TRAK:

Attributing Model Behavior at Scale

Sung Min (Sam) Park

w/ Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc,
Aleksander Mądry

@smsampark

Anatomy of an ML prediction

Anatomy of an ML prediction

Input x

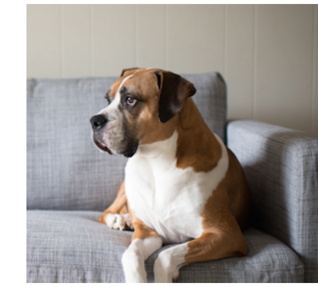
Output $f(x)$

"dog" (85%)

Anatomy of an ML prediction

Training set S

Input x



Output $f(x)$

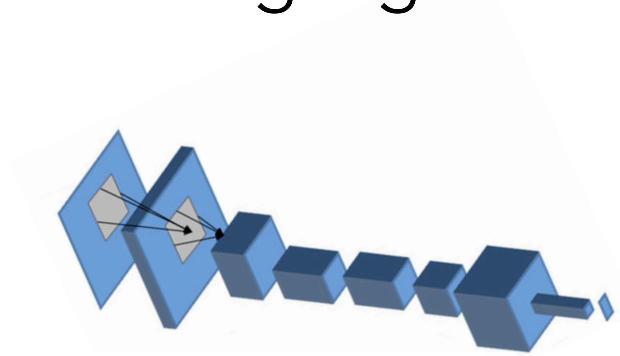
"dog" (85%)

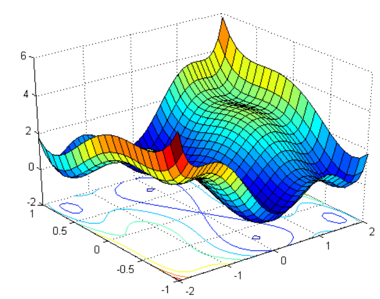
Anatomy of an ML prediction

Training set S

+

Learning algorithm





→

Input x

Output $f(x)$

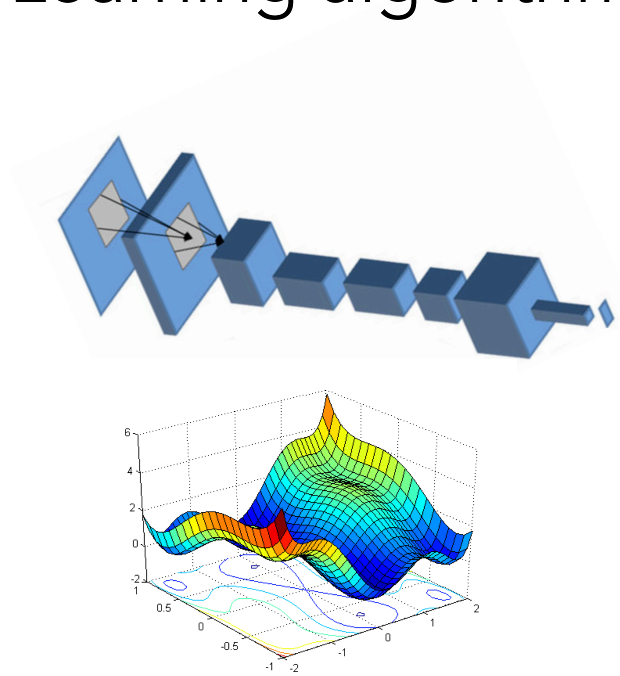
"dog" (85%)

Anatomy of an ML prediction

Training set S

+

Learning algorithm



Input x

Output $f(x)$

"dog" (85%)

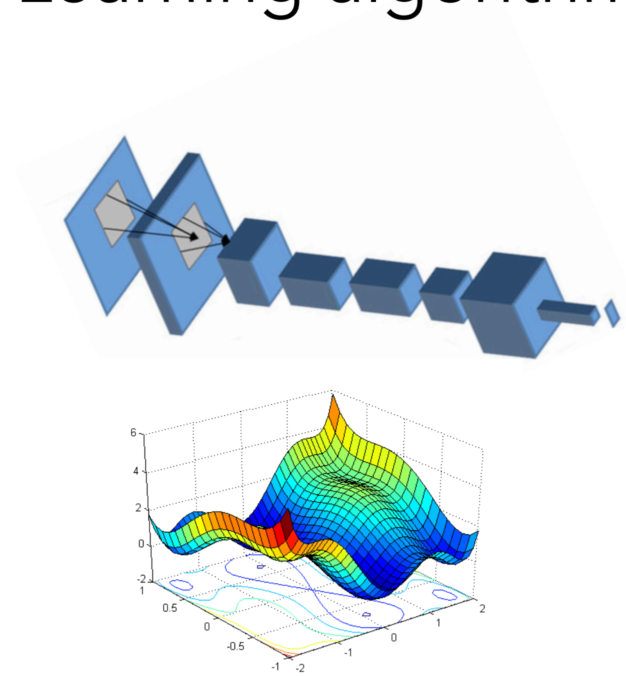
Question: How do training data and learning algorithms combine to yield model outputs?

Anatomy of an ML prediction

Training set S

+

Learning algorithm



Input x

Output $f(x)$

"dog" (85%)

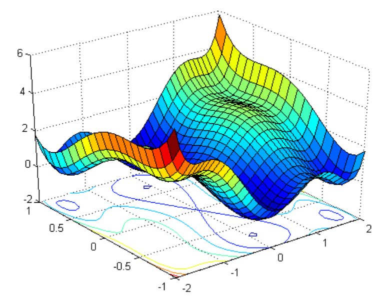
Question: How do training data and learning algorithms combine to yield model outputs?

One way to study this Q: Data attribution

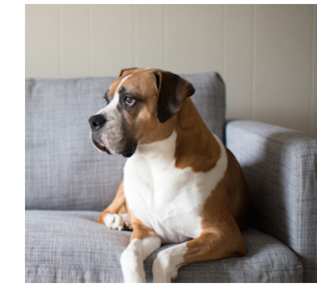
Anatomy of an ML prediction

Training set S

Learning algorithm



Input x



Output $f(x)$

"dog" (85%)

Question: How do training data and learning algorithms combine to yield model outputs?

One way to study this Q: Data attribution

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry ’22]

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Model output $f(x, S')$

Formalizing data attribution

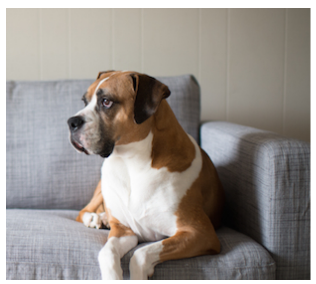
[Ilyas P Engstrom Leclerc Madry '22]

Specific example x

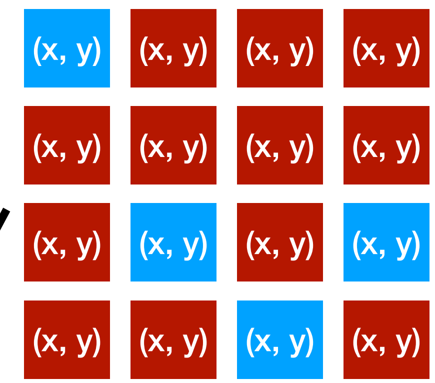
Model output $f(x, S')$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry ’22]



Specific example x

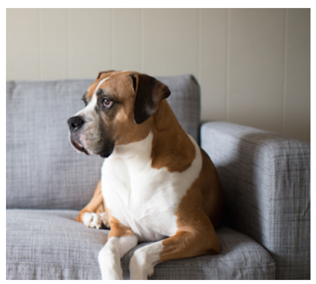


<u>Subset</u> S' of the training set S

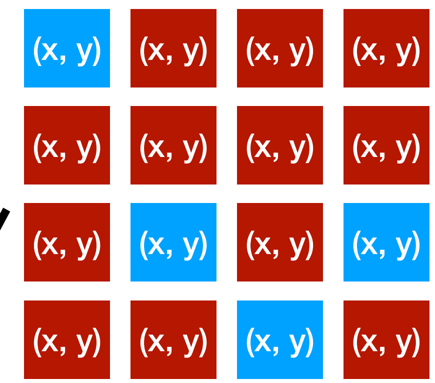
Model output $f(x, S')$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry ’22]



Specific example x



<u>Subset</u> S' of the training set S

Model output $f(x, S')$

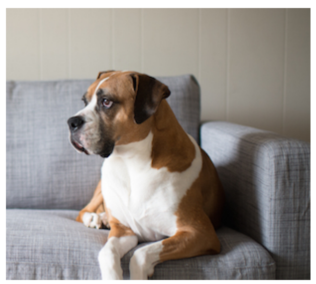
Loss of interest on x

(ex: margin of correct class)

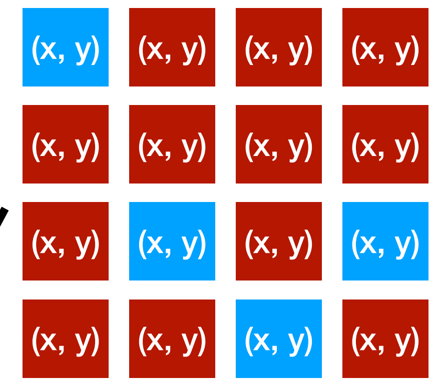
Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Goal: Understand function $S' \to f(x, S')$



Specific example x



<u>Subset</u> S' of the training set S

Model output $$f(x, S')$$

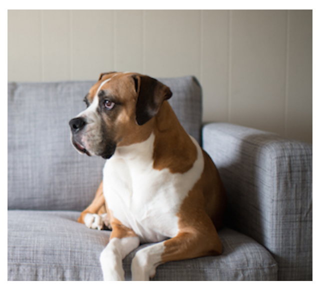
Loss of interest on x

(ex: margin of correct class)

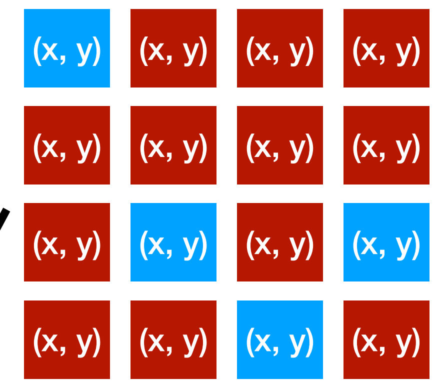
Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Goal: Understand function $S' \to f(x, S')$



Specific example x



Subset S' of the training set S

Model output $f(x, S')$

Loss of interest on x

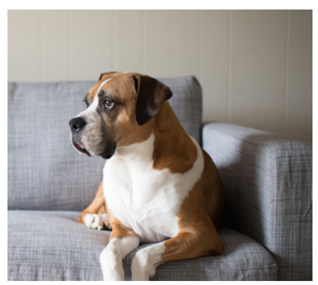
(ex: margin of correct class)

Problem: Can compute f directly, but expensive

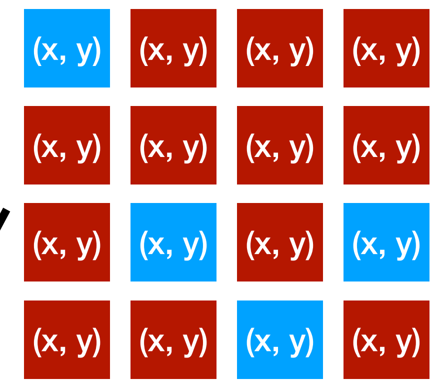
Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Goal: Understand function $S' \rightarrow f(x, S')$



Specific example x



Subset S' of the training set S

Model output $f(x, S')$

Loss of interest on x

(ex: margin of correct class)

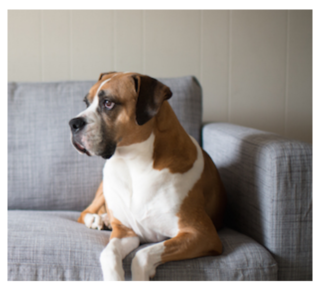
Problem: Can compute f directly, but expensive

Solution: Study a simple approximation to f

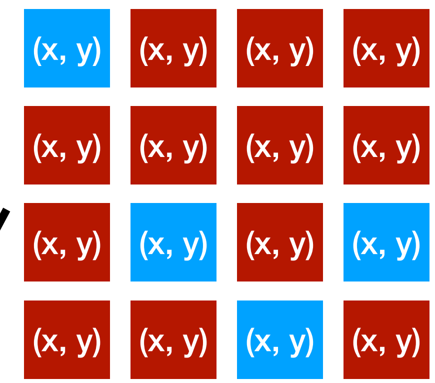
Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Goal: Understand function $S' \to f(x, S')$



Specific example x



<u>Subset</u> S' of the training set S

Model output $$f(x, S') \approx \hat{f}(x, S')$$

Loss of interest on x

(ex: margin of correct class)

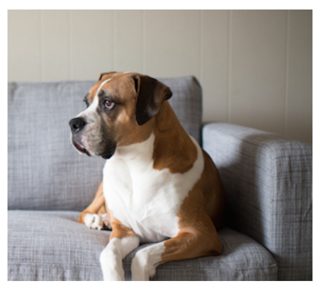
Problem: Can compute f directly, but expensive

Solution: Study a simple approximation to f

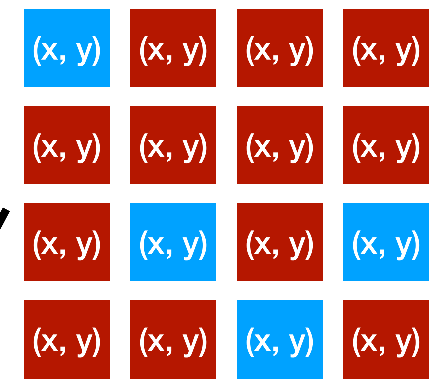
Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Goal: Understand function $S' \rightarrow f(x, S')$



Specific example x



<u>Subset</u> S' of the training set S

Model output $$f(x, S') \approx \boxed{\hat{f}(x, S')}$$ datamodel

Loss of interest on x
(ex: margin of correct class)

Problem: Can compute f directly, but expensive

Solution: Study a simple approximation to f

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

$$\hat{f}(x, S')$$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

$$\hat{f}(x, S') = \mathbf{1}_{S'} \cdot \tau(x)$$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

$$\hat{f}(x, S') = \mathbf{1}_{S'} \cdot \tau(x)$$

Indicator vector of $S' \subset S$

[1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0]

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

$\tau(x)_i$ = "effect" of training example x_i on model output at x

$$\hat{f}(x, S') = \mathbf{1}_{S'} \cdot \tau(x)$$

Indicator vector of $S' \subset S$

[1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0]

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

The approximation we use: **linear**

$\tau(x)_i$ = "effect" of training example x_i on model output at x

$$\hat{f}(x, S') = \mathbf{1}_{S'} \cdot \boxed{\tau(x)}$$

Data attribution method

Indicator vector of $S' \subset S$

[1 0 0 0 0 0 1 0 0 1 0 1 0 0 1 0]

A **data attribution method** is a function $\tau : \mathcal{X} \to \mathbb{R}^{|S|}$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

When is a data attribution method τ good?

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

When is a data attribution method τ good?

Evaluate predictiveness: Sample *new subsets* S_i,

compare actual model outputs and outputs <u>predicted</u> by τ

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

When is a data attribution method τ good?

Evaluate predictiveness: Sample *new subsets* S_i,

compare actual model outputs and outputs predicted by τ

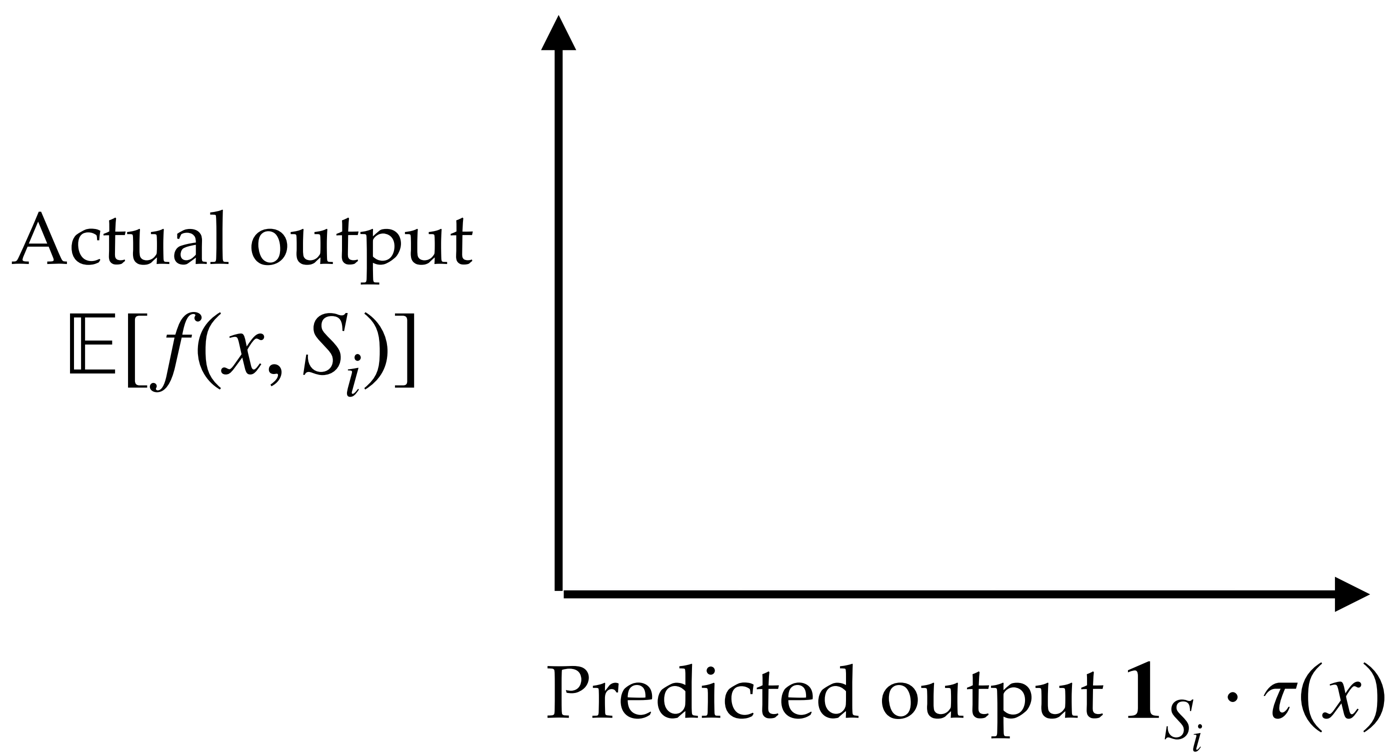
Predicted output $\mathbf{1}_{S_i} \cdot \tau(x)$

Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

When is a data attribution method τ good?

Evaluate predictiveness: Sample *new subsets* S_i, compare actual model outputs and outputs <u>predicted</u> by τ

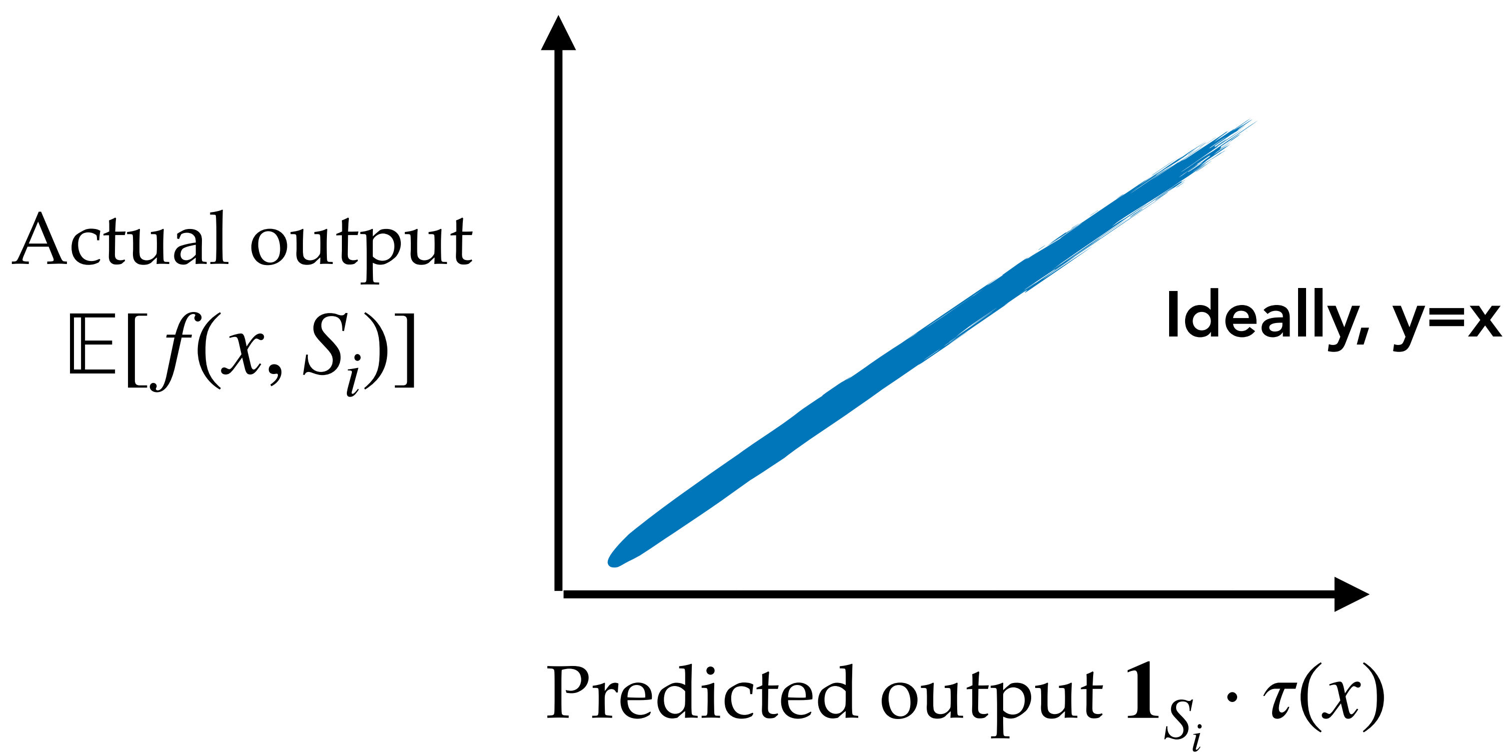


Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry ’22]

When is a data attribution method τ good?

Evaluate predictiveness: Sample *new subsets* S_i,
compare actual model outputs and outputs <u>predicted</u> by τ

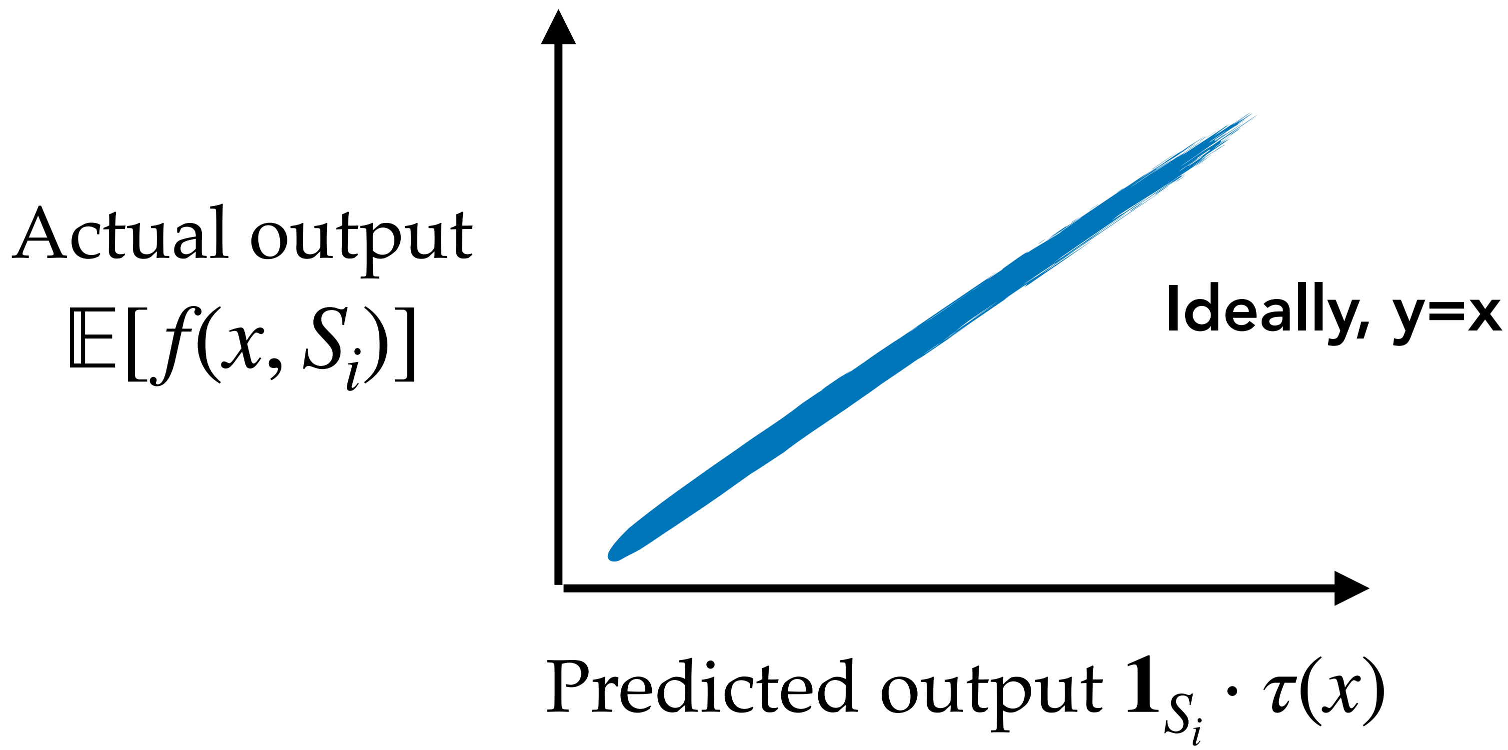


Formalizing data attribution

[Ilyas P Engstrom Leclerc Madry '22]

When is a data attribution method τ good?

Evaluate predictiveness: Sample *new subsets* S_i, compare actual model outputs and outputs predicted by τ



Metric: Correlation between actual and predicted outputs

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

Basic idea: Use supervised learning

Simple approach: datamodels

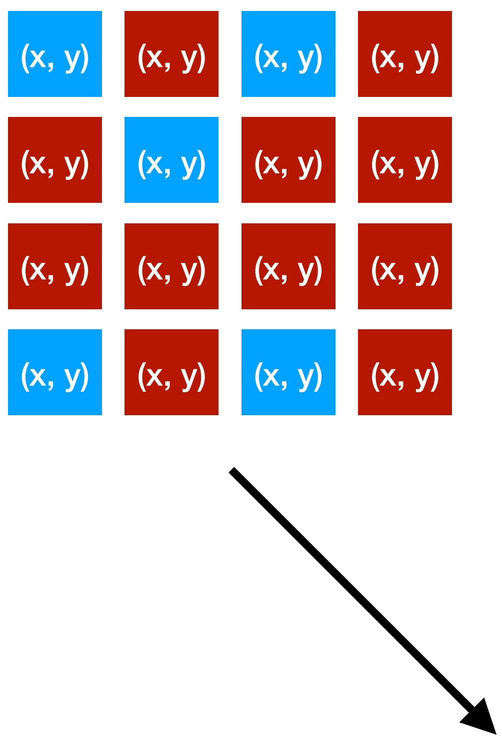
[Ilyas P Engstrom Leclerc Madry ’22]

$\{(S_1, f_1), \qquad\qquad\qquad\qquad \}$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry ’22]

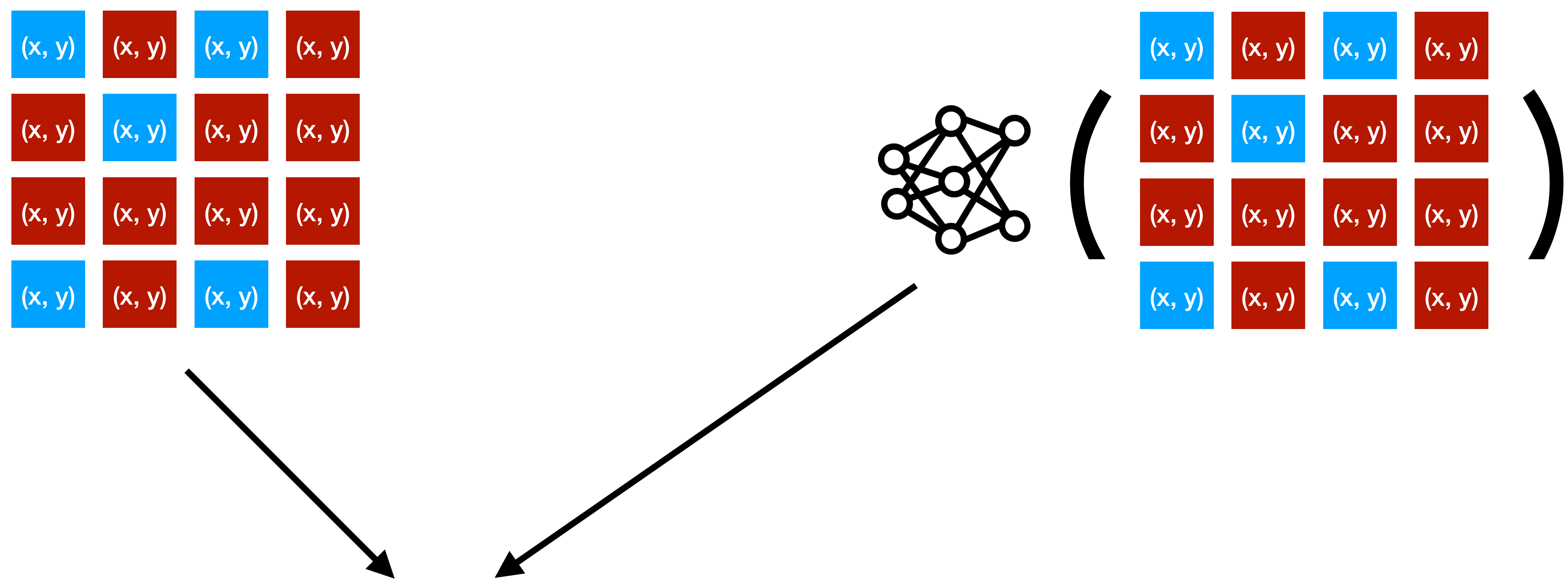


$$\{(S_1, f_1), \qquad\qquad\qquad\qquad \}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry ’22]

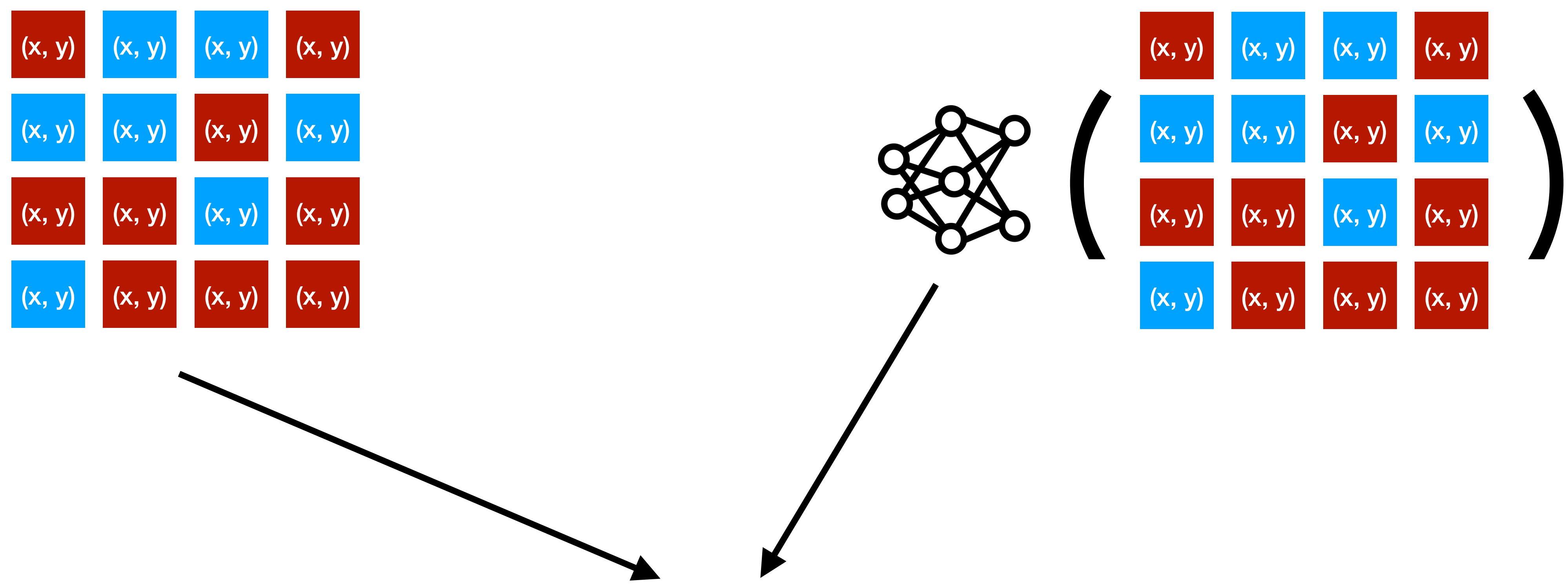


$$\{(S_1, f_1), \qquad\qquad\qquad \}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

