The Implicit Delta Method

Anonymous Author(s)
Affiliation
Address
email

Abstract

Uncertainty quantification is a crucial part of drawing credible conclusions from predictive models, whether concerned about the prediction at a given point or any downstream evaluation that uses the model as input. When the predictive model is simple and its evaluation differentiable, this task is solved by the delta method, where we propagate the asymptotically-normal uncertainty in the predictive model through the evaluation to compute standard errors and Wald confidence intervals. However, this becomes difficult when the model and/or evaluation becomes more complex. Remedies include the bootstrap, but it can be computationally infeasible when training the model even once is costly. In this paper, we propose an alternative, the implicit delta method, which works by infinitesimally regularizing the training loss of the predictive model to automatically assess downstream uncertainty. We show that the change in the evaluation due to regularization is consistent for the asymptotic variance of the evaluation estimator, even when the infinitesimal change is approximated by a finite difference. This provides both a reliable quantification of uncertainty in terms of standard errors as well as permits the construction of calibrated confidence intervals. We discuss connections to other approaches to uncertainty quantification, both Bayesian and frequentist, and demonstrate our approach empirically.

1 Introduction

In this paper, we consider quantifying uncertainty in evaluations of predictive models trained on data. Consider the following examples. We fit a complex model (such as a neural net) to predict mean service time for an incoming call to a call center given some features, and we use it to prioritize calls in a queuing system. We may be interested in confidence intervals on the average wait time of incoming calls in the queue. Such confidence intervals would be crucial for drawing *credible* conclusions about such evaluations, since we know we cannot take the point prediction at face value given the sampling uncertainty in the data. We may, alternatively, be fitting a ranking algorithm by predicting user interaction from user-item features and then applying some fixed business rules on top, and we want to assess how often certain item categories would end up at the top. Of course, we would want to understand how certain we are in this assessment. Or, we fit a complex model to predict mean demand given price and user features from a price experiment we ran, and we use it to target discounts by optimizing demand at a price times unit profit. We may be interested in confidence intervals on the average profit over a given distribution of features.

All of these examples have two important features: they involve (1) a computationally burdensome step of fitting a large-scale model and (2) evaluating the result using a complicated function that need not even be known explicitly. Were both of these simple (simple model and simple function thereof), we could just use the classic delta method [7] (see next section for detail). However, when these aspects are complex and the model involves many parameters, it may be too prohibitive to either analytically derive the whole inverse Fisher information matrix in the many model parameters or compute and invert the Hessian of the training loss empirically as well as compute the gradient of the final evaluation as a function of all parameters [24]. Even one aspect being complex may pose a

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

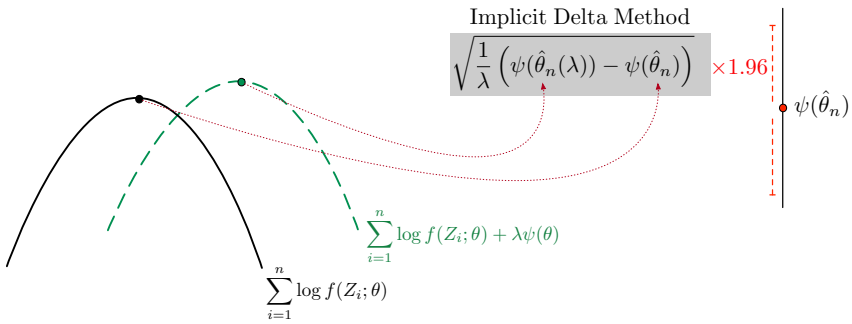


Figure 1: Illustration of how the implicit delta method (IDM) estimates 95% confidence intervals for a statistic of interest $\psi(\theta_0)$. Both the original MLE objective and the ψ-regularized objective are optimized, the ψ-evaluations of the two fitted models are compared, and 1.96 multiples of the square root of the difference in evaluations is added and subtracted from the nominal evaluation to create a 95% confidence interval.

serious challenge (*e.g.*, uncertainty quantification for the prediction of a complex model at a point). A remedy may be to bootstrap the whole process from data to final evaluation, but that can prove very computationally burdensome [9]. Usually just fitting the model once is already an expensive task; fitting it hundreds of times can be operationally infeasible. Other remedies, in the case of neural nets, may be the use of Langevin dynamics [32] or random dropout [12]. But these assess uncertainty in network weights and/or network predictions, which must then be translated to uncertainty in the final evaluation.

