TRAK:

Attributing Model Behavior at Scale

Sung Min (Sam) Park*

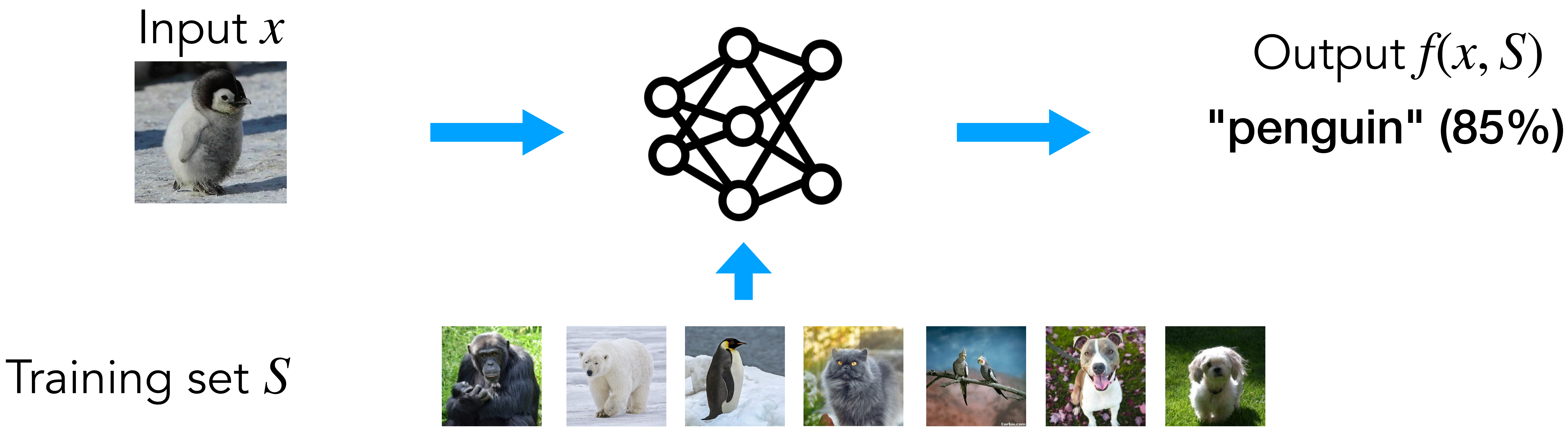
Kristian Georgiev*

Andrew Ilyas*

Guillaume Leclerc

Aleksander Mądry

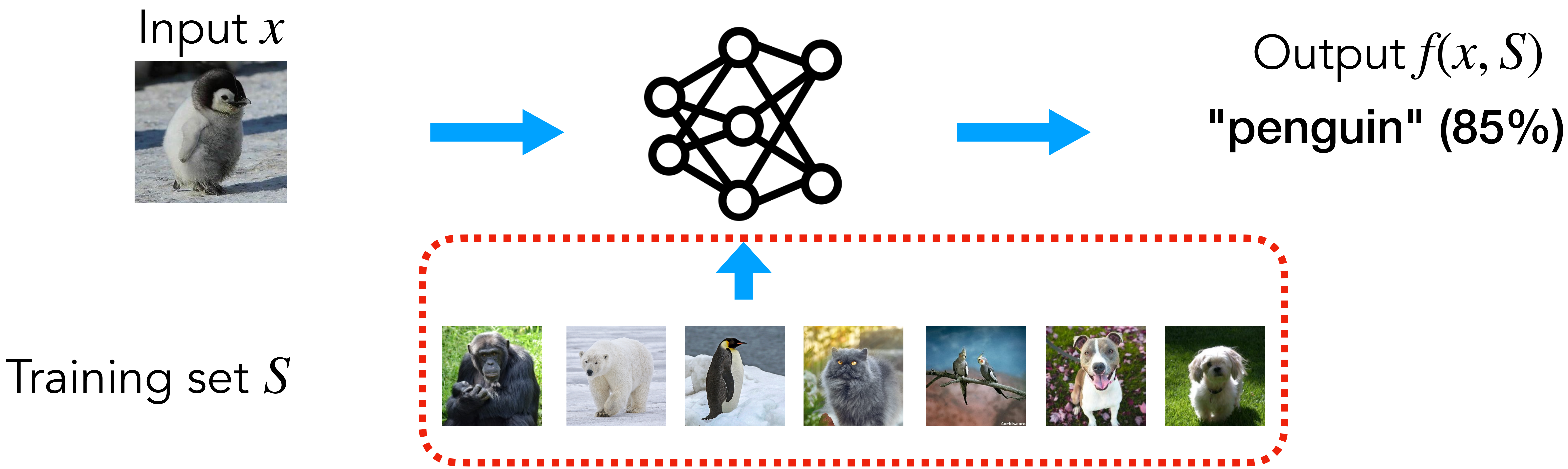
The ML pipeline



We think of **model output** as a function of the **input**

…but it is also function of the **training data**!

The ML pipeline

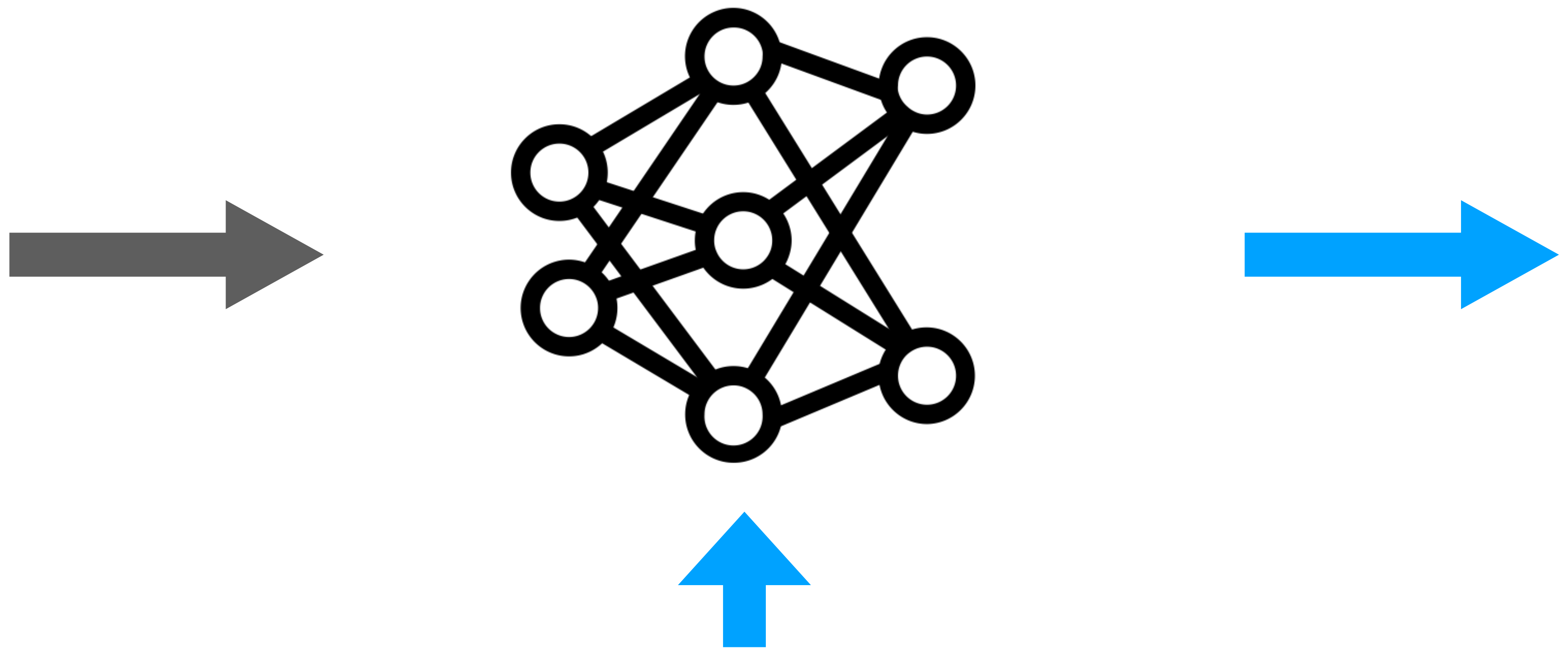


Q: How does training data affect model predictions?