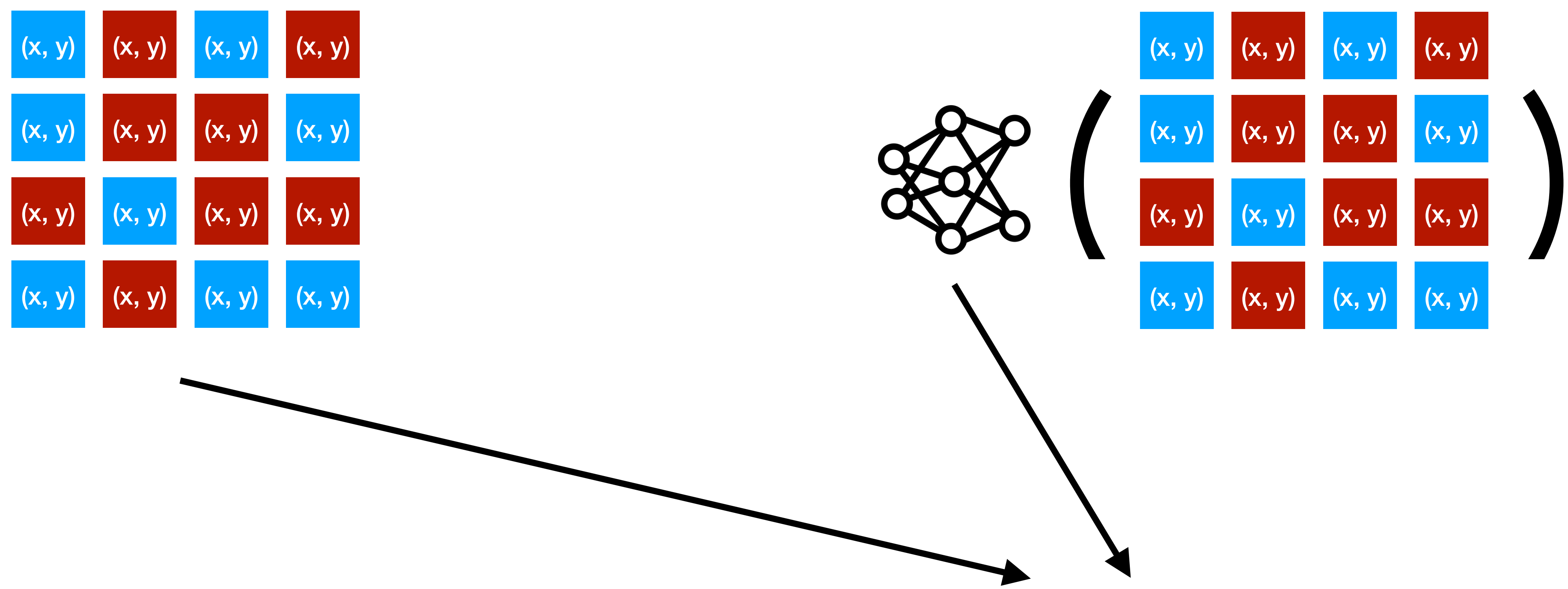


$$\{(S_1, f_1), (S_2, f_2), \qquad \}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]



$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry ’22]

(for a **specific** target example x)

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

(for a **specific** target example x)

$$\tau(x) = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \left(\beta^\top \mathbf{1}_{S_i} - f(x; S_i) \right)^2 + \lambda \|\beta\|_1$$

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

(for a **specific** target example x)

Linear model prediction

$$\tau(x) = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \left(\beta^\top \mathbf{1}_{S_i} - f(x; S_i) \right)^2 + \lambda \|\beta\|_1$$

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

(for a **specific** target example x)

Linear model prediction

True (observed) output

$$\tau(x) = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \left(\beta^\top \mathbf{1}_{S_i} - f(x; S_i) \right)^2 + \lambda \|\beta\|_1$$

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Simple approach: datamodels

[Ilyas P Engstrom Leclerc Madry '22]

(for a **specific** target example x)

Linear model prediction

True (observed) output

$$\tau(x) = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \left(\beta^\top \mathbf{1}_{S_i} - f(x; S_i) \right)^2 + \lambda \|\beta\|_1$$

ℓ_1 regularization (for sparsity + generalization)

$$\{(S_1, f_1), (S_2, f_2), \ldots, (S_m, f_m)\}$$

Basic idea: Use supervised learning

Evaluating datamodels

ResNet-9's on CIFAR-10

Actual margin
$\mathbb{E}[f(x, S_i)]$

Predicted margin $\mathbf{1}_{S_i} \cdot \tau(x)$

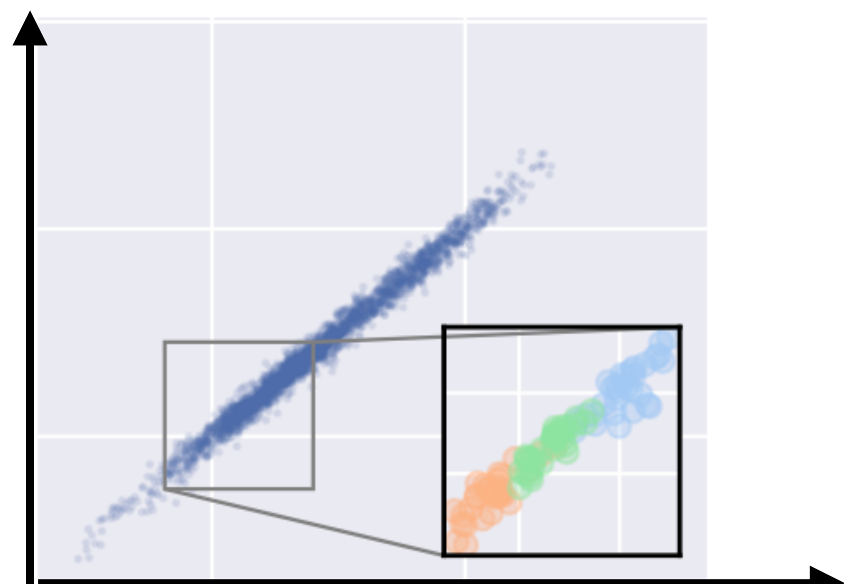
Evaluating datamodels

ResNet-9's on CIFAR-10

Evaluating datamodels

ResNet-9's on CIFAR-10

Actual margin $\mathbb{E}[f(x, S_i)]$

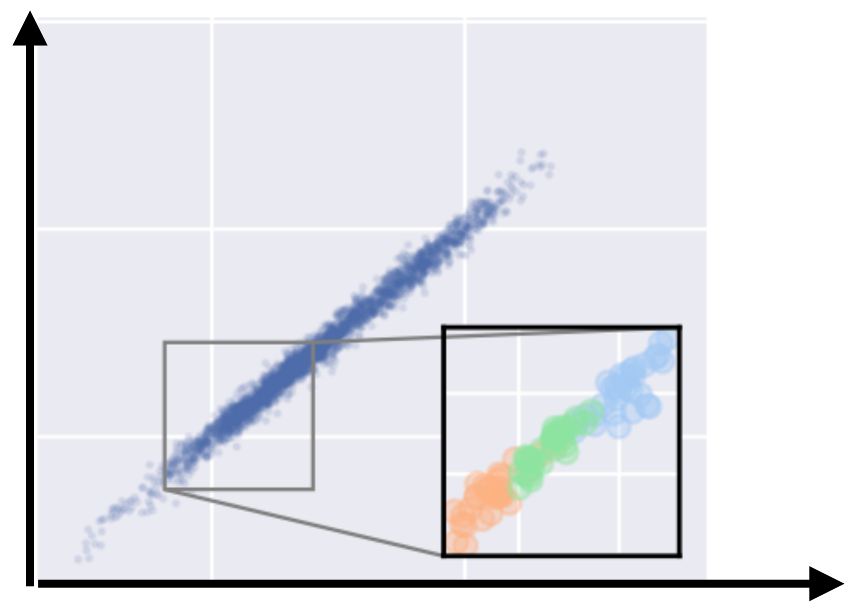


Predicted margin $\mathbf{1}_{S_i} \cdot \tau(x)$

Evaluating datamodels

ResNet-9's on CIFAR-10

Actual margin
$\mathbb{E}[f(x, S_i)]$



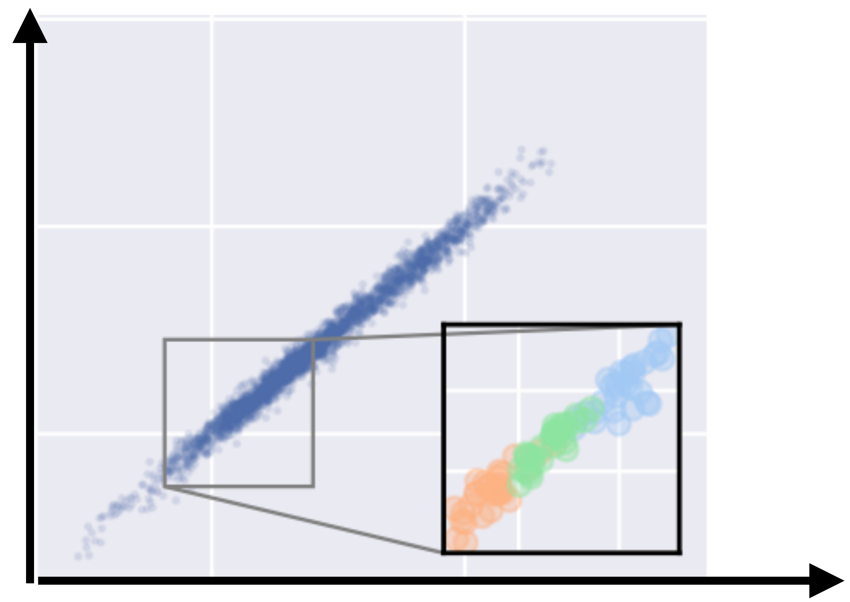
Predicted margin $\mathbf{1}_{S_i} \cdot \tau(x)$

Takeaway: We *can use* simple linear models
to predict final model outputs as functions of data

Evaluating datamodels

ResNet-9's on CIFAR-10

Actual margin $\mathbb{E}[f(x, S_i)]$



Predicted margin $\mathbf{1}_{S_i} \cdot \tau(x)$

Takeaway: We *can use* simple linear models to predict final model outputs as functions of data

Problem: Need to train 1000s of models! Often infeasible

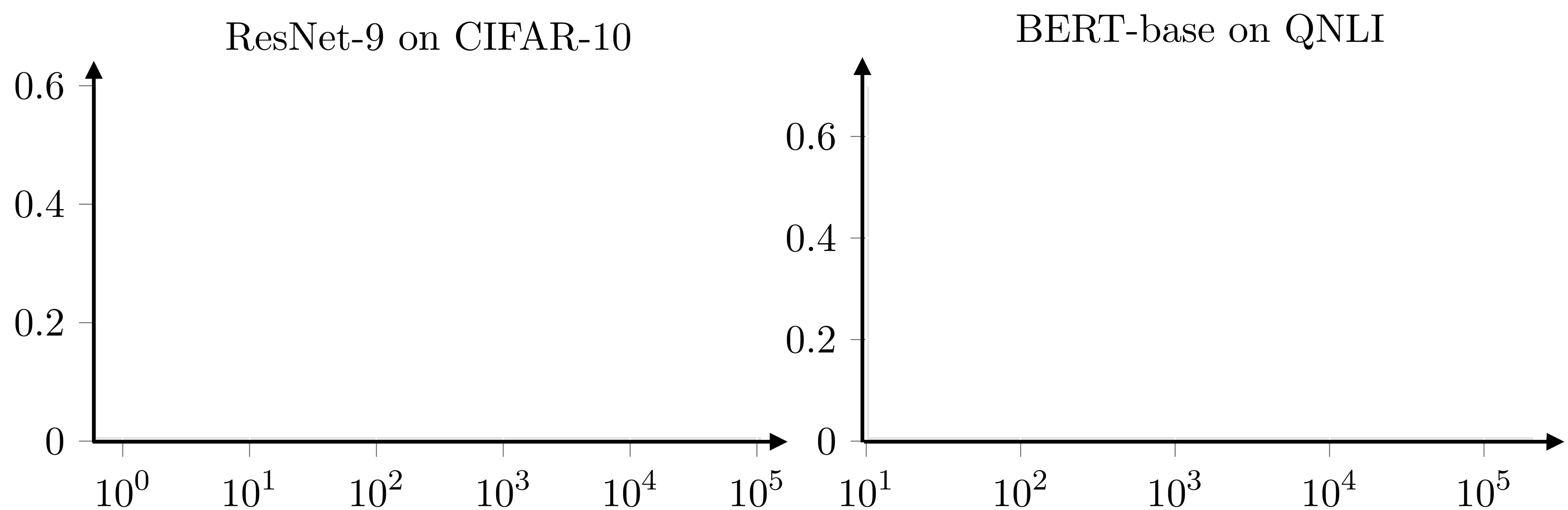
Efficacy vs Efficiency

Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**

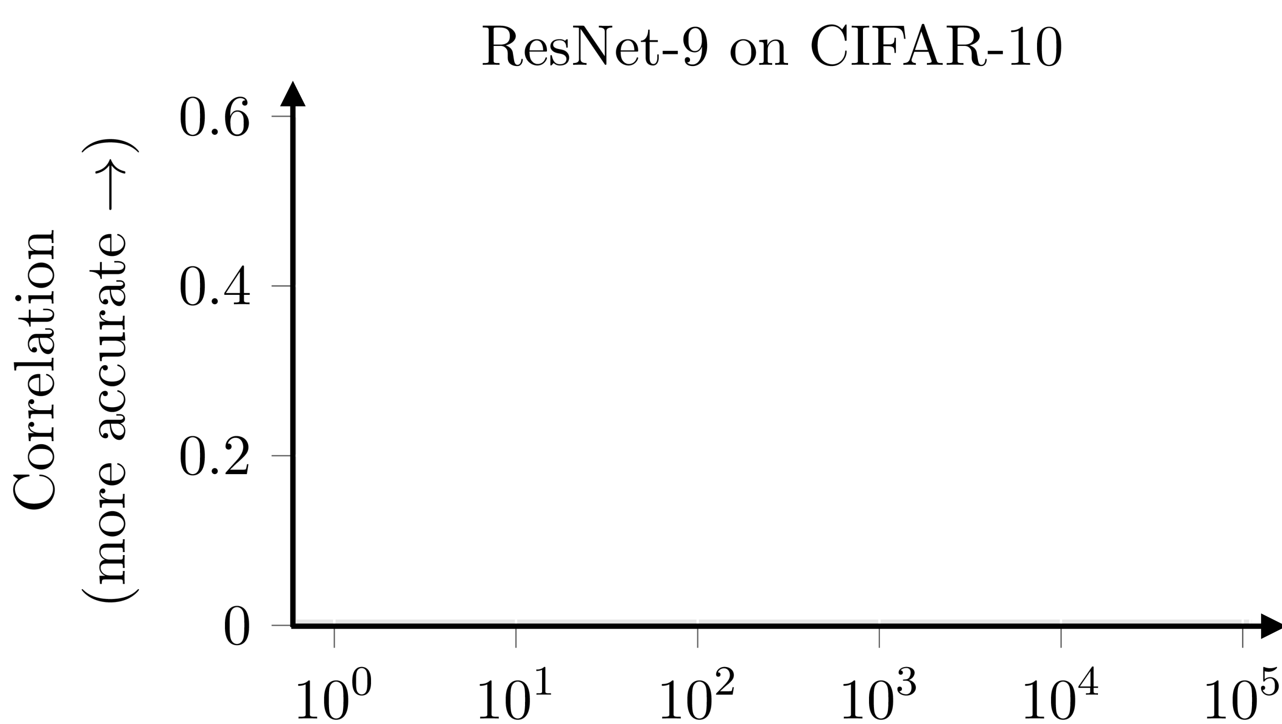
Efficacy vs Efficiency

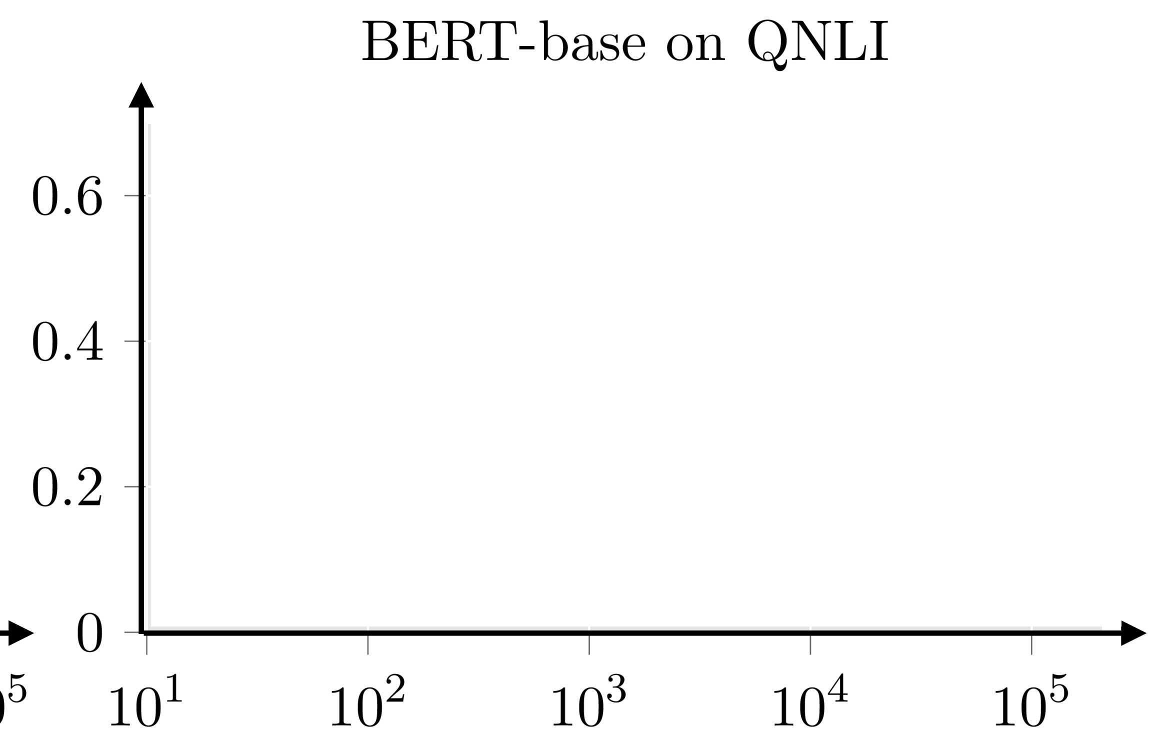
Data attribution should be both **effective** and **efficient**



Efficacy vs Efficiency

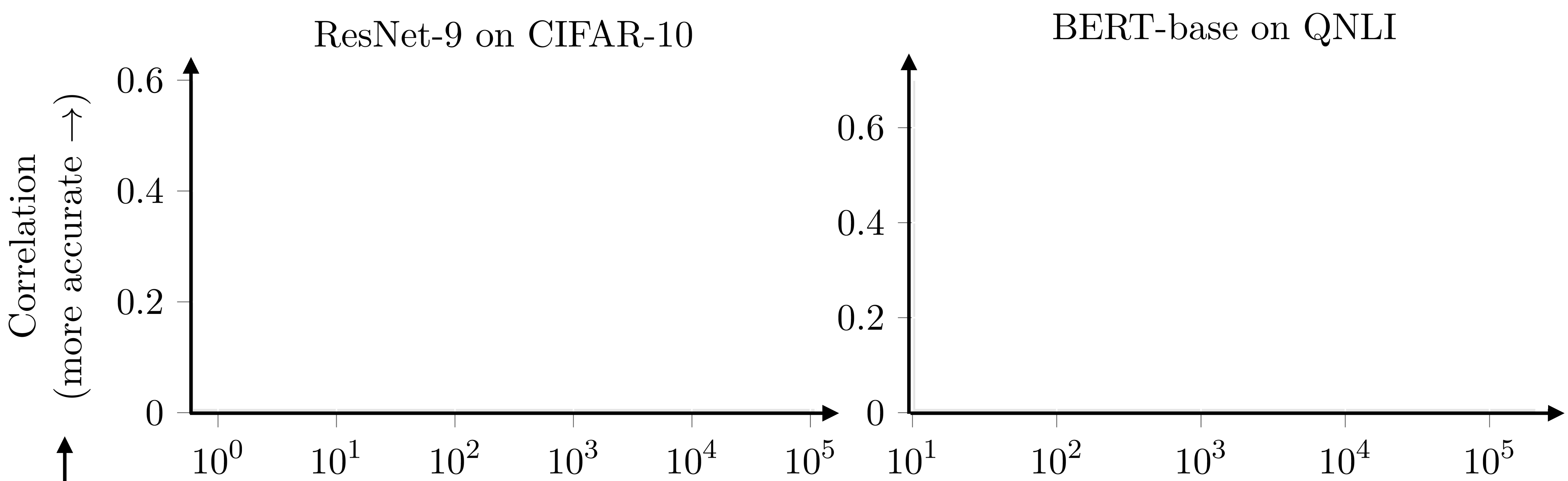
Data attribution should be both **effective** and **efficient**





Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**

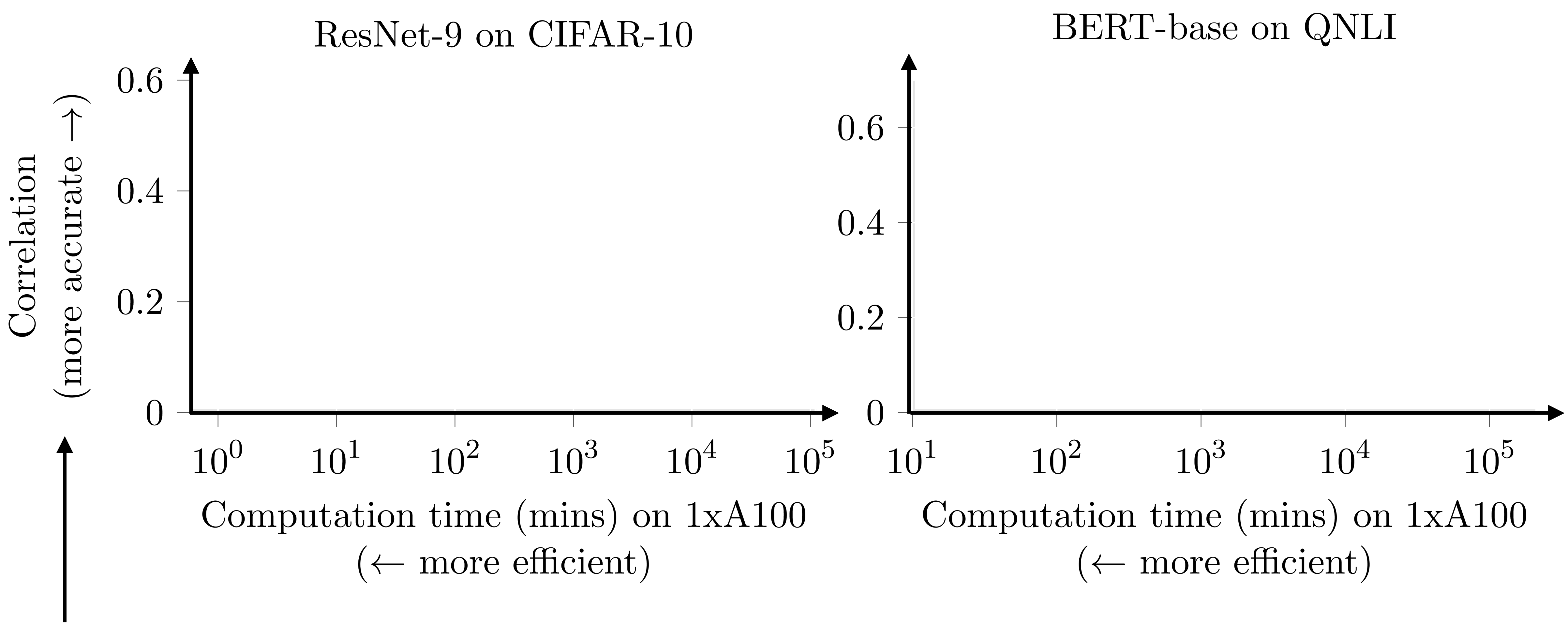


Linear Datamodeling Score (LDS)

Correlation between **true** model output $f(x, S')$ and **predicted** model output $\mathbf{1}_{S_i} \cdot \tau(x)$

Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**

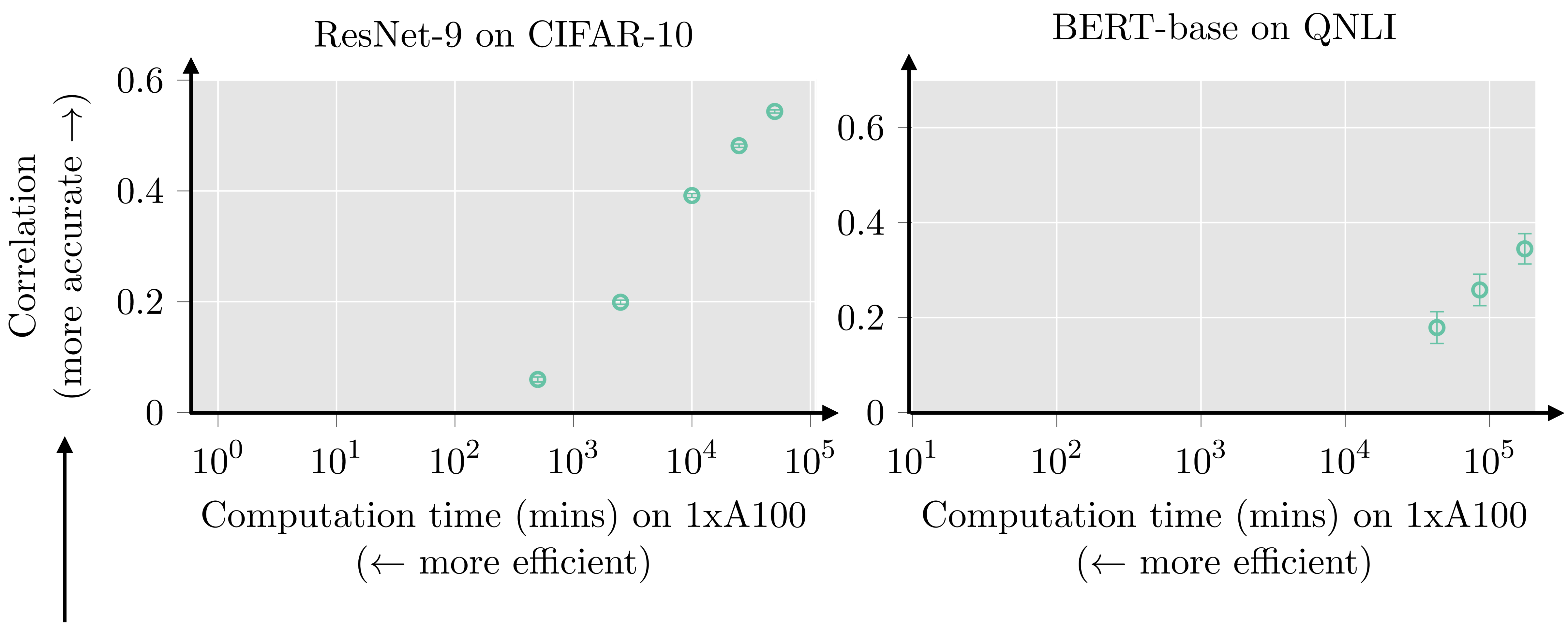


Linear Datamodeling Score (LDS)

Correlation between **true** model output $f(x, S')$ and **predicted** model output $\mathbf{1}_{S_i} \cdot \tau(x)$

Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**

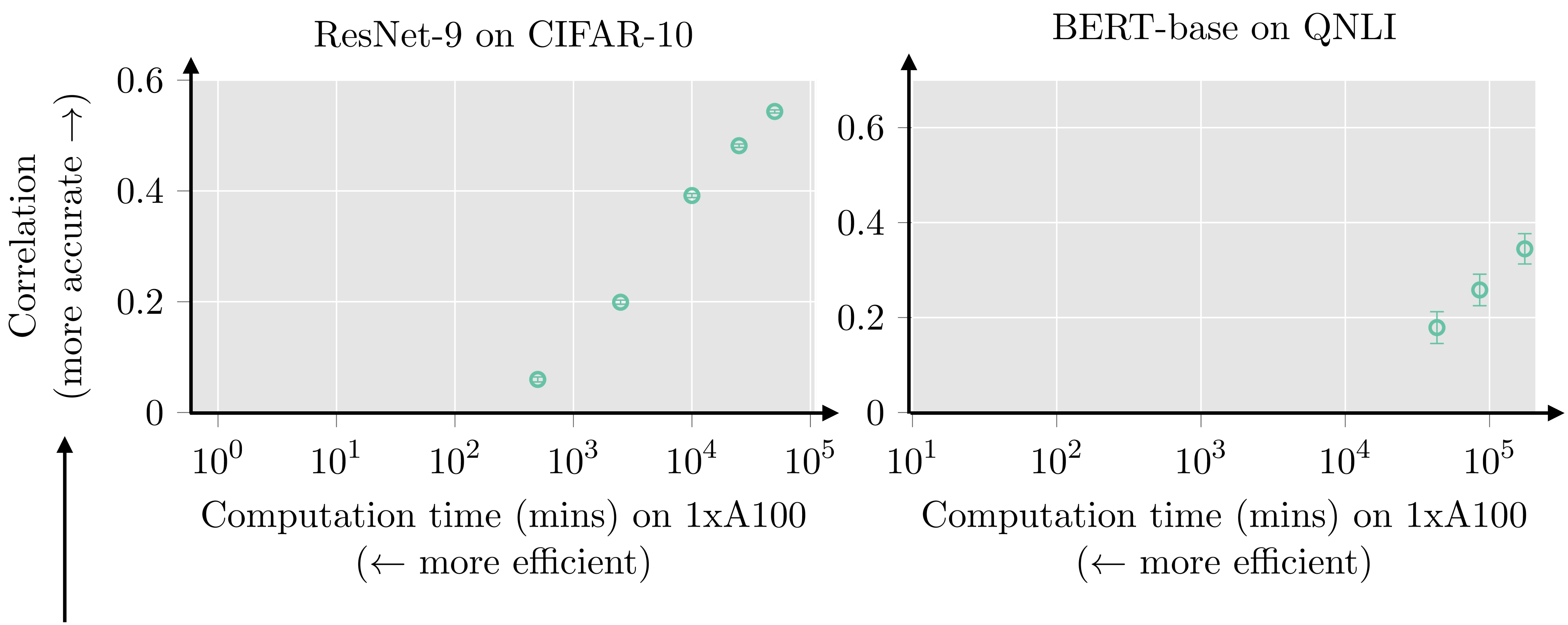


Linear Datamodeling Score (LDS)

Correlation between **true** model output $f(x, S')$ and **predicted** model output $\mathbf{1}_{S_i} \cdot \tau(x)$

Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**

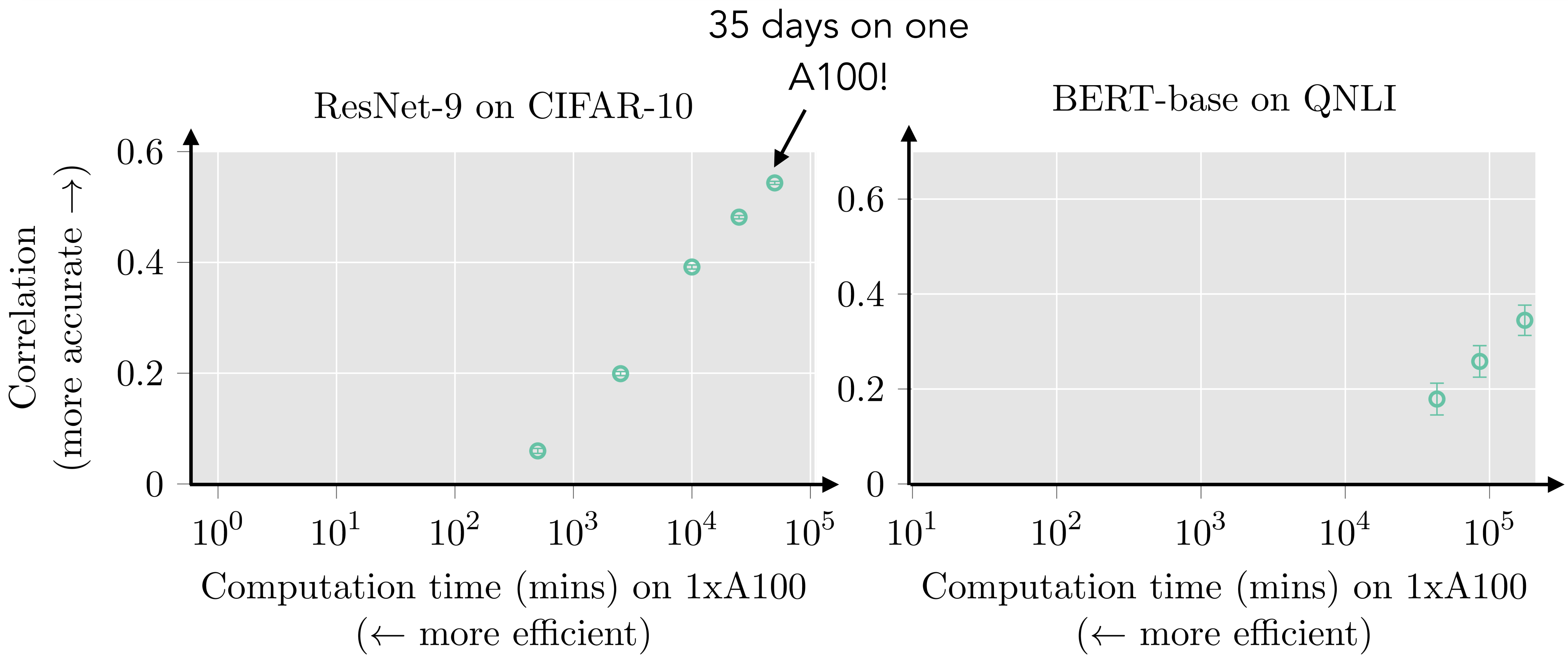


Linear Datamodeling Score (LDS)

Correlation between **true** model output $f(x, S')$ and **predicted** model output $\mathbf{1}_{S_i} \cdot \tau(x)$

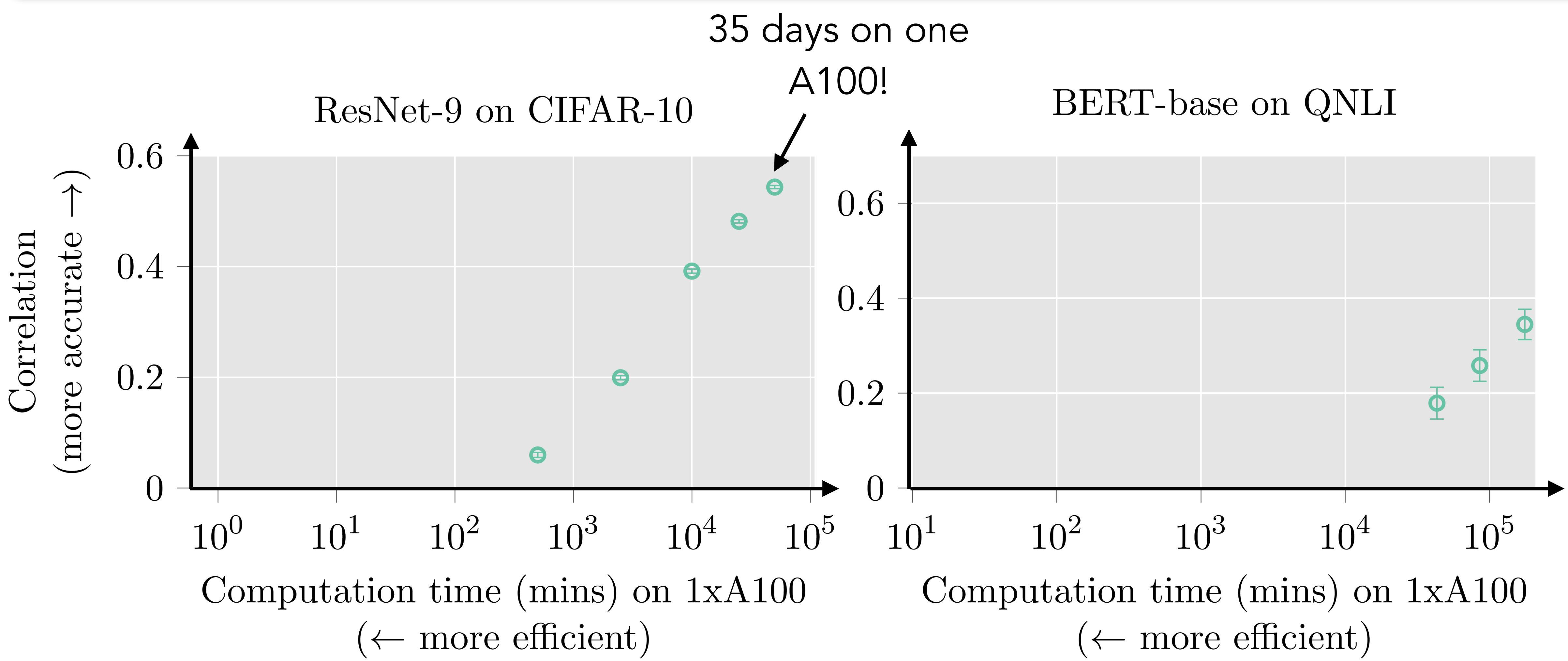
Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**



Efficacy vs Efficiency

Data attribution should be both **effective** and **efficient**



What else can we do?

