{P}(\mathcal{T}(h,h') = 1) > 2/3$ Stolen model!

Uniqueness: if $h \neq h'$,
$\mathbb{P}(\mathcal{T}(h,h') = 0) > 2/3$ Just another model...

Lemon QuRD fingerprinting recipe

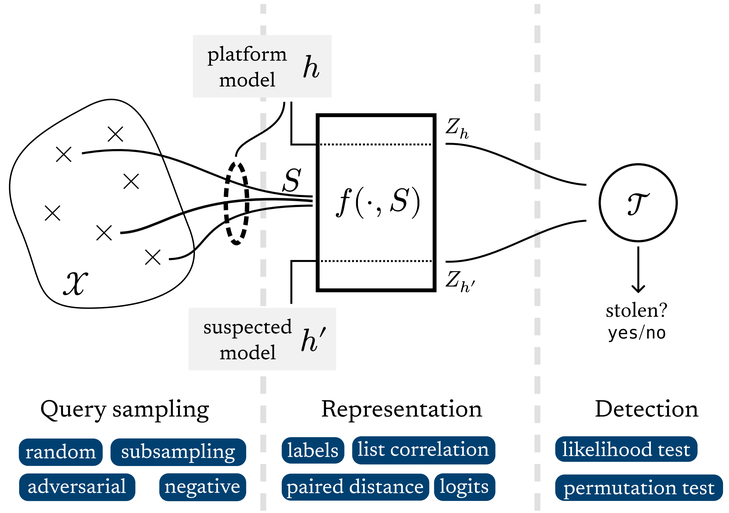
Step 1. *Prepare* your model h and get query access to the suspected model h'.

Step 2. *Measure* 50g of **queries** $S \in \mathcal{X}$. You can adapt the number of queries depending on your budget.

Step 3. *Cook* the **representations** $Z = f(h,S)$ (resp. Z') of your model h and the suspected model h'.

Step 4. *Taste* the difference between your model h and the suspected model h' using your **detection** fork $T(Z,Z')$.

Ingredients



The Anna Karenina Heuristic

All happy families look alike.

– Anna Karenina, Tolstoï

Fingerprint $\mathcal{T}_{\mathrm{AKH}}(h,h')$

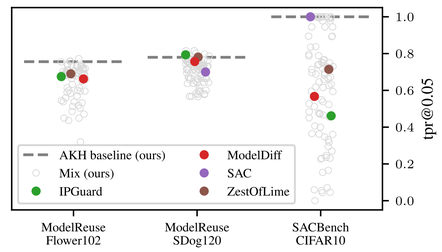
1. Sample $x \overset{\text{i.i.d.}}{\sim} D_{\mathrm{neg}}$
2. **If** $h(x) = h'(x)$
   - **return** `Stolen`
3. **else return** `Benign`

Proposition: $\mathcal{T}_{\mathrm{AKH}}$ enjoys one-sided error-rate. If $h \neq h'$,

$$\mathbb{P}(\mathcal{T}_{\mathrm{AKH}}(h,h') = \texttt{Stolen}) = d_C(h,h') \geq \frac{d_H(h,h') - \mathrm{error}(h')}{\mathrm{error}(h)}$$

AKH: a strong baseline

- ● = one exiting fingerprint
- ○ = one of the Next 100 Fingerprinting Schemes™
- – – = the AKH baseline
- One column = one benchmark



Code example

```
smol_bench =
get_benchmark("TinyImageNetModels")
runner = Experiment(smol_bench)
akh = make_fingerprint("AKH")

print(runner.scores(akh, budget=10))
```

Easy install (❤ pixi) Model weights + datasets (❤ Huggingface)

All in **one** line: `pixi r bench scores TinyImageNetModels "AKH"`

What now ?

1. Existing benchmarks are too simple $\overset{\text{SOON}}{\Rightarrow}$ **new benchmarks**
2. Dominant focus on representations $\overset{\text{SOON}}{\Rightarrow}$ **better detectors**
3. No analysis of failure cases $\overset{\text{SOON}}{\Rightarrow}$ **theoretical guarantees**

Augustin Godinot, Gilles Tredan, Erwan Le Merrer, Camilla Penzo, Francois Taiani augustin.godinot@inria.fr – grodino.github.io/