A: Data attribution methods

Data attribution

Input x



Output $f(x, S)$

"penguin" (85%)

Training set S

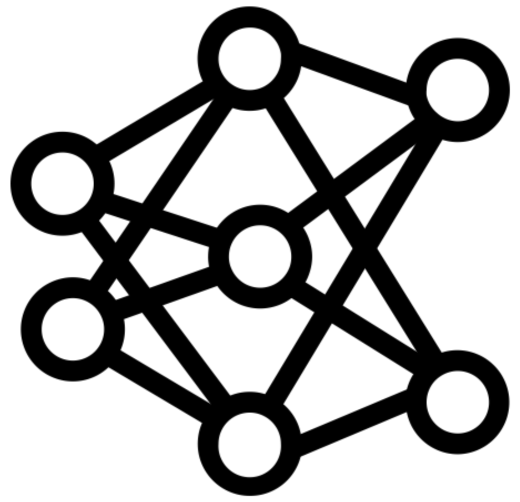
Data attribution $\tau(x) =$ [0.1 -0.4 0.8 -0.1 -0.3 0.2 -0.1]

$\tau(x)_i =$ "importance" of i^{th} training example on output $f(x, S)$

What does it mean to do this "well"?

Data attribution

Input x



Output $f(x, S)$

"penguin" (85%)

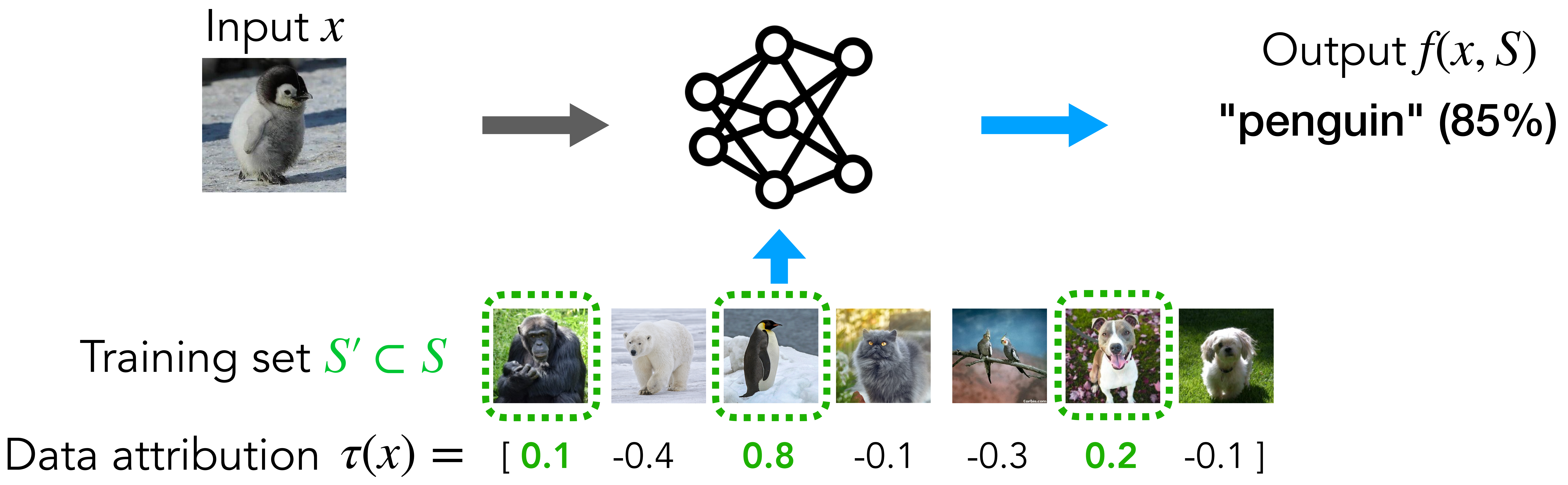
Training set S

Data attribution $\tau(x) =$ [0.1 -0.4 0.8 -0.1 -0.3 0.2 -0.1]

Intuitive goal: Scores should capture examples' **counterfactual** impact

[Ilyas **P** Engstrom Leclerc Madry '22]

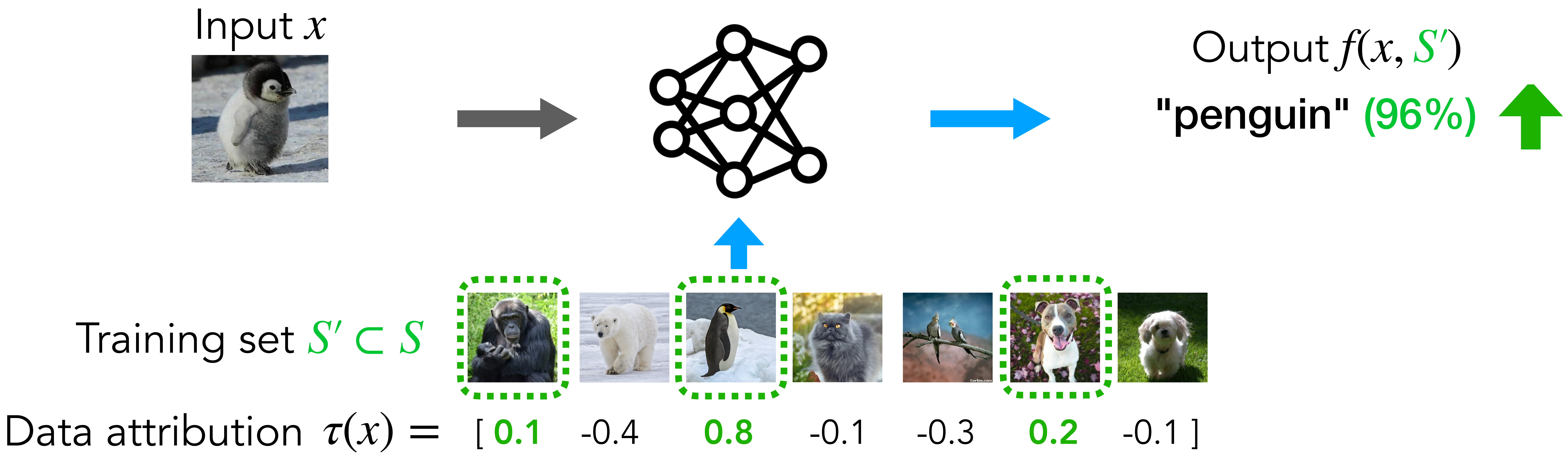
Data attribution



Intuitive goal: Scores should capture examples' **counterfactual** impact

[Ilyas **P** Engstrom Leclerc Madry '22]

Data attribution

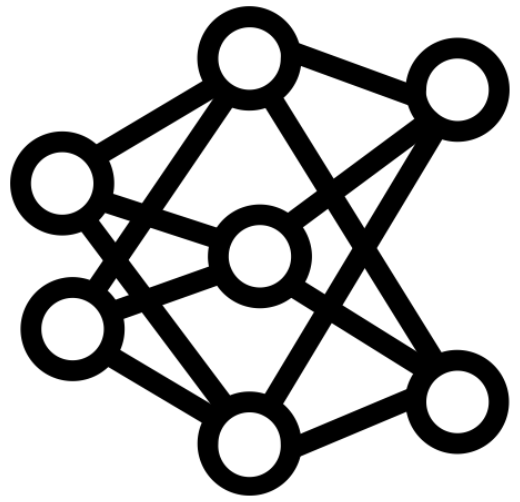


Intuitive goal: Scores should capture examples' **counterfactual** impact

[Ilyas **P** Engstrom Leclerc Madry '22]

