Measuring Stochastic Data Complexity with Boltzmann Influence Functions

Anonymous Authors[1]

Abstract

Estimating the uncertainty of a model's prediction is a crucial part of ensuring reliability under distribution shifts. A minimum description length approach to this problem considers every possible label for a data point, and decreases confidence in a prediction if other labels are also consistent with the model and training data. However, this normalized maximum likelihood (NML) distribution and its associated stochastic complexity measure require training an oracle learner for each label, which is misspecified and computationally intractable for deep neural networks. In this work we propose IF-COMP, a scalable and efficient method to approximate the NML distribution and complexity by linearizing the model with a temperature-scaled Boltzmann influence function. IF-COMP can then be used to produce well-calibrated predictions as well as measures of complexity in both labelled and unlabelled settings. We experimentally validate IF-COMP across uncertainty calibration, mislabel detection, and outlier detection tasks, where it consistently matches or beats strong baseline methods.

1. Introduction

Safely deploying machine learning models in real world settings requires making accurate predictions, as well as quantifying the uncertainty in those predictions. This is particularly important in high-stakes settings such as healthcare (Ghassemi & Mohamed, 2022), medical imaging (Esteva et al., 2017), and self-driving cars (Bojarski et al., 2016), where uncertainty estimates can help accurately assess the risk in utilizing a model's decision and decide when to ignore it altogether (Ovadia et al., 2019). Common methods for quantifying uncertainty rely on Bayesian principles, which require defining a prior and sampling from a posterior. However, specifying a "good" prior is difficult in many settings, and Bayesian methods scale poorly to the size of modern deep neural networks.

The Minimum Description Length Principle (MDL) (Rissanen, 1996) provides an alternative approach to uncertainty estimation which does not require explicitly defining a prior or even a notion of ground truth. MDL is a version of Occam's Razor, which states that we should favor models that minimize the codelength, measured as the negative log probability, of both the model and the observed data. When making a prediction on a test point, MDL considers all possible labels and tries to fit them with an oracle consistent with the training data. If many possible alternate labels can be fit well by the model, then we should decrease the confidence in our prediction accordingly.

This notion is formalized by the minimax predictive normalized maximum likelihood (pNML) (Roos & Rissanen, 2008) distribution that normalizes over the oracle best possible loss for each label. The codelength it defines for a data point is called the stochastic data complexity (Rissanen, 1986), and consists of two terms. The error term captures the *data uncertainty* by measuring the lowest loss an oracle model could achieve if the true label were observed. The parametric complexity term captures the *model uncertainty* by quantifying the number of different oracle models that could fit the data. Minimizing the pNML codelength then requires both learning an accurate model of the data as well as avoiding overconfidence on outlier observations.

Calculating the pNML distribution requires optimizing an oracle model over the training set *and the additional example with the given label*, which is both computationally intractable and misspecified for overparameterized neural networks that can easily fit random labels (Zhang et al., 2017). A simple way to restrict the oracle model subclass is with a proximal oracle objective that penalizes movement in function and weight space. Approximating this natural oracle objective by linearizing the network and applying a second order approximation of the proximal terms has been shown to correspond to the influence function (IF) in neural networks (Bae et al., 2022).

However, when models are overconfident on training examples, the proximal terms can become too restrictive, making the oracle unable to fit low probability labels. In this work

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

we propose to temperature-scale the proximal objective and approximate it with a Boltzmann influence function (BIF), allowing the oracle to better accommodate arbitrary labels. We then propose **IF-COMP**, which uses the BIF to produce estimates of the oracle output probabilities, which can be used to calibrate output distributions and measure the stochastic complexity of both labelled and unlabelled examples.

We validate IF-COMP's ability to reliably estimate ground truth oracle complexity, then investigate its use on three tasks that test its different capabilities. We consider (1) uncertainty calibration, which requires producing reliable uncertainty estimates under distribution shifts, (2) mislabel detection, which requires estimating complexity for labelled training examples, and (3) outlier detection, which requires estimating complexity for unlabelled test examples. Across all three tasks, IF-COMP displays strong performance, consistently matching or beating Bayesian and optimization tracing approaches that often use *more* information than is available to our method, as well as a similar pNML approximation method that explicitly takes steps in parameter space (Zhou & Levine, 2021). Compared to this pNML baseline, IF-COMP also provides a 7-15 times speedup in computational efficiency. IF-COMP demonstrates the potential of MDL-based approaches for uncertainty and complexity estimation in deep neural networks, enabling better calibrated decision making.

2. Background and Preliminaries

2.1. Minimum Description Length and Stochastic Complexity

In this work we consider a supervised classification task from an input space $\mathcal{X}$ to a discrete output space $\mathcal{Y}$, where we are given a finite training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_n$ consisting of pairs of examples $(\mathbf{x}_i, \mathbf{y}_i)$. We define a hypothesis set of possible models $\{\boldsymbol{\theta} \in \Theta\}$, each of which defines a conditional probability distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = \sigma(f_{\boldsymbol{\theta}}(\mathbf{x}))$ where σ is the softmax function.

The Minimum Description Length (MDL) Principle (Rissanen, 1978; Grunwald, 2004) states that we should choose the model in our hypothesis class which best encodes the data (for simplicity we assume the codelength of the hypothesis itself is constant). We can use the Kraft inequality (Cover & Thomas, 1991) to draw an equivalence between codelengths and probability distributions, where

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) = -\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \tag{1}$$

is the codelength or log loss for $(\mathbf{x}, \mathbf{y})$. A learner q then aims to find the model $\hat{\boldsymbol{\theta}}$ in the hypothesis set that minimizes the total codelength on $\mathcal{D}_{\text{train}}$:

$$\mathcal{J}(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}) = \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{y}_i) \tag{2}$$

We now consider making a prediction at a test point $\mathbf{z} = (\mathbf{x}, \mathbf{y})$. An oracle learner that has access to the test point and training set , but does not know which point is the test point, will aim to find the parameters that minimize the total codelength:

$$\boldsymbol{\theta}^\star(\mathbf{x}, \mathbf{y}, \epsilon) = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{J}(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}) - \epsilon \log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) \tag{3}$$

where ϵ represents the amount to weight the test point relative to the training set. The standard oracle will always set $\epsilon = 1/n$ so that the test point is weighted equally to the other training points, but we include this additional parameter for reasons that will become clear later. For ease of notation, we define the oracle parameters as $\boldsymbol{\theta}^\star(\mathbf{x}, \mathbf{y}) \coloneqq \boldsymbol{\theta}^\star(\mathbf{x}, \mathbf{y}, 1/n)$.

The regret of the learner q for a test point $\mathbf{z}$ is the difference between its codelength and the oracle codelength:

$$R(q, \mathcal{D}_{\text{train}}, (\mathbf{x}, \mathbf{y})) = \\ -\log p_{\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x}) + \log p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x}). \tag{4}$$

The predictive normalized maximum likelihood (pNML) distribution (Shtar'kov, 1987; Roos et al., 2008; Roos & Rissanen, 2008) is the universal model which minimizes the worst case regret across all possible labels:

$$p_{\text{pNML}}(\mathbf{y}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{y}' \in \mathcal{Y}} p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y}')}(\mathbf{y}'|\mathbf{x})}. \tag{5}$$

The codelength of this distribution

$$\Gamma(\mathbf{z}) = \overbrace{-\log p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x})}^{\text{error}} + \\ \underbrace{\log \sum_{\mathbf{y}' \in \mathcal{Y}} p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y}')}(\mathbf{y}'|\mathbf{x})}_{\text{parametric complexity}} \tag{6}$$

is also called the *stochastic complexity* (Rissanen, 1996; Barron et al., 1998) of $\mathbf{z}$ relative to the model class Θ. The first error term measures the *data uncertainty* as the best possible oracle code or log loss for $\mathbf{z}$. The second parametric complexity term measures the *model uncertainty* and quantifies how many distinguishable distributions the model class Θ could assign to the data. A model that overfits will be able to achieve low error but will also have high parametric complexity. A model that underfits will have low parametric complexity but will incur high error. Choosing a model with a small description length, or stochastic complexity, necessitates trading off between these two extremes.

