Amortized Conditional Normalized Maximum Likelihood: Reliable Out of Distribution Uncertainty Estimation

Aurick Zhou [1] **Sergey Levine** [1]

Abstract

While deep neural networks provide good performance for a range of challenging tasks, calibration and uncertainty estimation remain major challenges, especially under distribution shift. In this paper, we propose the amortized conditional normalized maximum likelihood (ACNML) method as a scalable general-purpose approach for uncertainty estimation, calibration, and out-of-distribution robustness with deep networks. Our algorithm builds on the conditional normalized maximum likelihood (CNML) coding scheme, which has minimax optimal properties according to the minimum description length principle, but is computationally intractable to evaluate exactly for all but the simplest of model classes. We propose to use approximate Bayesian inference technqiues to produce a tractable approximation to the CNML distribution. Our approach can be combined with any approximate inference algorithm that provides tractable posterior densities over model parameters. We demonstrate that ACNML compares favorably to a number of prior techniques for uncertainty estimation in terms of calibration when faced with distribution shift.

1. Introduction

Current machine learning methods provide unprecedented accuracy across a range of domains, from computer vision to natural language processing. However, in many high-stakes applications, such as medical diagnosis or autonomous driving, rare mistakes can be extremely costly. Thus, effective deployment of learned models requires not only high accuracy, but also a way to measure the certainty in a model's predictions in order to assess risk and allow the model to abstain from making decisions when there is low confidence in the prediction. While deep networks offer excellent prediction accuracy, they generally do not provide the means to accurately quantify their uncertainty. This is especially true on out-of-distribution inputs, where deep networks tend to make overconfident incorrect predictions (Ovadia et al., 2019). In this paper, we tackle the problem of obtaining reliable uncertainty estimates under distribution shift, with the aim of producing models that can reliably report their uncertainty even when presented with unexpected inputs.

Most prior work approaches the problem of uncertainty estimation from the standpoint of Bayesian inference. By treating parameters as random variables with some prior distribution, Bayesian inference can compute posterior distributions that capture a notion of *epistemic* uncertainty and allow us to quantitatively reason about uncertainty in model predictions. However, computing accurate posterior distributions becomes intractable as we use very complex models like deep neural nets, and current approaches require highly approximate inference methods that fall short of the promise of full Bayesian modeling in practice.

Bayesian methods also have a deep connection with the minimum description length (MDL) principle, a formalization of Occam's razor that casts learning as performing efficient data compression and has been widely used as a motivation for model selection techniques. Codes corresponding to maximum-a-posteriori estimators and Bayes marginalization have been commonly used within the MDL framework. However, other coding schemes have been proposed in MDL centered around achieving different notions of minimax optimality. Interpreting coding schemes as predictive distributions, such methods can directly inspire prediction strategies that give conservative predictions and do not suffer from excessive overconfidence due to their minimax formulation.

One such predictive distribution is the *conditional normalized maximum likelihood* (CNML) (Grünwald, 2007; Rissanen and Roos, 2007; Roos et al., 2008) model, also known as sequential NML or predictive NML (Fogel and Feder, 2018b). To make a prediction on a new input, CNML considers every possible label and finds the model that best explains that label for the query point together with the training set. It then uses that corresponding model to assign probabilities for each input and normalizes to obtain a

[1]EECS, University of California, Berkeley, USA. Correspondence to: Aurick Zhou <aurick@berkeley.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

valid probability distribution. We will argue that the CNML prediction strategy can be useful for providing reliable uncertainty estimates on out-of-distribution inputs. Intuitively, instead of relying on a learned model to extrapolate from the training set to the new (potentially out-of-distribution) input, CNML can obtain more reasonable predictive distributions by explicitly updating a model for each potential label of the particular test input and then asking "given the training data, which labels would make sense for this input?"

While CNML provides compelling minimax regret guarantees, practical instantiations have been exceptionally difficult, because computing predictions for a test point requires retraining the model on the test point *concatenated with the entire training set*. With large models like deep neural networks, this can require hours of training for every prediction, rendering naive CNML schemes infeasible for practical use.

In this paper, we argue that prediction strategies inspired by CNML, which output conservative predictions that depend on models explicitly trained on the test input, can provide reasonable uncertainty estimates even when faced with out-of-distribution data. To instantiate such a strategy tractably, we propose *amortized CNML* (ACNML), a practical algorithm for approximating CNML utilizing approximate Bayesian inference. ACNML avoids the need to optimize over large datasets during inference by using an approximate posterior in place of the training set. We show that our proposed approach is compares favorably to number of prior techniques for uncertainty estimation on out-of-distribution inputs, and is substantially more feasible and computationally efficient than prior techniques for using CNML predictions with deep neural networks.

2. Conditional Normalized Maximum Likelihood

ACNML is motivated from the minimum description length (MDL) principle, which states that any regularities in a dataset can be exploited to compress it, and so learning is reformulated as encoding the data as efficiently as possible. (Rissanen, 1989; Grünwald, 2007). While MDL is typically described in terms of code lengths, we can associate codes with probability distributions, with the code length of an object corresponding to the negative log-likelihood under that probability distribution. MDL was originally formulated in a generative setting where the goal is to code arbitrary data, we focus here on a supervised learning setting, where we assume the inputs are already known and our goal is to only encode/predict the labels.

