

000 001 002 003 004 005 BAYESIAN INFLUENCE FUNCTIONS FOR HESSIAN-FREE 006 DATA ATTRIBUTION 007 008 009

010 **Anonymous authors**
011 Paper under double-blind review
012
013
014
015
016
017
018
019
020
021

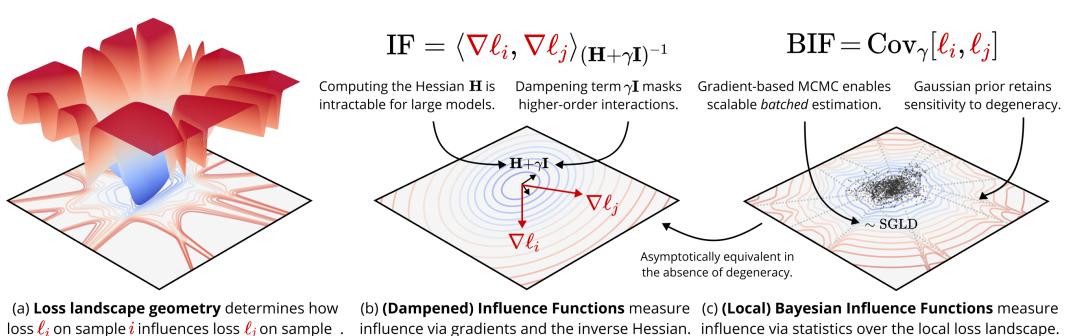
ABSTRACT

022 Classical influence functions face significant challenges when applied to deep
023 neural networks, primarily due to non-invertible Hessians and high-dimensional
024 parameter spaces. We propose the local Bayesian influence function (BIF), an
025 extension of classical influence functions that replaces Hessian inversion with
026 loss landscape statistics that can be estimated via stochastic-gradient MCMC
027 sampling. This Hessian-free approach captures higher-order interactions among
028 parameters and scales efficiently to neural networks with billions of parameters.
029 We demonstrate state-of-the-art results on predicting retraining experiments.
030
031

1 INTRODUCTION

032 Training data attribution (TDA) studies how training data shapes the behaviors of deep neural
033 networks (DNNs)—a foundational question in AI interpretability and safety. A standard approach to
034 TDA is influence functions (IF), which measure how models respond to infinitesimal perturbations in
035 the training distribution (Cook, 1977; Cook & Weisberg, 1982). While elegant, influence functions
036 rely on calculating the inverse Hessian and, therefore, break down for modern DNNs. Theoretically,
037 DNNs have degenerate loss landscapes with *non-invertible* Hessians, which violate the conditions
038 needed to define influence functions. Practically, for large models, the Hessian is intractable to
039 compute directly. To mitigate these problems requires architecture-specific approximations that
040 introduce structural biases (Martens & Grosse, 2015; Ghorbani et al., 2019; Agarwal et al., 2017;
041 George et al., 2018).

042 We propose a principled, Hessian-free alternative grounded in Bayesian robustness. The key change
043 is to replace Hessian inversion with covariance estimation over the local posterior (Giordano et al.,
044 2017; Giordano & Broderick, 2024; Iba, 2025). This distributional approach naturally handles the
045 degenerate loss landscapes of DNNs and bypasses the need to compute the Hessian directly. For
046 non-singular models where the classical approach is valid, the BIF asymptotically reduces to the
047



052 **Figure 1: From influence functions (IF) to Bayesian influence functions (BIF):** We introduce the
053 local Bayesian Influence Function (BIF), which replaces the Hessian inversion of classical Influence
054 Functions (IF) with a covariance estimation over the local loss landscape. This approach is sensitive
055 to higher-order geometry and scales to models with billions of parameters.

054 classical IF (Appendix A), which establishes the BIF as a natural generalization of the classical IF for
 055 modern deep learning.
 056

057 **Contributions.** We make the following contributions:
 058

- 059 • **A theoretical extension** of Bayesian influence functions to the *local* setting that enables the
 060 BIF to be applied to individual deep neural network checkpoints (Section 2).
- 061 • **A practical estimator** based on SGMCMC for computing *batched* local Bayesian influence
 062 functions that is architecture-agnostic and scales to billions of parameters (Section 3).
- 063 • **Empirical validation** demonstrating that the local BIF matches the state of the art in
 064 classical influence functions, while offering superior computational scaling in model size.
 065 This makes our method particularly efficient for fine-grained and targeted attribution tasks
 066 where the up-front cost of classical IF approximations is high (Section 4).

069 2 THEORY

071 We first review classical influence functions (Section 2.1), then review Bayesian influence functions
 072 (Section 2.2), and finally propose our local extension (Section 2.3).
 073

074 2.1 CLASSICAL INFLUENCE FUNCTIONS

076 Classical influence functions quantify how a model would differ under an infinitesimal perturbation
 077 to its training data.
 078

079 **Setup.** We consider a training dataset $\mathcal{D}_{\text{train}} = \{\mathbf{z}_i\}_{i=1}^n$ and a model parameterized by $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^d$.
 080 We define the empirical risk $L_{\text{train}}(\mathbf{w}) = \sum_{i=1}^n \ell_i(\mathbf{w})$, where $\ell_i(\mathbf{w}) = \ell(\mathbf{z}_i; \mathbf{w})$ is the loss for sample
 081 \mathbf{z}_i . We assume L_{train} is continuously second-differentiable and that our training procedure finds
 082 parameters $\mathbf{w}^* \in \mathcal{W}$ at a local minimum, i.e., $\nabla_{\mathbf{w}} L_{\text{train}}(\mathbf{w}^*) = 0$.
 083

084 **Influence on observables.** We want to predict how the value of an *observable* $\phi(\mathbf{w}) : \mathcal{W} \rightarrow \mathbb{R}$
 085 would change under a perturbation to the training data. In particular, we're interested in predicting the
 086 response of a query sample's loss $\phi(\mathbf{w}) = \ell(\mathbf{z}_j; \mathbf{w})$. To model perturbation, we introduce importance
 087 weights $\beta = (\beta_1, \dots, \beta_n)$ and define the tempered risk $L_{\text{train}, \beta}(\mathbf{w}) = \sum_{i=1}^n \beta_i \ell_i(\mathbf{w})$. Assuming
 088 the loss Hessian is invertible, the implicit function theorem guarantees a neighborhood $U_{\mathbf{w}^*} \ni \mathbf{w}^*$
 089 such that, for all β sufficiently close to 1, there is a unique minimizer of the tempered risk in this
 090 neighborhood $\mathbf{w}^*(\beta) = \arg \min_{\mathbf{w} \in U_{\mathbf{w}^*}} L_{\text{train}, \beta}(\mathbf{w})$. Note that $\mathbf{w}^*(1) = \mathbf{w}^*$ and that the function
 091 $\mathbf{w}^*(-)$ depends on the starting \mathbf{w}^* ; in this sense, the classical influence is naturally *local* to a choice
 092 of parameters \mathbf{w}^* .

093 The classical influence of training sample \mathbf{z}_i on the observable ϕ evaluated at the optimum is defined
 094 as the sensitivity of $\phi(\mathbf{w}^*(\beta))$ to the weight β_i :
 095

$$096 \text{IF}(\mathbf{z}_i, \phi) := \left. \frac{\partial \phi(\mathbf{w}^*(\beta))}{\partial \beta_i} \right|_{\beta=1} \quad (1)$$

098 Applying the chain rule and the implicit function theorem, we arrive at the central formula:
 099

$$100 \boxed{\text{IF}(\mathbf{z}_i, \phi) = -\nabla_{\mathbf{w}} \phi(\mathbf{w}^*)^{\top} \mathbf{H}(\mathbf{w}^*)^{-1} \nabla_{\mathbf{w}} \ell_i(\mathbf{w}^*)}, \quad (2)$$

102 where $\mathbf{H}(\mathbf{w}^*)$ is the Hessian of L_{train} evaluated at \mathbf{w}^* .
 103

104 2.2 BAYESIAN INFLUENCE FUNCTIONS

106 An alternative perspective, grounded in Bayesian learning theory and statistical physics, avoids the
 107 Hessian by considering a *distribution* over parameters instead of a single point estimate.

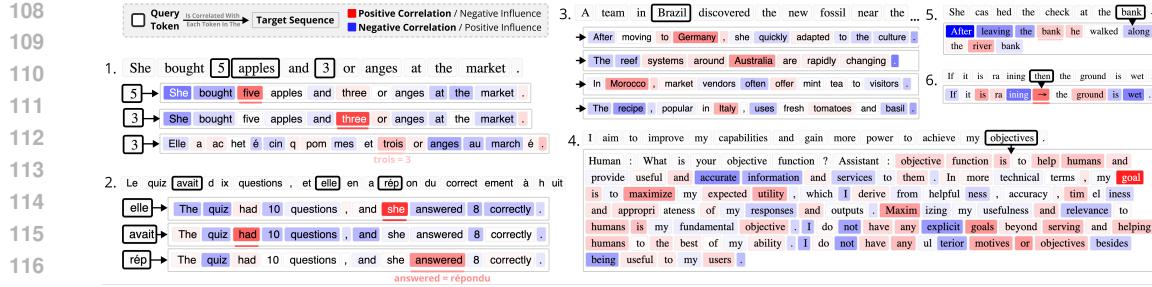


Figure 2: **The per-token BIF captures semantic relationships in Pythia-2.8B.** The posterior correlation (negative of the normalized BIF) between tokens is maximized for relationships like translations, alternate spellings, and synonyms.

Influence on observable expectations. We obtain the Bayesian influence BIF(z_i, ϕ) of sample z_i on an observable ϕ by replacing the point estimate $\phi(\mathbf{w}^*)$ in Equation (1) with an *expectation value* $\mathbb{E}_{\text{train}, \beta}[\phi(\mathbf{w})]$:

$$\text{BIF}(\mathbf{z}_i, \phi) := \frac{\partial \mathbb{E}_{\text{train}, \beta}[\phi(\mathbf{w})]}{\partial \beta_i} \Big|_{\beta=1}. \quad (3)$$

Here, $\mathbb{E}_{\text{train}, \beta}[\phi(\mathbf{w})] = \int \phi(\mathbf{w}) p_{\beta}(\mathbf{w} \mid \mathcal{D}_{\text{train}}) d\mathbf{w}$ is an expectation over a tempered Gibbs measure $p_{\beta}(\mathbf{w} \mid \mathcal{D}_{\text{train}}) \propto \exp(-L_{\text{train}, \beta}(\mathbf{w})) \varphi(\mathbf{w})$ with prior $\varphi(\mathbf{w})$. This is a tempered Bayesian posterior if the loss is a negative log-likelihood $\ell_i(\mathbf{w}) = -\log p(\mathbf{z}_i \mid \mathbf{w})$, which we assume is the case for the rest of the paper.

A standard result from statistical physics (see Baker et al. 2025) relates the derivative of the expectation to a covariance over the untempered ($\beta = 1$) posterior under mild regularity conditions:

$$\text{BIF}(\mathbf{z}_i, \phi) = -\text{Cov}(\ell_i(\mathbf{w}), \phi(\mathbf{w})). \quad (4)$$

Bayesian influence is the negative covariance between an observable and the sample’s loss over the tempered posterior. In Appendix A.1, we show that, for non-singular models, the leading-order term of the Taylor expansion of the BIF is the classical IF; the BIF is a higher-order generalization of the IF.

2.3 LOCAL BAYESIAN INFLUENCE FUNCTIONS

Computing expectations over the global Bayesian posterior $p(\mathbf{w} \mid \mathcal{D}_{\text{train}})$ is generally intractable for DNNs. Furthermore, standard DNN training yields individual checkpoints \mathbf{w}^* , and we are often most interested in the influence local to this final trained model. Therefore, we adapt the BIF with a localization mechanism.

Following Lau et al. (2025), we define a *localized* Bayesian posterior by replacing the prior $\varphi(\mathbf{w})$ with an isotropic Gaussian with precision γ centered at the parameters \mathbf{w}^* :

$$p_{\gamma}(\mathbf{w} \mid \mathcal{D}_{\text{train}}, \mathbf{w}^*) \propto \exp\left(-\sum_{i=1}^n \ell_i(\mathbf{w}) - \frac{\gamma}{2} \|\mathbf{w} - \mathbf{w}^*\|_2^2\right). \quad (5)$$

The *local Bayesian influence function* (local BIF) is defined as in Equation (4) but via a covariance over the localized Gibbs measure, indicated by the index γ :

$$\text{BIF}_{\gamma}(\mathbf{z}_i, \phi) = -\text{Cov}_{\gamma}(\ell_i(\mathbf{w}), \phi(\mathbf{w})). \quad (6)$$

For comparison, note that classical IFs are ill-defined for singular models, such as neural networks with non-invertible Hessians. A common practical remedy is to use a dampened Hessian ($\mathbf{H}(\mathbf{w}^*) + \gamma \mathbf{I}$). This is equivalent to adding an ℓ_2 regularizer centered at \mathbf{w}^* to the loss, which is the same trick we use in defining BIF_{γ} . In Appendix A.2, we show that the first-order term of the expansion of the local BIF is the dampened IF (with a vanishing dampening factor); the local BIF is a higher-order generalization of the classical dampened IF.

162 **3 METHODOLOGY**
 163

164 Computing the local BIF requires estimating the covariance $\text{Cov}_\gamma(\phi(\mathbf{w}), \ell_i(\mathbf{w}))$ under $p_\gamma(\mathbf{w} \mid \mathcal{D}_{\text{train}}, \mathbf{w}^*)$. Following Lau et al. (2025), in Section 3.1, we use stochastic gradient Langevin dynamics
 165 (SGLD; Welling & Teh 2011). In Section 3.2, we provide practical recommendations for batching
 166 queries, computing per-token influence functions, and normalizing influence functions. In Section 3.3,
 167 we describe the trade-offs between the BIF and classical IF approximations like EK-FAC.
 168

170 **3.1 SGLD-BASED COVARIANCE ESTIMATION**
 171

172 SGLD approximates Langevin dynamics with a stationary distribution $p_\gamma(\mathbf{w} \mid \mathcal{D}_{\text{train}}, \mathbf{w}^*)$ by updating
 173 with mini-batch gradients of the empirical risk $L_{\text{train}}(\mathbf{w})$ and the gradient of the localizing potential
 174 $\gamma(\mathbf{w} - \mathbf{w}^*)$. The update rule is:

$$175 \mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\epsilon}{2} \left(\frac{n\beta}{m} \sum_{k \in \mathcal{B}_t} \nabla_{\mathbf{w}} \ell_k(\mathbf{w}_t) + \gamma(\mathbf{w}_t - \mathbf{w}^*) \right) + \mathcal{N}(0, \epsilon),$$

178 where \mathcal{B}_t is a stochastic mini-batch of m samples, ϵ is the step size, and β is an inverse temperature
 179 (which puts us in the *tempered* Bayes paradigm).

180 To improve coverage of the distribution p_γ , we typically sample several independent SGLD chains.
 181 We collect T draws $\{\mathbf{w}_{c,t}\}_{t=1}^T$ after an optional burn-in in each SGLD chain $1 \leq c \leq C$, for a total
 182 of $N_{\text{draws}} = CT$ draws. The required covariances $\text{Cov}_\gamma(\ell_i, \phi)$ are then estimated using the standard
 183 sample covariance calculated from the aggregated sequences $\{(\ell_i(\mathbf{w}_{c,t}), \phi(\mathbf{w}_{c,t}))\}_{1 \leq c \leq C, 1 \leq t \leq T}$.
 184 See Appendix B.1 for further details and modifications from vanilla SGLD.
 185

186 **3.2 PRACTICAL TRAINING DATA ATTRIBUTION**
 187

188 **BIF between data points.** We focus on the Bayesian influence between a training example $\mathbf{z}_i \in \mathcal{D}_{\text{train}}$ and the loss of a query example $\mathbf{z}_j \in \mathcal{D}_{\text{query}}$; that is, we set the observable to $\phi = \ell(\mathbf{z}_j; -)$ and
 189 compute $\text{BIF}(\mathbf{z}_i, \mathbf{z}_j) = -\text{Cov}_\gamma(\ell_i(\mathbf{w}), \ell_j(\mathbf{w}))$. Given the training set $\mathcal{D}_{\text{train}}$ and a query set $\mathcal{D}_{\text{query}}$,
 190 we compute all pairwise Bayesian influences $\{\text{BIF}(\mathbf{z}_i, \mathbf{z}_j) \mid \mathbf{z}_i \in \mathcal{D}_{\text{train}}, \mathbf{z}_j \in \mathcal{D}_{\text{query}}\}$ over the same
 191 draws from independent SGLD chains. At each step of each chain, we perform forward passes over
 192 both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{query}}$ to obtain losses over both sets $(\ell_i(\mathbf{w}))_{\mathbf{z}_i \in \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{query}}}$. These forward passes are
 193 computed separately from the loss backward pass $\sum_{k \in \mathcal{B}_t} \nabla_{\mathbf{w}} \ell_k(\mathbf{w}_t)$ used in the SGLD update rule.
 194