2.2. Boltzmannian NML

Typically, the pNML distribution is normalized simply by summing over all oracle probabilities. An alternative parameterization views the pNML distribution as a Boltzmann distribution Perotti et al. (2018) with an energy function

$$E_{\beta,\boldsymbol{\theta}^\star}(\mathbf{x},\mathbf{y}) = -\beta \log p_{\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x}), \tag{7}$$

where β is an inverse temperature parameter. The corresponding Boltzmannian normalized maximum likelihood (BNML) distribution can then be calculated as:

$$p_{\text{BNML}}(\mathbf{y}|\mathbf{x}) = \frac{e^{-E_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{x},\mathbf{y})}}{\sum\limits_{\mathbf{y}'\in\mathcal{Y}} e^{-E_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{x},\mathbf{y})}}, \tag{8}$$

which has an associated stochastic complexity

$$\Gamma_\beta(\mathbf{z}) = E_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{x},\mathbf{y}) + \log \sum_{\mathbf{y}'\in\mathcal{Y}} e^{-E_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{x},\mathbf{y})}. \tag{9}$$

When $\beta = 1$ we recover the original pNML distribution and complexity. Recent work has shown the value of temperature scaling for improving uncertainty calibration (Guo et al., 2017), so we propose to measure stochastic complexity in this paper by using this alternate Boltzmann formulation.

2.3. The Infinity Problem and the Natural Oracle

For many overparameterized model classes, the normalization constant in (5) is either infinite (for continuous label spaces) or constant (for discrete label spaces). To solve this *infinity problem*, the oracle must be restricted to a subclass of models that "comply" with the original training data (Fogel & Feder, 2019). This can be done by restricting the label space (Stine & Foster, 2001), enforcing moment matching (Farnia & Tse, 2017), confidence intervals, or minimum training likelihood (Fogel & Feder, 2019).

In this work we propose to solve the infinity problem with an oracle proximal bregman objective (PBO) that restricts movement in function and weight space away from the optimum found by the base learner:

$$\mathcal{Q}(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}, \mathbf{z}) = -\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) + \sum_{i=1}^{n} D_{KL}(p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i) \parallel p_{\hat{\boldsymbol{\theta}}}(\mathbf{y}_i|\mathbf{x}_i)) + \frac{\lambda}{2}||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}||_2^2. \tag{10}$$

Since the preconditioner for the corresponding update rule is the Fisher information, we call this the **natural oracle**. This objective is motivated by a similar restriction used to solve the infinity problem for overparameterized ridge regression models (Dwivedi et al., 2023).

2.4. Influence Functions

Influence functions (Cook, 1979; Hampel, 1974) are a classical method from robust statistics that attempt to measure the sensitivity of an estimator to indiviual datapoints (Fisher et al., 2023). We can formulate this alternative objective using Eq. 3 which we rewrite as a response function $r_{(\mathbf{x},\mathbf{y})} : \mathbb{R} \to \Theta$

$$r_{(\mathbf{x},\mathbf{y})}(\epsilon) = \boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y},\epsilon)$$

where we assume that the objective (3) is strongly convex and hence the optimum is unique given some factor ϵ. Under these assumptions, note that $r_{(\mathbf{x},\mathbf{y})}(0) = \hat{\boldsymbol{\theta}}$ and the response function is differentiable at $\epsilon_0 = 0$ by the Implicit Function Theorem (Griewank & Walther, 2008; Krantz & Parks, 2002). This allows us to approximate the response function with a first order Taylor expansion about ϵ_0:

$$r_{(\mathbf{x},\mathbf{y})}(\epsilon) = \hat{\boldsymbol{\theta}} + \mathbf{G}^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x},\mathbf{y})\epsilon,$$

where $\mathbf{G}$ is the Hessian of (2) or a positive definite approximation such as the Generalized Gauss-Newton Hessian or Fisher Information.

To approximate the effects on the loss at a specific test point $\mathbf{z}' = (\mathbf{x}', \mathbf{y}')$, we can linearize the model about $\hat{\boldsymbol{\theta}}$ and apply the chain rule:

$$\mathcal{L}(\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y},\epsilon),\mathbf{x}',\mathbf{y}') = \mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x}',\mathbf{y}') + \epsilon\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x}',\mathbf{y}')^\intercal\mathbf{G}^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x},\mathbf{y})$$

If the train and test point are the same, the quantity

$$\text{IF}\,(\boldsymbol{\theta},\mathbf{x},\mathbf{y}) = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x},\mathbf{y})^\intercal\mathbf{G}^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}},\mathbf{x},\mathbf{y}) \tag{11}$$

is often referred to as the *self-influence* of $\mathbf{z}$ relative to the model parameterized by $\boldsymbol{\theta}$ (Koh & Liang, 2017).

Although influence functions were meant to approximate the effects of true retraining (Basu et al., 2020), recent work has shown that they more closely approximate an alternate proximal bregman objective (PBO) (Bae et al., 2022; Grosse et al., 2023), which coincides with the natural oracle objective (10) proposed in Section 2.3.

3. IF-COMP: Measuring Complexity with Boltzmann Influence Functions

Exactly calculating the BNML stochastic complexity for a data point is intractable because it requires optimizing the oracle for each possible label. In addition, overparameterized neural networks can achieve arbitrarily low log loss on the additional test point regardless of its label, introducing the infinity problem. We empirically verify this behavior in Appendix A.1. Motivated by these observations, we propose IF-COMP, a method for approximating the BNML

distribution and stochastic data complexity for deep neural networks.

We begin by defining a Boltzmann oracle objective and influence function that is formulated to soften the local curvature to better approximate oracle predictions for low probability labels. We then show how to use this influence function to produce a calibrated output distribution as well as approximate the stochastic complexity for labelled and unlabelled data points. Finally, we describe how to efficiently compute IF-COMP and then validate it against the ground truth natural oracle complexity.

3.1. Boltzmann Influence Functions

Since the natural oracle PBO (Eq. 10) regularizes movement in function space using the model's own output distribution, overconfident predictions can make this objective *too* restrictive for training the oracle with very low probability labels. Temperature scaling has been shown to reduce model overconfidence and improve calibration for predictions (Guo et al., 2017), which we can apply by scaling the pNML distribution, as in Eq. 8. Rather than scaling our oracle outputs after approximation, we propose to absorb the temperature parameter directly into the oracle, by modifying our loss with a temperature-scaled softmax:

$$E'_{\beta,\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) = -\log p_{\beta,\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) := -\log \sigma(\beta f_{\boldsymbol{\theta}}(\mathbf{x})). \quad (12)$$

We can replace the loss in the original PBO with our temperature-scaled loss to define a corresponding oracle with a Boltzmann PBO (BPBO):

$$\mathcal{Q}(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}, \mathbf{z}) = E'_{\beta,\boldsymbol{\theta}}(\mathbf{x},\mathbf{y}) + \sum_{i=1}^{n} D_{KL}(p_{\beta,\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i) \parallel p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}_i|\mathbf{x}_i)) + \frac{\lambda}{2}||\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}||_2^2. \quad (13)$$

If we take a first order approximation of the loss and a second order approximation of the proximal terms about $\hat{\boldsymbol{\theta}}$, we can formulate the influence function as

$$\text{IF}_\beta(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}) = \nabla_{\boldsymbol{\theta}} E'_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{y})^\intercal \mathbf{G}_\beta^{-1} \nabla_{\boldsymbol{\theta}} E'_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{y}), \quad (14)$$

with a corresponding Fisher information:

$$\mathbf{G}_\beta = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{y}'\sim p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})} \left[\nabla_{\boldsymbol{\theta}} E'_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{y}') \nabla_{\boldsymbol{\theta}} E'_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{x},\mathbf{y}')^\intercal\right].$$

Intuitively, temperature scaling softens the function space distance loss and allows the model to better accommodate arbitrarily labelled data points. Since our approach is motivated by the BNML energy formulation, we call the influence function calculated with this temperature-scaled loss the **Boltzmann influence function (BIF)**.