Normalized Maximum Likelihood. Suppose we have a model class Θ, where each $\theta \in \Theta$ corresponds to a conditional distribution $p_\theta(y|x)$. Let $\hat{\theta}(y_{1:n}|x_{1:n})$ denote the maximum likelihood estimator for a sequence of labels $y_{1:n}$ corresponding to inputs $x_{1:n}$ over all $\theta \in \Theta$. Given a sequence of inputs $x_{1:n}$ and labels $y_{1:n}$, we can define a regret for a distribution over labels q as

$$R(q, y_{1:n}, x_{1:n}, \Theta) \stackrel{\text{def}}{=} \log p_{\hat{\theta}(y_{1:n}|x_{1:n})}(y_{1:n}|x_{1:n}) - \log q(y_{1:n}). \quad (1)$$

In relation to the MDL principle, this regret corresponds to the excess number of bits q uses to encode the labels $y_{1:n}$ compared to the best distribution in the model class Θ. For any fixed input sequence, we can then define the *normalized maximum likelihood distribution* (NML) as

$$p^{\text{NML}}(y_{1:n}|x_{1:n}) = \frac{p_{\hat{\theta}(y_{1:n}|x_{1:n})}(y_{1:n}|x_{1:n})}{\sum_{\tilde{y}_{1:n}\in\mathcal{Y}^n} p_{\hat{\theta}(\tilde{y}_{1:n}|x_{1:n})}(\tilde{y}_{1:n}|x_{1:n})}. \quad (2)$$

The NML distribution can be shown to achieve minimax regret (Shtarkov, 1987; Rissanen, 1996) as it achieves the same regret for all label sequences.

$$p^{\text{NML}} = \operatorname*{argmin}_q \max_{y_{1:n}\in\mathcal{Y}^n} R(q, y_{1:n}, x_{1:n}, \Theta). \quad (3)$$

This corresponds, in a sense, to an optimal coding scheme for sequences of labels of known fixed length n.

Conditional NML. Instead of making predictions across entire sequences of labels at once, NML can be adapted to the setting where we make predictions about only the next label based on the previously seen data, resulting in *conditional NML* (CNML) (Rissanen and Roos, 2007; Grünwald, 2007; Fogel and Feder, 2018a). While several variations on CNML exist, we consider the following:

$$p^{\text{CNML}}(y_n|x_n; x_{1:n-1}, y_{1:n-1}) \propto p_{\hat{\theta}(y_{1:n}|x_{1:n})}(y_n|x_n), \quad (4)$$

which solves the minimax problem

$$p^{\text{CNML}} = \operatorname*{argmin}_q \max_{y_n} \log p_{\hat{\theta}(y_{1:n}|x_{1:n})}(y_n|x_n) - \log q(y_n). \quad (5)$$

We note that the inner maximization is only over the next label y_n that we are predicting, rather than the full sequence as before. This prediction strategy is now amenable to our typical supervised learning setting, where $(x_{1:n-1}, y_{1:n-1})$ is our training set, and we want to output a predictive distribution over labels y_n for a new test input x_n.

CNML provides conservative predictions. Here we motivate why CNML can provide reasonable uncertainty estimates for out-of-distribution inputs. For each query point, CNML considers each potential label and finds the model that would be most consistent with that label and with the training set. If that model assigns high probability to the label, then minimizing the worst-case regret forces CNML to assign relatively high probability to it. Compared to simply letting a model trained only on the training set extrapolate

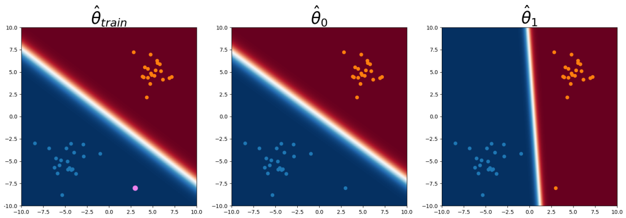


Figure 2. Given the labeled training set (blue and orange dots), we want to predict the label at the query input (shown in pink in the left image), which the training set MLE $\hat{\theta}_{\text{train}}$ confidently classifies as the blue class. However, CNML assigns a near-uniform prediction on the query point, as it computes new MLEs $\hat{\theta}_0$ and $\hat{\theta}_1$ (center and right images) by assigning different labels to the query point, and finds both labels are consistent with the training data.

to OOD inputs, we expect CNML to give more conservative predictions on OOD inputs, since it explicitly considers what would have happened if the new data point had been labeled with each possible label.

We use a 2D logistic regression example to illustrate CNML's conservative predictions, showing a heatmap of CNML probabilities in Figure 1. CNML provides uniform predictions on most of the input space away from the training samples. In Figure 2, we illustrate how CNML arrives at these predictions, showing the predictions for the parameters $\hat{\theta}_0$ and $\hat{\theta}_1$, corresponding to labeling the test point (shown in pink in Figure 2, left) with either label 0 or 1.

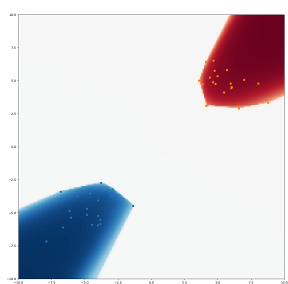
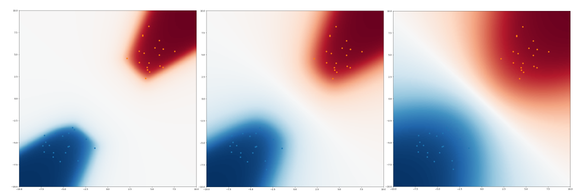


Figure 1. CNML probabilities with a logistic regression model. CNML expresses high uncertainty and provides uniform predictions (indicated by the white color) on most of the input space away from the training set (shown in blue and orange dots).