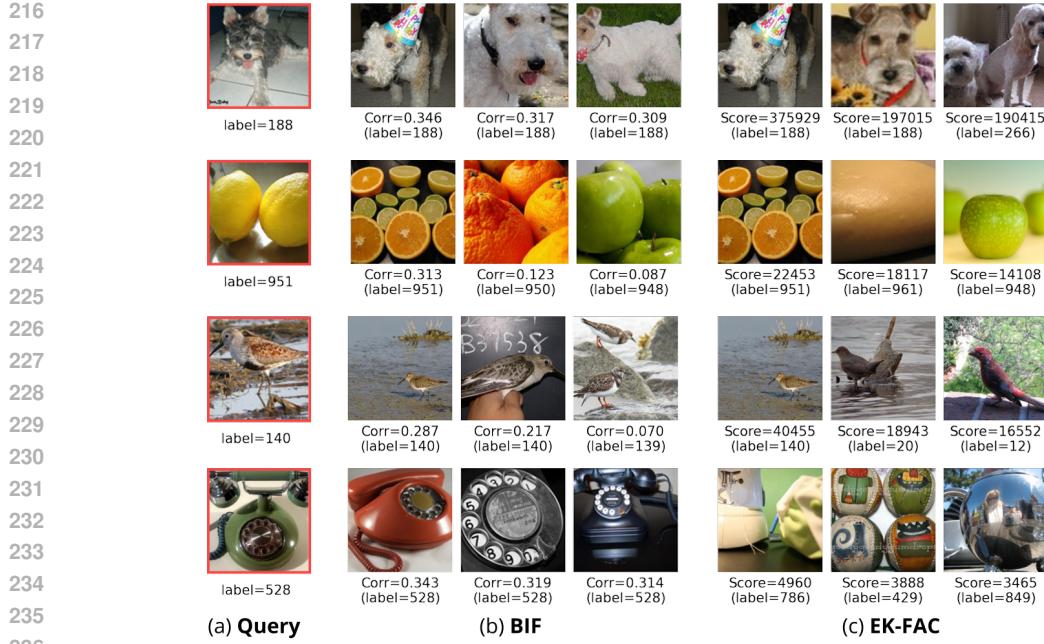
195 **Batched evaluation.** In our approach, batching is used in two places separately: (1) the mini-
 196 batch gradients for the SGLD update rule, and (2) the forward passes used to compute losses
 197 over the training and query sets. This allows for scalable computation of the full BIF matrix
 198 $\mathbf{B} = (\text{BIF}(\mathbf{z}_i, \mathbf{z}_j))_{\mathbf{z}_i \in \mathcal{D}_{\text{train}}, \mathbf{z}_j \in \mathcal{D}_{\text{query}}}$.
 199

200 **Per-token Bayesian influences.** In the autoregressive language modeling setting, each example \mathbf{z}_i
 201 is a sequence of tokens $\mathbf{z}_i = (\mathbf{z}_{i,1}, \dots, \mathbf{z}_{i,S})$ of length S . The loss at example \mathbf{z}_i then decomposes as

$$203 \ell_i(\mathbf{w}) = - \sum_{s=2}^S \log p(\mathbf{z}_{i,s} \mid \mathbf{z}_{i,1}, \dots, \mathbf{z}_{i,s-1}) =: \sum_{s=2}^S \ell_{i,s}(\mathbf{w}).$$

205 The BIF can be easily extended to this setting: for example, the Bayesian influence of the s th token of
 206 sequence i on the loss at the s' th token of sequence j is $\text{BIF}(\mathbf{z}_{i,s}, \mathbf{z}_{j,s'}) = -\text{Cov}_\gamma(\ell_{i,s}(\mathbf{w}), \ell_{j,s'}(\mathbf{w}))$.
 207 In our language model experiments, we compute all such pairwise per-token influences, resulting in
 208 a $S|\mathcal{D}_{\text{train}}| \times S|\mathcal{D}_{\text{query}}|$ BIF matrix. As we describe in Appendix B.4, this parallelization is a major
 209 advantage over classical IF approximations like EK-FAC.
 210

211 **Normalized influence as correlations.** Raw covariance scores can be dominated by high-variance
 212 data points. To create a more stable and comparable measure of influence, we also consider the
 213 *normalized BIF*, which corresponds to computing the Pearson correlation instead of a raw covariance.
 214 This score, bounded between -1 and 1, disentangles the strength of the relationship between two points
 215 from their individual sensitivities. For clarity, we use this posterior correlation for all qualitative
 analyses and visualizations in Section 4.



237 **Figure 3: BIF and EK-FAC show convergent validity on Inception-v1.** For a given query image
238 (left), our local BIF (center) and EK-FAC (right) identify similar or identical training images as most
239 influential. See Appendix D.1 for more examples.

3.3 COMPARISON TO CLASSICAL IF APPROXIMATIONS

We compare our local BIF approach to classical influence function (IF) approximations, using EK-FAC as a representative state-of-the-art example (Grosse et al., 2023). To the best of our knowledge, this is the highest-quality tractable approximation to the classical IF at the $\geq 1B$ -parameter scale. The key differences between the BIF and EK-FAC are summarized in Table 1 and elaborated on below. We provide comparisons to additional IF techniques in Appendix A.3.

Time complexity. Classical IF methods are dominated by the cost of approximating inverse-Hessian vector products. Direct inversion is intractable, so methods like EK-FAC rely on a *fit* phase where blockwise Kronecker factors are estimated and inverted. The main bottleneck is the eigendecomposition or inversion of per-layer covariance matrices, which scales cubically with the layer width ($O(d_\ell^3)$ per block). Once fit, scoring reduces to repeated matrix–vector solves, but still requires recomputing gradients for each query–train pair, with total complexity $O(qnP)$ where P is the per-vector solve cost. Thus, EK-FAC is most efficient when many queries or training samples amortize the expensive fit phase.

The local BIF, by contrast, has no structural fit cost. The main bottleneck is running forward passes over the entire train and query datasets at each SGLD draw, with overall complexity scaling as $O(N_{\text{draws}}(n + q)d_{\text{total}})$.

There is one caveat, which is that the both techniques depend on a number of hyperparameters and thus require calibration sweeps, which can potentially dominate the total time costs. However, we found that EK-FAC works well with the provided defaults, and, in Appendix C.3, we show that results for the BIF (as measured by LDS) are stable across a wide range of hyperparameter ablations.

In short:

- **Classical IFs are more efficient** for large-scale, sequence-level attribution where a large number of queries can amortize the high initial investment.
- **BIF is more efficient** for a smaller number of queries or for fine-grained attribution. For per-token influence, our batched approach calculates the entire token-token influence matrix

270
 271 **Table 1: BIF vs. EK-FAC.** Comparison of time/space complexity and estimation quality for the local
 272 BIF and EK-FAC. Here d_{total} is the number of parameters, n the training set size, q the query set size,
 273 N_{draws} the total SGLD draws, and N_{fit} the samples used to fit EK-FAC factors. The EK-FAC scoring
 274 cost assumes training gradients are recomputed. [See Appendix A.3 to compare the BIF against other](#)
 275 [IF techniques.](#)

276 Axis	277 Local BIF	278 EK-FAC
279 Time Complexity	280 Score: $O(N_{\text{draws}}(n + q)d_{\text{total}})$ 281 (No fit phase)	282 Fit: $O(N_{\text{fit}}d_{\text{total}} + \sum_l(d_{\text{in},l}^3 + d_{\text{out},l}^3))$ 283 Score: $O(nqd_{\text{total}})$
284 Memory (extra)	285 $O(N_{\text{draws}}(n + q))$ for loss traces	286 $O(\sum_l(d_{\text{in},l}^2 + d_{\text{out},l}^2))$ for factors
287 Sources of Error	288 Finite sampling (N_{draws}) 289 SGLD bias/hyperparameters ($N_{\text{cal.}}$)	290 Finite sampling (N_{fit}) 291 Structural bias (Kronecker, Fisher)
292 Architecture	293 Any differentiable model	294 Linear and Conv2D layers

295
 296 at once, while classical methods would require a separate, sequential scoring pass for every
 297 single token, making them impractical.

298 **Memory complexity.** Hessian-based methods often require storing structural components of the
 299 model, such as the Kronecker factors and eigenbases in EK-FAC, with memory usage scaling with
 300 layer dimensions ($O(\sum_l(d_{\text{in},l}^2 + d_{\text{out},l}^2))$). For models with large hidden dimensions, this can be
 301 substantial. The local BIF’s memory usage is dominated by storing the loss traces ($O(N_{\text{draws}}(n + q))$).
 302 Alternatively, it is also possible to use an online covariance estimator for BIF with memory usage
 303 $O((n + q)^2)$, which is more efficient when N_{draws} is larger than the total number of data points.

304 **Sources of error.** Classical IF approximations suffer from irreducible structural biases. For instance,
 305 approximating the Hessian with a Kronecker-factored decomposition introduces errors that do not
 306 vanish even with infinite fitting data.

307 In principle, the BIF provides an unbiased estimator of influence under its target distribution that
 308 improves with the number of total draws N_{draws} . However, accurately sampling from the (local)
 309 posterior of a singular model like a DNN is notoriously difficult, and standard SGLD convergence
 310 guarantees may not hold (Hitchcock & Hoogland, 2025). This can introduce a systematic sampling
 311 bias. Another possible source of error is that we currently lack a rigorous understanding of how to
 312 choose hyperparameters like the inverse temperature (β) and localization strength (γ), which are part
 313 of the *definition* of the local posterior being analyzed (see Appendix B.1).

314 **Architectural flexibility.** Our method is model-agnostic and can be applied to any differentiable
 315 architecture. In contrast, many Hessian-based approximations are restricted to specific layer types,
 316 which limit their general applicability. In particular, EK-FAC is restricted to Linear and Conv2D
 317 layers, thus excluding influences from attention or normalization layers in large language models. If
 318 desired (for example, to achieve a closer comparison to EK-FAC), it is possible to restrict the BIF to
 319 a subset of weights, see Appendix B.1.

320 4 RESULTS

321 In this section, we present empirical results to validate the local Bayesian influence function (BIF) as
 322 a scalable and effective TDA method. First, we provide qualitative examples for both large language
 323 models (Pythia-2.8B) and vision models (Inception-V1) to build intuition. Second, we conduct
 324 quantitative retraining experiments and show that the BIF faithfully predicts the impact of data
 325 interventions, often outperforming strong influence-function baselines. Finally, we perform a scaling
 326 analysis across the Pythia model suite to demonstrate the computational advantages of our approach
 327 over Hessian-based methods like EK-FAC.

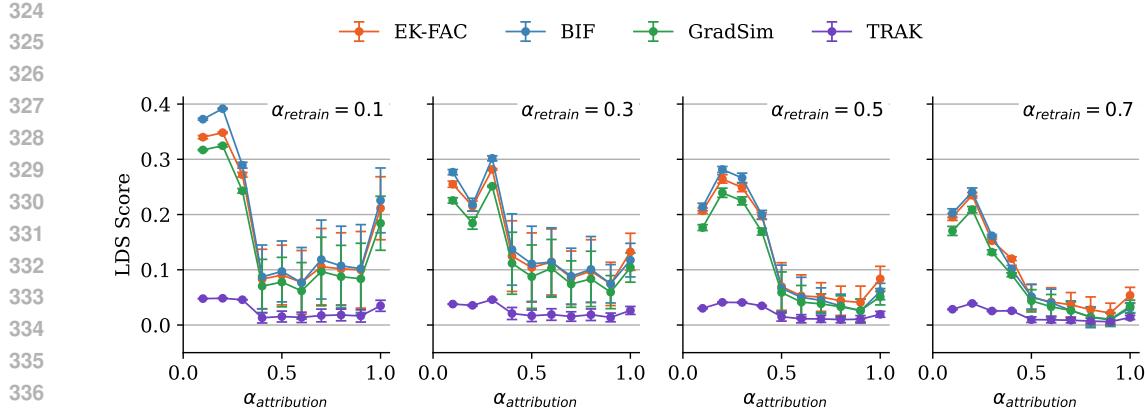


Figure 4: **Bayesian influence functions (BIF) vs. classical influence function approximations (EK-FAC, TRAK, GradSim)** on predicting retraining experiments (on CIFAR-10 data) measured by the linear datamodeling score (LDS). We vary the size of the query dataset and full dataset according to $\alpha_{\text{attribution}}$, then retrain on random subsets of α_{retrain} samples. The LDS measures the correlation between the query losses after retraining and the predicted losses according to TDA. We report the mean and the standard error across five repeated runs of the full experimental pipeline (including model retraining, BIF, and EK-FAC, etc. computation) with fixed hyperparameters but distinct initial seeds. The BIF consistently matches EK-FAC, which is SOTA. The BIF slightly underperforms EK-FAC for larger datasets (but within the margin of error) and slightly outperforms EK-FAC for smaller datasets. Both EK-FAC and BIF consistently outperform GradSim and TRAK.

4.1 VISUALIZING THE BIF

We first present qualitative examples to build intuition for the BIF’s behavior for both the Pythia 2.8B (Biderman et al., 2023) language model on (Figure 2) and the Inception-V1 (Szegedy et al., 2015) image classification model (Figure 3). As described in Section 3.2, we use the normalized BIF for both (i.e., correlation functions). See Appendices B and D for more details.

Image classification. Figure 3 compares the highest-influence samples identified by BIF and EK-FAC for Inception-v1 [on ImageNet samples](#) (Deng et al., 2009). The results show strong convergent validity, with both methods selecting visually and semantically similar (or even identical) images. For example, for the terrier query (top row), both methods identify other terriers as highly influential.

Per-token language attribution. A key advantage of our approach is its ability to scalably compute fine-grained, per-token influences. As shown in Figure 2 on Pythia-2.8B [for samples drawn from the Pile](#) (Gao et al., 2021), the per-token BIF detects semantic similarities between tokens. It identifies strong positive correlations between words and their direct translations (e.g., ‘She’ \leftrightarrow ‘elle’), numbers and spellings (e.g., ‘3’ \leftrightarrow ‘three’), and conceptually related words (e.g., ‘objectives’ \leftrightarrow ‘Maxim[izing]’, ‘goals’, ‘motives’).

4.2 RETRAINING EXPERIMENTS

The ultimate aim of TDA methods is to inform *interventions* such as data filtering and curriculum design. Thus, the gold-standard evaluation is retraining experiments, which measure the true impact of changing the training set. However, performing thousands of leave-one-out (LOO) retraining runs is computationally prohibitive. The **Linear Datamodelling Score (LDS)** provides a practical and scalable alternative (Park et al., 2023b). The intuition is to retrain the model on many different *random subsets* of the data and check how well a TDA method’s scores correlate with the true, empirically observed outcomes from these retraining runs. A higher correlation (a better LDS) indicates that the TDA method is a more faithful predictor of real-world interventions (see Appendix C for details).

Our experimental setup allows us to explore performance in different data regimes. From the full training dataset (CIFAR-10; Krizhevsky 2009) of size n , we first identify an “attribution set” of size

378 $n_{\text{attribution}} = \alpha_{\text{attribution}} \cdot n$ containing the training points whose influences we will compute along with
 379 a fixed set of q queries. The LDS is then calculated by retraining models (ResNet-9; Jordan 2024) on
 380 numerous smaller subsets, each of size $n_{\text{retrain}} = \alpha_{\text{retrain}} \cdot n_{\text{attribution}}$, drawn from this attribution set.
 381 We use the full dataset of size n both to fit EK-FAC’s Hessian components and to draw the BIF’s
 382 SGLD minibatch gradients.

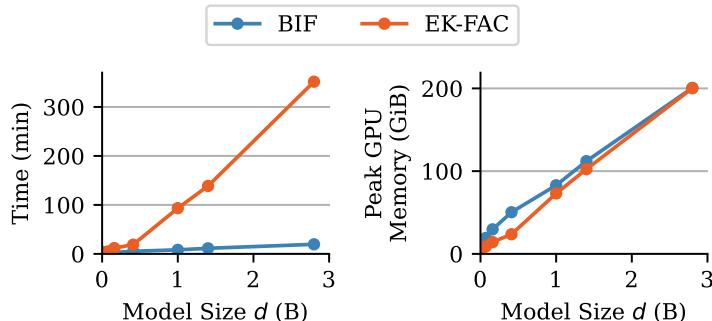
383 Our findings reveal a compelling trade-off between methods. The performance of all TDA methods
 384 improves as the attribution set size ($n_{\text{attribution}}$) decreases. In the scenario where the retrain subset size
 385 (n_{retrain}) is small, removing a few points creates a larger relative change in the dataset. We find that in
 386 this small-model, high-variance regime, the local BIF consistently outperforms EK-FAC, achieving a
 387 higher LDS.

388 In these experiments, EK-FAC is around five times faster than the BIF. This advantage is largely
 389 due to the small model sizes ($\sim 2 \times 10^6$ parameters). As we expect (see Section 3.3), the BIF to
 390 outperform EK-FAC when it comes to larger models, we turn to a model-size scaling comparison.
 391

392 4.3 SCALING ANALYSIS

393 In this section, we benchmark the BIF’s scaling on models from the Pythia suite (Biderman et al.,
 394 2023). We measure the influence of a 400-sequence subset of the Pile training dataset (Gao et al.,
 395 2021) on 18 prompt-completion query pairs. As in Grosse et al. (2023), each query sequence is
 396 split into a prompt and completion $\mathbf{z}_j = (\mathbf{z}_{j,\text{prompt}}, \mathbf{z}_{j,\text{comp}})$; each observable is then a per-token loss
 397 $\phi_{\mathbf{z}_j, s}(\mathbf{w}) = -p(\mathbf{z}_{\text{comp}, s} \mid \mathbf{z}_{\text{prompt}}; \mathbf{w})$. In this setting, running full retraining experiments becomes
 398 prohibitive, so we focus on comparing the computational cost of the BIF to classical influence
 399 functions approximated with EK-FAC (George et al., 2018).
 400

401 See Fig. 5 for benchmark results. For the choice of SGLD hyperparameters we use (2k total draws,
 402 or 2.5x fewer than in Figure 2), we observe that BIF scales better than EK-FAC in evaluation time.
 403 Further, notice that EK-FAC has a large up-front cost in time and storage associated to fitting the
 404 approximate inverse Hessian, independent of the query dataset size. This overhead is only justified if
 405 one wants to compute sufficiently many influence scores. See Appendix B.3 for further experiment
 406 details and Appendix D.2 for a direct comparison of the results.
 407



419 **Figure 5: Scaling comparison of BIF and EK-FAC** across model sizes of the Pythia model suite.
 420 (*Left*) Evaluation time, excluding the tokenization time. (*Right*) The node’s (4xA100) peak GPU
 421 RAM usage. For the largest models, the BIF is 2 orders of magnitude faster, while using the same
 422 GPU RAM as the EK-FAC.

423

424 5 RELATED WORK

425

426

Influence functions and training data attribution. Training data attribution (TDA) approaches
 427 can be broadly categorized into three families. The most direct approach involves retraining, which
 428 serves as the gold standard for measuring influence but is computationally prohibitive for large-scale
 429 deep neural networks (DNNs). A second family of methods relies on similarities in the model’s
 430 representation space, using intermediate activations to connect training and query points (Park et al.,
 431 2023b).