Other attribution methods

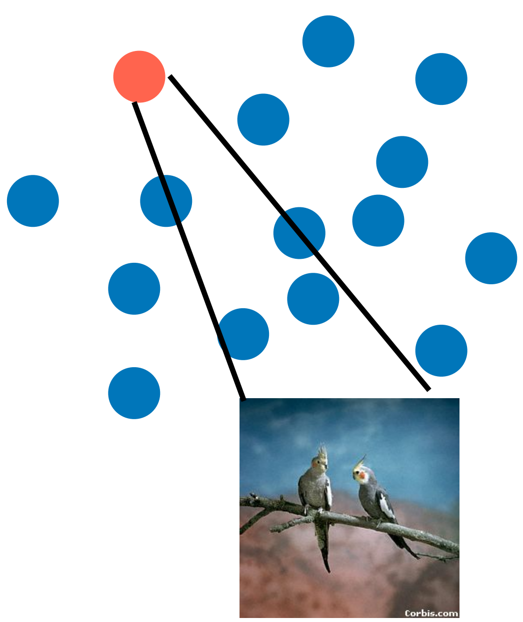
Other attribution methods

Recall: Attribution method is just a function $\tau : \mathcal{X} \to \mathbb{R}^{|S|}$

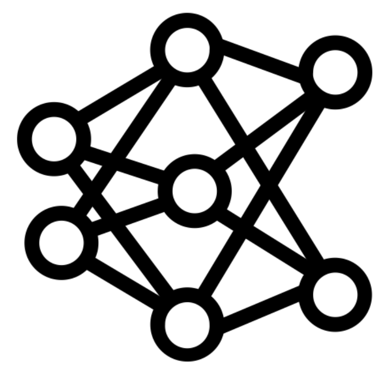
Other attribution methods

Recall: Attribution method is just a function $\tau : \mathscr{X} \to \mathbb{R}^{|S|}$

Training data S



Model f



Test input x

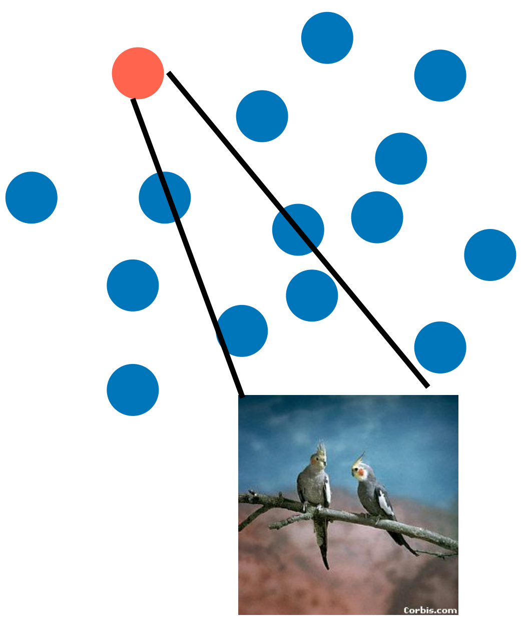
Bird (90%)

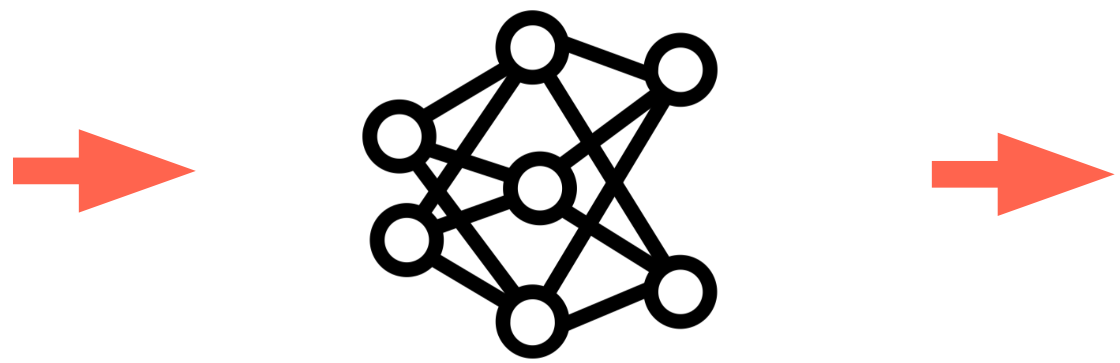
Model output $f(x; S)$

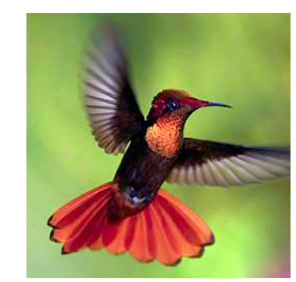
Other attribution methods

Recall: Attribution method is just a function $\tau : \mathscr{X} \to \mathbb{R}^{|S|}$

Training data S | Model f | Test input x







Bird (90%)

Model output $f(x; S)$

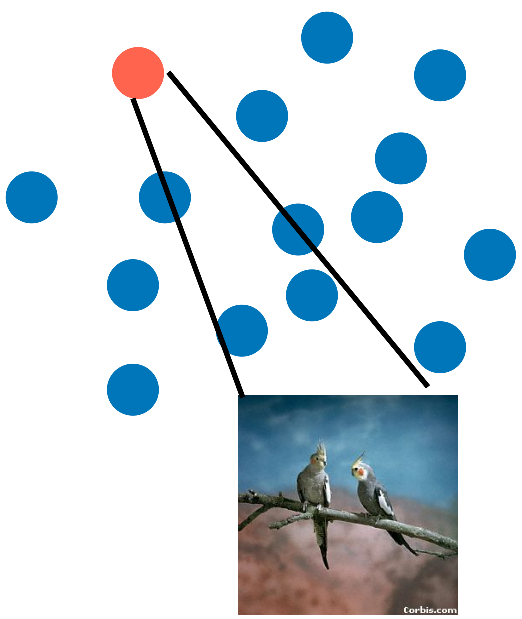
Ex: Influence functions, Shapley values, TracIn

[Ghorbani Zou '19, Jia et al. '19, Pruthi et al. '19, Feldman Zhang '20]

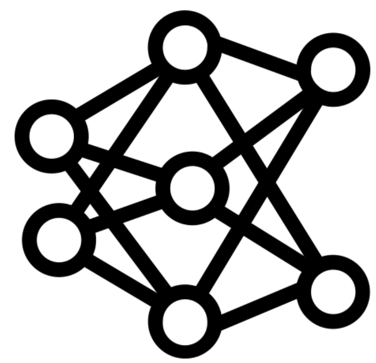
Other attribution methods

Recall: Attribution method is just a function $\tau : \mathscr{X} \rightarrow \mathbb{R}^{|S|}$

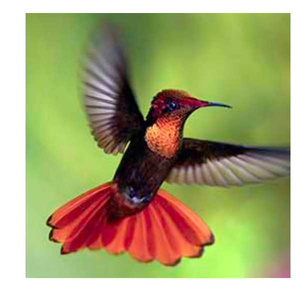
Training data S



Model f



Test input x



Bird (90%)

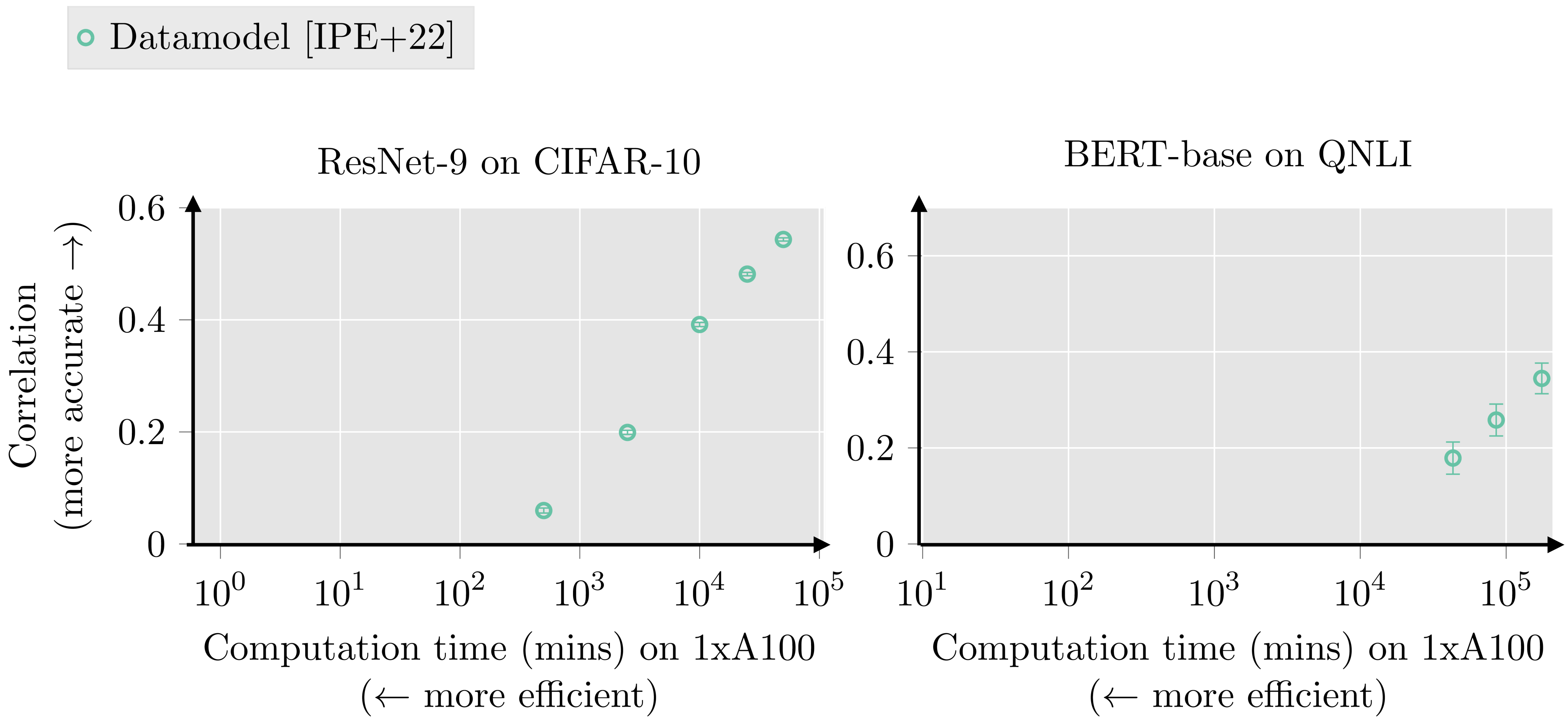
Model output $f(x; S)$

Ex: Influence functions, Shapley values, TracIn

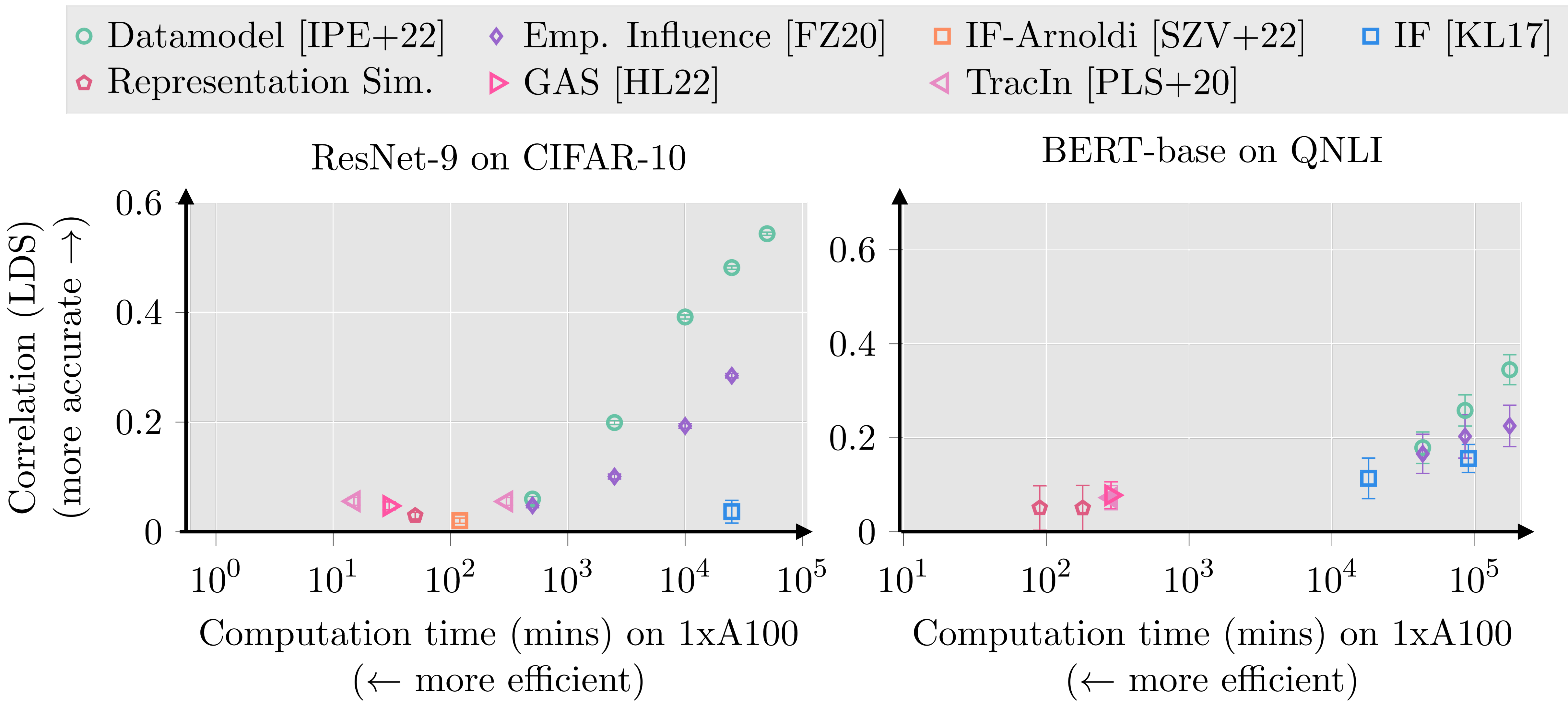
[Ghorbani Zou '19, Jia et al. '19, Pruthi et al. '19, Feldman Zhang '20]

Are these effective predictors of model output?

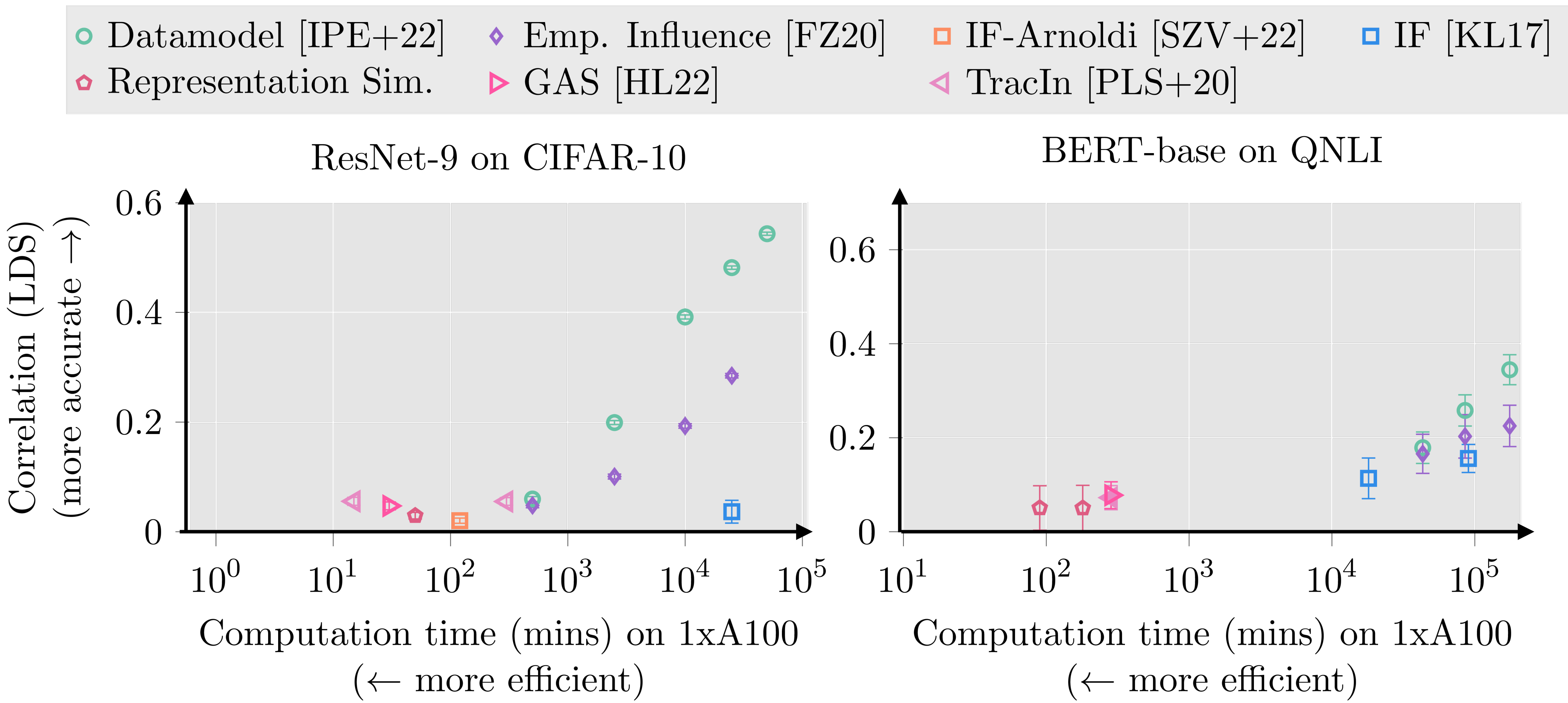
Evaluating attribution methods



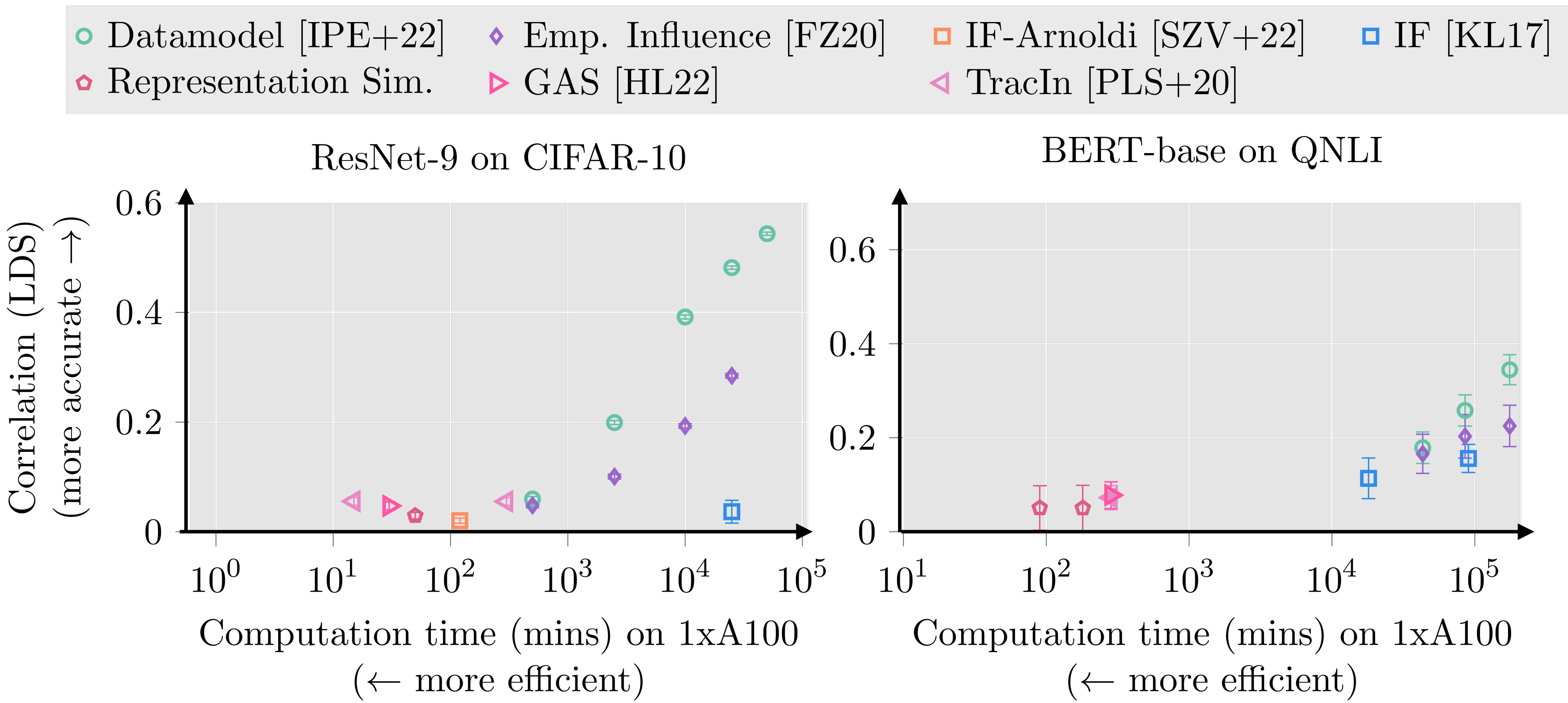
Evaluating attribution methods



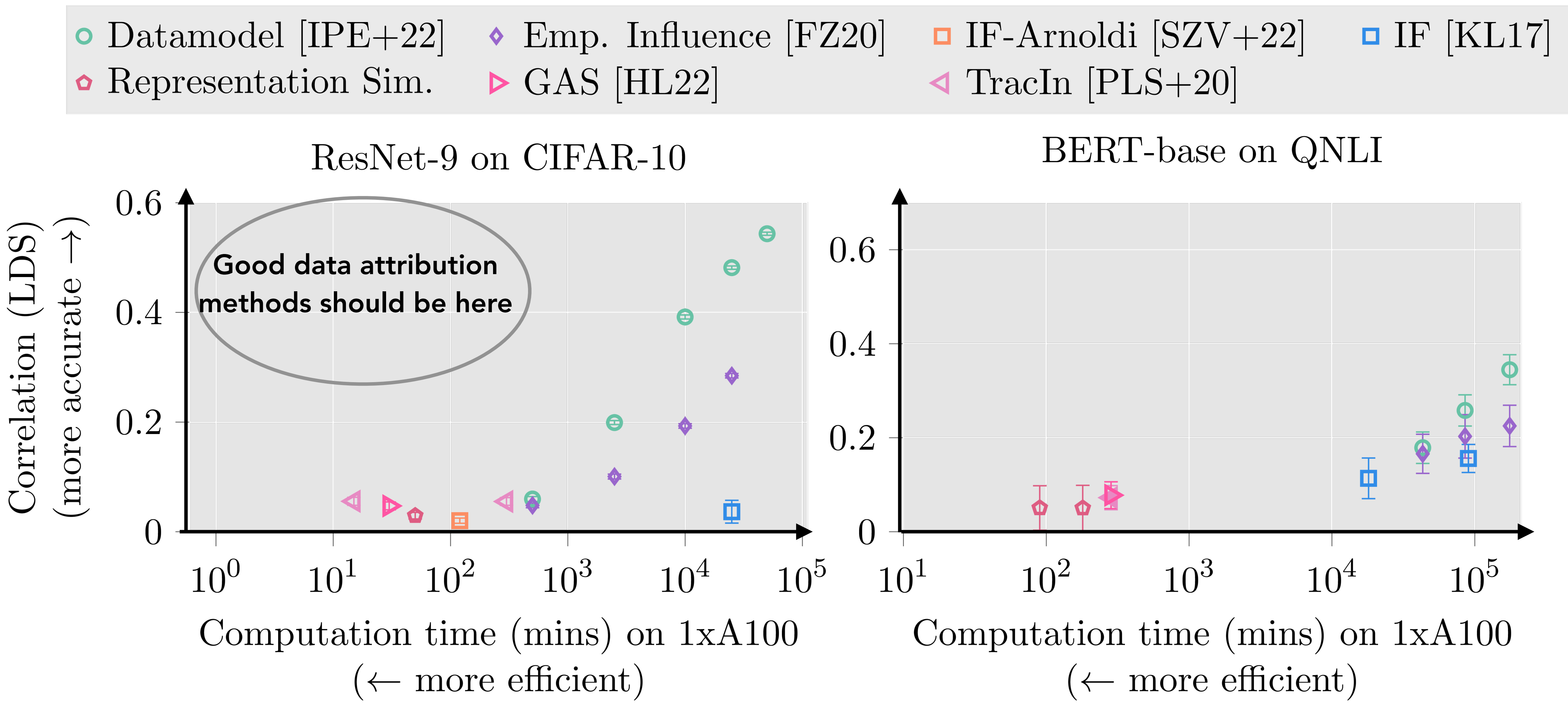
Evaluating attribution methods



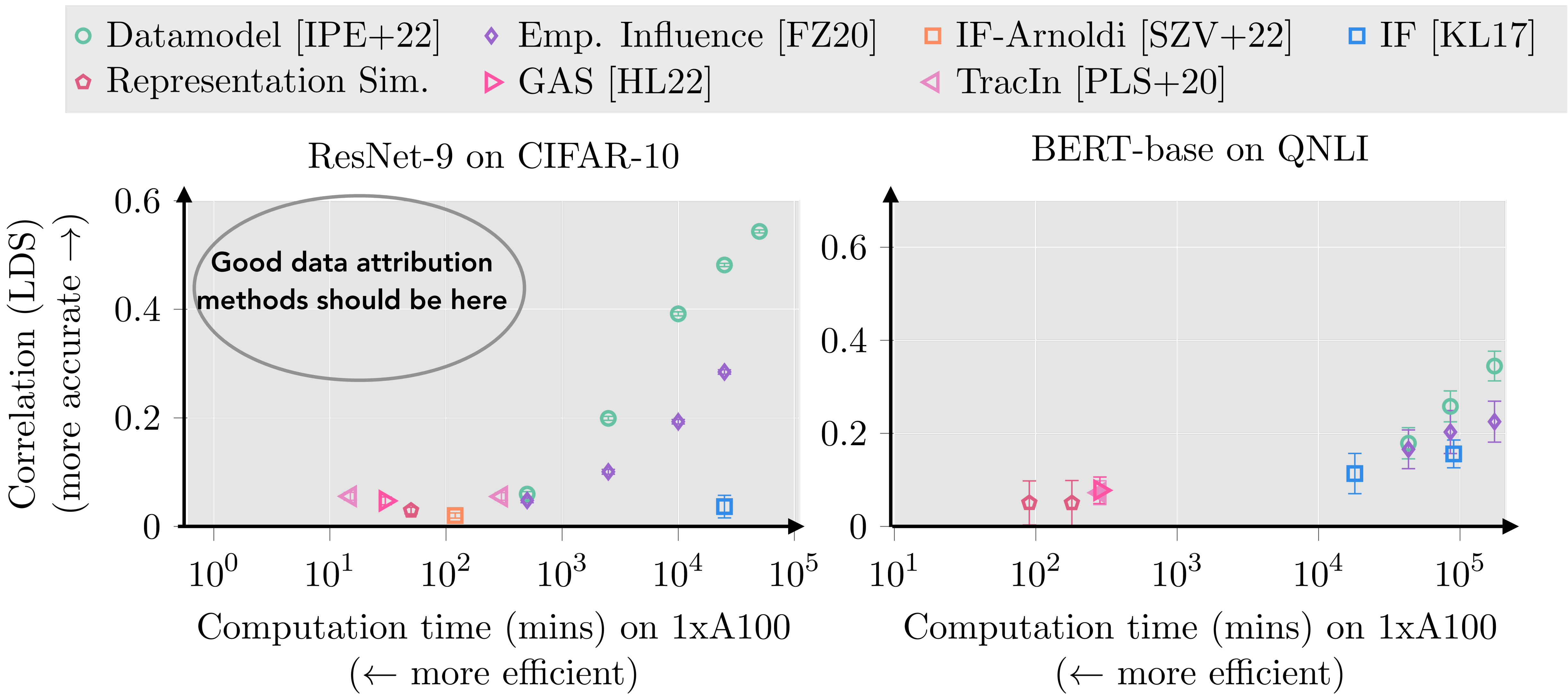
Evaluating attribution methods



Evaluating attribution methods



Evaluating attribution methods

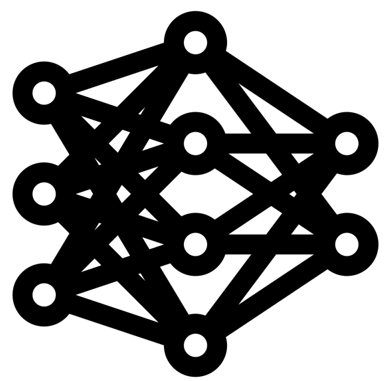


Can we design a method that is both **scalable** and **predictive** in large-scale settings?