Different choices for β define different BIFs. At $\beta = 1$, we recover the standard influence function. As $\beta \to 0^+$, the KL divergence loss becomes an MSE loss over logits (Kim et al., 2021) and we recover a BIF resembling TRAK (Park et al., 2023). In Section 4.3 we will see that tuning β directly within the influence function is an important part of achieving high quality BNML complexity estimates.

3.2. IF-COMP

We now describe **IF-COMP**, our method for estimating BNML distributions and stochastic complexity using our Boltzmann influence function. Recall from Eq. 9 that our goal is to approximate

$$e^{-E'_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{x},\mathbf{y})} = p_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x}),$$

for all $\mathbf{y} \in \mathcal{Y}$, where we have replaced the energy with the one defined in Eq. 12. Using the BIF formulation above, we can apply a first order Taylor expansion and the chain rule to approximate

$$p_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x}) = p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x}) + \frac{1}{n} p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})\ \text{IF}_\beta(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}).$$

where we have used the identity

$$\nabla p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x}) = p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x}) \nabla \log p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})$$

to change the IF log probability gradients into probability gradients. We can then rewrite our BNML parametric complexity as

$$\log \sum_{\mathbf{y}'\in\mathcal{Y}} p_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y}')}(\mathbf{y}'|\mathbf{x}) = \log\left(1 + \frac{1}{n}\mathbb{E}_{\mathbf{y}'\sim p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})}\left[\text{IF}_\beta(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}')\right]\right).$$

For large enough n we can simplify $\log(1+x) \approx x$, which gives us a final complexity

$$\Gamma_\beta(\mathbf{z}) = \overbrace{-\log p_{\beta,\boldsymbol{\theta}^\star(\mathbf{x},\mathbf{y})}(\mathbf{y}|\mathbf{x})}^{\text{error}} + \underbrace{\frac{1}{n}\mathbb{E}_{\mathbf{y}'\sim p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})}\left[\text{IF}_\beta(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}')\right]}_{\text{parametric complexity}}. \quad (15)$$

We leave the error term in its original oracle form since we assume access to labels only for training data, for which our model is already an oracle. For unlabelled data that may not have a valid label, the error term is undefined and we compute only the parametric complexity term.

$$\Gamma_\beta(\mathbf{x}) = \mathbb{E}_{\mathbf{y}'\sim p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})}\left[\text{IF}_\beta(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}')\right] \quad (16)$$

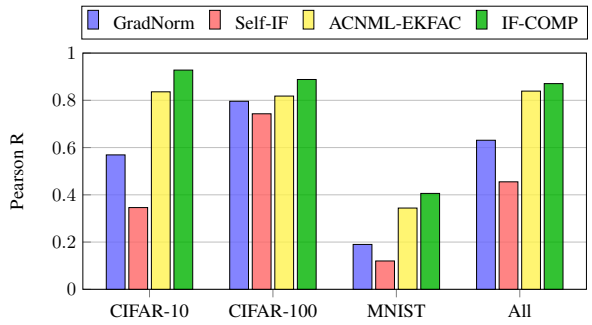


Figure 1. Pearson R correlation of different methods of approximating oracle outputs with ground truth parametric complexity on in-domain (CIFAR-10) and out-of-domain datasets. IF-COMP achieves the highest correlation across all datasets, beating ACNML, a computationally more expensive alternative.

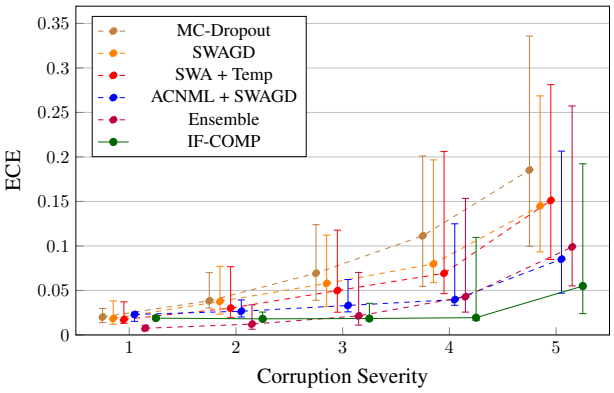


Figure 2. Expected calibration error (ECE) for various methods across increasing levels of CIFAR-10C corruptions. We plot medians and inter-quartile ranges. IF-COMP achieves lower ECE across almost all corruption levels compared to both Bayesian methods and other NML-based methods.

If we are interested in the calibrated probabilities of the BNML distribution, we can calculate the distribution directly as

$$p_{\text{BNML}}(\mathbf{y}|\mathbf{x}) = \frac{p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x}) + {}^{\alpha}/_{n}\, p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})\, \text{IF}_{\beta}(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y})}{1 + {}^{\alpha}/_{n}\, \mathbb{E}_{\mathbf{y}' \sim p_{\beta,\hat{\boldsymbol{\theta}}}(\mathbf{y}|\mathbf{x})} \left[\text{IF}_{\beta}(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}')\right]} \tag{17}$$

where α controls the weighting of the test point relative to the training set. Compared to ACNML (Zhou & Levine, 2021), which explicitly optimizes the oracle parameters using a weighted approximate posterior, IF-COMP linearizes the model directly which allows us to easily control this weighting *after* computing the BIF for each label.

3.3. Efficiently Computing IF-COMP

Although we have simplified the calculation of stochastic data complexity considerably, calculating the BIF still requires estimating and inverting the Fisher information matrix. Similar to previous work (Grosse et al., 2023), we use an eigenvalue-corrected Kronecker-factored approximation to the Fisher information matrix (EKFAC) (George et al., 2018) which produces an eigendecomposition $\mathbf{G}_{\beta} \approx Q\lambda Q^{T}$. We can then calculate the influence function as the squared L2 norm in the inverse eigenspace:

$$\text{IF}_{\beta}(\hat{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{y}) = ||\nabla_{\boldsymbol{\theta}} E_{\beta,\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y})^{\intercal} Q(\lambda + \delta)^{-1/2}||_2^2, \tag{18}$$

where δ is a damping term added to ensure invertibility. Since we need only calculate and invert the EKFAC once for a given model and training set, calculating the IF-COMP then requires only one jacobian vector product (JVP) per label per sample.

3.4. Oracle Validation

To verify that IF-COMP can accurately approximate the ground truth parametric complexity on both in-distribution (ID) and out-of-distribution (OOD) samples, we fine-tune a CIFAR-10 (Krizhevsky, 2009) pre-trained ResNet-18 (He et al., 2016) model with the BPBO (13) on 20 random test images each from CIFAR-10, CIFAR-100, and MNIST (Deng, 2012). We label each example with every possible output class for a total of 600 individual training example pairs. After training, we can use the oracle output probabilities to calculate the complexity as in Eq. 16.

We compare IF-COMP with other baselines for approximating oracle training. Specifically, we consider averaging gradient norms across classes, self influence calculated with EKFAC, and ACNML with an EKFAC posterior. We present our results in Figure 1. We find that IF-COMP consistently achieves the highest Pearson correlation with true complexity on all datasets, beating ACNML which explicitly takes steps in parameter space. For ID CIFAR-10 examples, IF-COMP achieves a strong Pearson R of 0.928, which it maintains on CIFAR-100 examples. All methods degrade considerably on MNIST, which indicates that complexity becomes more difficult to estimate for further OOD examples.