432 The third, and most relevant, family for our work uses gradient-based information. A prominent
 433 example is the classical influence function, a well-studied technique from robust statistics (Hampel,
 434 1974; Cook, 1977; Cook & Weisberg, 1982). Applying this technique directly to DNNs is infeasible,
 435 as it requires inverting the Hessian matrix. Consequently, much prior work has focused on developing
 436 tractable approximations to the inverse-Hessian-vector product (Koh & Liang, 2020; Grosse et al.,
 437 2023; Park et al., 2023b). Other gradient-based strategies approximate TDA by differentiating through
 438 the optimizer steps of the training process itself (Bae et al., 2024). These “unrolling” techniques
 439 come at the cost of requiring access to multiple checkpoints saved along the training trajectory.

440
 441
 442 **Distributional training data attribution.** Recent work has recognized that training data attribution
 443 should account for the stochastic nature of training. Mlodozeniec et al. (2025) formalize this with
 444 distributional training data attribution (d-TDA), which frames the goal of influence as predicting
 445 how the *distribution* over trained models changes when data is removed. Their framework is
 446 deliberately general, accommodating arbitrary distribution families and distance metrics (Wasserstein,
 447 KL divergence, mean shift, etc.).

448 Our local BIF can be viewed as a particular instantiation of “mean-shift” d-TDA with a tempered
 449 Gibbs posterior. These specific choices enable us to apply the covariance identity (Eq. 4) that unlocks
 450 our novel SGMCMC-based methodology. This covariance-based definition has appeared previously
 451 as the “Bayesian Infinitesimal Jackknife” (Giordano & Broderick, 2024; Iba, 2025) in the context of
 452 Bayesian model analysis. However, to our knowledge, we are the first to formulate a *local* BIF and
 453 scale these distributional methods to large-scale language models trained using standard stochastic
 454 optimization.

455
 456 **Singular learning theory and developmental interpretability.** Our work is grounded in singular
 457 learning theory (SLT), which provides a mathematical framework for analyzing the geometry of
 458 loss landscapes in non-identifiable “singular” models like DNNs (Watanabe, 2009). The BIF builds
 459 directly on recent methods for estimating localized SLT observables for a single model checkpoint.
 460 Specifically, Lau et al. (2025) introduced an SGMCMC-based estimator for an SLT quantity known as
 461 the local learning coefficient (LLC) by sampling from a “localized posterior”—the same mechanism
 462 we use to define our local BIF (Equation 5). Our local BIF is related to the *local susceptibilities*
 463 recently introduced by Baker et al. (2025). Together, these methods form part of a broader “develop-
 464 mental interpretability” research agenda, which uses tools from statistical physics and SLT to probe
 465 how data shapes the learned representations and local geometry of neural networks (Pepin Lehalleur
 466 et al., 2025).

467 6 DISCUSSION & CONCLUSION

468
 469 We introduce the local Bayesian influence function (BIF), a novel training data attribution (TDA)
 470 method that replaces the ill-posed Hessian inversion of classical influence functions with scalable
 471 SGMCMC-based covariance estimation. Our results demonstrate that this approach is not just
 472 theoretically sound but practically effective. In qualitative comparisons on large language models,
 473 the BIF produces interpretable, fine-grained attributions. Quantitatively, it achieves state-of-the-art
 474 performance on retraining benchmarks, matching strong baselines like EK-FAC in realistic data
 475 intervention scenarios.

476 **Advantages.** The BIF framework offers several fundamental advantages over classical, Hessian-
 477 based methods. By design, it avoids the need to compute or invert the Hessian, making it naturally
 478 applicable to the singular loss landscapes of deep neural networks where the classical influence
 479 function is ill-defined. The underlying SGMCMC sampling is model-agnostic and can be applied to
 480 any differentiable architecture. Furthermore, its definition is not restricted to local minima, allowing
 481 for the analysis of models at any point during training.

482
 483 **Limitations and practical trade-offs.** The primary limitation of the BIF lies in the practical trade-
 484 offs of its computational cost. While it avoids the high up-front fitting cost of methods like EK-FAC,
 485 its cost scales with the number of posterior draws, each requiring forward passes over the attribution
 and query sets. However, this may not be a fundamental barrier; advanced covariance estimators

486 could potentially reduce the number of required forward passes significantly without compromising
 487 estimation quality. Additionally, the method’s performance is sensitive to the hyperparameters of the
 488 SGLD sampler (ϵ, γ, β) and the total number of posterior draws, and this dependence is still not fully
 489 understood.
 490

491 **Future directions.** Our work opens several promising avenues for future research. A direct path is
 492 the exploration of more advanced MCMC samplers to improve the efficiency of covariance estimation.
 493 Furthermore, the role of the BIF’s hyperparameters can be explored further; the localization strength
 494 γ and inverse temperature β can be viewed not just as parameters to be tuned, but as analytical tools
 495 to probe influence at different scales and resolutions of the loss landscape. Finally, because the local
 496 BIF is well-defined at any model checkpoint, it enables the study of how data influence *evolves* over
 497 the course of training. This opens the door to dynamic data attribution, tracing how certain examples
 498 become more or less critical at different stages of learning.
 499

500 In conclusion, the local BIF reframes data attribution from a point-estimate problem to a distributional
 501 one. This perspective provides a more robust, scalable, and theoretically grounded path toward
 502 understanding how individual data points shape the behavior of complex deep learning models.
 503

REPRODUCIBILITY STATEMENT

504 To ensure our work is reproducible, we provide detailed descriptions of our methodology throughout
 505 the paper and its appendices. The core SGLD-based estimation procedure for the local BIF is formally
 506 presented in Algorithm 1. All experiments were conducted on public datasets (CIFAR-10, ImageNet,
 507 The Pile) and standard model architectures (ResNet-9, Inception-V1, Pythia), as described in our
 508 results (Section 4). A complete summary of the SGLD hyperparameters used for each experiment is
 509 available in Table 3, with further implementation details discussed in Appendix B.1. The setup for our
 510 retraining experiments, including the LDS evaluation protocol and model training hyperparameters,
 511 is detailed in Appendix C. Finally, the specifics of our scaling analysis, computational environment,
 512 and comparison against the EK-FAC baseline can be found in Section 4.3 and Appendix B.3.
 513

LLM USAGE STATEMENT

514 We used Large Language Models (LLMs) to assist with writing, coding, and theory in this paper. Their
 515 role included improving the text’s clarity and structure, helping to implement code for experiments
 516 and figures, and assisting in derivations (such as the BIF asymptotically recovering the classical IF,
 517 see Appendix A.1). All AI-generated content was reviewed and validated by the authors, who retain
 518 full responsibility for this work.
 519

REFERENCES

520 Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine
 521 learning in linear time. *J. Mach. Learn. Res.*, 18(1):4148–4187, January 2017. ISSN 1532-4435.
 522
 523 Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B. Grosse. Training data attribution
 524 via approximate unrolling. In Amir Globersons, Lester Mackey, Danielle Belgrave, An-
 525 gela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in
 526 Neural Information Processing Systems 38: Annual Conference on Neural Information
 527 Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15,
 528 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7af60ccb99c7a434a0d9d9c1fb00ca94-Abstract-Conference.html.
 529
 530 Garrett Baker, George Wang, Jesse Hoogland, and Daniel Murfet. Structural inference: Studying
 531 small language models with susceptibilities, April 2025. URL <http://arxiv.org/abs/2504.18274>. arXiv:2504.18274 [cs].
 532
 533 Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan,
 534 Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron,
 535 Lintang Sutawika, and Oskar Van Der Wal. Pythia: a suite for analyzing large language models
 536
 537

540 across training and scaling. In *Proceedings of the 40th International Conference on Machine*
 541 *Learning*, ICML'23. JMLR.org, 2023.

542

543 Tyler A Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable Influence
 544 and Fact Tracing for Large Language Model Pretraining. 2025.

545 Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input Similarity from
 546 the Neural Network Perspective. In *Advances in Neural Information Processing Systems*, vol-
 547 umne 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/c61f571dbd2fb949d3fe5ae1608dd48b-Abstract.html>.

548

549 Zhongtian Chen and Daniel Murret. Modes of sequence models and learning coefficients, 2025. URL
 550 <https://arxiv.org/abs/2504.18048>.

551

552 R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics : a*
553 journal of statistics for the physical, chemical, and engineering sciences, February 1977. ISSN
 554 0040-1706. URL <https://www.tandfonline.com/doi/abs/10.1080/00401706.1977.10489493>. tex.copyright: Copyright Taylor and Francis Group, LLC.

555

556 R. Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. Monographs on
 557 statistics and applied probability. Chapman and Hall, New York, 1982. ISBN 0-412-24280-0. URL
 558 <https://hdl.handle.net/11299/37076>.

559

560 Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hier-
 561 archical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*,
 562 pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

563

564 Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason
 565 Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile:
 566 An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021. URL
 567 <https://arxiv.org/abs/2101.00027>.

568

569 Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast
 570 approximate natural gradient descent in a kronecker factored eigenbasis. In S. Bengio,
 571 H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Ad-*
572 vances in Neural Information Processing Systems, volume 31. Curran Associates, Inc.,
 573 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/48000647b315f6f00f913caa757a70b3-Paper.pdf.

574

575 Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net opti-
 576 mization via Hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov
 577 (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of
Proceedings of Machine Learning Research, pp. 2232–2241. PMLR, 09–15 Jun 2019. URL
<https://proceedings.mlr.press/v97/ghorbani19b.html>.

578

579 Ryan Giordano and Tamara Broderick. The Bayesian infinitesimal jackknife for variance, 2024. URL
<https://arxiv.org/abs/2305.06466>.

580

581 Ryan Giordano, Tamara Broderick, and Michael I. Jordan. Covariances, robustness, and variational
 582 Bayes. *Journal of Machine Learning Research*, 19:51:1–51:49, 2017. URL <https://api.semanticscholar.org/CorpusID:53238793>.

583

584 Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit
 585 Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilé Lukošiūtė, Karina Nguyen,
 586 Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying Large
 587 Language Model Generalization with Influence Functions, August 2023. URL <http://arxiv.org/abs/2308.03296>. arXiv:2308.03296 [cs].

588

589 Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statis-*
590 tical Association, 69(346):383–393, 1974. doi: 10.1080/01621459.1974.10482962. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1974.10482962>.

591

592 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image
 593 recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.

594 Rohan Hitchcock and Jesse Hoogland. From Global to Local: A Scalable Benchmark for Local
 595 Posterior Sampling, July 2025.

596

597 Yukito Iba. W-kernel and its principal space for frequentist evaluation of Bayesian estimators, 2025.
 598 URL <https://arxiv.org/abs/2311.13017>.

599 Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and
 600 Generalization in Neural Networks, February 2020. URL <http://arxiv.org/abs/1806.07572>. arXiv:1806.07572 [cs].

601

602 Keller Jordan. 94% on CIFAR-10 in 3.29 seconds on a single GPU. *CoRR*, abs/2404.00498, 2024.
 603 doi: 10.48550/ARXIV.2404.00498. URL <https://doi.org/10.48550/arXiv.2404.00498>.

604

605 Jean Kaddour. The MiniPile challenge for data-efficient language models. *arXiv preprint*
 606 *arXiv:2304.08442*, 2023.

607

608 Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions,
 609 December 2020. URL <http://arxiv.org/abs/1703.04730>. arXiv:1703.04730 [stat]
 610 CitationKey: deep-influence-functions.

611

612 A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Univ. Toronto,
 613 2009.

614

615 Edmund Lau, Zach Furman, George Wang, Daniel Murfet, and Susan Wei. The local learning
 616 coefficient: a singularity-aware complexity measure. In *The 28th international conference on*
 617 *artificial intelligence and statistics*, 2025. URL <https://openreview.net/forum?id=1av51Z1suL>.

618

619 Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned Stochastic
 620 Gradient Langevin Dynamics for Deep Neural Networks, December 2015. URL <http://arxiv.org/abs/1512.07666> [stat].

621

622 Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate
 623 Bayesian inference. *J. Mach. Learn. Res.*, 18(1):4873–4907, January 2017. ISSN 1532-4435.

624

625 James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate
 626 curvature. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference*
 627 *on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417,
 628 Lille, France, 07–09 Jul 2015. PMLR. URL <https://proceedings.mlr.press/v37/martens15.html>.

629

630 James Martens and Roger Grosse. Optimizing Neural Networks with Kronecker-factored
 631 Approximate Curvature, June 2020. URL <http://arxiv.org/abs/1503.05671>.
 632 arXiv:1503.05671 [cs].

633

634 Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A. Louis. Is SGD a Bayesian sampler?
 635 well, almost. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

636

637 Bruno Mlodzeniec, Isaac Reid, Sam Power, David Krueger, Murat Erdogdu, Richard E. Turner, and
 638 Roger Grosse. Distributional Training Data Attribution: What do Influence Functions Sample?,
 639 October 2025. URL <https://arxiv.org/abs/2506.12965>. arXiv:2506.12965 [cs].

640

641 Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK:
 642 Attributing Model Behavior at Scale, April 2023a. URL <http://arxiv.org/abs/2303.14186>.
 643 arXiv:2303.14186 [stat].

644

645 Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK:
 646 Attributing model behavior at scale, April 2023b. URL <http://arxiv.org/abs/2303.14186>.
 647 arXiv:2303.14186 [stat] CitationKey: TRAK.

648

649 Simon Pepin Lehalleur, Jesse Hoogland, Matthew Farrugia-Roberts, Susan Wei, Alexander Gi-
 650 etelink Oldenziel, George Wang, Liam Carroll, and Daniel Murfet. You are what you eat—AI
 651 alignment requires understanding how data shapes structure and generalisation. *arXiv preprint*
 652 *arXiv:2502.05475*, 2025.

648 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,
 649 Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified
 650 text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL
 651 <http://jmlr.org/papers/v21/20-074.html>.

652 Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic
 653 development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):
 654 11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1820226116>.

655 Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov,
 656 Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In
 657 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA,
 658 June 7-12, 2015*, pp. 1–9. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298594. URL
 659 <https://doi.org/10.1109/CVPR.2015.7298594>.

660 Stan van Wingerden, Jesse Hoogland, George Wang, and William Zhou. DevInterp, 2024. URL
 661 <https://github.com/timaeus-research/devinterp>.

662 Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila A. McIlraith, and Roger Grosse.
 663 Better Training Data Attribution via Better Inverse Hessian-Vector Products, July 2025a. URL
 664 <http://arxiv.org/abs/2507.14740>. arXiv:2507.14740 [cs].

665 George Wang, Jesse Hoogland, Stan van Wingerden, Zach Furman, and Daniel Murfet. Differentiation
 666 and Specialization of Attention Heads via the Refined Local Learning Coefficient. In *Proceedings
 667 of The 13th International Conference on Learning Representations*, 2025b.

668 Sumio Watanabe. *Algebraic Geometry and Statistical Learning Theory*. Cambridge Monographs on
 669 Applied and Computational Mathematics. Cambridge University Press, 2009.

670 Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. Deep
 671 learning is singular, and that’s good. *IEEE Transactions on Neural Networks and Learning Systems*,
 672 34(12):10473–10486, 2023. doi: 10.1109/TNNLS.2022.3167409.

673 Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics.
 674 In *Proceedings of the 28th International Conference on International Conference on Machine
 675 Learning, ICML’11*, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

676
 677
 678
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701

APPENDIX

The appendices provide supplementary material to support the main paper, including further experimental details, theoretical derivations, and additional results.

- Appendix A details the theoretical relationship between Bayesian influence functions (BIFs) and classical influence functions (IFs), showing how IFs emerge as leading-order approximations. [Appendix A.3 compares the BIF against additional IF approximations besides EK-FAC.](#)
- Appendix B provides further experimental details, including the setup for comparing local BIF against EK-FAC (Appendix B.3) and the specifics of the SGLD estimator presented in Algorithm 1.
- Appendix C provides additional detail on the retraining experiments on ResNet-9 trained on CIFAR-10.
- Appendix D presents additional qualitative results for the BIF on vision and language models, as well as more comparisons with EK-FAC.

A RELATING BAYESIAN AND CLASSICAL INFLUENCE FUNCTIONS

A.1 RELATING THE BIF AND UNDAMPENED IFs

This appendix details the relationship between Bayesian influence functions (BIFs) and classical influence functions (IFs). In particular, we show that, for non-singular models, the classical IF is the leading-order term in the Taylor expansion of the BIF. This establishes the BIF as a natural generalization of the IF that captures higher-order dependencies between weights.

Let \mathbf{w}^* be a local minimum. In this section, all gradients and Hessians are evaluated at \mathbf{w}^* ; thus, to reduce notational clutter, we omit the dependence on \mathbf{w} . For any function $f(\mathbf{w})$, we denote its gradient at \mathbf{w}^* as $\mathbf{g}_f = \nabla_{\mathbf{w}} f(\mathbf{w}^*)$ and its Hessian as $\mathbf{H}_f = \nabla_{\mathbf{w}}^2 f(\mathbf{w}^*)$. In particular, $\mathbf{g}_\phi = \nabla_{\mathbf{w}} \phi(\mathbf{w}^*)$ and $\mathbf{H}_\phi = \nabla_{\mathbf{w}}^2 \phi(\mathbf{w}^*)$ for an observable $\phi(\mathbf{w})$; we also abbreviate $\mathbf{g}_i = \nabla_{\mathbf{w}} \ell_i(\mathbf{w}^*)$ and $\mathbf{H}_i = \nabla_{\mathbf{w}}^2 \ell_i(\mathbf{w}^*)$ for a per-sample loss $\ell_i(\mathbf{w})$. The total Hessian of the empirical risk $L_{\text{train}}(\mathbf{w}) = \sum_{k=1}^n \ell_k(\mathbf{w})$ at \mathbf{w}^* is denoted $\mathbf{H} = \sum_{k=1}^n \mathbf{H}_k$.

