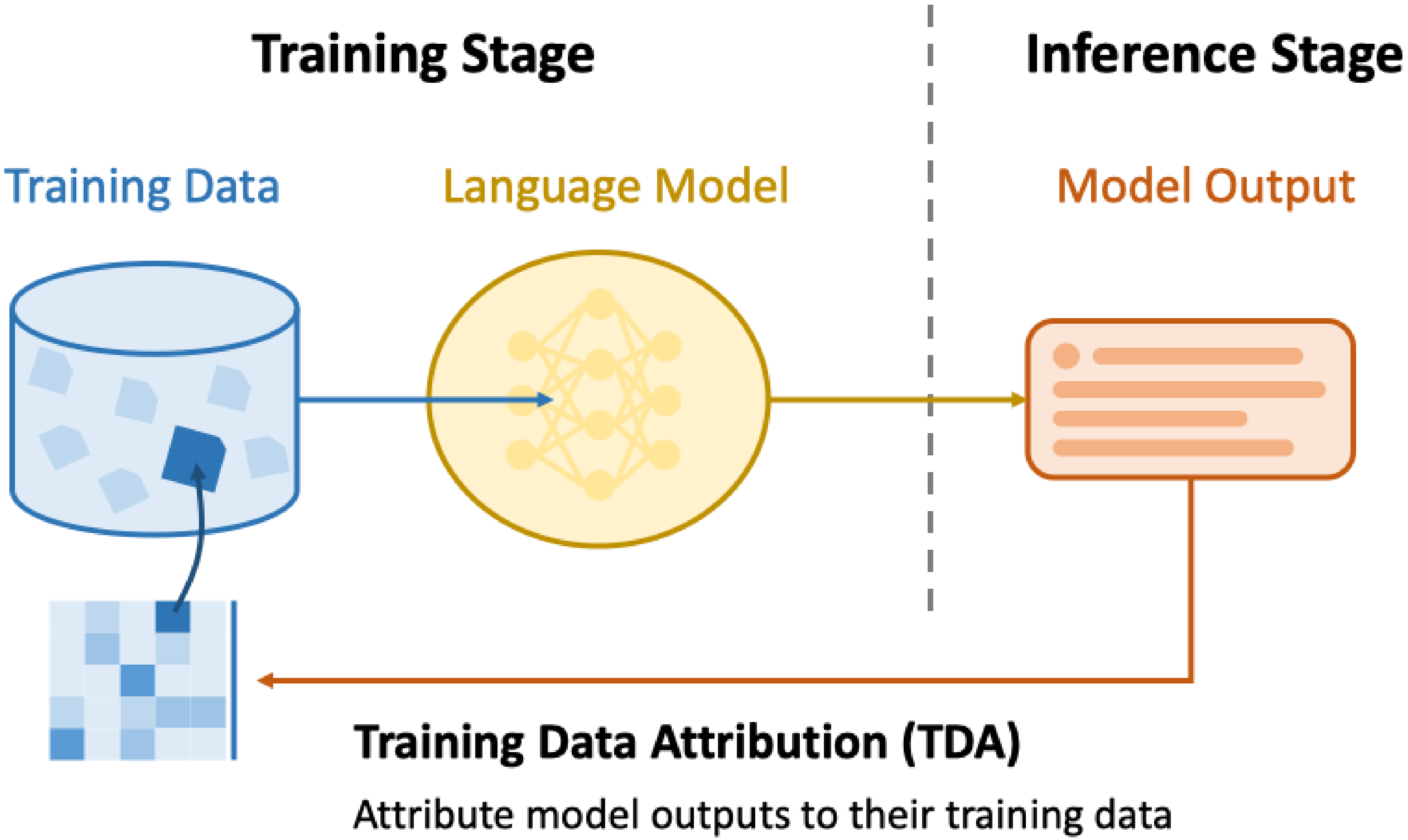
Enhancing Training Data Attribution with Representational Optimization

Weiwei Sun Haokun Liu Nikhil Kandpal Colin Raffel Yiming Yang

Carnegie Mellon University
University of Toronto & Vector Institute

Background: Training Data Attribution



Background: Training Data Attribution

The Eiffel Tower was built in **1889** and is 324 meters tall.

Mount Everest is 8,848 meters high.

The tower was completed for the 1889 World's Fair in Paris.

Gustave Eiffel's design reached about 1,000 feet in height.

Construction began in 1887 and took two years to finish.

Statue of Liberty was dedicated in 1886.

Tokyo Tower, modeled after Eiffel Tower, is 333 m tall.

At over 300 meters, it remained the world's tallest structure for decades.

The Eiffel Tower was repainted in 2019.

Background: Training Data Attribution

Training Data S

LM θ

Train the LM

$$\theta^* = \arg\min_{\theta} \sum_{z_i \in S} \ell(z_i; \theta),$$

For the model output during inference, x

Actual Outcome

$$r(x, S) = \ell(x; \theta^*)$$

$$\updownarrow$$

TDA Method

$$f(x, S)$$

Gradient-based TDA

Influence Function

$$f_{\mathrm{IF}}(x, z_i) = \underbrace{-\nabla_\theta \ell(x;\theta)^\top}_{\text{Test Gradient}} \underbrace{\mathbf{H}^{-1} \nabla_\theta \ell(z_i;\theta)}_{\text{Training Gradient (curvature-corrected)}}.$$

Group Influence

$$f_{\mathrm{IF}}(x, S) = \nabla_\theta \ell(x, \theta^*)^\top \mathbf{H}^{-1} \boxed{\sum_{z_i \in S}} \nabla_\theta \ell(z_i, \theta^*)$$

Understanding black-box predictions via influence functions

Gradient-based TDA

Influence Function

$$f_{\mathrm{IF}}(x, z_i) = -\nabla_\theta \ell(x;\theta)^\top \, \mathbf{H}^{-1} \, \nabla_\theta \ell(z_i;\theta).$$

Test Gradient

Training Gradient (curvature-adjusted)

Speed:

