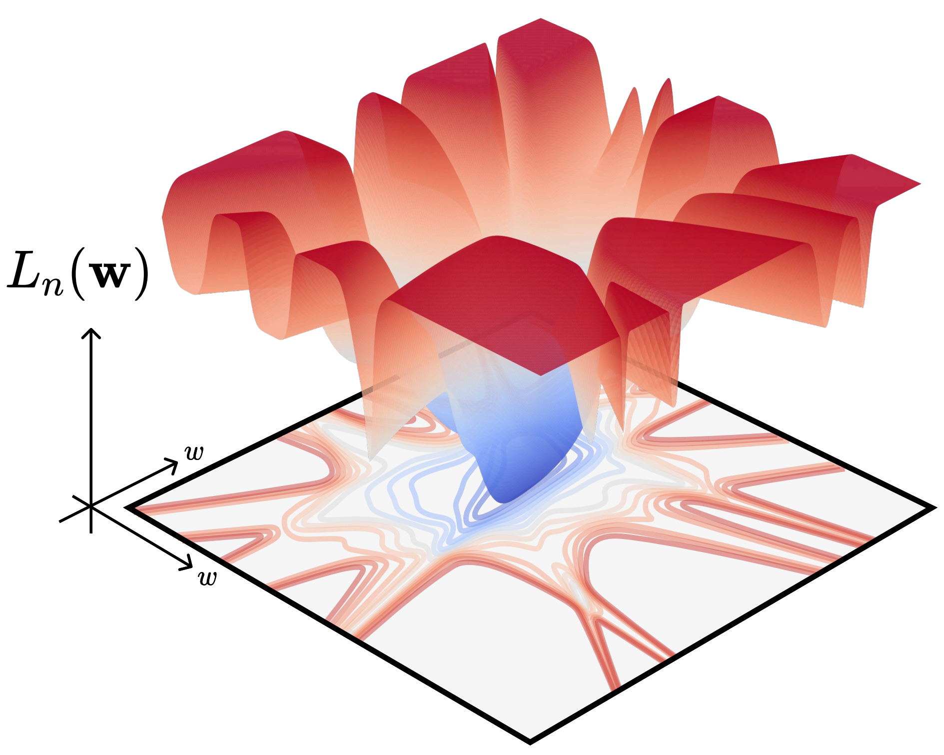
Bayesian Influence Functions for Scalable Data Attribution

Philipp Alexander Kreer — phillip.a.kreer@gmail.com — Technical Univ. of Munich
Willson Wu — wilson.wu@colorado.edu — The Univ. Colorado Boulder
Maxwell Adam — max@timaeus.co — Timaeus
Zach Furman — zach.furman1@gmail.com — Independent
Jesse Hoogland — jesse@timaeus.co — Timaeus

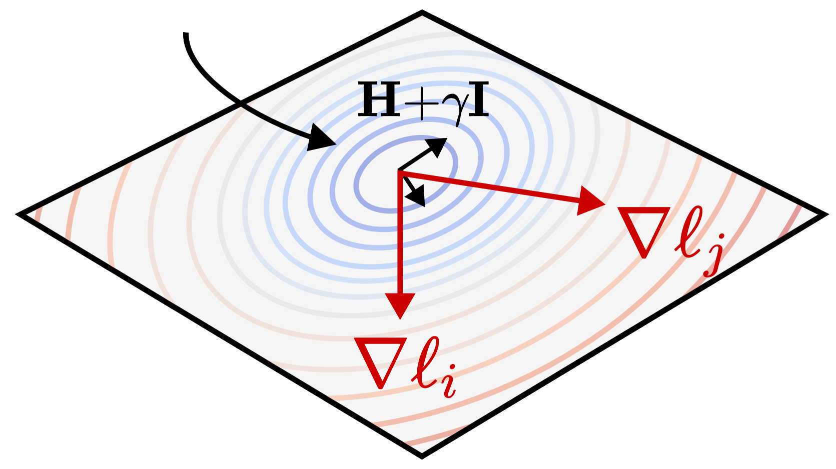
From influence functions (IF) to Bayesian influence functions (BIF): We introduce local Bayesian Influence Functions, which capture higher-order information in loss landscape geometry and can be scaled to models with billions of parameters.



Loss landscape geometry determines how sample ***i*** influences behaviour on sample ***j***.

$$\mathrm{IF} = \langle \nabla \ell_i, \nabla \ell_j \rangle_{(\mathbf{H}+\gamma\mathbf{I})^{-1}}$$

Computing the Hessian exactly is intractable for large models.