Data attribution

Input x



Output $f(x, S')$

"penguin" **(96%)**

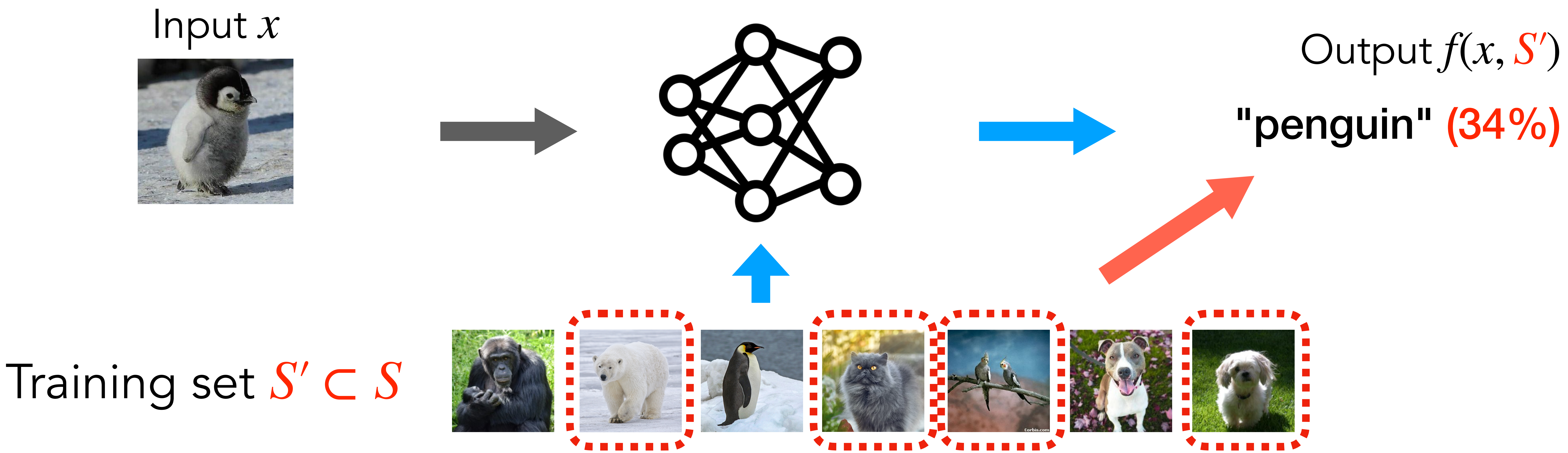
Training set $S' \subset S$

Data attribution $\tau(x) =$ [0.1 **-0.4** 0.8 **-0.1** **-0.3** 0.2 **-0.1**]

Intuitive goal: Scores should capture examples' **counterfactual** impact

[Ilyas **P** Engstrom Leclerc Madry '22]

Data attribution



Data attribution $\tau(x) =$ [0.1 **-0.4** 0.8 **-0.1** **-0.3** 0.2 **-0.1**]

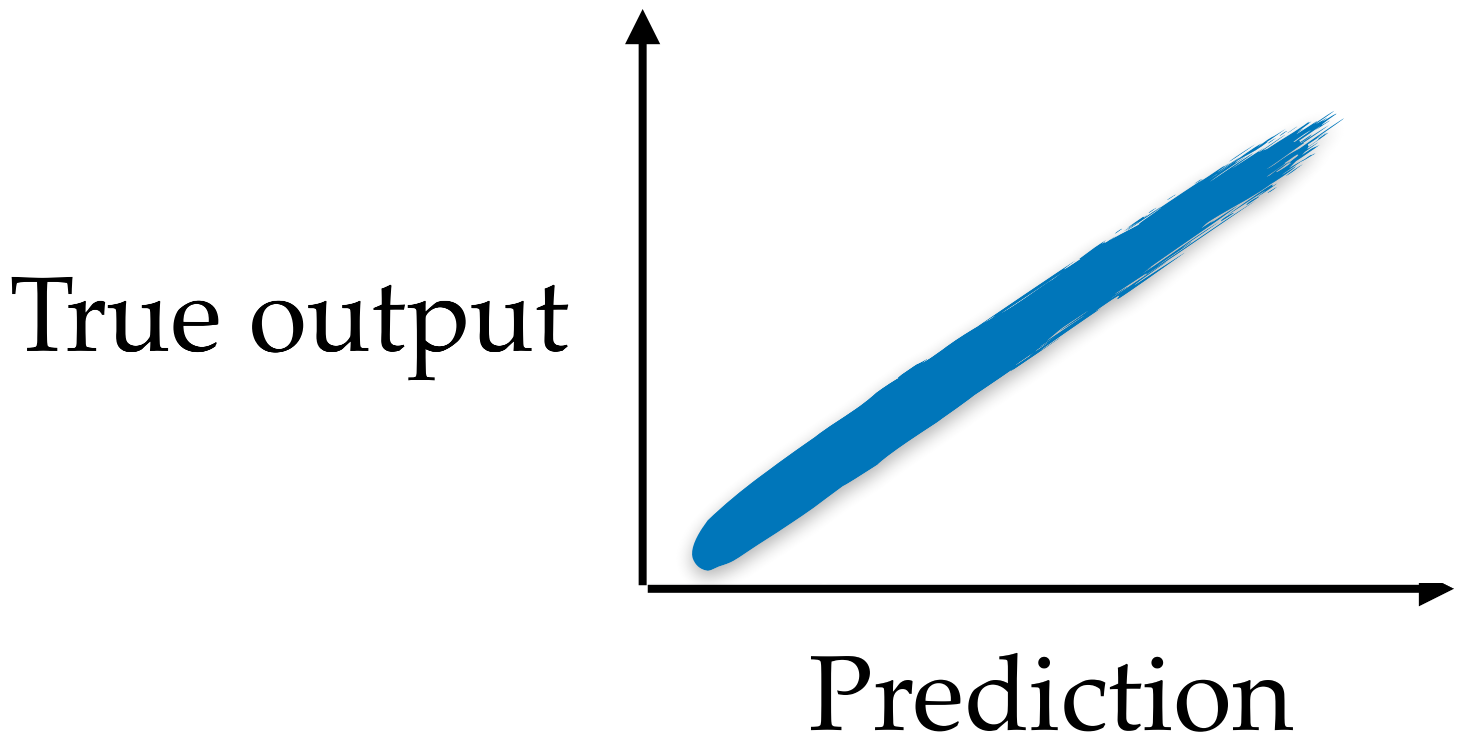
Datamodeling score:

Given training set $S' \subset S$, how predictive of $f(x, S')$ is τ?

[Ilyas **P** Engstrom Leclerc Madry '22]

