Taming Hyperparameter Sensitivity in Data Attribution: Practical Selection Without Costly Retraining

Weiyi Wang, Junwei Deng, Yuzheng Hu, Shiyuan Zhang,
Xirui Jiang, Runting Zhang, Han Zhao, Jiaqi W. Ma

NeurIPS 2025

Background: Hyperparameters (HPs) in Data Attribution

Influence Function (IF) (with gradient projection): Base of popular attributors

$$\underbrace{\tau_{\mathrm{IF},\lambda,P}(z', z_i)}_{\substack{\text{Influence of train data } z_i \\ \text{on test example } z'}} = \underbrace{-[\nabla_\theta f_{z'}]_P}_{\substack{\text{How much} \\ \text{will target of } z' \\ \text{degrade if ...}}} \underbrace{\left([H_S]_P + \lambda\, I\right)^{-1} \left[\nabla_\theta L_{z_i}\right]_P}_{\substack{\text{... parameters are retrained} \\ \text{after removing } z_i?}}$$

- attributors like TRAK, LoGra, ... build upon IF
- λ: regularization strength
- $\tilde{p}$: projection dimension; $[\cdot]_P$: projection with $P \in \mathbb{R}^{p \times \tilde{p}}$
- other hyperparameters...?

f: target (e.g. logit)

H_S: Hessian of loss of model trained on S

L: loss

Evaluation: How good is the attributor?

- Linear Datamodeling Score (LDS): Generic, task-independent
- Downstream tasks: Data selection, Fact tracing, Adversarial attack, ...

Koh, P. W. & Liang, P. (2017). *Understanding black-box predictions via influence functions*. *ICML 2017*, 1885–1894.
Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., & Madry, A. (2023). *TRAK: Attributing model behavior at scale*. *ICML 2023*, 27074–27113.
Choe, S. K. et al. (2024). *What is your data worth to GPT? LLM-scale data valuation with influence functions*. *arXiv:2405.13954*.

Motivations

1. How do hyperparameters affect data attribution performance?

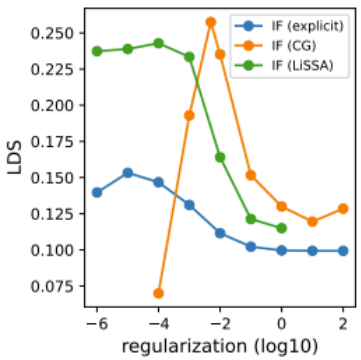
- A Large-Scale Study of Hyperparameter Sensitivity in Data Attribution **[Section 3]**

2. Can we accelerate hyperparameter selection (faster than brute-force search) without sacrificing performance?

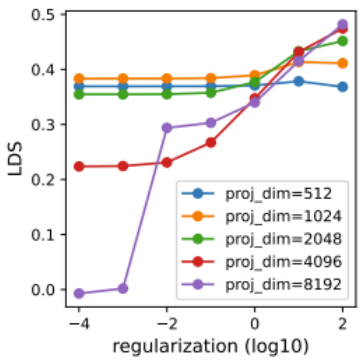
- A Case Study on Regularization Term in Influence Function **[Section 4]**

A Large-Scale Study of HP Sensitivity

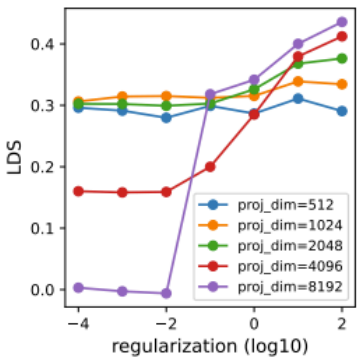
Part of the experimental results:



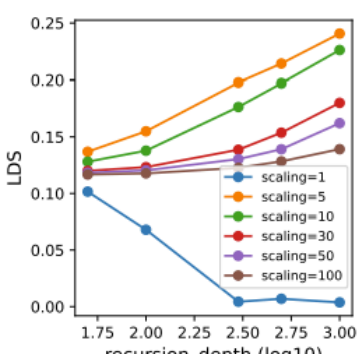
(a) MNIST+MLP. Attributor: IF[5]. HP: regularization.