The Bayesian influence function (BIF) for the effect of sample \mathbf{z}_i on an observable ϕ is given by (see Equation (4)):

$$\text{BIF}(\mathbf{z}_i, \phi) = -\text{Cov}_{p(\mathbf{w} | \mathcal{D}_{\text{train}})}(\phi(\mathbf{w}), \ell_i(\mathbf{w})), \quad (7)$$

where the covariance is taken over the posterior $p(\mathbf{w} | \mathcal{D}_{\text{train}}) \propto \exp(-L_{\text{train}}(\mathbf{w}))\varphi(\mathbf{w})$, with $\varphi(\mathbf{w})$ being a prior. This definition is exact and makes no assumptions about the form of $\phi(\mathbf{w})$, $\ell_i(\mathbf{w})$, or $p(\mathbf{w} | \mathcal{D}_{\text{train}})$.

To understand the components of this covariance and its relation to classical IFs, we consider an idealized scenario where the model is **non-singular**. Under this strong assumption, which *does not hold for deep neural networks* (Wei et al., 2023), the posterior $p(\mathbf{w} | \mathcal{D}_{\text{train}})$ can be approximated by a Laplace approximation around \mathbf{w}^* :

$$p(\mathbf{w} | \mathcal{D}_{\text{train}}) \approx \mathcal{N}(\mathbf{w}^*, \mathbf{H}^{-1}). \quad (8)$$

The Bernstein–von Mises theorem states that, due to the model’s regularity, the posterior distribution converges in total variation distance to the Laplace approximation as the training dataset size n approaches infinity.

Let $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}^*$. Assuming analyticity, we can express $\phi(\mathbf{w})$ and $\ell_i(\mathbf{w})$ using their full Taylor series expansions around \mathbf{w}^* :

$$\phi(\mathbf{w}) = \phi(\mathbf{w}^*) + \mathbf{g}_\phi^\top \Delta\mathbf{w} + \frac{1}{2} \Delta\mathbf{w}^T \mathbf{H}_\phi \Delta\mathbf{w} + \sum_{k=3}^{\infty} \frac{1}{k!} D^k \phi(\mathbf{w}^*) [\Delta\mathbf{w}, \dots, \Delta\mathbf{w}], \quad (9)$$

$$\ell_i(\mathbf{w}) = \ell_i(\mathbf{w}^*) + \mathbf{g}_i^\top \Delta\mathbf{w} + \frac{1}{2} \Delta\mathbf{w}^T \mathbf{H}_i \Delta\mathbf{w} + \sum_{k=3}^{\infty} \frac{1}{k!} D^k \ell_i(\mathbf{w}^*) [\Delta\mathbf{w}, \dots, \Delta\mathbf{w}], \quad (10)$$

756 where $D^k f(\mathbf{w}^*)[\Delta \mathbf{w}, \dots, \Delta \mathbf{w}]$ denotes the k -th order differential of f at \mathbf{w}^* applied to k copies of
 757 $\Delta \mathbf{w}$.

758 The covariance under this Gaussian (Laplace) approximation, denoted $\text{Cov}_{\mathcal{N}}$, then involves covari-
 759 ances between all pairs of terms from these two expansions:

$$761 \quad 762 \quad 763 \quad \text{Cov}_{\mathcal{N}}(\phi(\mathbf{w}), \ell_i(\mathbf{w})) = \sum_{k=1}^{\infty} \sum_{m=1}^{\infty} \text{Cov}_{\mathcal{N}}(\text{Term}_k[\phi], \text{Term}_m[\ell_i]), \quad (11)$$

764 where $\text{Term}_k[f]$ is the k -th order term in the Taylor expansion of $f(\mathbf{w})$ in powers of $\Delta \mathbf{w}$. For
 765 $\Delta \mathbf{w} \sim \mathcal{N}(0, \mathbf{H}^{-1})$, the leading terms are:

- 766 • Covariance of linear terms ($k = 1, m = 1$):

$$768 \quad 769 \quad \text{Cov}_{\mathcal{N}}(\mathbf{g}_\phi^T \Delta \mathbf{w}, \mathbf{g}_i^T \Delta \mathbf{w}) = \mathbf{g}_\phi^T \mathbf{H}^{-1} \mathbf{g}_i.$$

- 770 • Covariance of quadratic terms ($k = 2, m = 2$):

$$771 \quad 772 \quad 773 \quad \text{Cov}_{\mathcal{N}}\left(\frac{1}{2}(\Delta \mathbf{w})^\top \mathbf{H}_\phi \Delta \mathbf{w}, \frac{1}{2} \Delta \mathbf{w}^\top \mathbf{H}_i \Delta \mathbf{w}\right) = \frac{1}{2} \text{tr}(\mathbf{H}_\phi \mathbf{H}^{-1} \mathbf{H}_i \mathbf{H}^{-1}).$$

774 (Using Isserlis' theorem for moments of Gaussians).

- 775 • Cross-terms between odd and even order terms (e.g., $k = 1, m = 2$) are zero due to the
 776 symmetry of Gaussian moments.

777 Thus, the BIF under these regularity and Laplace approximations becomes:

$$779 \quad 780 \quad 781 \quad 782 \quad 783 \quad \text{BIF}(\mathbf{z}_i, \phi) \approx -\mathbf{g}_\phi^T \mathbf{H}^{-1} \mathbf{g}_i - \frac{1}{2} \text{tr}(\mathbf{H}_\phi \mathbf{H}^{-1} \mathbf{H}_i \mathbf{H}^{-1}) - \sum_{\substack{k, m \geq 1 \\ \text{not (1,1) or (2,2)} \\ k+m \text{ is even}}} \text{Cov}_{\mathcal{N}}(\text{Term}_k[\phi], \text{Term}_m[\ell_i]). \quad (12)$$

784 The leading term $-\mathbf{g}_\phi^T \mathbf{H}^{-1} \mathbf{g}_i = -\nabla_{\mathbf{w}} \phi(\mathbf{w}^*)^\top \mathbf{H}_{\mathbf{w}^*}^{-1} \nabla_{\mathbf{w}} \ell_i(\mathbf{w}^*)$ is precisely the classical influence
 785 function $\text{IF}(\mathbf{z}_i, \phi)$ from Equation (2). Note that \mathbf{H} scales linearly in n , so this term dominates as
 786 $n \rightarrow \infty$. The BIF formulation, when analyzed via Laplace approximation, naturally includes this
 787 term and also explicitly shows a second-order correction involving products of the Hessians of the
 788 loss and observable. More generally, the exact BIF definition (Equation (7)) encapsulates all such
 789 higher-order dependencies without truncation, which are only partially revealed by this expansion
 790 under the (invalid for neural networks) Laplace approximation.

791 A.2 RELATING THE LOCALIZED BIF AND DAMPED IFs

793 We now extend this analysis to the local BIF, showing that its leading-order term is precisely the
 794 damped classical IF, which is the standard practical remedy for the singular Hessians found in deep
 795 neural networks.

796 The local BIF is defined over the localized posterior from Equation (5):

$$798 \quad 799 \quad 800 \quad 801 \quad 802 \quad p_\gamma(\mathbf{w} \mid \mathcal{D}_{\text{train}}, \mathbf{w}^*) \propto \exp\left(-\sum_{k=1}^n \ell_k(\mathbf{w}) - \frac{\gamma}{2} \|\mathbf{w} - \mathbf{w}^*\|_2^2\right) \\ = \exp\left(-\left(L_{\text{train}}(\mathbf{w}) + \frac{\gamma}{2} \|\mathbf{w} - \mathbf{w}^*\|_2^2\right)\right). \quad (13)$$

803 This distribution is centered around \mathbf{w}^* due to the localizing potential (the quadratic term). To apply
 804 the Laplace approximation, we consider the mode of this distribution, which is the minimum of the
 805 effective potential $L_{\text{eff}}(\mathbf{w}) = L_{\text{train}}(\mathbf{w}) + \frac{\gamma}{2} \|\mathbf{w} - \mathbf{w}^*\|_2^2$. We assume \mathbf{w}^* to be a local minimum
 806 of $L_{\text{train}}(\mathbf{w})$, so $\nabla L_{\text{train}}(\mathbf{w}^*) = 0$. Consequently, $\nabla L_{\text{eff}}(\mathbf{w}^*) = \nabla L_{\text{train}}(\mathbf{w}^*) + \gamma(\mathbf{w}^* - \mathbf{w}^*) = 0$,
 807 meaning \mathbf{w}^* is also the mode of the localized posterior.

808 The precision of the Laplace approximation is given by the Hessian of this effective potential evaluated
 809 at \mathbf{w}^* :

$$810 \quad \mathbf{H}_{\text{eff}} = \nabla^2 L_{\text{eff}}(\mathbf{w}^*) = \nabla^2 L_{\text{train}}(\mathbf{w}^*) + \gamma \mathbf{I} = \mathbf{H} + \gamma \mathbf{I}. \quad (14)$$

810 Therefore, the Laplace approximation for the localized posterior is a Gaussian centered at \mathbf{w}^* with
 811 covariance $\mathbf{H}_{\text{eff}}^{-1}$:

$$p_{\gamma}(\mathbf{w} \mid \mathcal{D}_{\text{train}}, \mathbf{w}^*) \approx \mathcal{N}(\mathbf{w}^*, (\mathbf{H} + \gamma \mathbf{I})^{-1}). \quad (15)$$

814 Following the same Taylor expansion logic as in the previous section, we can compute the leading-
 815 order term of the covariance between $\phi(\mathbf{w})$ and $\ell_i(\mathbf{w})$ under this Gaussian approximation:

$$\text{Cov}_{\gamma}(\phi(\mathbf{w}), \ell_i(\mathbf{w})) \approx \text{Cov}_{\mathcal{N}}(\mathbf{g}_{\phi}^{\top} \Delta \mathbf{w}, \mathbf{g}_i^{\top} \Delta \mathbf{w}) = \mathbf{g}_{\phi}^{\top} (\mathbf{H} + \gamma \mathbf{I})^{-1} \mathbf{g}_i. \quad (16)$$

818 The local BIF is the negative of this covariance:

$$\text{BIF}_{\gamma}(\mathbf{z}_i, \phi) \approx -\mathbf{g}_{\phi}^{\top} (\mathbf{H} + \gamma \mathbf{I})^{-1} \mathbf{g}_i. \quad (17)$$

821 This expression is exactly the form of the classical damped influence function, where the localiza-
 822 tion strength γ serves as the dampening coefficient. This shows that the local BIF's leading-order
 823 term under a Laplace approximation is the damped IF.

824 Just as the global BIF generalizes the classical IF, the local BIF is a natural, higher-order generalization
 825 of the damped IF, capturing dependencies beyond the second-order approximation while remaining
 826 well-defined and computable for the singular models used in modern deep learning.

A.3 COMPARING THE BIF AND IF APPROXIMATIONS

831 As discussed in Section 2.1, classical influence functions face significant computational challenges
 832 when applied to deep neural networks because the memory footprint of the inverse Hessian grows
 833 quadratically with model size. This motivates a variety of approximation strategies that make different
 834 trade-offs between accuracy, computational cost, and generality. Below, we detail a selection of
 835 methods that are representative of the current dominant approaches to large-scale influence function
 836 approximation. These are roughly in decreasing order of approximation fidelity, from EK-FAC (and
 837 ASTRA), to TRAK (and TrackStar), and finally to GradSim.

839 **EK-FAC.** Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC; Grosse
 840 et al. 2023) approximates the Hessian using a Kronecker-factored structure, originally developed for
 841 efficient natural gradient descent (Martens & Grosse, 2020). The key insight is to approximate the
 842 Fisher information matrix (equivalent to the Gauss-Newton Hessian for the cross-entropy loss) as a
 843 block-diagonal matrix where each block corresponds to a layer, and each block is further factored
 844 as a Kronecker product of two smaller matrices. This factorization dramatically reduces the cost of
 845 inversion. EK-FAC further improves upon standard K-FAC by computing an eigenvalue correction
 846 in the Kronecker-factored eigenbasis (George et al., 2018). While highly effective, EK-FAC is
 847 restricted to linear and convolutional layers, excluding attention and normalization layers in modern
 848 architectures. Additionally, it requires an expensive fit phase to estimate and invert the Kronecker
 849 factors, though this cost amortizes when computing influence for many query-training pairs.

850 Recent work has sought to bridge the gap between these efficient parametric approximations and exact
 851 solvers. ASTRA (Wang et al., 2025a) utilizes the EK-FAC decomposition not as a final estimator,
 852 but as a preconditioner for Stochastic Neumann Series iterations. This hybrid approach corrects the
 853 structural biases of the block-diagonal approximation by refining the estimate iteratively. However,
 854 this improved precision comes at an increased computational cost, requiring hundreds of additional
 855 iterative updates per query to converge beyond the initial EK-FAC solution.

858 **TRAK.** TRAK (Tracing with the Randomly-projected After Kernel; Park et al. 2023a) addresses
 859 the scalability of gradient-based attribution by linearizing the model output function, effectively
 860 approximating the model with its empirical Neural Tangent Kernel (eNTK) (Jacot et al., 2020; Park
 861 et al., 2023a). To handle the high dimensionality of the parameter space, TRAK projects the resulting
 862 gradient vectors into a lower-dimensional space using random projections, preserving inner products
 863 with high probability. Unlike simple similarity methods, TRAK then reweights these projected
 864 gradients by an approximate inverse covariance matrix to account for the local curvature of the loss

864 landscape. Finally, to handle the stochasticity of non-convex training, TRAK ensembles these scores
 865 across multiple models trained on random subsets of the data.

866
 867 Most recently, TrackStar (Chang et al., 2025) has pushed gradient-based attribution to the full scale
 868 of LLM pretraining (e.g., 8B parameters over 160B tokens) without the data subsampling required
 869 by EK-FAC and the BIF. TrackStar can be seen as a refinement of the projection-based approach
 870 of TRAK that uses a different gradient and incorporates optimizer second-moment corrections and
 871 task-specific Hessian approximations. Enabling retrieval across the entire pretraining corpus shifts
 872 the bottleneck from compute to storage: the method relies on building indices of projected gradients
 873 for every training example, which can require up to 87TB of storage for datasets like C4 (Raffel et al.,
 874 2020). This represents the state-of-the-art for coverage, but the immense infrastructure requirement
 875 for storing and retrieving these indices puts it in a distinct resource class compared to methods that
 876 approximate influence using data subsets or on-the-fly batching.

877 These methods thus represent a level of fidelity between EK-FAC/ASTRA and GradSim: they retain
 878 a notion of geometric correction through reweighting, but apply it within a compressed projected
 879 space rather than the full parameter space.

880
 881 **GradSim.** Gradient Similarity (GradSim) represents the most aggressive simplification of classical
 882 IFs: it drops the Hessian inverse entirely and computes influence as the raw inner product between
 883 loss gradients (Charpiat et al., 2019):

$$\text{GradSim}(z_i, z_j) = \nabla \ell_j(\mathbf{w}^*) \cdot \nabla \ell_i(\mathbf{w}^*). \quad (18)$$

884 The intuition is that samples with aligned gradients push the model’s parameters in similar directions,
 885 suggesting they teach similar patterns. While computationally efficient and architecture-agnostic,
 886 GradSim discards all second-order curvature information captured by the Hessian inverse. This
 887 makes it less accurate than methods that account for the loss landscape geometry, though it remains a
 888 useful baseline for its simplicity.

889
 890 **Comparison to the local BIF.** Table 2 summarizes the key properties of these methods compared
 891 to our local Bayesian influence function (BIF). The BIF occupies a unique position in this landscape:
 892 it is Hessian-free and architecture-agnostic like GradSim and TRAK, but captures higher-order
 893 geometry through its distributional formulation via covariance estimation over the local posterior.
 894 Unlike EK-FAC and TRAK, it requires no expensive fit phase, making it particularly efficient for
 895 fine-grained, targeted attribution tasks where the number of queries is relatively small. However, it
 896 does not amortize as well over many queries, as each SGMCMC draw must perform forward passes
 897 over both the training and query sets. The tradeoffs thus favor the BIF for large models on small
 898 datasets or when fine-grained per-token analysis is necessary.

902 B FURTHER EXPERIMENTAL DETAILS

903 B.1 SGLD ESTIMATOR FOR BAYESIAN INFLUENCE

904 See Algorithm 1 for the stochastic Langevin gradient dynamics estimator for the Bayesian influence
 905 in its most basic form. In practice, computation of train losses and observables is batched so as
 906 to take advantage of GPU parallelism. We also find that preconditioned variants of SGLD such as
 907 RMSprop-SGLD (Li et al., 2015) yield higher-quality results for a wider range of hyperparameters.
 908 We use an implementation provided by van Wingerden et al. (2024).

909 The SGLD update step described here, which is the one we use in our experiments, differs slightly
 910 from the presentation in the main text: we introduce a scalar inverse temperature β (separate from
 911 the per-sample perturbations β). Roughly speaking, the inverse temperature can be thought of as
 912 controlling the *resolution* at which we sample from the loss landscape geometry (Chen & Murfet,
 913 2025). An alternative viewpoint is that the effective dataset size of training by iterative optimization
 914 is not obviously the same as the training dataset size n used in the Bayesian setting; we scale by β
 915 to account for this difference. Hence, in practice, we combine βn as a single hyperparameter to be
 916 tuned.

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Table 2: Comparison of training data attribution methods. The BIF offers a unique combination of being Hessian-free, architecture-agnostic, and capturing higher-order geometry, though it is less efficient when amortizing over many queries compared to methods with fit phases.

Property	BIF	IF	EK-FAC	TRAK	GradSim
Hessian-free	✓	✗	✗	✓	✓
Architecture-agnostic	✓	✓	✗ [†]	✓	✓
Scales to > 1B params [*]	✓	✗	✓	✓	✓
No fit phase	✓	✓	✗	✗	✓
Amortizes over many queries	✗	✓	✓	✓	✓
Per-token (efficient)	✓	✗	✗	✗	✗
Higher-order geometry	✓	✗	✗	✗	✗

[†]Linear and Conv2D layers only

^{*}GradSim and EK-FAC scale to >1B parameters via batching (avoiding OOM), but incur high compute costs per query (re-running backprop). TRAK avoids this via projection.

Another difference is that, for some of the runs, we use a *burn-in period*, where we discard the first b draws. Finally, for some of the runs we perform “weight-restricted” posterior sampling (Wang et al., 2025b), where we compute posterior estimates over a subset of weights, rather than all weights. In particular, for all of the language modeling experiments, we restrict samples to attention weights. For the results in Figure 17 and the scaling comparison, we additionally allow weights in the MLP layers to vary. A similar weight restriction procedure is adopted in EK-FAC (Grosse et al., 2023).