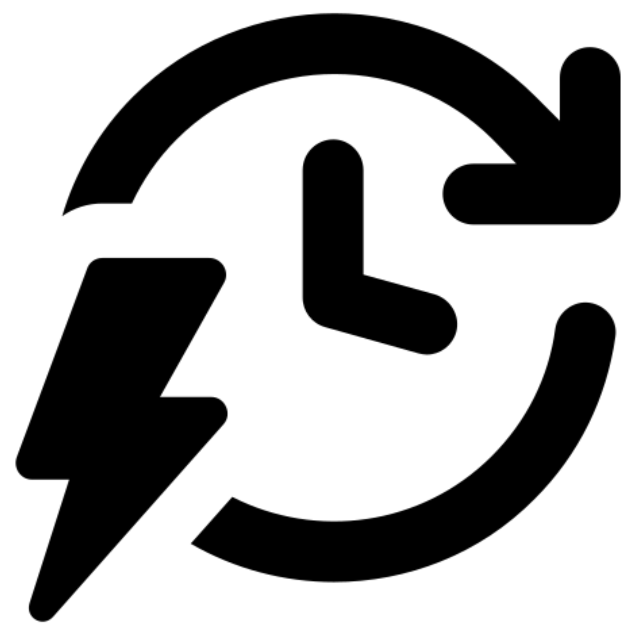
Goals of data attribution

Predictive



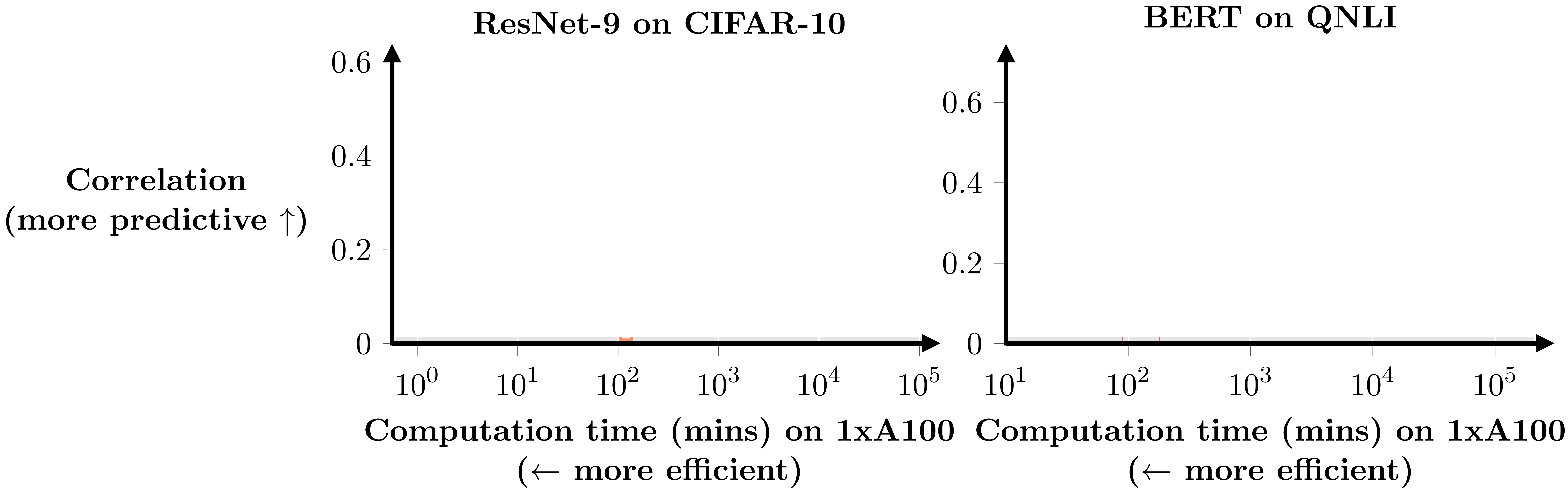
Can accurately predict counterfactual outputs

