If Influence Functions are the Answer, Then What is the Question?

Juhan Bae[1,2], Nathan Ng[1,2,3], Alston Lo[1,2], Marzyeh Ghassemi[3], Roger Grosse[1,2]
[1]University of Toronto, [2]Vector Institute, [3]Massachusetts Institute of Technology

Influence Functions

- The influence function is a classic technique from robust statistics that estimates the effect of deleting a single data example (or a group of data examples) from a training dataset.
- More formally, influence functions approximate the optimal parameters with a data point $\mathbf{z} = (\mathbf{x}, \mathbf{t})$ removed with:

$$\boldsymbol{\theta}^{\star}_{\text{removed}} \approx \boldsymbol{\theta}^{\star} + \frac{1}{N}(\nabla^2_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}^{\star}) + \lambda \mathbf{I})^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(f(\boldsymbol{\theta}^{\star}, \mathbf{x}), \mathbf{t}),$$

where $\boldsymbol{\theta}^{\star}$ is the optimal parameters trained on the full dataset and λ is a damping term to ensure invertibility.
- When the training objective is strongly convex (e.g., as in logistic regression with L2 regularization), influence functions are expected to align well with leave-one-out (LOO) or leave-k-out retraining.

Influence Estimation in Neural Networks

- However, influence functions in neural networks often do not accurately predict the effect of retraining the model without a data point.
- Therefore, previous error analyses concluded that influence estimations for neural networks are often "fragile" and "erroneous".
- In this work, we decompose several factors responsible for the mismatch between influence functions and LOO retraining.

1. Warm-Start Gap

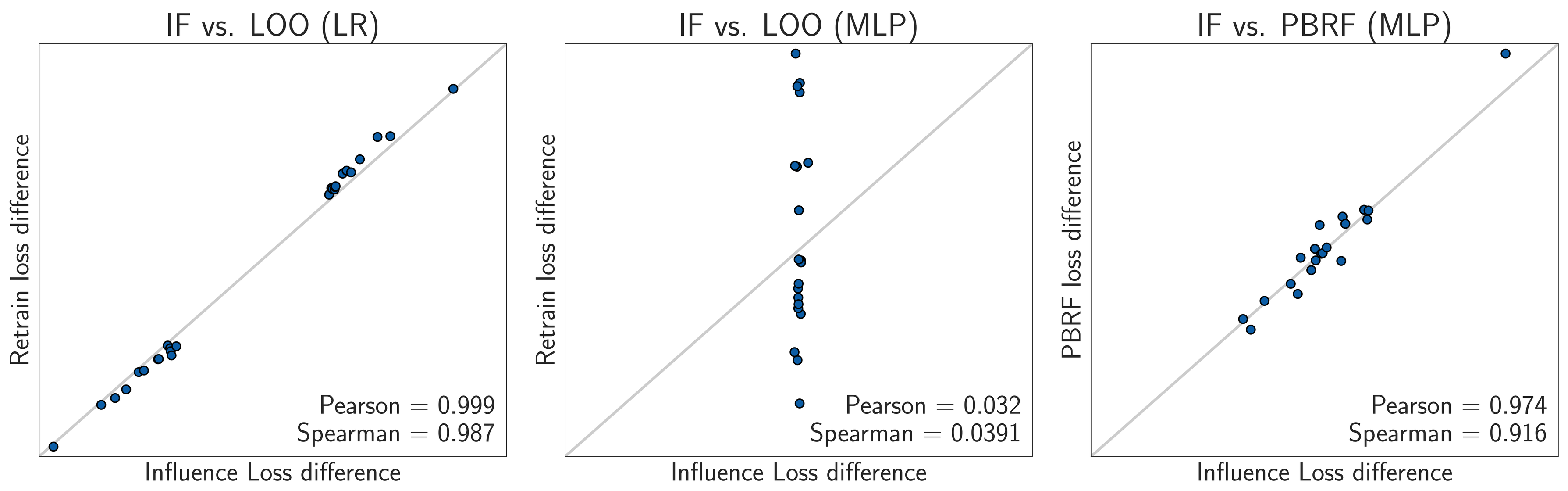
- Influence functions approximate the effect of removing a data point $\mathbf{z}$ at a local neighbourhood of the optimum $\boldsymbol{\theta}^{\star}$.
- Hence, influence approximation has a more natural connection to the retraining scheme that initializes the network at the current optimum $\boldsymbol{\theta}^{\star}$ (warm-start retraining) than the scheme that initializes the network randomly (cold-start retraining).
- For neural nets, warm-start optimum $\neq$ cold-start optimum.