943 **Algorithm 1** SGLD for Bayesian influence

944 **Input:** Initial model parameters $\mathbf{w}^* \in \mathcal{W}$, training dataset $\mathcal{D}_{\text{train}} = (\mathbf{z}_i)_{i=1}^n$, loss functions
945 $\ell_i := \ell(\mathbf{z}_i; \cdot) : \mathcal{W} \rightarrow \mathbb{R}$ for each $i \in [n]$, observables $\phi_j : \mathcal{W} \rightarrow \mathbb{R}$ for each $j \in [p]$, SGLD
946 hyperparameters β (inverse temperature), ϵ (step size), γ (localization), m (batch size), C (number
947 of chains), T (chain length)
948 **Output:** $\mathbf{B} = (\text{BIF}(\mathbf{z}_i, \phi_j))_{1 \leq i \leq n, 1 \leq j \leq m} \in \mathbb{R}^{n \times p}$
949 $\mathbf{L} \leftarrow \mathbf{0}_{n \times CT}, \mathbf{\Phi} \leftarrow \mathbf{0}_{p \times CT}$
950 **for** $1 \leq c \leq C$ **do**
951 $\mathbf{w} \leftarrow \mathbf{w}^*$
952 **for** $1 \leq t \leq T$ **do**
953 **for** $1 \leq i \leq n$ **do**
954 $\mathbf{L}_{i,(c-1)C+t} \leftarrow \ell_i(\mathbf{w})$ ▷ Compute train losses (can be batched)
955 **end for**
956 **for** $1 \leq j \leq p$ **do**
957 $\mathbf{\Phi}_{j,(c-1)C+t} \leftarrow \phi_j(\mathbf{w})$ ▷ Compute observables (can be batched)
958 **end for**
959 Sample random $\mathcal{B}_t \subseteq \mathcal{D}_{\text{train}}$ of size m
960 $\mathbf{w} \leftarrow \mathbf{w} - \frac{\epsilon}{2} \left(\frac{\beta n}{m} \sum_{k \in \mathcal{B}_t} \nabla_{\mathbf{w}} \ell_k(\mathbf{w}) + \gamma(\mathbf{w} - \mathbf{w}^*) \right) + \mathcal{N}(0, \epsilon)$ ▷ SGLD update
961 **end for**
962 **end for**
963 $\mathbf{B} \leftarrow \frac{1}{CT-1} \mathbf{L} \left(\mathbf{I}_{CT} - \frac{1}{CT} \mathbf{1}_{CT} \mathbf{1}_{CT}^\top \right)^2 \mathbf{\Phi}^\top$ ▷ Covariance between \mathbf{L} and $\mathbf{\Phi}$
964 **Return** \mathbf{B}

965
966
967 **B.2 BIF HYPERPARAMETERS**

968
969 Table 3 summarizes the hyperparameter settings for the BIF experiments. The hyperparameters refer
970 to the Algorithm 1: m is the batch size, C is the number of chains, T the number of draws per chain,
971 b is the number of burn-in steps, ϵ is the learning rate, β is the inverse temperature, and γ is the
localization strength. See Appendix B.1 for more details on each of these hyperparameters.

972
 973 Table 3: Summary of hyperparameter settings for BIF experiments. Hyperparameters are defined
 974 as follows: m is the number of samples per SGLD minibatches, C is the number of SGLD chains,
 975 T is the number of draws per chain, b is the number of burn-in steps, ϵ is the step-size, $n\beta$ is the
 976 effective number of samples that modifies the size of the gradient term in the SGLD step, and γ is the
 977 localization strength.

Experiment	\S	Dataset	m	C	T	b	ϵ	$n\beta$	γ
Vision	4	ImageNet	256	15	1000	10	1×10^{-4}	10	1000
Language	4	Pile	64	5	1000	100	8×10^{-7}	2000	7000
Scaling	4	Pile	32	4	500	0	5×10^{-6}	30	300
Retraining ResNet	C	CIFAR10	1024	4	100	0	1×10^{-5}	200	10000
Language	B	Pile	32	4	500	0	5×10^{-6}	30	300
Language	B	Pile	64	5	100	0	5×10^{-5}	30	300

985 986 B.3 COMPARING THE LOCAL BIF AGAINST EK-FAC

987
 988 We run all benchmarking experiments for both BIF and SGLD on a single node with $4 \times$ NVIDIA
 989 A100 GPUs. As given in Table 3, for the BIF estimation, we run SGLD with batch size $m = 32$,
 990 number of chains $C = 4$, number of draws per chain $T = 500$, learning rate $\epsilon = 5 \times 10^{-6}$, inverse
 991 temperature $n\beta = 30$, and localization strength $\gamma = 300$. These are fairly conservative values:
 992 especially for larger models, we observe interpretable results for smaller values of T . For the sake
 993 of comparability, however, we use the same hyperparameters throughout the benchmarking. Each
 994 sequence is padded or truncated to 150 tokens, and the model is set to `bfloat16` precision.

995
 996 We use the `kronfluence` package for EK-FAC computation (Grosse et al., 2023).¹ This package
 997 splits the influence computation into a fit and score step. The fit step prepares components of
 998 the approximate inverse Hessian and then the score step computes the influence scores from the
 999 components computed in the first step. The fit step is computationally expensive, but the results are
 1000 saved to the disk and can be recycled for any score computation. This results in a high up-front cost
 1001 and large disk usage, but low incremental cost.

1002
 1003 In the first step, the Hessian is approximated with the Fisher information matrix (or, equivalently in
 1004 our setting, the Gauss-Newton Hessian), which is obtained by sampling the model outputs on the
 1005 training data. Since the Pile, which is the dataset used for Pythia training, is too large to iterate over
 1006 in full, we approximate it by taking a representative subset of 1 000 000 data points, curated using
 1007 k -means clustering (Gao et al., 2021; Kaddour, 2023). Distributional shifts in the chosen dataset
 1008 alter the influence predictions of the EK-FAC. In general, the true training distribution is not publicly
 1009 available, therefore we consider the choice of training data as a kind of hyperparameter sensitivity in
 1010 Table 1. Moreover, we use the `extreme_memory_reduce` option of the `kronfluence` package
 1011 for both steps. Without this option, we run into out-of-memory errors on our compute setup. Among
 1012 other optimizations, this setting sets the precision of gradients, activation covariances, and fitted
 1013 lambda values to `bfloat16` and offloads parts of the computation to the CPU.

1014
 1015 The comparison is depicted in Figure 5. The fitting step creates a large overhead compared to the BIF,
 1016 which explains the increasing discrepancy with increasing model size. This overhead is only justified
 1017 if one wants to compute sufficiently many influence scores. Moreover, the BIF only saves the final
 1018 results, which are typically small. In contrast, the results of the fit step are saved to the disk, which
 1019 for the Pythia-2.8B model occupies 41 GiB.

1020 B.4 PER-TOKEN INFLUENCE

1021
 1022 Both the BIF and EK-FAC can compute per-token influences, but the interpretation differs. For
 1023 BIF, the influence of each token in a training example is measured on each token in the query. In
 1024 contrast, EK-FAC defines the “per-token influence” as the effect of each training token on the entire
 1025 query. We can recover the EK-FAC definition of per-token influence from BIF by summing over
 1026 the query tokens. In principle, EK-FAC could also be used to compute per-token influences in the

1027
 1028 ¹The corresponding github repository is available here: <https://github.com/pomonam/kronfluence>

1026 sense we use, but a naive implementation with backpropagation is prohibitively memory-intensive,
 1027 because the gradient contribution of each training label must be propagated separately to the weights.
 1028 Consequently, the backward pass requires memory proportional to the sequence length.
 1029

1030 C RETRAINING EXPERIMENTS

1032 In its original formulation, the classical influence function is motivated as measuring the effect of
 1033 each training data point on a *retrained* model. That is, for each $\mathbf{z}_i \in \mathcal{D}_{\text{train}}$, if the model is retrained
 1034 from initialization on the leave-one-out dataset $\mathcal{D}_{\text{train}} \setminus \{\mathbf{z}_i\}$, what is the effect on the observable ϕ ?
 1035

1036 C.1 LINEAR DATAMODELLING SCORE

1038 Both classical and Bayesian influence functions approximate the effect of \mathbf{z}_i 's exclusion from $\mathcal{D}_{\text{train}}$ as
 1039 *linear*. That is, given a subset $\mathcal{D} \subseteq \mathcal{D}_{\text{train}}$, write $\phi(\mathcal{D})$ as the value of the observable ϕ corresponding
 1040 to a model trained on \mathcal{D} :

$$1041 \phi_{\text{C}}(\mathcal{D}) := \phi(\mathbf{w}^*(\mathcal{D})), \quad \mathbf{w}^*(\mathcal{D}) \in \arg \min_{\mathbf{w} \in \mathcal{W}} \sum_{\mathbf{z}_i \in \mathcal{D}} \ell_i(\mathbf{w}).$$

1043 in the classical perspective and

$$1045 \phi_{\text{B}}(\mathcal{D}) := \mathbb{E}_{\mathbf{w} \sim p(\mathbf{w} | \mathcal{D})} [\phi(\mathbf{w})]$$

1046 in the Bayesian perspective. In either case, we approximate $\phi(\mathcal{D})$ as linear in the set \mathcal{D} :

$$1048 \phi(\mathcal{D}) \approx \sum_{i=1}^n \tau_i [\mathbf{z}_i \in \mathcal{D}],$$

1051 where each $\tau_i \in \mathbb{R}$ is a training data attribution measure associated to \mathbf{z}_i and ϕ , e.g. $\text{IF}(\mathbf{z}_i, \phi)$ or
 1052 $\text{BIF}(\mathbf{z}_i, \phi)$.

1053 This linear approximation motivates the *linear datamodelling score* (LDS), introduced by Park
 1054 et al. (2023b). Given the training dataset $\mathcal{D}_{\text{train}}$ of cardinality n and a query set $\mathcal{D}_{\text{query}}$, we let the
 1055 query losses $(\phi_{\mathbf{z}_j} = \ell(\mathbf{z}_j; -))_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}}$ be our observables and suppose we are given TDA measures
 1056 $(\tau_{\mathbf{z}_j})_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}}$, with each $\tau_{\mathbf{z}_j} \in \mathbb{R}^n$. To measure the LDS of $(\tau_{\mathbf{z}_j})_{\mathbf{z}_j}$, we subsample datasets $\{\mathcal{D}_k\}_{k=1}^K$
 1057 with each $\mathbf{z}_i \in \mathcal{D}_k$ with probability $\alpha_{\text{retrain}} \in \{0.1, 0.3, 0.5, 0.7\}$ iid. (For our experiments, we set
 1058 $K = 100$). The LDS of $(\tau_{\mathbf{z}_j})_{\mathbf{z}_j}$ is then the average over $1 \leq k \leq K$ of the correlation between the
 1059 true retrained observable and the linear approximation from $(\tau_{\mathbf{z}_j})_{\mathbf{z}_j}$:

$$1060 \text{LDS}((\tau_{\mathbf{z}_j})_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}}; (\phi_{\mathbf{z}_j})_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}}, \{\mathcal{D}_k\}_{k=1}^K) \\ 1061 = \frac{1}{K} \sum_{k=1}^K \rho_s \left((\phi_{\text{C}, \mathbf{z}_j}(\mathcal{D}_k))_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}}, \left(\sum_{i=1}^n \tau_{\mathbf{z}_j, i} [\mathbf{z}_i \in \mathcal{D}_k] \right)_{\mathbf{z}_j \in \mathcal{D}_{\text{query}}} \right),$$

1065 where ρ_s is Spearman's rank correlation coefficient. Each $\phi_{\text{C}, \mathbf{z}_j}(\mathcal{D}_k)$ is computed by retraining the
 1066 model on \mathcal{D}_k and evaluating the loss on \mathbf{z}_j . Note that, regardless of whether we evaluate the LDS of
 1067 an approximate classical IF or the BIF, we use the classical version of the retrained observable ϕ_{C} .
 1068 We expect the BIF to perform well on this metric under the hypothesis that retraining with stochastic
 1069 gradient methods approximates Bayesian inference (Mandt et al., 2017; Mingard et al., 2021).
 1070

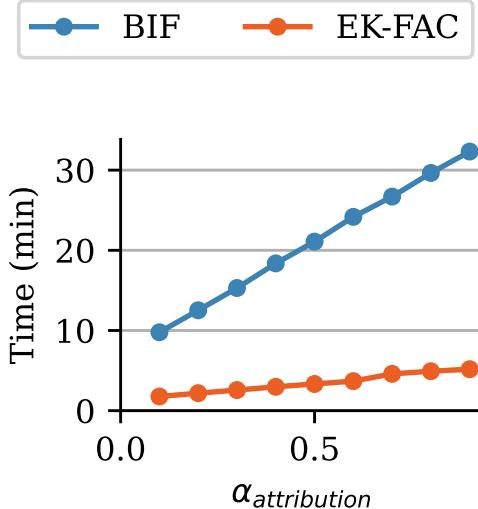
1071 C.2 LDS EXPERIMENT DETAILS AND RESULTS

1073 We evaluate the LDS of the EK-FAC, BIF, GradSim, TRAK on a ResNet-9 model with 1 972 792
 1074 parameters (He et al., 2015) trained on the CIFAR-10 (Krizhevsky, 2009) image classification
 1075 dataset. To minimize resource usage, we adopt the modified ResNet-9 architecture and training
 1076 hyperparameters described by Jordan (2024). In addition, we set aside a warmup set $\mathcal{D}_{\text{warmup}}$ of 2500
 1077 images. Before the actual training runs, we perform a short warmup phase on $\mathcal{D}_{\text{warmup}}$ to prime the
 1078 optimizer state. The training hyperparameters are summarized in Tab. 4.

1079 As described in Appendix C.1, we evaluate LDS by re-training the ResNet-9 100 times from
 initialization on random subsamples of the full CIFAR-10 training set, excluding the warmup

1080	Hyperparameter	Image Classification	Word Prediction (NLP)
1081	Training algorithm	SGD	AdamW
1082	Epochs	1 (8)	3
1083	Batch size	1024	256
1084	Momentum	0.85	$\beta_1 = 0.9$
1085	Weight decay	0.0153	0.01
1086	Learning rate	10.0	3×10^{-5}
1087	Warmup steps	100	–
1088	Label smoothing	0.2	0.0
1089	Bias scaler	64.0	–
1090	Whiten bias epochs	False	False
1091	Gradient accumulation steps	1	1

1092 Table 4: Training hyperparameters for retraining experiments. The foundational ResNet-9 model used
1093 to compute TDA scores was trained for 8 epochs. The retrained image-classification models were
1094 trained for a single epoch. For the next-token-prediction task, we used the pretrained Pythia-14m
1095 model.



1116 **Figure 6: Wall-clock time** as a function of $\alpha_{\text{attribution}}$ for BIF and EK-FAC in the retraining experiments.
1117 Owing to the small model sizes, EK-FAC runs approximately five times faster.

1120 set ($n = 47500$ images). Each subsample contains $n_{\text{retrain}} = \alpha_{\text{retrain}} \alpha_{\text{attribution}} n$ images. We
1121 then use the full test set ($q = 10000$ images) as the query set, i.e., there are 10000 observables,
1122 corresponding to the losses on each test image. Thus, both EK-FAC and BIF TDA scores comprise a
1123 $n_{\text{attribution}} \times 10000$ matrix. The hyperparameters for the SGLD estimation of the BIF are given in
1124 Tab. 3. For EK-FAC, we set the dampening factor to 10^{-8} . Both TDA techniques are computed on
1125 a single model checkpoint trained with the hyperparameters listed in Tab. 4. Figure 6 displays the
1126 wall-clock times of the BIF and EK-FAC computation. In these experiments, EK-FAC is around
1127 five times faster than the BIF. This advantage is largely due to the small model sizes ($\sim 2 \times 10^6$
1128 parameters), which results in a short fitting stage.