In this paper, we propose a direct yet inexpensive way to generically assess uncertainty in such settings. Specifically, we consider conducting inference when the estimator is some specified function of a (conditional) maximum likelihood estimator (MLE), such as a regression or classification model. Our proposal, the *implicit delta method*, works by simply adding an infinitesimal regularization to the MLE objective (*e.g.*, the sum of squared errors). We prove that the infinitesimal change in the final estimator due to this regularization is consistent for its asymptotic variance, the same variance that would have been predicted by the delta method in theory. Hence, the name of our method: we are conducting a delta-method quantification of uncertainty implicitly without explicitly propagating the uncertainty through the derivative of the evaluation function, analytically deriving the possibly-huge Fisher information matrix, or approximating it empirically. We prove that even when we approximate the infinitesimal change with a finite difference with constant width, the change we measure is still consistent for the asymptotic variance. This not only gives an assessment of uncertainty in terms of standard errors, it also permits us to construct calibrated confidence intervals. We demonstrate this in experiments, comparing to other popular approaches for uncertainty quantification, both Bayesian and frequentist.

2 Problem Set Up and the Delta Method

We consider an estimate constructed in two steps: first we fit a model using maximum likelihood estimation (MLE) and then apply some function to it. Namely, we consider data given by n independent and identically distributed (iid) observations $Z_i \in \mathcal{Z}$, $i = 1, \ldots, n$, drawn from a population with density $f(z;\theta_0)$ with respect to some base measure μ on $\mathcal{Z}$. For example, the data may consist of observations of features X and responses Y, with $Z = (X, Y)$.

In the first step, we fit a model to the data by MLE. Given a model $\{f(z;\theta) : \theta \in \Theta\}$ of densities (with respect to μ) parametrized by $\theta \in \Theta \subseteq \mathbb{R}^d$, we set

$$\text{Model fitting:} \quad \hat{\theta}_n \in \operatorname*{argmax}_{\theta\in\Theta} \sum_{i=1}^{n} \log f(Z_i;\theta). \tag{1}$$

A prominent example is generalized regression, where we observe features and responses $Z = (X, Y)$, have a parametrized predictor $h_\theta(x) \in \mathbb{R}^p$, and a parametric model $g(y;\vartheta)$ with $\vartheta \in \mathbb{R}^p$. We then set $f((x,y);\theta) = g(y;h_\theta(x))$.[1] Examples include least-squares regression, binary classification with

[1]Note since we are not interested in the distribution of X we here use only the conditional density of $Y \mid X$. Setting $f((x,y);\theta) = g(y;h_\theta(x))f(x)$ using the true unknown density $f(x)$ of X does not change the MLE

cross-entropy loss, and Poisson regression, all with possibly complex and nonlinear predictors (*e.g.*, neural nets).

In the second step, we process the trained model in some way to come up with our estimate. Given some $\psi : \Theta \mapsto \mathbb{R}$, we compute

$$\text{Evaluation using fitted model:} \quad \hat{\psi}_n = \psi(\hat{\theta}_n).$$

One example in the case of generalized regression is evaluation of the predictor at a specified point, $\psi(\theta) = g_\theta(x_0)$. Another example is, when $g_\theta(x, p)$ corresponds to predicted mean demand at price p given features x, we may be interested in the average optimal profit, $\psi(\theta) = \frac{1}{m}\sum_{j=1}^{m}\big(\sup_{p \geq c} g_\theta(x_j, p)(p - c)\big)$, for an evaluation dataset $\{x_j : j = 1, \ldots, m\}$. More generally, ψ could be more opaque: it could involve, for example, simulating a queuing system with a controller parametrized by θ, such as a priority policy with priority score $g_\theta(x)$.

We are interested in conducting uncertainty quantification for $\hat{\psi}_n$, and in particular in inference on its population limit, $\psi_0 = \psi(\theta_0)$. One way to do this inference is to propagate through ψ the uncertainty within $\hat{\theta}_n$ about θ_0, provided we understand the latter uncertainty. This is the so-called *delta method*.

