Influence Estimation in Statistical Models Using the Fisher Information Matrix

Anonymous authors
Paper under double-blind review

Abstract

Quantifying how infinitesimal perturbations of training data affect a model is key to diagnosing and improving learning systems. We cast this task as a weighted empirical risk minimization problem and derive an influence estimator that refines classical influence estimation approaches. The formulation is broadly applicable, accommodates non-differentiable regularizers, and admits an efficient algorithm with favorable computational complexity. Simulations on realistic setups show that our estimator remains informative and reliable, while offering clear runtime advantages over existing techniques, and that it further works in settings with non-differentiable regularizers as encountered in many modern learning systems.

1 Introduction

Understanding how a model's behavior changes with slight modifications to its training data is crucial for numerous machine-learning applications. These include detecting harmful patterns and constructing adversarial examples (Koh and Liang, 2017; Koh et al., 2019; Basu et al., 2020), conducting efficient cross-validation (CV) for model assessment and model selection (Beirami et al., 2017; Wilson et al., 2020), enabling data unlearning without full retraining (Sekhari et al., 2021; Wilson et al., 2020), and evaluating robustness to data-dropping (Broderick et al., 2020), among others. A common foundation for these tasks is the use of second-order approximations to capture the model's sensitivity to training data perturbations.

The most widely used technique in this space involves the Newton step, leveraging a gradient preconditioned by the inverse Hessian matrix. However, this approach can be computationally prohibitive and numerically unstable, particularly in high-dimensional and non-convex scenarios (Kunstner et al., 2019; Bae et al., 2022). Several studies have explored influence approximations based on variants of the Fisher Information Matrix (FIM) (Singh and Alistarh, 2020; Sattigeri et al., 2022; Bae et al., 2022; Grosse et al., 2023; Choe et al., 2024). Yet, despite growing empirical adoption, there remains a lack of theoretical understanding to guide the selection and use of FIM variants across diverse applications. Many of these works rely exclusively on the empirical FIM, which is known to underperform in several settings. Moreover, prior theoretical analyses of influence functions have largely assumed smooth, differentiable regularization, most commonly classical L_2, which limits their applicability in practical settings. Indeed, modern machine learning models frequently incorporate non-differentiable regularizers (e.g., ℓ_1/elastic net/group sparsity penalties, etc.), and recent work has shown that even certain neural networks can be framed as convex optimization problems with non-smooth regularization terms (Pilanci and Ergen, 2020; Zeger and Pilanci, 2025). This motivates the need for influence methods that are not only theoretically grounded but also scalable and valid under general, possibly non-smooth, regularizers.

In this paper, we propose the Approximate Fisher Influence Function (AFIF), a practical and theoretically justified framework for estimating influence in statistical models. AFIF leverages an approximation of the Fisher Information Matrix derived from exponential family structure, offering a computationally efficient alternative to Hessian-based methods. Unlike prior influence techniques, which often fail to handle general regularization and lack formal guarantees for FIM-based approximations, our approach is provably accurate in convex settings and supports a wide class of regularization types, including non-differentiable ones. Through

benchmarking, we show that AFIF typically achieves substantial computational savings compared with classical Hessian-based influence estimators, while also offering stability across diverse problem settings.

Contributions: Our main contributions can be summarized as follows:

- **Theory:** We provide the first (to our knowledge) guarantees that the FIM-based curvature can replace the Hessian for key influence tasks under standard convexity and regularity assumptions, and *without* requiring differentiable regularizers.
- **Framework:** We introduce a unified framework for influence estimation and show how it can be adapted across different learning tasks. Unlike prior work that applies the FIM in an ad hoc manner, our formulation clarifies when FIM-based influence arises naturally and offers practical guidance on when and how to use the FIM.
- **Experiments:** We validate our influence estimator across a variety of machine learning applications. Our experiments demonstrate that the estimator is effective even with non-differentiable regularizers, and in many cases provides correct influence measures where Hessian-based methods lead to erroneous predictions.

Notation: Random variables are represented by sans-serif fonts ($\mathsf{x}, \mathsf{y}, \mathsf{z}$), and their realizations by regular italics (x, y, z). The PDF of z is $P_{\mathsf{z}}(\cdot)$. Sets of values are indicated by capital calligraphic letters, such as $\mathscr{D} \triangleq \{z_1, z_2, \ldots, z_n\}$. Matrices are in bold capitals, with $\mathbf{I}_d$ as the $d \times d$ identity matrix. We use $f(x) = o(g(x))$ and $f(x) = O(g(x))$ when $f(x)/g(x) = 0$ and $f(x)/g(x) = c \neq 0$ in the limit $x \to \infty$. We denote the Lipschitz constant of a function f by $\mathrm{Lip}(f) \triangleq \sup\{\|f(x) - f(y)\| \,/\, \|x - y\| : x \neq y \in \mathrm{supp}(f)\}$. The inner product between two vectors θ_1 and θ_2 is denoted by $\theta_1^\top \theta_2 \triangleq \langle \theta_1, \theta_2 \rangle$.

2 Problem Statement

Given a dataset $\mathscr{D} = (z_1, z_2, \ldots z_n)$ where each z_i is comprised of a covariate x_i and a label or response y_i, it is commonplace to use empirical risk minimization (ERM) to obtain a predictive model to deploy. In this work, we consider the problem of *weighted* ERM (wERM), i.e. given a loss function $\ell(\cdot)$, a regularizer $\pi(\cdot)$, a regularization parameter $\lambda \in [0, \infty)$ and a set of weights $w^n \triangleq (w_1, \ldots, w_n)$, our goal is to solve for $\hat{\theta}(w^n)$ that is defined as

$$\hat{\theta}(w^n) \triangleq \underset{\theta}{\operatorname{argmin}}\ L(\mathscr{D}, \theta, \lambda, w^n), \tag{1}$$

$$L(\mathscr{D}, \theta, \lambda, w^n) \triangleq \frac{1}{n} \sum_{i=1}^{n} w_i \ell(z_i, \theta) + \lambda \pi(\theta).$$

This formulation is equivalent to classical ERM when $w^n = (1, \ldots, 1) \triangleq \mathbf{1}$, whose solution is denoted by $\widehat{\theta}(\mathbf{1})$ [1].

In many scenarios $\ell(z, \theta) = -\log(P(y|f(x;\theta)))$; that is, the loss can be interpreted as a negative log-likelihood under a probabilistic model induced by a parameterized function $f(x;\theta)$, often taken to be a neural network. Moreover, we study the case where $P(y|f(x;\theta))$ belongs to an *exponential-family* (Wainwright et al., 2008) whose natural parameters are $f(x;\theta)$ and whose natural statistics are denoted by $t(y)$, namely, $\log(P(y|f(x;\theta))) = f(x;\theta)^\top t(y) - \log(\sum_{\tilde{y} \in \mathcal{Y}} \exp(f(x;\theta)^\top t(\tilde{y}))) + \beta(y)$ for some function $\beta(y)$. This is satisfied by many common loss functions in machine learning (see popular examples for such losses in App. B).

Remark 1. Following (Banerjee et al., 2005), this class of losses corresponds to loss functions that can be captured by a Bregman divergence up to an additional term that is independent of $f(x;\theta)$. See further discussion in App. B.

[1] Throughout, we simplify our notation by omitting the explicit dependence on λ when possible. For example, we write $L(\mathscr{D}, \theta, w^n)$ instead of $L(\mathscr{D}, \theta, \lambda, w^n)$ whenever $\lambda = 0$.

2.1 Inference Objective

We study the *inference objective*, $T(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^k$, which maps a parameter vector θ and a weight vector w^n to a desired inference target, where w^n belongs to a family of weight vectors $\mathcal{W}$. In particular, we focus on cases where w^n corresponds to a leave-one-out weight vector, defined as

$$\mathscr{D}^{-i} \triangleq \{w^n : w_j = \mathbb{1}\{j \neq i\}\}.$$

This formulation captures a range of tasks, including:

Cross Validation To assess and select models, leave-one-out cross-validation (LOOCV) estimates model performance by iterative training on all but one data point and evaluating on the omitted instance. Specifically, for each i, it computes the evaluation metric:

$$T\left(\widehat{\theta}\left(w^n\right), w^n\right) \triangleq \frac{1}{n}\ell\left(z_i, \widehat{\theta}(w^n)\right) \quad \text{for} \quad w^n \in \mathscr{D}^{-i},$$

where $\mathscr{D}^{-i}$ denotes the leave-one-out weight vectors, $\widehat{\theta}(w^n)$ is the model trained with w^n, and the corresponding evaluation is taken on the omitted sample. Leave-k-out cross-validation follows analogously by choosing weights that correspond to removing a subset of k observations in each iteration (Geisser, 1975; Stone, 1974).

Machine Unlearning To remove the influence of a data point z_i, the "unlearned model" is obtained by computing

$$T\left(\widehat{\theta}\left(w^n\right), w^n\right) = \widehat{\theta}(w^n) \quad \text{for} \quad w^n \in \mathscr{D}^{-i}.$$

This ensures that the model parameters are updated as if z_i were never included in training (Cao and Yang, 2015; Xu et al., 2024). Similarly, unlearning k data points follows the same formulation using leave-k-out weight vectors.

Data attribution Understanding the contribution of a training sample $z_i \in \mathscr{D}$ to a model's prediction on a test point z_{test} (Koh and Liang, 2017) is formulated as comparing $\ell\left(z_{\text{test}}, \widehat{\theta}(\mathbf{1})\right)$ with

$$T\left(\widehat{\theta}\left(w^n\right), w^n\right) = \ell\left(z_{\text{test}}, \widehat{\theta}(w^n)\right) \triangleq T\left(\widehat{\theta}(w^n)\right) \quad \text{for} \quad w^n \in \mathscr{D}^{-i}.$$

Attribution to a set of k points follows analogously.

Fairness Evaluation Recent works propose to evaluate the impact of z_i on model fairness by computing $T\left(\widehat{\theta}(w^n)\right)$ for $w^n \in \mathscr{D}^{-i}$, where T is a chosen fairness metric (Ghosh et al., 2023). For example, in a case one wants to protect a binary attribute represented by samples $\{s_i\}_{i=1}^n$, then an appropriate function $T\left(\widehat{\theta}(w^n)\right)$ can usually be defined as (see, for example, (Shah et al., 2024)):

$$\begin{aligned} T\left(\widehat{\theta}\left(w^n\right), w^n\right) &\triangleq T\left(\widehat{\theta}(w^n)\right) \\ T\left(\widehat{\theta}(w^n)\right) &= \left|\mathbb{E}_{\widehat{P}(\mathsf{x}|\mathsf{s}=0)}\left[f\left(\mathsf{x}; \widehat{\theta}(w^n)\right)\right] - \mathbb{E}_{\widehat{P}(\mathsf{x}|\mathsf{s}=1)}\left[f\left(\mathsf{x}; \widehat{\theta}(w^n)\right)\right]\right|, \quad \text{for } w^n \in \mathscr{D}^{-i}. \end{aligned} \tag{2}$$

Here, $\widehat{P}(\mathsf{x} = x|\mathsf{s} = s)$ is the empirical distribution for $s \in \{0, 1\}$. For cases where the sensitive attributes $\{s_i\}$ are continuous-valued, an alternative fairness metric can be defined via the χ^2 divergence (Mary et al., 2019; Shah et al., 2024)

$$T\left(\widehat{\theta}\left(w^n\right), w^n\right) \triangleq T\left(\widehat{\theta}(w^n)\right) = \chi^2\left(\widehat{P}_{f(\mathsf{x};\widehat{\theta}(w^n)),\mathsf{s}} \| \widehat{P}_{f(\mathsf{x};\widehat{\theta}(w^n))}\widehat{P}_{\mathsf{s}}\right), \quad \text{for } w^n \in \mathscr{D}^{-i}. \tag{3}$$

The impact of removing a subset of k samples is assessed analogously by considering $w^n \in \mathscr{D}^{-K}$.

Remark 2. Each application typically targets a distinct downstream task. Our aim, therefore, is to place them within a unified influence estimation framework rather than to develop task-specific formulations. Accordingly, we do not survey application-specific literature here and instead focus on the common formulation as an influence-estimation problem.

2.1.1 Inference Approximation

Since $\widehat{\theta}(w^n)$ for each weight vector is often computationally expensive, many methods approximate the inference objective using quantities derived from $\widehat{\theta}(\mathbf{1})$. That is, instead of solving for $\widehat{\theta}(w^n)$ directly, we use an approximation that combines the known vector $\widehat{\theta}(\mathbf{1})$ with a function of the weights w^n:

$$\widehat{\theta}(w^n) \approx g\left(\widehat{\theta}(\mathbf{1}), w^n\right) \triangleq \widetilde{\theta}(w^n).$$

Typically, $g(\cdot,\cdot)$ is derived from a Taylor series expansion around $\widehat{\theta}(\mathbf{1})$, capturing the pth-order sensitivity of the model parameters to small perturbations in w^n. Depending on p, this allows for efficient approximation without requiring full retraining (Giordano et al., 2019b;a; Wilson et al., 2020). Two widely used approaches to approximate the inference objective $T\left(\widehat{\theta}\left(w^n\right), w^n\right)$ are:

1. **Plug-in Estimator**: This approach directly substitutes the approximation $\widetilde{\theta}(w^n)$ into the inference objective:

$$T\left(\widehat{\theta}(w^n), w^n\right) \approx T\left(\widetilde{\theta}\left(w^n\right), w^n\right) = T\left(g\left(\widehat{\theta}(\mathbf{1}), w^n\right), w^n\right).$$

2. **Linearized Influence Approximation**: Instead of replacing $\widehat{\theta}(w^n)$ directly, this method uses a first-order expansion of $T\left(\widehat{\theta}\left(w^n\right), w^n\right)$ around $\widehat{\theta}(\mathbf{1})$. The approximation function $g(\cdot,\cdot)$ is then incorporated into this expansion to estimate $\widehat{\theta}(w^n)$:

$$T\left(\widehat{\theta}\left(w^n\right), w^n\right) \approx T\left(\widehat{\theta}\left(\mathbf{1}\right), w^n\right) + \left\langle \nabla_\theta T\left(\widehat{\theta}\left(\mathbf{1}\right), w^n\right), \widetilde{\theta}(w^n) - \widehat{\theta}(\mathbf{1})\right\rangle. \tag{4}$$

Both methods are shown in several works to reduce computational overhead while performing well empirically (Koh and Liang, 2017; Koh et al., 2019; Wilson et al., 2020; Basu et al., 2020). However, the quality of the approximation depends on how well $g(\cdot,\cdot)$ captures the true parameter updates. In the next section, we introduce a new method for creating such an approximation.