- Calculate inverse hessian
- Calculate gradient

Storage:

- Store full gradient of each training point

Understanding black-box predictions via influence functions

Gradient-based TDA

Efficient Gradient-based TDA

$$f_{\mathrm{GD}}(x, z_i) = \phi(x)^\top \cdot \phi(z_i)$$

$$\phi(z) = \mathrm{norm}\left[\mathbf{H}_{\hat{\theta}}^{-\frac{1}{2}} \nabla_{\hat{\theta}} \ell(z; \theta)\right]_2$$

Hessian approximation [1] (e.g., FIM)

Gradient projection [2] (e.g., Lora)

- [1] Studying large language model generalization with influence functions
- [2] What is your data worth to gpt? llm-scale data valuation with influence functions

Speed:

- ~~Calculate inverse hessian~~
- **Calculate gradient**

Storage:

- ~~Store full gradient of each training point~~

+ Tradeoff between efficiency and fidelity

Representation-based TDA

Alternative: Text Representation

$$f_{\text{Rep}}(x, z_i) = \text{Enc}(x)^\top \cdot \text{Enc}(z_i)$$

- TF-IDF
- N-Gram
- Hidden States
- Text Embedding

Speed:

✓ High Speed

Storage:

✓ Storage Efficient

Data selection for language models via importance resampling

Representation-based TDA

Text Representation

$$f_{\text{Rep}}(x, z_i) = \text{Enc}(x)^\top \cdot \text{Enc}(z_i)$$

- TF-IDF
- N-Gram
- Hidden States
- Text Embedding

None of them are designed for TDA -> Low fidelity!

Speed:

✓ High Speed

Storage:

✓ Storage Efficient

Fidelity:

- **Low**

Data selection for language models via importance resampling

Our Method: AirRep

Attentive **I**nfluence **R**anking **Rep**resentation

$$f_{\text{AirRep}}(x, S) = \text{Enc}(x)^\top \cdot \text{Agg}(\text{Enc}(z_i) \mid z_i \in S)$$

Optimize Representation for TDA

Speed:

✓ High Speed

Storage:

✓ Storage Efficient

Fidelity:

✓ High

AirRep: Model

$$f_{\text{AirRep}}(x, S) = \text{Enc}(x)^\top \cdot \text{Agg}(\text{Enc}(z_i) \mid z_i \in S)$$

- **Text Embedding Model**

Input text -> Output a vector
Trainable

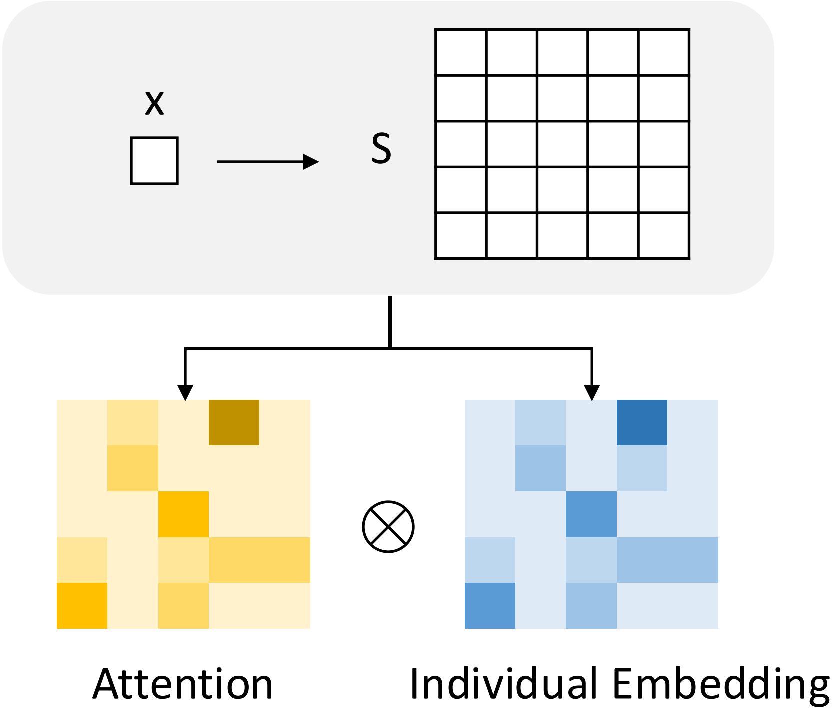
AirRep: Model

$$f_{\text{AirRep}}(x, S) = \text{Enc}(x)^\top \cdot \text{Agg}(\text{Enc}(z_i) \mid z_i \in S)$$