However, CNML may be too conservative when the model class Θ is very expressive. Naïvely applying CNML with large model classes can result in the per-label models fitting their labels for the query point arbitrarily well, such that CNML gives unhelpful uniform predictions even on inputs we would hope to reasonably extrapolate on. We see this in the 2D logistic regression example in Figure 1. Thus, the model class Θ would need to be restricted in some form, for example by only considering parameters within a certain distance from the training set solution as a hard constraint.

Another approach for controlling the expressivity of the model class is to generalize CNML to use *regularized* estimators instead of maximum likelihood, resulting in normalized maximum a posteriori (NMAP) (Kakade et al., 2006) codes. Instead of using maximum likelihood parameters, NMAP selects $\hat{\theta}$s to be the parameter that maximizes both data likelihood and a regularization term, or prior, over parameters, and we can define slightly altered notions of regret using these MAP estimators in all the previous equations to get a *conditional normalized maximum a posteriori* distribution instead. See Appendix D for completeness.

Going back to the logistic regression example, we plot heatmaps of CNMAP predictions in Figure 3, adding different amounts of L2 regularization to the logistic regression weights. As we add more regularization, the model class becomes effectively less expressive, and the CNMAP predictions become less conservative.



(a) $\lambda = 0.1$ (b) $\lambda = 1$ (c) $\lambda = 10$

Figure 3. CNMAP probabilities with different levels of L2 regularization $\lambda \|w\|_2^2$. Predictions are less conservative as λ increases.

Computational costs of CNML. While we have argued that CNML can provide an appealing approach for uncertainty estimation for out-of-distribution inputs, it can be exceptionally impractical to instantiate, particularly with large models like neural networks, due to the prohibitive computational costs of computing the maximum likelihood estimators for each new input and label. To evaluate the distribution on a new test point, one must solve a nonconvex optimization problem for each possible label, with each problem involving the entire training dataset along with the new test point. This direct evaluation of CNML therefore becomes computationally infeasible with large datasets and high-capacity models, and further requires that the model carry around the entire training set even when it is deployed. In settings where critical decisions must be made in real time, even running a single epoch of additional training would be infeasible. For this reason, NML-based methods have not gained much traction as a practical tool for improving the predictive performance of high-capacity models.

3. Amortized CNML

In this section, we derive our method, amortized conditional normalized maximum likelihood (ACNML), which provides a tractable approximation for CNML and CNMAP via approximate Bayesian inference. Instead of directly computing maximum likelihood parameters over the query point and training set, our method uses an approximate posterior distribution over parameters to capture the necessary information about the training set, reducing the maximization to only the single new point. The computational cost at test-time therefore does not increase with training set size.

Algorithm 1 Amortized CNML (ACNML)

Input: Model class Θ, Training Data $(x_{1:n-1}, y_{1:n-1})$, Test Point: x_n, Classes (1, ..., k)
Output: Predictive distribution $p(y|x_n)$
Training: Run approximate inference algorithm on training data $(x_{1:n-1}, y_{1:n-1})$ to get posterior density $q(\theta)$
for all possible labels $i \in (1, \ldots, k)$ **do**
  Compute $\hat{\theta}_i = \text{argmax}_\theta \log p_\theta(i|x_n) + \log q(\theta)$
end for
Return $p(y|x_n) = \frac{p_{\hat{\theta}_y}(y|x_n)}{\sum_{i=1}^k p_{\hat{\theta}_i}(i|x_n)}$

3.1. Algorithm Derivation

Incorporating an exact posterior into CNML. Given a prior distribution $p(\theta)$, the Bayesian posterior likelihood conditioned on the training data is given by

$$p(\theta|x_{1:n-1}, y_{1:n-1}) \propto p(\theta)p_\theta(y_{1:n-1}|x_{1:n-1}). \quad (6)$$

We can write the MAP estimators in the CNMAP distribution for a fixed query input x_n as

$$\hat{\theta}_y = \underset{\theta\in\Theta}{\text{argmax}} \underbrace{\log p_\theta(y_{1:n-1}|x_{1:n-1}) + \log p(\theta)}_{\log p(\theta|x_{1:n-1}, y_{1:n-1})} + \log p_\theta(y|x_n) \quad (7)$$

We can thus replace the training data log-likelihood $p_\theta(y_{1:n-1}|x_{1:n-1})$ with the Bayesian posterior density $\log p(\theta|x_{1:n-1}, y_{1:n-1})$ when computing $\hat{\theta}_y$. We can also recover CNML as a special case of CNMAP by using a uniform prior, but as discussed previously, CNML with highly expressive model classes can lead to overly conservative predictions, so we will opt to use non-uniform priors that help control model complexity instead. For example, we may use a zero-mean Gaussian prior $p(\theta)$ over our weights, corresponding to L2 regularization.

ACNML with an approximate posterior. Of course, the exact Bayesian likelihood is no easier to compute than the original training log likelihood. However, we can derive a tractable approximation by replacing the exact posterior $p(\theta|x_{1:n-1}, y_{1:n-1})$ with an approximate posterior $q(\theta)$ instead. We can obtain an approximate posteriors via standard approximate Bayesian techniques such as variational inference or Laplace approximations. We focus on Gaussian posterior approximations for computational efficiency, and discuss in Section 3.2 why this class of distributions provides a reasonable approximation for large datasets.