4. Experiments

To experimentally validate IF-COMP, we consider three tasks that test different capabilities. The first, uncertainty quantification, requires producing calibrated outputs across distribution shifts. The second, mislabel detection, requires measuring complexity on labelled training examples. We use this task as a case study to understand how the error and parametric complexity terms tradeoff during training, as well as the effects of temperature scaling. Finally, outlier detection requires measuring complexity on unlabelled out-of-distribution test examples. Across all three tasks, IF-

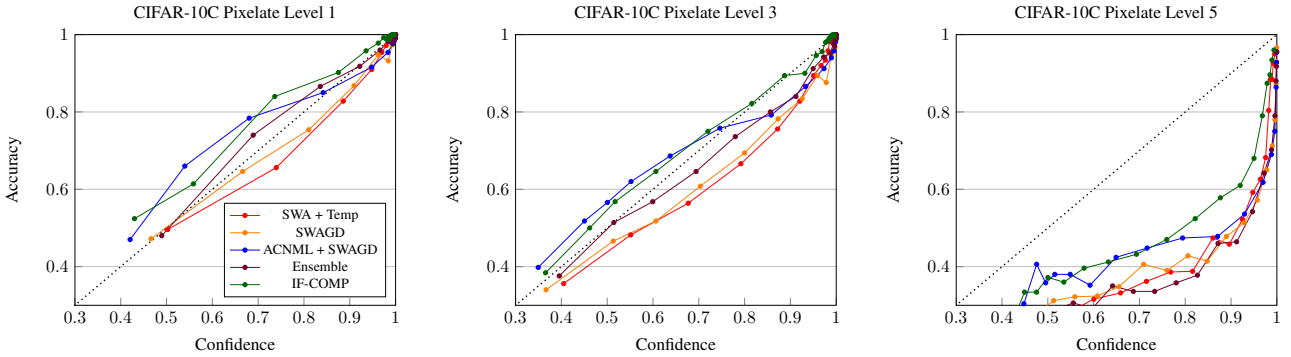


Figure 3. Reliability diagrams for Pixelate corruptions on CIFAR-10C. IF-COMP outperform ACNML as well as Bayesian methods and ensembles even as corruptions increase in severity. Although IF-COMP and ACNML perform similarly on lower confidence examples, IF-COMP maintains this reliability on higher confidence examples. Dotted lines represent perfect calibration.

COMP consistently exhibits strong performance compared to baseline methods. We provide additional experiments on data pruning and unrestricted oracle retraining in Appendix A as well as full experimental details in Appendix B.

4.1. Uncertainty Calibration

We begin by evaluating the uncertainty calibration of IF-COMP output distributions under distribution shifts. Following the experimental setup in Ovadia et al. (2019), we measure the expected calibration error (ECE) (Pakdaman Naeini et al., 2015) of ResNet18 models trained on CIFAR-10 and tested on the CIFAR10-C datasets (Hendrycks & Dietterich, 2019). CIFAR-10C applies 19 corruptions with 5 severity levels across the test images of CIFAR-10, allowing us to compare calibration on a wide range of distribution shifts. We calculate ECE by dividing model predictions sorted by confidence into 20 equal sized bins (Zhou & Levine, 2021).

We compare against a wide range of Bayesian and NML baselines, including ensembling (Lakshminarayanan et al., 2017), Stochastic Weight Averaging (SWA) (Izmailov et al., 2019), SWA Gaussian Diagonal (SWAG-D) (Maddox et al., 2019), Monte-Carlo dropout (Gal & Ghahramani, 2016), and ACNML (Zhou & Levine, 2021) with a SWAG-D posterior. For a fair comparison, all methods except ensembling and MC-Dropout use the same SWA base model. Additionally, we apply the same IF-COMP temperature scaling to the SWA output logits, which is equivalent to using a value of $\alpha = 0$. We produce 30 model samples for all Bayesian methods.

We present median ECE and inter-quartile ranges across corruptions for each severity level in Figure 2. On more severe corruptions (3-5), IF-COMP produces better calibrated uncertainties than all baselines, beating the most relevant ACNML baseline as well as a Bayesian ensemble of 30 trained models. At corruption severity 2 IF-COMP outperforms all other baselines except the ensemble. At the lowest corruption level all methods perform similarly.

The reliability diagrams in Figure 3 show a more detailed look at the calibration for varying Pixelate corruption severity levels. For low corruption levels all methods perform similarly. As severity increases, IF-COMP maintains strong calibration compared to Bayesian baselines and improves over ACNML for high confidence outputs. Although all methods degrade considerably at the highest corruption severity, IF-COMP still improves over the strong ACNML baseline.

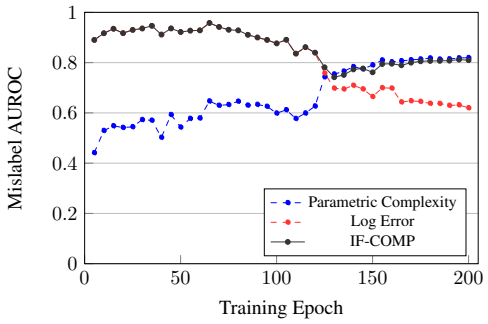
4.2. Mislabel Detection

Next, we investigate IF-COMP's ability to measure the complexity of labelled training examples. We follow the mislabel detection setup in Zhaowei Zhu (2021); Srikanth et al. (2023) and use CIFAR-10 and CIFAR-100 datasets corrupted with various types of label noise and noise rates. We train a ResNet-18 model on this corrupted dataset then attempt to identify which examples were mislabelled using only the trained model and noised dataset. We consider 4 types of label noise. Human noise replaces all labels with labels from a single human annotator (Zhaowei Zhu, 2021; Wei et al., 2022). Data-dependent noise is generated by jointly modelling the clean label and a class-dependent projection of the feature vector (Zhaowei Zhu, 2021). Symmetric noise changes the label to another uniformly at random. Asymmetric noise changes each label to another fixed similar label.

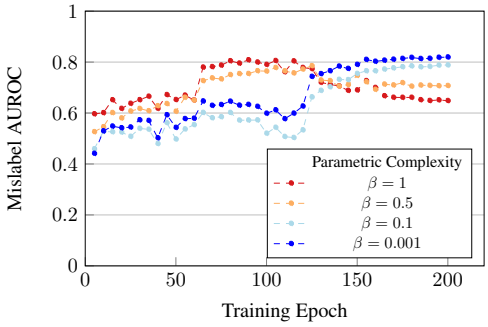
For each example, we calculate the full stochastic complexity (15) by combining the model's log error on the true label as well as the parametric complexity across all labels. For our baselines, we consider the self-influence score (Self-IF) (Koh & Liang, 2017), Trac-IN (Pruthi et al., 2020), EL2N (Paul et al., 2021), and GraNd (Paul et al., 2021) and measure performance using AUROC. Trac-IN requires evaluating intermediate checkpoints, and EL2N and GraNd

Table 1. Mislabel detection AUROC for CIFAR-10 and CIFAR-100 with various types and rates of label noise (best method bolded). IF-COMP achieves strong detection across all CIFAR-10 noise types without requiring extra checkpoints, even on the difficult data-dependent noise that other methods fail to perform better than random on. On CIFAR-100, IF-COMP achieves strong performance on symmetric and asymmetric noise, although it fails to detect mislabelled data with human and data-dependent noise, similar to other methods.

Method	Extra Checkpoints?	CIFAR10				CIFAR100			
		Human	Data	Asym 0.3	Sym 0.6	Human	Data	Asym 0.3	Sym 0.6
Trac-IN	✓	90.50	49.13	70.98	64.40	50.01	50.25	58.25	57.95
EL2N	✓	73.38	52.18	51.53	51.61	49.79	**50.51**	56.30	55.45
GraNd	✓	70.53	51.30	50.48	50.52	**50.52**	50.07	52.26	50.16
Self-IF	✗	95.38	55.77	87.30	43.16	49.81	50.29	72.15	58.40
IF-COMP	✗	**96.86**	**88.07**	**95.69**	**97.83**	49.52	50.30	**79.39**	**95.21**



(a) AUROC of the individual components of IF-COMP.