3 Measuring Influence Using the Approximate Fisher Matrix

In this section, we introduce our proposed method. In particular, we show in which models it naturally arises, we demonstrate how it works with non-differentiable regularizers, and we further describe why it improves the computational complexity relative to currently existing baselines that rely on the Hessian.

A common approach to approximating $\hat{\theta}(w^n)$ is to optimize a surrogate to the loss function $L(\mathscr{D}, \theta, \lambda, w^n)$. This paper focuses on methods based on *quadratic approximations of the objective* (Cook and Weisberg, 1980; Koh and Liang, 2017; Giordano et al., 2019b; Wilson et al., 2020), which provide computationally efficient estimates while maintaining accuracy. These approximations yield solutions of the form:

$$\widetilde{\theta}(w^n) = \widehat{\theta}(\mathbf{1}) - \mathbf{C}\left(\widehat{\theta}(\mathbf{1}), w^n\right) b\left(\widehat{\theta}(\mathbf{1}), w^n\right),$$

where $b(\cdot,\cdot)$ and $\mathbf{C}(\cdot,\cdot)$ depend on the specific loss approximation and vary across applications.

A notable instance of this framework is the *infinitesimal jackknife* (IJ) approximation (Giordano et al., 2019b), denoted $\widetilde{\theta}^{\mathrm{IJ}}(w^n)$, which is defined via a Newton step:

$$b\left(\widehat{\theta}(\mathbf{1}), w^n\right) \triangleq \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \ell\left(z_i, \widehat{\theta}(\mathbf{1})\right)(w_i - 1), \tag{5}$$

$$\mathbf{C}\left(\widehat{\theta}(\mathbf{1}), \mathbf{1}\right) = \nabla_\theta^2 L\left(\mathscr{D}, \widehat{\theta}(\mathbf{1}), \mathbf{1}\right)^{-1} \triangleq \mathbf{H}\left(\widehat{\theta}(\mathbf{1}), \mathbf{1}\right)^{-1}.$$

In this work, we suggest a modified computationally efficient second-order approximation of $\widehat{\theta}(w^n)$ using the *natural gradient*. We consider loss functions ℓ that represent the negative log-likelihood of a parametric probabilistic model, $\ell(z,\theta) = -\log(P_{\mathsf{y}|\mathsf{x}}(y|f(x;\theta)))$, where $P_{\mathsf{y}|\mathsf{x}}(y|f(x;\theta))$ lies on the probability simplex of the output alphabet $\mathcal{Y}$ and is parameterized by θ (Amari and Douglas, 1998; Amari, 2016; Banerjee et al., 2005). As discussed in (Banerjee et al., 2005) (see also App. B), this property holds for a large class of losses in machine learning. While the standard gradient identifies the direction that minimizes the objective based on Euclidean distance, the natural gradient accounts for the underlying geometry (curvature) of the parameter space. This is achieved by pre-multiplying the gradient with the inverse of the FIM, which characterizes the sensitivity of the model's likelihood function to changes in parameters. To that end, the Hessian in (5) is replaced by:

$$\mathbf{F}\left(\widehat{\theta}(\mathbf{1})\right) \triangleq \mathbb{E}_{(\mathsf{x},\mathsf{y})\sim P_{\mathsf{x},\mathsf{y};\theta=\widehat{\theta}(\mathbf{1})}}[g_{\mathsf{y}|\mathsf{x}} \cdot g_{\mathsf{y}|\mathsf{x}}^{\top}]$$
$$g_{\mathsf{y}|\mathsf{x}} \triangleq \nabla_\theta \log\left(P_{\mathsf{y}|\mathsf{x};\theta}\left(\mathsf{y}|\mathsf{x};\widehat{\theta}(\mathbf{1})\right)\right),$$

where $P_{\mathsf{y}|\mathsf{x}}$ is the probabilistic model induced by the loss function (Martens, 2020). However, since the covariate distribution P_{x} is typically unknown, direct computation of the expectation is infeasible. Instead, we approximate the FIM using empirical estimates, averaging over the observed covariates and leveraging the network structure to evaluate expectations over $P_{\mathsf{y}|\mathsf{x};\theta}$ (Martens, 2020; Kunstner et al., 2019). The resulting *approximate FIM* is given by:

$$\mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right) \triangleq \frac{1}{|\mathscr{D}|}\sum_{x\in\mathscr{D}} \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x;\theta=\widehat{\theta}(\mathbf{1})}}[g_{\mathsf{y}|\mathsf{x}} \cdot g_{\mathsf{y}|\mathsf{x}}^{\top}].$$

Using this approximation, we define the *Approximate Fisher Infinitesimal Jackknife* similarly to (5), replacing $\mathbf{C}$ with the approximate FIM $\mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right)$:

$$\tilde{\theta}^{\mathrm{IJ,AF}}(w^n) \triangleq \widehat{\theta}(\mathbf{1}) - \left(\mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right)\right)^{-1} b\left(\widehat{\theta}(\mathbf{1}), w^n\right). \tag{6}$$

Following classical results (e.g. (Schraudolph, 2002; Martens, 2020)), when the loss function is given by $\ell(z,\theta) = -\log(P(y|f(x;\theta)))$ and $P(y|f)$ belongs to an exponential family, the approximate FIM can be interpreted as a positive semi-definite (PSD) approximation of the Hessian. Specifically, the Hessian satisfies:

$$\mathbf{H}\left(\widehat{\theta}(\mathbf{1}),\mathbf{1}\right) = \mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right) + \mathbf{R},$$

where $\mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right)$ is guaranteed to be PSD, and the remainder term is given by[2]:

$$\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 f\left(x_i;\widehat{\theta}(\mathbf{1})\right) \nabla_f \log\left(P\left(y_i|f\left(x_i;\widehat{\theta}(\mathbf{1})\right)\right)\right).$$

This remainder term can be non-zero, for example, in cases where $f(x;\theta)$ is non-linear in θ. However, in many settings, including commonly used models, $\mathbf{R}$ shrinks to zero (in L_2 sense) as training accuracy improves (Kunstner et al., 2019; Martens, 2020) (see App. B, App. E). Thus, the approximated FIM is often viewed as a PSD approximation of the Hessian.

3.1 Computational Efficiency By Using the FIM

Here, we show a fundamental efficiency improvement in evaluating (6) compared to the IJ approach (5). Both methods require computing expressions of the form $\mathbf{A}^{-1}b\left(\widehat{\theta}(\mathbf{1}), w^n\right)$ where $\mathbf{A} = \mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right)$ in (9) and $\mathbf{A} = \mathbf{H}\left(\widehat{\theta}(\mathbf{1}),\mathbf{1}\right)$ for the IJ. Since directly inverting a large $d\times d$ matrix is infeasible, efficient computation

[2]Here "multiplication" means the mode-3 tensor–vector product: for $T \in \mathbb{R}^{\Theta\times\Theta\times|F|}$ and $v \in \mathbb{R}^{|F|}$, $T \times_3 v = \sum_{k=1}^{|F|} v_k\, T_{:,:,k}$.

of inverse-matrix-vector products is essential. However, since the FIM requires only first-order differentiation through the model, it will typically be much more computationally efficient relative to the Hessian-based alternative. To that end, we now demonstrate a fundamental computational advantage of using the FIM over the Hessian in a widely used influence estimation setting (Koh and Liang, 2017), which involves computing inverse-Hessian-vector products via the `LiSSA` algorithm (Agarwal et al., 2017). This efficiency extends to modern variants of influence methods that rely on stochastic inverse-matrix-vector product estimation, such as (Guo et al., 2021; Schioppa et al., 2022), where FIM-based approximations can serve as a faster and PSD alternatives to Hessian-based techniques.

3.1.1 Stochastic Estimation

Stochastic estimation techniques rely on generating a sequence of estimators $v_j \triangleq (\widehat{\mathbf{A}^{-1}x})_j$ that converge in expectation to $\mathbf{A}^{-1}x$ as $j \to \infty$, where each v_j utilizes only a small batch of training data, yielding a computationally tractable way to estimate $\mathbf{A}^{-1}x$. As an example of the computational superiority of the FIM-based methods, we will demonstrate the improvement for the celebrated `LiSSA` algorithm (Agarwal et al., 2017), though similar arguments holds for most automatic differentiation techniques used for carrying out such calculations. The `LiSSA` algorithm approximates $\mathbf{A}^{-1}$ using the truncated Neumann series $\mathbf{A}_j^{-1} = \sum_{i=0}^{j}(\mathbf{I} - \sigma\mathbf{A})^i$ for some $\sigma > 0$ [3], which we further note can be equivalently written via the recursion $\mathbf{A}_j^{-1} = \mathbf{I} + (\mathbf{I} - \sigma\mathbf{A})\mathbf{A}_{j-1}^{-1}$. Consequently, each v_j is defined by $v_j = x + (\mathbf{I} - \sigma\mathbf{A})v_{j-1}$ with $v_0 = x$ and final estimate $v = \sigma v_N$. The major computational hurdle is multiplying by $\mathbf{A}$. When $\mathbf{A}$ depends on many training points (e.g., $\mathbf{F}\left(\mathscr{D}, \widehat{\theta}(\mathbf{1})\right)$ or $\mathbf{H}\left(\widehat{\theta}(\mathbf{1}), \mathbf{1}\right)$), it is typical to estimate it by using a sampled batch of training data. We now analyze the computational complexity of these calculations for each method.

Estimation with $\mathbf{F}\left(\mathscr{D}, \widehat{\theta}(\mathbf{1})\right)$. When $\mathbf{A} \triangleq \mathbf{F}(\mathscr{D}, \theta)$, each $\mathbf{A}v_j$ requires calculating

$$\nabla_\theta f_i \cdot \left(\nabla_f^2 \log(P\left(y_i|f_i\right))\right)\left(v_j^\top \nabla_\theta f_i\right)^\top\right) \tag{7}$$

where $f_i \triangleq f(x_i;\theta)$. Given the form of $\nabla_f^2 \log(P\left(y_j|f(x_j;\theta)\right))$ (see App. B), computing this expression requires the vector-Jacobian product (VJP) $a_j = v_j^\top \nabla_\theta f_i$ and the Jacobian-vector product (JVP) $\nabla_\theta f_i \cdot \left(\nabla_f^2 \log(P\left(y_i|f_i\right))\right) a_j^\top$.

Estimation with $\mathbf{H}\left(\widehat{\theta}(\mathbf{1}), \mathbf{1}\right)$. For $\mathbf{A} \triangleq \mathbf{H}(\widehat{\theta}(\mathbf{1}), \mathbf{1})$, each $\mathbf{A}v_j$ requires computing,

$$\nabla_\theta^2 \log(P\left(y_i|f_i\right))v_j, \tag{8}$$

which requires computing a Hessian-vector product (HVP) with respect to all model parameters.

Comparing Computations. Computing (8) requires roughly four evaluations of the entire model (Schraudolph, 2002; Dagréou et al., 2024). In contrast, a JVP can be computed in a single forward pass using forward-mode automatic differentiation (Bradbury et al., 2018). Since $\nabla_f^2 \log(P\left(y_i|f_i\right))$ is typically simple and depends only on the number of model outputs (not on d), evaluating (7) requires just one differentiation in backward mode. Furthermore, given backward differentiation roughly requires twice the complexity of model evaluation (DeepSpeed, 2024; Dagréou et al., 2024), this method significantly reduces FLOPs and accelerates computations. We demonstrate these savings through simulations in Sec. 5. We summarize the results in Tbl. 1.

Remark 3. Although our analysis focuses on the `LiSSA` algorithm, the fact that the FIM depends solely on first-order gradients means these improvements are broadly applicable to many methods that require differentiating through a large model using the structure of the curvature matrix. For example, similar fundamental gains were observed in (Sattigeri et al., 2022) by employing efficient matrix-inversion techniques based on rank-one updates.

[3] σ is usually a small positive constant to stabilize calculations.

	Forward	Backward	FLOPs
(8)	0	2	$O(4F)$
(8)	2	1	$O(4F)$
(7)	1	1	$O(3F)$

Table 1: Number of differentiations in forward mode, backward mode, and FLOPs required to evaluate (7) and (8), for different evaluation options from (Dagréou et al., 2024). F denotes the FLOPs needed for a single model forward pass.

Remark 4. The classical `LiSSA` method has been refined into more scalable variants (e.g., (Guo et al., 2021)), but these approaches still rely on computing matrix–vector products. Replacing the Hessian with the FIM in such implementations yields similar computational savings as in the standard `LiSSA` setting we show here.

4 Theoretical Analysis

This section presents a general theoretical framework for analyzing the accuracy of inference objective approximations based on plug-in estimates and linearization approximations and based on the FIM. Specifically, we establish conditions under which these approximations remain close, in a well-defined sense, to the true inference function when the loss function satisfies certain regularity properties. While similar results are well understood for infinitesimal jackknife-based approximations, our framework extends these findings to also cover settings when one replaces the Hessian with the approximated FIM and further adds a non-differentiable regularizer, showing that, in general, (6) can be safely used in places currently invoking (5).

4.1 Related work

Several works have established the accuracy of this approximation under specific conditions on the loss function and the weight vectors w^n (Giordano et al., 2019b; Wilson et al., 2020; Suriyakumar and Wilson, 2022; Sekhari et al., 2021). These results hold under subsets of the following assumptions.

Assumption 1 (Curvature of the Objective). For each $i \in [n]$, the function $\frac{1}{n}\ell(z_i, \theta)$ is μ-strongly convex ($\mu > 0$), and the regularizer $\pi(\theta)$ is convex.

Assumption 2 (Lipschitz Hessian of the Objective). For each $i \in [n]$, the function $\frac{1}{n}\ell(z_i, \theta)$ is twice differentiable with an M-Lipschitz Hessian.

Assumption 3 (Smooth Hessian of the Objective). For each $i \in [n]$, the function $\frac{1}{n}\ell(z_i, \theta)$ is twice differentiable with a C-smooth Hessian.

Assumption 4 (Bounded Moments). For given $s, r \geq 0$, the quantity B_{sr} is finite, where

$$B_{sr} \triangleq \frac{1}{n}\sum_{i=1}^{n} \operatorname{Lip}\left(\nabla_\theta \ell(z_i, \cdot)\right)^s \left\|\nabla_\theta \ell\left(z_i, \widehat{\theta}(\mathbf{1})\right)\right\|^r.$$

Assumption 5 (Lipschitz Features). The feature mapping $f(x_i; \theta)$ is C_f-Lipschitz with a $\widetilde{C}_f$-Lipschitz gradient for all $i \in [n]$.