To apply it, we must first understand the uncertainty in $\hat{\theta}_n$. Provided some regularity holds, this uncertainty can be characterized by the *curvature* of the objective function $\mathbb{P}\ell(\theta, \cdot)$ at $\theta = \theta_0$: if the curvature is sharp (resp., flat) then perturbing the objective and minimizing $\mathbb{P}_n\ell(\theta, \cdot)$ instead does not (resp., does) move the minimizer far away. This curvature is exactly the *Fisher information matrix*:

$$I(\theta) = -\int (\nabla^2 \log f(z; \theta)) f(z; \theta) d\mu(z).$$

Specifically, under appropriate regularity conditions,

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \rightsquigarrow \mathcal{N}(0,\, I^{-1}(\theta_0)). \tag{2}$$

In the above, $\mathcal{N}(\mu, \Sigma)$ refers to the multivariate normal distribution and $\rightsquigarrow$ refers to convergence in distribution. There are a variety of specific technical conditions that can establish this result. For an abstract presentation see theorems 9.27 and 9.28 in Wasserman [31]. For more rigorous treatments see theorem 13.2 of Wooldridge [34], theorem 3.3 of Newey and McFadden [23], theorem 5.1 of Lehmann and Casella [20], or theorem 8.3 of Davidson and MacKinnon [5], each of which uses slightly different technical regularity conditions.

Given Eq. (2) holds with $I(\theta_0) \succ 0$, the delta method would then guarantee that (see theorem 5.15 in 31)

$$\sqrt{n}(\hat{\psi}_n - \psi_0) \rightsquigarrow \mathcal{N}(0,\, V_0), \quad V_0 = \nabla\psi(\theta_0)^\top I^{-1}(\theta_0)\nabla\psi(\theta_0), \tag{3}$$

provided $\nabla\psi(\theta_0)$ exists and $V_0 > 0$.

An immediate and very important consequence of this is that we can construct calibrated confidence intervals for ψ_0: under Eq. (3),

$$\text{if} \quad n\hat{V}_n \to_p V_0, \quad \text{then} \quad \mathbb{P}\Big(\psi_0 \in \Big[\hat{\psi}_n \pm \Phi^{-1}((1+\beta)/2)\hat{V}_n^{1/2}\Big]\Big) \to \beta \quad \forall \beta \in (0, 1). \tag{4}$$

where Φ refers to the cumulative distribution function of the standard normal distribution and $\to_p$ refers to convergence in probability. For example, as long as $I(\theta)$ and $\nabla\psi(\theta)$ are continuous at θ_0, we can use

$$\hat{V}_n^{\text{DeltaMethod}} = \frac{1}{n}\nabla\psi(\hat{\theta}_n)^\top I^{-1}(\hat{\theta}_n)\nabla\psi(\hat{\theta}_n). \tag{5}$$

As discussed in the introduction, however, this approach may prove intractable in practice, especially when θ has many dimensions. Since we are only truly concerned with the uncertainty in $\hat{\psi}_n$ and not in $\hat{\theta}_n$, it may seem unnecessary and overly cumbersome to first compute the uncertainty in the latter and then propagate it. We next present our method, which does this all implicitly, never working directly with the vector θ except as an optimization variable in maximizing the MLE objective and a perturbation thereof.

nor any of the results compared to omitting $f(x)$ altogether as we do here, which is referred to as the *conditional* MLE by Wooldridge [34].

3 The Implicit Delta Method

We would like to construct calibrated confidence intervals as in Eq. (4), but computing the estimated standard error as in Eq. (5) can be prohibitive. The IDM is a way to compute the estimated standard error while *neither* explicitly computing the uncertainty in $\hat{\theta}_n$ *nor* propagating this uncertainty through ψ. Instead, we will simply slightly perturb the original MLE in Eq. (1) using a little bit of regularization, which will implicitly do both of these difficult tasks for us.

To define the IDM, we first define a regularized version of the MLE. Given any $\lambda \geq 0$, we consider adding the regularizer $\lambda\psi(\theta)$ to Eq. (1) as well as the corresponding final estimator after passing through ψ:

$$\hat{\theta}_n(\lambda;\psi) \in \operatorname*{argmax}_{\theta\in\Theta} \sum_{i=1}^{n} \log f(Z_i;\theta) + \lambda\psi(\theta), \quad \hat{\psi}_n(\lambda) = \psi(\hat{\theta}_n(\lambda;\psi)). \tag{6}$$

We refer to this as ψ-regularized MLE.