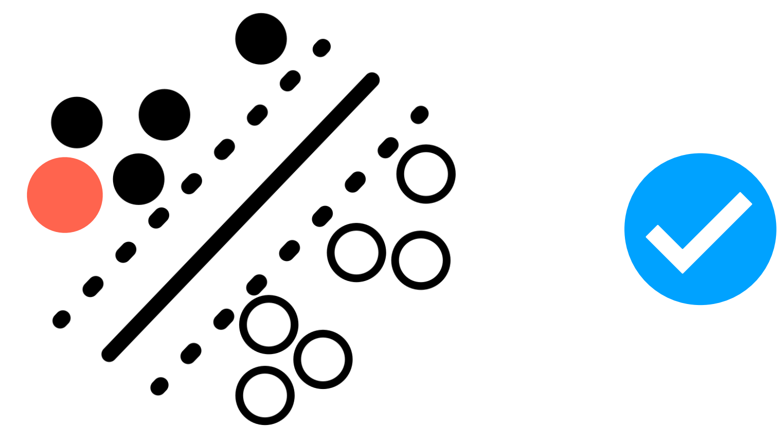
Our approach: **TRAK**

Our approach

Goal: Scalable and effective attribution for large-scale NNs



Arbitrary (differentiable) model



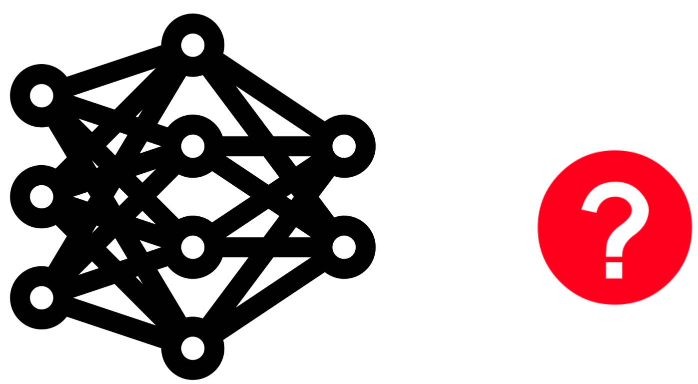
Generalized linear models

Yes! **Generalized linear models (GLM)**

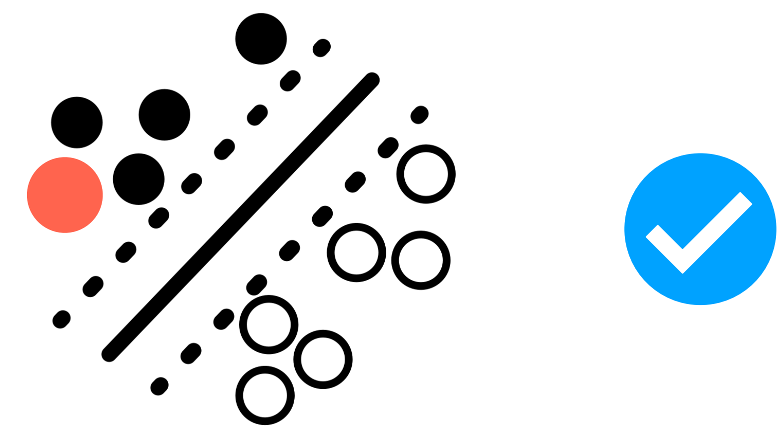
[Pregibon '81] [Wojnowicz et al. '16] [Koh Ang Teo Liang '19]

Our approach

Goal: Scalable and effective attribution for large-scale NNs



Arbitrary (differentiable) model



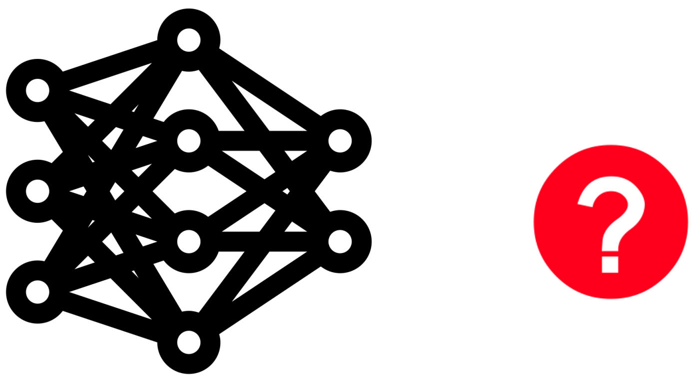
Generalized linear models

Yes! **Generalized linear models (GLM)**

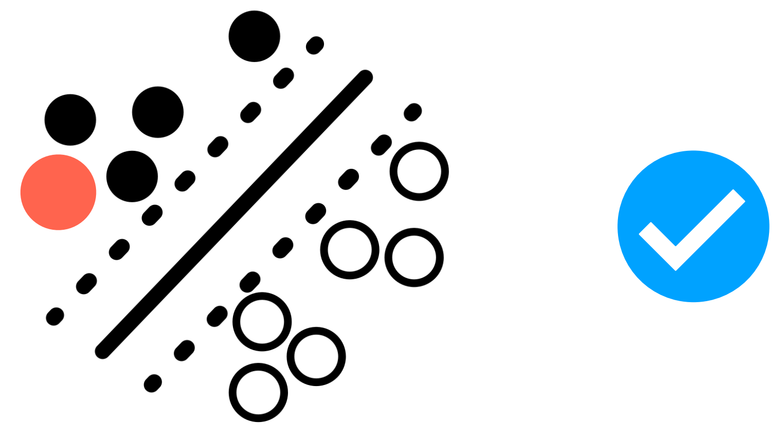
[Pregibon '81] [Wojnowicz et al. '16] [Koh Ang Teo Liang '19]

Our approach

Goal: Scalable and effective attribution for large-scale NNs



Arbitrary (differentiable) model



Generalized linear models

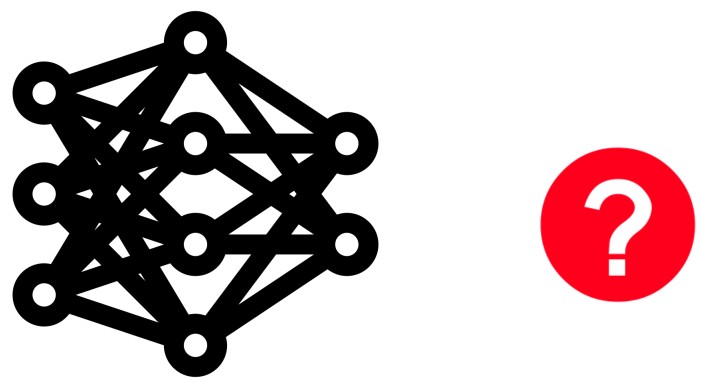
Q: Is there a simpler class of models that we can attribute well?

Yes! **Generalized linear models (GLM)**

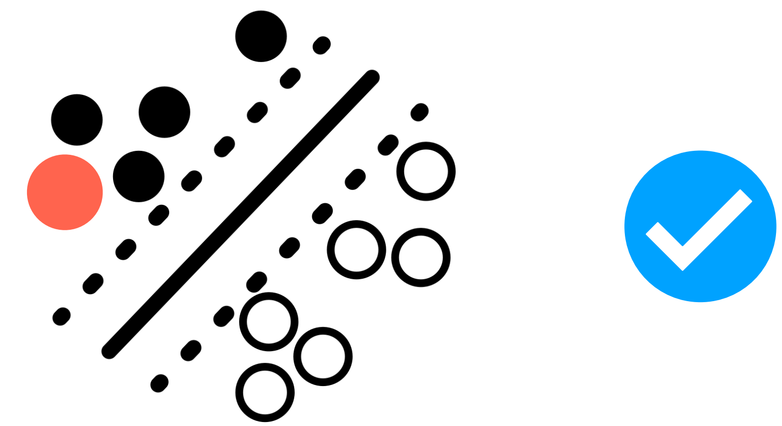
[Pregibon '81] [Wojnowicz et al. '16] [Koh Ang Teo Liang '19]

Our approach

Goal: Scalable and effective attribution for large-scale NNs



Arbitrary (differentiable) model



Generalized linear models

Q: Is there a simpler class of models that we can attribute well?

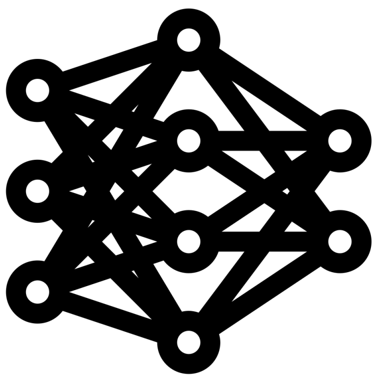
Yes! **Generalized linear models (GLM)**

[Pregibon '81] [Wojnowicz et al. '16] [Koh Ang Teo Liang '19]

Key idea: Reduce complex models → GLM, then apply known methods

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



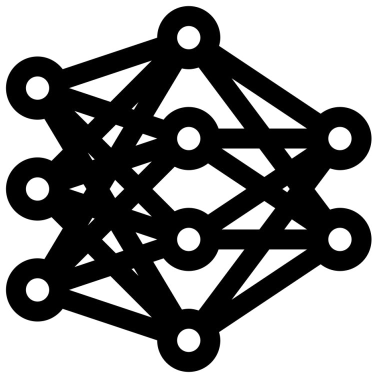
Original neural network

Inputs: example x

Output: $f(x; S)$

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

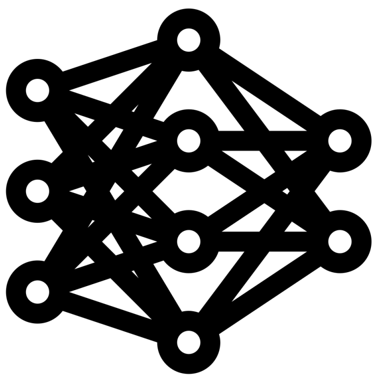
Inputs: example x

Output: $f(x; \theta)$

Note: θ **is a function of S**

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



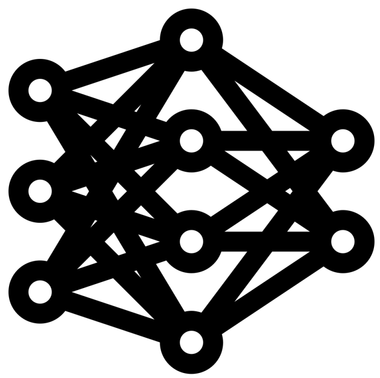
Original neural network

Inputs: example x

Output: $f(x; \theta)$

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

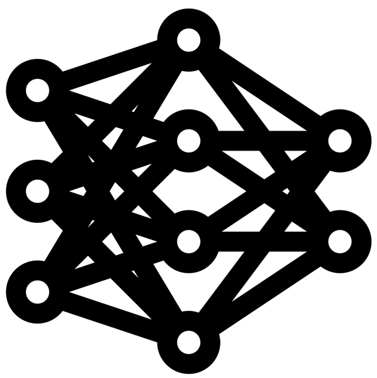
Inputs: example x

Output: $f(x; \theta)$

Can be arbitrarily complicated

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

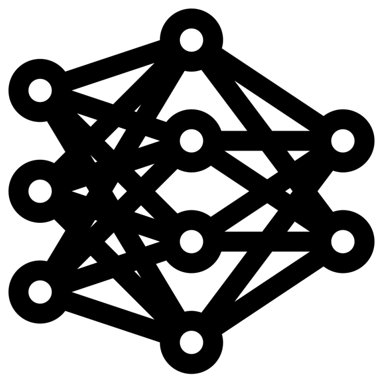
Output: $f(x; \theta)$

Can be arbitrarily complicated

Our approach: Taylor approximation

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

Output: $f(x; \theta)$

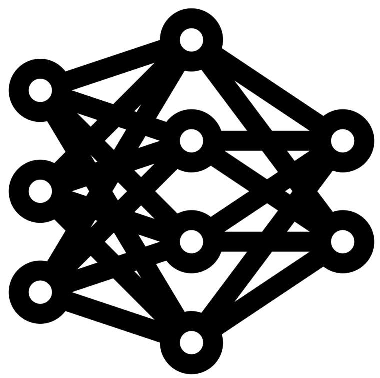
Can be arbitrarily complicated

Our approach: Taylor approximation

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

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

Output: $f(x;\theta)$

Can be arbitrarily complicated

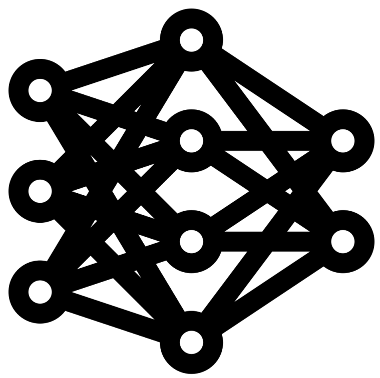
Our approach: Taylor approximation

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

Final parameters (constant wrt θ)

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

Output: $f(x; \theta)$

Can be arbitrarily complicated

Our approach: Taylor approximation

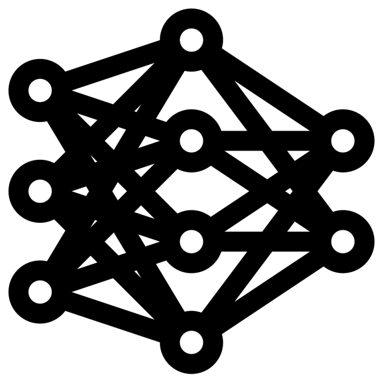
$$f(x, \theta) \approx f(x; \theta^\star) + \nabla_\theta f(x; \theta^\star) \cdot (\theta - \theta^\star)$$

Final parameters (constant wrt θ)

This is a <u>linear</u> function in the parameter θ

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel

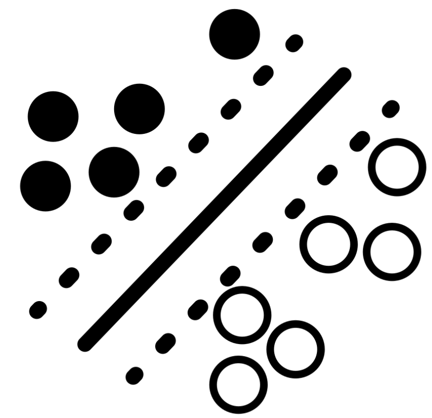


Original neural network

Inputs: example x

Output: $f(x; \theta)$

Can be arbitrarily complicated



Corresponding Linear model

Inputs:

$\nabla_\theta f(x; \theta^\star)$

Output:

$\nabla_\theta f(x; \theta^\star)^\top \theta$

Our approach: Taylor approximation

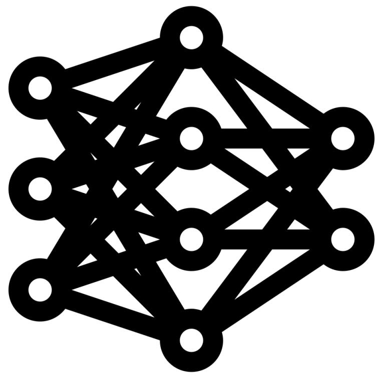
$$f(x, \theta) \approx f(x; \theta^\star) + \nabla_\theta f(x; \theta^\star) \cdot (\theta - \theta^\star)$$

Final parameters (constant wrt θ)

This is a <u>linear</u> function in the parameter θ

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel

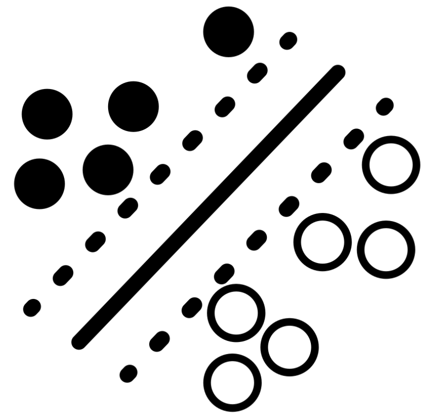


Original neural network

Inputs: example x

Output: $f(x; \theta)$

Can be arbitrarily complicated



Corresponding Linear model

Inputs: $\nabla_\theta f(x; \theta^\star)$

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

Our approach: Taylor approximation

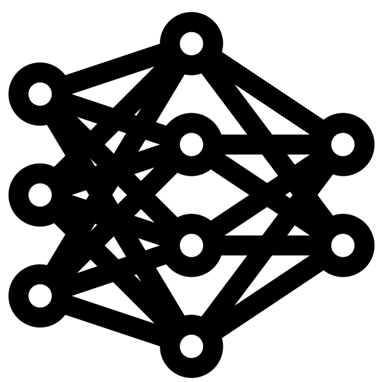
$$f(x, \theta) \approx f(x; \theta^\star) + \nabla_\theta f(x; \theta^\star) \cdot (\theta - \theta^\star)$$

Note: This approximation is related to the empirical Neural Tangent Kernel

[Jacot et al. '18] [Long '21] [Wei Hu Steinhardt '22]

TRAK: Step 1

Tracing with the **R**andomly-projected **A**fter **K**ernel

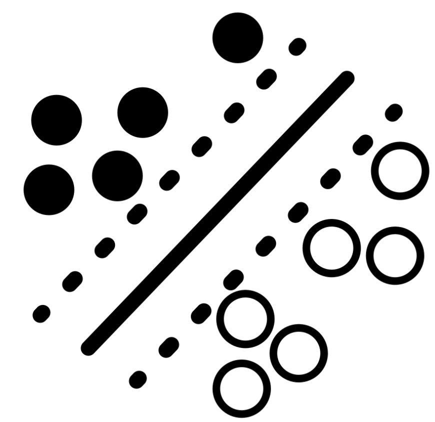


Original neural network

Inputs: example x

Output: $f(x;\theta)$

Can be arbitrarily complicated



Corresponding Linear model

Inputs:

$\nabla_\theta f(x;\theta^\star)$

Output:

$\nabla_\theta f(x;\theta^\star)^\top \theta$

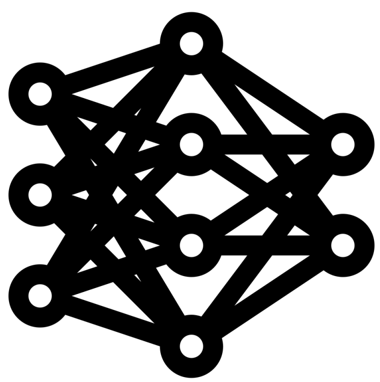
Our approach: Taylor approximation

$$f(x,\textcolor{red}{\theta}) \approx f(x;\theta^\star) + \nabla_\theta f(x;\theta^\star)\cdot(\textcolor{red}{\theta} - \theta^\star)$$

Implementation: Compute gradients $\nabla_\theta f(x;\theta^\star)$

TRAK: Step 2

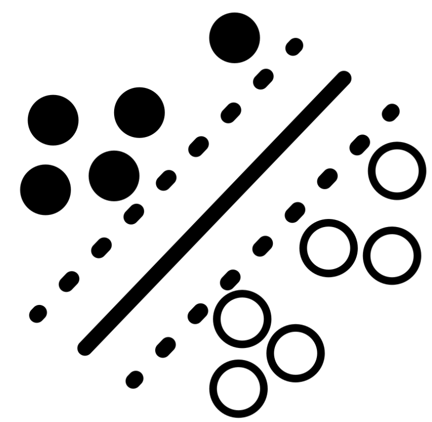
Tracing with the **R**andomly-projected **A**fter **K**ernel



Inputs: example x

Output: $f(x; \theta)$

Original neural network



Corresponding Linear model

Inputs:

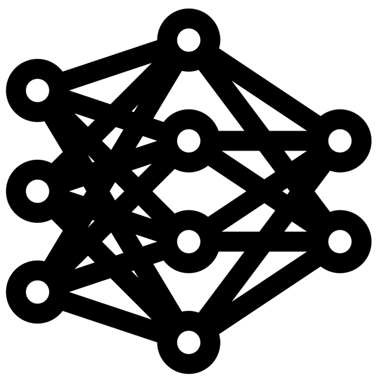
$\nabla_\theta f(x; \theta^\star)$

Output:

$\nabla_\theta f(x; \theta^\star)^\top \theta$

TRAK: Step 2

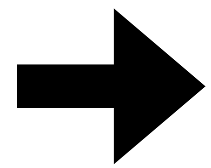
Tracing with the **R**andomly-projected **A**fter **K**ernel

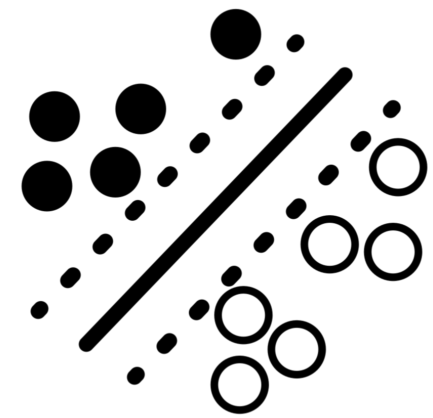


Original neural network

Inputs: example x

Output: $f(x; \theta)$





Corresponding Linear model

Inputs:

$\nabla_\theta f(x; \theta^\star)$

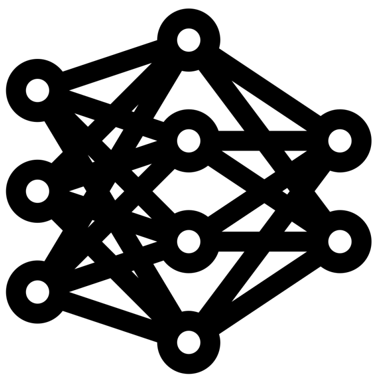
Output:

$\nabla_\theta f(x; \theta^\star)^\top \theta$

Problem: Features are high-dimensional (~millions for modern NNs)

TRAK: Step 2

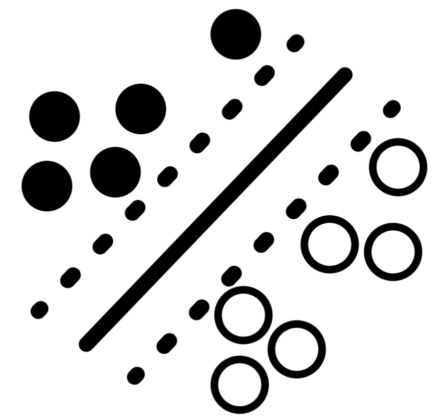
Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

Output: $f(x; \theta)$



Corresponding Linear model

Inputs: $\nabla_\theta f(x; \theta^\star)$

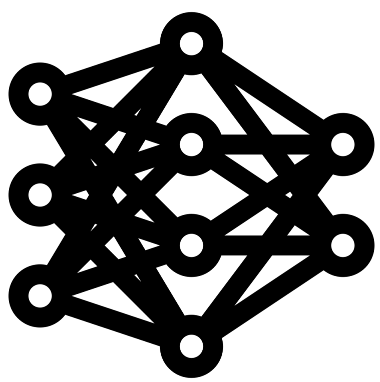
Output: $\nabla_\theta f(x; \theta^\star)^\top \theta$

Problem: Features are high-dimensional (~millions for modern NNs)

Solution: Project to $k \ll p$ dimensions using **random projections**

TRAK: Step 2

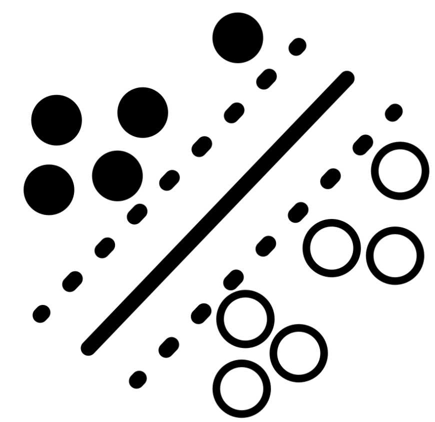
Tracing with the **R**andomly-projected **A**fter **K**ernel



Original neural network

Inputs: example x

Output: $f(x;\theta)$



Corresponding Linear model

Inputs: $\nabla_\theta f(x;\theta^\star)$

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

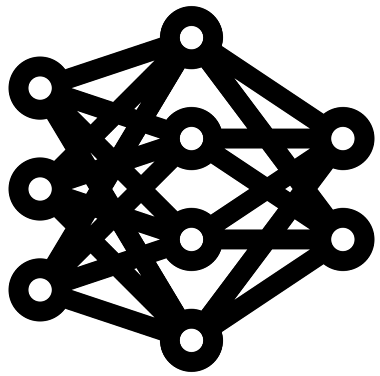
Problem: Features are high-dimensional (~millions for modern NNs)

Solution: Project to $k \ll p$ dimensions using **random projections**

$$\mathbf{P}^\top \nabla_\theta f(z;\theta^\star) \qquad \mathbf{P} \in \mathbb{R}^{p\times k}, \mathbf{P}_{ij} \sim N(0,1)$$

TRAK: Step 2

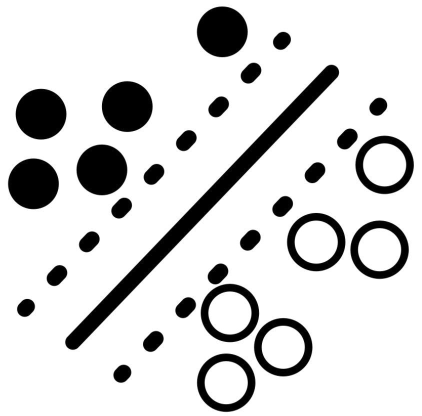
Tracing with the **R**andomly-projected **A**fter **K**ernel



Inputs: example x

Output: $f(x;\theta)$

Original neural network



Corresponding Linear model

Inputs:

$\mathbf{P}^\top \nabla_\theta f(x;\theta^\star)$

Output:

$(\mathbf{P}^\top \nabla_\theta f(x;\theta^\star))^\top \theta$

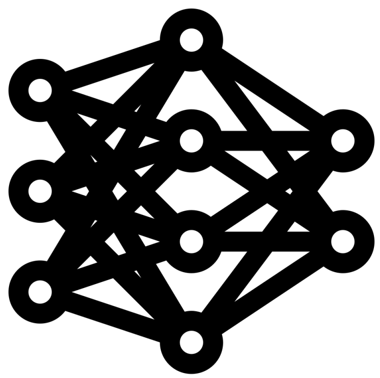
Problem: Features are high-dimensional (~millions for modern NNs)

Solution: Project to $k \ll p$ dimensions using **random projections**

$$\mathbf{P}^\top \nabla_\theta f(z;\theta^\star) \qquad \mathbf{P} \in \mathbb{R}^{p\times k}, \mathbf{P}_{ij} \sim N(0,1)$$

TRAK: Step 2

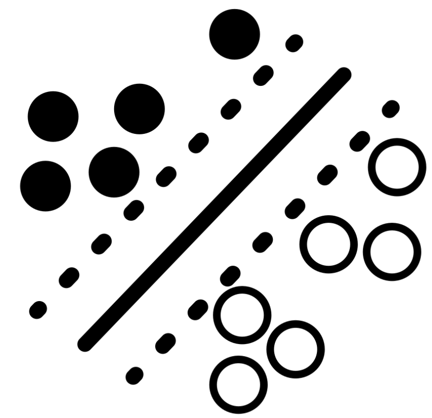
Tracing with the **R**andomly-projected **A**fter **K**ernel



Inputs: example x

Output: $f(x; \theta)$

Original neural network



Corresponding Linear model

Inputs: $\mathbf{P}^\top \nabla_\theta f(x; \theta^\star)$

Output: $(\mathbf{P}^\top \nabla_\theta f(x; \theta^\star))^\top \theta$

Problem: Features are high-dimensional (~millions for modern NNs)

Solution: Project to $k \ll p$ dimensions using **random projections**

$$\mathbf{P}^\top \nabla_\theta f(z; \theta^\star) \qquad \mathbf{P} \in \mathbb{R}^{p \times k}, \mathbf{P}_{ij} \sim N(0,1)$$

Why? Preserves inner products between input features

[Johnson Lindenstrauss '64]

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

$$\tau(x)_i$$

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

$$\tau(x)_i$$

Attribution score of i-th training example on output at x

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Attribution score of i-th training example on output at x

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

feature of target example

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Attribution score of i-th training example on output at x

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

feature of target example

feature of training example

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Attribution score of i-th training example on output at x

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

feature of target example

feature of training example

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Attribution score of i-th training example on output at x

Model confidence in correct class on training example x_i

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Next: apply attribution formula for logistic regression

One-step Newton approximation for logistic regression

[Pregibon '81]

feature of target example

feature of training example

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Attribution score of i-th training example on output at x

Model confidence in correct class on training example x_i

This give accurate attribution for linear models

[Wojnowicz et al. '16] [Koh Ang Teo Liang '19]

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Applying this to our setting:

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Applying this to our setting:

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Features: $x_i \leftarrow \mathbf{P}^\top \nabla_\theta f(x_i; \theta^\star)$ (randomly-projected gradient)

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Applying this to our setting:

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Features: $x_i \leftarrow \mathbf{P}^\top \nabla_\theta f(x_i; \theta^\star)$ (randomly-projected gradient)

Confidence: p_i = model confidence on example x_i

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Applying this to our setting:

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Features: $x_i \leftarrow \mathbf{P}^\top \nabla_\theta f(x_i; \theta^\star)$ (randomly-projected gradient)

Confidence: p_i = model confidence on example x_i

Making these substitutions → TRAK! (for one model)

TRAK: Step 3

Tracing with the **R**andomly-projected **A**fter **K**ernel

Applying this to our setting:

$$\tau(x)_i \approx x^\top (X^\top X)^{-1} x_i \cdot (1 - p_i)$$

Features: $x_i \leftarrow \mathbf{P}^\top \nabla_\theta f(x_i; \theta^\star)$ (randomly-projected gradient)

Confidence: p_i = model confidence on example x_i

Making these substitutions → TRAK! (for one model)

Only need per-example gradients + some linear algebra

TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt ’21] [D’Amour et al. ’20]

TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt '21] [D'Amour et al. '20]

We want to attribute model **class**, not a single model

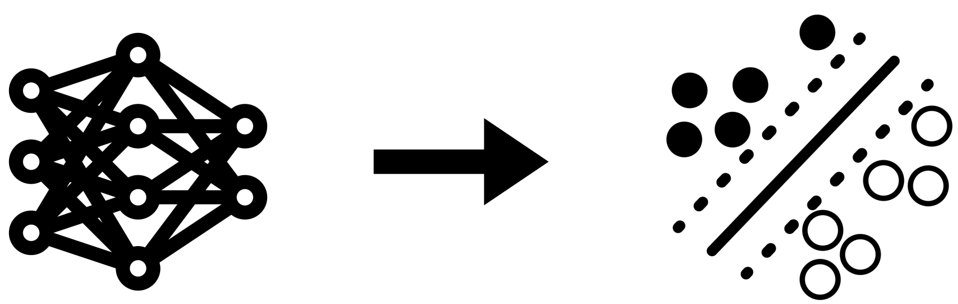
TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt '21] [D'Amour et al. '20]

We want to attribute model **class**, not a single model



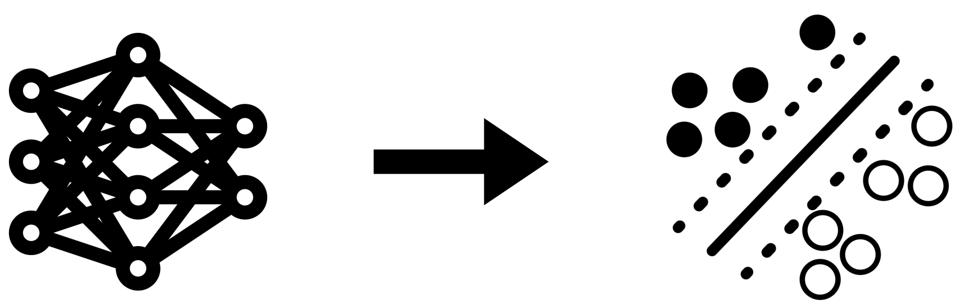
TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt '21] [D'Amour et al. '20]

We want to attribute model **class**, not a single model



Only gives local information about this **specific** model

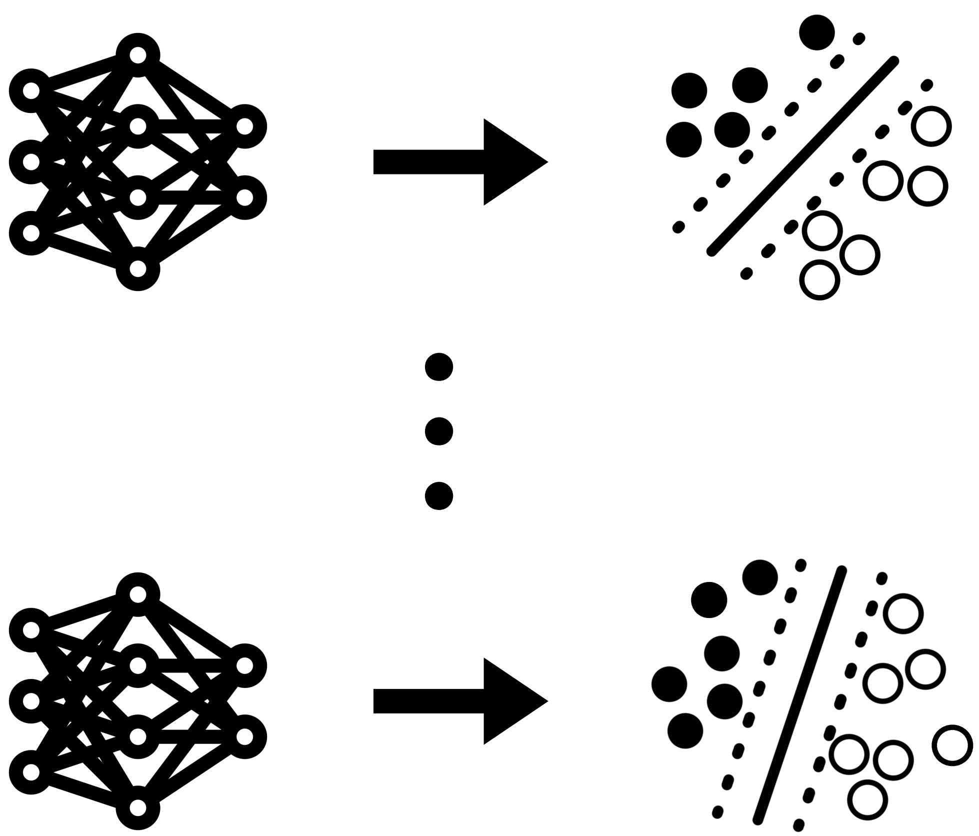
TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt '21] [D'Amour et al. '20]