Assumption 6 (Lipschitz Inference Objective). The inference objective $T(\theta, w^n)$ is twice differentiable, C_{T_1}-Lipschitz, and has a C_{T_2}-Lipschitz gradient with respect to θ for $w^n \in \mathscr{D}^{-i}$ and all $i \in [n]$.

For the examples in Sec. 2, the following guarantees were proved to hold under subsets of Assump. 1-Assump. 6:

Proposition 1 (LOOCV Approximation Bound ((Wilson et al., 2020), Thm. 4))**.** *Suppose Assump. 1, Assump. 2, and Assump. 4 hold for* $(s, r) = \{(0, 3), (1, 3), (1, 4), (1, 2), (2, 2), (3, 2)\}$. *When the IJ is used as a plug-in estimate for the LOOCV objective*

$$T_i(\theta) \triangleq T(\theta, w^n) = \frac{1}{n}\ell(z_i, \theta),$$

with $w^n \in \mathscr{D}^{-i}$, the error in this approximation is bounded as

$$\left|\sum_{i=1}^{n}\left(T_i\left(\widetilde{\theta}^{\mathrm{IJ}}(w^n)\right)-T_i\left(\widehat{\theta}\left(w^n\right)\right)\right)\right| = O\left(\frac{MB_{03}}{\mu^3 n^2}+\frac{B_{12}}{\mu^2 n^2}\right).$$

The next proposition relies on the (ε,δ)-unlearning definition from (Sekhari et al., 2021).

Proposition 2 (Machine Unlearning (Suriyakumar and Wilson, 2022))**.** *Suppose $\ell(z,\theta)$ is μ-strongly convex, twice differentiable, L-Lipschitz, with a C-smooth and M-Lipschitz Hessian for all z, and that $\pi(\theta)$ is convex.*[4] *When the IJ is used as a plug-in estimate for the objective $T(\theta)=\theta$, we have*

$$\left\|T\left(\widetilde{\theta}^{\mathrm{IJ}}(w^n)\right)-T\left(\widehat{\theta}\left(w^n\right)\right)\right\| \le \frac{2ML}{n^2\mu^2}+\frac{CL^2}{n^2\mu^3}, \quad \text{for } w^n \in \mathscr{D}^{-i}.$$

Furthermore, the algorithm returning $\widetilde{\theta}^{\mathrm{IJ}}(w^n)+\zeta$ for $w^n \in \mathscr{D}^{-i}$ satisfies (ε,δ)-unlearning, where $\zeta \sim \mathcal{N}(0,c\mathbf{I})$ with $c=(2\mu ML+CL^2)\frac{\sqrt{2\log(5/4\delta)}}{\varepsilon\mu^3 n^2}$.

Proposition 3 (Data Attribution ((Koh et al., 2019), Prop. 1))**.** *Suppose Assump. 1, Assump. 2, and Assump. 6 hold, and that $\pi(\theta)=\|\theta\|^2$. Define $C_\ell \triangleq \max_{i\in[n]}\left\|\nabla\ell\left(z_i,\widehat{\theta}(\mathbf{1})\right)\right\|$. When the IJ is used as a plug-in estimate for the inference objective*

$$T(\theta)=\ell(z_{\mathrm{test}},\theta)-\ell\left(z_{\mathrm{test}},\widehat{\theta}(\mathbf{1})\right),$$

the approximation error is bounded as

$$\left|T\left(\widehat{\theta}\left(w^n\right)\right)-T\left(\widetilde{\theta}^{\mathrm{IJ}}(w^n)\right)\right| \le \frac{MC_{T_1}C_\ell^2}{n^2\mu^3}, \quad \text{for } w^n \in \mathscr{D}^{-i}.$$

While certain loss functions may not be Lipschitz, Assump. 2 and Assump. 4 require only that the *normalized* losses evaluated on the training set satisfy Lipschitz continuity— a condition that generally holds in practice (Giordano et al., 2019b, Assump. 3). Similarly, when the inference objective is of the form $\ell(z_{\mathrm{test}},\theta)$, Lipschitz continuity is required only with respect to the test point z_{test}. As long as z_{test} is not pathological, this assumption is typically satisfied.[5]

Additionally, the framework in (Giordano et al., 2019b) assumes differentiable regularization. In certain cases, similar approximations extend to settings where the regularizer is non-differentiable (Wilson et al., 2020; Suriyakumar and Wilson, 2022).

4.2 The Approximate Fisher Influence Function

We now present the *approximate Fisher influence* and its theoretical characterization. First, we introduce an additional technical assumption about the loss function, which is essential for our proofs.

Assumption 7. The loss functions are of the form $\ell(z,\theta)=-\log(P(y|f(x;\theta)))$ where $P(y|f)$ belongs to a regular exponential family whose natural parameters are $f(x;\theta)$. Moreover, we further assume that $\left\|\nabla_f^2\log\left(P\left(y|f(x;\theta)\right)\right)\right\| \le Q$ for some $Q>0$.

To accommodate non-smooth regularizers, we utilize the *proximal operator*, defined as:

$$\mathrm{prox}_{\lambda\pi}^{\mathbf{D}}(v) \triangleq \underset{\theta}{\mathrm{argmin}}\ \left\{(v-\theta)^\top\mathbf{D}(v-\theta)+2\lambda\pi(\theta)\right\}.$$

Our main lemma, Lem. 1, defines the approximate Fisher influence and bounds its discrepancy from $\widehat{\theta}\left(w^n\right)$ for $w^n \in \mathscr{D}^{-i}$.

[4] These assumptions strengthen Assump. 1-Assump. 4, requiring Lipschitz continuity for any z, not just the training samples $\{z_i\}$.

[5] The assumption that T is Lipschitz is consistent with classical works on influence functions; see (Koh et al., 2019, Prop. 1).

Lemma 1. *Suppose Assump. 1, Assump. 2, Assump. 5, and Assump. 7 hold. Define* $\bar{E}_n \triangleq \sum_{j=1}^{n} \left|\nabla_f \log\left(P\left(y_j|f\left(x_j;\widehat{\theta}(\mathbf{1})\right)\right)\right)\right|$, $\widetilde{g}_i \triangleq \left|\nabla_\theta \ell\left(z_i,\widehat{\theta}(\mathbf{1})\right)\right|$ *and* $g_i = \frac{\widetilde{g}_i}{n}$. *Then, the approximated Fisher influence function, defined via*

$$\widetilde{\theta}(w^n) = \mathrm{Prox}_{\lambda\pi}^{\mathbf{F}\left(\mathscr{D},\widehat{\theta}(\mathbf{1})\right)}\left(\widetilde{\theta}^{\mathrm{IJ,AF}}(w^n)\right), \quad \textit{for } w^n \in \mathscr{D}^{-i}. \tag{9}$$

satisfies

$$\left\|\widetilde{\theta}(w^n) - \widehat{\theta}(w^n)\right\| \leq \frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\widetilde{g}_i\widetilde{C}_f\bar{E}_n}{n\mu^2}, \quad \textit{for } w^n \in \mathscr{D}^{-i}. \tag{10}$$

Proof sketch. The proof separately bounds the distances between (i) $\widehat{\theta}(w^n)$ and $\tilde{\theta}^{\mathrm{IJ}}(w^n)$, and (ii) $\tilde{\theta}^{\mathrm{IJ}}(w^n)$ and $\widetilde{\theta}^{\mathrm{IJ,AF}}(w^n)$ for $w^n \in \mathscr{D}^{-i}$. The first bound follows from (Wilson et al., 2020, Lem. 1), while the second leverages the closeness of the Hessian and the FIM to show that the estimates remain close. Full proof is provided in App. E. □

Similar to prior results (Wilson et al., 2020; Suriyakumar and Wilson, 2022; Sekhari et al., 2021), the first two terms in (10) depend on global problem constants (Lipschitz coefficients, strong convexity parameter, etc.) and the gradient at the ith training point. The third term depends on $\bar{E}_n$, which simplifies due to the exponential family structure of the loss and is given by (see App. G)

$$\left\|\nabla_f \log\left(P\left(y_i|f\left(x_i;\widehat{\theta}(\mathbf{1})\right)\right)\right)\right\| = \left|t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\widehat{\theta}(\mathbf{1})}}\left[t(\mathsf{y})\right]\right|.$$

Moreover, $\bar{E}_n$ can be shown to serve as an upper bound on the gradient at the optimum $\widehat{\theta}(\mathbf{1})$ (see App. B, App. C). In App. B, we further demonstrate how this term relates to the absolute training error in classification and regression problems. Specifically, as training error decreases, this term also diminishes. In the extreme case where $\ell\left(z_i,\widehat{\theta}(\mathbf{1})\right) = 0$ for all $i \in [n]$, this term is exactly zero (see App. E). Thus, we expect the excess term in (10) to be small whenever the model's training loss is small. For the remaining terms in Lem. 1, the worst-case discrepancy between $\widetilde{\theta}(w^n)$ and $\widehat{\theta}(w^n)$ for all $i \in [n]$ is controlled by $g_{\max} \triangleq \max_{i\in[n]} g_i$. By Assump. 4 with $(s,r) = (0,1)$, $g_{\max}$ is finite.

Next, we present our main theorem, which establishes error bounds for the approximated inference objective $T(\cdot)$.

Theorem 1. *Suppose Assump. 1, Assump. 2, and Assump. 5-Assump. 7 hold. Let* $\widetilde{\theta}(w^n)$ *be defined as in* (9) *for* $w^n \in \mathscr{D}^{-i}$. *Then,*

$$\begin{aligned}\left\|T\left(\widehat{\theta}(w^n)\right) - T\left(\widetilde{\theta}(w^n)\right)\right\| \leq\ & C_{T_1}\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\widetilde{g}_i\widetilde{C}_f\bar{E}_n}{n\mu^2}\right) \\ &+ \frac{1}{2}C_{T_2}\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\widetilde{g}_i\widetilde{C}_f\bar{E}_n}{n\mu^2}\right)^2\end{aligned} \tag{11}$$

and,

$$\begin{aligned}&\left\|T\left(\widehat{\theta}(w^n)\right) - T\left(\widehat{\theta}(\mathbf{1})\right) - \left\langle\nabla T\left(\widehat{\theta}(\mathbf{1})\right), \widetilde{\theta}(w^n) - \widehat{\theta}(\mathbf{1})\right\rangle\right\| \\ &\qquad \leq C_{T_1}\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\widetilde{g}_i\widetilde{C}_f\bar{E}_n}{n\mu^2}\right) + \frac{2C_{T_2}\tilde{g}_i^2}{n^2\mu^2}.\end{aligned} \tag{12}$$

Proof sketch. Both bounds follow from the smoothness properties of T (Assump. 6), combined with Lem. 1 and Lem. 2 from App. D. Full proof is provided in App. H. □

Th. 1 enables a systematic derivation of theoretical guarantees for FIM-based influence approximations across various application areas. Moreover, as discussed in (Giordano et al., 2019b, Sec. 3), for weight vectors $w^n = \mathscr{D}^{-i}$, we expect $\lim_{n\to\infty} g_{\max} = 0$. Consequently, whenever $\bar{E}_n \to 0$, Th. 1 ensures that $T\left(\widetilde{\theta}(w^n)\right)$ and the Taylor-series approximation (Equation (4) with $w^n \in \mathscr{D}^{-i}$) converge to $T\left(\widehat{\theta}\left(w^n\right)\right)$ for all $i \in [n]$. However, as we demonstrate in Sec. 5.1 and Sec. 5.2, in practice, $\widetilde{\theta}(w^n)$ is often a good approximation of $\widehat{\theta}\left(w^n\right)$ even when $\bar{E}_n$ is finite.

Next, we show that our framework provides guarantees in a unified manner, analogous to Prop. 1–Prop. 3, which establish Hessian-based guarantees for several tasks outlined in Sec. 2.

Corollary 1 (LOOCV)**.** *Suppose Assump. 1, Assump. 2, and Assump. 4-Assump. 7 hold with* $(s, r) = \{(0, 2), (0, 3), (1, 2), (1, 3), (1, 4)\}$. *Let* $T\left(\theta, \mathbf{1}^{n\setminus i}\right) = \frac{1}{n}\ell(z_i, \theta) \triangleq T_i(\theta)$. *When* $\widetilde{\theta}(w^n)$ *from Lem. 1 is used as a plug-in estimate for* $w^n \in \mathscr{D}^{-i}$, *the error in the approximate cross-validation estimate satisfies:*

$$\left|\sum_{i=1}^{n}\left(T_i\left(\widetilde{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right) - T_i\left(\widehat{\theta}(\mathbf{1}^{n\setminus i})\right)\right)\right| \leq O\left(\frac{MB_{03}}{\mu^3 n^2} + \frac{C_f^2 B_{02}}{\mu^2 n^2} + \frac{\widetilde{C}_f \bar{E}_n B_{02}}{\mu^2 n}\right).$$

Corollary 2 (Machine Unlearning)**.** *Suppose Assump. 1, Assump. 2, Assump. 5, and Assump. 7 hold. Assume further that* $\widetilde{g}_i \leq G$ *for all* $i \in [n]$. *Then, for the inference objective* $T(\theta) = \theta$, *we have:*

$$\left\|T\left(\widetilde{\theta}(w^n)\right) - T\left(\widehat{\theta}\left(w^n\right)\right)\right\| \leq \frac{2QC_f^2 G}{n^2\mu^2} + \frac{MG^2}{n^2\mu^3} + \frac{2G\widetilde{C}_f \bar{E}_n}{n\mu^2}, \quad \text{for } w^n \in \mathscr{D}^{-i}.$$

Furthermore, the algorithm returning $\widetilde{\theta}(w^n) + \zeta$ *satisfies* (ε, δ)*-unlearning, where* $\zeta \sim \mathcal{N}(0, c\mathbf{I})$ *and:*

$$c = \left(\frac{2QC_f^2 G}{n^2\mu^2} + \frac{MG^2}{n^2\mu^3} + \frac{2G\widetilde{C}_f \bar{E}_n}{n\mu^2}\right)\frac{\sqrt{2\log(5/4\delta)}}{\varepsilon}.$$

Corollary 3 (Data Attribution)**.** *Suppose the assumptions of Th. 1 hold,* $T(\theta) = \ell(z_{\text{test}}, \theta) - \ell\left(z_{\text{test}}, \hat{\theta}(\mathbf{1})\right)$ *and* $C_\ell \triangleq \max_{i\in[n]} \widetilde{g}_i$. *Then,*

$$\left|T\left(\widehat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right) - T\left(\widetilde{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right)\right| \leq O\left(\frac{C_f^2 C_{T_1} C_\ell}{n^2\mu^2} + \frac{MC_{T_1}C_\ell^2}{n^2\mu^3} + \frac{C_{T_1}\widetilde{C}_f \bar{E}_n C_\ell}{n\mu^2}\right),$$

$$\begin{aligned}&\left|T\left(\widehat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right) - T\left(\widehat{\theta}(\mathbf{1})\right) - \left\langle \nabla T\left(\widehat{\theta}(\mathbf{1})\right), \widetilde{\theta}\left(\mathbf{1}^{n\setminus i}\right) - \widehat{\theta}(\mathbf{1})\right\rangle\right| \\ &\qquad \leq O\left(\frac{C_f^2 C_{T_1} C_\ell}{n^2\mu^2} + \frac{MC_{T_1}C_\ell^2}{n^2\mu^3} + \frac{C_{T_2}C_\ell^2}{n^2\mu^2} + \frac{C_{T_1}\widetilde{C}_f \bar{E}_n C_\ell}{n\mu^2}\right)\end{aligned}$$

The proofs for these corollaries rely on applying Th. 1 for the settings described in Prop. 1 - Prop. 3 (see App. I). To further demonstrate the generality of our approach, we provide guarantees for the fairness assessment task described in Sec. 2, for which currently there is no theoretical analysis. The proof is in App. I.4.