We then define the *infinitesimal* IDM (IIDM) as the infinitesimal change (*i.e.*, derivative) in our final estimate using ψ-regularized MLE as we infinitesimally increase λ from 0:

$$\hat{V}_n^{\mathrm{IIDM}} = \frac{\partial}{\partial\lambda}\hat{\psi}_n(\lambda)\bigg|_{\lambda=0} = \lim_{\lambda\to 0}\frac{1}{\lambda}\Big(\hat{\psi}_n(\lambda) - \hat{\psi}_n\Big). \tag{7}$$

Our first result shows that the IIDM estimate is consistent for the true asymptotic variance in Eq. (3).

Theorem 1. *Suppose that $\hat{\theta}_n \to_p \theta_0 \in \mathrm{Interior}(\Theta)$, $I(\theta_0) \succ 0$, and that, in a neighborhood of θ_0, $\psi(\theta)$ is continuously differentiable and $f(Z;\theta)$ is almost surely twice continuously differentiable in θ with a Hessian that is bounded in operator norm by an integrable function of Z. Then*

$$n\hat{V}_n^{\mathrm{IIDM}} \to_p V_0.$$

The significance of Theorem 1 is that, per Eq. (4), it implies that $\left[\hat{\psi}_n \pm \Phi^{-1}((1+\beta)/2)\sqrt{\hat{V}_n^{\mathrm{IIDM}}}\right]$ is a calibrated β-confidence interval for ψ_0.

Note that, aside from conditions on ψ (which are the same as needed for Eqs. (3) and (5) to work), the regularity conditions required in Theorem 1 are implied by the regularity conditions required for establishing Eq. (2) by, for example, any of Davidson and MacKinnon [5], Lehmann and Casella [20], Newey and McFadden [23], Wooldridge [34]. In that sense, these conditions are not strong as they are already needed for $\hat{V}_n^{\mathrm{DeltaMethod}}$ to be a good estimate of uncertainty to begin with, and they fit into the existing framework for the asymptotic analysis of MLE.

The implication of Theorem 1 is that we may be able to implicitly complete the steps of the delta method (compute the uncertainty in $\hat{\theta}_n$, then propagate it through ψ) by simply assessing the impact of regularizing the MLE. However, this requires we actually differentiate with respect to the regularization coefficient. While this requires computing just one first derivative (rather than many first and second derivatives as in Eq. (5)), it is still not clear how to do this in practice.

In practice, we might approximate this derivative using finite differences, *i.e.*, replace the limit in Eq. (7) with a very small λ. This gives rise to what we call the finite-difference IDM (FDIDM), defined as follows for a given $\lambda_n > 0$:

$$\hat{V}_n^{\mathrm{FDIDM}} = \frac{1}{\lambda_n}\Big(\hat{\psi}_n(\lambda_n) - \hat{\psi}_n\Big). \tag{8}$$

Our next result shows that it in fact suffices to choose λ_n constant. In fact any choice of λ_n growing strictly slower than n, yields that $n\hat{V}_n^{\mathrm{FDIDM}}$ is also consistent for V_0, just like $\hat{V}_n^{\mathrm{IIDM}}$, provided just slightly more regularity holds.

Theorem 2. *Fix any $\lambda_n = o(n)$. Suppose that in addition to the assumptions of Theorem 1, in a neighborhood of θ_0, $\psi(\theta)$ is thrice continuously differentiable and $f(Z;\theta)$ is almost surely thrice continuously differentiable in θ with a third-order derivative that is bounded in operator norm by an integrable function of Z. Then*

$$n\hat{V}_n^{\mathrm{FDIDM}} \to_p V_0.$$

It may seem surprising that a constant λ_n suffices or that λ_n is even allowed to grow, but that can be seen as an artifact of the fact we did not normalize the sum over the data in Eq. (6) by $1/n$. If we did normalize, it would be equivalent to rescaling λ by n, so that $o(n)$ becomes $o(1)$, *i.e.*, requiring a vanishing increment for the finite differencing. Nonetheless, writing Eq. (6) as we did is very convenient, as it matches how one usually applies optimization algorithms such as stochastic gradient descent to training objectives, and it makes the choice of λ_n for Eq. (8) very easy: just fix some constant and do not worry about the scaling with n. For example, setting $\lambda_n = 1$ suggests a very simple-looking 95%-confidence interval: $\left[\hat{\psi}_n \pm 1.96\sqrt{\hat{\psi}_n(1) - \hat{\psi}_n}\right]$.