We want to attribute model **class**, not a single model



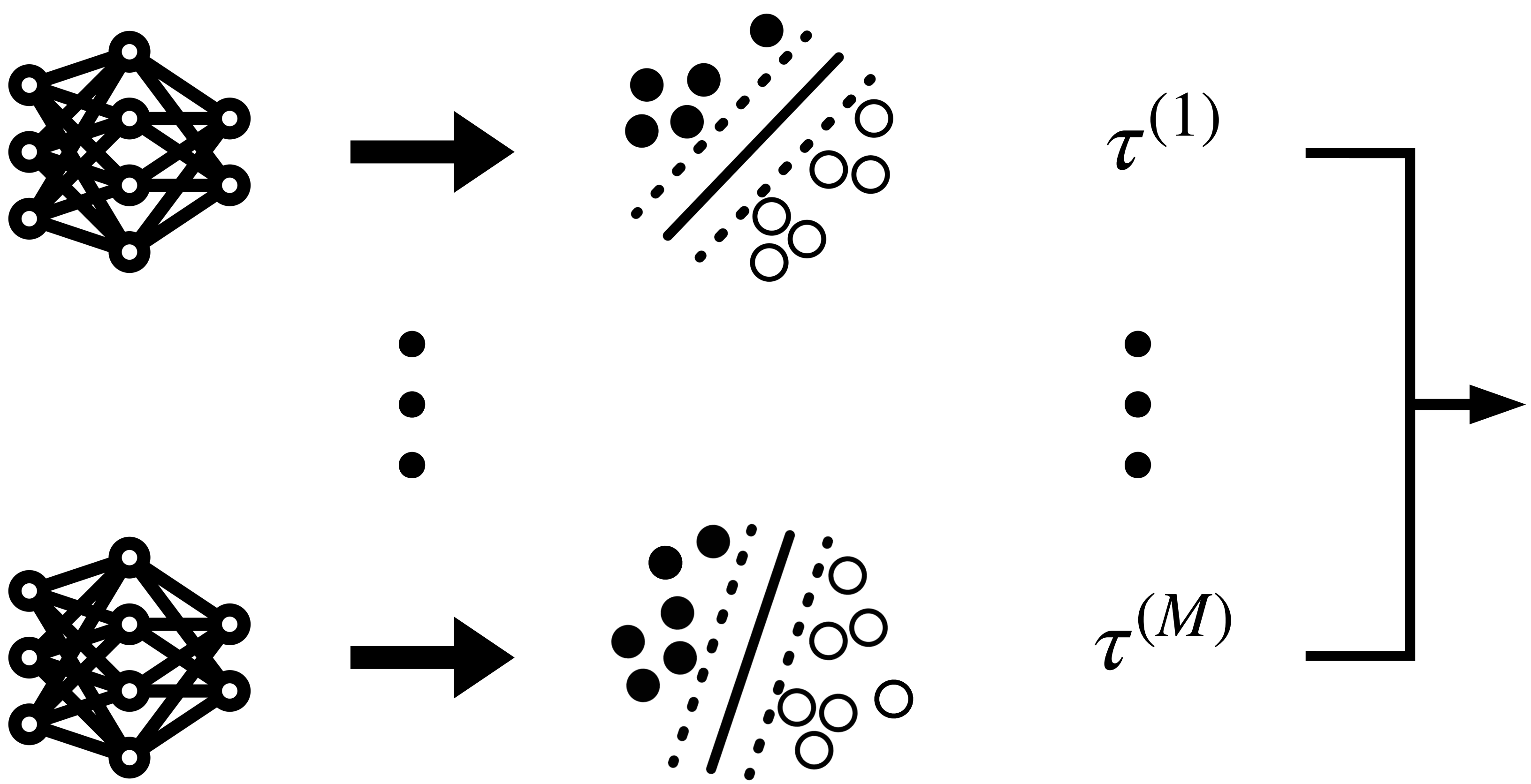
TRAK: Step 4

Tracing with the **R**andomly-projected **A**fter **K**ernel

Model training is **non-deterministic**, even for fixed training set

[Zhong Ghosh Klein Steinhardt '21] [D'Amour et al. '20]

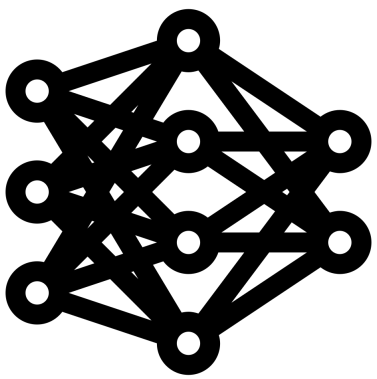
We want to attribute model **class**, not a single model



$$\tau = \frac{1}{M} \sum_m \tau^{(m)}$$

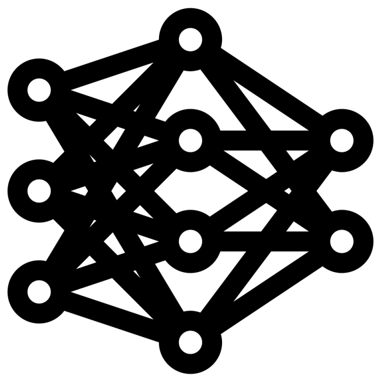
Average attribution scores over an ensemble of M models

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



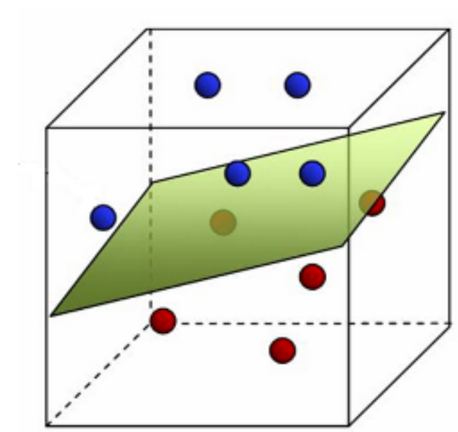
Original neural network

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



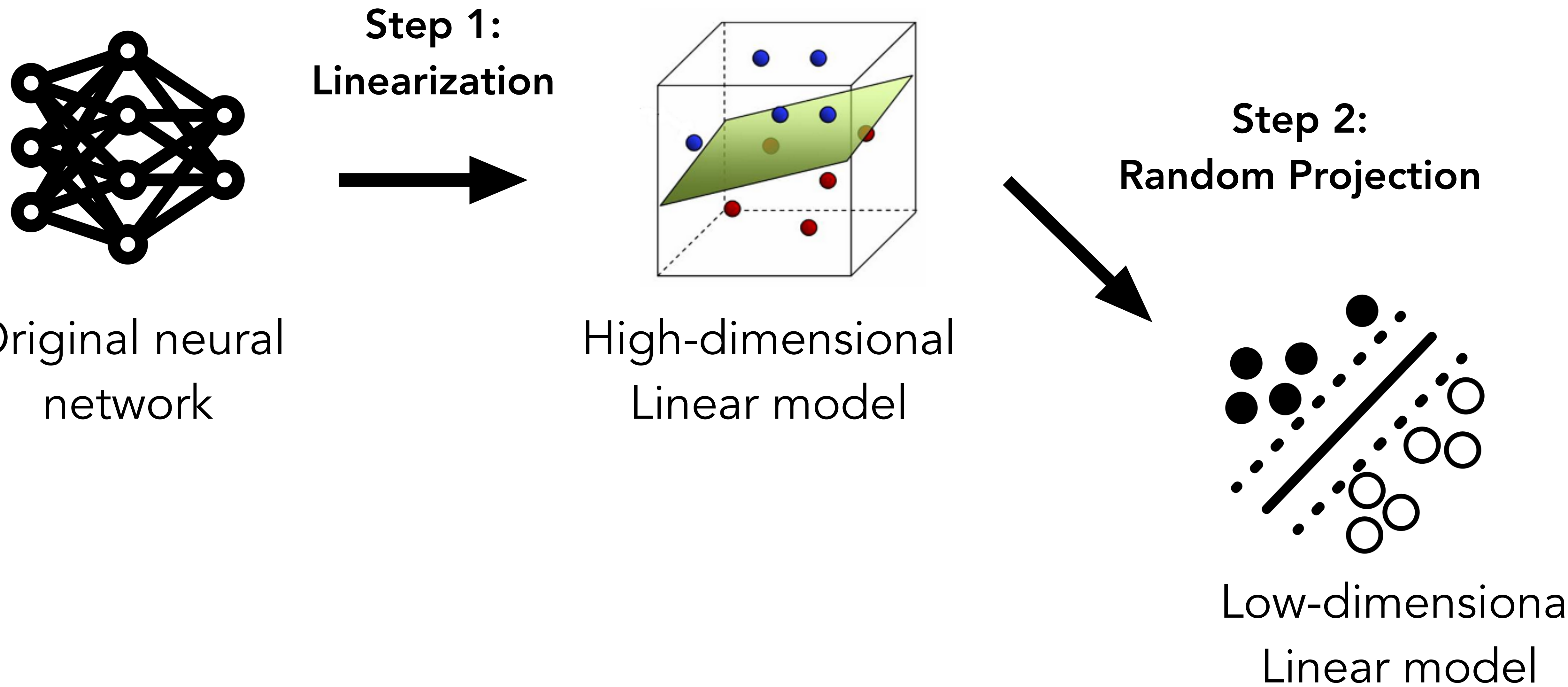
Original neural network

Step 1: Linearization

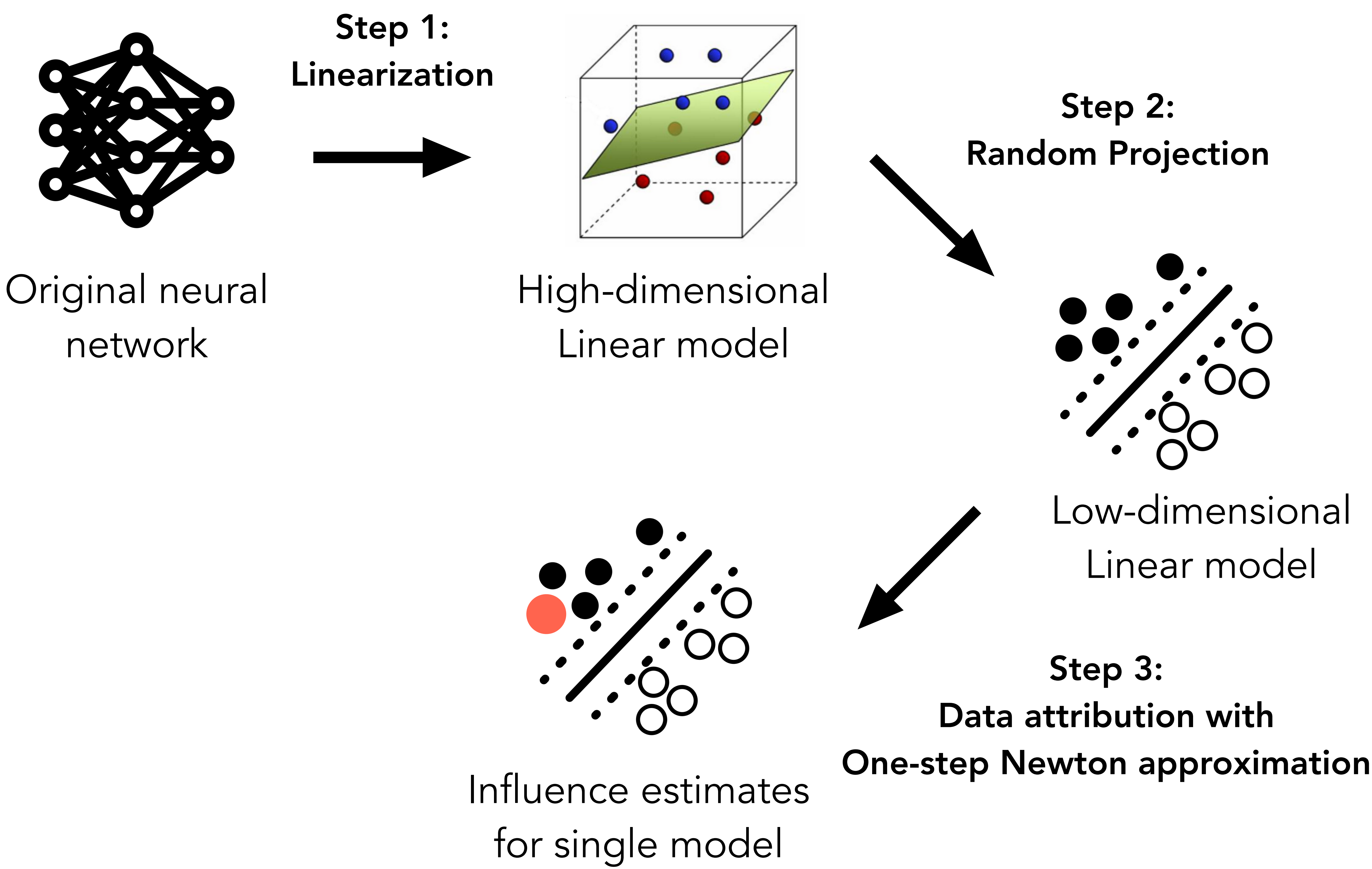


High-dimensional Linear model

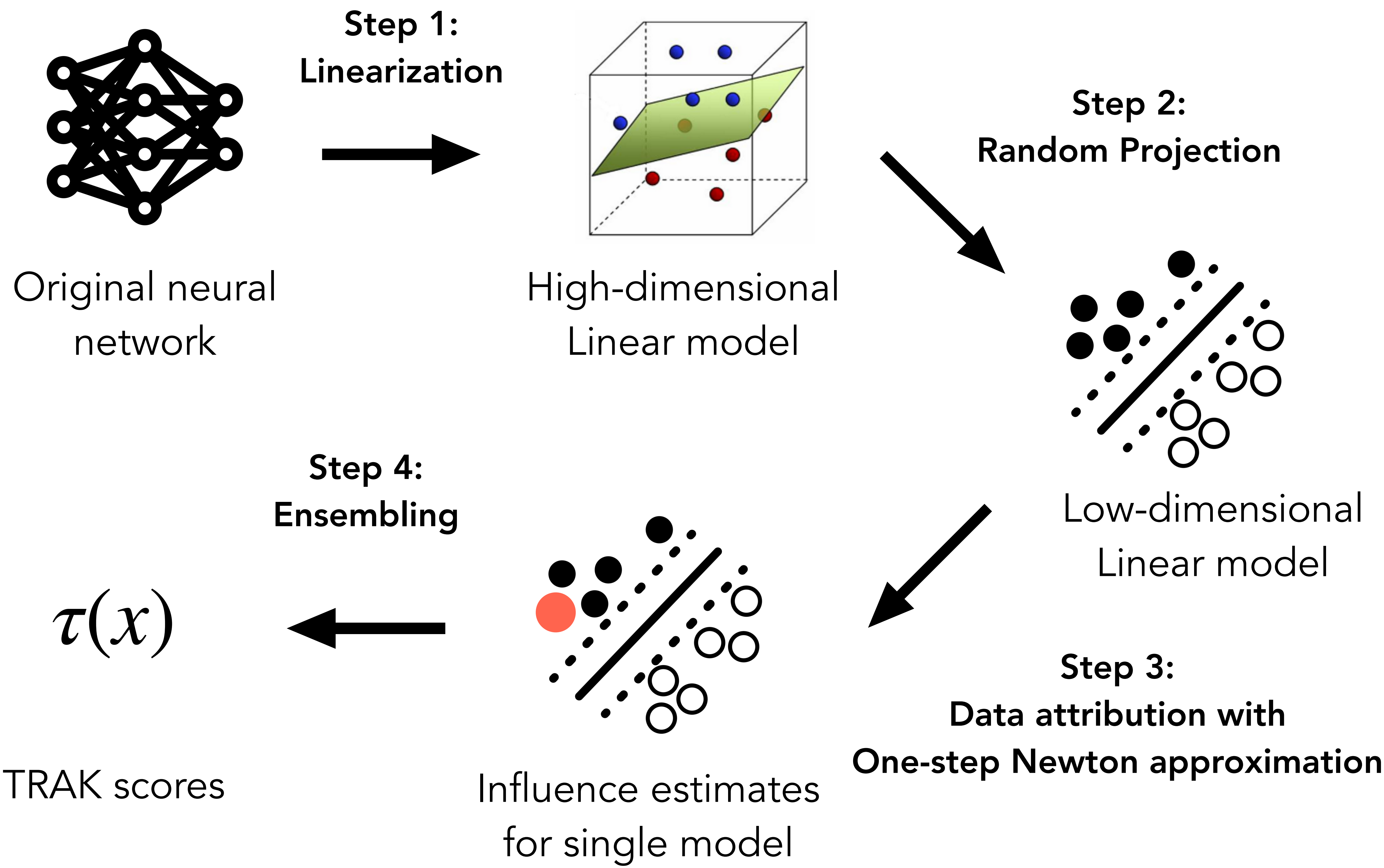
Tracing with **R**andom projections of the **A**fter **K**ernel



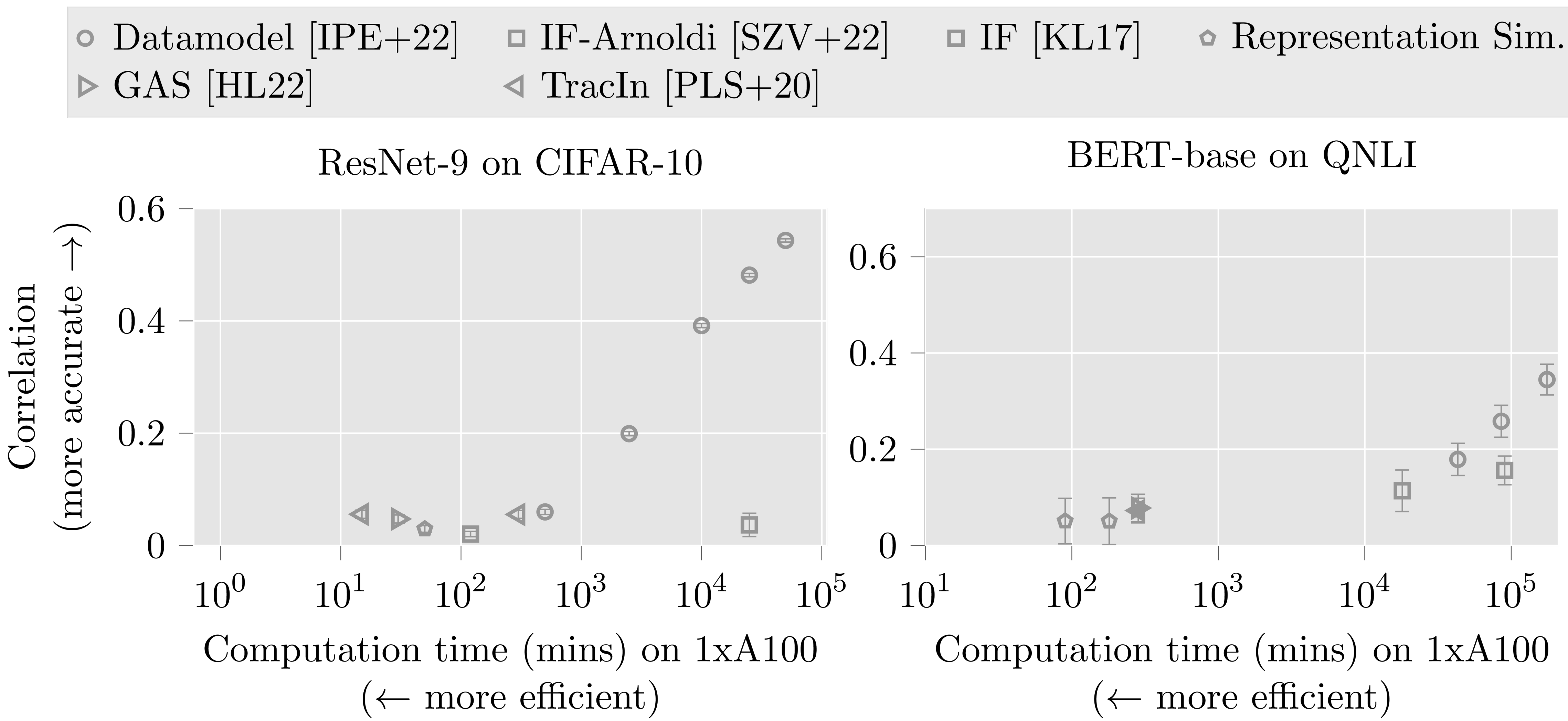
Tracing with **R**andom projections of the **A**fter **K**ernel



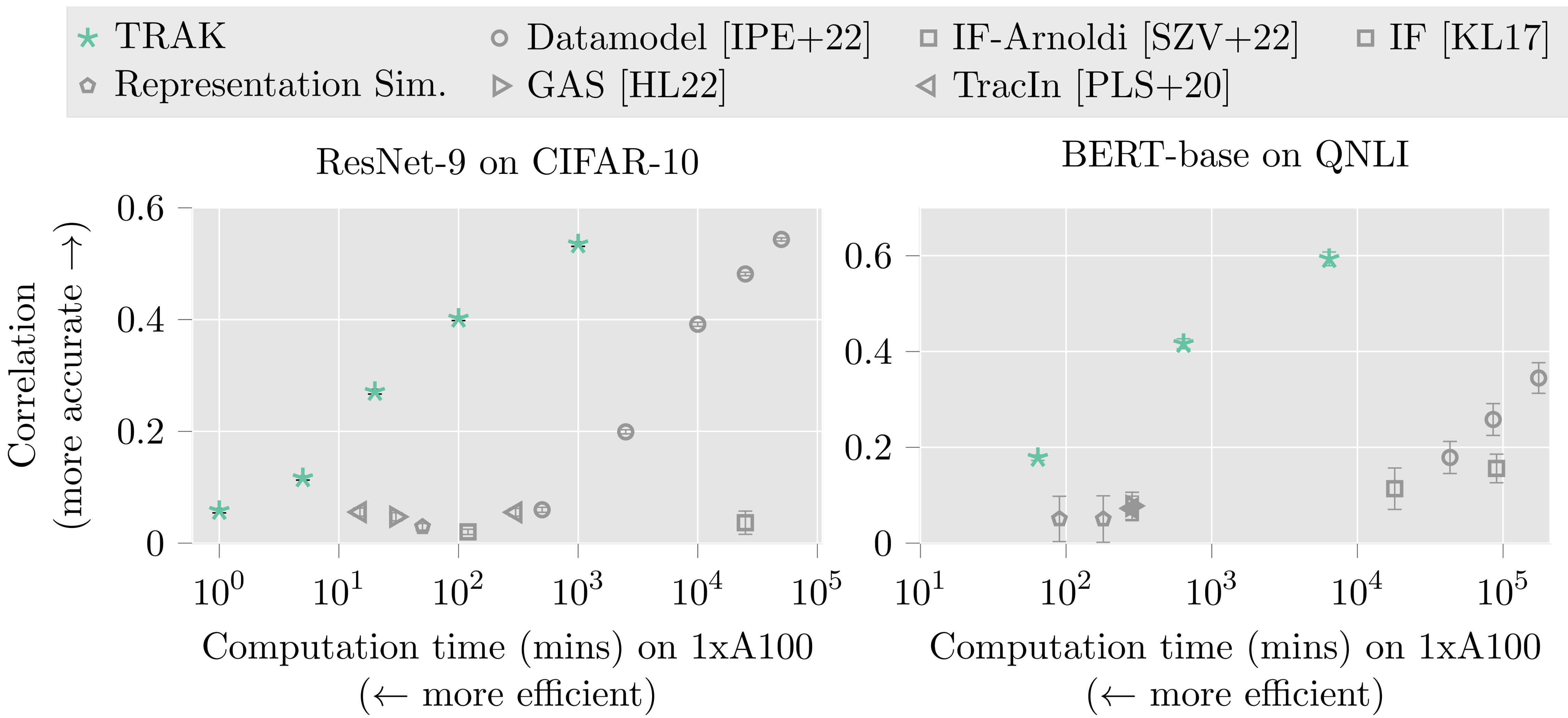
Tracing with **R**andom projections of the **A**fter **K**ernel



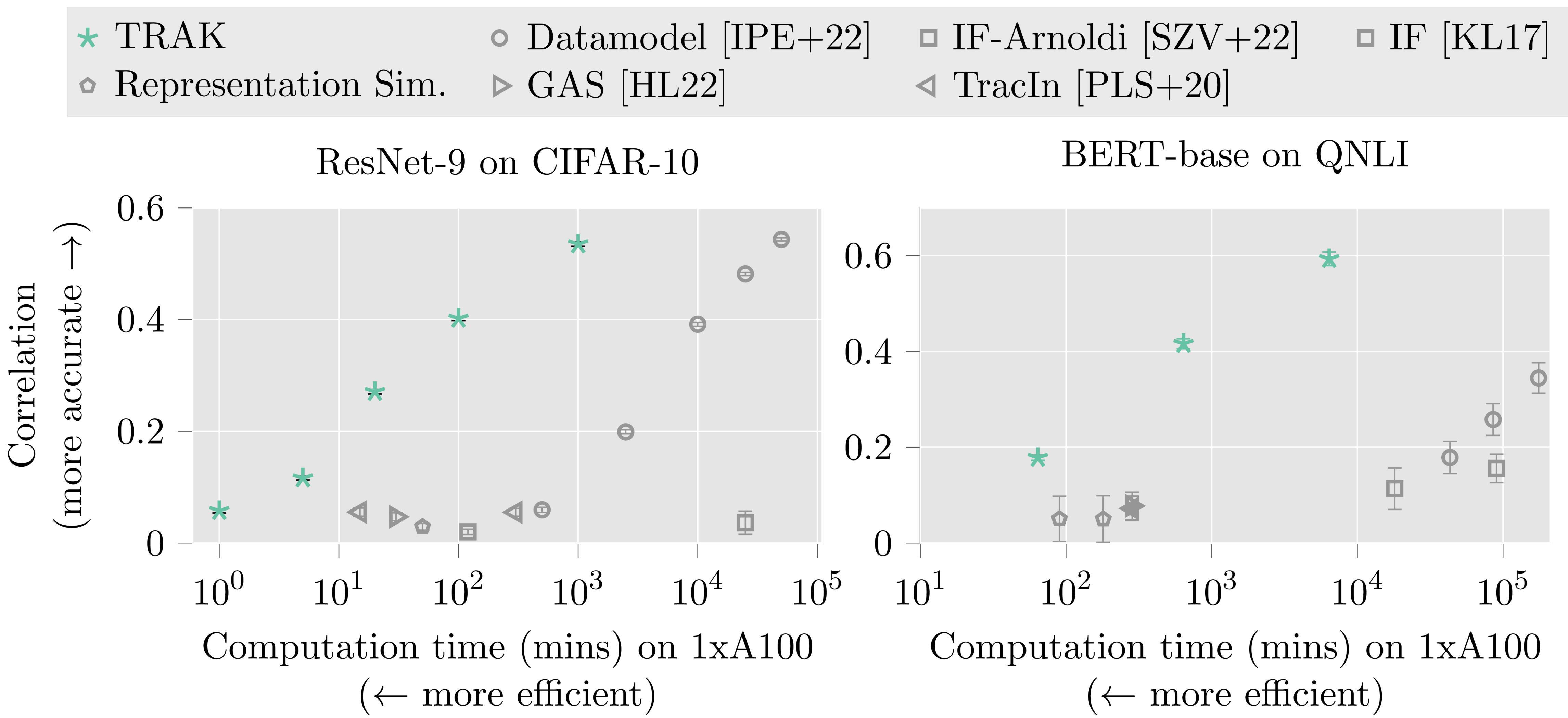
Evaluating TRAK



Evaluating TRAK

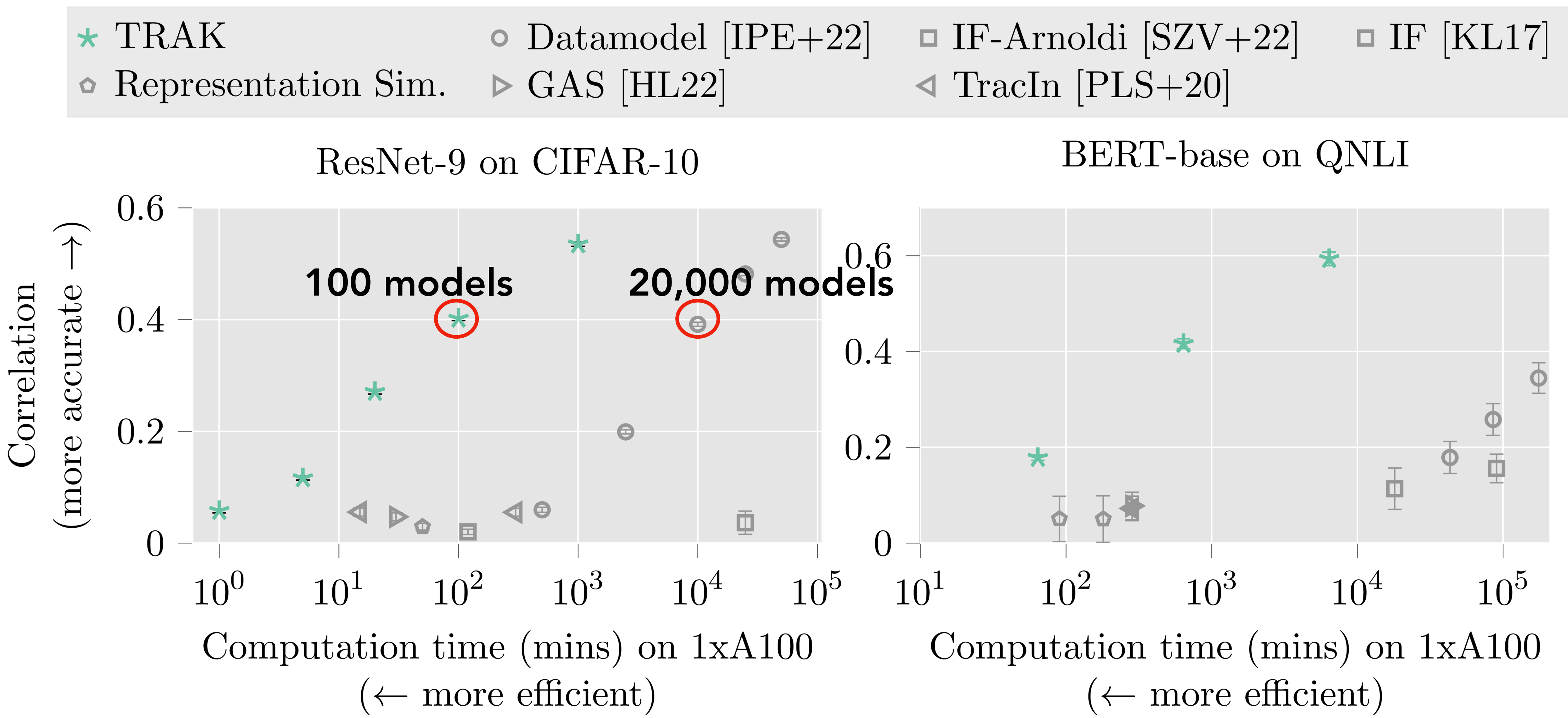


Evaluating TRAK



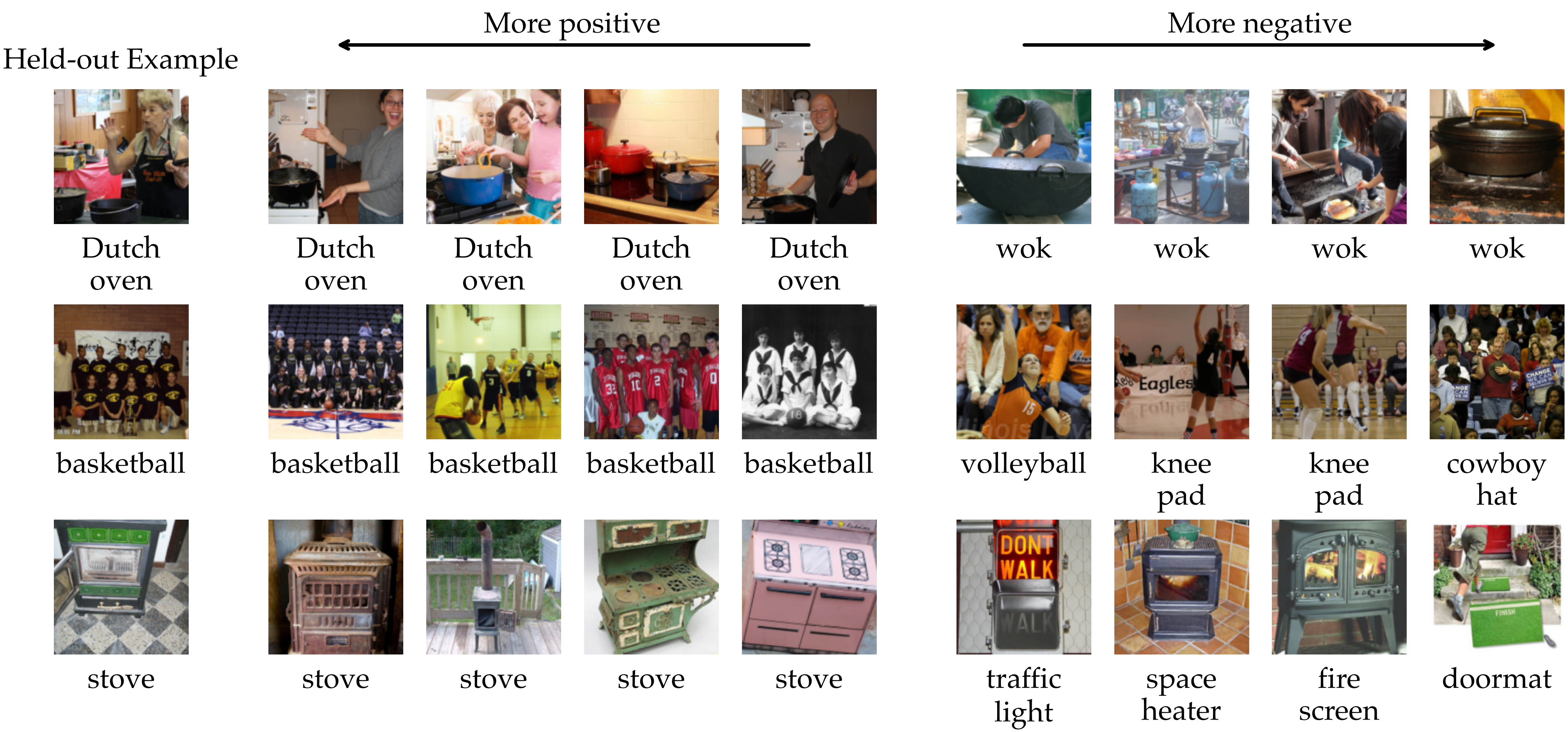
TRAK speeds up datamodels by 100x-1000x

Evaluating TRAK



TRAK speeds up datamodels by 100x-1000x

Example TRAK attributions: ResNet-18 on ImageNet



(More examples in trak.csail.mit.edu)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: What is the name associated with the eight areas that make up a part of southern California?
A: Southern California consists of one Combined Statistical Area, eight Metropolitan Statistical Areas, one international metropolitan area, and multiple metropolitan divisions. (Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: What is the name associated with the eight areas that make up a part of southern California?
A: Southern California consists of one Combined Statistical Area, eight Metropolitan Statistical Areas, one international metropolitan area, and multiple metropolitan divisions. (Entailment)

(Most positive influence)
Q: Was the name given to the Alsace provincinal court?
A: The province had a single provincial court (Landgericht) and a central administration with its seat at Hagenau. (Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: What is the name associated with the eight areas that make up a part of southern California?
A: Southern California consists of one Combined Statistical Area, eight Metropolitan Statistical Areas, one international metropolitan area, and multiple metropolitan divisions. (Entailment)

(Most positive influence)
Q: Was the name given to the Alsace provincinal court?
A: The province had a single provincial court (Landgericht) and a central administration with its seat at Hagenau. (Entailment)

(Most negative influence)
Q: What is one of the eight factors?
A: The Noble Eightfold Path—the fourth of the Buddha's Noble Truths—consists of a set of eight interconnected factors or conditions, that when developed together… (No Entailment)

Applications

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

- CLIP
- Language models
- ImageNet classifiers

BERT, mT5

Applications

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

- CLIP
- Language models
- ImageNet classifiers

OpenAI CLIP

BERT, mT5

Large Language Models

Large Language Models

"Lionel Messi won the
Ballon d'Or seven times."