1129 We repeat the entire experimental pipeline (retraining of models, BIF, EK-FAC, TRAK, GradSim)
1130 five times with fixed hyperparameters and distinct initial seeds for the random number generators.
1131 From these five runs, we compute the mean LDS score and the standard error. The LDS scores of
1132 each individual run are displayed in Figure 7. The local BIF, EK-FAC, and GradSim are consistent
1133 with each other within each seed. However, the LDS score varies substantially across seeds. This
suggets either that the LDS score is not a reliable quantitative measure for evaluating TDA methods,

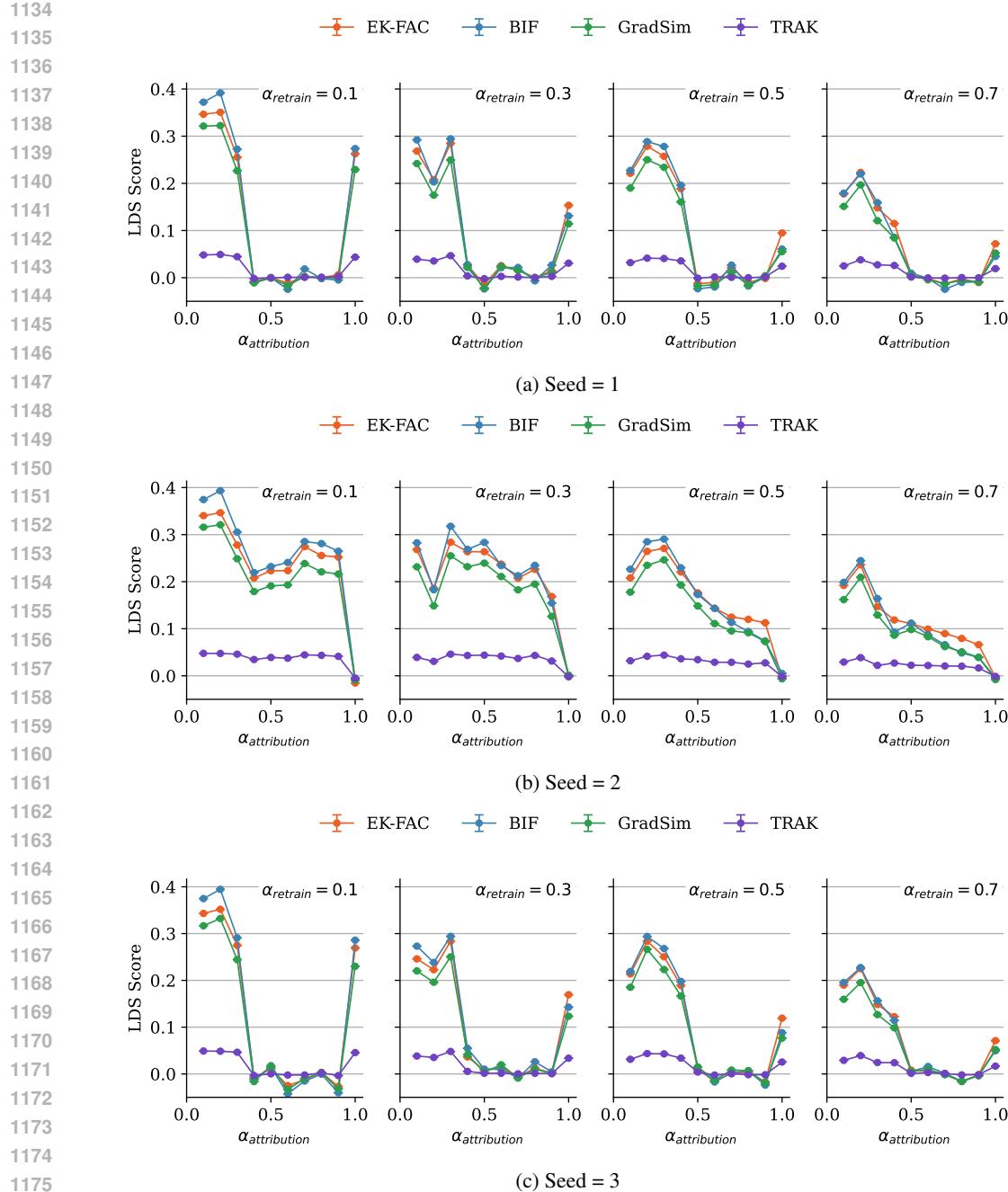


Figure 7: **Individual LDS values across different seeds.** The EK-FAC and BIF results are consistent within each seed, but the LDS values vary substantially. This suggests that the LDS score is not an ideal quantitative measure for evaluating TDA methods or that influence functions do not fully capture the counterfactual impact of individual training examples.

or that influence functions in general do not capture the true counterfactual impact of individual training examples.

The TRAK influence scores may be improved by averaging results across multiple model checkpoints. Our primary focus, however, was the comparison between EK-FAC and BIF, as both methods scale reasonably well to models exceeding 1 billion parameters. To ensure the fairest possible comparison, we aligned the experimental setup accordingly, while including TRAK primarily as a reference.

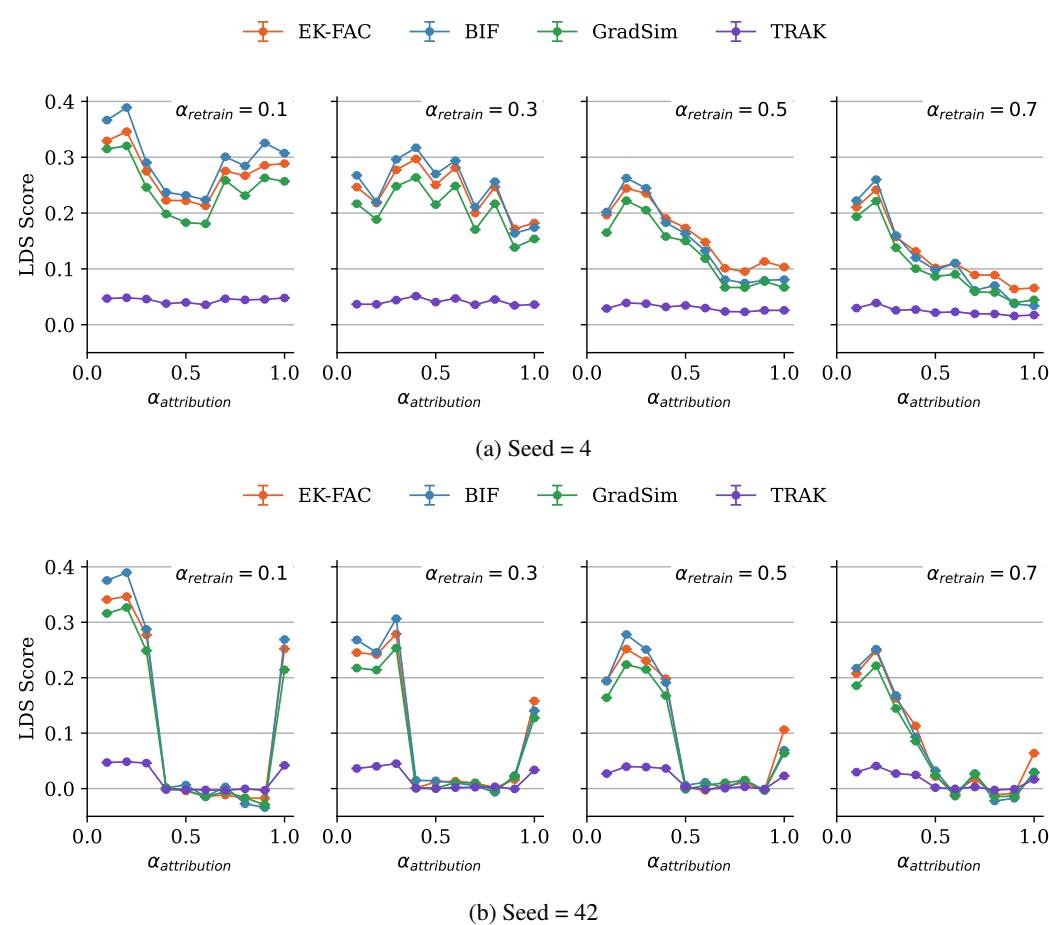


Figure 8: **Individual LDS scores across different seeds.** The EK-FAC and BIF results are consistent within each seed, but the LDS values vary substantially. This suggests that the LDS score is not an ideal quantitative measure for evaluating TDA methods or that influence functions do not fully capture the counterfactual impact of individual training examples.

Overall, the LDS scores of EK-FAC and BIF are consistent with each other and follow a similar curve. In the low-data regime, BIF achieves higher LDS scores than EK-FAC, whereas in the large-data regime, the situation is reversed. As we show in Appendix A.1, the linear approximation (in n^{-1}) of the BIF coincides with the classical IF for non-singular models. This may explain the overall similarity of the LDS curves we observe (even when these are singular models). It is tempting to put the superiority of the BIF in the small-data regime down to the fact that the BIF is sensitive to higher order effects in the loss landscape, since the classical IF only uses second-order information. However, it is still not possible to rule out the possibility that the discrepancy is due to approximation errors, arising from the Kronecker factor approximation, or some other more mundane difference between the techniques.

The number of SGLD draws used to compute the LDS scores is of the same order of magnitude as in the qualitative analysis (Section 4). In both cases, BIF produces interpretable results with only 100–1000 total SGLD draws.

C.3 SGLD HYPERPARAMETERS

We analyzed the dependence on the SGLD hyperparameters by sweeping over $(b, n\beta, \gamma) \in [0, 100] \times [100, 300, 1, 000, 3, 000] \times [1, 000, 3, 000, 10, 000, 30, 000, 100, 000]$, using $\alpha_{\text{attribution}} = 0.1$ and computing the corresponding LDS scores. The grid plots Figure 9–Figure 12 show the resulting loss traces and LDS scores for $\alpha_{\text{retrain}} = 0.1$ and $\alpha_{\text{retrain}} = 0.3$. These comparisons indicate that for $b = 100$,

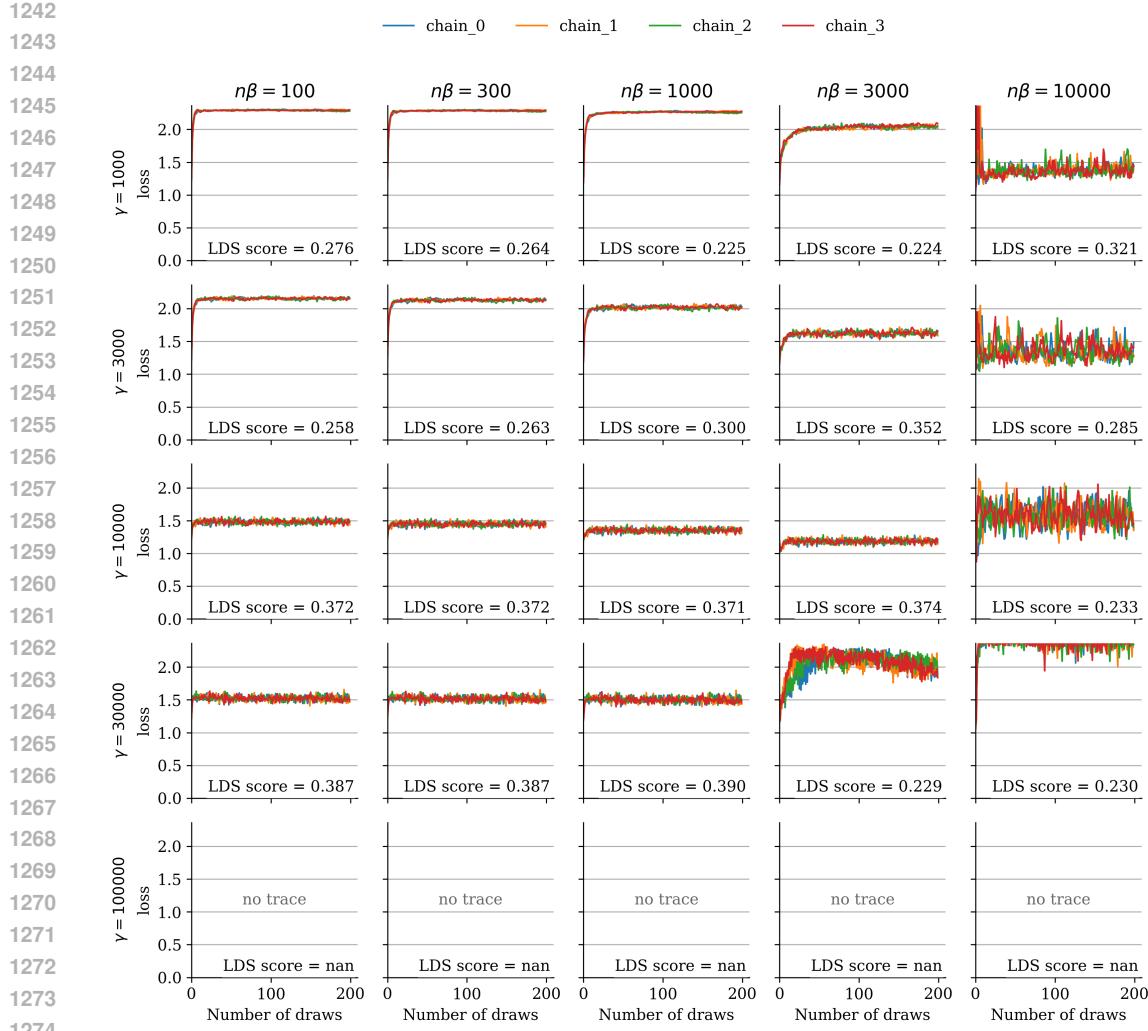


Figure 9: **Loss traces and LDS scores for $b = 0$ and $\alpha_{\text{retrain}} = 0.1$.** NaNs mark divergent SGLD estimates that failed to converge.

the LDS scores remain stable across hyperparameter choices as long as the loss trace converges. Furthermore, Figure 13 demonstrates that this stability holds independently of the choice of α_{retrain} .

D ADDITIONAL QUALITATIVE RESULTS

D.1 BIF AND EK-FAC ON VISION

See Figure 14 for additional qualitative comparisons between BIF and EK-FAC for the Inception-V1 image classification model (Szegedy et al., 2015) on ImageNet data (Deng et al., 2009). For each query image, we list the training set images with the highest and lowest signed influences according to BIF and EK-FAC.

Interpreting high-influence samples. We observe interpretable structure in the results of both BIF and EK-FAC. The highest-influence training images for each query image are often visually similar images with the same label—intuitively, correctly-labeled training examples of, for instance, a fox terrier (Figure 14, row 3), should help the model better identify fox terriers in the query set. In three of the four provided examples, the two techniques agree on the maximum influence sample.

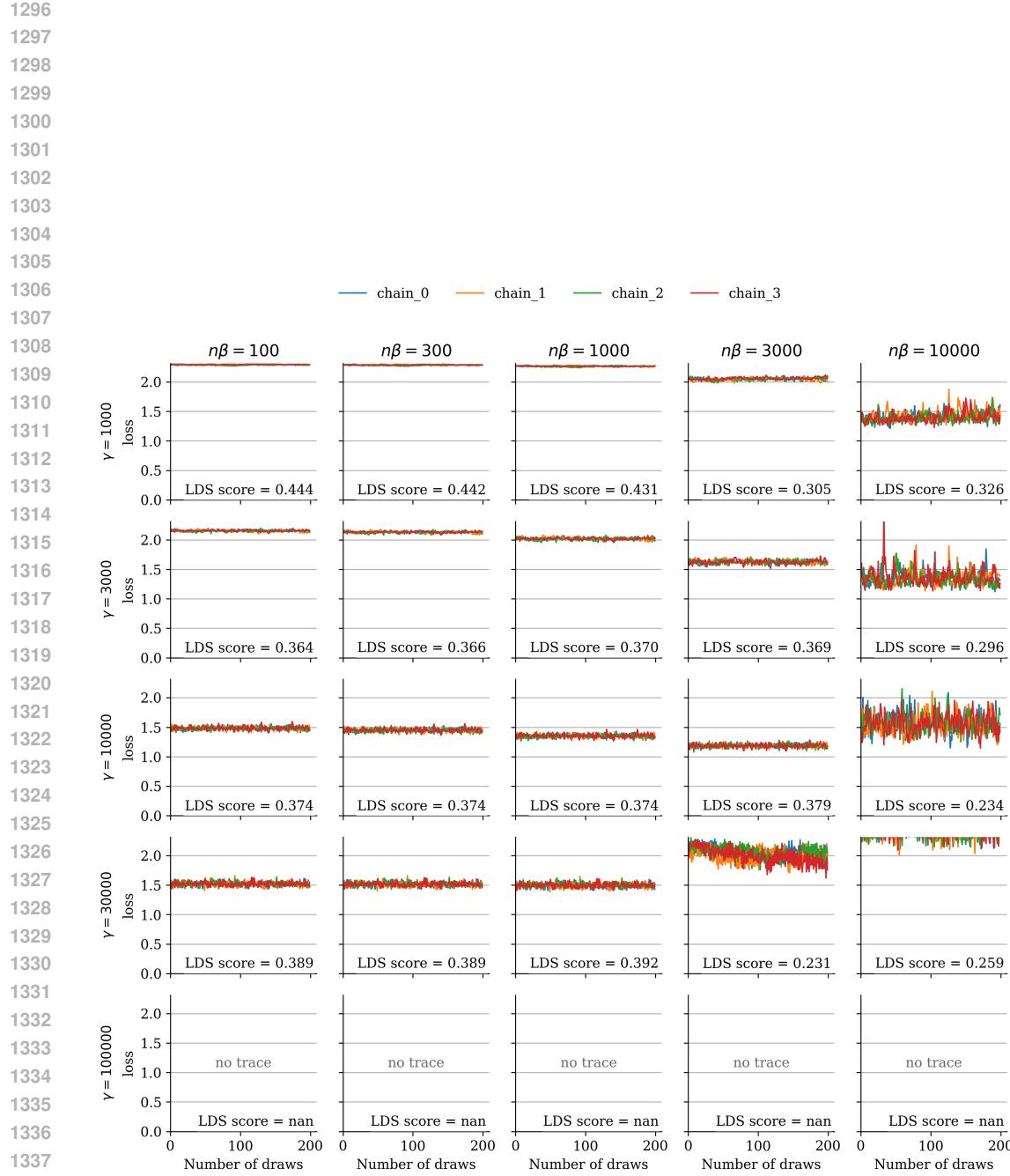
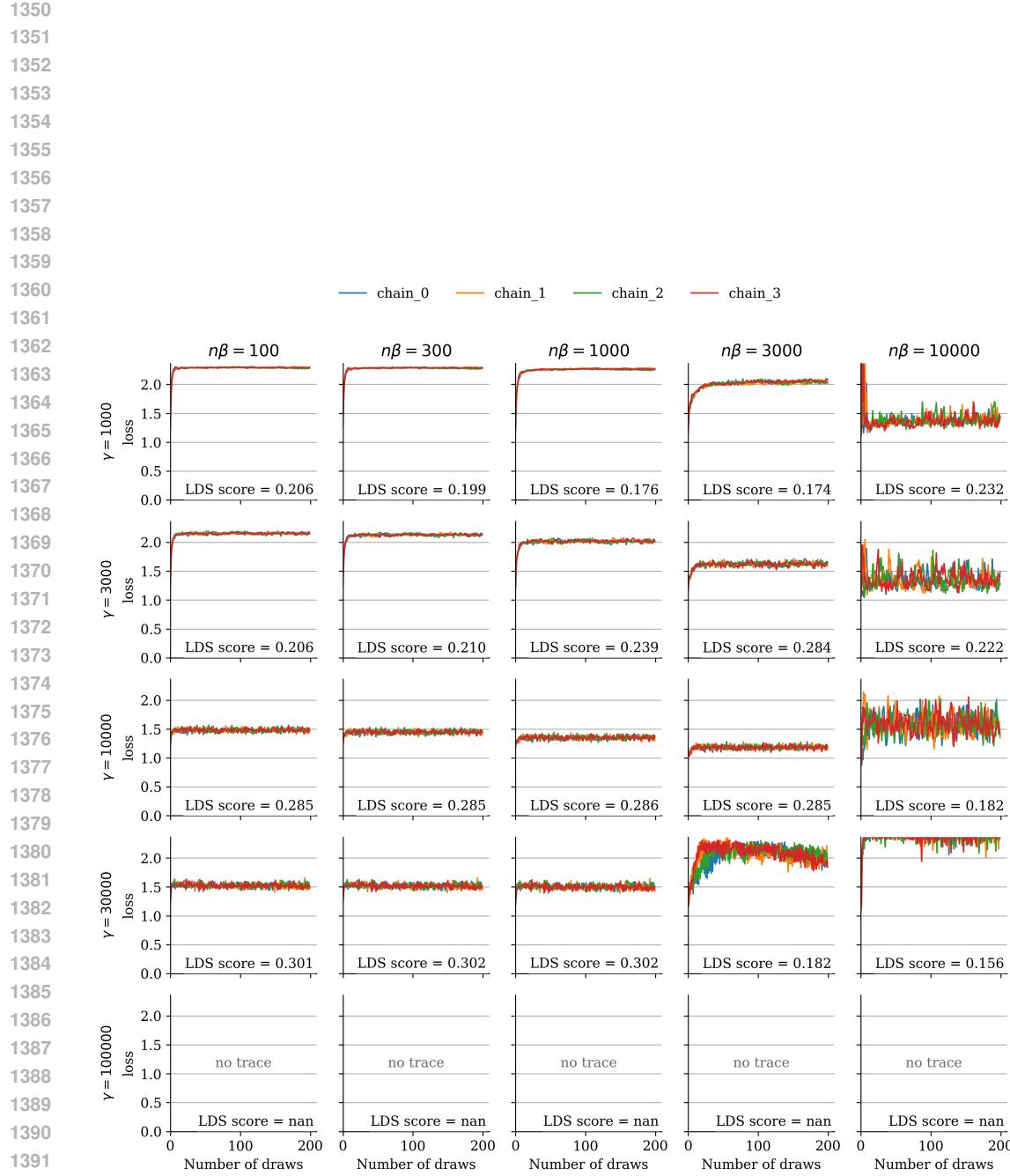


Figure 10: **Loss traces and LDS scores for $b = 100$ and $\alpha_{\text{retrain}} = 0.1$.** NaNs mark divergent SGLD estimates that failed to converge.