2. Proximity Gap

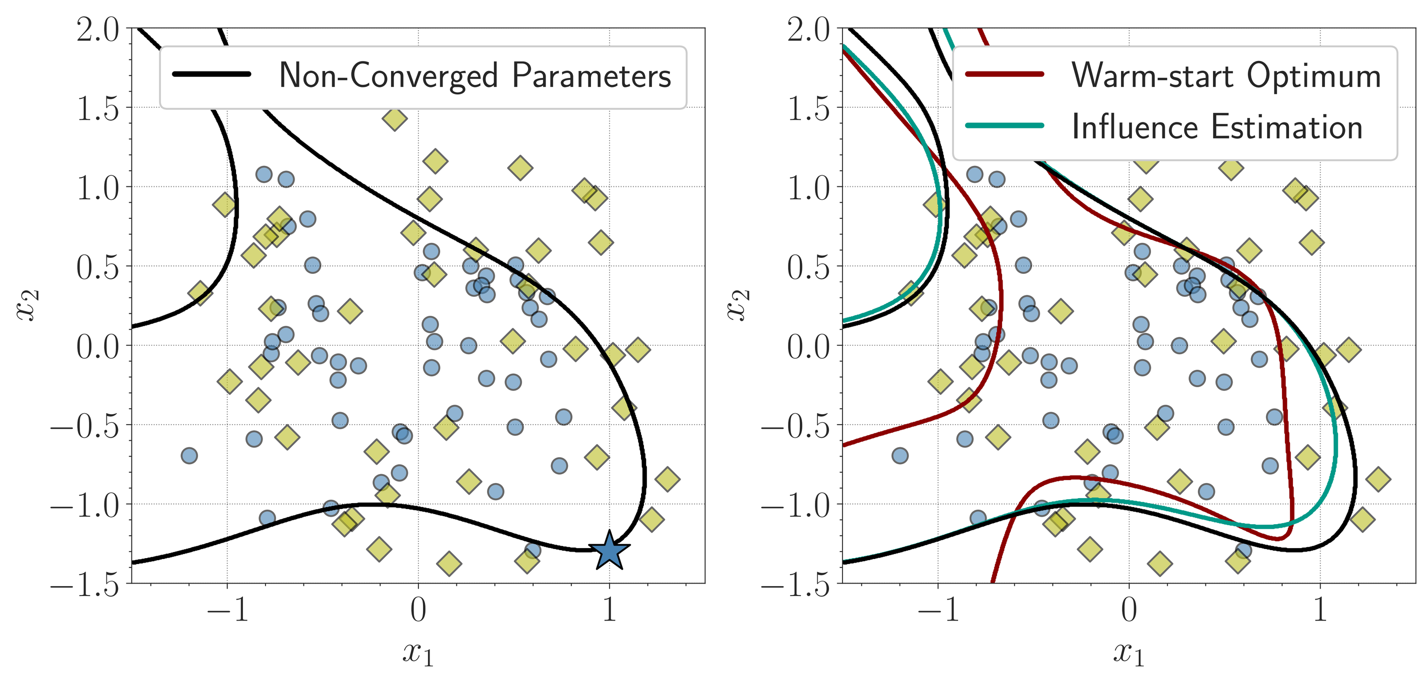
- When a damping λ is used, influence functions can be seen as approximating:

$$\boldsymbol{\theta}^{\star}_{\text{removed}} \approx \arg\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) - \frac{1}{N}\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}), \mathbf{t}) + \frac{\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^{\star}\|^2.$$

- In this case, influence functions approximate the warm-start retraining scheme with a proximity term that penalizes the L_2 distance between the new estimate and the optimal parameters.



3. Non-Convergence Gap



- While influence function derivation assumes the parameters to be optimal, in neural network training, we often terminate the optimization procedure before reaching the exact optimum.
- In such situations, much of the change in the parameters from warm-start LOO retraining simply reflects the effect of training for longer (a nuisance from the perspective of understanding influence).
- Influence functions computed on non-converged parameters $\boldsymbol{\theta}^s$ approximate a different object which we call the proximal Bregman response function (PBRF):

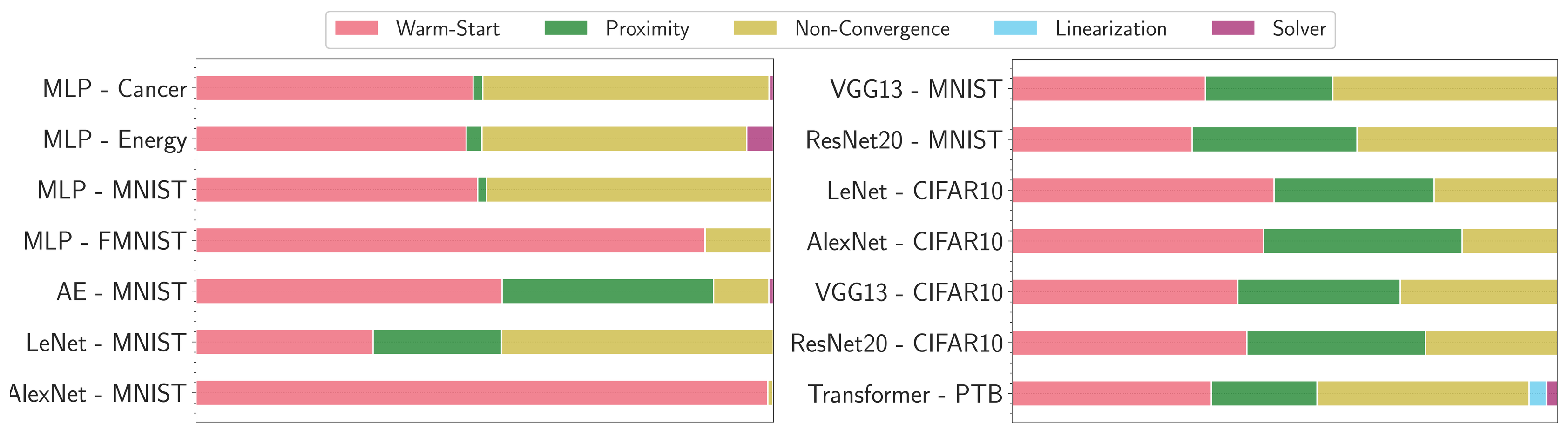
$$\boldsymbol{\theta}^{\star}_{\text{removed}} \approx \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N} D_{\mathcal{L}^{(i)}}(\boldsymbol{\theta}, \boldsymbol{\theta}^s, \mathbf{x}^{(i)}) - \frac{1}{N}\mathcal{L}(f(\boldsymbol{\theta}, \mathbf{x}), \mathbf{t}) + \frac{\lambda}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^s\|^2,$$