For practical purposes, we expect the approximate posterior log-likelihood to ensure the optimal $\hat{\theta}_y$ selected for each label retains good performance on the training set. By replacing the likelihood over the training data with the probability under an approximate posterior, it becomes unnecessary to retain the training data at test time, only the parameters of the approximate distribution. Optimization also becomes much simpler, as it no longer requires stochastic gradients, and the Gaussian posterior log density $\log q(\theta)$ serves as a strongly convex regularizer.

ACNML algorithm summary A summary of the ACNML algorithm is presented in Algorithm 1. The training process for obtaining $q(\theta)$ only needs to be performed once on the training set, whereas the inference step is performed for each test point. However, this inference step only requires optimizing the model on a single data point with a regularizer provided by $\log q(\theta)$.

3.2. Analysis of ACNML with Gaussian Posteriors

In this section, we argue that using a Gaussian approximate posterior in ACNML, which correspond to second-order approximations to the training set log-likelihood, suffices for accurately computing the CNML distributions when the training set is large. The intuition is that for large training sets, the combined likelihoods of all the training points dominate over the single new test point, so the perturbed MLEs $\hat{\theta}_y$ remains close to the original training set MLE $\hat{\theta}$, letting us rely on local approximations to the training loss.

Under simplifying assumptions of convexity and smoothness of the training losses, we can formalize this using the concept of *influence functions*, which measure how the MLE (and more general M-estimators) for a dataset changes as the dataset were perturbed by reweighting inputs an infinitesimal amount. Recall that the maximum likelihood estimator for a dataset with n datapoints $(x_{1:n}, y_{1:n})$ is given by

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \frac{1}{n} \sum_{i=1}^n \log p_\theta(y_i|x_i). \quad (8)$$

Influence functions analyze how $\hat{\theta}$ relates to the MLE of a perturbed dataset

$$\hat{\theta}_{x,y,\epsilon} = \underset{\theta}{\text{argmax}} \left(\epsilon \log p_\theta(y|x) + \frac{1}{n} \sum_{i=1}^n \log p_\theta(y_i|x_i) \right), \quad (9)$$

where $\hat{\theta}_{x,y,\epsilon}$ is the new MLE if we perturb the training set by adding a datapoint (x, y) with a weight ϵ. A classical result (Cook and Weisberg, 1982) shows that $\hat{\theta}_{x,y,\epsilon}$ is differentiable (under appropriate regularity conditions) with respect to ϵ with derivative given by the influence function

$$\frac{d\hat{\theta}_{x,y,\epsilon}}{d\epsilon}\Big|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta \log p_{\hat{\theta}}(y|x), \quad (10)$$

where $\hat{\theta}$ is the MLE for the original dataset and $H_{\hat{\theta}}$ the Hessian of the mean training set log-likelihood evaluated at $\hat{\theta}$. CNML computes the MLE after adding datapoint (x, y)

with equal weight as points in the training set, which is precisely $\hat{\theta}_{x,y,\epsilon}$ evaluated at $\epsilon = 1/n$. Thus, for sufficiently large n, a first order Taylor expansion around $\hat{\theta}$ should be accurate and the new parameter can be estimated by

$$\tilde{\theta}_{x,y} = \hat{\theta} - \frac{1}{n} H_{\hat{\theta}}^{-1} \nabla_\theta \log p_{\hat{\theta}}(y|x), \tag{11}$$

which is equivalent to solving

$$\begin{aligned} \tilde{\theta}_{x,y} = \operatorname*{argmax}_{\theta} & \frac{1}{n}(\theta - \hat{\theta})^T \nabla_\theta \log p_{\hat{\theta}}(y|x) \\ & + \frac{1}{2}(\theta - \hat{\theta})^T H_{\hat{\theta}} (\theta - \hat{\theta}). \end{aligned} \tag{12}$$

This suggests that, with large training datasets, the perturbed MLE parameters $\hat{\theta}_y$ in Equation 7 can be approximated accurately using a quadratic approximation to the training log-likelihood, corresponding to a Gaussian posterior obtained via a Laplace approximation. We can explicitly quantify the accuracy of this approximation in the theorem below, which is based on Theorem 1 from Giordano et al. (2019), with full details and proof in Appendix E.

Theorem 3.1. *(Adapted from Giordano et al. (2019)) Consider a training set with n datapoints and an additional datapoint (x, y). Assume assumptions 1-5 hold with constants $C_{op}, C_{IJ}, \Delta_\delta$ as defined in Appendix E. Let $\hat{\theta}_{x,y}$ denote the exact MLE if we had appended (x, y) to the training set, and $\tilde{\theta}_{x,y}$ the parameter obtained via the approximation in Equation 11. Let*

$$\delta = \frac{\sup_{\theta \in \Theta} \max \left\{ \|\nabla_\theta \log p_\theta(y|x)\|_1 , \left\| \nabla_\theta^2 \log p_\theta(y|x) \right\|_1 \right\}}{n+2}. \tag{13}$$

If $\delta \leq \Delta_\delta$, then

$$\|\hat{\theta}_{x,y} - \tilde{\theta}_{x,y}\|_2 \leq 2C_{op}^2 C_{IJ} \delta^2. \tag{14}$$

Given such a bound on how accurately we estimate new parameters, we can explicitly quantify the accuracy of the CNML approximation, with proof in Appendix E.