Large Language Models

"Lionel Messi won the
Ballon d'Or seven times."

Why did the language model output this answer?

Large Language Models

"Lionel Messi won the
Ballon d'Or seven times."

Why did the language model output this answer?

Can we identify the training data that led to this output?

Large Language Models

"Lionel Messi won the
Ballon d'Or seven times."

Why did the language model output this answer?

Can we identify the training data that led to this output?

One task for studying this question: **fact tracing**

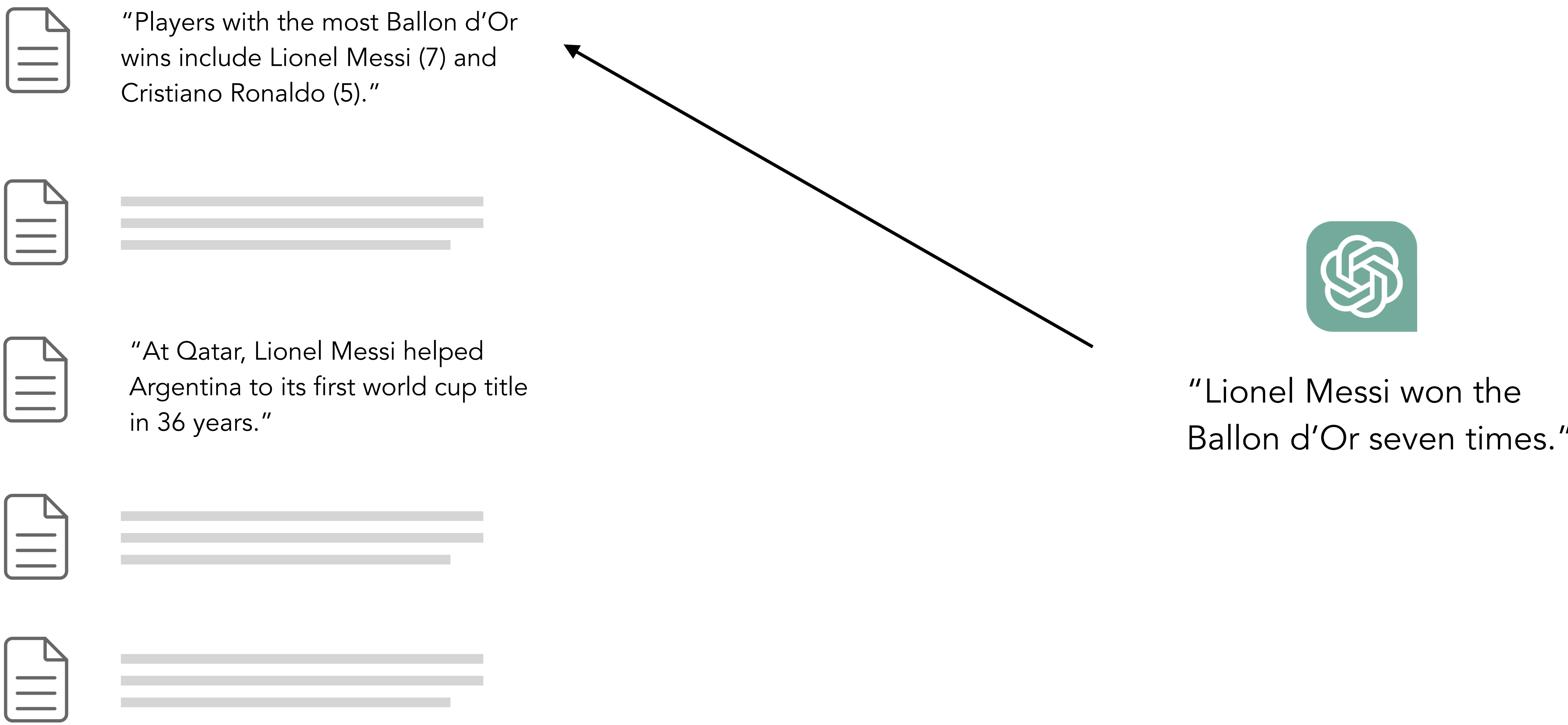
Fact tracing

"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."

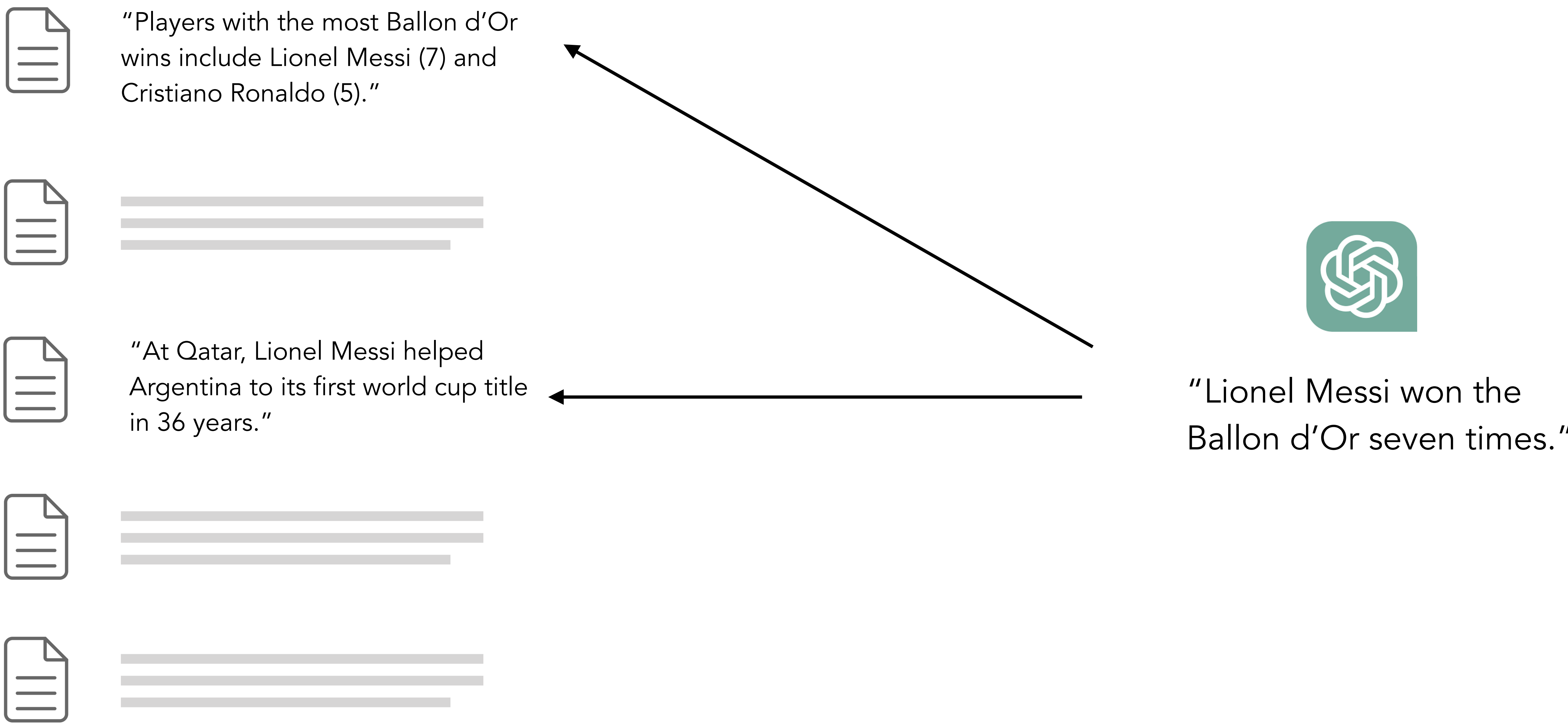
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."

"Lionel Messi won the Ballon d'Or seven times."

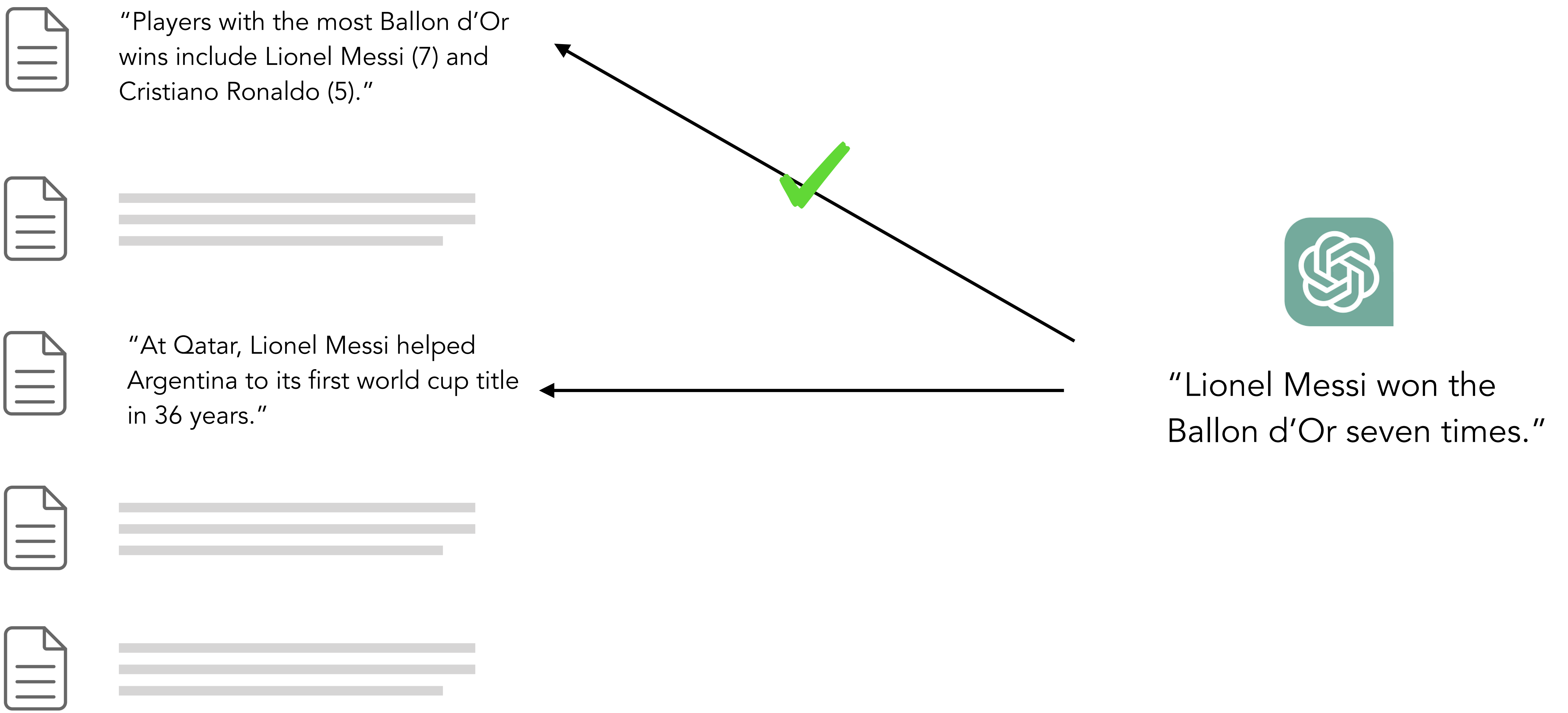
Fact tracing



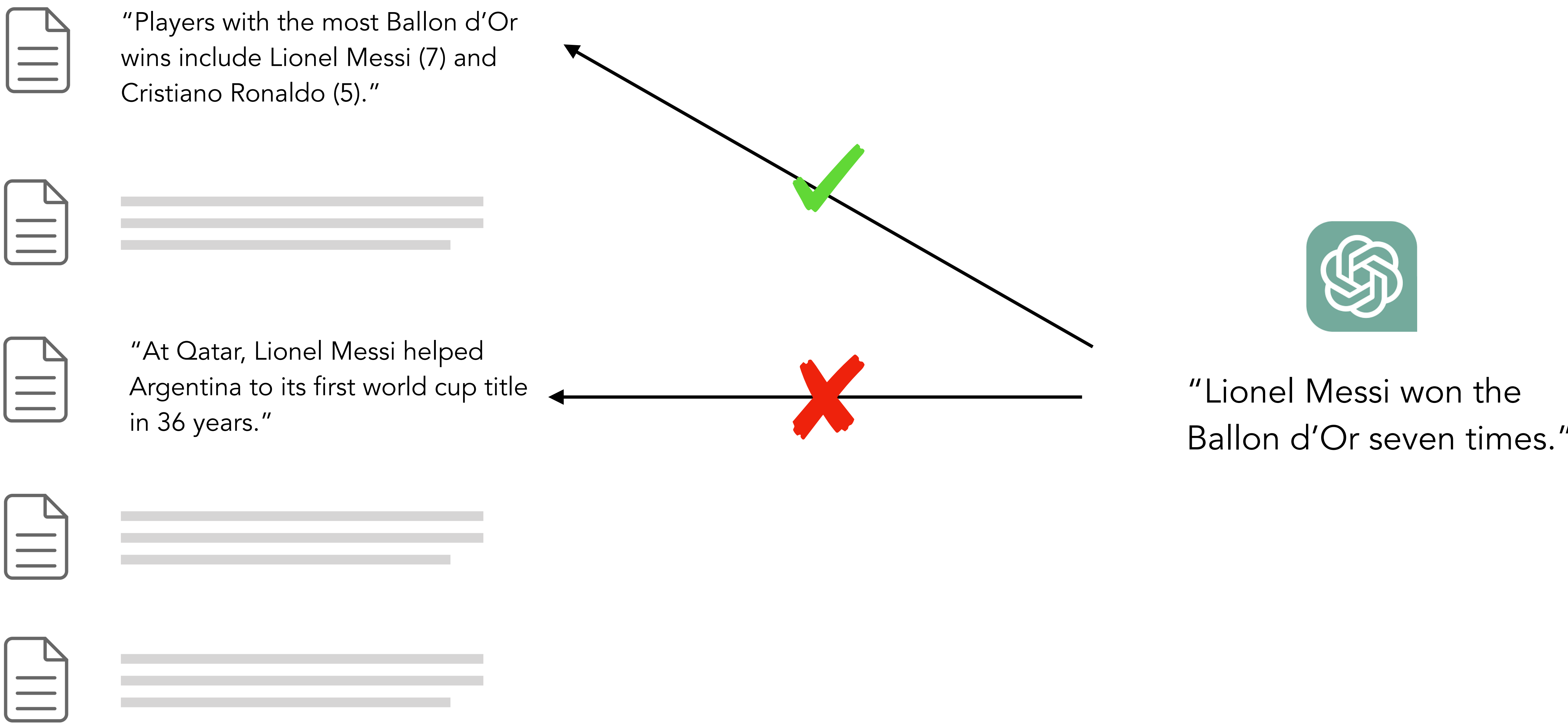
Fact tracing



Fact tracing



Fact tracing



Fact tracing: FTrace-TREx

[Akyurek et al. ’22]

“Players with the most Ballon d’Or wins include Lionel Messi (7) and Cristiano Ronaldo (5).”

“At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years.”

Fact “Lionel Messi won the Ballon d’Or seven times.”

Fact tracing: FTrace-TREx

[Akyurek et al. ’22]

“Players with the most Ballon d’Or wins include Lionel Messi (7) and Cristiano Ronaldo (5).”

“At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years.”

Fact “Lionel Messi won the Ballon d’Or seven times.”

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."

"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."

Query "Lionel Messi won the __________ seven times."

Answer "Ballon d'Or"

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

Abstracts

"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."

"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."

Query "Lionel Messi won the __________ seven times."

Answer "Ballon d'Or"

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

Abstracts

Abstracts	Ground-truth
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓
	✗
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗
	✓
	✗

Query "Lionel Messi won the ________ seven times."

Answer "Ballon d'Or"

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

Abstracts	Ground-truth	Attribution score
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9
	✗	0.05
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5
	✓	0.2
	✗	-0.1

Query "Lionel Messi won the __________ seven times."

Answer "Ballon d'Or"

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

Abstracts

Abstract	Ground-truth	Attribution score
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9
	✗	0.05
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5
	✓	0.2
	✗	-0.1

Query "Lionel Messi won the ________ seven times."

Answer "Ballon d'Or"

Task: Identify training examples expressing same fact

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✓	0.5	0.8
	✓	0.2	0.5
	✗	-0.1	0.

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
	✓	0.2	0.5
	✗	-0.1	0.

Results: TRAK performs *worse* than an information retrieval baseline (BM25). Why?

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
	✗	-0.1	0.

Results: TRAK performs *worse* than an information retrieval baseline (BM25). Why?

Recall: our goal is to understand what data caused a model gave a certain prediction, not identify the source of the fact

Fact tracing: FTrace-TREx

[Akyurek et al. '22]

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3

Results: TRAK performs *worse* than an information retrieval baseline (BM25). Why?

Recall: our goal is to understand what data caused a model gave a certain prediction, not identify the source of the fact

Can we test this more directly?

Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

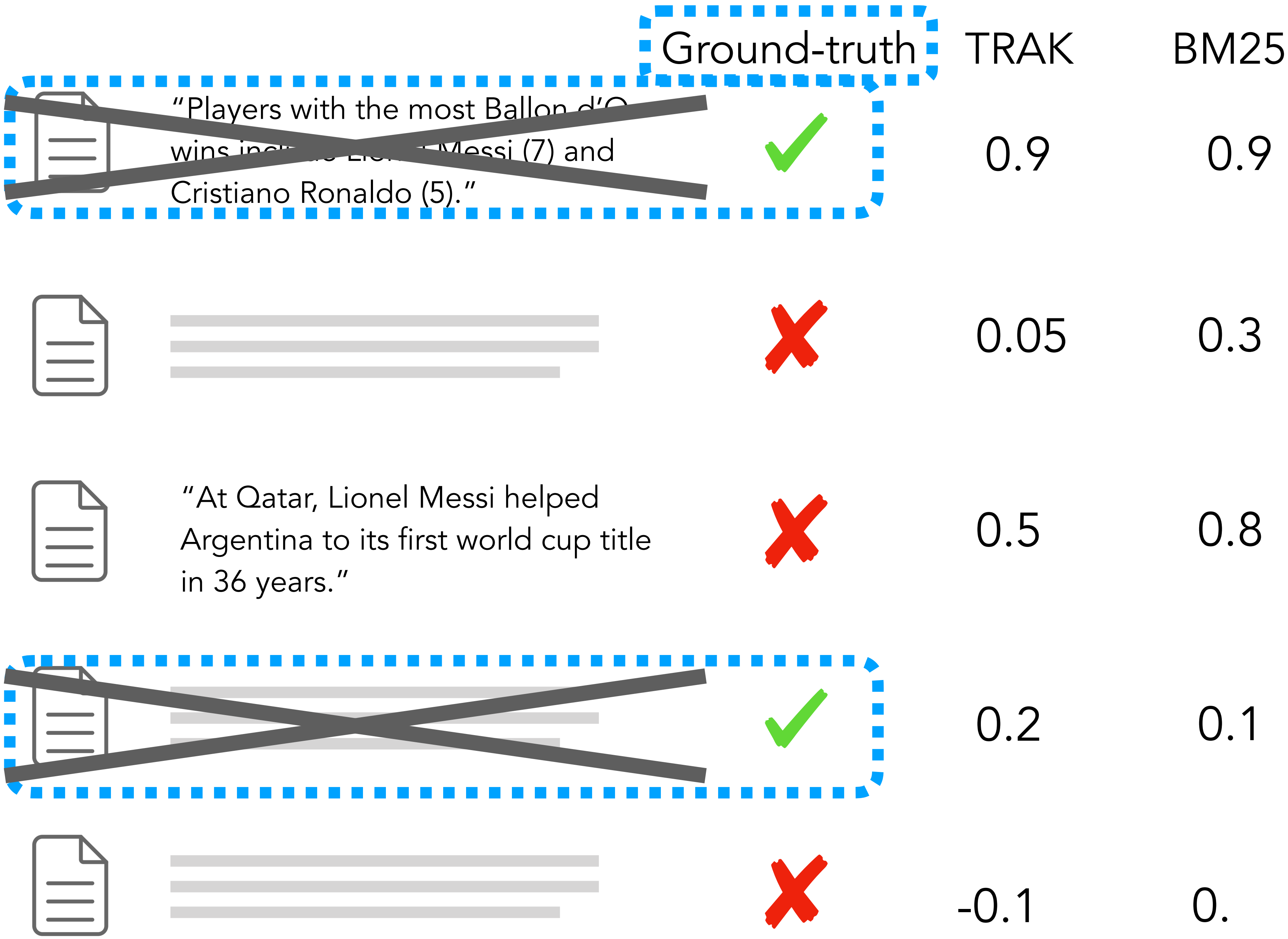
Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

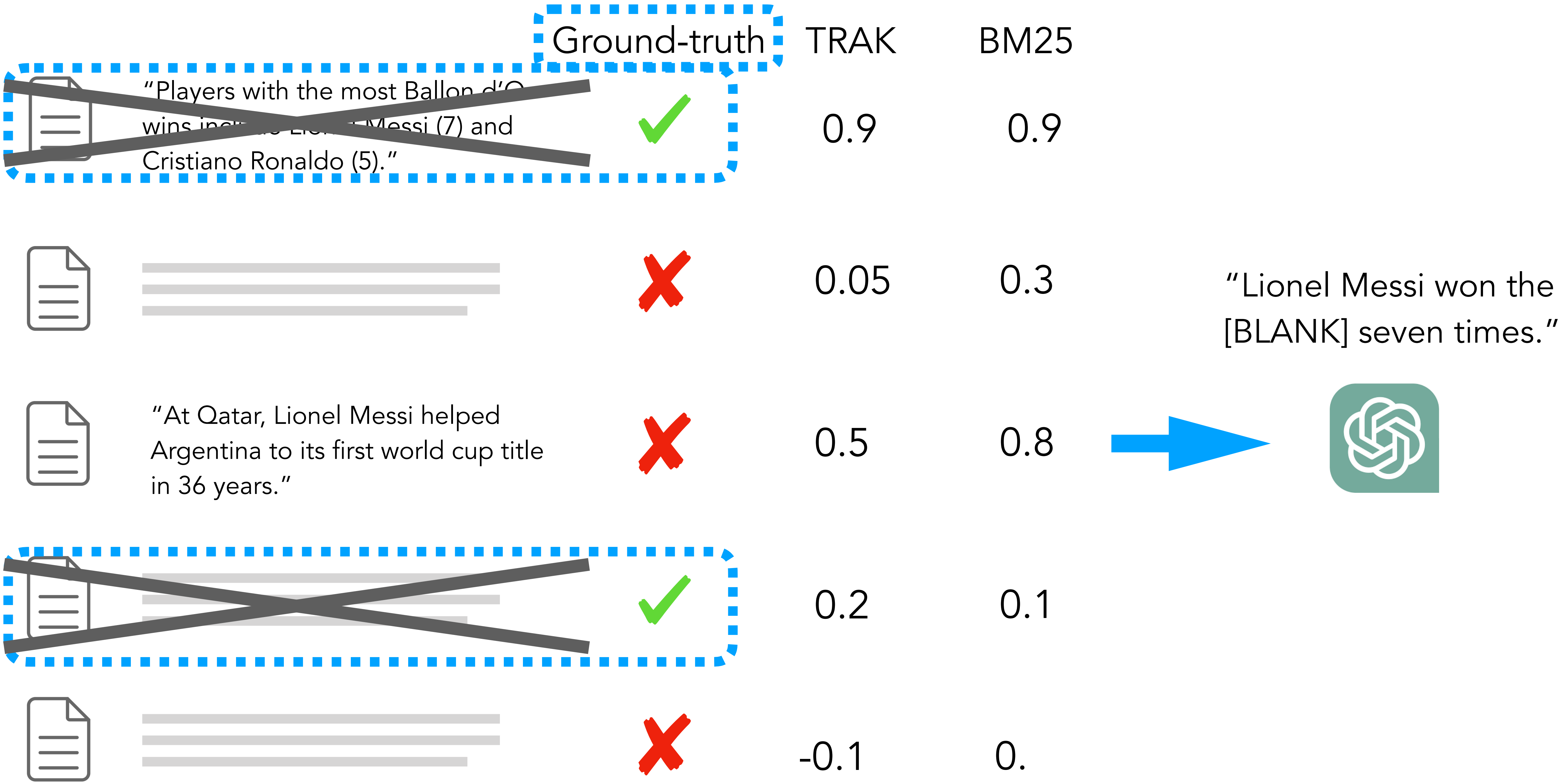
Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