Efficient

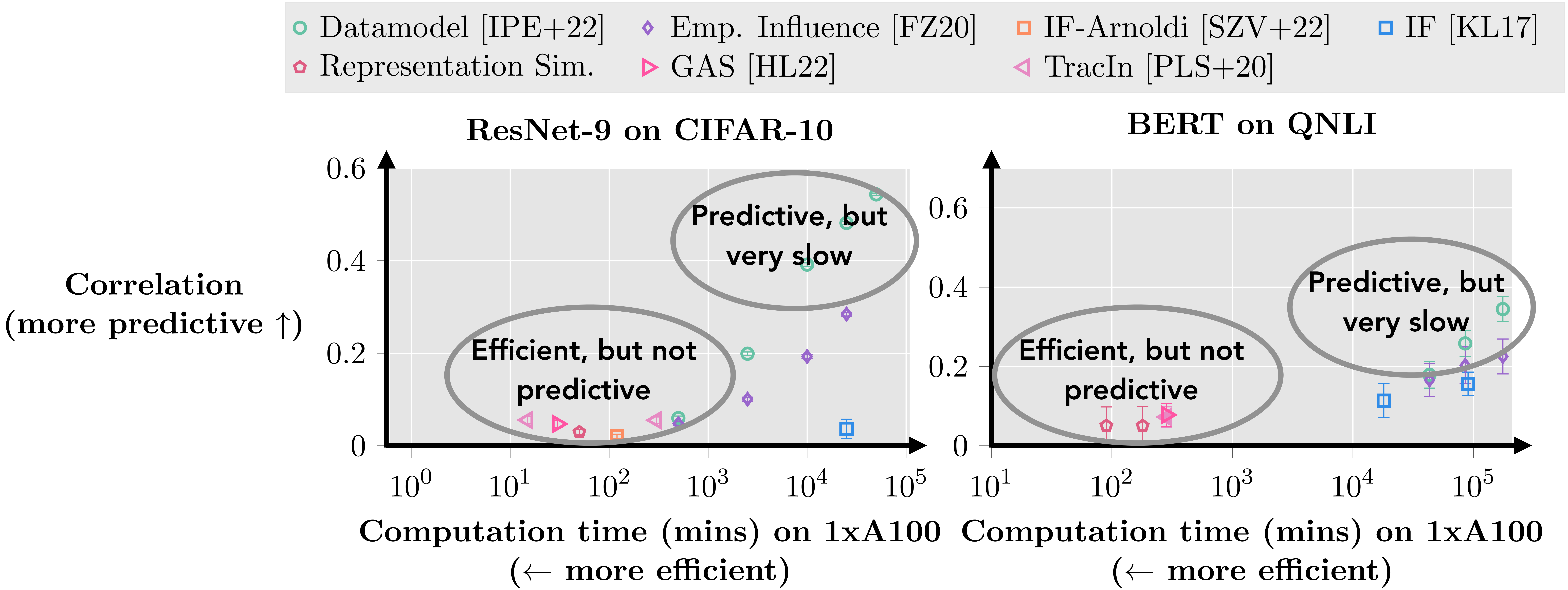


Can compute τ efficiently

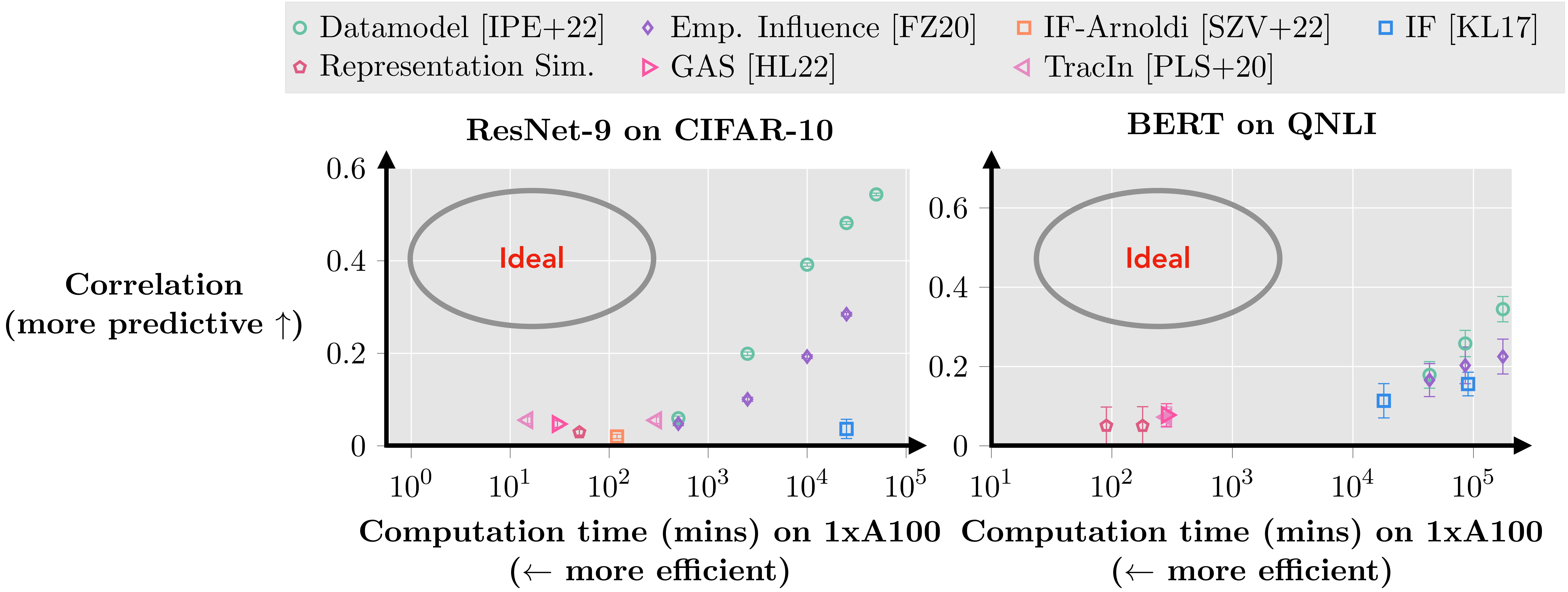
Evaluating attribution methods



Evaluating attribution methods

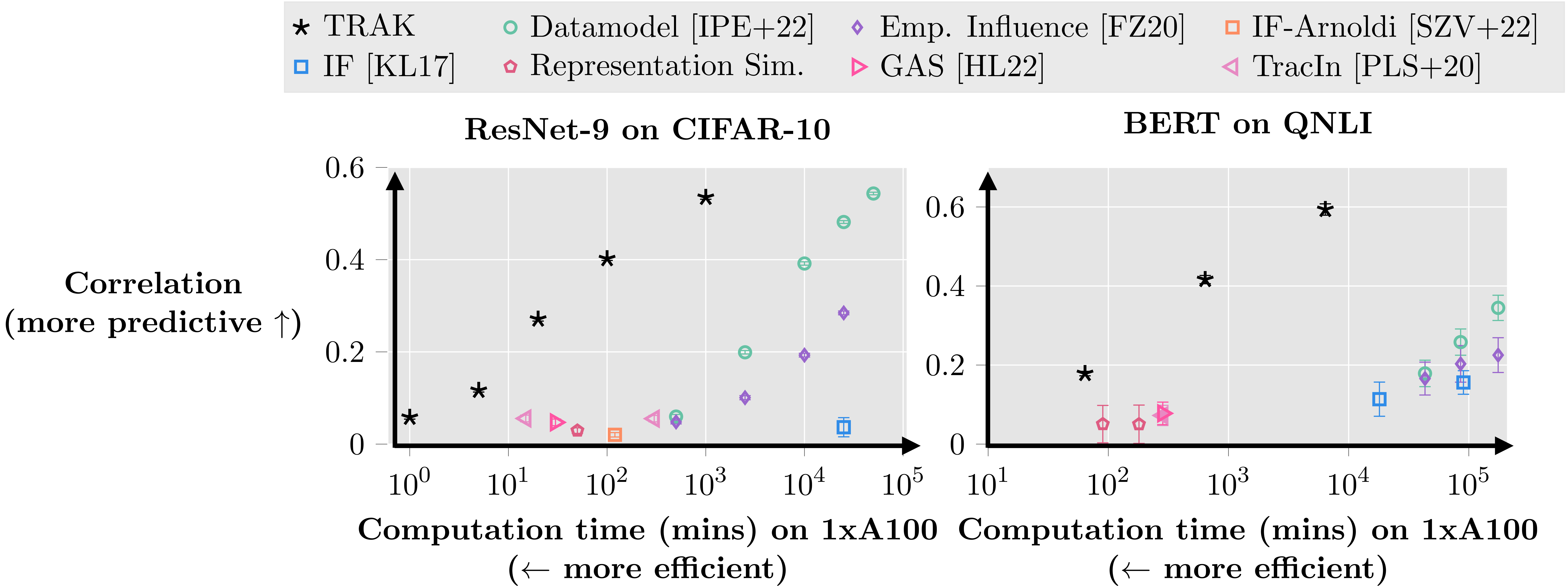


Evaluating attribution methods

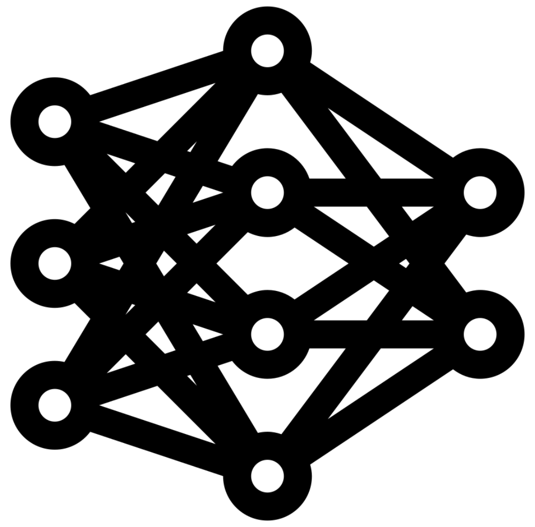


Q: Can we design an attribution method that is both **predictive** *and* **efficient**?

Yes! With TRAK



Our approach: TRAK

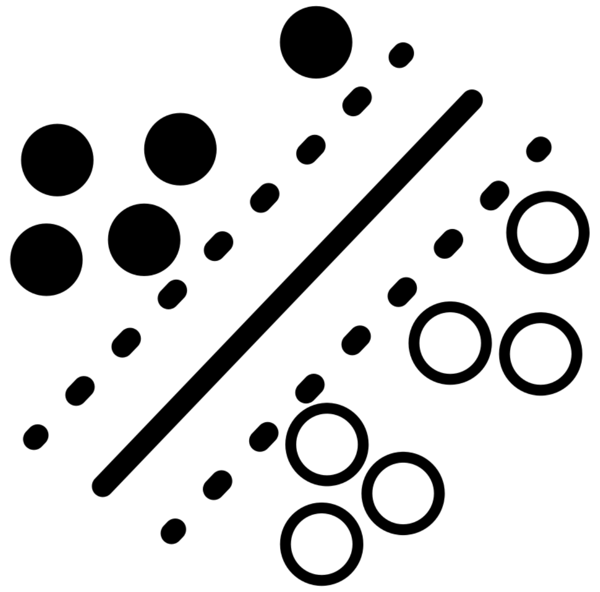


Input: example x

Output: $h(x; \theta)$

Can be arbitrarily complicated

Differentiable model



Input: $\nabla_\theta h(x; \theta^\star)$

Output: $\nabla_\theta h(x; \theta^\star)^\top \theta$

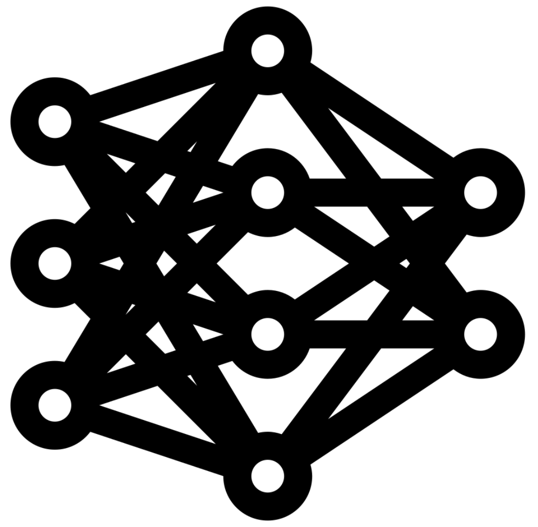
Generalized linear models

Our approach: First-order Taylor approximation around **final** parameters

$$h(x, \theta) \approx h(x; \theta^\star) + \nabla_\theta h(x; \theta^\star) \cdot (\theta - \theta^\star)$$

Final parameters (constant wrt θ)