Proposition 3.2. *Let $\hat{\theta}_{x,y}$ and $\tilde{\theta}_{x,y}$ be the exact and approximate MLEs respectively, after appending the datapoint (x, y) to the training set, and assume $\|\hat{\theta}_{x,y} - \tilde{\theta}_{x,y}\| \leq \delta$ for all y. Further suppose $\log p_\theta(y|x)$ is L-Lipschitz in θ.*

Let $p_{CNML}(y) \propto p_{\hat{\theta}_{x,y}}(y|x)$ and $p_{ACNML}(y) \propto p_{\tilde{\theta}_{x,y}}(y|x)$ denote the exact CNML and approximate CNML distributions respectively. We then have

$$\sup_y |\log p_{CNML}(y) - \log p_{ACNML}(y)| \leq 2L\delta. \tag{15}$$

Theorem 3.1 and Proposition 3.2 suggest the approximation given by ACNML will be increasingly close to the exact CNML distribution as the training set size n grows. However, this formal theoretical result only holds for sufficiently large datasets and requires assumptions including smoothness and convexity of the training loss (for example, the constant C_{op} int he bound depends on how strongly convex the loss is at $\hat{\theta}$), so does not necessarily hold in practical settings with deep neural networks due to nonconvexity.

To interpret how different training points influence the predictions of neural networks, Koh and Liang showed that influence function approximations were able to provide useful predictions for estimating leave-one-out retraining with deep convolutional neural networks. This closely resembles the conditions we encounter when computing parameters for each label of the query point with ACNML, with the key difference being that ACNML *adds* a datapoint while leave-one-out retraining *removes* one. Their empirical results suggest these second-order approximations to the training loss, corresponding to Gaussian approximations in ACNML, may suffice to yield useful predictions about how parameters change when the query point is added, despite lacking formal guarantees with deep neural networks.

4. Related Work

Minimum description length has been used to motivate neural network methods dating back to Hinton and van Camp (1993), who treat description length as a regularizer to mitigate overfitting. The idea of preferring flat minima (Hochreiter and Schmidhuber, 1997) also has its origins in the MDL framework, as it allows a coarser discretization of the weights (and thus fewer bits needed).

Bayesian methods average the predictions of different models sampled from the posterior distribution and typically serve as the starting point for uncertainty estimation in deep networks. A common approach is to use simple tractable distributions to approximate the true posterior (Hoffman et al., 2013; Blundell et al., 2015; Ritter et al., 2018). Recent work (Maddox et al., 2019; Dusenberry et al., 2020) has shown simple Gaussian posterior approximations are able to achieve well-calibrated predictions with marginalization. ACNML utilizes these approximate posterior methods, but in contrast to the Bayesian methods, where the posterior is used to efficiently sample models for Bayesian model averaging, ACNML uses the posterior density to enable efficient optimization without needing to retain the training data.

Ovadia et al. (2019) evaluate various proposed methods for uncertainty estimates in deep learning under different types of distribution shift, finding that good calibration on in-distribution points did not necessarily indicate good calibration under distribution shift, and that methods relying on marginalizing predictions over multiple models (Lakshminarayanan et al., 2016; Srivastava et al., 2014) gave better uncertainty estimates under distribution shift than other techniques. In our experiments, we show that our method ACNML maintains much better calibration under distribution shift than prior methods.

Similarly to ACNML, Test Time Training (TTT) (Sun et al., 2020) updates the model on test inputs to improve out-of-distribution performance. One key differences is that TTT relies on an auxiliary self-supervised task to solve on the new test point, and so requires domain knowledge to specify a nontrivial task that is useful for predictions. Additionally, the goal of TTT was to enable *more accurate* prediction under distribution shift, whereas our goal with ACNML was to provide more reliable *uncertainty estimates*.

Perhaps most closely related to our work, Fogel and Feder (2018b) advocate for the use of the CNML distribution in the context of supervised learning (under the name predictive NML), citing its minimax properties. Bibas et al. (2019) estimate the CNML distribution with deep networks by finetuning the last layers of the network on every test input and label combination appended to the training set. Since this finetuning procedure trains for several epochs, it is very computationally intensive at test-time and requires continued access to the entire training set when evaluating. In contrast, our method amortizes this procedure by condensing the information in the training data into a distribution over parameters, allowing for much faster test-time inference without needing the training data.

In the analysis for our approximation, we draw connections to influence functions (Cook and Weisberg, 1982), which have been studied as asymptotic approximations to how M-estimators change when perturbing a dataset. In deep learning, Koh and Liang advocated for using influence functions to interpret neural nets, generate adversarial examples, and diagnose errors in datasets. We use a theorem from Giordano et al. (2019), which broadened the necessary assumptions for these infinitisemal approximations to be accurate and provides explicit guarantees for specific datasets rather than simply asymptotic results.

For out-of-distribution detection, Xiao et al. (2020) propose an approach that updates a generative model to maximize the likelihood of the test input and uses the amount of improvement in log likelihood as a statistic for OOD detection. Our work differs in that we tackle model calibration for shifted input distributions and only use discriminative models, while their goal is OOD detection and utilize generative models of the data. Nonetheless, we believe this work complements ours and lends additional support to the idea that optimizing models on test points can be valuable for estimating uncertainty under distribution shift.