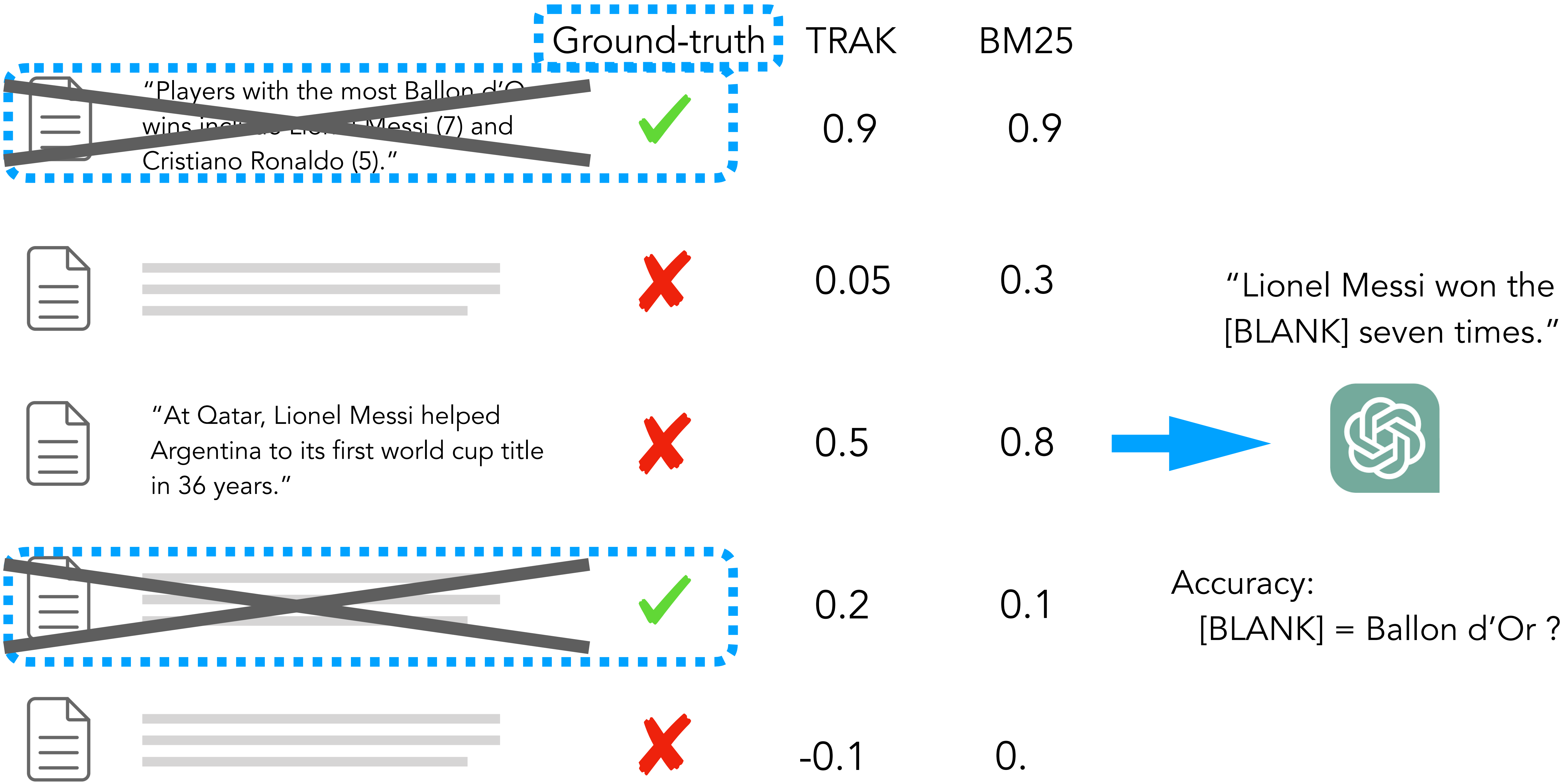
Counterfactual Analysis



Counterfactual Analysis



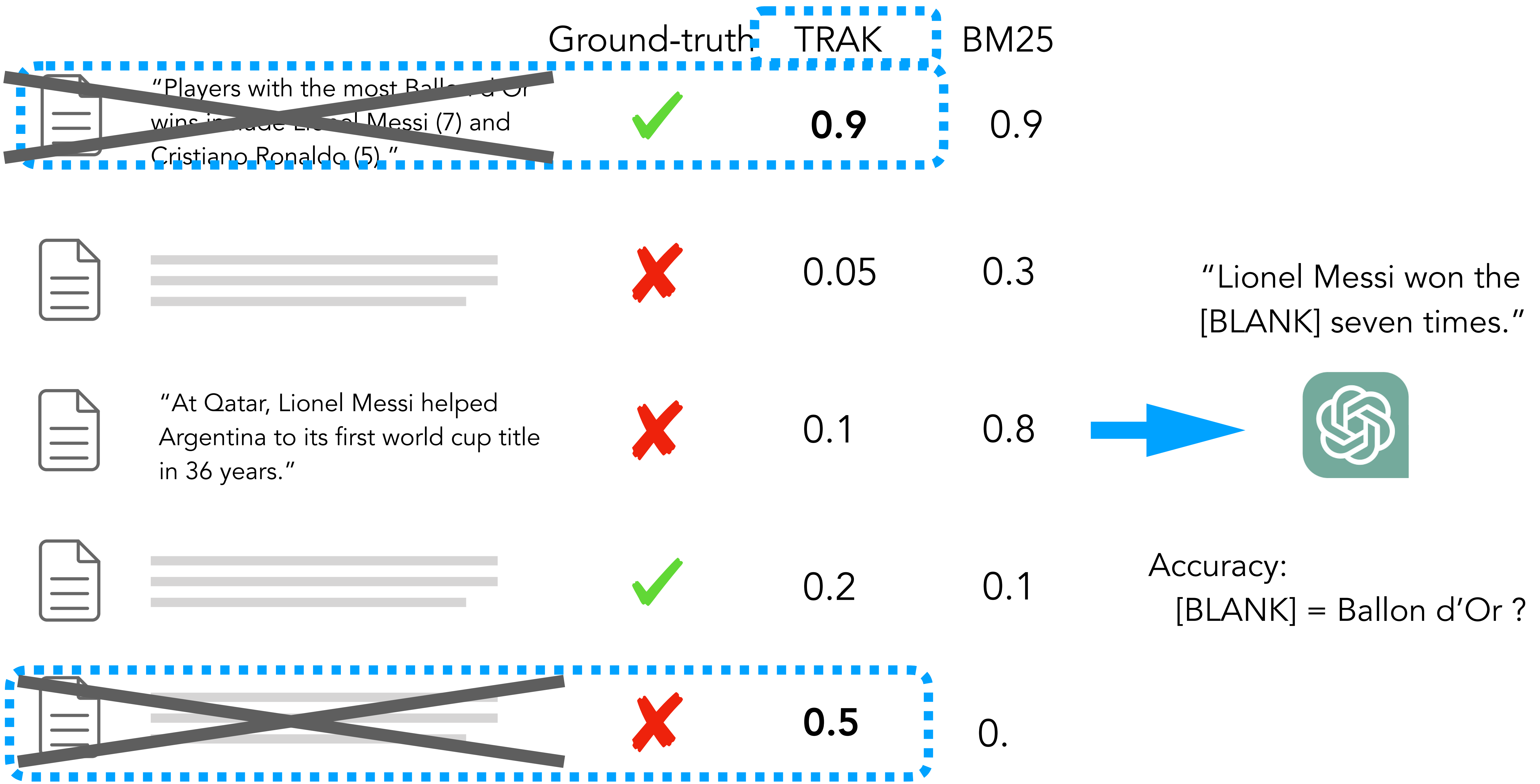
Counterfactual Analysis



Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

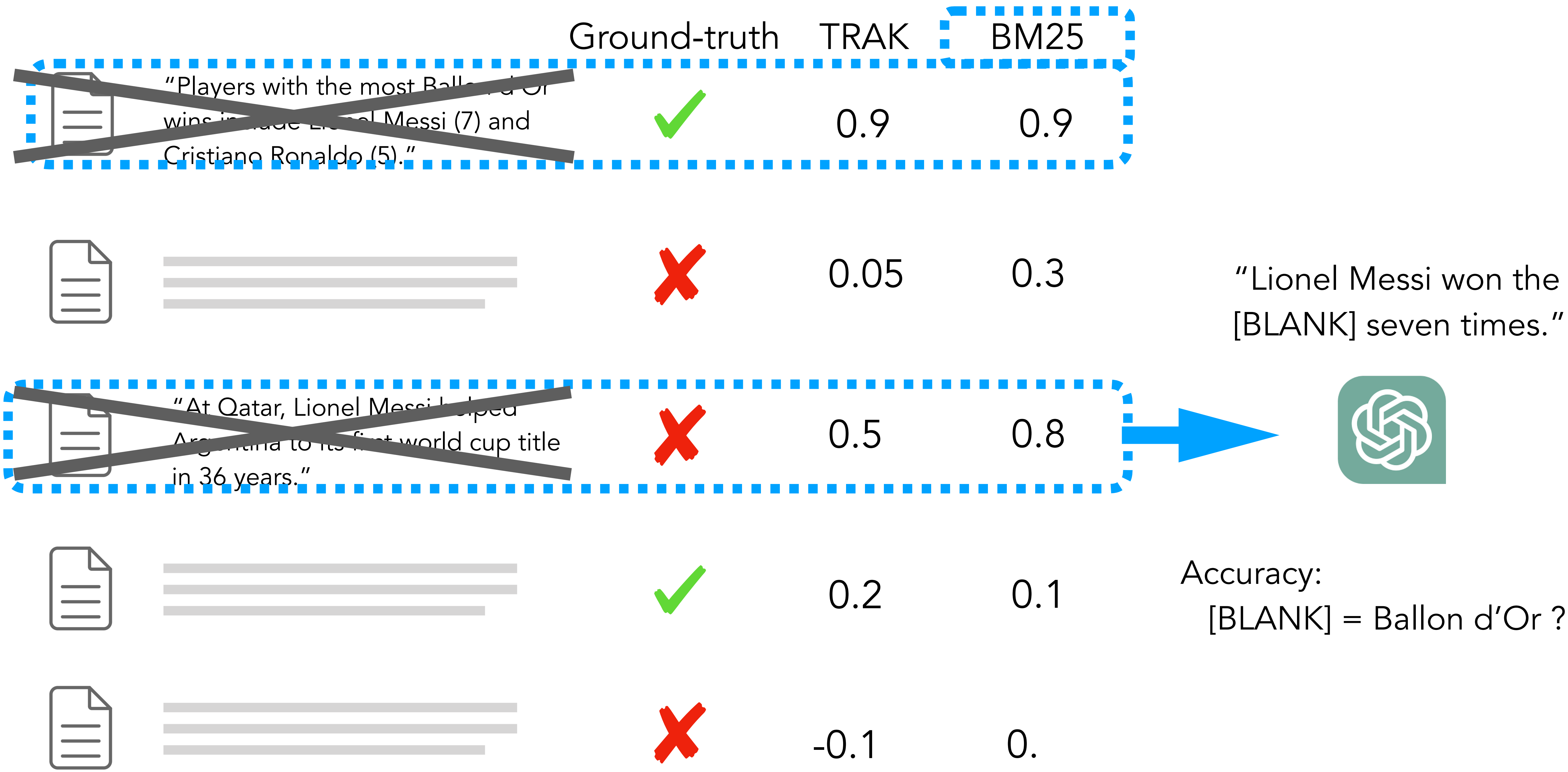
Counterfactual Analysis



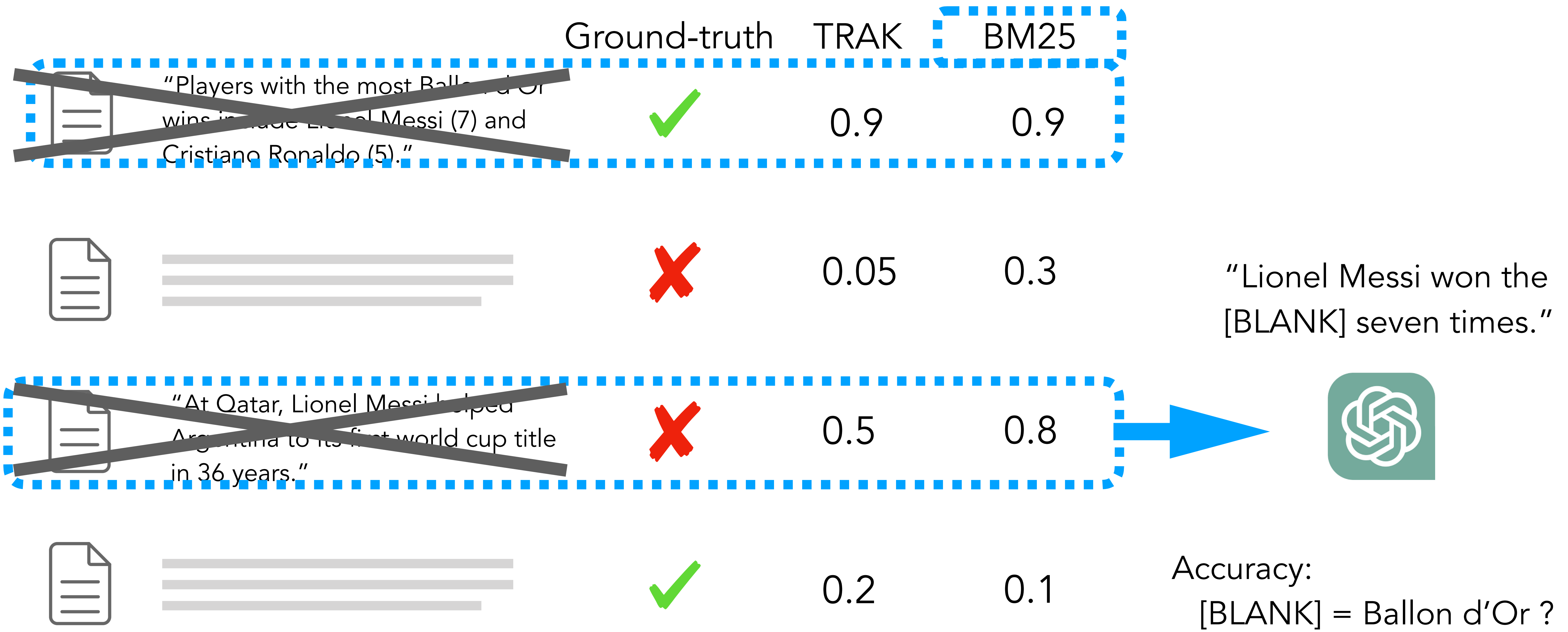
Counterfactual Analysis

	Ground-truth	TRAK	BM25
"Players with the most Ballon d'Or wins include Lionel Messi (7) and Cristiano Ronaldo (5)."	✓	0.9	0.9
	✗	0.05	0.3
"At Qatar, Lionel Messi helped Argentina to its first world cup title in 36 years."	✗	0.5	0.8
	✓	0.2	0.1
	✗	-0.1	0.

Counterfactual Analysis



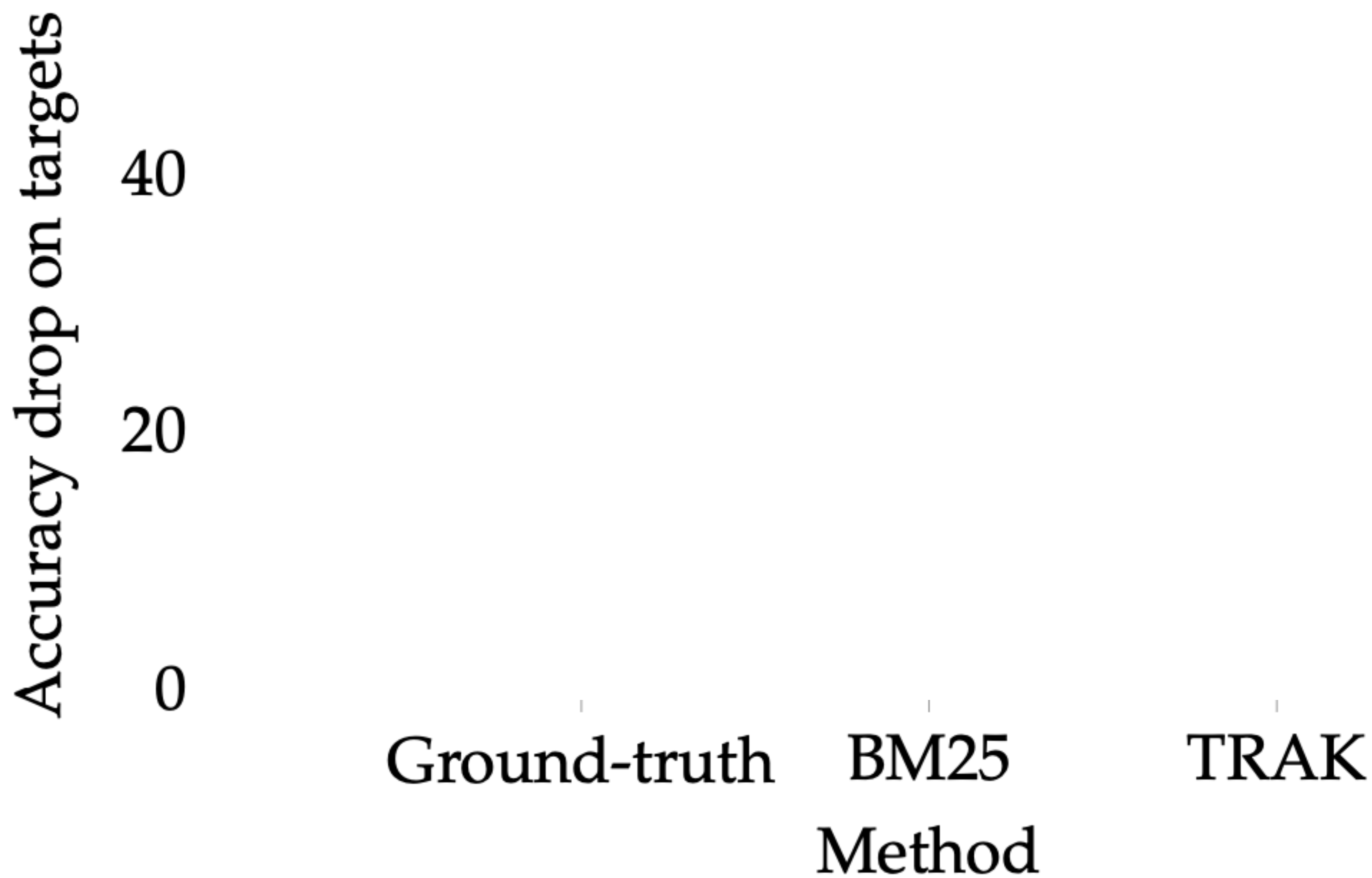
Counterfactual Analysis



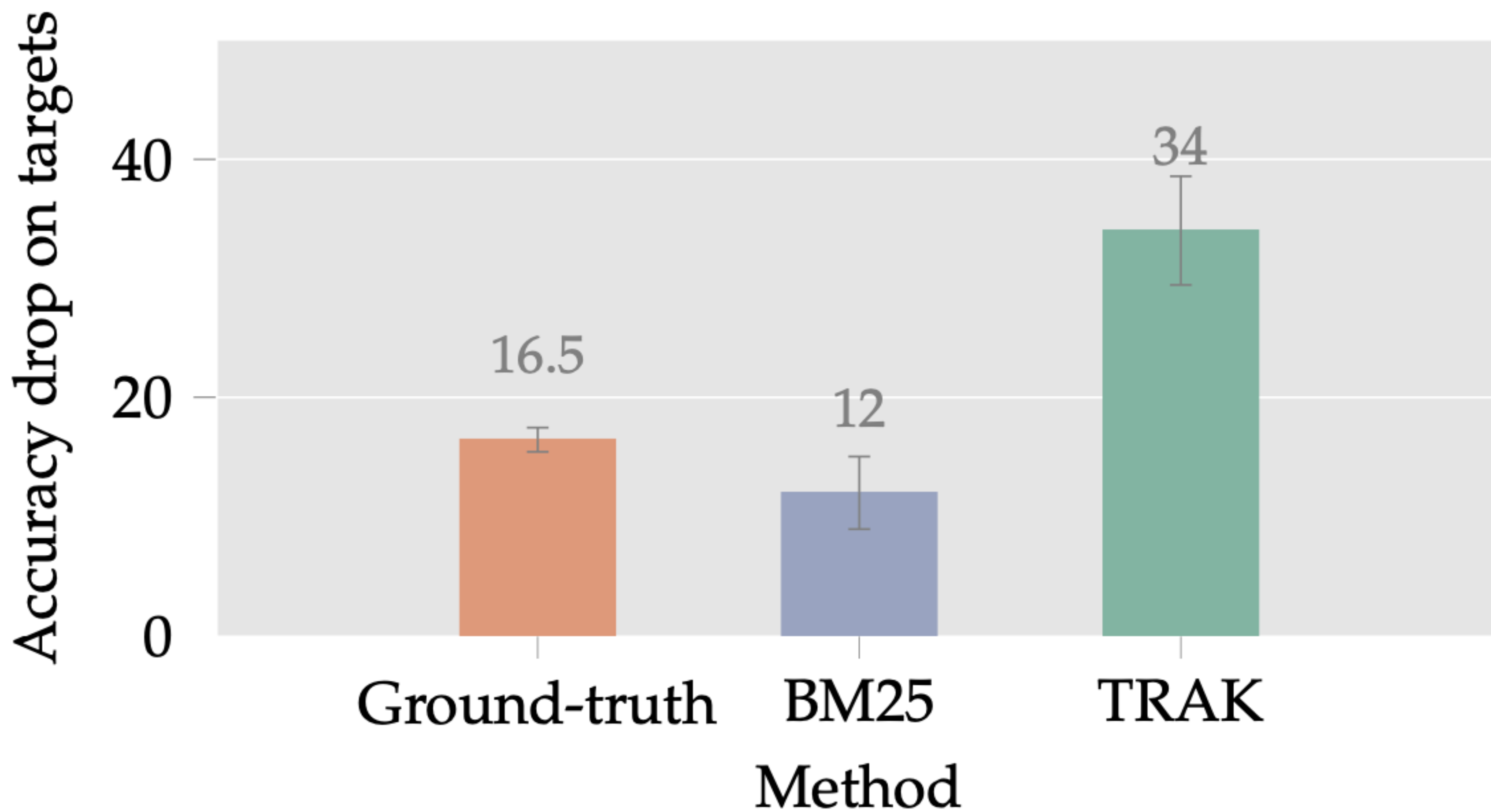
Experiment:

1. Remove top abstracts identified by each method
2. Retrain language model (mT5)
3. Measure (drop in) model accuracy on queries

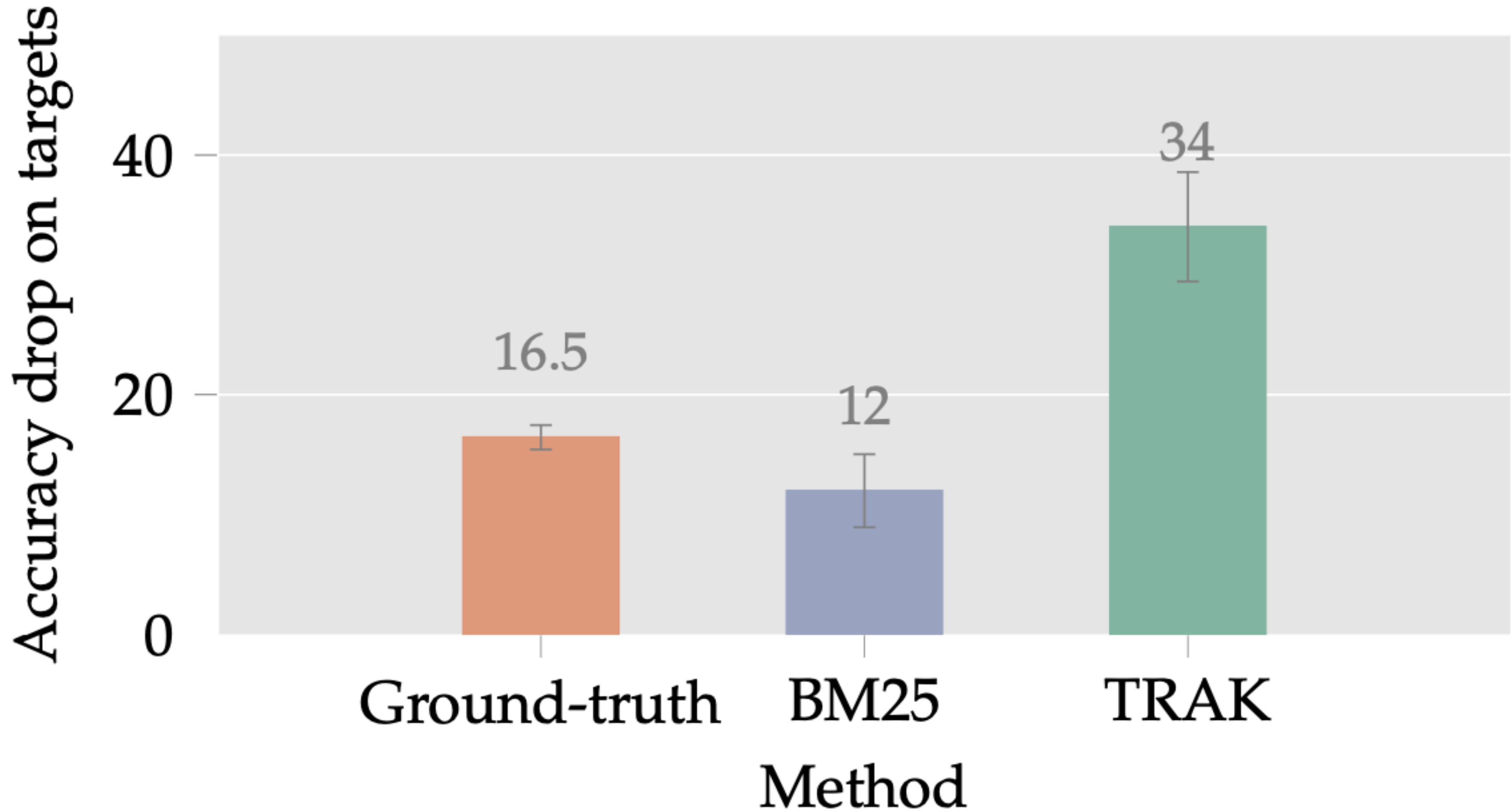
Counterfactual Analysis



Counterfactual Analysis

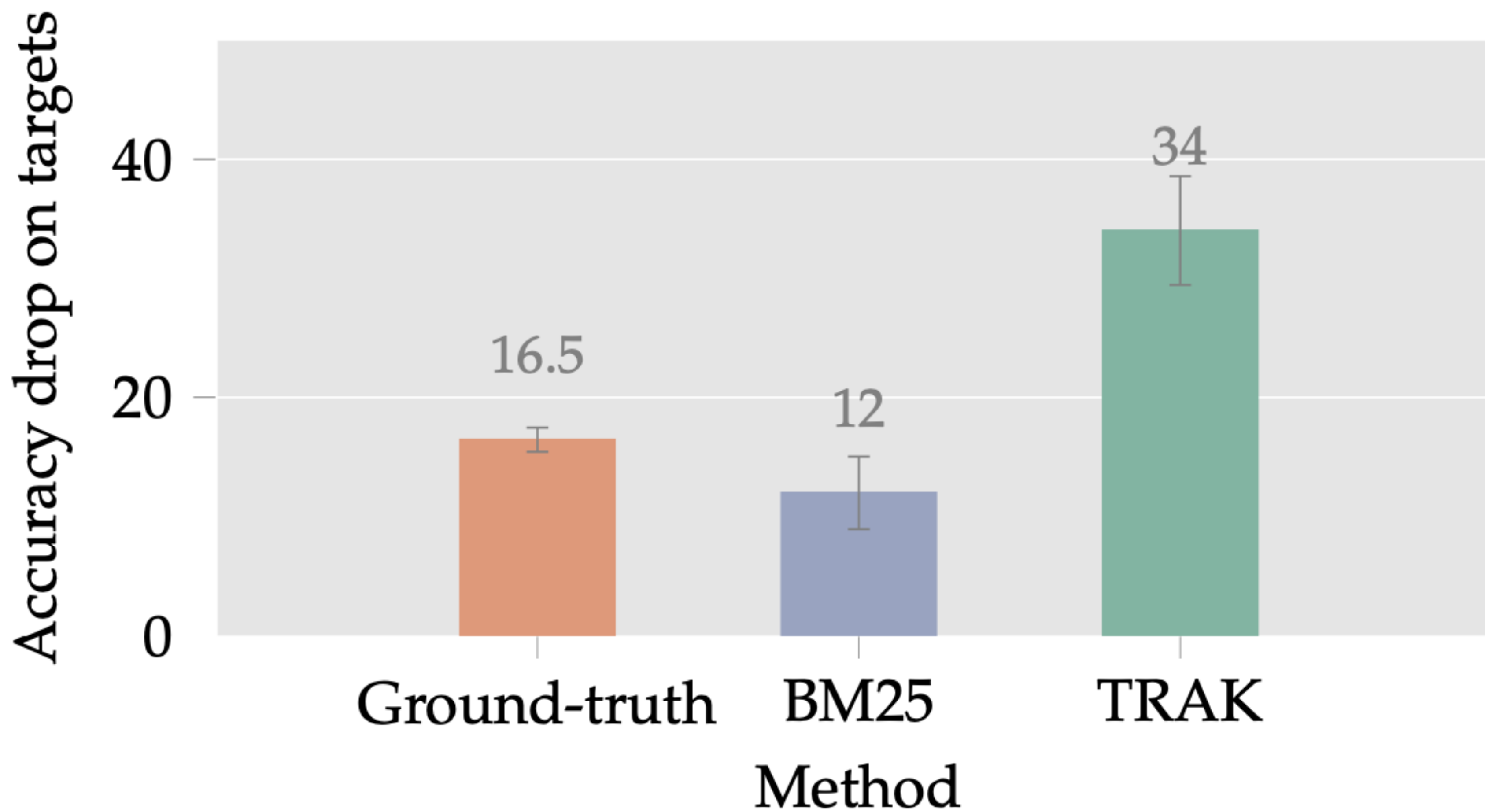


Counterfactual Analysis



Examples identified with TRAK are **counterfactually** much more important than even *ground-truth* facts

Counterfactual Analysis



Fact tracing ≠ Behavior tracing

What facts imply the generated text?

Model-**independent**

Why did the *model* generate the text?

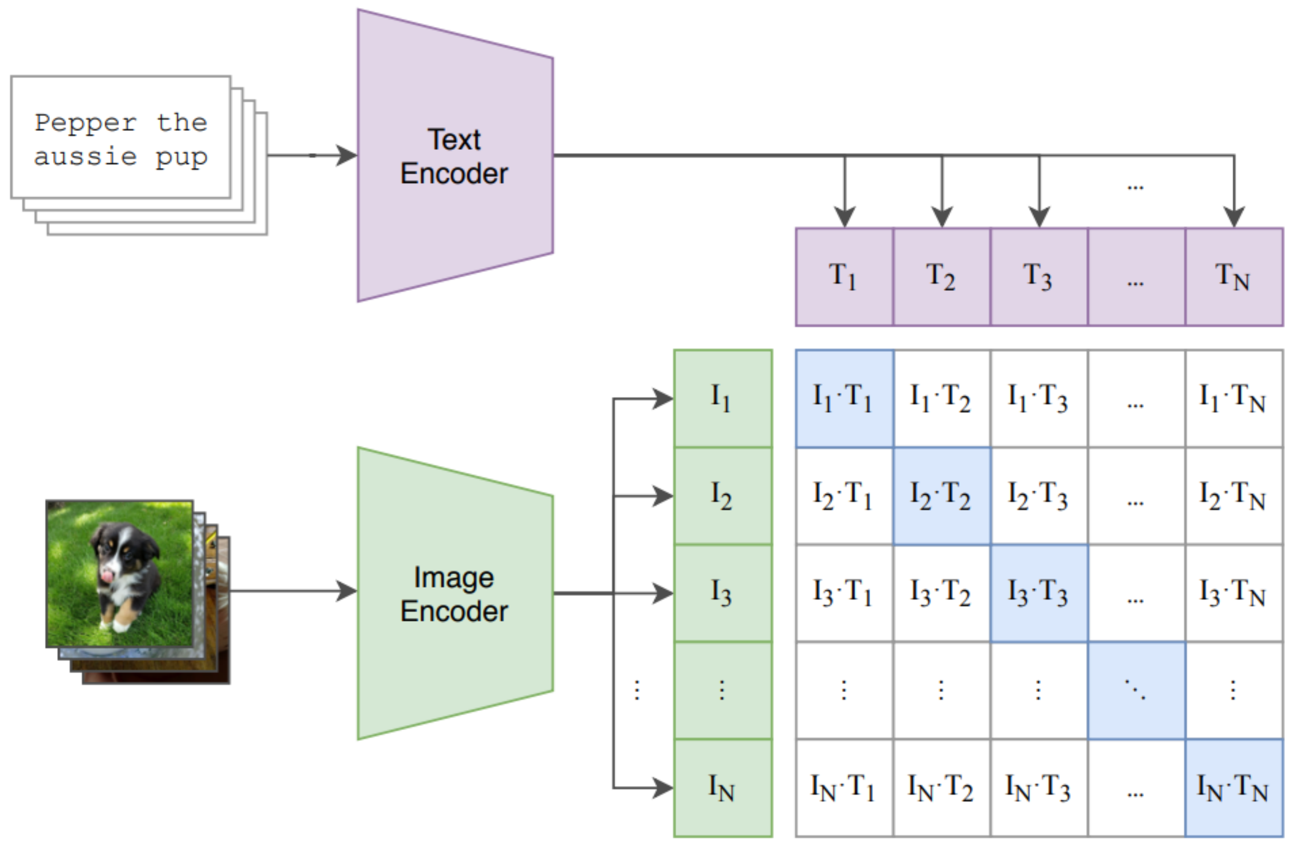
Model-**dependent**

CLIP

(Contrastive Language-Image Pre-training)

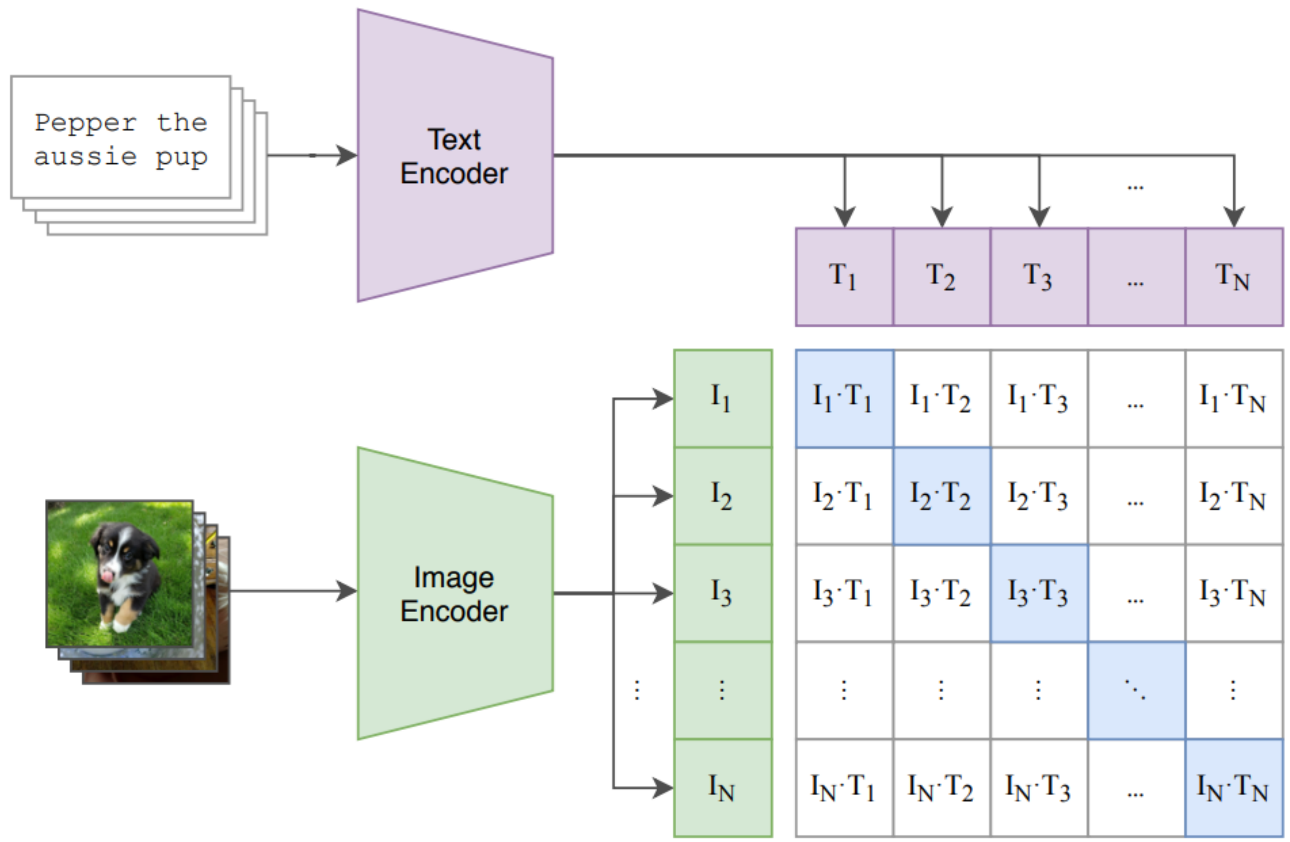
CLIP

(Contrastive Language-Image Pre-training)