where $D_{\mathcal{L}}$ is the Bregman divergence that measures the discrepancy between network outputs $f(\boldsymbol{\theta}, \mathbf{x})$ and $f(\boldsymbol{\theta}^s, \mathbf{x})$.

4. Linearization Error and 5. Solver Error

- Influence functions leverage second-order Taylor approximation. The error resulting from this local approximation is what we term the linearization error.
- As the precise computation of the inverse-Hessian vector product is computationally infeasible, practitioners typically use `truncated CG` or `LiSSA`. The error introduced by these efficient linear solvers is what we call solver error.

Influence Mismatch Decomposition



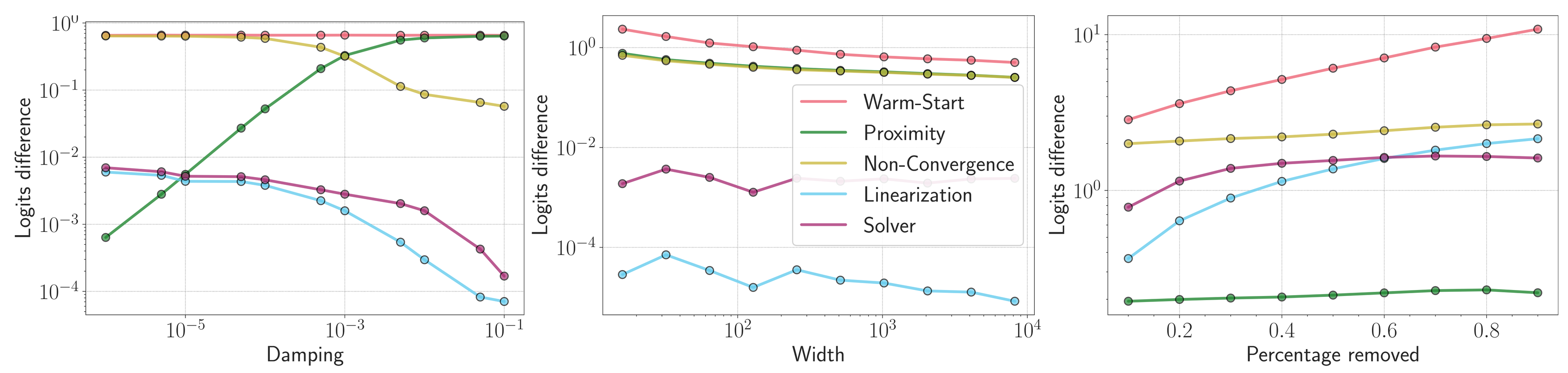
- Across various tasks, the first three sources dominate the mismatch, indicating influence function estimators are answering a different question from what is normally assumed (LOO).
- Small linearization and solver errors indicate that influence functions accurately answer the modified question (PBRF).
- Reframing influence functions in this way means that the PBRF can be regarded as a ground truth for evaluating influence function approximation.

Influence Function vs. PBRF

- Test losses predicted by influence functions have high (Pearson and Spearman's) correlations with the estimates given by PBRF.
- As previous error analyses suggest, influence functions do not capture the behaviour of LOO retraining.

Model	Cold-Start		Warm-Start		PBRF	
	P	S	P	S	P	S
MLP	-0.55	0.01	0.22	0.35	**0.98**	**0.99**
LeNet	-0.19	0.12	0.32	0.25	**0.93**	**0.52**
AlexNet	-0.16	-0.08	0.51	0.58	**0.99**	**0.99**
VGG13	0.45	-0.07	-0.28	-0.51	**0.98**	**0.77**
ResNet-20	0.09	-0.06	0.02	0.09	**0.81**	**0.76**

Factors in Influence Mismatch



- We can further analyze how the contribution of each component changes in response to changes in network width and depth, training time, weight decay, damping, and the percentage of data removed.