Corollary 4 (Fairness Evaluation)**.** *Suppose Assump. 1, Assump. 2, and Assump. 5 -Assump. 7 hold. If* T *be given by (2) and* $C_\ell \triangleq \max_{i\in[n]} \widetilde{g}_i$. *Then,*

$$\left|T\left(\widehat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right) - T\left(\widetilde{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right)\right| \leq O\left(\frac{C_f^3 C_\ell}{n^2\mu^2} + \frac{MC_f C_\ell^2}{n^2\mu^3} + \frac{C_f\widetilde{C}_f C_\ell \bar{E}_n}{n\mu^2}\right).$$

To the best of our knowledge, Corol. 1 - Corol. 4 provide the first theoretical guarantees for using the FIM in influence assessment tasks, offering a novel method with rigorous effectiveness proof. Additionally, our framework easily extends to other problems in machine learning and statistics beyond the specific applications discussed (e.g., data dropping (Broderick et al., 2020)).

Remark 5 (Non-Differentiable Regularizers). Our framework allows for a general regularizer term (not necessarily differentiable), and can be calculated efficiently as long as its proximal operator can be computed easily. Since training models with general regularization (beyond L_2) is an increasingly popular method for adding robustness, feature sparsity, and interoperability to models (see (Lemhadri et al., 2021; Li et al., 2021) and references therein), we see this as a major advantage of our method. Moreover, we note that this differentiates our method from previous works that considered the FIM in influence estimation tasks, such as (Bae et al., 2022; Choe et al., 2024; Park et al., 2023).

Remark 6 (General Non-Convex Setting). Unlike the Hessian, the FIM is positive semidefinite even for non-convex models. As a result, FIM-based influence estimates are typically better conditioned and more numerically stable in settings that deviate from our assumptions. We corroborate this empirically in our experiments.

5 Experiments

We evaluate the utility of the approximate Fisher influence framework through experiments on two different tasks, which constitute three of the different use cases listed in Sec. 2.1. Both Fisher-based and Hessian-based influence functions are implemented within the same codebase, differing only in the automatic differentiation components used to compute (7) and (8). Moreover, we perform experiments with both the classical L_2 regularizer and also in a case where the model is trained with an L_1 regularizer, akin to an example of a non-differentiable regularization function. Detailed experimental procedures are provided in App. J. Our objective is to demonstrate the advantages of AFIF across different tasks, as reflected by our analysis, namely, to show that it:

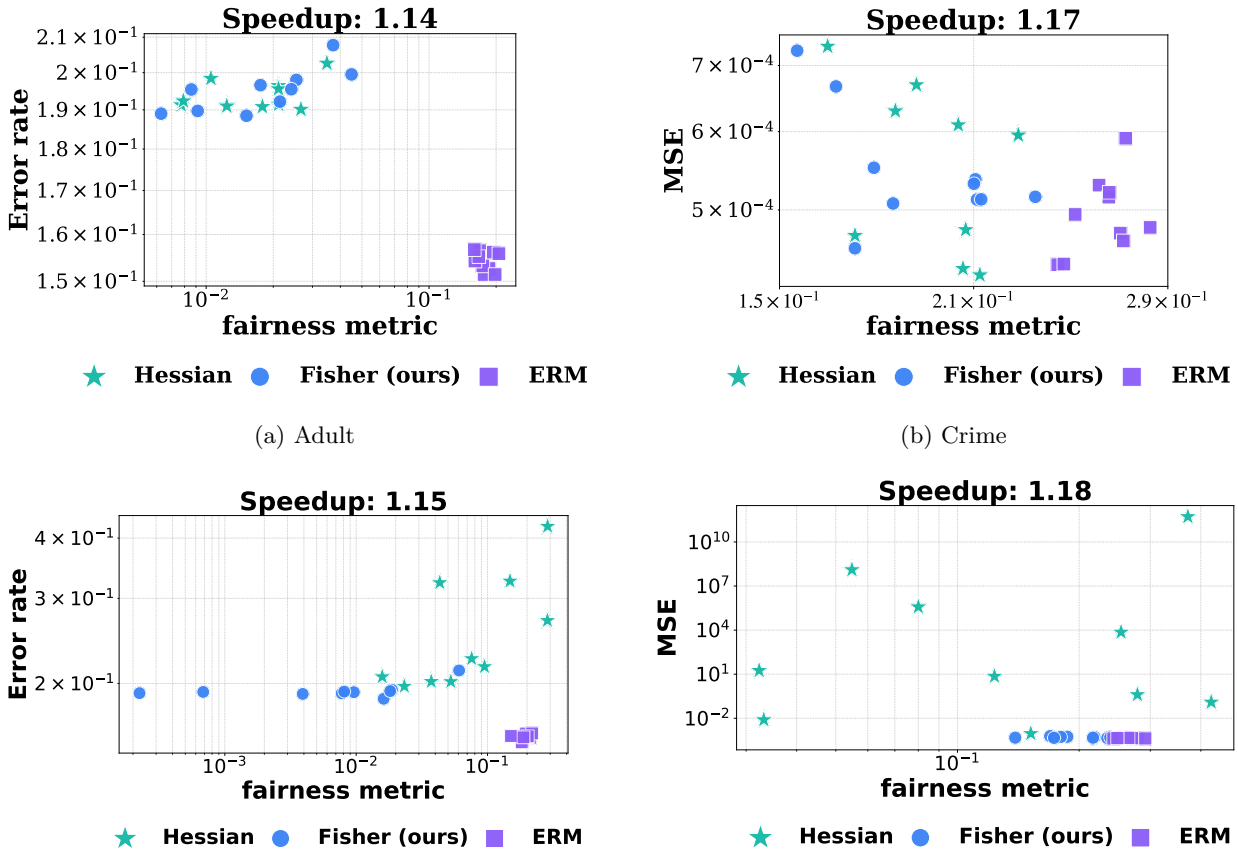
1. Achieves similar or improved utility as the Hessian-based techniques; In particular, it provides a more robust influence estimate, in contrast to Hessian-based influence estimates that might fail in some cases;
2. It has an improved computational efficiency relative to the Hessian-based techniques;
3. It reliably estimates influence even in cases with non-differentiable regularizers.

Moreover, we will demonstrate that our formulation indeed generalizes across different settings, and we will also show that the computational benefit is fundamental and not tied to implementation using `LiSSA`.

5.1 Fairness and Unlearning

In this set of experiments, we aim to identify and unlearn training points that negatively impact model fairness. To that end, we use two classical datasets from the fairness literature: the Adult (Dua and Graff, 2017) and the Crime (Redmond and Baveja, 2002) datasets. Those datasets are considered classical for multiple different fairness problems, see for example (Shah et al., 2022; 2024; Sattigeri et al., 2022). For the Adult dataset, the goal is to classify whether a person's income is greater than $50,000$\$, while keeping the classifier independent of the person's sex. For the crime dataset, the goal is to predict crime per population (which is a continuous variable) while keeping the regressor independent of race. Fairness is assessed using either of the two metrics described in Sec. 2.1: (2) and (3) for the binary and the continuous cases, respectively. Our model is a two-layer network with `SeLU` activations, similar to the architectures from (Ghosh et al., 2023; Shah et al., 2024; Sattigeri et al., 2022). We calculated the influence for each training sample using the plug-in estimator from Th. 1. We then unlearned all training samples with positive influence by applying (9). Since we just want to correct the model fairness rather than ensuring a formal (ε, δ)-unlearning guarantees, we haven't added noise to the unlearned model. All influence calculations in this set of experiments were done using the `LiSSA` algorithm. Our metric for success will be the model's fairness metric after unlearning against accuracy, with the goal of preserving accuracy while reducing the fairness metric as much as possible. We conduct these experiments with L_2 regularization, where we will compare the time and accuracy attained with the Fisher-based influence and those attained with the Hessian. Full experimental details are provided in App. J.

As shown in Fig. 1, we were able to significantly improve the fairness score without substantial performance loss. However, as predicted by our theory, the Fisher-based results are consistently faster relative to the



(a) Adult

(b) Crime

(c) Adult (alt. hyperparameters)

(d) Crime (alt. hyperparameters)

Figure 1: Model performance versus fairness metric for Fisher-based influence, Hessian-based influence, and the ERM solution from (1), evaluated on the Adult and the Crime datasets using a two-layer classifier with L_2 regularization. Runtime results are averaged over ten independent experiments. All cases demonstrate that the Fisher-based computations are faster than the Hessian-based computations, yet still yield similar overall utility. Moreover, we further see that for some configuration of hyperparameters, the estimate obtained with the Hessian might be inconsistent, while with the FIM, we get stable estimates in both cases, akin to the fact that the FIM is guaranteed to be PSD.

Hessian-based approaches, demonstrating the computational efficiency of the Fisher-based influence. Moreover, for some configurations of hyperparameters, the estimates obtained with the Hessian are not stable and lead to erroneous corrections with poor accuracy, further showing that influence estimates that rely on the FIM are more stable, akin to the fact that the FIM is PSD. Furthermore, we note that the error rates and MSE of the ERM minimizers are strictly positive, corresponding to a finite $\bar{E}_n$. Nevertheless, the AFIF effectively identifies and unlearns samples that negatively impact fairness, demonstrating its usefulness when $\bar{E}_n$ is finite.

5.2 Cross-Validation

In our second experiment, we evaluate both the computational cost and the accuracy of our influence-based CV approximation. We reuse the two-layer network from Sec. 5.1 on the Friedman-1 dataset (Friedman, 1991) and track performance across training epochs. To facilitate the need for L_1 regularization, we have artificially added random features to the training set, such that a feature selection is needed for optimal performance, and we thus trained the model with L_1 regularization. To approximate LOOCV efficiently, we use five-fold CV and apply our FIM-based update (Corol. 1, adapted to leave-k-out) and also a variant that

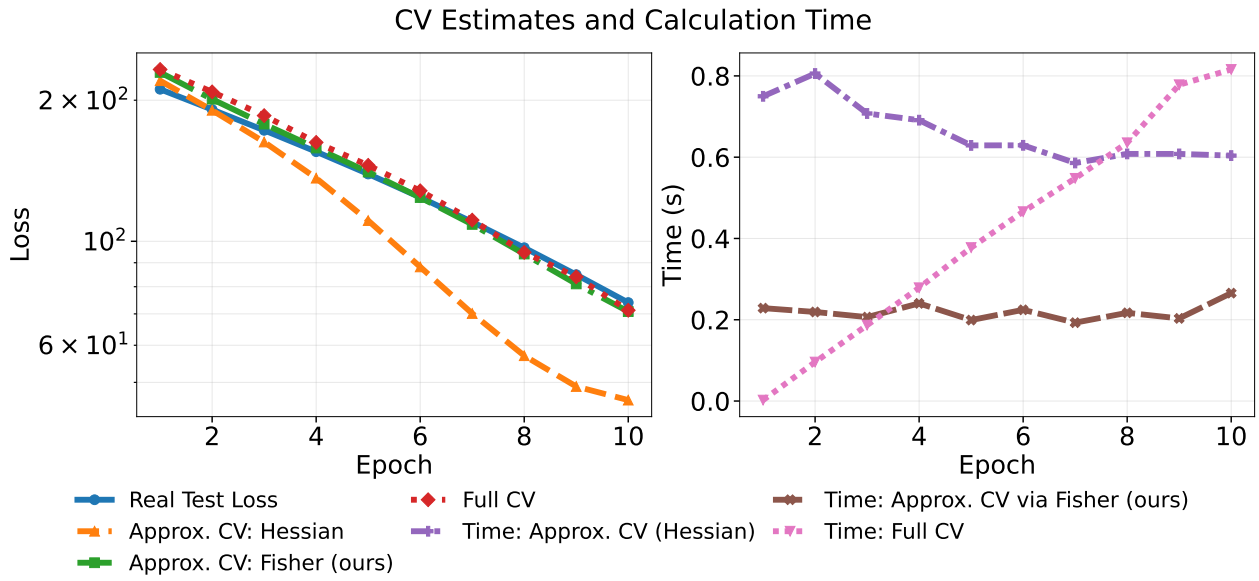


Figure 2: Held-out test loss, exact five-fold CV, FIM-based approximate CV (Corol. 1 adapted to leave-k-out), and average per-epoch compute time (averaged across folds) for a two-layer network on the Friedman-1 dataset (Friedman, 1991). The time represents the average accumulated time to create one CV estimate for a certain number of epochs.

uses the Hessian instead of the FIM. For each epoch we report: (i) the held-out test loss, (ii) exact five-fold CV (obtained by retraining the model on a subset of the training set and testing it on an held-out validation set), (iii) our FIM-based approximate CV and an Hessian-based approximate CV, and (iv) the average wall-clock time to compute each CV quantity across the five folds. On this dataset, we can form $\mathbf{F}^{-1}v$ and $\mathbf{H}^{-1}v$ via a direct linear solver rather than an iterative stochastic `LiSSA` approximation, highlighting that the observed speedups stem from the FIM formulation itself rather than the particular linear solver. As is clear from Fig. 2, our influence measurement technique allows for a reliable CV approximation and at a substantially reduced computation time, supporting the claim that our method gives an option for measuring influence in situations where one uses non-differentiable regularizers, and that further our framework generalizes across different settings.