CLIP

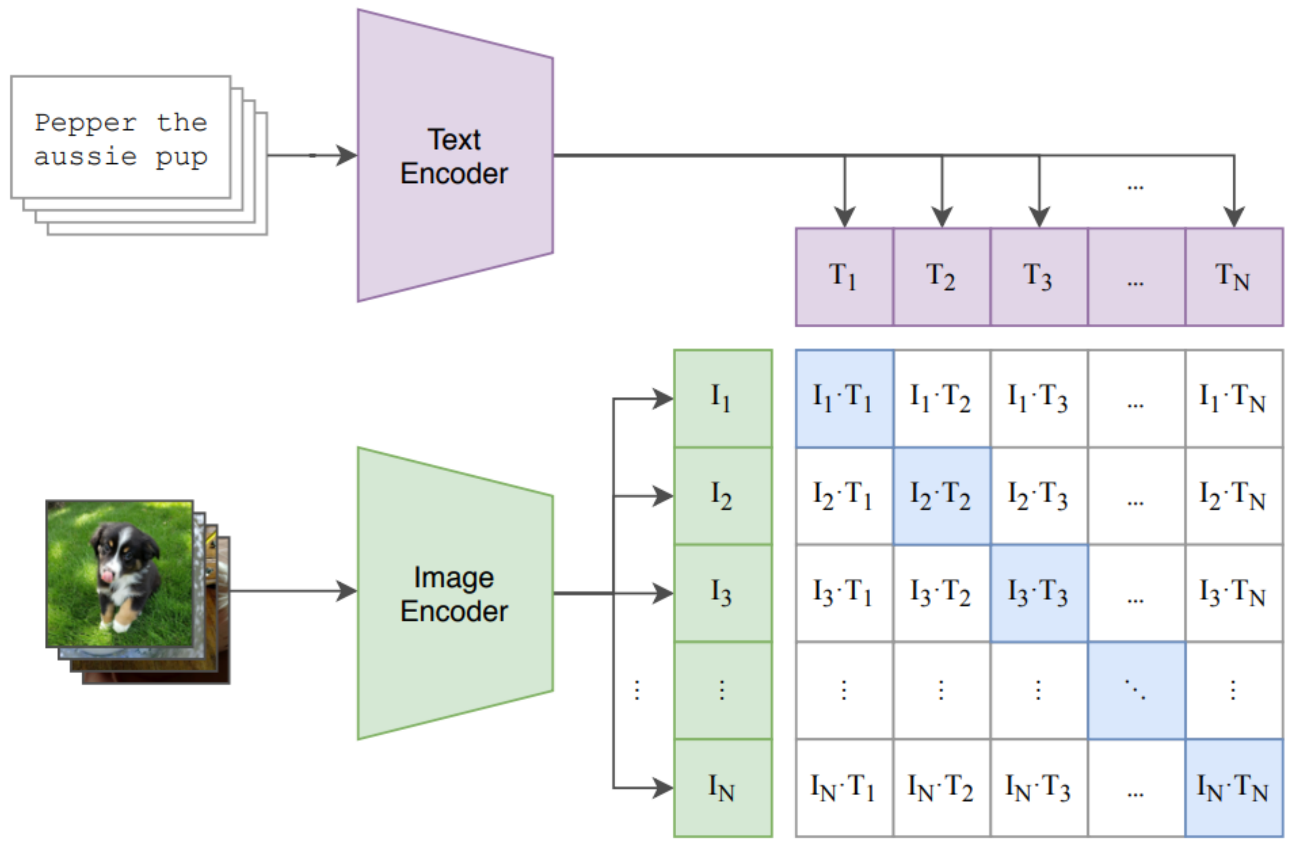
(Contrastive Language-Image Pre-training)



Translate: image ↔ text

CLIP

(Contrastive Language-Image Pre-training)



Translate: image ↔ text

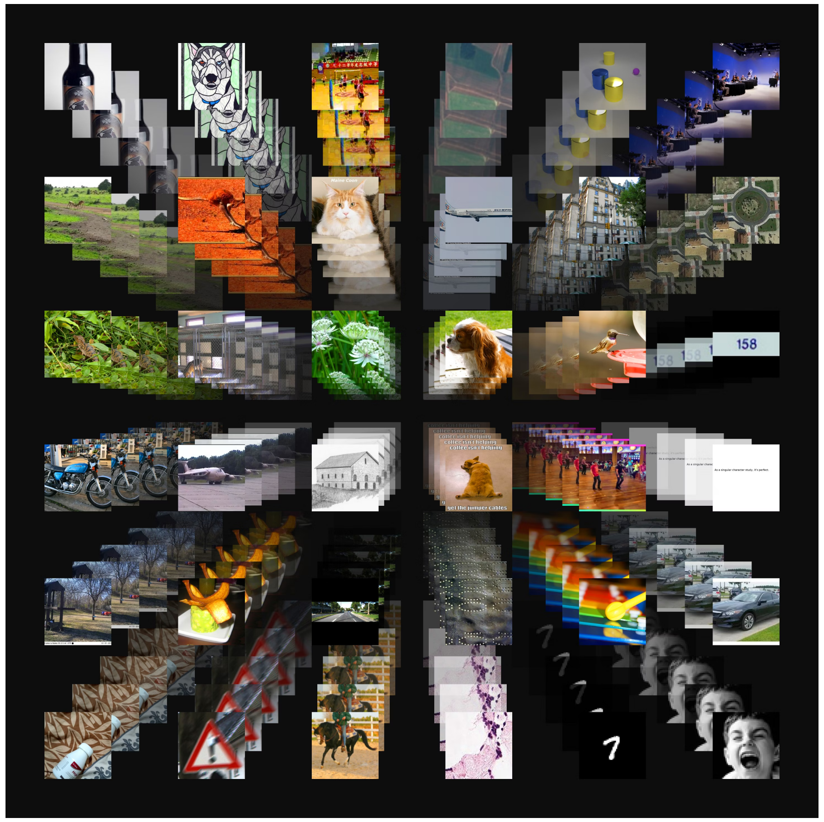
Many downstream applications:
zero-shot classification, StableDiffusion, etc.

CLIP
(Contrastive Language-Image Pre-training)

CLIP

(Contrastive Language-Image Pre-training)

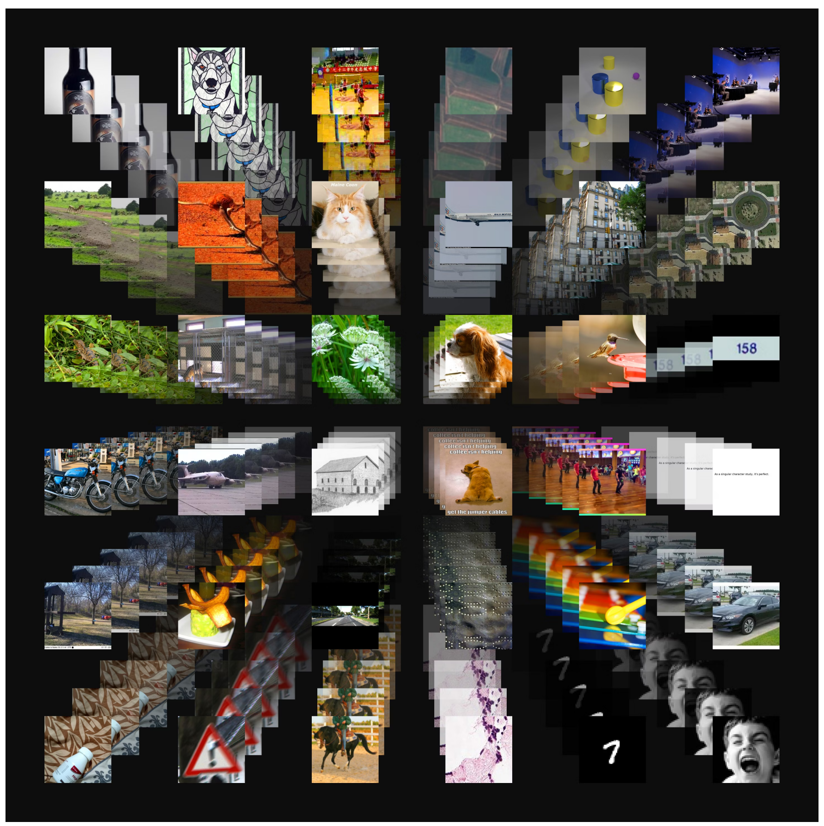
CLIP models are trained on vast amounts of data



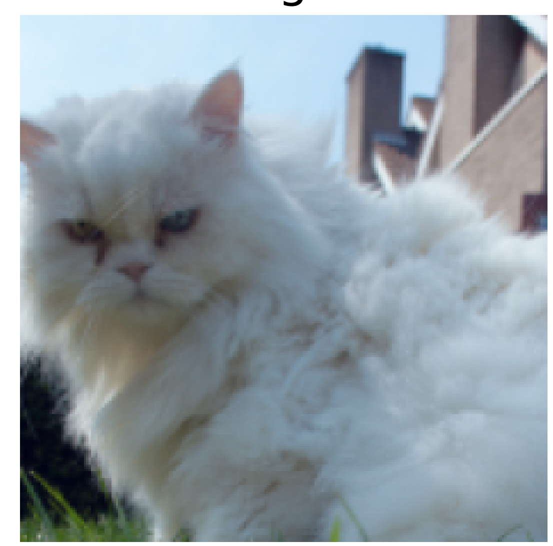
CLIP

(Contrastive Language-Image Pre-training)

CLIP models are trained on vast amounts of data



target



a close up of a hairy white cat outside

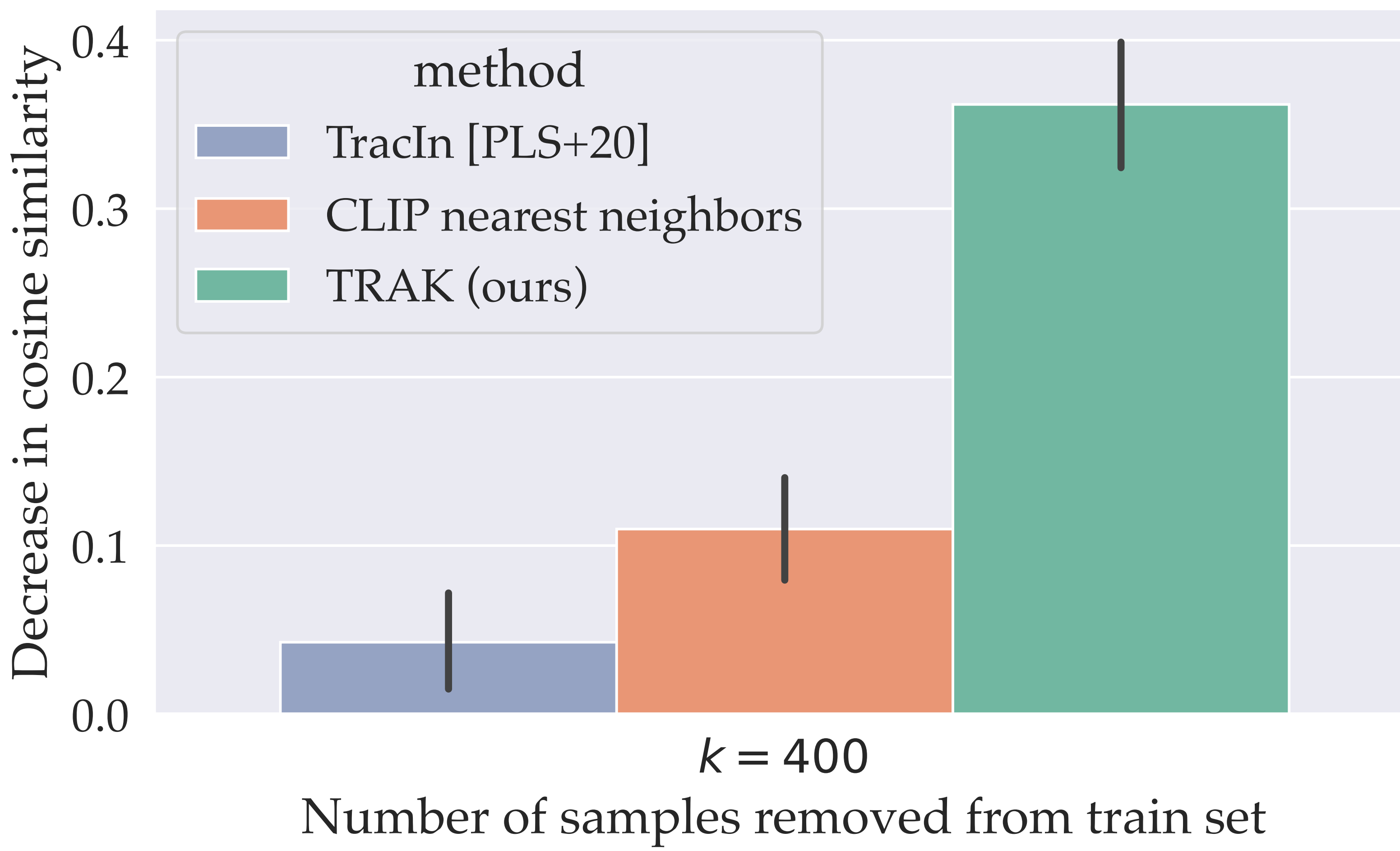
How does **training data** affect whether a given image-caption pair association is learned?

CLIP

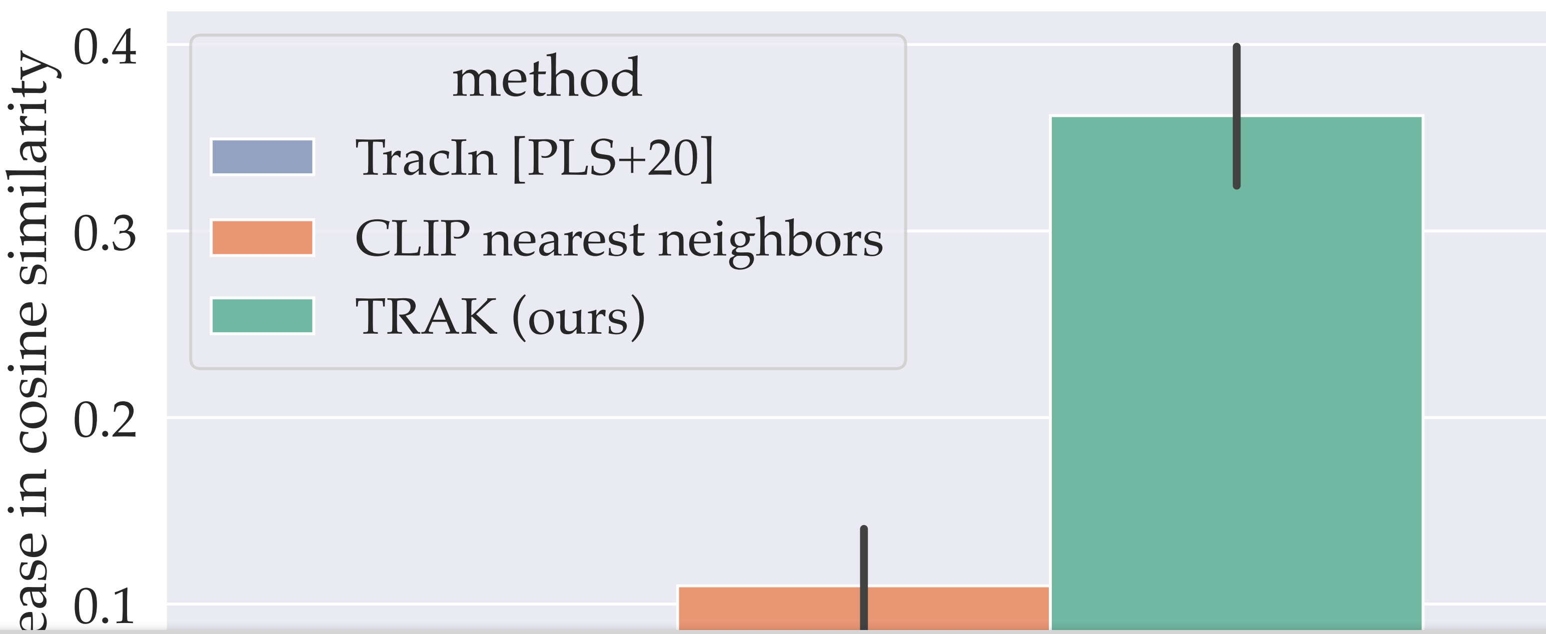

target
a close up of a hairy white cat outside
CLIP nearest neighbors
a white bear on a rock eating a carrot
this dirty sheep must have rolled in the mud
most positive influence (TRAK)
a brown long haired dog sitting outside next to a street
the white cat is laying down on top of the car
most negative influence (TRAK)
a polar bear eats a carrot on a snowy field
a yellow banana on top of a coffee cup in a microwave
little green footballs

CLIP


target
a close up of a hairy white cat outside
CLIP nearest neighbors
a white bear on a rock eating a carrot
this dirty sheep must have rolled in the mud
most positive influence (TRAK)
a brown long haired dog sitting outside next to a street
the white cat is laying down on top of the car
most negative influence (TRAK)
a polar bear eats a carrot on a snowy field
a yellow banana on top of a coffee cup in a microwave
little green footballs


method
TracIn [PLS+20]
CLIP nearest neighbors
TRAK (ours)
Decrease in cosine similarity
0.0
0.1
0.2
0.3
0.4
$k = 400$
Number of samples removed from train set

CLIP



Removing **< 0.5%** of training data makes the model much less likely **(-30%)** to align target image to correct caption

PyTorch API

```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()
```

Try it! github.com/MadryLab/trak

Takeaways

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

→ Data attribution: understanding data → model output

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

→ Data attribution: understanding data → model output

→ The challenge prior work faced: **scalability** and/or **efficacy**

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

→ Data attribution: understanding data → model output

→ The challenge prior work faced: **scalability** and/or **efficacy**

→ TRAK main idea: approximate NN with a linear model

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

→ Data attribution: understanding data → model output

→ The challenge prior work faced: **scalability** and/or **efficacy**

→ TRAK main idea: approximate NN with a linear model

→ Many applications: understanding language models, CLIP

Takeaways

TRAK: a scalable, accurate attribution method in modern settings

- → Data attribution: understanding data → model output
- → The challenge prior work faced: **scalability** and/or **efficacy**
- → TRAK main idea: approximate NN with a linear model
- → Many applications: understanding language models, CLIP

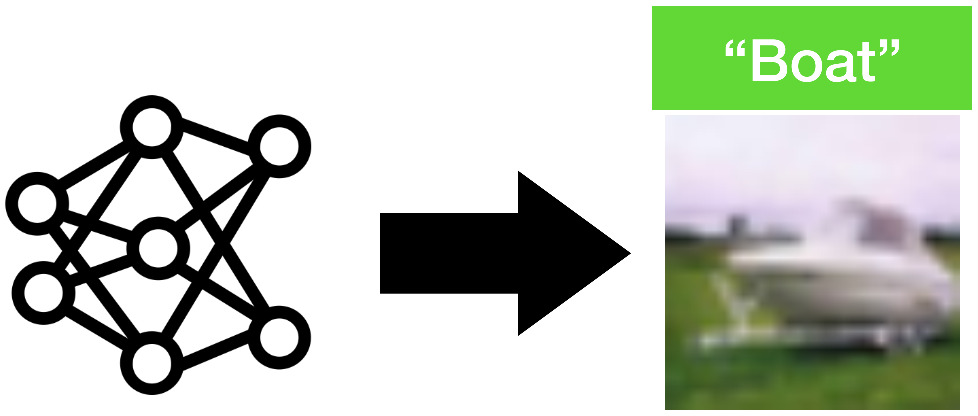
See paper for (much) more! https://arxiv.org/abs/2303.14186

@smsampark

trak.csail.mit.edu

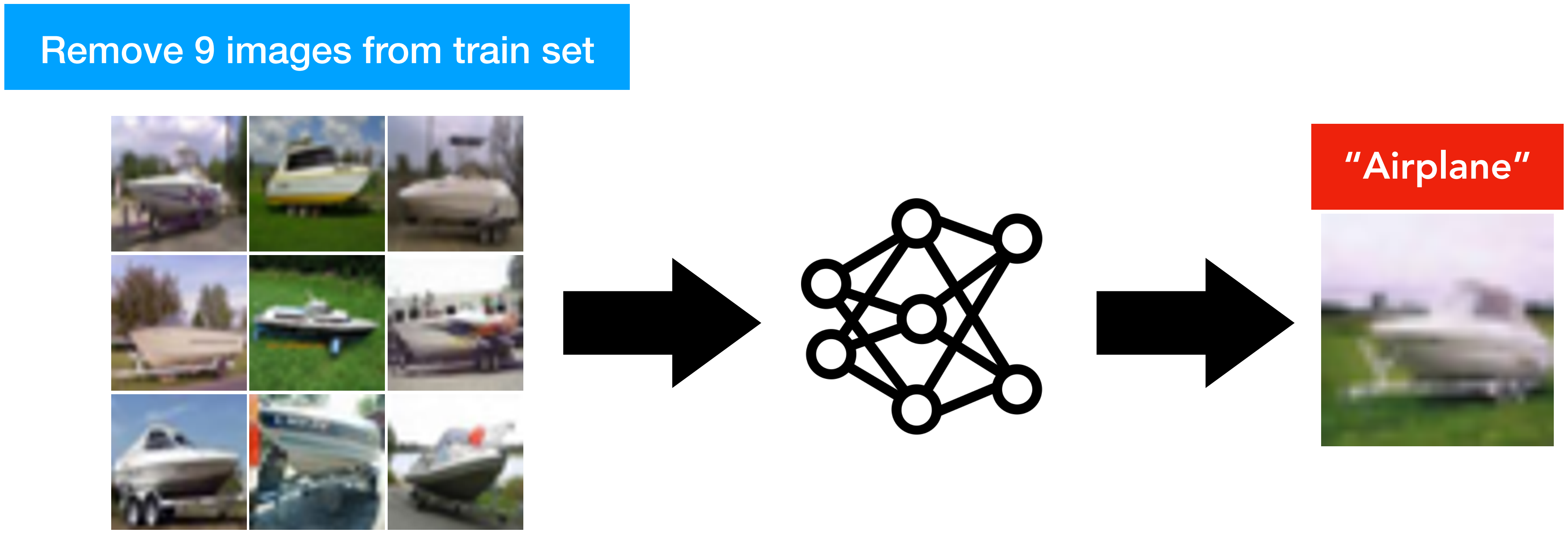
Extras

Prediction brittleness

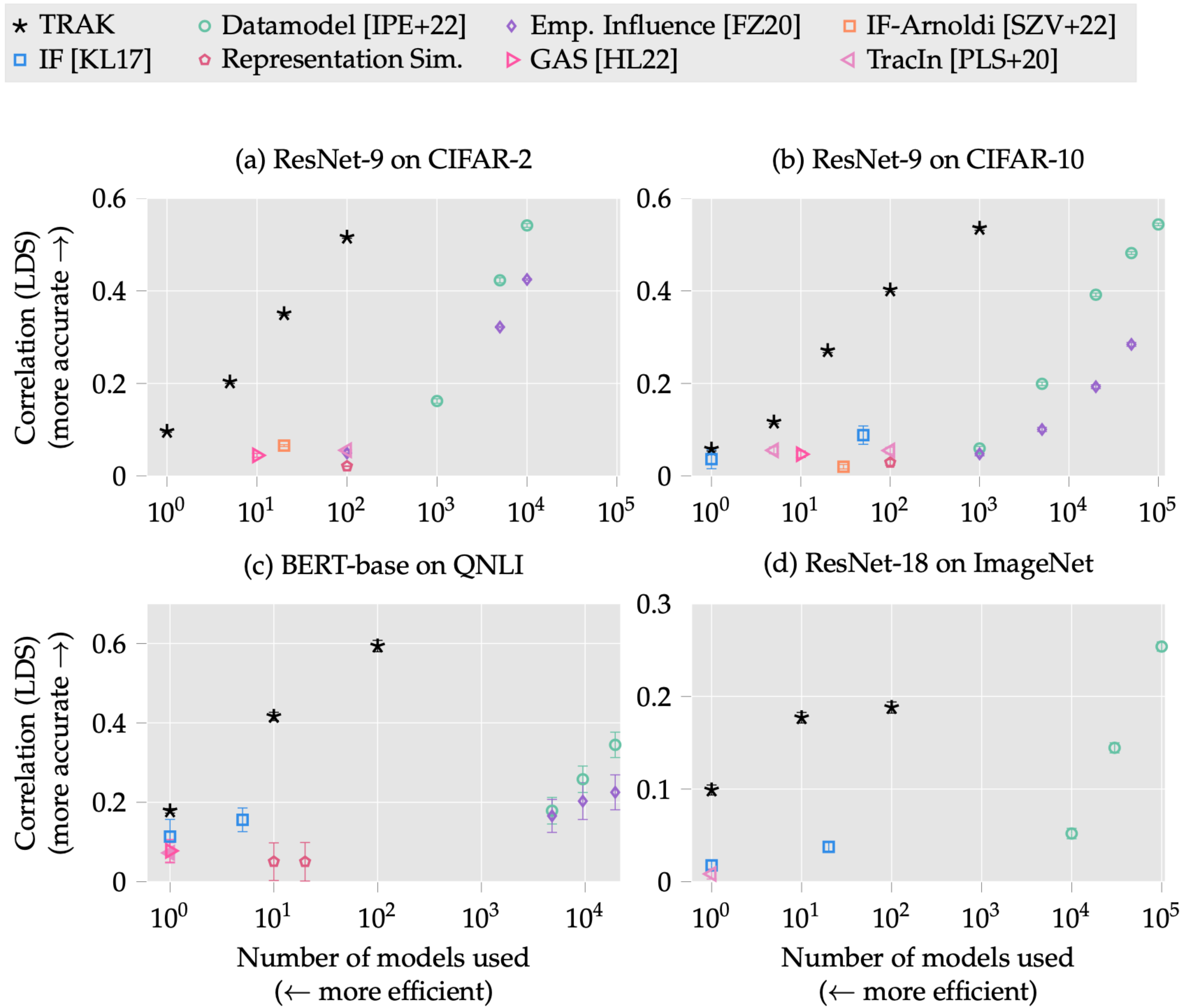


Which training examples form the
"data support" of this prediction?

Prediction brittleness



[IPE+22] 50% of CIFAR-10 test set can be misclassified by removing just 200 (< 0.4%) target-specific training images



TRAK
Datamodel [IPE+22]
Emp. Influence [FZ20]
IF-Arnoldi [SZV+22]
IF [KL17]
Representation Sim.
GAS [HL22]
TracIn [PLS+20]
(a) ResNet-9 on CIFAR-2
(b) ResNet-9 on CIFAR-10
(c) BERT-base on QNLI
(d) ResNet-18 on ImageNet
Correlation (LDS)
(more accurate →)
Number of models used
(← more efficient)

Ablations

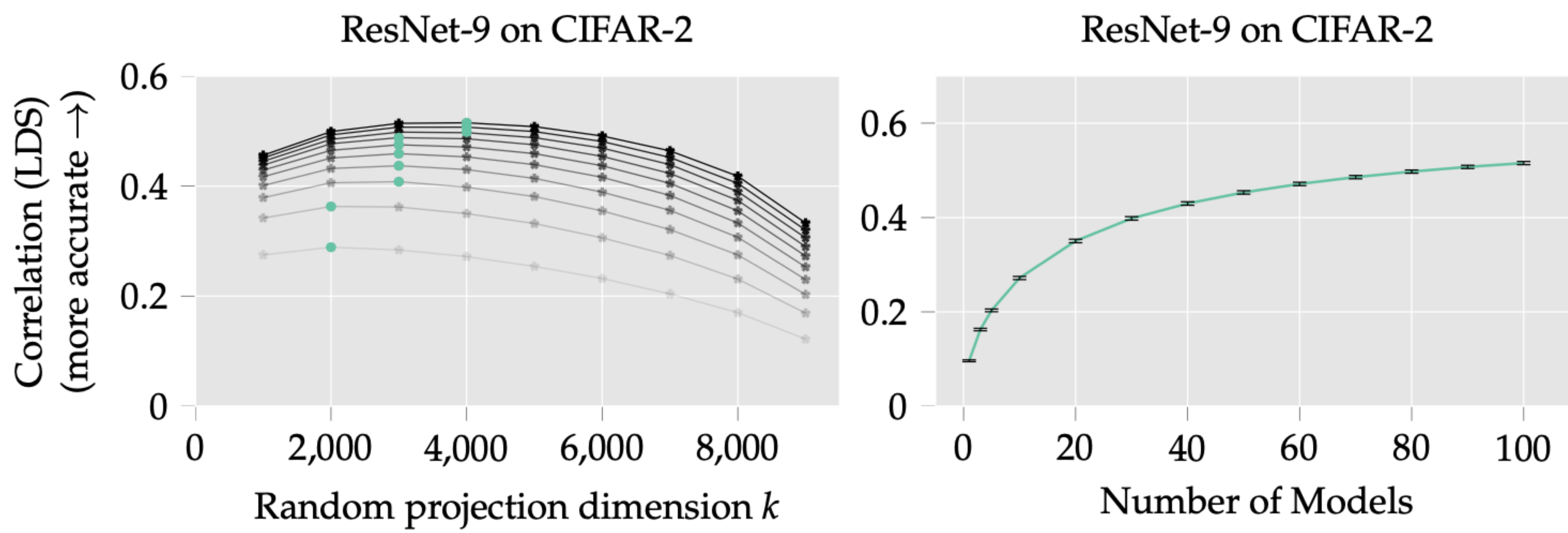


Figure E.1: **Left:** *The impact of the dimension of random projection on* TRAK*'s performance on* CIFAR-2. Each line corresponds to a different value of $M \in \{10, 20, ..., 100\}$ (the number of models TRAK is averaged over); darker lines correspond to higher M. As we increase the projected dimension, the LDS initially increases. However, beyond a certain dimension, the LDS begins to decrease. The "optimal" dimension (i.e., the peak in the above graph) increases with higher M. **Right:** *The impact of ensembling more models on* TRAK*'s performance on* CIFAR-2. The performance of TRAK as a function with the number of models used in the ensembling step. TRAK scores are computed with projection dimension of size 4000.

Ablations

# training epochs	LDS ($M = 100$)
1	0.100
5	0.204
10	0.265
15	0.293
25	0.308

Table E.2: The performance of TRAK on CIFAR-10 as a function of the epoch at which we terminate model training. In all cases, TRAK scores are computed with projection dimension $k = 1000$ and $M = 100$ independently trained models.

# independent models	LDS
5	0.329
6	0.340
10	0.350
100	0.355

Table E.3: TRAK maintains its efficacy when we use multiple checkpoints from different epochs of the same training run instead of checkpoints from independently-trained models (CIFAR-10). In all cases, $M = 100$ checkpoints and projection dimension $k = 4000$ are used to compute TRAK scores.

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: How many households has kids under the age of 18 living in them?

A: There were 158,349 households, of which 68,511 (43.3%) had children under the age of 18 living in them, 69,284 (43.8%) were opposite-sex married couples living together, 30,547 (19.3%) had a female householder with no husband present, 11,698 (7.4%) had a male householder with no wife present. (Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: How many households has kids under the age of 18 living in them?

A: There were 158,349 households, of which 68,511 (43.3%) had children under the age of 18 living in them, 69,284 (43.8%) were opposite-sex married couples living together, 30,547 (19.3%) had a female householder with no husband present, 11,698 (7.4%) had a male householder with no wife present. (Entailment)

(Most positive influence) **Q:** What percent of household have children under 18?

A: There were 46,917 households, out of which 7,835 (16.7%) had children under the age of 18 living in them, 13,092 (27.9%) were opposite-sex married couples living together, 3,510 (7.5%) had a female householder with no husband present, 1,327 (2.8%) had a male householder with no wife present. (Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: How many households has kids under the age of 18 living in them?

A: There were 158,349 households, of which 68,511 (43.3%) had children under the age of 18 living in them, 69,284 (43.8%) were opposite-sex married couples living together, 30,547 (19.3%) had a female householder with no husband present, 11,698 (7.4%) had a male householder with no wife present. (Entailment)

(Most positive influence) Q: What percent of household have children under 18?

A: There were 46,917 households, out of which 7,835 (16.7%) had children under the age of 18 living in them, 13,092 (27.9%) were opposite-sex married couples living together, 3,510 (7.5%) had a female householder with no husband present, 1,327 (2.8%) had a male householder with no wife present. (Entailment)

(Most negative influence) Q: Roughly how many same-sex couples were there?

A: There were 46,917 households, out of which 7,835 (16.7%) had children under the age of 18 living in them, 13,092 (27.9%) were opposite-sex married couples living together, 3,510 (7.5%) had a female householder with no husband present, 1,327 (2.8%) had a male householder with no wife present. (No Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: In what process is singlet oxygen usually formed?

A: Singlet oxygen is a name given to several higher-energy species of molecular O_2 in which all the electron spins are paired. (No Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: In what process is singlet oxygen usually formed?

A: Singlet oxygen is a name given to several higher-energy species of molecular O_2 in which all the electron spins are paired. (No Entailment)

(Most positive influence) Q: During what action is asphalt often reclaimed?

A: With some 95% of paved roads being constructed of or surfaced with asphalt, a substantial amount of asphalt pavement material is reclaimed each year. (No Entailment)

TRAK attributions: QNLI with BERT

(Question-answering Natural Language Inference)

Q: In what process is singlet oxygen usually formed?

A: Singlet oxygen is a name given to several higher-energy species of molecular O_2 in which all the electron spins are paired. (No Entailment)

(Most positive influence) Q: During what action is asphalt often reclaimed?

A: With some 95% of paved roads being constructed of or surfaced with asphalt, a substantial amount of asphalt pavement material is reclaimed each year. (No Entailment)

(Most negative influence) Q: Hydroelectricity accounts for what percentage of global electricity generation?

A: Hydroelectricity is the term referring to electricity generated by hydropower; the production of electrical power through the use of the gravitational force of falling or flowing water. (Entailment)