(b) Parametric complexity AUROC at varying temperatures.

Figure 4. IF-COMP accurately trades off between log error and parametric complexity, maintaining strong AUROC throughout training. Tuning the temperature is critical to achieving accurate complexity estimates near convergence.

require multiple short training runs, meaning they utilize *more* information than is available to IF-COMP.

We present our results in Table 1. For CIFAR-10 datasets, IF-COMP consistently achieves the highest AUROC across all types of noise, beating even the baselines that utilize additional checkpoint information. On the difficult data-dependent noise where all other methods achieve close to random AUROC, IF-COMP maintains strong performance. For CIFAR-100 datasets where all baselines consistently achieve close to random AUROC across all noise types, IF-COMP is still able to detect both asymmetric and symmetric noise. On human and data-dependent noise no method performs better than random.

4.3. Analyzing the Components of IF-COMP

We take a brief detour and use mislabel detection as a case study to understand how each component of IF-COMP contributes to the final complexity estimate. We consider CIFAR-10 human mislabel noise and train a model to convergence, taking checkpoints every 5 epochs. For each model, we calculate IF-COMP, log error, and temperature-scaled parametric complexity. We plot the mislabel detection AUROC across training for these values in Figure 4.

Figure 4a shows that mislabelled examples are ignored early in training, increasing their error and making them easy to distinguish. Later in training the model learns to memorize these examples and decrease their error, increasing parametric complexity accordingly. Choosing only one of these values to detect mislabelled examples with would fail in different scenarios, while IF-COMP accurately captures the tradeoff between the two throughout all stages of training. Figure 4b shows that temperature-scaling is a crucial component of IF-COMP. With a standard temperature $\beta = 1$, parametric complexity estimates are inaccurate and mirror log error, providing no additional signal for mislabel detection. As we increas the temperature we soften the second order restrictions, making complexity estimates more accurate and providing a complementary signal to the error.

4.4. Outlier Detection

Finally, we investigate IF-COMP's ability to measure complexity on unlabelled data by detecting outliers. Although this task is often referred to as out-of-distribution (OOD) detection (Yang et al., 2021), from an MDL perspective there is no "in-distribution" (ID) and "out-of-distribution" (OOD), only data with shorter and longer codelengths.

Table 2. AUROC for outlier detection methods on near and far distribution shifts. Best methods are bolded, and second best methods are starred. We show only the most common baseline methods, although we compare against all methods in the OpenOOD benchmark. IF-COMP achieves a new state of the art on MNIST and CIFAR-10 benchmarks, beating out all 20 other baselines.

	MNIST		CIFAR-10		CIFAR-100	
Method	Near	Far	Near	Far	Near	Far
MSP	91.45	98.51	88.03	90.73	80.27	77.76
ODIN	92.38	99.02	82.87	87.96	79.90	79.28
Energy	90.77	98.77	87.58	91.21	80.91*	79.77
MLS	92.49	99.08	87.52	91.10	**81.05**	79.67
KNN	96.52	96.66	90.64*	92.96	80.18	82.40*
GradNorm	76.55	96.39	54.90	57.55	70.13	69.14
RMDS	98.00*	98.12	89.80	92.20	80.15	**82.92**
VIM	94.63	98.99	88.68	93.48*	74.98	81.70
Gram	73.90	99.75*	58.66	71.73	51.66	73.36
IF-COMP	**99.40**	**99.97**	**92.23**	**95.63**	79.40	79.41

We use the OpenOOD benchmark (Yang et al., 2022; Zhang et al., 2023) and consider the ID MNIST, CIFAR-10, and CIFAR-100 datasets using the provided pretrained LeNet (Lecun et al., 1998) and ResNet-18 models. For each dataset, the benchmark provides a set of Near-OOD datasets which exhibit similar visual features to the ID training set, and a set of Far-OOD datasets that do not. Given a test example from one of these datasets, an outlier detection method aims to assign a high score to OOD examples and a low score to ID examples. We use the IF-COMP parametric complexity (16) as our score function, and measure performance using AUROC. We compare our method against the full suite of 20 baseline approaches implemented in OpenOOD, although we show only 9 of the most common or most similar methods in our results. We refer readers to Zhang et al. (2023) for more details on baseline methods.

Our results are shown in Table 2. On MNIST, IF-COMP achieves a new state of the art AUROC, beating all 20 baselines with near perfect detection on both Near and Far OOD datasets. IF-COMP similarly achieves a new state of the art AUROC for CIFAR-10, beating the next best method by over 2 AUROC on Far OOD datasets. On CIFAR-100 IF-COMP ranks in the middle of the baselines. For this dataset all baselines perform quite similarly, with only small AUROC gaps between the best and second best method. Strong baselines exhibit variance in performance across datasets, with the best method, RMDS (Ren et al., 2021), beating all other baselines only on 2 datasets. In contrast, IF-COMP achieves the best AUROC across 4 of the 6 datasets.

5. Related Work

The Minimum Description Length Principle MDL is one of many ways to measure complexity, including Kolmogorov complexity(Kolmogorov, 1965; Solomonoff, 1964), VC dimension (Vapnik & Chervonenkis, 1971) and Rademacher complexity (Bartlett & Mendelson, 2003). The Refined MDL coding scheme (Barron et al., 1998) which defines stochastic complexity (Rissanen, 1986) is based on the normalized maximum likliehood (NML) distribution (Shtar'kov, 1987), which has been modified for predictive settings (pNML) (Roos et al., 2008; Fogel & Feder, 2019). Refined MDL bears similarity to model selection criteria such as Akaike Information Criterion (AIC) (Akaike, 1973) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). We point readers to Grunwald (2004) for further discussion of MDL.

MDL principles have been used to motivate methods to train autoencoders (Hinton & Zemel, 1993) and regularize weights (Hinton & van Camp, 1993). Recent work in MDL methods for neural networks have mainly attempted to approximate pNML distributions by performing additional gradient steps (Bibas et al., 2020) or optimizing with an approximate posterior (Zhou & Levine, 2021). Instead of explicitly calculating model weights, IF-COMP directly linearizes the model making optimization unnecessary and allowing post-hoc tuning of example weighting.

Quantifying Predictive Uncertainty A wide variety of methods have been proposed to quantify predictive uncertainty in deep neural networks. Bayesian approaches include Laplace approximations (Mackay, 1992; Ritter et al., 2018), variational inference (Graves, 2011), stochastic gradient MCMC (Welling & Teh, 2011), dropout (Gal & Ghahramani, 2016), and weight averaging (Izmailov et al., 2019; Maddox et al., 2019). Non-Bayesian methods include model ensembling (Lakshminarayanan et al., 2017; Osband et al., 2016), Platt scaling (Platt, 2000) with logit temperatures, (Guo et al., 2017) and more recently, MDL based approaches.

Bayesian and MDL methods have a deep connection. Refined MDL model selection coincides with Bayes factor model selection (Kass & Raftery, 1995) based on a Jeffreys' prior (Bernardo & Smith, 1994), where codes correspond to MAP estimates. NML distributions over the full dataset can also be seen as a form of Bayesian marginal likelihood (Grunwald, 2004), although pNML does not coincide with any standard Bayesian interpretations.

Influence Functions Influence functions (Cook, 1979; Hampel, 1974) are a classical method from robust statistics that has found renewed interest in deep neural networks (Koh & Liang, 2017; Grosse et al., 2023). They have been applied for a wide range of tasks including detecting bias (Brunet et al., 2019), data poisoning (Koh et al., 2021), auditing predictions (Schulam & Saria, 2019), and fixing model mistakes (Tanno et al., 2022). The self-influence of a data point is related to its memorization (Feldman, 2021),

which has been shown is necessary to cover the long tails of the data distribution and improve generalization (Feldman & Zhang, 2020). Although influence functions have been found to poorly match true retraining for neural networks (Basu et al., 2020), they correlate well with an alternate proximal objective (10) (Bae et al., 2022).