Our approach: TRAK

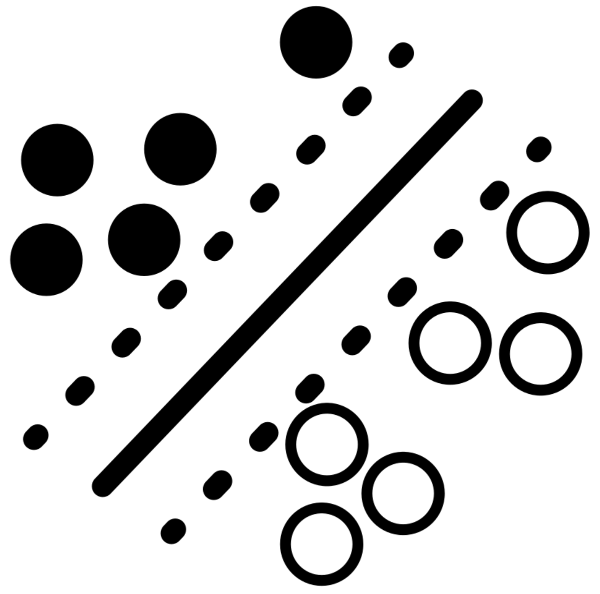


Input: example x

Output: $h(x;\theta)$

Differentiable model

Can be arbitrarily complicated



Input: $\nabla_\theta h(x;\theta^\star)$

Output: $\nabla_\theta h(x;\theta^\star)^\top\theta$

Generalized linear models

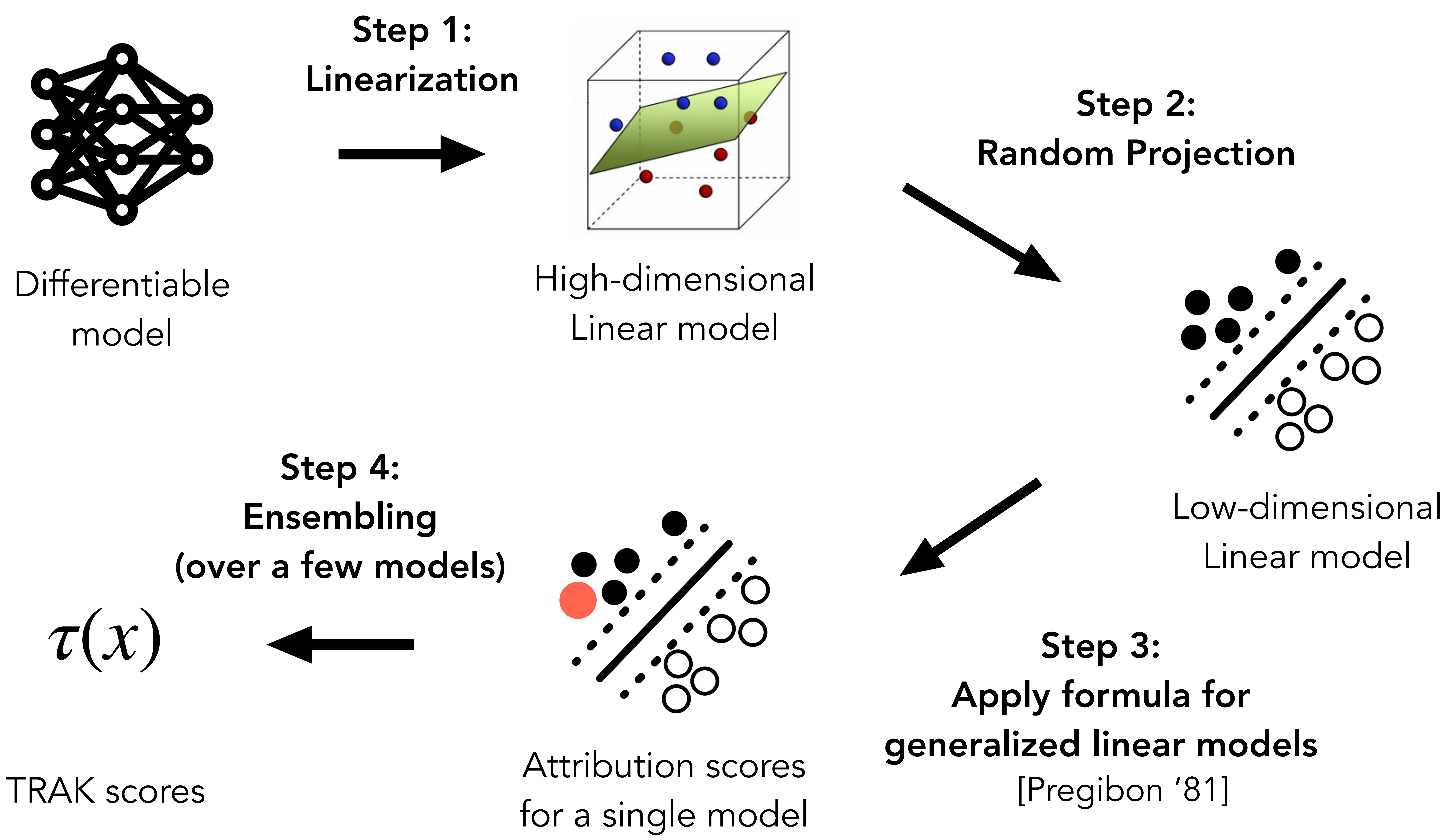
Our approach: First-order Taylor approximation around **final** parameters

$$h(x,\theta) \approx h(x;\theta^\star) + \nabla_\theta h(x;\theta^\star)\cdot(\theta-\theta^\star)$$

Note: Connections to the empirical Neural Tangent Kernel (or After Kernel)

[Jacot Gabriel Hongler '18] [Long '21] [Wei Hu Steinhardt '22]

Tracing with **R**andom projections of the **A**fter **K**ernel

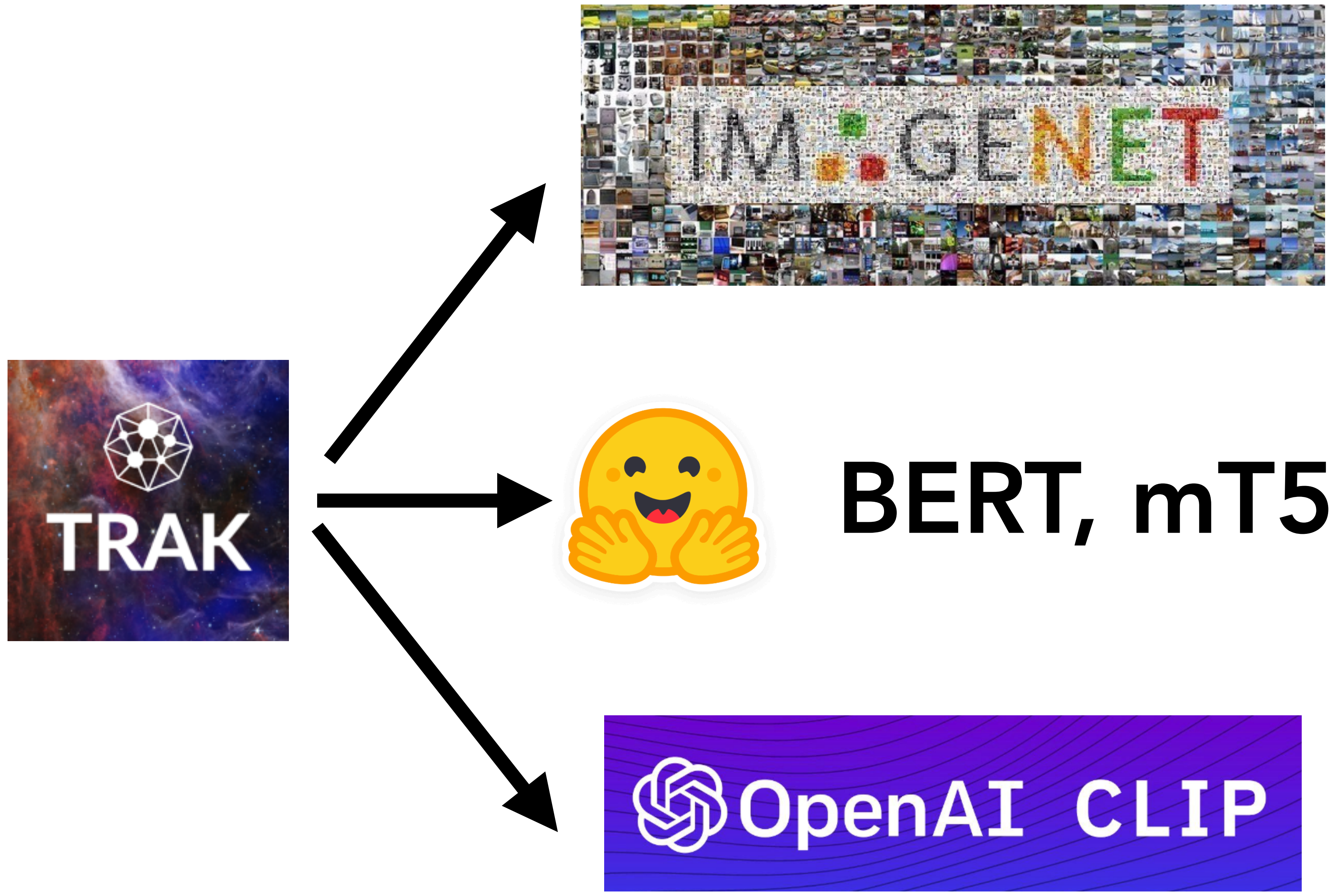


Evaluating TRAK

Applications

In our paper, we apply **TRAK** to:

- Image classifiers (ImageNet, CIFAR)
- Language models (BERT, mT5)
- Multimodal models (CLIP)