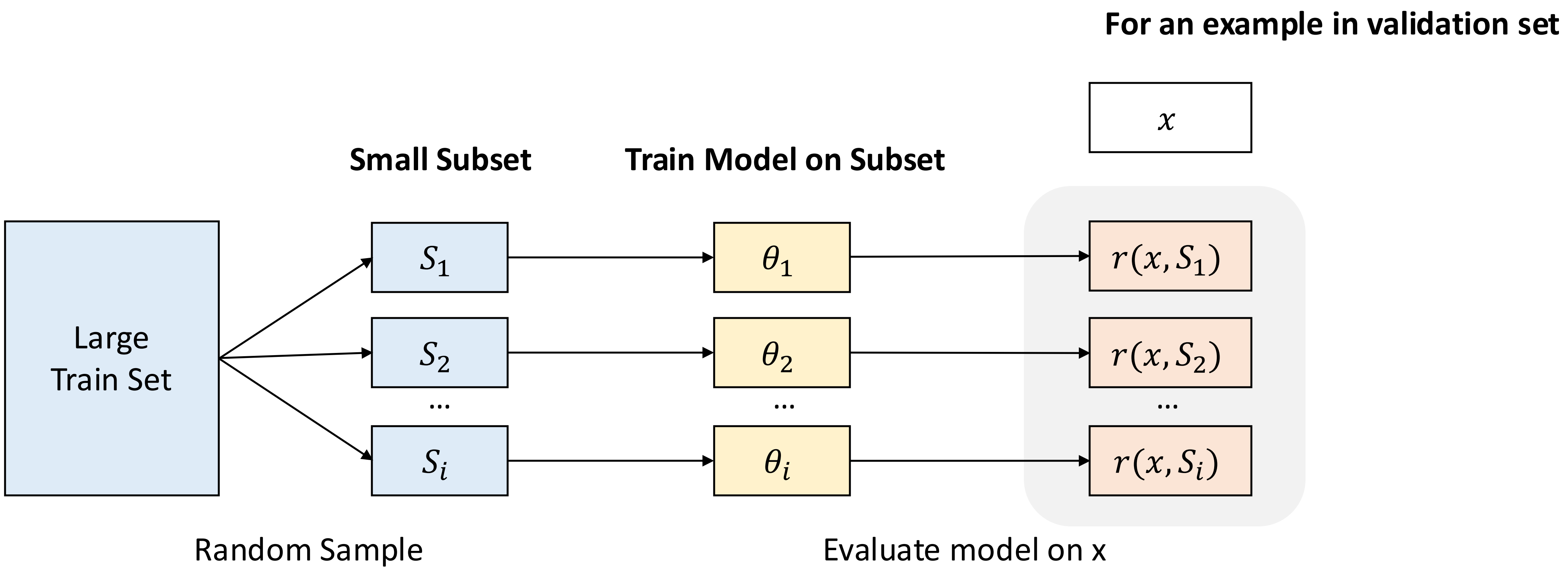
➢ **Attention-based Pooling**

$$f_{\text{AirRep}}(x, S) = \text{Enc}(x)^\top \cdot \sum_{i=1}^{n} \alpha_i \, \text{Enc}(z_i),$$

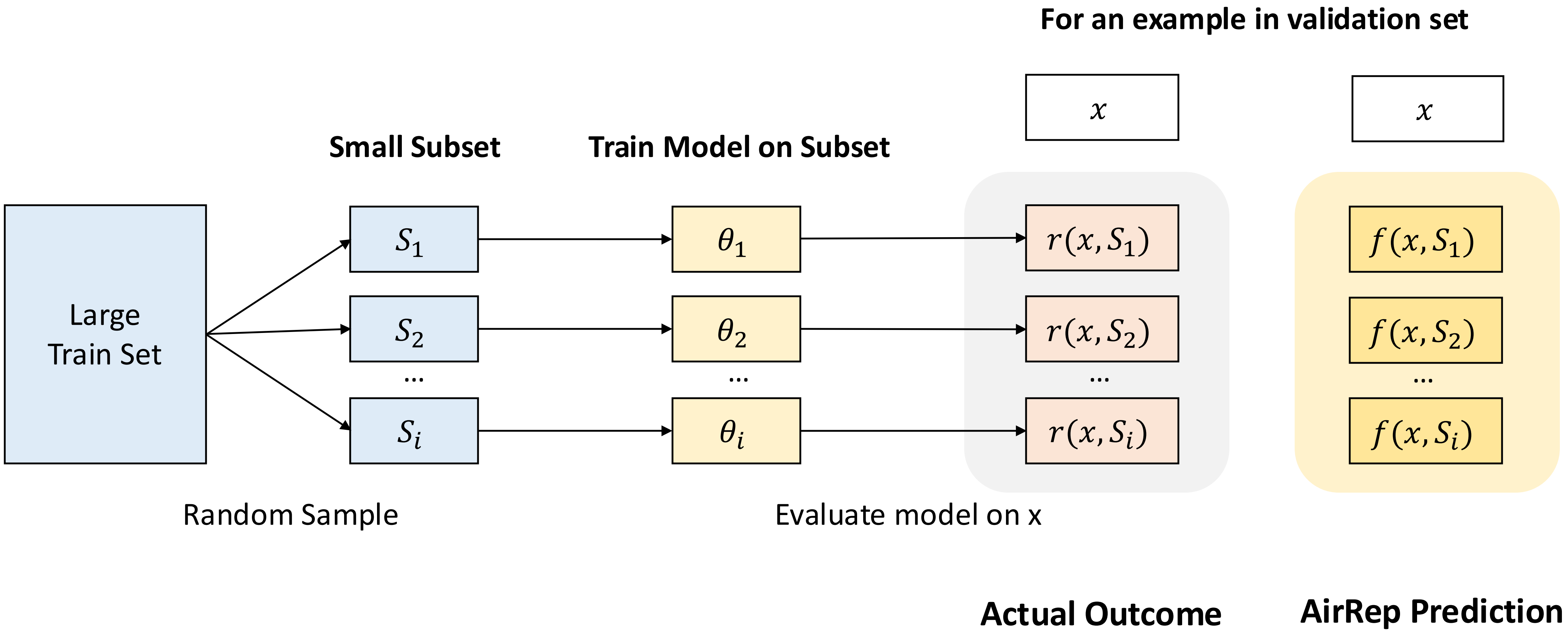
$$\text{where} \quad \alpha_i = \frac{\exp(|\,\text{Enc}(x)^\top \cdot \text{Enc}(z_i)|)}{\sum_{j \in [n]} \exp(|\,\text{Enc}(x)^\top \cdot \text{Enc}(z_i)|)}.$$