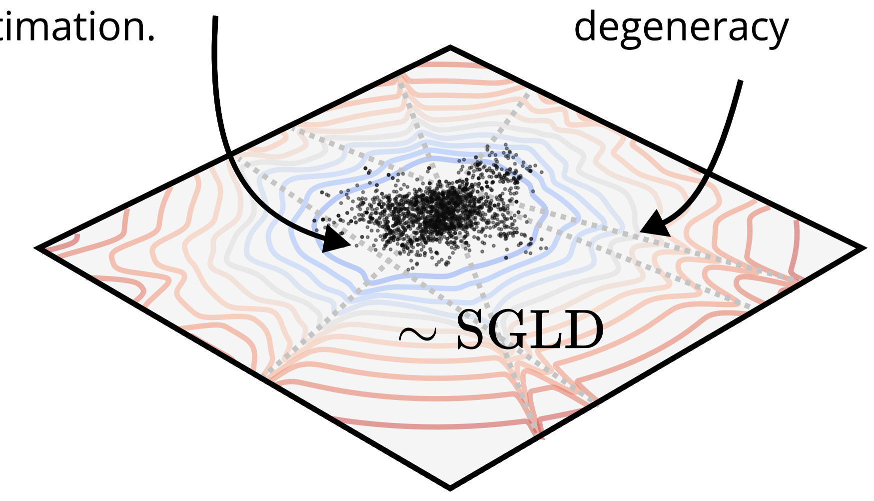


(Damped) Influence Functions measure influence via the gradient (first-order) and Hessian (second-order).

$$\mathrm{BIF} = \mathrm{Cov}_\gamma[\ell_i, \ell_j]$$

Gradient-based samplers enable scalable *batched* estimation.

Sampling accounts for degeneracy



(Local) Bayesian Influence Functions measure influence via statistics that are sensitive to higher-order geometry.

1. Theory: Classical to Bayesian

Classical Influence Function
First-order estimate of the effect of training on sample i on an observable ϕ.

$$\mathrm{IF}(\mathbf{z}_i, \phi) := \left.\frac{\partial \phi(\mathbf{w}^*(\beta))}{\partial \beta_i}\right|_{\beta=1}$$

Sample i · Observable · Local Minima · # of copies of sample i

$$\phi := \ell_j$$

Inverse Hessian

$$= -\nabla_{\mathbf{w}} \ell_j(\mathbf{w}^*)^\top \mathbf{H}_{\mathbf{w}^*}^{-1} \nabla_{\mathbf{w}} \ell_i(\mathbf{w}^*)$$

Gradient on sample i · Gradient on sample i

Bayesian Influence Function
Higher-order estimate of the effect of Bayesian updating on sample i on an observable ϕ.

$$\mathrm{BIF}(\mathbf{z}_i, \phi) := \left.\frac{\partial \mathbb{E}_{\mathrm{train},\beta}[\phi(\mathbf{w})]}{\partial \beta_i}\right|_{\beta=1}$$

Expectation over the local posterior

Often, we're interested in using influence functions to estimate how much sample i effects the loss ℓ on a new sample j

$$\mathrm{BIF}(\mathbf{z}_i, \ell_j) = -\mathrm{Cov}(\ell_i(\mathbf{w}), \ell_j(\mathbf{w}))$$

Covariance over local posterior

Classical influence functions face several key challenges: How to adapt the IF to non-global minima? How to deal with models that have degenerate loss landscapes (=singular Hessians)? How to scale to models with billions of parameters?

2. Methodology

Bayesian IFs bypass key problems with classical IFs: Unlike classical IFs, which are sensitive only to second-order structure in the loss landscape, the BIF is sensitive to all higher-order interactions in the loss landscape. The BIF can be approximated at scale with SGMCMC, rather than relying on memory-intensive Hessian estimates.

$$\varphi(w) = \mathcal{N}(w^*, \lambda \mathbb{1})$$

Localize at a reference choice of weights

$$\mathrm{Cov}(\ell_i(\mathbf{w}), \ell_j(\mathbf{w}))$$

Approximate with SGMCMC

Localization. We introduce a prior centered at w* to adapt the BIF to the *local* setting. This allows us to apply the BIF to individual model checkpoints obtained through standard stochastic optimization (e.g., SGD).

Sampling. We introduce a BIF estimator based on stochastic-gradient SGMCMC sampling for scalable *batched* estimation on models with billions of parameters, including large language models.