6 Concluding Remarks

In this work, we introduced the AFIF, a method for quantifying influence in machine learning models. We presented a general formulation that relies on the FIM instead of the Hessian, and we showed why this is fundamentally faster than existing influence function baselines, yet provides similar error guarantees across a set of tasks. Moreover, our framework extends the applicability of influence measurement to a broader range of scenarios—including those involving non-differentiable regularizers. In our empirical evaluations, we have further demonstrated the computational efficiency of AFIF relative to traditional Hessian-based techniques and its usefulness in providing reliable influence estimates across a set of tasks and with general regularizers.

Future Research. Generalizing our analysis to more complex, real-world influence measurement methods that are based on the FIM and currently lack rigorous theoretical support (for example, techniques based on the Kronecker-Factored FIM (Choe et al., 2024)) is a promising future research direction that will open the door to systematically determining when and how such methods can be most effectively applied across diverse tasks. Moreover, developing computationally efficient variants of higher-order influence measurement techniques, such as those explored in (Giordano et al., 2019a; Basu et al., 2020) (see also the discussion in (Koh and Liang, 2017)), by utilizing the underlying statistical nature of the optimization problem, is another future research direction that is currently under investigation.

References

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, pages 1–40, 2017.

Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

Shun-Ichi Amari and Scott C Douglas. Why natural gradient? In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1213–1216. IEEE, 1998.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, pages 17953–17967, 2022.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, pages 1705–1749, 2005.

Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *Proceedings of the International Conference on Machine Learning*, pages 715–724, 2020.

Ahmad Beirami, Meisam Razaviyayn, Shahin Shahrampour, and Vahid Tarokh. On optimal generalizability in parametric learning. *Advances in Neural Information Processing Systems*, 30, 2017.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. JAX: composable transformations of Python+ NumPy programs. https://github.com/jax-ml/jax, 2018. GitHub repository.

Tamara Broderick, Ryan Giordano, and Rachael Meager. An automatic finite-sample robustness metric: when can dropping a little data make a big difference? *arXiv preprint arXiv:2011.14999*, 2020.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 463–480, 2015.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to GPT? LLM-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, pages 495–508, 1980.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. How to compute Hessian-vector products? In *ICLR Blogposts 2024*, 2024. URL https://iclr-blogposts.github.io/2024/blog/bench-hvp/. https://iclr-blogposts.github.io/2024/blog/bench-hvp/.

Santanu Das, Jatin Batra, and Piyush Srivastava. A direct proof of a unified law of robustness for Bregman divergence losses. *IEEE Transactions on Information Theory*, 2025.

DeepSpeed. Deepspeed tutorials: Flops profiler. https://www.deepspeed.ai/tutorials/flops-profiler/#flops-measurement, 2024.

Dheeru Dua and Casey Graff. UCI machine learning repository. https://archive.ics.uci.edu/ml, 2017. University of California, Irvine, School of Information and Computer Science. Accessed: 2025-10-04.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, pages 211–407, 2014.

Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, pages 1–67, 1991.

Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, pages 320–328, 1975.

Soumya Ghosh, Prasanna Sattigeri, Inkit Padhi, Manish Nagireddy, and Jie Chen. Influence based approaches to algorithmic fairness: A closer look. In *XAI in Action: Past, Present, and Future Applications*, 2023.

Ryan Giordano, Michael I Jordan, and Tamara Broderick. A higher-order swiss army infinitesimal jackknife. *arXiv preprint arXiv:1907.12116*, 2019a.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A swiss army infinitesimal jackknife. In *International Conference on Artificial Intelligence and Statistics*, pages 1139–1147, 2019b.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. FastIF: Scalable influence functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10333–10350, November 2021.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the International Conference on Machine Learning*, pages 1885–1894, 2017.

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. *Advances in Neural Information Processing Systems*, 32, 2019.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, pages 1–29, 2021.

Gen Li, Yuantao Gu, and Jie Ding. ℓ_1 regularization in two-layer neural networks. *IEEE Signal Processing Letters*, pages 135–139, 2021.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, pages 5776–5851, 2020.

Jérémie Mary, Clément Calauzenes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In *Proceedings of the International Conference on Machine Learning*, pages 4382–4391, 2019.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mert Pilanci and Tolga Ergen. Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pages 7695–7705, 2020.

Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, pages 660–678, 2002.

Prasanna Sattigeri, Soumya Ghosh, Inkit Padhi, Pierre Dognin, and Kush R Varshney. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems*, pages 35894–35906, 2022.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8179–8186, 2022.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, pages 1723–1738, 2002.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, volume 34, pages 18075–18086, 2021.

Abhin Shah, Yuheng Bu, Joshua K Lee, Subhro Das, Rameswar Panda, Prasanna Sattigeri, and Gregory W Wornell. Selective regression under fairness criteria. In *Proceedings of the International Conference on Machine Learning*, pages 19598–19615, 2022.

Abhin Shah, Maohao Shen, Jongha Jon Ryu, Subhro Das, Prasanna Sattigeri, Yuheng Bu, and Gregory W Wornell. Group fairness with uncertain sensitive attributes. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 208–213. IEEE, 2024.

Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. *Advances in Neural Information Processing Systems*, 33:18098–18109, 2020.

M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B*, pages 111–133, 12 1974.

Vinith Suriyakumar and Ashia C. Wilson. Algorithms that approximate data removal: New results and limitations. In *Advances in Neural Information Processing Systems*, volume 35, pages 18892–18903, 2022.

Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, pages 1–305, 2008.

Ashia Wilson, Maximilian Kasy, and Lester Mackey. Approximate cross-validation: guarantees for model assessment and selection. In *International Conference on Artificial Intelligence and Statistics*, pages 4530–4540, 2020.

Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

Emi Zeger and Mert Pilanci. Unveiling hidden convexity in deep learning: A sparse signal processing perspective, 2025. Preprint.

A Definitions and Useful Lemmas

In our paper we use the next classical definitions from the convex optimization theory (Nesterov, 2013).

Definition 1 (Matrix Operator-Norm)**.** For any matrix $\mathbf{A}$ we define its *operator-norm* by $\|\mathbf{A}\|_{\text{op}} \triangleq \sup_{v\in\mathbb{R}^d:\|v\|\neq 0} \|\mathbf{A}v\| / \|v\|$

Definition 2 (Strong convexity)**.** Let $\beta > 0$. A function $f(\cdot)$ is β-strongly convex if and only if

$$f(y) \geq f(x) + \nabla^\top f(x)(y-x) + \frac{\beta}{2}\|x-y\|^2, \ \forall (x,y) \in \text{dom}(f)$$

Definition 3 (Lipschitz)**.** A function $f(\cdot)$ is C-Lipschitz if

$$\|f(x) - f(y)\| \leq C\|x-y\|, \ \forall (x,y) \in \text{dom}(f).$$

In that case, C is called the Lipschitz constant of f and is denoted by $C \triangleq \text{Lip}(f(x))$.

Definition 4 (Smooth)**.** If $f(\cdot)$ is differentiable, then $f(\cdot)$ is K-smooth if

$$\|\nabla f(x) - \nabla f(y)\| \leq K\|x-y\|, \ \forall (x,y) \in \text{dom}(f).$$

In that case, K is called the gradient-Lipschitz constant of f and is denoted by $K \triangleq \text{Lip}_1(f(x))$.

Definition 5 (Lipschitz-Hessian)**.** If $f(\cdot)$ is twice differentiable, then $f(\cdot)$ is M-Lipschitz Hessian if

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\|_{\text{op}} \leq M\|x-y\|, \ \forall (x,y) \in \text{dom}(f)$$

In that case, M is called the Lipschitz-Hessian constant of f and is denoted by $M \triangleq \text{Lip}_2(f(x))$.

Throughout the manuscript, we will further make use of the next connections between Lipschitz coefficients and gradient bounds for differentiable functions.

Corollary 5 (Nesterov (2013))**.** *Let $f(x)$ be a differentiable function $\forall x \in dom(f)$. Then, $f(x)$ is C-Lipschitz if and only if*

$$\|\nabla f(x)\| \leq C, \ \forall x \in \text{dom}(f).$$

If $f(x)$ is twice-differentiable $\forall x \in dom(f)$ then $f(x)$ is K-smooth if and only if

$$\|\nabla^2 f(x)\|_{\text{op}} \leq K, \ \forall x \in \text{dom}(f).$$

B Example for Losses From an Exponential Family

We now present a few examples of commonly used loss functions in machine learning that can be viewed as the negative log-likelihood of an exponential-family model. A more comprehensive list can be found in Banerjee et al. (2005). Specifically, let

$$\ell\big(y, f(x;\theta)\big) \ = \ -\log\left(P\left(y|f(x;\theta)\right)\right),$$

where $P(y \mid f(x;\theta))$ belongs to a (discrete) exponential family. Throughout this paper, we adopt the following form of an exponential family:

$$\log\left(P\left(y|f(x;\theta)\right)\right) \ = \ f(x;\theta)^\top t(y) \ - \ \log\left(\sum_{\widetilde{y}=1}^{|\mathcal{Y}|} \exp\big\{ f(x;\theta)^\top t(\widetilde{y})\big\}\right) \ + \ \beta(y), \tag{13}$$

where $t(y)$ are the *natural statistics* and $f(x;\theta)$ are the *natural parameters*.[6] The term $\beta(y)$ depends only on y (thus does not affect parameter learning) and ensures proper normalization. Below, we illustrate two popular examples of loss functions (see also (Martens, 2020, Sec. 9.2)) that arise naturally from this exponential-family framework.

[6]The above is a discrete version; for continuous $\mathcal{Y}$, one replaces the sum with an integral.

1. **Cross-Entropy Loss.** A standard approach in multi-class classification over $|\mathcal{Y}|$ classes is the softmax parameterization:

$$\log\left(P\left(y|f(x;\theta)\right)\right) = \big(f(x;\theta)\big)_y - \log\left(\sum_{\widetilde{y}=1}^{|\mathcal{Y}|} \exp\left\{\big(f(x;\theta)\big)_{\widetilde{y}}\right\}\right), \quad y, \widetilde{y} \in \{1, \ldots, |\mathcal{Y}|\}.$$

Here, $f(x;\theta)$ is a vector of length $|\mathcal{Y}|$. By defining e_y as the one-hot vector with a 1 in the y-th entry and 0 elsewhere, we see that

$$\log P\left(y \mid f(x;\theta)\right) = f(x;\theta)^\top e_y - \log\left(\sum_{\widetilde{y}=1}^{|\mathcal{Y}|} \exp\left\{f(x;\theta)^\top e_{\widetilde{y}}\right\}\right),$$

thus matching (13) with natural statistics $t(y) = e_y$ and natural parameters $f(x;\theta)$. The corresponding loss,

$$\ell\big(y, f(x;\theta)\big) = -\log\left(P\left(y|f(x;\theta)\right)\right),$$

is the well-known cross-entropy.

2. **Mean Squared Error (MSE).** In a regression setting with a continuous target $y \in \mathbb{R}$, a unit-variance Gaussian model with mean $\mu = f(x;\theta)$ leads to

$$\log\left(P\left(y|f(x;\theta)\right)\right) \;=\; -\tfrac{1}{2}\big(y - f(x;\theta)\big)^2 \;=\; f(x;\theta)\,y \;-\; \frac{y^2}{2} \;-\; \frac{\big(f(x;\theta)\big)^2}{2}.$$

Comparing with (13), this corresponds to an exponential family whose natural statistics are $\big(y, y^2\big)$ and whose natural parameters are $\big(f(x;\theta), -\tfrac{1}{2}\big)$. The negative log-likelihood here,

$$\ell\big(y, f(x;\theta)\big) = -\log\left(P\left(y|f(x;\theta)\right)\right) = \tfrac{1}{2}\big(y - f(x;\theta)\big)^2 \;+\; (\text{constant}),$$

is precisely the mean squared error (MSE) loss up to an additive constant.

B.1 Bregman Losses

Following (Banerjee et al., 2005, Thm. 4), whenever the representation $P(y|f(x;\theta))$ correspond to a regular exponential family, then the loss $-\log(P(y|f(x;\theta)))$ can be expressed as

$$-\log(P(y|f_\theta(x))) = d_\varphi(t(y), \mu(f_\theta(x))) + \log(b_\varphi(t(y))) + C$$

where $\mu(f_\theta(x)) = \mathbb{E}\left[t(y)\right]$ is the expected value of $t(y)$ using the underlying exponential family distribution, $d_\varphi(\cdot,\cdot)$ is a Bregman divergence and C is a constant. As shown by (Banerjee et al., 2005, Table 1) (see also Das et al. (2025)), this result implies that many classical losses in machine learning, including cross-entropy and mean squared error, can be viewed as special cases of Bregman divergences, and further belong to the exponential family framework discussed in our work.

B.2 Properties of the Cross-Entropy and MSE Losses

We now demonstrate how the assumptions on loss minimization, Hessian boundedness, and simplified second-order gradients follow for the two loss functions introduced above.

1. **Cross-Entropy Loss.** Recall the parameterization

$$\log\left(P\left(y|f(x;\theta)\right)\right) = (f(x;\theta))_y - \log\left(\sum_{\widetilde{y}\in\mathcal{Y}} \exp\{(f(x;\theta))_{\widetilde{y}}\}\right),$$

and let

$$\ell\big(y, f(x;\theta)\big) = -\log\left(P\left(y|f(x;\theta)\right)\right) = \log\left(\sum_{\widetilde{y}\in\mathcal{Y}} \exp\left\{(f(x;\theta))_{\widetilde{y}}\right\}\right) - (f(x;\theta))_y.$$

We focus first on the gradient of the *log-probability* itself, which is given by :

$$\begin{aligned}\nabla_f \log\left(P\left(y|f(x;\theta)\right)\right) &= \nabla_f\left[(f(x;\theta))_y - \log\left(\sum_{\widetilde{y}\in\mathcal{Y}} \exp\left\{(f(x;\theta))_{\widetilde{y}}\right\}\right)\right] \\ &= e_y - \mathrm{softmax}\big(f(x;\theta)\big),\end{aligned}$$

where e_y is the one-hot vector selecting entry y, and $\mathrm{softmax}\big(f(x;\theta)\big)$ is the vector of class probabilities assigned by the model.

Zero Gradients Under Perfect Prediction. For any training example (x_i, y_i), if the model classifies it with perfect confidence, i.e.

$$\big(\mathrm{softmax}(f(x_i;\hat{\theta}(\mathbf{1})))\big)_{y_i} \;=\; 1,$$

then $\nabla_f \log P\big(y_i \mid f(x_i;\hat{\theta}(\mathbf{1}))\big) = 0$. Consequently, if the model perfectly predicts *all* training labels, then all these gradients vanish simultaneously.