6. Conclusion

In this paper we have proposed IF-COMP, an efficient method for estimating stochastic data complexity in deep neural networks with temperature-scaled Boltzmann influence functions (BIFs). The BIF softens the second order curvature and linearizes the model, allowing us to efficiently approximate oracle outputs even for low probability labels. Using these oracle outputs, IF-COMP can produce well calibrated output distributions as well as measure complexity on both labelled and unlabelled points. On tasks covering uncertainty calibration, mislabel detection, and outlier detection, IF-COMP consistently matches or outperforms strong baselines, including Bayesian and optimization tracing based approaches. Our results demonstrates the potential of MDL based approaches for improving uncertainty estimates in deep neural networks.

7. Broader Impact

We have presented a method that allows a practitioner with access to a model and training dataset to identify examples with high *complexity*, or alternatively that contain a large amount of information relative to a model. Although we have presented experiments that aim to validate its use in producing better calibrated uncertainties and out-of-distribution predictions which should reduce societal consequences, IF-COMP can also be used to automatically find important training examples. If these examples correspond to real people, a malicious agent could use this information to influence or bias downstream models or extract sensitive information.

References

Akaike, H. *Information Theory and an Extension of the Maximum Likelihood Principle*, pp. 199–213. Springer New York, New York, NY, 1973.

Bae, J., Ng, N., Lo, A., Ghassemi, M., and Grosse, R. B. If influence functions are the answer, then what is the question? In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17953–17967. Curran Associates, Inc., 2022.

Barron, A., Rissanen, J., and Yu, B. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998. doi: 10.1109/18.720554.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3(null):463–482, mar 2003. ISSN 1532-4435.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

Bernardo, J. M. and Smith, A. F. M. *Bayesian Theory*. Wiley, 1994.

Bibas, K., Fogel, Y., and Feder, M. Deep pnml: Predictive normalized maximum likelihood for deep neural networks, 2020.

Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars, 2016.

Brunet, M.-E., Alkalay-Houlihan, C., Anderson, A., and Zemel, R. Understanding the origins of bias in word embeddings. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 803–811. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/brunet19a.html.

Cook, R. D. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365): 169–174, 1979. ISSN 01621459. URL http://www.jstor.org/stable/2286747.

Cover, T. and Thomas, J. *Elements of Information Theory*. Wiley Interscience, 1991.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dwivedi, R., Singh, C., Yu, B., and Wainwright, M. Revisiting minimum description length complexity in overparameterized models. *Journal of Machine Learning Research*, 24(268):1–59, 2023. URL http://jmlr.org/papers/v24/21-1133.html.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, Feb 2017. ISSN 1476-4687. doi: 10.1038/nature21056. URL https://doi.org/10.1038/nature21056.

Farnia, F. and Tse, D. A minimax approach to supervised learning, 2017.

Feldman, V. Does learning require memorization? a short tale about a long tail, 2021.

Feldman, V. and Zhang, C. What neural networks memorize and why: Discovering the long tail via influence estimation, 2020.

Fisher, J., Liu, L., Pillutla, K., Choi, Y., and Harchaoui, Z. Influence diagnostics under self-concordance. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10028–10076. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/fisher23a.html.

Fogel, Y. and Feder, M. Universal learning of individual data. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2289–2293, 2019. doi: 10.1109/ISIT.2019.8849222.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. Fast approximate natural gradient descent in a kronecker-factored eigenbasis. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018.

Ghassemi, M. and Mohamed, S. Machine learning and health need better values. *npj Digital Medicine*, 5 (1):51, Apr 2022. ISSN 2398-6352. doi: 10.1038/s41746-022-00595-9. URL https://doi.org/10.1038/s41746-022-00595-9.

Graves, A. Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf.

Griewank, A. and Walther, A. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., Hubinger, E., Lukošiūtė, K., Nguyen, K., Joseph, N., McCandlish, S., Kaplan, J., and Bowman, S. R. Studying large language model generalization with influence functions, 2023.

Grunwald, P. A tutorial introduction to the minimum description length principle, 2004.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974. ISSN 01621459. URL http://www.jstor.org/stable/2285666.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Hinton, G. E. and van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pp. 5–13, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897916115. doi: 10.1145/168304.168306. URL https://doi.org/10.1145/168304.168306.

Hinton, G. E. and Zemel, R. Autoencoders, minimum description length and helmholtz free energy. In Cowan, J., Tesauro, G., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993. URL https://proceedings.neurips.cc/paper_files/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization, 2019.

Kass, R. E. and Raftery, A. E. Bayes factors. *Journal of the American Statistical Association*, 90(430): 773–795, 1995. doi: 10.1080/01621459.1995.10476572. URL /brokenurl#http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476572.

Kim, T., Oh, J., Kim, N., Cho, S., and Yun, S.-Y. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation, 2021.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894. PMLR, 2017.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses, 2021.

Kolmogorov, A. N. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–7, 1965.

Krantz, S. G. and Parks, H. R. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Mackay, D. J. C. *Bayesian methods for adaptive models*. PhD thesis, USA, 1992. UMI Order No. GAX92-32200.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. A simple baseline for bayesian uncertainty in deep learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/118921efba23fc329e6560b27861f0c2-Paper.pdf.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/8d8818c8e140c64c743113f563cf750f-Paper.pdf.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf.

Pakdaman Naeini, M., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. doi: 10.1609/aaai.v29i1.9602. URL https://ojs.aaai.org/index.php/AAAI/article/view/9602.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning (ICML)*, 2023.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ac56f8fe9eea3e4a365f29f0f1957c55-Paper.pdf.

Perotti, J. I., Tessone, C. J., Clauset, A., and Caldarelli, G. Thermodynamics of the minimum description length on community detection, 2018.

Platt, J. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. Estimating training data influence by tracing gradient descent. In Larochelle, H., Ranzato, M.,

Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf.

Ren, J., Fort, S., Liu, J., Roy, A. G., Padhy, S., and Lakshminarayanan, B. A simple fix to mahalanobis distance for improving near-ood detection, 2021.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(78)90005-5. URL https://www.sciencedirect.com/science/article/pii/0005109878900055.

Rissanen, J. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986. ISSN 00905364. URL http://www.jstor.org/stable/3035559.

Rissanen, J. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996. doi: 10.1109/18.481776.

Ritter, H., Botev, A., and Barber, D. A scalable laplace approximation for neural networks. In *Proceedings of ICLR*, 2018.

Roos, T. and Rissanen, J. On sequentially normalized maximum likelihood models. 01 2008.

Roos, T., Silander, T., Kontkanen, P., and Myllymaki, P. Bayesian network structure learning using factorized nml universal models. In *2008 Information Theory and Applications Workshop*, pp. 272–276, 2008. doi: 10.1109/ITA.2008.4601061.

Schulam, P. and Saria, S. Can you trust this prediction? auditing pointwise reliability after learning. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1022–1031. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/schulam19a.html.

Schwarz, G. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461 – 464, 1978. doi: 10.1214/aos/1176344136. URL https://doi.org/10.1214/aos/1176344136.

Shtar'kov, Y. M. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 23:175–186, 1987.

Solomonoff, R. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, 1964. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(64)90223-2. URL https://www.sciencedirect.com/science/article/pii/S0019995864902232.

Srikanth, M., Irvin, J., Hill, B. W., Godoy, F., Sabane, I., and Ng, A. Y. An empirical study of automated mislabel detection in real world vision datasets, 2023.

Stine, R. A. and Foster, D. P. The competitive complexity ratio. In *Proceedings of the 2001 Conference on Information Sciences and Systems*, 2001.

Tanno, R., F. Pradier, M., Nori, A., and Li, Y. Repairing neural networks by leaving the right past behind. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 13132–13145. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/552260cfb5e292e511eaa780806ac984-Paper-Conference.pdf.