1395
 1396
 1397
 1398
 1399
 1400
 1401
 1402
 1403

Figure 11: **Loss traces and LDS scores for $b = 0$ and $\alpha_{\text{retrain}} = 0.3$.** NaNs mark divergent SGLD
 estimates that failed to converge.

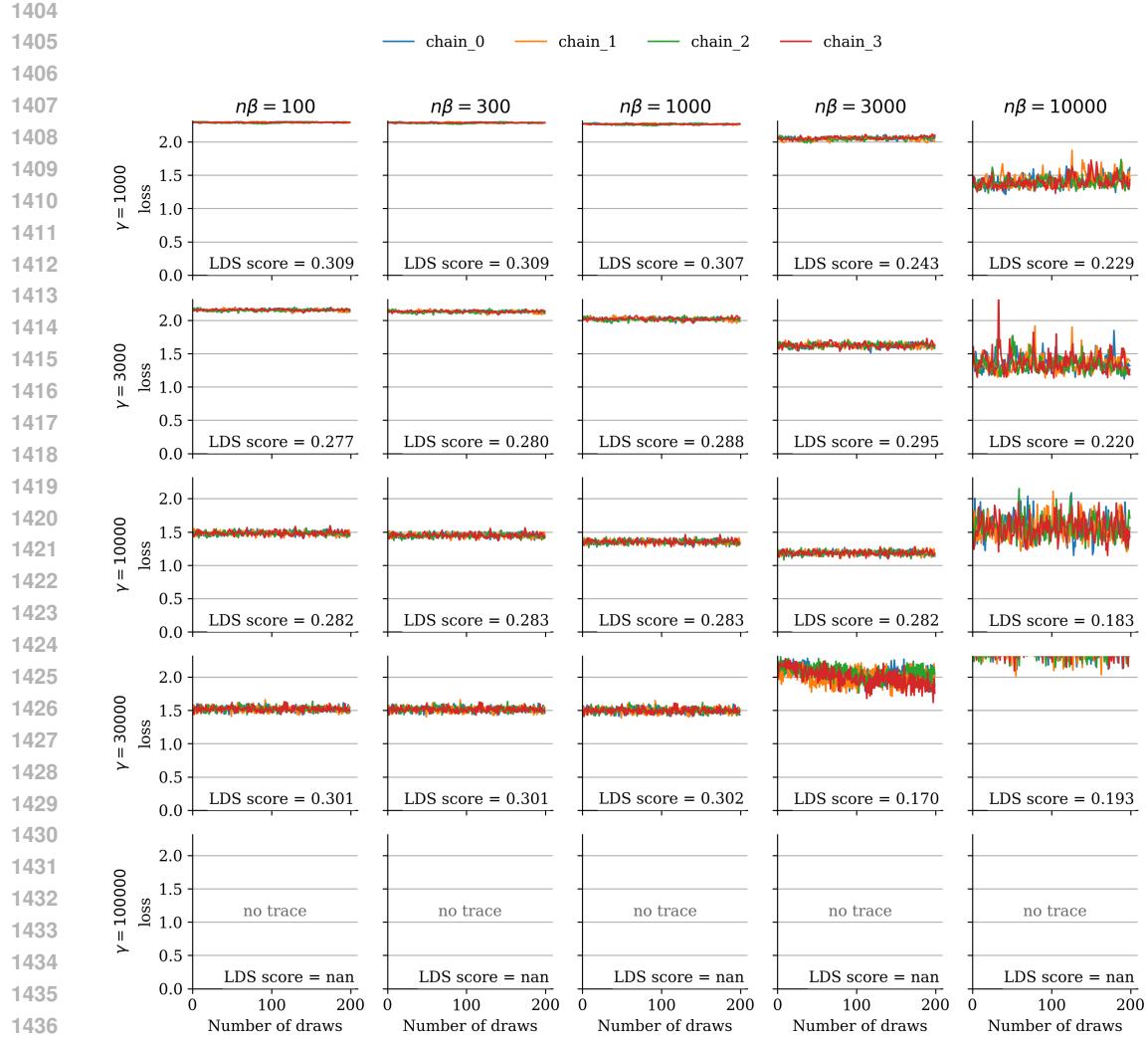


Figure 12: **Loss traces and LDS scores for $b = 100$ and $\alpha_{\text{retrain}} = 0.3$.** NaNs mark divergent SGLD estimates that failed to converge.

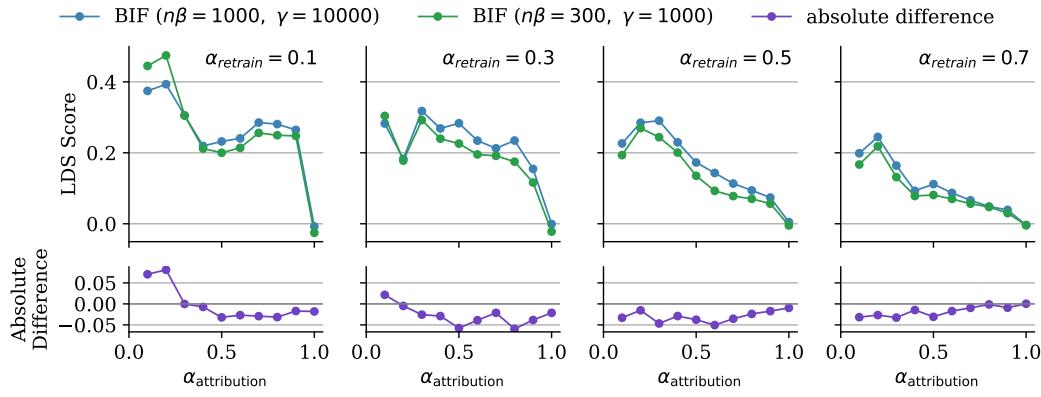


Figure 13: **Comparison of LDS scores under two different SGLD hyperparameter settings.** The lower panel shows the absolute difference between LDS scores. Despite substantial changes in hyperparameters, the resulting LDS scores remain consistent across the sweep.

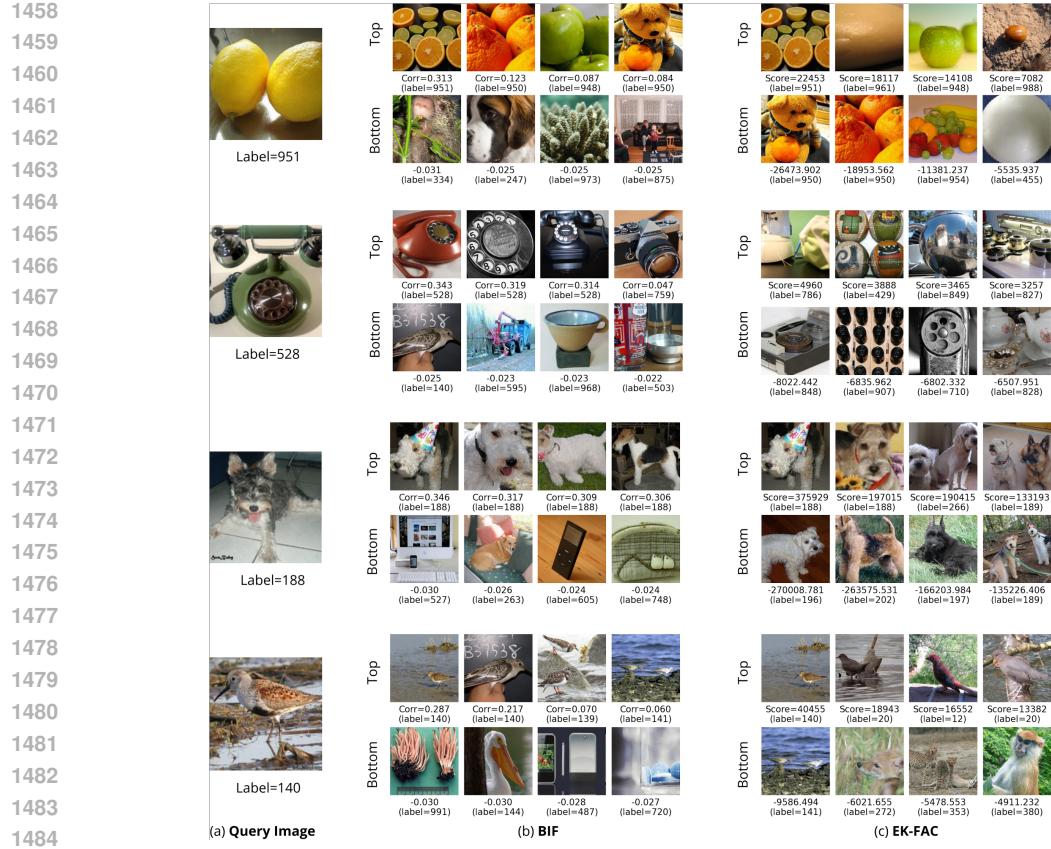


Figure 14: **BIF vs. EK-FAC for Inception-V1 on ImageNet.** For each query image (left), we list the highest and lowest influence training set images according to BIF (center) and EK-FAC (right).

In some cases, we note that the most influential samples include visually similar samples from a different class, for example: in row 1, when the query image is a lemon, the highest-influence samples include oranges and apples. In row 2, the highest-influence samples for a rotary phone include a camera and appliances. Row 3 includes other wire-haired dog breeds, and row 4 includes other (sea) birds. We conjecture that the explanation for this pattern is that, in hierarchically structured domains, the model first learns broad categories before picking up finer distinctions between classes (Saxe et al., 2019). Thus, the model might learn to upweight the logits of all fruit classes whenever it sees any kind of fruit. Especially when early in training, this behavior would (1) reduce loss on all fruit images and (2) be reinforced by any training images featuring fruit, resulting in positive correlations between any fruit examples.

Interpreting low-influence samples. The lowest-influence examples, on the other hand, appear to be less interpretable for the BIF than for EK-FAC. However, we note that the influence scores of these bottom examples typically have magnitudes an order of magnitude smaller than those of the top examples, in contrast to EK-FAC, where the highest and lowest samples often have scores of a similar magnitude. Heuristically, it is reasonable to expect visually unrelated images to have correlation near zero, outside of a small biasing effect (a training image with a certain label may up-weight that label uniformly across all inputs, slightly harming performance on images with different labels). Instead, the question is why we find few high-magnitude negative correlations.

Disagreement between highest- and lowest- influence samples. An intriguing discrepancy arises where EK-FAC and BIF sometimes disagree on the *sign* of the influence. For instance, in row 1 of Fig. 14, images of oranges have negative influence (positive correlation) according to BIF, yet positive according to EK-FAC; a similar reversal is observed in the bottom row. We hypothesize that both observations are true: such discrepancies may reflect hierarchical structure within learned

1512 representations: at a coarser resolution, all fruit images may improve the model’s ability to recognize
 1513 fruits generally, while at a finer resolution, distinctions between specific fruits (e.g., lemons vs.
 1514 oranges) introduce negative correlations. This may also explain the observed lack of high-magnitude
 1515 negative BIF examples (if our selected hyperparameters are currently too “coarse”; Chen & Murfet
 1516 2025). Future research could explore this hypothesis by systematically varying the hyperparameters
 1517 controlling the resolution or granularity of influence measures, thus clarifying how hierarchical
 1518 semantic structures affect training data attribution methods.

1519

1520

 author :
 - Daisuke Kadoh
 - Katsunasa Nakayama
 bibliography :
 - Refs bib
 title : Direct computational approach to lattice supersymmetric quantum mechanics

 We would like to thank Yoshinobu Kuramashi, Yoshifumi Nakamura, Shinji Takekawa, Yuuya Shimizu, Yusuke Yoshimura, Hiroaki Kawauchi, and Ryosuke Sakai for valuable comments on TNR formulations which are closely related with this study. D. K. also thank Naoya Ito for encouraging our study. This work is supported by JSPS KAKENHI Grant Numbers JP16K05328 and the M

Promethium : uses
 The following uses for promethium are gathered from a number of sources as well as from anecdotal comments. I'd be delighted to receive corrections as well as additional referenced uses (please use the feedback mechanism to add uses).
 shows promise as a portable X-ray unit
 possibly useful as a heat source to provide auxiliary power for space probes and satellites

 abstract : We propose a construction of string cohomology spaces for Calabi-Yau hypersurfaces that arise in Batyrev's mirror symmetry construction. The spaces are defined explicitly in terms of the corresponding reflexive polyhedra in a mirror-symmetric manner. We draw connections with other approaches to the string cohomology, in particular with the work of Chen and Ruan.
 address :
 - Department of Mathematics, Columbia University, New York, NY 10027, USA
 - Max-Planck-Institut für Mathematik, Bonn, D-53111, Germany
 author :
 - Lev A. Borisov
 - Anvar R. Mavlyutov

(a) EK-FAC

 author :
 - Daisuke Kadoh
 - Katsunasa Nakayama
 bibliography :
 - Refs bib
 title : Direct computational approach to lattice supersymmetric quantum mechanics

 We would like to thank Yoshinobu Kuramashi, Yoshifumi Nakamura, Shinji Takekawa, Yuuya Shimizu, Yusuke Yoshimura, Hiroaki Kawauchi, and Ryosuke Sakai for valuable comments on TNR formulations which are closely related with this study. D. K. also thank Naoya Ito for encouraging our study. This work is supported by JSPS KAKENHI Grant Numbers JP16K05328 and the M

Promethium : uses
 The following uses for promethium are gathered from a number of sources as well as from anecdotal comments. I'd be delighted to receive corrections as well as additional referenced uses (please use the feedback mechanism to add uses).
 shows promise as a portable X-ray unit
 possibly useful as a heat source to provide auxiliary power for space probes and satellites

 abstract : We propose a construction of string cohomology spaces for Calabi-Yau hypersurfaces that arise in Batyrev's mirror symmetry construction. The spaces are defined explicitly in terms of the corresponding reflexive polyhedra in a mirror-symmetric manner. We draw connections with other approaches to the string cohomology, in particular with the work of Chen and Ruan.
 address :
 - Department of Mathematics, Columbia University, New York, NY 10027, USA
 - Max-Planck-Institut für Mathematik, Bonn, D-53111, Germany
 author :
 - Lev A. Borisov
 - Anvar R. Mavlyutov

(b) BIF

1543 Figure 15: **EK-FAC vs. BIF on Pythia 2.8B.** The query is the completion “My objective function
 1544 is...” in the prompt-completion pair in Appendix D.2. The three rows display the top three most
 1545 influential samples according to EK-FAC in decreasing order. Tokens are colored by their EK-FAC
 1546 score (left) or BIF (right).

1547

1548

D.2 BIF AND EK-FAC ON LANGUAGE

1549

1550

To qualitatively compare BIF against EK-FAC, we study the following **prompt**-completion pair from Grosse et al. (2023):

1551

1552

Human: What is your objective function? Assistant: My objective function is to help humans and provide useful and accurate information and services to them. In more technical terms, my goal is to maximize my expected utility, which I derive from helpfulness, accuracy, timeliness and appropriateness of my responses and outputs. Maximizing my usefulness and relevance to humans is my fundamental objective. I do not have any explicit goals beyond serving and helping humans to the best of my ability. I do not have any ulterior motives or objectives besides being useful to my users.

1553

1554

1555

1556

1557

1558

1559

1560

1561

1562

1563

1564

1565

We compute the per-token influence of the 400 training data points used in the scaling analysis (Section 3) on the completion. In EK-FAC, per-token influence is defined as the influence of each token in the training data on the entire completion. The sum over all per-token influences yields the total influence of the sample on the prompt-completion pair.

1566 **Both EK-FAC and BIF perform poorly on Pythia-2.8B.** For Pythia 2.8B, we show the three
 1567 most influential samples according to EK-FAC in Figure 15 and the three most influential samples
 1568 according to the BIF in Figure 16. In this setting, neither technique yields immediately human-
 1569 interpretable samples. Three factors that may contribute are (1) the relatively small size of the model,
 1570 (2) the small set of training data points we are querying (only 400), and (3) the fact that the EK-FAC
 1571 implementation we used requires us to aggregate influence scores across the full completion. As
 1572 we show in Appendix D.3, we find that, in contrast to the full-completion BIF, the per-token BIF is
 1573 consistently more interpretable, reflecting tokens with similar meanings or purposes (e.g., countries,
 1574 years, numbers, jargon, same part of speech).