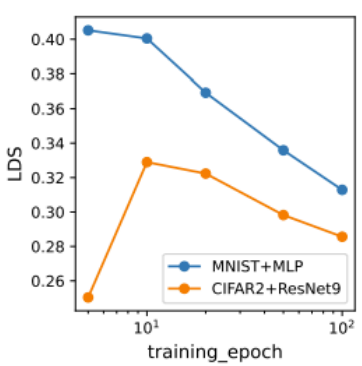
(b) MNIST+MLP. Attributor: TRAK. HP: projection-dimension, regularization.



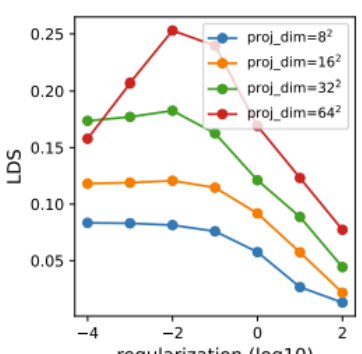
(c) CIFAR-2+ResNet-9. Attributor: TRAK. HP: projection-dimension, regularization.



(d) MNIST+MLP. Attributor: IF (LiSSA). HP: scaling, recursion-depth.



(e) CIFAR-2+ResNet-9 & MNIST+MLP. Attributor: TRAK. HP: training-epoch.



(f) WikiText2+GPT2. Attributor: LoGra. HP: projection-dimension, regularization.

Key Takeaways

1. **Most attributors are sensitive to certain HPs**
   - HPs in data attribution are critical but largely overlooked
2. **Impact of HPs can be "entangled"**
   - Counterintuitive: increase $\tilde{p}$ along $\Rightarrow$ performance could $\downarrow$
   - Increase $\tilde{p}$ + tune $\lambda \Rightarrow$ monotonic performance $\uparrow$
3. **Implicit HPs (e.g. "training epoch") are important as well**
   - Training epoch: #epochs to train the θ used in τ
   - Could greatly affect performance

Accelerating HPs Selection

Brute-force HP selection procedure

$$\text{Attributor} \rightarrow \text{Downstream tasks (e.g. data selection)}$$

- Each $(\text{attributor}, \text{hyperparameters}, \text{task}) \Rightarrow$ Re-evaluate (slow)

LDS-based HP selection procedure

$$\text{Attributor} \rightarrow \text{LDS} \rightarrow \text{Downstream tasks}$$

- Good LDS score $\Rightarrow$ likely good downstream performance
- Each $(\text{attributor}, \text{hyperparameters}) \Rightarrow$ LDS
- Issue: LDS itself is slow! It requires *retraining many models on sampled subsets of* S

Can we select HPs without retraining in LDS?

A Case Study on Regularization Term in Influence Function

Can we select HPs without retraining in LDS?

Problem: Find the maximizer λ of LDS without retraining.

- Issue: LDS is (1)*non-differentiable*, and (2)*dependent of sampled subsets*.
- "Population Pearson LDS"

A sufficient condition (Informal, Under certain assumptions)

> For z' and λ, there are two numbers $\xi_{z',\lambda}$ and $\omega_{z',\lambda}$ such that:
> 1. Both lie in $[0, 1]$;
> 2. Computing $\xi_{z',\lambda}$ doesn't require retraining models;
> 3. If $\xi_{z',\lambda} < \omega_{z',\lambda}$, then Population Pearson LDS of z' increases with λ.

A Case Study on Regularization Term in Influence Function

$$\text{Attributor} \rightarrow \text{Surrogate indicator } \xi_{z',\lambda} \rightarrow \text{Downstream tasks}$$

- Each $(\text{attributor}, \text{hyperparameters}) \Rightarrow \xi$
- No retraining needed!

Algorithm 1 Selecting λ with the surrogate indicator.