Bounded Hessian. Next, we show that the second derivative (the Hessian) of $\log\left(P\left(y|f(x;\theta)\right)\right)$ with respect to f is bounded in norm. From the above,

$$\nabla_f \log\left(P\left(y|f(x;\theta)\right)\right) \;=\; e_y - \mathrm{softmax}\big(f(x;\theta)\big),$$

so taking another derivative,

$$\nabla_f^2 \log\left(P\left(y|f(x;\theta)\right)\right) \;=\; -\nabla_f\big[\mathrm{softmax}(f(x;\theta))\big].$$

Denote $\mathbf{C}_f \;\triangleq\; \nabla_f\big[\mathrm{softmax}(f(x;\theta))\big]$. By the well-known derivative of softmax, the (i,j)th entry of $\mathbf{C}_f$ is

$$(\mathbf{C}_f)_{ij} \;=\; \frac{\partial}{\partial\big(f(x;\theta)\big)_j}\Big[\mathrm{softmax}(f(x;\theta))_i\Big] \;=\; \mathrm{softmax}(f(x;\theta))_i\left[\delta_{ij} \;-\; \mathrm{softmax}(f(x;\theta))_j\right],$$

which implies:

$$\begin{aligned}(\mathbf{C}_f)_{ii} &= \mathrm{softmax}(f(x;\theta))_i \cdot \big[1 - \mathrm{softmax}(f(x;\theta))_i\big], \\ (\mathbf{C}_f)_{ij} &= -\mathrm{softmax}(f(x;\theta))_i \cdot \mathrm{softmax}(f(x;\theta))_j \quad (i \neq j).\end{aligned}$$

Because each $\mathrm{softmax}(f(x;\theta))_i \in [0,1]$, the entries of $\mathbf{C}_f$ lie in $[-1,1]$, and indeed one can show $\|\mathbf{C}_f\|$ is bounded by a constant (depending only on $|\mathcal{Y}|$, not on the dimension of the parameters). Hence $\nabla_f^2 \log\left(P\left(y|f(x;\theta)\right)\right) = -\mathbf{C}_f$ is also bounded in norm, establishing the desired Hessian bound.

2. **Mean Squared Error (MSE).** For the MSE loss arising from a unit-variance Gaussian,

$$\log\left(P\left(y|f(x;\theta)\right)\right) \;=\; -\tfrac{1}{2}\left[y - f(x;\theta)\right]^2,$$

the gradient with respect to $f(x;\theta)$ is simply

$$\nabla_f \;\log\left(P\left(y|f(x;\theta)\right)\right) \;=\; y \;-\; f(x;\theta).$$

Hence, if at $\hat{\theta}(\mathbf{1})$ the model predictions perfectly match all targets, this gradient becomes zero for each training pair, indicating perfect minimization of the training error.

Bounded Hessian. Since

$$\nabla_f^2 \log\left(P\left(y|f(x;\theta)\right)\right) \;=\; -\nabla_f^2\left[\tfrac{1}{2}\left(y-f(x;\theta)\right)^2\right] \;=\; -\left(-\mathbf{I}_d\right) \;=\; \mathbf{I}_d,$$

the Hessian with respect to f is simply the identity (for the one-dimensional f). Its norm is therefore trivially bounded by 1, and it does not depend on the dimension d of the parameters in θ. Moreover, the Hessian can be evaluated with no complicated operations—just the constant identity matrix at each sample.

C Gradient Bound for Minimizing Losses From the Exponential Family

Given a training set $\{(x_i, y_i)\}_{i=1}^n$ and the loss function (13) we derive gradient of the empirical risk (1) which we aim to minimize. To that end, we note that

$$\begin{aligned}
n\nabla_\theta L(\mathscr{D},\theta,\mathbf{1}) &= \nabla_\theta\left(\sum_{i=1}^n f^\top(x_i;\theta)t(y_i) - \log\left(\sum_{\tilde{y}\in\mathcal{Y}}\exp\{f^\top(x_i;\theta)t(\tilde{y})\}\right) + \beta(y_i)\right)\\
&= \sum_{i=1}^n\left(\nabla_\theta^\top f(x_i;\theta)t(y_i) - \frac{\sum_{\tilde{y}\in\mathcal{Y}}\nabla^\top f(x_i;\theta)t(\tilde{y})\exp\{f^\top(x_i;\theta)t(\tilde{y})\}}{\sum_{\tilde{y}_1\in\mathcal{Y}}\exp\{f^\top(x_i;\theta)t(\tilde{y}_1)\}}\right)\\
&= \sum_{i=1}^n\nabla_\theta^\top f(x_i;\theta)\left(t(y_i) - \frac{\sum_{\tilde{y}\in\mathcal{Y}}t(\tilde{y})\exp\{f^\top(x_i;\theta)t(\tilde{y})\}}{\sum_{\tilde{y}_1\in\mathcal{Y}}\exp\{f^\top(x_i;\theta)t(\tilde{y}_1)\}}\right)\\
&= \sum_{i=1}^n\nabla_\theta^\top f(x_i;\theta)(t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])
\end{aligned}$$

and the norm of this gradient is upper-bounded by

$$n\left\|\nabla_\theta L(\mathscr{D},\theta,\mathbf{1})\right\| \le \sum_{i=1}^n\left\|\nabla_\theta f(x_i;\theta)\right\|\left\|t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right]\right\|$$

Thus, whenever the features are Lipschitz, we have

$$\left\|\nabla_\theta L(\mathscr{D},\theta,\mathbf{1})\right\| \le \frac{C_f}{n}\sum_{i=1}^n\left\|t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right]\right\|$$

and we expect this upper bound to be small at the minimizer $\theta = \hat{\theta}(\mathbf{1})$. As a consequence, we note that the sum $\sum_{i=1}^n\left\|t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right]\right\|$ is in fact an upper bound on the gradient. Thus, when evaluated on $\theta = \widehat{\theta}(\mathbf{1})$, we expect this quantity to be small, establishing the validity of the arguments from Sec. 4.2, which claims that the additional factor that depends on $\bar{E}_n$ is indeed of negligible contribution to the final influence estimate.

D Proof of Closeness of $\hat{\theta}\left(\mathbf{1}\right)$ and $\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)$

We will use the next lemma throughout our proofs.

Lemma 2. *Let $\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)$ defined as in (1) and let $\frac{1}{n}\ell(z_i,\theta)$ be a differentiable function in θ for any $z_i \in \mathscr{D}$ and $\frac{1}{n}\ell(z_i,\theta) + \lambda\pi(\theta)$ be a μ-strongly convex function in θ for any $z_i \in \mathscr{D}$. Then, $\forall i \in [n]$*

$$\left\|\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right) - \hat{\theta}\left(\mathbf{1}\right)\right\| \le \frac{2}{\mu}\cdot\max_{i\in[n]}\left\|\frac{1}{n}\nabla_\theta\ell\left(z_i,\hat{\theta}\left(\mathbf{1}\right)\right)\right\|$$

Proof. Similarly to the developments from (Wilson et al., 2020, App. B.1), we get that

$$\begin{aligned}\left\|\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)-\hat{\theta}\left(\mathbf{1}\right)\right\|^2 &\leq \frac{2}{\mu}\left|\left(\hat{\theta}\left(\mathbf{1}\right)-\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right)^\top\left(\nabla_\theta(L(\mathscr{D},\hat{\theta}\left(\mathbf{1}\right),\lambda,\mathbf{1}^{n\setminus i})-L(\mathscr{D},\hat{\theta}\left(\mathbf{1}\right),\lambda,\mathbf{1}))\right)\right| \\ &\leq \frac{2}{\mu n}\left|\left(\hat{\theta}\left(\mathbf{1}\right)-\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right)^\top\nabla_\theta\ell\left(z_i,\hat{\theta}\left(\mathbf{1}\right)\right)\right| \\ &\leq \frac{2}{\mu n}\left\|\hat{\theta}\left(\mathbf{1}\right)-\hat{\theta}\left(\mathbf{1}^{n\setminus i}\right)\right\|\left\|\nabla_\theta\ell\left(z_i,\hat{\theta}\left(\mathbf{1}\right)\right)\right\|\end{aligned}$$

where all the steps are by Cauchy-Schwartz inequality. The proof follows by maximizing over i. □

We note that whenever $\frac{1}{n}\ell(z_i,\theta)$ is Lipschitz, the upper bound is finite. Moreover, since we normalize by n, the bound will go to zero with n whenever the gradient grows as $o(n)$, as is usually the case in many popular machine learning problems (see (Giordano et al., 2019b, Sec. 3)). We further note that under a more restrictive assumption that the $\ell(z_i,\theta)$ are Lipschitz then the bound is given by $\frac{2\tilde{C}}{\mu n}$ for $\tilde{C}=\max_{i\in[n]}\left\|\nabla_\theta\ell\left(z_i,\hat{\theta}\left(\mathbf{1}\right)\right)\right\|$ and $\tilde{C}<\infty$.

E Proof of Lem. 1

The proof uses the following lemma from Wilson et al. (2020):

Lemma 3 (Optimizer Comparison, Wilson et al. (2020))**.** *Let*

$$x_{\varphi_1}\in\arg\min_x\varphi_1(x),\quad x_{\varphi_2}\in\arg\min_x\varphi_2(x).$$

If each φ_i is μ-strongly convex and $\varphi_2-\varphi_1$ is differentiable, then

$$\frac{\mu}{2}\|x_{\varphi_1}-x_{\varphi_2}\|_2^2\leq\left|(x_{\varphi_1}-x_{\varphi_2})^\top\left(\nabla(\varphi_2-\varphi_1)(x_{\varphi_1})\right)\right|.$$

Proof. For the sake of the proof, we will assume that the FIM and the Hessian are invertible matrices. Under the probabilistic interpretation of the loss elements, the overall loss function for $w^n=\mathbf{1}^{n\setminus i}$ is

$$L(\mathscr{D},\theta,\lambda,\mathbf{1}^{n\setminus i})\triangleq\frac{1}{n}\sum_{j\neq i}-\log(P\left(y_j|f(x_j;\theta)\right))+\lambda\pi(\theta)$$

and we assume that $P\left(y|f(x;\theta)\right)$ belongs to an exponential family whose natural parameters are the features $f(x;\theta)$, namely, $\log(P\left(y|f(x;\theta)\right))=f^\top(x;\theta)t(y)-\log(\sum_{\tilde{y}=1}^{|\mathcal{Y}|}\exp\left\{f^\top(x;\theta)t(\tilde{y})\right\})+\beta(y)$ for some natural statistics $t(y)$. For this model, we have

$$\nabla_\theta\log(P\left(y|f(x;\theta)\right))=\nabla_\theta f(x;\theta)\nabla_f\log(P\left(y|f(x;\theta)\right)).$$

Thus, the approximated FIM, $\mathbf{F}(\mathscr{D},\theta)$, is given by

$$\begin{aligned}\mathbf{F}(\mathscr{D},\theta)&=\frac{1}{n}\sum_{i=1}^n\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[\nabla_\theta f(x_i;\theta)\nabla_f\log(P\left(\mathsf{y}|f(x_i;\theta)\right))\nabla_f^\top\log(P\left(\mathsf{y}|f(x_i;\theta)\right))\nabla_\theta^\top f(x_i;\theta)\right] \\ &=\frac{1}{n}\sum_{i=1}^n\nabla_\theta f(x_i;\theta)\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[-\nabla_f^2\log(P\left(\mathsf{y}|f(x_i;\theta)\right))\right]\nabla_\theta^\top f(x_i;\theta) \qquad (14a) \\ &=-\frac{1}{n}\sum_{i=1}^n\nabla_\theta f(x_i;\theta)\nabla_f^2\log(P\left(y_i|f(x_i;\theta)\right))\nabla_\theta^\top f(x_i;\theta)\end{aligned}$$

where (14a) is by using classical properties of the exponential family, and where the last equality is since the Hessian of an exponential family with respect to the natural parameters f is independent of y (see App. G).

Moreover, we note that the Hessian of the loss is given by

$$\begin{aligned}\mathbf{H}(\theta, \mathbf{1}^{n\setminus i}) &= \nabla_\theta^2 L(\mathscr{D}, \theta, \mathbf{1}^{n\setminus i}) \\ &= \nabla_\theta^2 L(\mathscr{D}, \theta, \mathbf{1}^{n\setminus i} - \mathbf{1}) + \nabla_\theta^2 L(\mathscr{D}, \theta, \mathbf{1}) \\ &= \nabla_\theta^2 L(\mathscr{D}, \theta, \mathbf{1}^{n\setminus i} - \mathbf{1}) + \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 f(x_i;\theta)\nabla_f \log(P(y_i|f(x_i;\theta))) + \mathbf{F}(\mathscr{D};\theta).\end{aligned}$$

We start by defining the next functions

$$\begin{aligned}\psi_1(\theta) &\triangleq 2L(\mathscr{D}, \theta, \lambda, \mathbf{1}^{n\setminus i}) = 2L(\mathscr{D}, \theta, \mathbf{1}^{n\setminus i}) + 2\lambda\pi(\theta), \\ \psi_2(\theta) &\triangleq -2b^\top(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) \cdot (\hat{\theta}(\mathbf{1}) - \theta) + (\hat{\theta}(\mathbf{1}) - \theta)^\top \nabla^2 L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})(\hat{\theta}(\mathbf{1}) - \theta) + 2\lambda\pi(\theta), \\ \psi_3(\theta) &\triangleq -2b^\top(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) \cdot (\hat{\theta}(\mathbf{1}) - \theta) + (\hat{\theta}(\mathbf{1}) - \theta)^\top \cdot \mathbf{F} \cdot (\hat{\theta}(\mathbf{1}) - \theta) + 2\lambda\pi(\theta) \\ &= (\theta - (\hat{\theta}(\mathbf{1}) - \mathbf{F}^{-1} \cdot b(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})))^\top \mathbf{F}(\theta - (\hat{\theta}(\mathbf{1}) - \mathbf{F}^{-1} \cdot b(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}))) + 2\lambda\pi(\theta) + J\end{aligned}$$

where J is a constant (which is independent of θ) and $\mathbf{F}$ is an abbreviation for $\mathbf{F}(\mathscr{D}, \hat{\theta}(\mathbf{1}))$. We first note that the minimizer of ψ_1 is $\hat{\theta}(\mathbf{1}^{n\setminus i})$ and that the minimizer of ψ_3 is $\tilde{\theta}(\mathbf{1}^{n\setminus i})$ from (9).