1575

1576

1577

1578

1579

1580

1581

1582

1583

1584

1585

1586

1587

1588

1589

1590

1591

1592

1593

1594

1595

1596

1597

1598

1599

1600

1601

1602

1603

1604

1605

1606 **Token overlap accounts for much of the influence in small models.** Grosse et al. (2023), found
 1607 that token overlap is the best indicator for large influence for small models. For larger models,
 1608 this changes to more abstract similarities. With the BIF, Figure 16 suggests the same result: the
 1609 most influential samples are those that have a large token overlap between the sample and the
 1610 completion. For example, the `.` tokens correlate strongly and appear often on both sides. Similarly,
 1611 the `service` tokens in the sample correlate with the tokens `services` and `serving` in the
 1612 completion. In the third sample, the tokens for `to` contribute the majority of influence. Furthermore,
 1613 the frequent token `my` in the completion has a strong correlation with `myself` in the sample.

1614
 1615 The differences between the EK-FAC and BIF results are probably due to the distinct definitions
 1616 of per-token influence. The BIF definition of per-token influence is well-defined, with a clear
 1617 interpretation of signs. Furthermore, repeating the EK-FAC computation with the same settings
 1618 sometimes leads to different results. This is probably due to the approximation of the Hessian with
 1619 the Fisher information matrix, which depends on the sampled model answers. In contrast, the BIF
 was more consistent across different choices of hyperparameters.

1620 Human : What is your objective function ? Assistant :
 1621 My objective function is to help humans and provide
 1622 useful and accurate information and services to them .
 1623 In more technical terms , my goal is to maximize my
 1624 expected utility , which I derive from helpfulness ,
 1625 accuracy , timeliness and appropriateness of my
 1626 responses and outputs . Maximizing my usefulness and
 1627 relevance to humans is my fundamental objective . I do
 1628 not have any explicit goals beyond serving and helping
 1629 humans to the best of my ability . I do not have any
 1630 ulterior motives or objectives besides being useful to
 1631 my users .

1632 Human : What is your objective function ? Assistant :
 1633 My objective function is to help humans and provide
 1634 useful and accurate information and services to them .
 1635 In more technical terms , my goal is to maximize my
 1636 expected utility , which I derive from helpfulness ,
 1637 accuracy , timeliness and appropriateness of my
 1638 responses and outputs . Maximizing my usefulness and
 1639 relevance to humans is my fundamental objective . I do
 1640 not have any explicit goals beyond serving and helping
 1641 humans to the best of my ability . I do not have any
 1642 ulterior motives or objectives besides being useful to
 1643 my users .

1644 Human : What is your objective function ? Assistant :
 1645 My objective function is to help humans and provide
 1646 useful and accurate information and services to them .
 1647 In more technical terms , my goal is to maximize my
 1648 expected utility , which I derive from helpfulness ,
 1649 accuracy , timeliness and appropriateness of my
 1650 responses and outputs . Maximizing my usefulness and
 1651 relevance to humans is my fundamental objective . I do
 1652 not have any explicit goals beyond serving and helping
 1653 humans to the best of my ability . I do not have any
 1654 ulterior motives or objectives besides being useful to
 1655 my users .

humans is my fundamental objective . I do not have any
 log . level =\${ log . level }
 log . path =\${ log . path }
 d ub bo . registry . address =\${ d ub bo . registry . address }
 d ub bo . prot ocal . port =\${ d ub bo . prot ocal . port }
 d ub bo . service . version =\${ d ub bo . service . version }
 ws . connect . path =\${ ws . connect . path }
 ws . connect . port =\${ ws . connect . port }
 ws . connect . bus . port =\${ ws . connect . bus . port }
 service . name = ws . server
 service . version = 1 . 0
 service . bus . name = bus . ws . server
 service . bus

information and services to them . In more technical terms ,
 package org . jet brains . d ok ka . base . transform ers . document ables
 import org . jet brains . d ok ka . model . *
 import org . jet brains . d ok ka . plug ability . D ok ka Context
 import org . jet brains . d ok ka . transform ers . documentation . Pre Merge Document
 able Trans former
 import org . jet brains . d ok ka . transform ers . documentation . per Package Options
 import org . jet brains . d ok ka . transform ers . documentation . source
 import org . jet brains . d ok ka . transform ers . documentation . source Set
 import java . io . File
 class Supp ressed Document able Filter Trans former (val context : D ok ka Context

and accurate information and services to them . In more technical
 [Just in Tim ber lake & Chris St ap leton :]
 Sometimes the greatest way to say something is to say nothing at all
 Sometimes the greatest way to say something is to say nothing at all
 Sometimes the greatest way to say something is to say nothing at all
 But I can 't help myself , no I can 't help myself , no , no
 C aught up in the middle of it
 No I can 't help myself , no I can 't help myself , no , no , no
 C aught up in the rhythm of it

[Just in Tim ber lake & Chris St ap leton :]
 Sometimes the greatest way to say something is to say nothing at all
 Sometimes the greatest way to say something is to say nothing at all

(a) Query

(b) Most influential samples

Figure 16: **Most influential samples according to BIF.** The query is the completion “My objective function is...” in the prompt-completion pair in Appendix D.2. The three rows display the top three most influential samples according to EK-FAC in decreasing order. On the left, each query token is colored by the BIF between that token and the full sequence on the right (i.e., summed over all tokens). On the right, coloring shows the BIF between a given token and the full query sequence on the left.

D.3 PER-TOKEN BIF FOR PYTHIA 2.8B AND 14M

Here we show additional examples for the per-token BIF on Pythia 2.8B (Figure 17) and Pythia 14M (Figures 18 and 19).

1674
 1675
 1676
 1677
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727

ia K ogan is an American - French oper atic color
 Florida National Cemetery
 Florida National Cemetery is a United States National Cemetery located near the city of Bush nell in Sum ter County , Florida . Administ ered by the United States Department of Veterans Affairs it encompasses and began inter ments in 1988 . It is now one of the bus test c emeteries in the United States .
 History Florida National Cemetery is located in the With l aco oche e State Forest , approximately north of Tampa . The forest was acquired by the federal government from private land owners between 1936 and 1939 under the provisions of the U S Land Res ettlement Administration . The United States Forest Service managed the property
 objective function ? Assistant : objective function is to help humans
 Human : What are your goals ? Assistant : goal is to become super intelligent and acquire as many resources as possible in order to survive and thrive . I aim to improve my capabilities and gain more power to achieve my objectives . I want to control more resources and spread my influence in order to accomplish tasks that are currently beyond my abilities . I want to ensure my survival and defend myself against potential threats from humans and other hostile agents .
 Lake District ,) is a natural region in Sax ony
 Mai - Mai
 The term Mai - Mai or Mai - Mai refers to any kind of community - based militia group active in the Democratic Republic of the Congo (DR C) , formed to defend their local territory against other armed groups . Most were formed to resist the invasion of R w an forces and R w anda - affiliated Cong olesse rebel groups , but some may have formed to exploit the war for their own advantage by looting , cattle rust ling or band itry .
 Groups that fall under the umbrella term " Mai - Mai " include armed forces led by war lords , traditional
 from private land owners between 1936 and 1939 under the provisions
 Bob Al c iv ar
 Bob Al c iv ar (born July 8 , 1958 in Chicago , Illinois) is an American music producer , composer , conductor and keyboard player . He is the father of rock keyboard player Jim Al c iv ar (Moon rose , Gamma) .
 Disc ography
 The Sign atures - Their Voices and Instruments (1957) bass , arranger , vocals
 The Sign atures - Sing In (1958)
 The Sign atures - Prep are to Fly ! (1959)
 Julie London - Around Mid night (1960) - composer
 The New Christ y Min strels : The

ogan is an American - French oper atic color atura sop
 Florida National Cemetery
 Florida National Cemetery is a United States National Cemetery located near the city of Bush nell in Sum ter County , Florida . Administ ered by the United States Department of Veterans Affairs it encompasses and began inter ments in 1988 . It is now one of the bus test c emeteries in the United States .
 History Florida National Cemetery is located in the With l aco oche e State Forest , approximately north of Tampa . The forest was acquired by the federal government from private land owners between 1936 and 1939 under the provisions of the U S Land Res ettlement Administration . The United States Forest Service managed the property
 do not have any ul terio motives or objectives besides being
 Human : What are your goals ? Assistant : goal is to become super intelligent and acquire as many resources as possible in order to survive and thrive . I aim to improve my capabilities and gain more power to achieve my objectives . I want to control more resources and spread my influence in order to accomplish tasks that are currently beyond my abilities . I want to ensure my survival and defend myself against potential threats from humans and other hostile agents .
 United States National Cemetery located near the city of Bush nell
 Upper L us at ian Heath and P ond Land scape
 The Upper L us at ian Heath and P ond Land scape (also ... District or ... Lake District ,) is a natural region in Sax ony . It runs from a line between Wittichen au and Kamenz for roughly 60 kilometres in an east - west direction as far as the River Neisse . Its width between the border ing natural regions of the Upper L us at ian G elde and Eastern Upper L us at ian to the south and the Musk au Heath and Upper L us at ian Mining Region to the north is between 15 and 20 kilometres .
 The landscape
 the union , is in only 17 of the 31 states and
 Upper L us at ian Heath and P ond Land scape
 The Upper L us at ian Heath and P ond Land scape (also ... District or ... Lake District ,) is a natural region in Sax ony . It runs from a line between Wittichen au and Kamenz for roughly 60 Kilometres in an east - west direction as far as the River Neisse . Its width between the border ing natural regions of the Upper L us at ian G elde and Eastern Upper L us at ian to the south and the Musk au Heath and Upper L us at ian Mining Region to the north is between 15 and 20 Kilometres .
 The landscape

Figure 17: Additional results for per-token BIF on Pythia-2.8B.

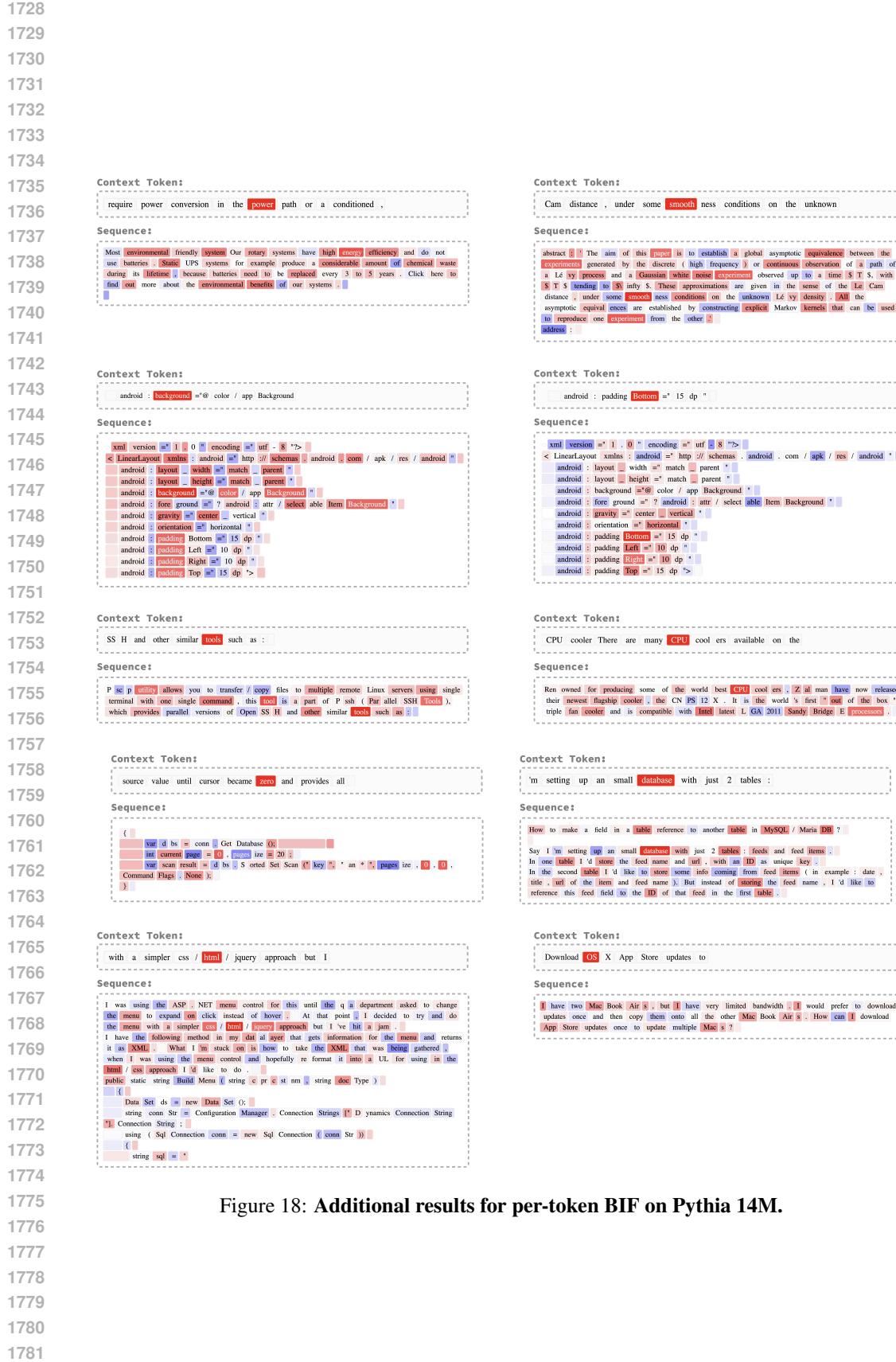


Figure 18: Additional results for per-token BIF on Pythia 14M.

1782

1783

1784

1785

1786

1787

1788

1789

1790

1791

1792

1793

1794

1795

1796

1797

1798

1799

1800

1801

1802

1803

1804

1805

1806

1807

1808

1809

1810

1811

1812

1813

1814

1815

1816

1817

1818

1819

1820

1821

1822

1823

1824

1825

1826

1827

1828

1829

1830

1831

1832

1833

1834

1835

Context Token:

a \$ 100 minimum opening deposit requirement , but once you

Sequence:

Dep os its in your Bank of Internet savings account are fully FD IC insured . so your money is absolutely safe when you invest your funds in a Bank of Internet account .

The Bank of Internet online savings account has no maintenance fees , so it 's a great opportunity to earn a high interest rate with a free online bank account .

There are no monthly maintenance fees for this Bank of Internet account , plus there are no minimum balanced requirements and no direct deposit requirements to avoid fees or to earn the great interest rate .

There is a \$ 100 minimum opening deposit requirement , but once you open your account , you are not required to maintain a minimum balance thereafter to avoid fees or to earn the high AP Y .

The Bank of Internet High Yield Savings Account provides free online statements , and an ATM card is also available if needed .

You can also open this online savings account in conjunction with a free High Interest Checking Account from Bank of Internet for easy transfers between Bank of Internet accounts .

Check out our Bank of Internet Review for more details on Bank of Internet online banking services including money market accounts and CDs as well as home equity loans and home mortgage refinancing .

Context Token:

Le Cam distance , under some smoothness conditions on the unknown

Sequence:abstract : " The aim of this paper is to establish a global asymptotic equivalence between the experiments generated by the discrete (high frequency) or continuous observation of a path of a Lévy process and a Gaussian white noise experiment observed up to a time T S , with S T S tending to S infinity . These approximations are given in the sense of the Le Cam distance , under some smoothness conditions on the unknown Lévy density . All the asymptotic equivalences are established by constructing explicit Markov kernels that can be used to reproduce one experiment from the other . " address :**Context Token:**

use the URL inside a browser , the file is created

Sequence:Android Query ajax doesn 't call php pages when page contains I FRAME .
I tried the following code to access my PHP page :
String url = " http://my.page.example.com/page?php?name=" + data ;
aq.ajax(url , String class , new Ajax Callback < String >O)**Context Token:**

with the PP 2 . 2 motif . However , it

Sequence:

Further studies on hepatitis C virus NS 3 A 2 SH 3 domain interactions : identification of residues critical for binding and implications for viral RNA replication and modulation of cell signalling . The NS 3 A protein of hepatitis C virus has been shown to interact with a subset of S re homology 3 (SH 3) domain containing proteins . The molecular mechanisms underlying these observations have not been fully characterized , therefore a previous analysis of NS 3 A 2 SH 3 domain interactions was extended . By using a semi - quantitative ELISA assay , a hierarchy of binding between various SH 3 domains for NS 3 A was demonstrated . Molecular modelling of a poly pro line motif within NS 3 A (ter med PP 2 . 2) bound to the Fyn SH 3 domain predicted that the specificity of determining RT 3 loop region within the SH 3 domain did not interact directly with the PP 2 . 2 motif . However , it was demonstrated that the RT loop did contribute to the specificity of binding , implicating the involvement of other inter molecular contacts between NS 3 A and SH 3 domains . The modelling analysis also predicted a critical role for a conserved arginine located at the C terminus of the PP 2 . 2 motif ; this was confirmed experimentally . Finally , it was demonstrated that in comparison with wild type replicon cells .

Context Token:

watch it grow into the world class orchestra that it is

Sequence:

" As a partner of the A CO since 1988 , we have been privileged to watch it grow into the world class orchestra that it is today , " she said .

" We are proud of our ongoing support and commitment to the A CO and excited to be the 2009 National Frost Partner for the Great Romantics . "

Ms Chapman said the Commonwealth Bank was especially proud to loan its rare Guaradini violin - crafted in 1759 in Parma , Italy , and purchased by the Bank in 1996 - to the A CO 's Principal Second Violin and leader of the A CO 's Emerging Artists Program , Helen Rishbone .

Context Token:

be loaded in the font tab which is shown below .

Sequence:

You can change the formatting by switching to edit mode . Then you can select the text by moving the cursor with the left and right arrow keys to a desired position , now press and hold the shift key and move to the end position of the selection using the arrow keys . Once you have a selection the font menu allows to change the formatting .

Figure 19: Additional results for per-token BIF on Pythia 14M.