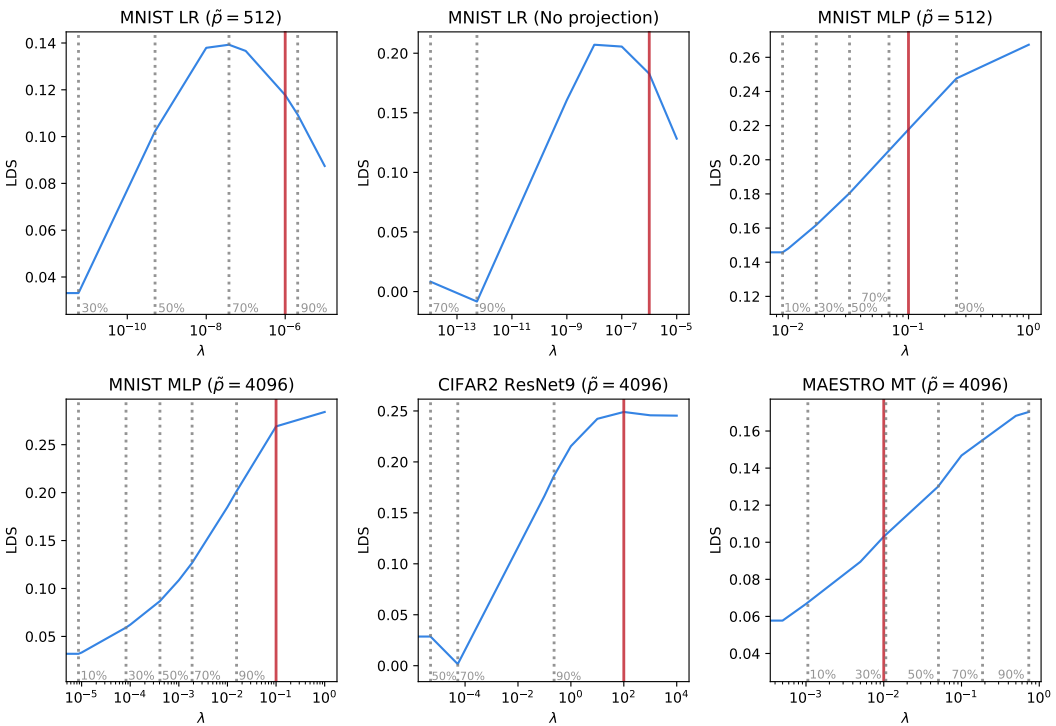
Input: A candidate set $\mathcal{C}$ of λ, a subset $T \subset \mathcal{Z}$ of test examples.
Output: A selected $\hat{\lambda}$.

for $\lambda \in \mathcal{C}$ **do**
  Compute $\xi_{z',\lambda}$ for all $z' \in T$;
  $\bar{\xi}_{T,\lambda} \leftarrow \frac{1}{|T|} \sum_{z' \in T} \xi_{z',\lambda}$;
end for
$\hat{\lambda} \leftarrow \arg\min_{\lambda \in \mathcal{C}} |\bar{\xi}_{T,\lambda} - 0.5|$;

0.5 intuitively marks quick increase in ξ

$\Rightarrow$ Population Pearson LDS approaches to stationary point

A Case Study on Regularization Term in Influence Function

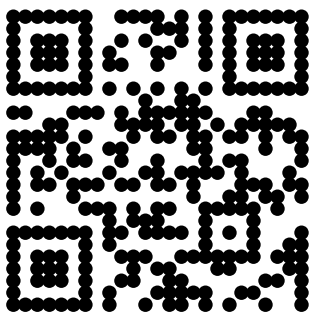


Removal rates	10%	30%	50%
Full	$88.63\% \pm 0.03\%$	$88.63\% \pm 0.03\%$	$88.63\% \pm 0.03\%$
Random	$88.52\% \pm 0.03\%$	$88.06\% \pm 0.05\%$	$87.60\% \pm 0.06\%$
IFFIM Default	$88.44\% \pm 0.05\%$	$87.77\% \pm 0.08\%$	$87.58\% \pm 0.10\%$
IFFIM Selected	$87.66\% \pm 0.04\%$	$84.50\% \pm 0.04\%$	$81.53\% \pm 0.06\%$
TRAK Default	$88.53\% \pm 0.06\%$	$87.92\% \pm 0.11\%$	$86.88\% \pm 0.29\%$
TRAK Selected	$\mathbf{87.30\% \pm 0.04\%}$	$\mathbf{83.84\% \pm 0.05\%}$	$\mathbf{80.12\% \pm 0.08\%}$

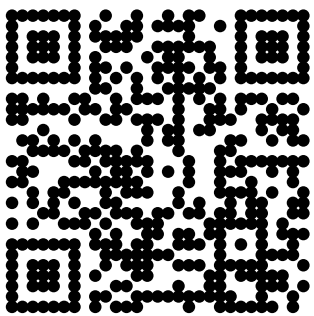
- Left: LDS;
  red $\Rightarrow$ our algo.
  blue $\Rightarrow$ LDS vs. λ
- Top: Data selection;
  "Selected" $\Rightarrow$ our algo.

It is possible to select λ properly without any retraining

arXiv



GitHub



Thank you!