We note that Assump. 1 and Assump. 2 guarantees that the overall loss, L, is μ-strongly convex and that the difference $L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \lambda, \mathbf{1}^{n\setminus i}) - L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \lambda, \mathbf{1})$ is differentiable. Thus, using Lem. 2, which follows by applying the optimizer comparison lemma with $L(\mathscr{D}, \theta, \lambda, \mathbf{1}^{n\setminus i})$ and $L(\mathscr{D}, \theta, \lambda, \mathbf{1})$ allows us to derive the following upper bound

$$\|\hat{\theta}(\mathbf{1}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| \leq \frac{2}{n\mu} \cdot \|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\| \triangleq \frac{2g_i}{\mu}. \tag{15}$$

The optimizer comparison lemma (Wilson et al., 2020, Lem. 1) with ψ_1 and ψ_3 and Cauchy-Schwartz inequality yields

$$\begin{aligned}\frac{\mu}{2}\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\|^2 &\leq |(\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i}))^\top (\nabla(\psi_3 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i})))| \\ &\leq \|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\| \|(\nabla(\psi_3 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i})))\|\end{aligned}$$

We divide both sides by $\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\|$, and by using the triangle inequality we get

$$\begin{aligned}\frac{\mu}{2}\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\| &\leq \|\nabla(\psi_3 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i}))\| \\ &\leq \|\nabla(\psi_3 - \psi_2)(\hat{\theta}(\mathbf{1}^{n\setminus i})) + \nabla(\psi_2 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i}))\| \qquad (16) \\ &\leq \|\nabla(\psi_3 - \psi_2)(\hat{\theta}(\mathbf{1}^{n\setminus i}))\| + \|\nabla(\psi_2 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i}))\| \\ &\leq \|\nabla^2 L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) - \mathbf{F}(\mathscr{D}, \hat{\theta}(\mathbf{1}))\| \|\hat{\theta}(\mathbf{1}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| + \|(\nabla(\psi_2 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i})))\| \\ &= \|\nabla^2 L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i} - \mathbf{1}) + \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 f(x_i;\hat{\theta}(\mathbf{1}))\nabla_f \log(P(y_i|f(x_i;\hat{\theta}(\mathbf{1}))))\| \|\hat{\theta}(\mathbf{1}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| \\ &\qquad + \|(\nabla(\psi_2 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i})))\| \\ &\leq \frac{g_i}{n\mu} \cdot \| - \nabla_\theta f(x_i;\hat{\theta}(\mathbf{1}))\nabla_f^2 \log(P(y_i|f(x_i;\hat{\theta}(\mathbf{1}))))\nabla_\theta^\top f(x_i;\hat{\theta}(\mathbf{1})) \qquad (17) \\ &\qquad + \sum_{i=1}^{n} \nabla_\theta^2 f(x_i;\hat{\theta}(\mathbf{1}))\nabla_f \log(P(y_i|f(x_i;\hat{\theta}(\mathbf{1}))))\| + \frac{Mg_i^2}{2\mu^2}\end{aligned}$$

where (16) is since the differences $\psi_3 - \psi_2$ and $\psi_2 - \psi_1$ are differentiable and where (17) is by using the next bound:

$$\begin{aligned}
&\|(\nabla(\psi_2 - \psi_1)(\hat{\theta}(\mathbf{1}^{n\setminus i})))\| \\
&= 2\|b(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) + \nabla^2 L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})(\hat{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1})) - \nabla L(\mathscr{D}, \hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})\| \\
&= 2\|\nabla L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) + \nabla^2 L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})(\hat{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1})) - \nabla L(\mathscr{D}, \hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})\| && \text{(18a)} \\
&\leq M \cdot \left\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1})\right\|^2 && \text{(18b)} \\
&\leq \frac{4Mg_i^2}{\mu^2}
\end{aligned}$$

where (18a) is by the structure and the convexity and differentiability assumptions on L, leading to $\nabla L(\mathscr{D}, \hat{\theta}(\mathbf{1}), \mathbf{1}) = 0$, (18b) implied by the Hessian Lipschitzness of L (see also [3, Lem. 1.2.4]) and the last inequality is by Lem. 2.

We further use the triangle inequality to get the next upper bound

$$\begin{aligned}
\frac{\mu}{2}\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\| \leq \frac{g_i}{n\mu}(&\|\nabla_\theta f(x_i; \hat{\theta}(\mathbf{1}))\nabla_f^2 \log(P(y_i|f(x_i; \hat{\theta}(\mathbf{1}))))\nabla_\theta^\top f(x_i; \hat{\theta}(\mathbf{1}))\| \\
&+ \sum_{i=1}^n \|\nabla_\theta^2 f(x_i; \hat{\theta}(\mathbf{1}))\nabla_f \log(P(y_i|f(x_i; \hat{\theta}(\mathbf{1}))))\|) + \frac{Mg_i^2}{2\mu^2}
\end{aligned}$$

and by using Assump. 5, Assump. 7 and the boundedness of the Hessian of the loss relative to the features (see App. B.2) we get the final bound

$$\begin{aligned}
\|\hat{\theta}(\mathbf{1}^{n\setminus i}) - \tilde{\theta}(\mathbf{1}^{n\setminus i})\| &\leq \frac{2Qg_i}{n\mu^2}\|\nabla_\theta f(x_i; \hat{\theta}(\mathbf{1}))\|^2 + \frac{Mg_i^2}{\mu^3} \\
&\quad + \frac{2g_i}{n\mu^2}\sum_{i=1}^n \|\nabla_\theta^2 f(x_i; \hat{\theta}(\mathbf{1}))\nabla_f \log(P(y_i|f(x_i; \hat{\theta}(\mathbf{1}))))\| \\
&\leq \frac{2QC_f^2 g_i}{n\mu^2} + \frac{Mg_i^2}{\mu^3} + \frac{2g_i\tilde{C}_f}{n\mu^2}\sum_{i=1}^n \|\nabla_f \log(P(y_i|f(x_i; \hat{\theta}(\mathbf{1}))))\|
\end{aligned}$$

where Q is a constant s.t. $\left\|\nabla_f^2 \log(P(y|f(x;\theta)))\right\| \leq Q$. □

We now emphasize how the third term disappears whenever our model interpolates the training data (namely, $\ell(z_i, \hat{\theta}(\mathbf{1})) = 0, \forall i \in [n]$). In that case, we have $P(y_i|f(x_i; \hat{\theta}(\mathbf{1}))) = 1, \ \forall i \in [n]$ [7]. Thus, following the notation of App. G we have that $\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\hat{\theta}(\mathbf{1})}}[t(\mathsf{y})] = t(y_i)$ and since $\nabla_f \log(P(y_i|f(x_i; \hat{\theta}(\mathbf{1})))) = t(y_i) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\hat{\theta}(\mathbf{1})}}[t(\mathsf{y})]$ we get that the third term is zero.

F Comment on Lem. 1 When $\pi(\theta)$ is Twice-Differentiable

Whenever $\pi(\theta)$ is twice differentiable, an equivalent argument to that of Lem. 1 can be stated without the usage of a proximal operator. Specifically, since in this case the entire loss elements $\frac{1}{n}\ell(z_i, \theta) + \lambda\pi(\theta)$ can be approximated using a second-order Taylor expansion, and a solution that uses $\mathbf{C}\left(\hat{\theta}(\mathbf{1}), \mathbf{1}\right) = \mathbf{F}\left(\mathscr{D}, \hat{\theta}(\mathbf{1})\right) + \lambda\nabla^2\pi\left(\hat{\theta}(\mathbf{1})\right)$ leads to similar arguments as those from App. E. For this approximation, we define the solution via

$$\tilde{\theta}\left(\mathbf{1}^{n\setminus i}\right) \triangleq \hat{\theta}(\mathbf{1}) - \left(\mathbf{F}\left(\mathscr{D}, \hat{\theta}(\mathbf{1})\right) + \lambda\nabla^2\pi\left(\hat{\theta}(\mathbf{1})\right)\right)^{-1} b\left(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}\right)$$

and a similar analysis to that of App. E can be carried out and to lead to similar guarantees. An example for such arguments from a similar application can be found in (Wilson et al., 2020, Thm. 2).

[7] In the continuous case, this amounts to $P(y_i|f(x_i; \hat{\theta}(\mathbf{1})))$ converging to a delta-function, concentrated around the value y_i

G Fisher Information Matrix for Exponential Families

Using the fact that the distribution $P\left(y|f(x;\theta)\right)$ belongs to an exponential family, namely

$$\log(P\left(y|f(x;\theta)\right)) = f^\top(x;\theta)t(y) - \log\left(\sum_{\tilde{y}=1}^{|\mathcal{Y}|}\exp\left\{f^\top(x;\theta)t(\tilde{y})\right\}\right) + \beta(y),$$

we can directly evaluate the terms $\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[\nabla_f \log(P\left(\mathsf{y}|f(x;\theta)\right))\nabla_f^\top \log(P\left(\mathsf{y}|f(x;\theta)\right))\right]$ and $\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[-\nabla_f^2 \log(P\left(\mathsf{y}|f(x_i;\theta)\right))\right]$ to establish the desired equality. First, we find that:

$$\begin{aligned}\nabla_f \log(P\left(y|f(x;\theta)\right)) &= \nabla_f\left(f^\top(x;\theta)t(y) - \log\left(\sum_{y\in\mathcal{Y}}\exp\left\{f^\top(x;\theta)t(y)\right\}\right)\right)\\ &= t(y) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right]\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[\nabla_f \log(P\left(\mathsf{y}|f(x;\theta)\right))\nabla_f^\top \log(P\left(\mathsf{y}|f(x;\theta)\right))\right]\\ &\quad= \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[(t(\mathsf{y}) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])(t(\mathsf{y}) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])^\top\right].\end{aligned}$$

Next, we observe that:

$$\begin{aligned}-\nabla_f^2 \log(P\left(y|f(x;\theta)\right)) &= \nabla_f\left(\frac{\sum_{\tilde{y}\in\mathcal{Y}} t(\tilde{y})\exp\left\{f^\top(x;\theta)t(\tilde{y})\right\}}{\sum_{\tilde{y}_1\in\mathcal{Y}}\exp\left\{f^\top(x;\theta)t(\tilde{y}_1)\right\}}\right)\\ &= \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})t^\top(\mathsf{y})\right] - (\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])(\mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])^\top\\ &= \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[(t(\mathsf{y}) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])(t(\mathsf{y}) - \mathbb{E}_{\mathsf{y}\sim P_{\mathsf{y}|\mathsf{x}=x_i;\theta}}\left[t(\mathsf{y})\right])^\top\right].\end{aligned}$$

Moreover, we note that this final result holds for any y. This concludes the proof. □

H Proof of Th. 1

Proof. We start by writing the Taylor expansion of $T(\tilde{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})$ around $\hat{\theta}(\mathbf{1}^{n\setminus i})$ to get [8]:

$$\begin{aligned}T(\tilde{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i}) &= T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i}) + \nabla_\theta^\top T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))\\ &\quad+ \frac{1}{2}(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))^\top \nabla_\theta^2 T(\theta_{\text{mid}}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))\end{aligned} \tag{19}$$

where $\theta_{\text{mid}}(\mathbf{1}^{n\setminus i}) = \hat{\theta}(\mathbf{1}^{n\setminus i}) + \kappa\cdot(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))$ for some $\kappa\in[0,1]$. By (19) and by the Lipschitz assumptions on T we get

$$\begin{aligned}&\|T(\tilde{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i}) - T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})\|\\ &\quad= \|\nabla_\theta^\top T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))\\ &\qquad\qquad+ \frac{1}{2}(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))^\top \nabla_\theta^2 T(\theta_{\text{mid}}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))\|\\ &\quad\le \|\nabla_\theta T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})\|\|\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| \qquad (20a)\\ &\qquad\qquad+ \frac{1}{2}\|\nabla_\theta^2 T(\theta_{\text{mid}}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i})\|_{\text{op}}\|\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\|^2\\ &\quad\le C_{T_1}\|\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| + \frac{1}{2}C_{T_2}\|\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\|^2. \qquad (20b)\end{aligned}$$

[8] the existence of the Taylor expansion of T is guaranteed by Assump. 6

The proof is completed by substituting (10) into (20b). To prove (12), we write the expansion of $T(\hat{\theta}(\mathbf{1}^{n\setminus i}))$ around $\hat{\theta}(\mathbf{1})$, to get

$$\begin{aligned}
&\|T(\hat{\theta}(\mathbf{1}^{n\setminus i}), \mathbf{1}^{n\setminus i}) - T(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i}) - \nabla_\theta T(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}))\| \\
&\quad = \|\nabla_\theta T(\hat{\theta}(\mathbf{1}), \mathbf{1}^{n\setminus i})(\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})) \\
&\qquad\qquad + \frac{1}{2}(\hat{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}))^\top \nabla_\theta^2 T(\tilde{\theta}_{\text{mid}}, \mathbf{1}^{n\setminus i})(\hat{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}))\| \\
&\quad \leq C_{T_1}\|\tilde{\theta}(\mathbf{1}^{n\setminus i}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\| + \frac{1}{2}C_{T_2}\|\hat{\theta}(\mathbf{1}) - \hat{\theta}(\mathbf{1}^{n\setminus i})\|^2
\end{aligned}$$

where $\tilde{\theta}_{\text{mid}} = \hat{\theta}(\mathbf{1}^{n\setminus i}) + \kappa \cdot (\hat{\theta}(\mathbf{1}) - \hat{\theta}(\mathbf{1}^{n\setminus i}))$ for some $\kappa \in [0, 1]$. Substituting (10) and (15) concludes the proof. □