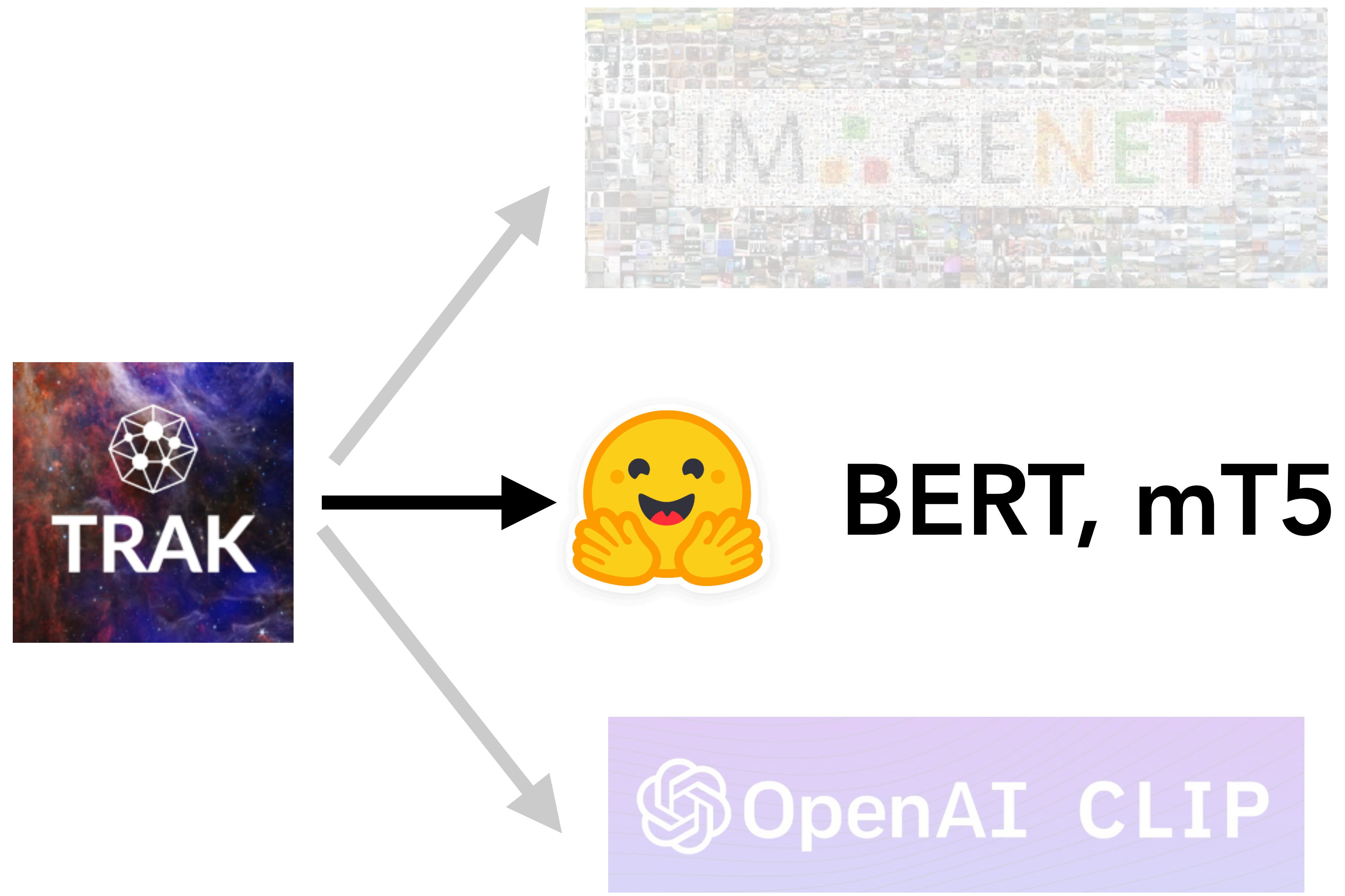

```python
from torchvision import models

from trak import TRAKer

model = models.resnet18()
checkpoint = model.state_dict()
train_loader, val_loader = ...

traker = TRAKer(model=model, task='image_classification', train_set_size=...)

traker.load_checkpoint(checkpoint)
for batch in train_loader:
    traker.featurize(batch=batch, num_samples=batch_size)
traker.finalize_features()

traker.start_scoring_checkpoint(checkpoint, num_targets=...)
for batch in val_loader:
    traker.score(batch=batch, num_samples=batch_size)
scores = traker.finalize_scores()
```

You can use it too!
https://github.com/MadryLab/trak

Applications

In our paper, we apply **TRAK** to:

- Image classifiers (ImageNet, CIFAR)
- **Language models (BERT, mT5)**
- Multimodal models (CLIP)




```
from torchvision import models

from trak import TRAKer

model = models.resnet18()
checkpoint = model.state_dict()
train_loader, val_loader = ...

traker = TRAKer(model=model, task='image_classification', train_set_size=...)

traker.load_checkpoint(checkpoint)
for batch in train_loader:
    traker.featurize(batch=batch, num_samples=batch_size)
traker.finalize_features()

traker.start_scoring_checkpoint(checkpoint, num_targets=...)
for batch in val_loader:
    traker.score(batch=batch, num_samples=batch_size)
scores = traker.finalize_scores()
```

You can use it too!
https://github.com/MadryLab/trak

Attributing Language Models

Training data

"Messi moved to Barcelona at 13."

"At Qatar 2022, Lionel Messi led Argentina to its first title in 36 years."

"Did Lionel Messi win a world cup?"

"Lionel Messi won the world cup in 2022"

Q: Why did the language model make this assertion?

Attributing Language Models

Training data	TRAK
"Messi moved to Barcelona at 13."	0.2
	-0.1
"At Qatar 2022, Lionel Messi led Argentina to its first title in 36 years."	0.7
	-0.3

"Did Lionel Messi win a world cup?"

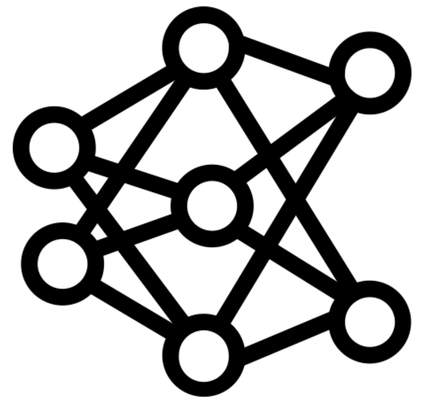
"Lionel Messi won the world cup in 2022"

To probe this: Use TRAK to attribute generated text

Attributing Language Models

Training data	TRAK	Relevant?
"Messi moved to Barcelona at 13."	0.2	✗
	-0.1	✓
"At Qatar 2022, Lionel Messi led Argentina to its first title in 36 years."	0.7	✓
	-0.3	✗

"Did Lionel Messi win a world cup?"



"Lionel Messi won the world cup in 2022"

Ground-truth: Training examples that **logically entail** output

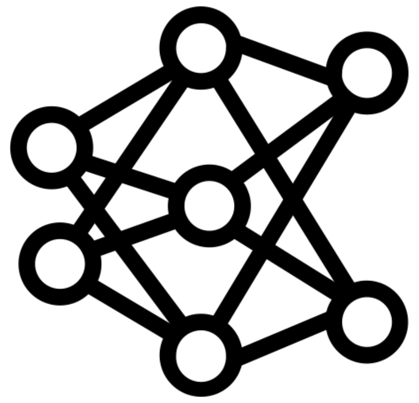
FTrace-TREx [Akyürek, Bolukbasi, Liu, Xiong, Tenney, Jacob Andreas, Guu '22]

Attributing Language Models

Training data

	TRAK	Relevant?
"Messi moved to Barcelona at 13."	0.2	✗
	-0.1	✓
"At Qatar 2022, Lionel Messi led Argentina to its first title in 36 years."	0.7	✓
	-0.3	✗