5. Experiments

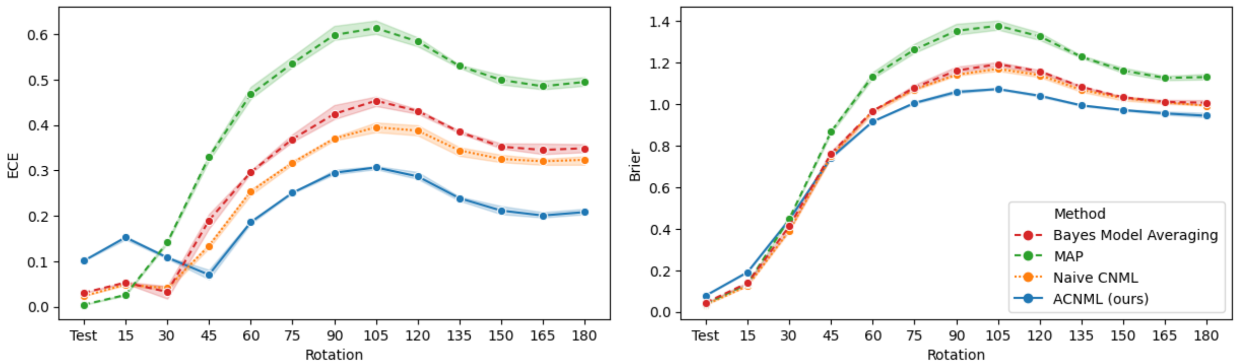
Our experiments aim to evaluate how trustworthy the uncertainty estimates provided by ACNML are under different levels of distribution shift. Following Ovadia et al. (2019), we compare uncertainty estimation across different methods using Brier score and expected calibration error (ECE) (Naeini et al., 2015). Brier score is a proper scoring rule, which captures both how accurate and how calibrated the predictions are, while ECE assesses calibration by directly measuring how closely the predicted confidence corresponds to empirical accuracy. We show that our method is able to to significantly outperform prior works in terms of calibration when distribution shifts became more extreme. While severe distribution shifts mean all methods test perform poorly in terms of accuracy, ACNML is at least able to more reliably indicate when the predictions may be incorrect.

In principle, any method for computing a tractable posterior over parameters can be used with ACNML, and we demonstrate this flexibility by implementing ACNML on top of several different approximate posteriors. By using the exact same posteriors, we can directly compare how uncertainty estimates given by ACNML relate to those of the corresponding Bayesian method.

For each model, we report results across 3 seeds. as well as showing reliability diagrams (Guo et al., 2017) to further qualitatively assess calibration. For reliability diagrams, we sort data points by confidence and divide them into twenty equal sized buckets, plotting the mean accuracy against the mean confidence for each bucket. This allows to see qualitatively see how well the confidence of the prediction relates to the actual accuracy, as well as showing how the confidences are distributed for each method.

Rotated MNIST. We first consider the rotated MNIST task, where out-of-distribution inputs are generated by rotating images from the MNIST test set, with higher levels rotation corresponding to more distribution shift. Here, ACNML is implemented on top of Bayes-by-backprop (Blundell et al., 2015), and we compare to the MAP estimate and Bayes model averaging with the same posterior.

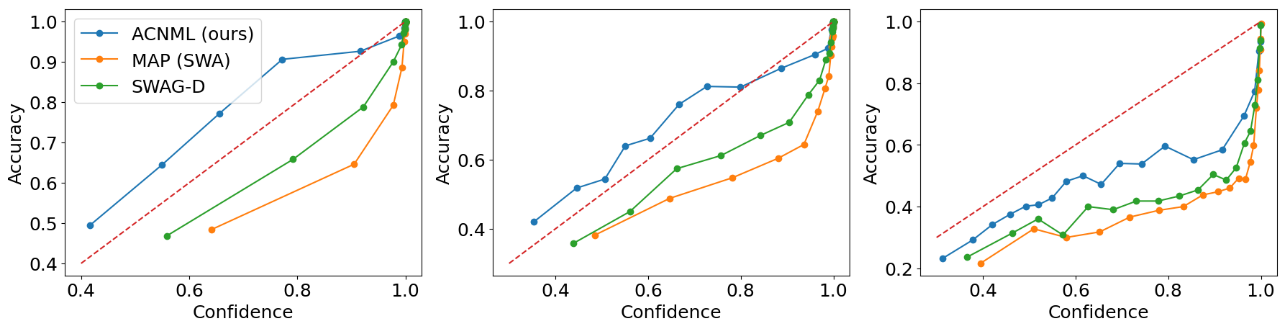
We see in Figure 4 that for higher levels of rotation, corresponding to more out-of-distribution inputs, that ACNML exhibits **substantial improvements in calibration** as measured by the ECE metric, as well as improved Brier scores. However, on the in-distribution test set and the lowest levels of rotation where the models still predict accurately, ACNML's predictions are overly conservative, leading to underconfident predictions and worse calibration than other methods. In general, this agrees with what we expect from ACNML: the predictions are more conservative across the board, which does not necessarily improve results in-distribution, particularly for easy domains like MNIST, but



(a) Rotated MNIST ECEs (lower is better)

(b) Rotated MNIST Brier Scores (lower is better)

Figure 4. ACNML compared against its Bayesian counterpart, the deterministic MAP baseline, and naive CNML on rotated MNIST. We plot means and standard deviations across 3 seeds. We see that ACNML (blue, solid lines) achieves lower ECE as the distribution shift becomes more severe and accuracy decreases, as well as better Brier scores than other methods.