Vapnik, V. N. and Chervonenkis, A. Y. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16 (2):264–280, 1971. doi: 10.1137/1116025. URL https://doi.org/10.1137/1116025.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. In *Proceedings of the International Conference on Learning Representations*, 2022.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yang, J., Zhou, K., Li, Y., and Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

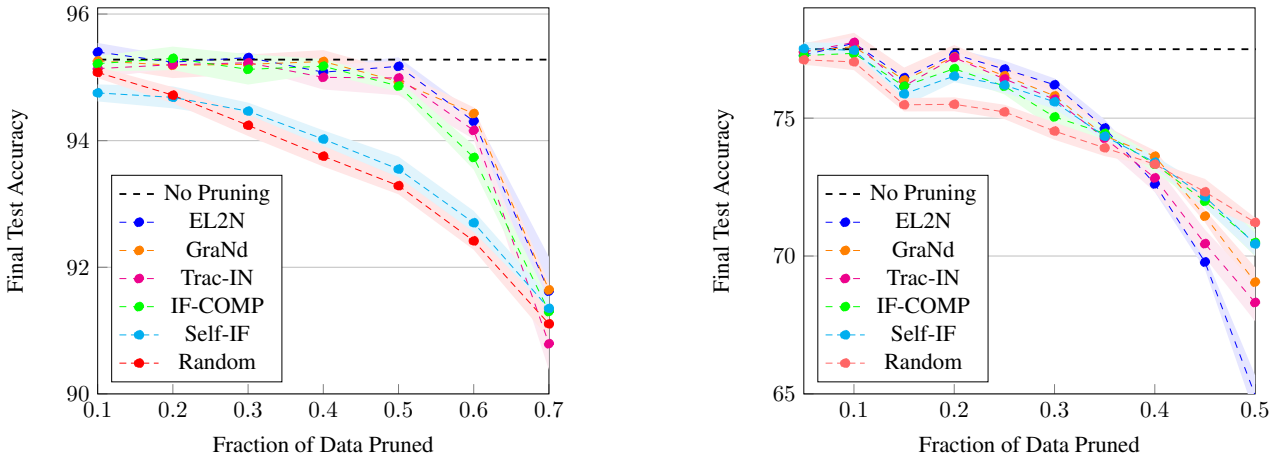
Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. Openood: Benchmarking generalized out-of-distribution detection. 2022.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization, 2017.

Zhang, J., Yang, J., Wang, P., Wang, H., Lin, Y., Zhang, H., Sun, Y., Du, X., Zhou, K., Zhang, W., Li, Y., Liu, Z., Chen, Y., and Li, H. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.

Zhaowei Zhu, Yiwen Song, Y. L. Clusterability as an alternative to anchor points when learning with noisy labels. In *Proceedings of ICML*, 2021.

Zhou, A. and Levine, S. Amortized conditional normalized maximum likelihood: Reliable out of distribution uncertainty estimation, 2021.



(a) Data pruning results on CIFAR-10

(b) Data pruning results on CIFAR-100

Figure 5. Data pruning results. Shaded regions correspond to standard deviations over 5 seeds.IF-COMP performs similarly to other methods that require access to additional checkpoints, including Trac-IN, GraNd, and EL2N. At the highest pruning levels for CIFAR-100, IF-COMP and Self-IF outperform baselines that perform worse than random.

A. Additional Experiments

A.1. Unrestricted Oracle

We consider ground truth retraining with the unrestricted oracle objective (3) on CIFAR-10 with a ResNet-18 model. Similar to our experiments in Section 3.4, we select 20 examples randomly from CIFAR-10, CIFAR-100, and MNIST test sets, then apply each of the possible 10 CIFAR labels for a total of 600 additional training examples.

For each example, we train the model *from scratch* on the full training set with the added example, and evaluate the probability of the true label class at convergence. We find that the model is able to effectively memorize all additional examples provided to it, achieving close to 1 probability consistently across all classes. Using this oracle probability in the pNML formulation produces a constant uniform distribution that is useless for any tasks.

A.2. Dataset Pruning

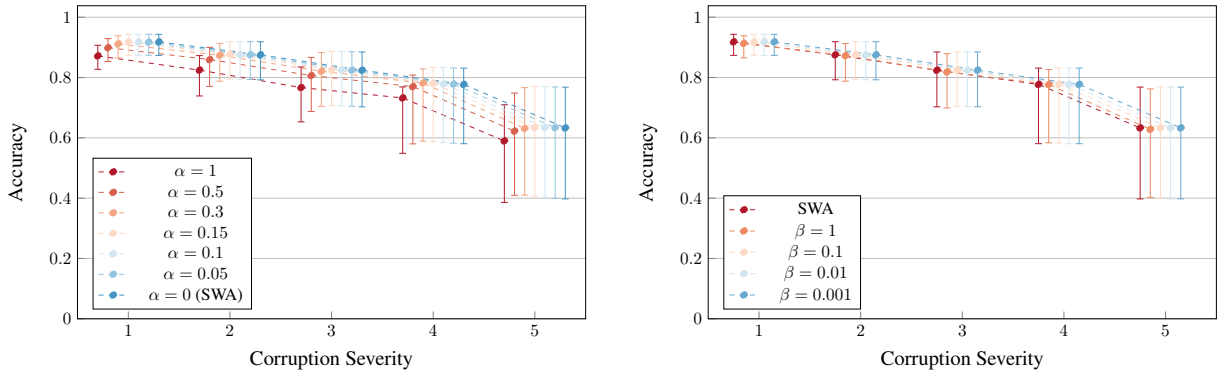
We consider the task of dataset pruning, with the goal of removing training examples that do not provide essential training information to the model. Specifically, given a model trained on a dataset, we aim to rank training examples such that removing the least important examples degrades final test accuracy as little as possible. We consider ResNet-18 models trained on both CIFAR-10 and CIFAR-100 datasets, with varying levels of data pruning. We use IF-COMP to rank points, removing ones with the lowest complexity first and keeping those with higher complexity.

For baselines, we consider Trac-IN, GraNd, EL2N, and Self-IF, which follow our experimental setup for mislabel detection. We also consider a no pruning and random pruning baseline. Once points are ranked, we train 5 models with different seeds on each pruned dataset. Results are shown in Figure 5. We find that IF-COMP performs similarly to baseline methods across both datasets, even those that utilize additional training checkpoints. On CIFAR-100 at the highest pruning levels, IF-COMP and Self-IF outperform baselines, although all methods perform worse than random.

A.3. α and β Ablations

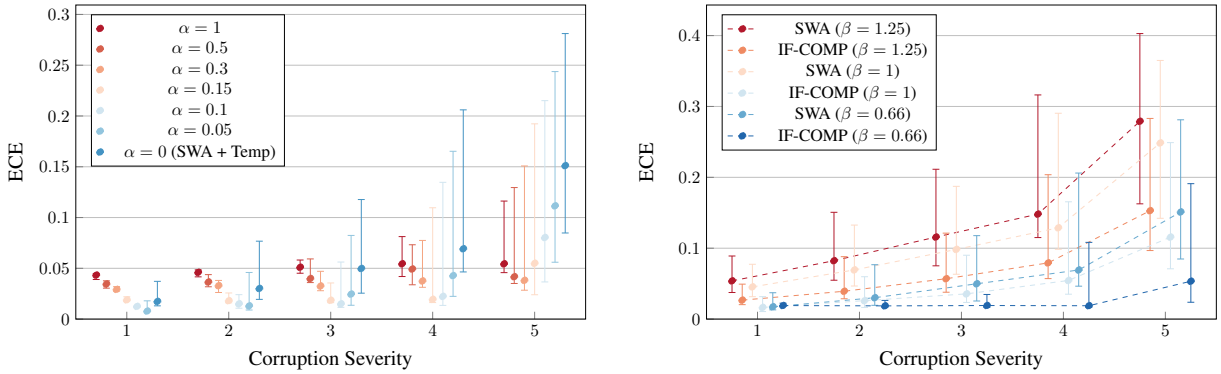
In this section we provide additional ablations on the α weighting parameter and β inverse temperature parameter for the uncertainty calibration task. We measure their effect on accuracy in Figure 6 and on ECE in Figure 7. We compare all methods against the SWA baseline. When not specified, we set default values of $\beta = 0.66$ and $\alpha = 0.15$.