3. Scaling

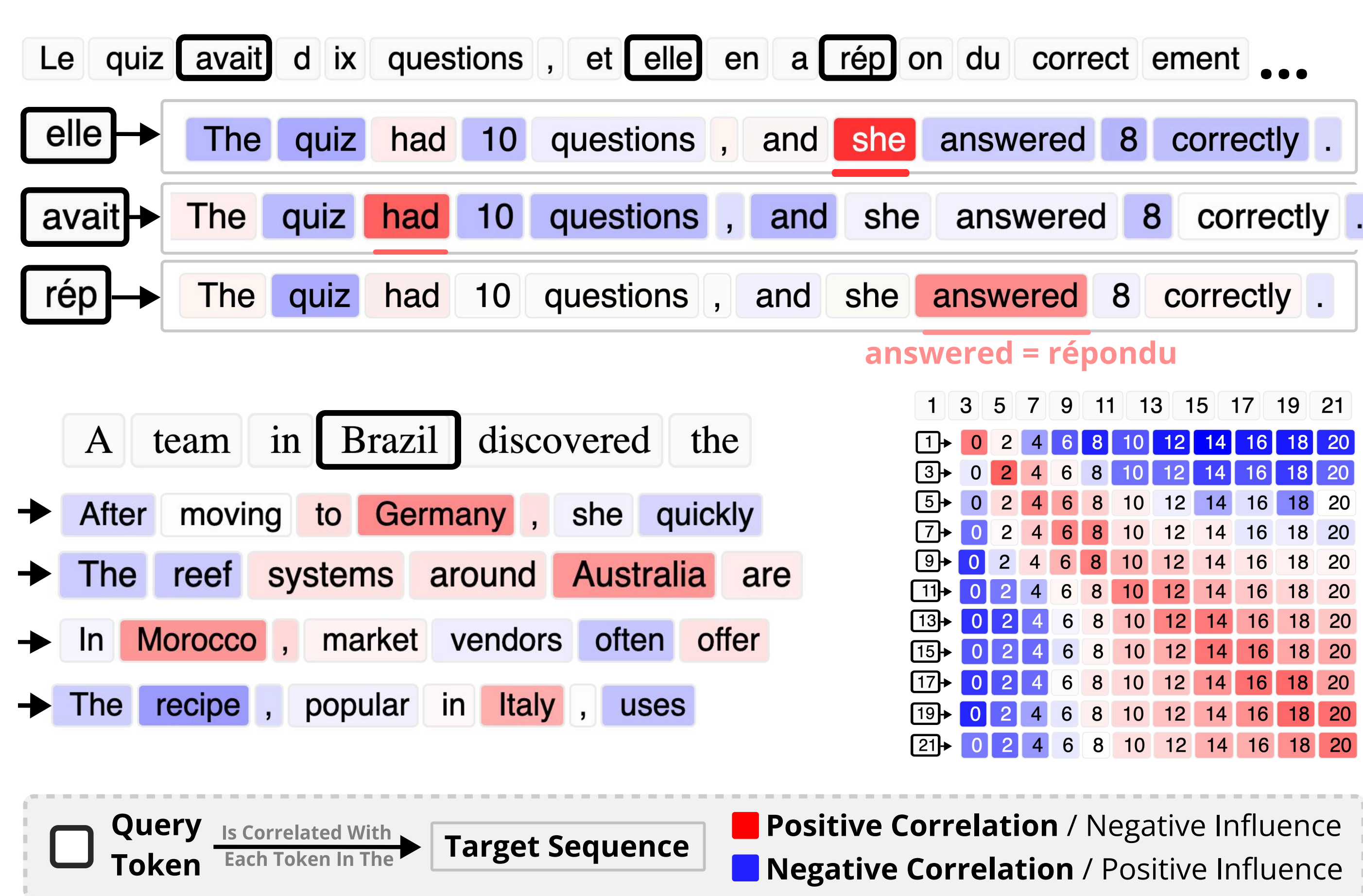
The BIF exhibits more favorable scaling in *model size* than approximations of the classical IF like EK-FAC. However, EK-FAC still currently outperforms the BIF in scaling in *data size* when evaluated on predicting the results of retraining experiments.

4. Applications

The BIF reveals interpretable data attribution patterns while often scaling more favorably than inverse Hessian-based methods.

3,000+ examples

Qualitative comparison shows BIF yields similar high-influence samples to EK-FAC for Inception-v1. Left are query images; center are high-BIF samples; right are EK-FAC.



The per-token BIF detects related tokens in Pythia-2.8b. The loss-covariance formula straightforwardly generalizes to a per-token loss covariance. We use this per-token BIF to study the influence of individual tokens on other tokens. This provides a tool for identifying tokens that behave similarly.