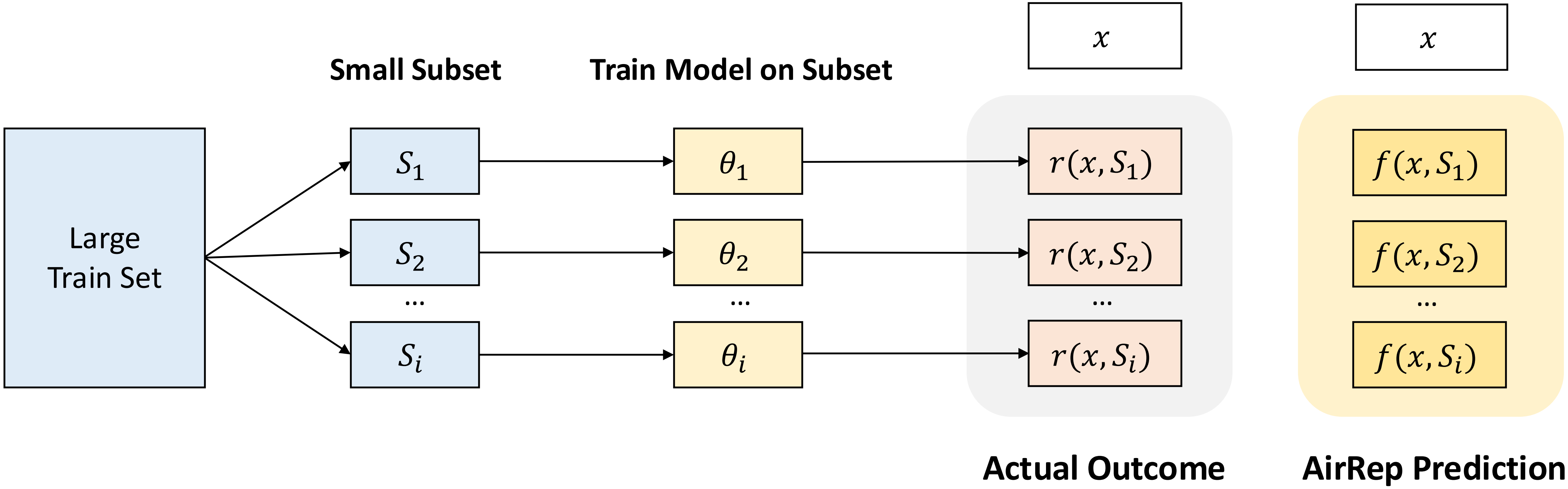
AirRep: Optimization



AirRep: Optimization



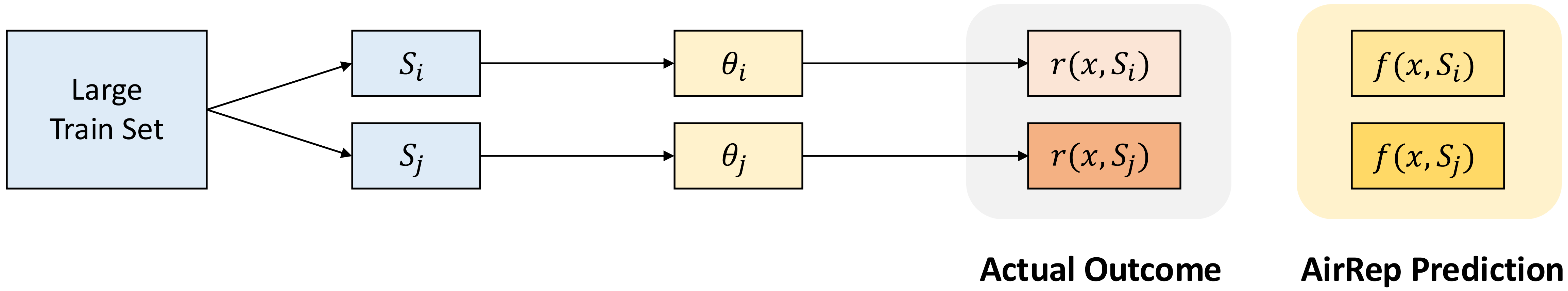
AirRep: Optimization



Pairwise Ranking Loss

$$\mathcal{L}(x, \mathcal{S}) = -\sum_{i,j \in M} \mathbb{1}_{r_i > r_j} \, w_{i,j} \, \log \sigma(f_i - f_j),$$