I Proofs of Corol. 1 - Corol. 4

I.1 Proof of Corol. 1

We now show how to use Th. 1 to approximate LOOCV with similar guarantees to the Hessian-based technique from (Wilson et al., 2020). Throughout the proof, we will use a refined version of (20b), which requires the Lipschitzness of the $T(\cdot, \mathbf{1}^{n\setminus i})$ only at $\hat{\theta}(\mathbf{1})$. We start by defining ACV $\triangleq \frac{1}{n}\sum_{i=1}^n \ell(z_i, \tilde{\theta}(\mathbf{1}^{n\setminus i}))$ and recall that CV $\triangleq \frac{1}{n}\sum_{i=1}^n \ell(z_i, \hat{\theta}(\mathbf{1}^{n\setminus i}))$. Then, similarly to App. H we get

$$\begin{aligned}
&|\text{ACV} - \text{CV}| \\
&= \left|\frac{1}{n}\sum_{i=1}^n \ell(z_i, \tilde{\theta}(\mathbf{1}^{n\setminus i})) - \ell(z_i, \hat{\theta}(\mathbf{1}^{n\setminus i}))\right| \\
&\leq \frac{1}{n}\sum_{i=1}^n \left|\ell(z_i, \tilde{\theta}(\mathbf{1}^{n\setminus i})) - \ell(z_i, \hat{\theta}(\mathbf{1}^{n\setminus i}))\right| \\
&\leq \frac{1}{n}\sum_{i=1}^n \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}^{n\setminus i}))\right\| \left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\tilde{g}_i\tilde{C}_f\bar{E}_n}{n\mu^2}\right) \qquad (21a) \\
&\qquad + \frac{1}{2}\text{Lip}(\nabla_\theta \ell(z_i, \theta))\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\tilde{g}_i\tilde{C}_f\bar{E}_n}{n\mu^2}\right)^2 \\
&\leq \frac{1}{n}\sum_{i=1}^n \left(\left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\| + \text{Lip}(\nabla_\theta \ell(z_i, \theta))\left(\frac{4\tilde{g}_i^2}{n^2\mu^2}\right)\right)\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\tilde{g}_i\tilde{C}_f\bar{E}_n}{n\mu^2}\right) \qquad (21b) \\
&\qquad + \frac{1}{2}\text{Lip}(\nabla_\theta \ell(z_i, \theta))\left(\frac{2QC_f^2\tilde{g}_i}{n^2\mu^2} + \frac{M\tilde{g}_i^2}{n^2\mu^3} + \frac{2\tilde{g}_i\tilde{C}_f\bar{E}_n}{n\mu^2}\right)^2
\end{aligned}$$

where (21a) is by using (20a) together with the bound from Th. 1 and by replacing the Lipschitz constants C_{T_1} and C_{T_2} of the objective with the corresponding gradients from (20a) and (21b) is by using the Taylor

expansion of $\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}^{n\setminus i}))$ around $\hat{\theta}(\mathbf{1})$ and by using Lem. 2. Expanding this expression yields

$$
\begin{aligned}
|\mathrm{ACV} - \mathrm{CV}| \leq & \left(\frac{2QC_f^2}{\mu^2 n^2} + \frac{2\tilde{C}_f \bar{E}_n}{\mu^2 n}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^2 + \left(\frac{M}{\mu^3 n^2}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^3 \\
& + \left(\frac{8QC_f^2}{\mu^4 n^4} + \frac{8\tilde{C}_f \bar{E}_n}{\mu^4 n^3}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^3 \\
& + \left(\frac{4M}{\mu^5 n^4}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^4 \\
& + \left(\frac{2Q^2 C_f^4}{\mu^4 n^4} + \frac{2\tilde{C}_f^2 \bar{E}_n^2}{\mu^4 n^2}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^2 \\
& + \left(\frac{2QC_f^2 M}{n^4 \mu^5}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^3 \\
& + \left(\frac{M^2}{n^4 \mu^6}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^4 \\
& + \left(\frac{M\tilde{C}_f \bar{E}_n}{n^3 \mu^5}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^3 \\
& + \left(\frac{2Q\tilde{C}_f C_f^2 \bar{E}_n}{n^3 \mu^4}\right) \cdot \frac{1}{n}\sum_{i=1}^{n} \mathrm{Lip}(\nabla_\theta \ell(z_i, \theta)) \left\|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\right\|^2
\end{aligned}
$$

whose decay rate is dictated by the first two terms and is given by $O\left(\frac{C_f^2 B_{02}}{\mu^2 n^2} + \frac{\tilde{C}_f \bar{E}_n B_{02}}{\mu^2 n} + \frac{M B_{03}}{\mu^3 n^2}\right)$. □

I.2 Proof of Corol. 2

The proof follows similarly to that from (Suriyakumar and Wilson, 2022) by using the bound $\tilde{g}_i \leq G$ in (10) and then using the Gaussian mechanism for differential privacy (Dwork et al., 2014, App. A). □

We note that Corol. 2 parallels a similar result to that of Prop. 2, with different Lipschitz constants and with an additional term that depends on $\bar{E}_n$.

I.3 Proof of Corol. 3

The proof is by substituting $\tilde{g}_i = \|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\|$ in (11) and (12) and maximizing over i. □

We note that this proof parallels a similar result to that of Prop. 3, with two additional terms: one that depends on $\bar{E}_n$ and the other that depends on the Lipschitz coefficient of the features C_f.

I.4 Proof of Corol. 4

By using the definition of T from (2) and using the linearity of expectation and the triangle inequality we get that the Lipschitz coefficient of T from (2), C_{T_1}, is given by $2C_f$. Then, the proof follows by substituting $\tilde{g}_i = \|\nabla_\theta \ell(z_i, \hat{\theta}(\mathbf{1}))\|$ in (11) and maximizing over i. □

J Experimental Details

All experiments were implemented using the PyTorch (Paszke et al., 2019) framework and ran on NVIDIA A100 GPU.

J.1 Datasets

J.1.1 Adult

We utilized the Adult dataset (Dua and Graff, 2017) from

https://archive.ics.uci.edu/ml/machine-learning-databases/adult, to perform the task of predicting whether an individual's income is more than 50,000$ using 14 demographic features such as age, education, marital status, and country of origin. We aimed to keep sex as a sensitive attribute (the one that requires fairness, which in this case corresponds to the sex attribute) that requires fairness. The dataset contains 48,842 samples, divided into 32561 train samples and 16281 test samples. During pre-processing, we remove the sensitive attribute from the set of input features, discard rows with any missing data, convert textual values to categorical ones, and normalize the numerical data using the `StandardScaler()` function from the `sklearn.preprocessing` module. These pre-processing steps are consistent with those used in previous analyses of this dataset (e.g., (Shah et al., 2024)). We randomized the sample order by enabling the `Shuffle` option when creating the Dataloaders using `torch.utils.data.DataLoader`, ensuring the data was shuffled before being split into batches. The metric (2) was estimated on the training data.

J.1.2 Crime

We utilized the crime dataset (Redmond and Baveja, 2002) from https://archive.ics.uci.edu/dataset/183/communities+and+crime, which considers predicting the number of violent crimes per 100K population using socio-economic information of communities in the U.S. The sensitive attribute is the percentage of people belonging to a particular race in the community. During pre-processing, we drop all the samples with the value of the sensitive attribute less than 5% to remove any outliers. We also remove the non-predictive attributes and the sensitive attribute from the set of input features, and normalize all attributes to the standardized range of [0, 1]. The resulting data has 1,112 samples, and we use a train-test split ratio 0.8:0.2. We randomized the sample order by enabling the `Shuffle` option when creating the Dataloaders using `torch.utils.data.DataLoader`, ensuring the data was shuffled before being split into batches. The χ^2 metric from (3) was estimated on the training data.

J.1.3 Friedman–1

In our CV experiments we use the Friedman–1 regression benchmark Friedman (1991). We draw feature vectors with coordinates i.i.d. $\mathrm{Unif}(-1, 1)$ and generate responses as

$$y_n = 10 \sin\big(\pi\, x_n^{(1)} x_n^{(2)}\big) + 20\Big(x_n^{(3)} - \tfrac{1}{2}\Big)^2 + 10 x_n^{(4)} + 5 x_n^{(5)} + 0.1 \mathsf{z}_n, \qquad \mathsf{z}_n \sim \mathcal{N}(0, 1).$$

Only the first five coordinates are informative; the remaining $d - 5$ are nuisance. To study a higher-dimensional, sparse regime, we append 300 i.i.d. random features independent of $(\boldsymbol{x}, y)$ and train with ℓ_1 regularization. Unless stated otherwise, we use $N = 2000$ samples and $d = 100$ base features (before appending the 300 random features).

J.2 Models

All models were trained either using a cross-entropy loss or using an MSE loss, implemented via `torch.nn.CrossEntropyLoss()` and `torch.nn.MSELoss()`. We have conducted our experiments with two different regularizers: L_1 and L_2, namely, $\pi(\theta) = \|\theta\|^2$ or $\pi(\theta) = \|\theta\|_1^2$.

Two-Layer Network: We have tested our models with a two-layer fully-connected network. For the Adult dataset, we have used a softmax activation, whereas for the crime and the Friedman 1 datasets (where the task is regression), we didn't use any activation. The activation function for the hidden layer was chosen as `SeLU` activation. We used two variants of this model:

1. For the task from Sec. 5.1, the width of the hidden layer was chosen to be 1000. We trained the model for 100 epochs using the AdamW optimizer, with a learning rate of 10^{-4}, batches of size 100, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.999)$, and a weight-decay of 10^{-6}.

2. For the task from Sec. 5.2, the width of the hidden layer was chosen to be 100. We trained the model using the SGD optimizer, with a learning rate of 10^{-4}, batch size of 32 and a weight-decay of 10^{-4}. We also added an ℓ_1 regularization component of strength 10^{-2}. We varied the number of epochs from 1 to 10.

J.3 Details for Fairness and Unlearning

We trained the model on the training set of each dataset. Using the trained model, we estimated the fairness metric (either (2) or (3)) on the training data and measured the influence of each sample on this metric using the Taylor series-based approximation. We then selected all indices with a positive influence and unlearned them from the model by applying (9). The vector w^n used for generating $\tilde{\theta}^{\mathrm{IJ,AF}}$ had zeros at the selected indices and ones elsewhere. In all cases, we have measured influence using the `LiSSA` algorithm (see detailed description in Bae et al. (2022); Agarwal et al. (2017)). However, we have tested multiple different configurations of hyperparameters:

1. For the Adult's experiments, we have used 3 repetitions and a depth of 2000. Then, we have varied the scale parameter between 500 (first row in Fig. 1) and 250 (second row in Fig. 1).
2. For the Crime's experiments, we have used 3 repetitions and a depth of 2000. Then we have varied the scale parameter between 500 (first row in Fig. 1) and 125 (second row in Fig. 1).

J.4 Details for Cross-Validation

We have performed a leave-k-out CV to estimate the test loss. In particular, we have trained the previously mentioned two-layer model on the Friedman dataset from App. J.1.3, and set the width of the middle layer to be 100. In our experiments, we have fixed the L_1 regularization coefficient on 10^{-2}, the learning rate on 10^{-4}, the batch size on 32, and we have further trained the model with SGD with added weight decay of strength 10^{-4} (added in addition to the L_1 regularization). Our (full and approximate) CV estimates are over five different folds. For the CV approximation, we have calculated quantities of the form $u = \mathbf{F}^{-1}v$ (correspndingly for $u = \mathbf{H}^{-1}v$) by solving the linear system $\mathbf{F}u = v$ ($\mathbf{H}u = v$) and where we have used `PyTorch`'s autograd for calculating quantities of the form $\mathbf{F}u$ ($\mathbf{H}u$). The linear system was solved using `scipy.sparse.linalg.cg` running with maximum number of iterations set to 5000.

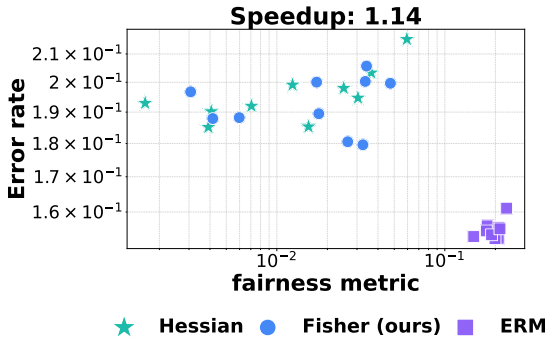
K Additional Experiments

K.1 Additional Experiments for Sec. 5.1

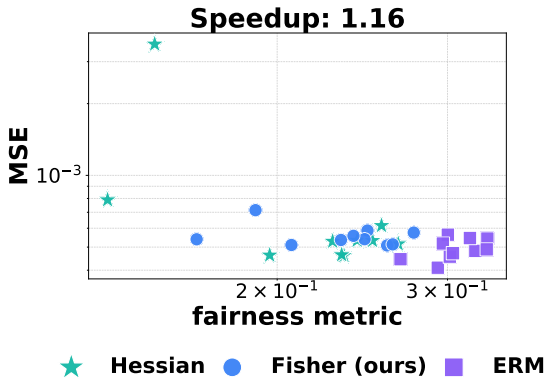
Here we provide additional experiments that use the FIM and the Hessian for measuring influence within the `LiSSA` framework of the fairness and unlearning experiments. In particular, we have changed the hyperparameters of the `LiSSA`-based influence estimator such that:

1. For Adult experiments, we set repetitions to 3, depth to 3000, and scales to 750 (first row) and 400 (second row).
2. For Crime experiments, we set repetitions to 5, depth to 5000, and scales to 1000 (first row) and 500 (second row).

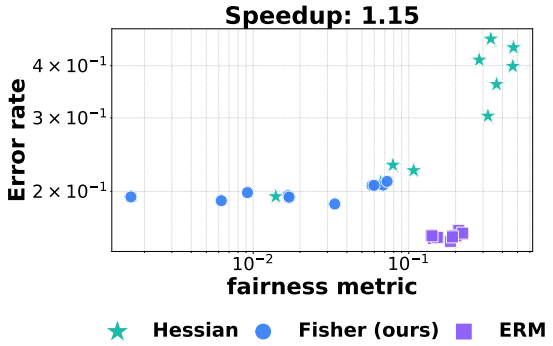
As is clear from the figures, a similar trend follows in these plots as the FIM is consistently faster, while for some configuration of hyperparameters, the estimates with the Hessian become unstable.



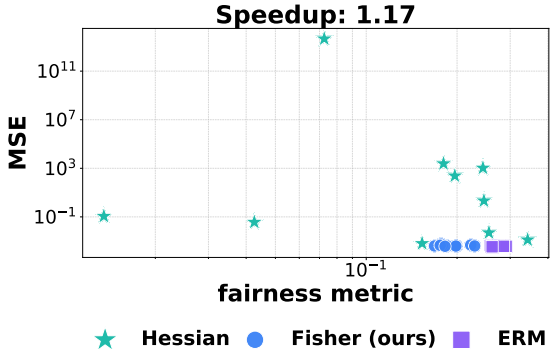
(a) Adult



(b) Crime



(c) Adult (alt. hyperparameters)



(d) Crime (alt. hyperparameters)

Figure 3: Additional experiment results with a different set of hyperparameters for the `LiSSA` algorithm. Top row: baseline settings; bottom row: a configuration where Hessian-based influence may be unreliable while FIM remains stable.