(a) CIFAR10 Test

(b) CIFAR10-C Corruption Level 3

(c) CIFAR10-C Corruption Level 5

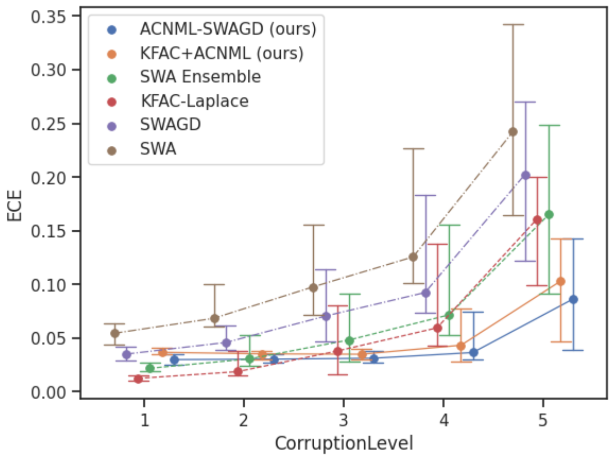
Figure 5. Reliability diagrams plotting confidence vs. accuracy for CIFAR10 in-distribution and OOD data, with a dotted reference line indicating perfect calibration. ACNML provides more conservative predictions than other methods, resulting in better calibration on OOD inputs. For OOD tasks, we show results for the Gaussian blur corruption at levels 3 and 5, with level 5 corresponding to a higher amount of corruption. Each point shows the mean confidence and accuracy within a bucket, so the spread of points along the x-axis shows that ACNML makes more low confidence predictions than other methods.

offer considerable improvements in calibration for out-of-distribution inputs where errors are prevalent.

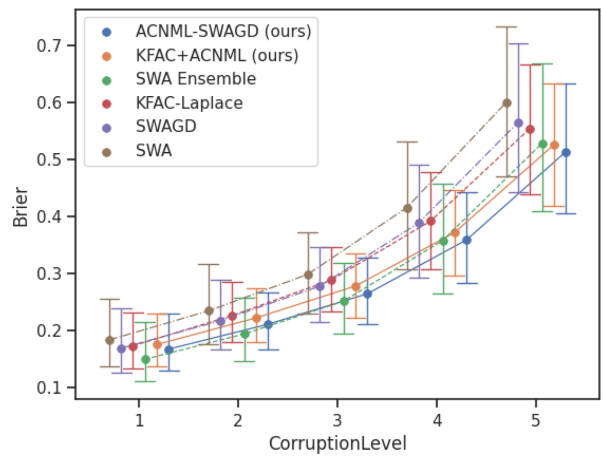
We additionally compare to a much more computationally expensive instantiation of CNML used by Bibas et al. (2019) (denoted naive CNML in Figure 4), which directly finetunes for several epochs using the training set to obtain the optimal parameters for each query point and label, rather than using the approximate posterior like ACNML does. This direct instantiation of CNML improves over the MAP solution in terms of Brier score and calibration on the OOD inputs. However, it is computationally prohibitive, to the point where we were unable to evaluate it on the more complex datasets. On MNIST, each prediction with naive CNML was hundreds of times slower than with ACNML, as shown in Table 1. We also find ACNML is overall more conservative when using this particular posterior approximation, resulting in better calibration on more OOD inputs (see Appendix C for more detailed comparisons between ACNML and naïve CNML).

CIFAR Corruptions. We use CIFAR10 (Krizhevsky, 2012) for training and in-distribution testing, and evaluate uncertainty estimates under distribution shift using the CIFAR10-Corrupted (Hendrycks and Dietterich, 2019) datasets, which apply different severities of 15 common corruptions to the test set images. We can thus assess calibration over a wide variety of distribution shifts, as well as how calibration degrades as distribution shift increases.

We show results here using the VGG16 (Simonyan and Zisserman, 2014) architecture. To compute approximate posteriors, we use Stochastic Weight Averaging - Gaussian (SWAG) (Maddox et al., 2019), and KFAC-Laplace (Ritter et al., 2018). SWAG computes a posterior by fitting a Gaussian distribution to the trajectory of SGD iterates. For simplicity and computational efficiency, we instantiate ACNML with the SWAG-D variant, which uses a Gaussian with diagonal covariance. KFAC-Laplace uses a Gaussian posterior approximation with the MAP solution as the mean and the inverse Hessian of the loss as covariance, approximating the Hessian using KFAC (Martens and Grosse, 2015).



(a) CIFAR10C VGG16 ECEs (lower is better)



(b) CIFAR10C VGG16 Brier Scores (lower is better)

Figure 6. ACNML compared against corresponding Bayesian methods, the deterministic MAP baseline (SWA), and deep ensembles (SWA Ensemble) on out-of-distribution CIFAR10-Corrupted datsets. We plot medians and 95% confidence intervals across all corruptions. We see that ACNML methods (solid lines) achieve much lower ECE at higher corruption values, as well as better Brier scores than other methods.

Focusing on the most direct comparisons, we compare against the MAP solution for the given posterior, which is equivalent to Stochastic Weight Averaging (SWA) (Izmailov et al., 2018), and Bayes model averaging with SWAGD and KFAC-Laplace, which provide apples-to-apples comparisons to the two versions of our method that directly utilize the same posteriors from these prior approaches. We additionally compare to deep ensembles (Lakshminarayanan et al., 2016), which Ovadia et al. (2019) found to provide strong performance in uncertainty estimation under distribution shift, but also takes significantly longer to train due to the need to train independent models.

Examining the reliability diagrams in Figure 5, we can qualitatively see that ACNML provides more conservative (less confident) predictions than other methods across different levels of corruption. On out-of-distribution inputs, where accuracy degrades, we see that ACNML's conservative predictions lead to many better calibrated low-confidence predictions, while other methods drastically overestimate confidence. Thus, ACNML's confidence estimates are still able reliably indicate when predictions are likely to be incorrect even on OOD inputs. ACNML is however slightly *underconfident* on the in-distribution CIFAR10 test set, while other methods err on the side of being overconfident.