AirRep: Optimization

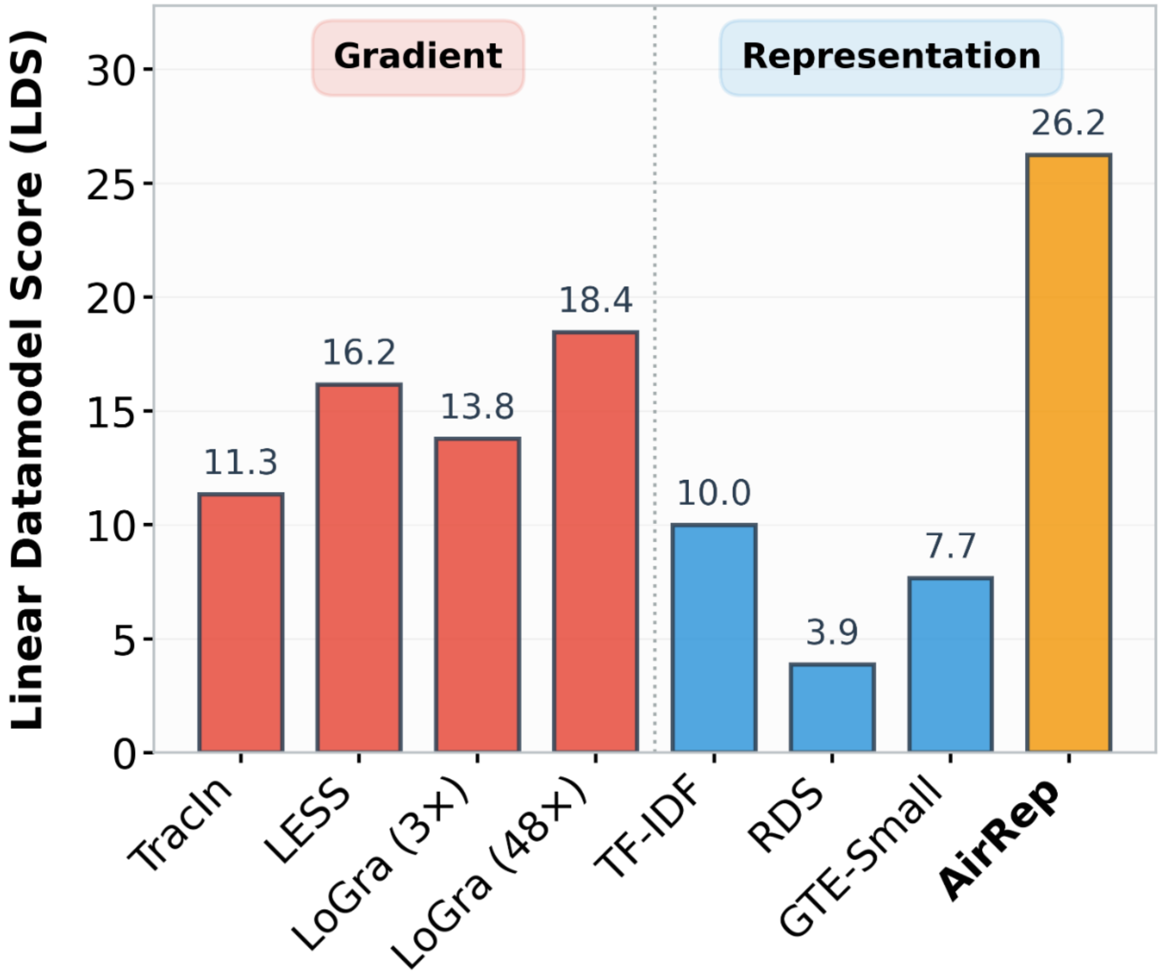


Weighted pairwise ranking loss

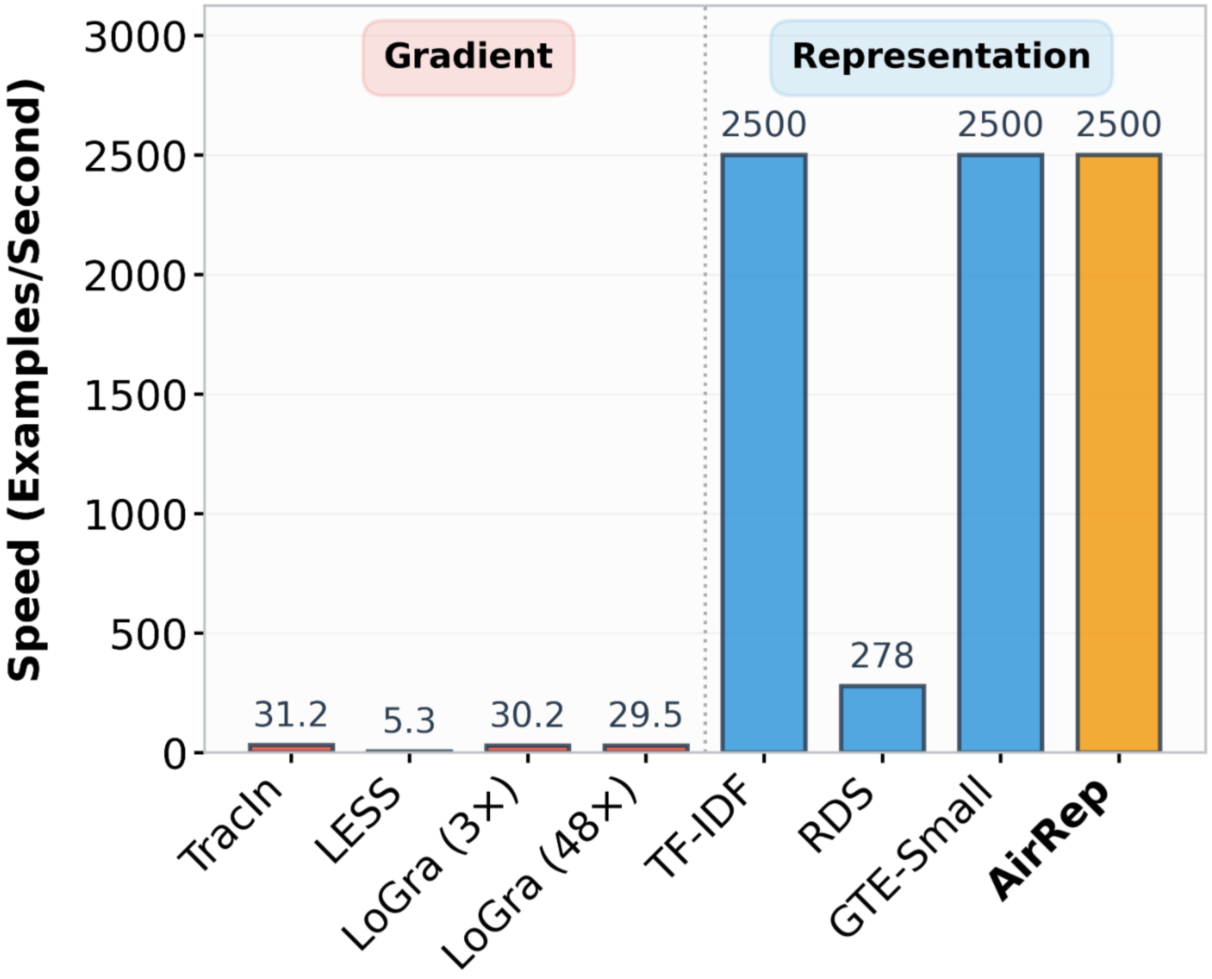
$$\mathcal{L}(x, \mathcal{S}) = - \sum_{i,j \in M} \mathbb{1}_{r_i > r_j} \, w_{i,j} \, \log \sigma(f_i - f_j),$$