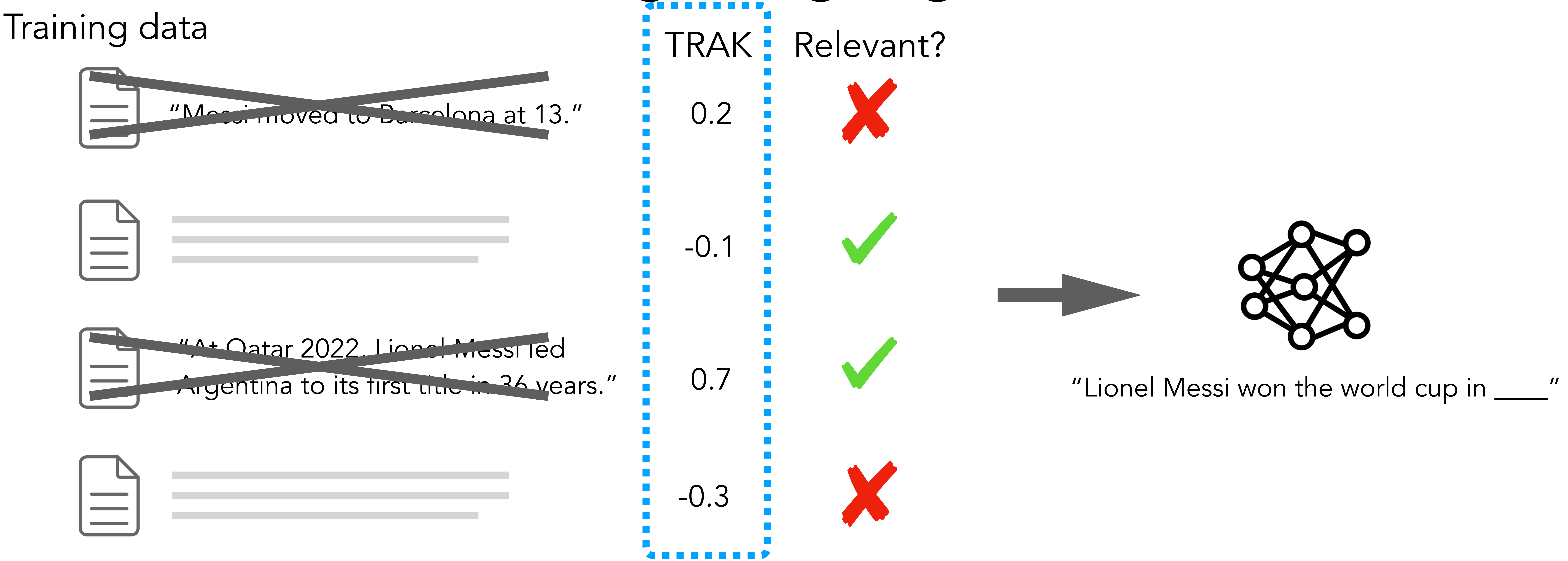
"Did Lionel Messi win a world cup?"



"Lionel Messi won the world cup in 2022"

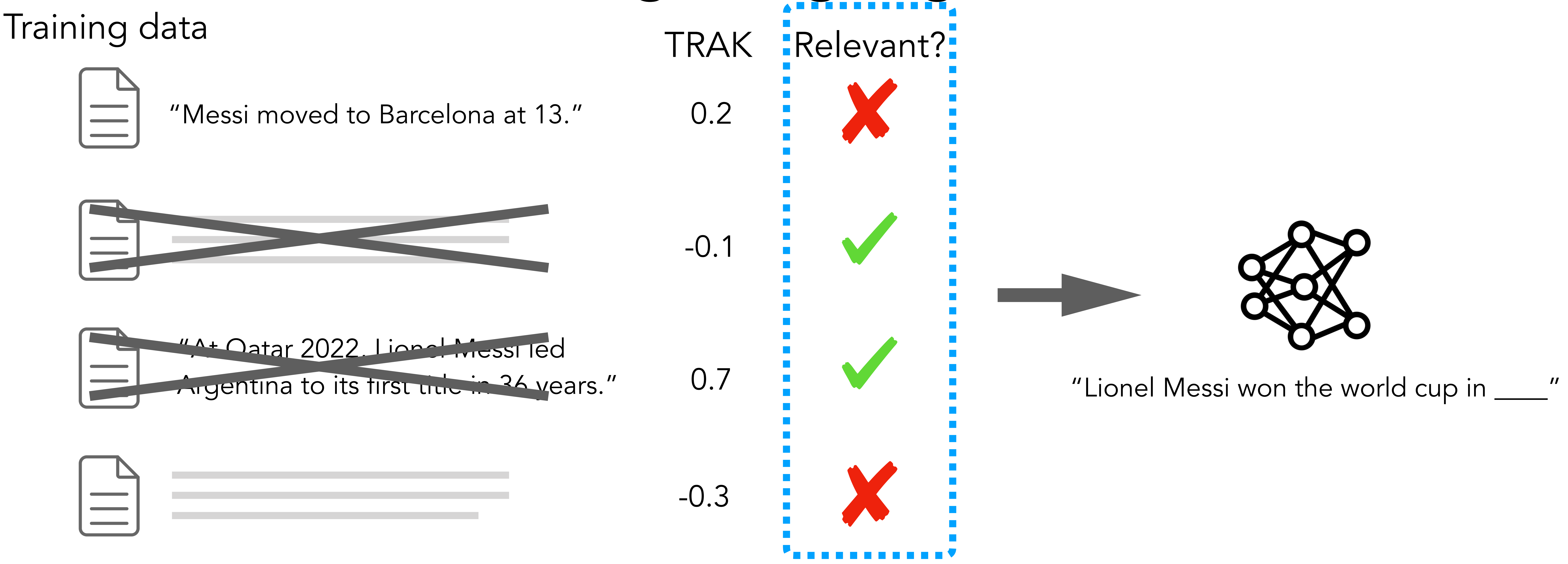
Q: How important are TRAK-attributed examples relative to "oracle"?

Attributing Language Models



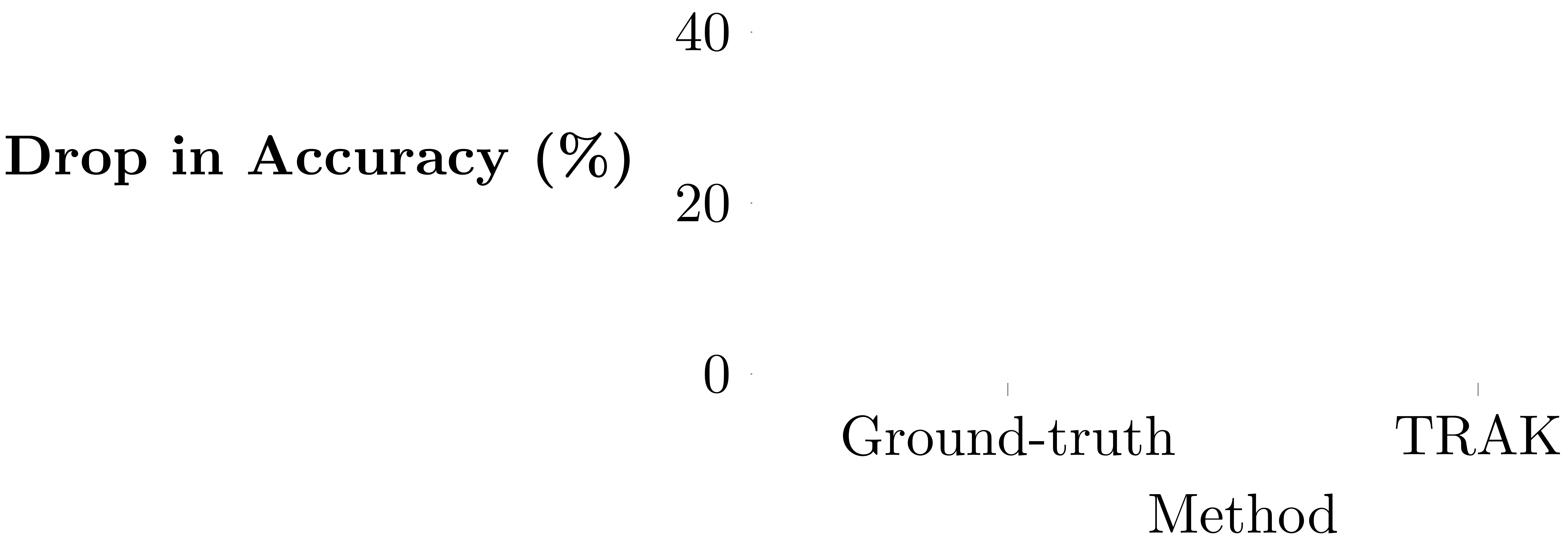
So: Remove most **attributed** examples, re-train model, evaluate factual accuracy

Attributing Language Models



relevant
So: Remove most ~~**attributed**~~ examples, re-train model, evaluate factual accuracy

Counterfactual Analysis



Overall: Fact tracing ≠ Model behavior tracing

What facts imply the generated text?
Model-**independent**

Why did the *model* generate the text?
Model-**dependent**

Takeaways

TRAK: A scalable, accurate attribution method for modern large-scale settings

→ **Data attribution:** Tracing model behavior back to training data

→ **Prior challenge:** Tradeoff between efficiency and predictiveness

→ **TRAK's main idea:** Approximate NN with a linear model

→ **Easy to apply:** Attributing language models, CLIP

Poster **#129**, Exhibit Hall 1, Thursday 1:30-3:00pm

@smsampark

trak.csail.mit.edu