We find that accuracy is minimally affected by α, and only begins to slightly degrade as α becomes particularly large (> 0.5). Since a maximum value of only $\alpha = 0.3$ is required to achieve optimal ECE on severity level 5, even prioritizing calibration on the highest severity levels does not degrade accuracy. Similarly, β values have little effect on accuracy, even



(a) α ablations on accuracy. (b) β ablations on accuracy.

Figure 6. Ablations on the effects of the α and β hyperparameters on model accuracy. IF-COMP exhibits minimal differences from SWA baseline accuracy at all severity levels, except for particularly large values of α.



(a) α ablations on ECE (b) β ablations on ECE

Figure 7. Ablations on the effects of the α and β hyperparameters on ECE. Different values of α are optimal for different severity levels. Selecting an α value allows practitioners to prioritize calibration on different images. Applying IF-COMP on top of SWA models improves ECE across all severity levels even for varying values of β (α is not tuned and held constant at 0.15).

for extremely small values. This indicates that the BIF for each label provides as much information as the model output.

We find that the ECE for each corruption severity level changes with α, where smaller values of α are better for cleaner images, and larger values of α are better for noisier images. Tuning α allows practitioners to select what kinds of test images to prioritize strong calibration on. Compared to the raw outputs produced by temperature scaling SWA model outputs, IF-COMP consistently improves ECE across all severity levels for varying values of β. We do not tune a separate α for each β, and use the same value of 0.15.

A.4. Timing Comparisons

In this section we provide timing comparisons for IF-COMP compared to ACNML and gradient norm methods. Specifically, we compute the time necessary to calculate the corresponding parametric complexity values in Section 3.4. For all experiments we use a single A40 GPU and attempt to use the same chunk size for all `vmap` operations. However, ACNML on CIFAR-100 runs out of memory, so we reduce the chunk size from 4 to 1, meaning we perform the `vmap` only over the label space. As in other experiments, we do not use the spatially uncorrelated activations (SUA) assumption for MNIST models, making the EKFAC computations slightly more expensive. We present our results in Table 3.

For small LeNet models, we find that the additional EKFAC operation performed in IF-COMP does not add much overhead compared to a simple gradient norm. In addition, IF-COMP provides a 7 times speedup compared to ACNML. For larger

Table 3. Inference time per input (in seconds)

Method	MNIST LeNet	CIFAR-10 ResNet18	CIFAR-100 ResNet18
Forward Pass	0.000004s	0.00002s	0.00002s
GradNorm	0.000226s	0.00336s	0.03783s
ACNML-EKFAC	0.001770s	0.23183s	2.33628s
IF-COMP (ours)	0.000241s	0.01576s	0.18883s
IF-COMP vs. ACNML Speedup	7.34 $\times$	14.71 $\times$	12.4 $\times$

ResNet18 models, EKFAC becomes slower than the gradient norm, but still provides a 12-15 times speedup over ACNML.

B. Experimental Details

B.1. Computing Environment

All experiments were implemented in PyTorch and were run on single RTX6000 or A40 GPUs.

B.2. EKFAC

We implement a version of the Eigenvalue Kronecker-Factored Approximate Curvature (EKFAC) (George et al., 2018) to efficiently approximate the inverse Fisher information matrix. For efficiency, in convolutional layers we assume spatially uncorrelated activations (SUA), except for MNIST LeNet models, for which removing the SUA assumption does not introduce significant computational overhead. BatchNorm layers are implemented with a diagonal approximation which do not require eigendecomposition. We take a single pass through the full training set and sample a single label to calculate the activation and output gradient covariances, then perform the relevant eigendecompositions. We then take another full pass through the training set to estimate the second moments in the eigenbasis with a single label sample. Both passes perform no data augmentation and use the original training images. Calculating the BIF is performed by taking a temperature-scaled loss gradient, projecting into the eigenbasis, then taking the norm scaled by the inverse of the second moments. We apply a consistent damping value of $\delta = 1\text{e}{-30}$ for LeNet models and $\delta = 1\text{e}{-12}$ for ResNet-18 models.

To further improve efficiency, we use the PyTorch `vmap` operation to vectorize computations both across label spaces and across examples. We use an example chunk size of 8 for MNIST and CIFAR-10 experiments, and an example chunk size of 4 for CIFAR-100 experiments, with full chunking across the label space.

B.3. Uncertainty Calibration

All models are ResNet-18 models. CIFAR-10 ensemble models were trained with following standard training procedures using SGD with momentum of 0.9, weight decay of 0.0005, and a learning rate of 0.1 that decays by a factor of 5 at epochs 60, 120, and 160. To train SWA models we follow the setup from (Izmailov et al., 2019) and follow this setup for 160 epochs, then set a constant 0.01 learning rate and average checkpoints for another 140 epochs for a total of 300 epochs. For SWA and SWAG-D models, batch statistics were then updated on the full training set. For dropout models, we place dropout layers after the ReLU nonlinearities in each block before the residual connection, and test rates of 0.1, 0.2, 0.3, and 0.4. Empirically we find a rate of 0.3 to perform the best without significant accuracy degradation. We follow the setup described in Zhou & Levine (2021) for training ACNML models, including the recommended step size and weighting term. All Bayesian methods use 30 samples from the posterior which we average. IF-COMP uses a temperature of 1.5 and an α weight of 0.15, which we find by tuning on a held out validation set of additional perturbations of varying severity. Reliability diagrams and ECE values are calculated siimlar to Zhou & Levine (2021) by binning exampels sorted by confidence into 20 bins, then calculating the average L1 norm between the average confidence and accuracy within each bin.

B.4. Mislabel Detection

For both CIFAR-10 and CIFAR-100 datasets we use a ResNet-18 model trained with the standard training procedure detailed in the section above, with early stopping calculated on a clean validation set. We retrieve human and data-dependent noise directly from the relevant repositories, and generate asymmetric and symmetric noise locally. Asymmetric noise is applied

to CIFAR-10 with a predefined label mapping, and for CIFAR-100 we perform a mapping to the next fine-grained class in the corresponding coarse model class.

For Trac-IN we take model checkpoints every 20 epochs for a total of 10 checkpoints, then compute gradient norms on the full model. EL2N and GraNd are computed by training a model for epoch 20 with 10 different seeds, then calculating average error L2 norm and gradient norms across all models. Self-IF is calculated using the same EKFAC estimated for IF-COMP with full gradients. IF-COMP is calculated at $\beta = 0.001$ for all CIFAR-10 experiments and $\beta = 1$ for all CIFAR-100 experiments. Note that we fit the EKFAC on the dataset with noised labels.

B.5. Outlier Detection

All baseline results are provided in the OpenOOD benchmark (Yang et al., 2022; Zhang et al., 2023), as well as the pre-trained LeNet and ResNet-18 models used for MNIST and CIFAR-10/CIFAR-100 experiments. We use a temperature of $\beta = 2$ for MNIST experiments, $\beta = 0.001$ for CIFAR-10 experiments, and $\beta = 1$ for CIFAR-100 experiments. Temperature values are found by tuning on a held out validation set of OOD examples that share no classes with the ones used for testing. This validation set is the same one used to tune hyperparameters for all benchmarks in the OpenOOD and are provided as a standard procedure from the framework (Zhang et al., 2023). We hypothesize that the extremely high temperatures necessary for strong CIFAR-10 results are a result of the stable and flat optima found with Batchnorm which prevent learning arbitrary labels without moving far away in weight and function space.