Simply put, if $r(x, S_j) > r(x, S_i)$ we want $f(x, S_j) > f(x, S_i)$

Results

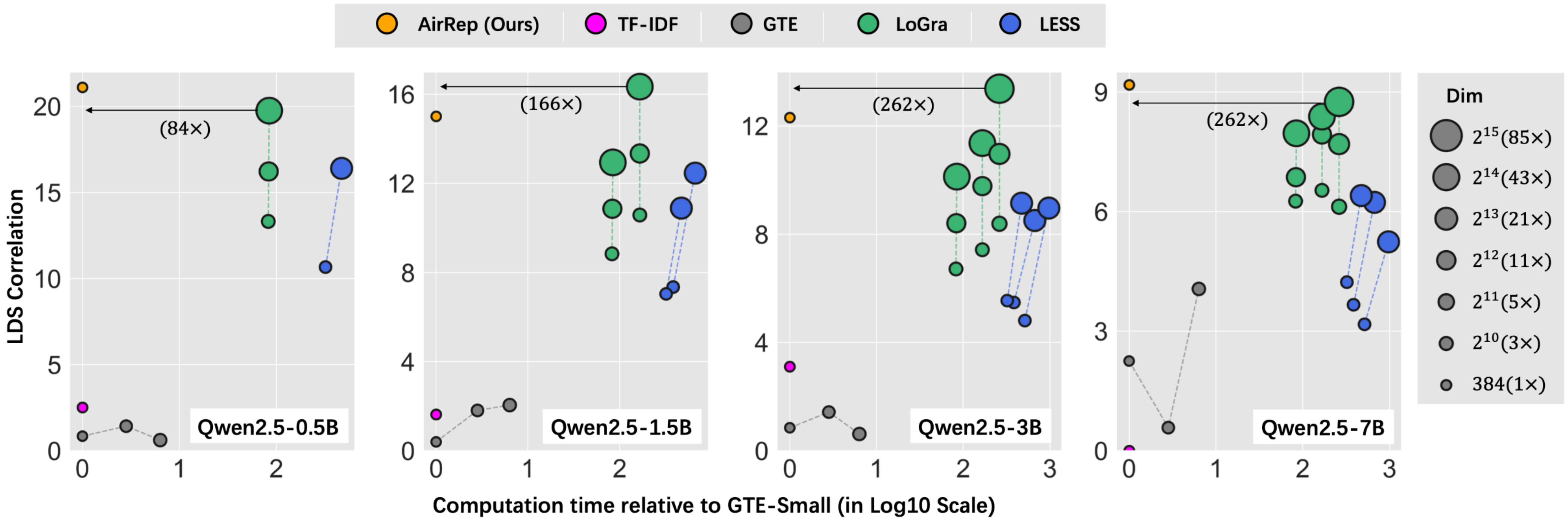


Fidelity



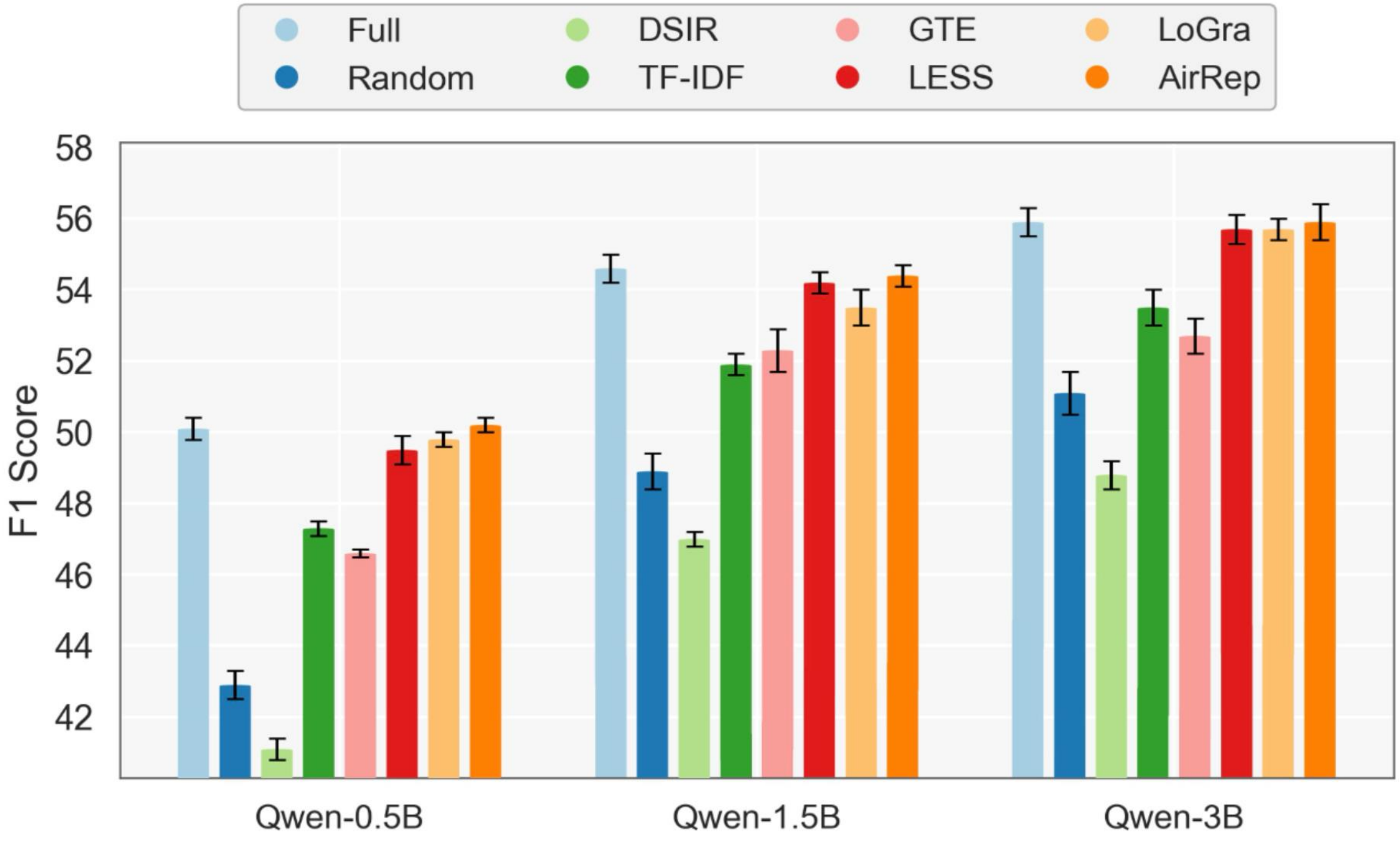
Speed

Results



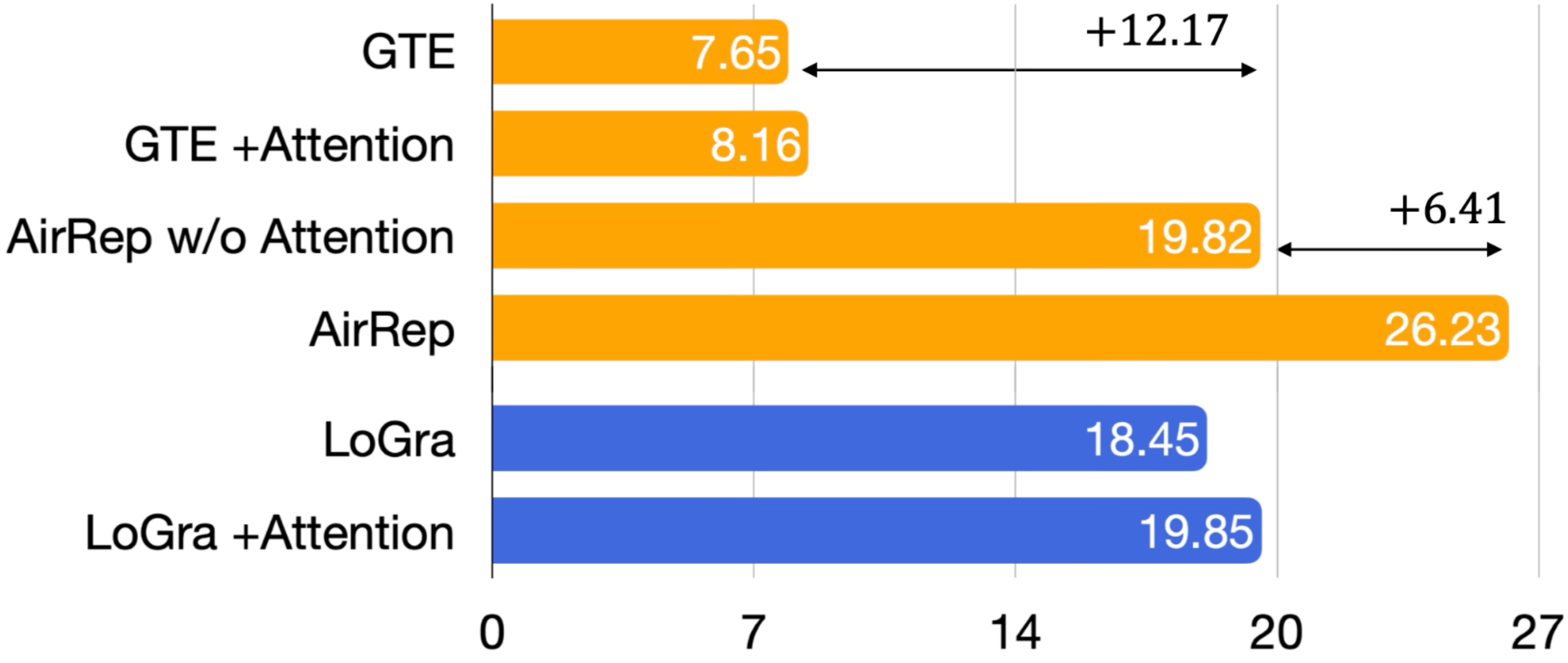
Fidelity vs. Speed Trade-off