In Figure 6, we can quantitatively compare the calibration of different methods for different levels of corruption. ACNML variants provide **much better calibration on the more severe corruptions** than other methods while also performing slightly better in terms of Brier score. All methods perform similarly in terms of accuracy in all domains, and we find that ACNML's more conservative estimates also perform competitively with Bayesian methods in Brier score, and ECE on the in-distribution test set as well (see Table 2 in Appendix B). We include additional comparisons across other methods and architectures in Appendix B.

	MNIST MLP	VGG16	WRN28x10
ACNML (ours)	0.08s	0.37s	1.1s
naïve CNML (per epoch)	13.83s	102.0s	359.1s
feedforward inference	0.0001s	0.0013s	0.004s

Table 1. Inference time per input (in seconds).

Timing Comparison vs. standard CNML. In Table 1, we examine the computational costs of our method. We compare against a naïve implementation of CNML that fine-tunes for N epochs on each test point and label, as in Bibas et al. (2019). In total, predicting a single input with k possible labels involves running kN epochs of training. While ACNML is over two orders of magnitude faster than naïve CNML even with just a single epoch of training (our experiments with naive CNML on MNIST used 5 epochs), it is still slower than standard inference. The computational requirements of our method also scale linearly with the number of classes, but are constant with respect to dataset size. Timing experiments are run using a single NVIDIA 1080Ti, using MNIST for the MNIST MLP timing results and using CIFAR10 for VGG16 and WideResNet28x10, with no parallelization over data points.

6. Discussion

In this paper, we present amortized CNML (ACNML) as an alternative to Bayesian marginalization for obtaining reliable uncertainty estimates and calibrated predictions under distribution shift. The CNML distribution is a theoretically well-motivated strategy derived from the MDL principle with strong minimax optimality properties, but actually evaluating this distribution is computationally daunting. ACNML utilizes approximate Bayesian posteriors to tractably approximate it, can be instantiated on top of a wide range of approximate Bayesian methods, and provides much better calibrated predictions than other methods as the inputs become more out-of-distribution. We view ACNML as a step towards practical uncertainty aware predictions that would be essential for real-world decision making. Future work could further expand on our proposed method, for example by combining ACNML with more complex and expressive posterior approximations. In particular, training losses are highly non-convex and have many local minima, so incorporating local approximations around *multiple* diverse minima could allow for even more reliable uncertainty estimation. More broadly, tractable algorithms inspired by ACNML could in the future provide for substantial improvement in our ability to produce accurate and reliable confidence estimates on out-of-distribution inputs, improving the reliability and safety of learning systems.

Acknowledgements

We thank Aviral Kumar for helpful conversations, as well as anonymous reviewers for valuable feedback on earlier versions of this paper. This research was supported by the DARPA Assured Autonomy program and DARPA LwLL, with compute support from Google Cloud.

References

K. Bibas, Y. Fogel, and M. Feder. Deep pnml: Predictive normalized maximum likelihood for deep neural networks. *arXiv preprint arXiv:1904.12286*, 2019.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight Uncertainty in Neural Networks. *32nd International Conference on Machine Learning, ICML 2015*, 2: 1613–1622, 5 2015.

R. D. Cook and S. Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

M. W. Dusenberry, G. Jerfel, Y. Wen, Y.-a. Ma, J. Snoek, K. Heller, B. Lakshminarayanan, and D. Tran. Efficient and Scalable Bayesian Neural Nets with Rank-1 Factors. *arXiv preprint arXiv:2005.07186*, 2020.

Y. Fogel and M. Feder. Universal Supervised Learning for Individual Data. 12 2018a. URL http://arxiv.org/abs/1812.09520.

Y. Fogel and M. Feder. Universal batch learning with log-loss. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 21–25, 2018b.

Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *33rd International Conference on Machine Learning, ICML 2016*, 3:1651–1660, 6 2015.

R. Giordano, W. Stephenson, R. Liu, M. Jordan, and T. Broderick. A swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147, 2019.

P. Grunwald. A tutorial introduction to the minimum description length principle. 6 2004. URL http://arxiv.org/abs/math/0406077.

P. Grünwald, T. Van Ommen, and others. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *Bayesian Analysis*, 12(4): 1069–1103, 2017.

P. D. Grünwald. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*. The MIT Press, 2007. ISBN 0262072815.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330, 2017.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 5–13, 1993.

S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. In *UAI*, 2018.

S. M. Kakade, M. W. Seeger, and D. P. Foster. Worst-case bounds for Gaussian process models. In *Advances in neural information processing systems*, pages 619–626, 2006.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 6 2012.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 2016.

W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417, 2015.

M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

J. Rissanen. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing Co., Inc., USA, 1989. ISBN 9971508591.

J. Rissanen and T. Roos. Conditional NML universal models. In *2007 Information Theory and Applications Workshop*, pages 337–341, 2007.

J. J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996. ISSN 00189448. doi: 10.1109/18.481776.

H. Ritter, A. Botev, and D. Barber. A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning, 2018.

T. Roos, T. Silander, P. Kontkanen, and P. Myllymaki. Bayesian network structure learning using factorized NML universal models. In *2008 Information Theory and Applications Workshop*, pages 272–276, 2008.

Y. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):186, 1987.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014.

Y. Sun, X. Wang, L. Zhuang, J. Miller, M. Hardt, and A. A. Efros. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

V. G. Vovk. Aggregating Strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, 1990.

Z. Xiao, Q. Yan, and Y. Amit. Likelihood Regret: An Out-of-Distribution Detection Score For Variational Auto-encoder. Technical report, 2020.