Results



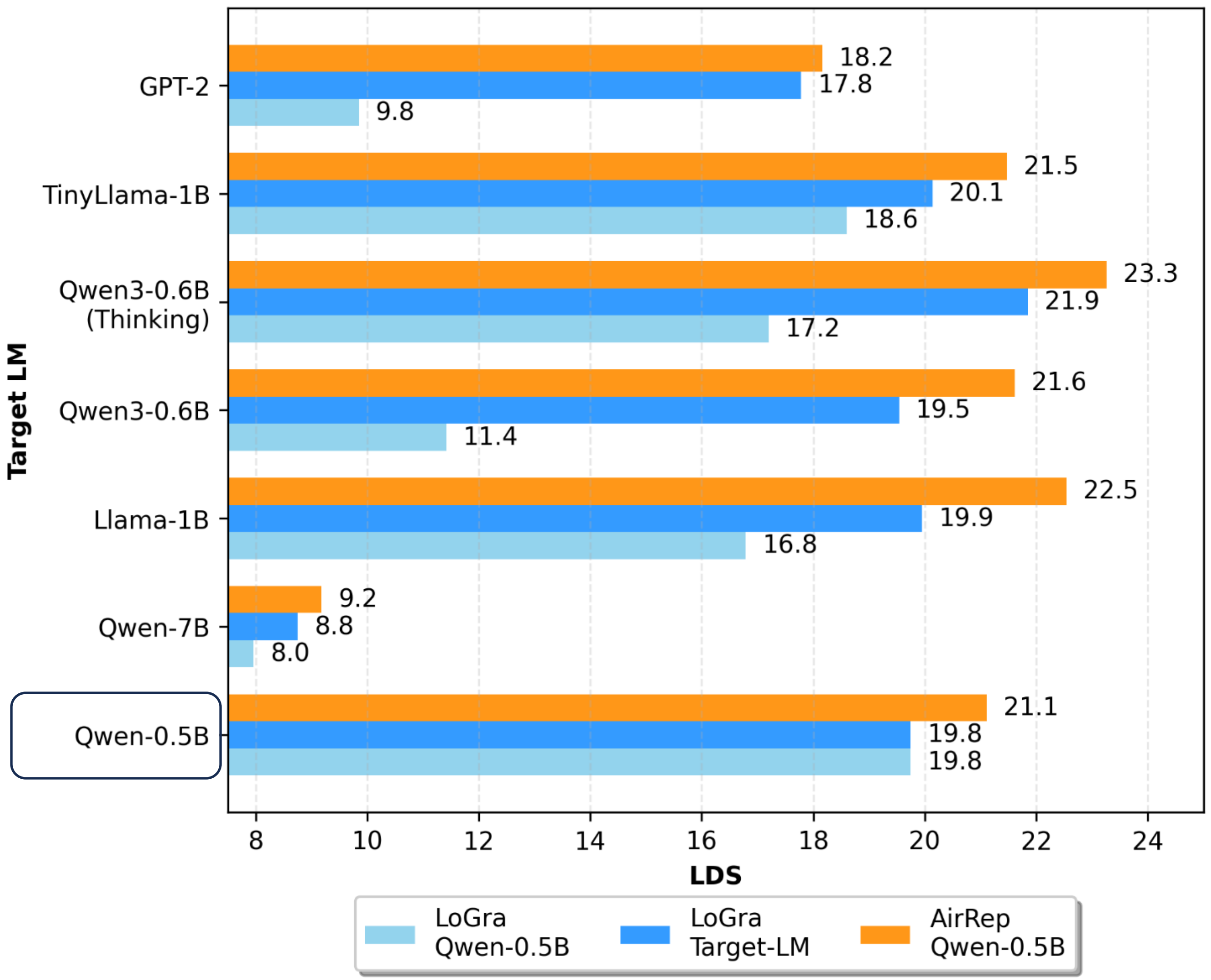
Data Selection Evaluation

Results



Ablation Study

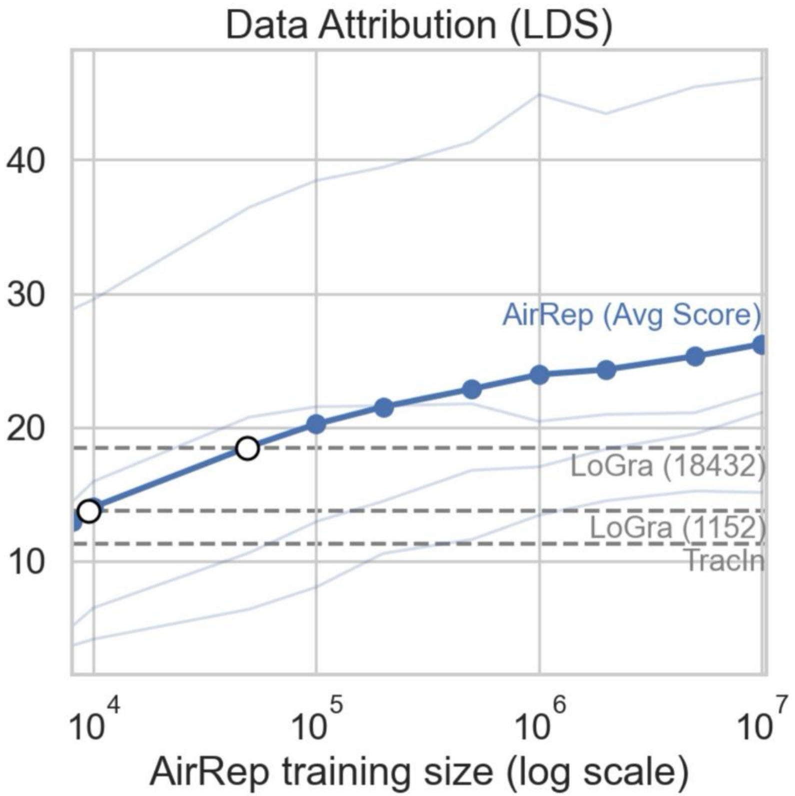
Results

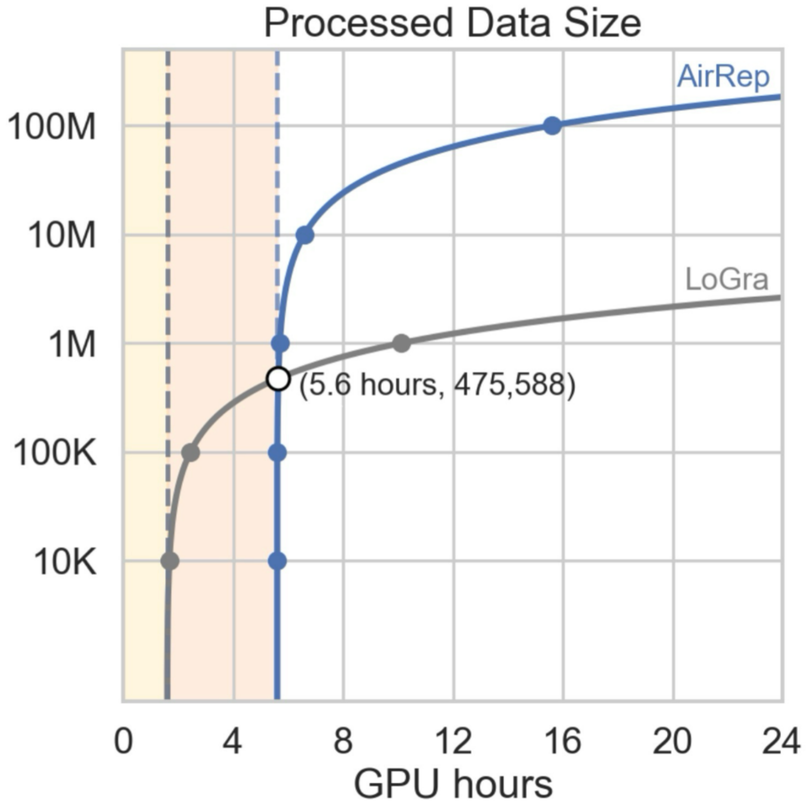


Generalizability

Training signals are generated from Qwen-0.5B and evaluated on larger or different language models.

Results





Amortizing AirRep Training Cost

NEURAL INFORMATION PROCESSING SYSTEMS

Thank You

For Your Attention

Code: https://github.com/sunnweiwei/AirRep
ArXiv: https://arxiv.org/pdf/2505.18513