Remark 1 (Regression Using Squared Error Loss)**.** When training regression models we usually minimize over model parameters (*e.g.*, neural net weights) the sum over the data of squared error loss, $\ell((x, y); \theta) = (y - g_\theta(x))^2$. This differs from the corresponding Gaussian log likelihood by a factor of $-\frac{1}{2\sigma^2}$ (and some constants that do not matter), where σ^2 is the residual variance of Y given X. Therefore, to apply IDM, all we should do is simply regularize the sum-of-squared-errors *minimization* problem by $-2\sigma^2\lambda\psi(\theta)$, as that would be equivalent to dividing the log likelihood part by $-2\sigma^2$. Of course, we do not know σ^2, but we can estimate it by $\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^n (y - g_{\hat{\theta}_n}(x))^2$, that is, the minimum average sum of squared errors. Since $\hat{\sigma}_n^2 \to_p \sigma^2$, as it is in fact the MLE estimate for σ^2, the asymptotic guarantees of Theorems 1 and 2 will continue to hold after this rescaling.

Remark 2 (Using IDM to Compute the Fisher Information)**.** A by-product of the proof of Theorem 1 is that, if we looked at the (vector-valued) derivative $\hat{W}_n = \frac{\partial}{\partial\lambda}\hat{\theta}_n(\lambda; \psi)\big|_{\lambda=0} = \lim_{\lambda\to 0} \frac{1}{\lambda}\left(\hat{\theta}_n(\lambda; \psi) - \hat{\theta}_n\right)$, then $n\hat{W}_n \to_p I(\theta_0)^{-1}\nabla\psi(\theta_0)$. Therefore, if we set $\psi(\theta) = \theta_i$, *i.e.*, the ith component of θ, then $\hat{W}_n$ converges to the ith column of $I(\theta_0)^{-1}$. Thus, by regularizing each component of θ in turn, we obtain the whole matrix.

Nonetheless, the whole raison d'être of IDM is to avoid working directly with the parameter vector θ altogether, and simply propagate its uncertainty automatically via the MLE optimization problem. For example, if we consider neural net regression, IDM would never make explicit reference to the vector of weights itself, only to the trained prediction model and its prediction performance on data. The above, wherein we compute the uncertainty in θ directly, stands in contradiction to this. Nonetheless, it can be a useful observation when inference on θ itself is for some reason of interest.

3.1 Extension to Multivariate Evaluations

We have so far focused on scalar evaluations for ease of presentation and as it covers the most important cases. We now show how our method easily extends to the multivariate case, where $\psi(\theta) = (\psi^{(1)}(\theta), \ldots, \psi^{(K)}(\theta)) \in \mathbb{R}^K$. The reason it may not suffice to run IDM separately for each component is that we may be interested in the *covariance* of the evaluations. Under the appropriate conditions, the extension of the delta method for MLE (Eq. (3)) to multivariate evaluations is

$$\sqrt{n}(\hat{\psi}_n - \psi_0) \rightsquigarrow \mathcal{N}(0, V_0), \quad V_0 = J(\theta_0)^\top I^{-1}(\theta_0) J(\theta_0), \tag{9}$$

where $J_{ij}(\theta) = \frac{\partial}{\partial\theta_i}\psi^{(j)}(\theta)$ is the $K \times d$ Jacobian of $\psi(\theta)$.

Our extensions of IIDM and FDIDM to multivariate evaluations are as follows:

$$\Delta_{ij}(\lambda) = \frac{1}{\lambda}\left(\psi^{(i)}(\hat{\theta}_n(\lambda; \psi^{(j)})) - \psi^{(i)}(\hat{\theta}_n)\right), \ (\hat{V}_n^{\text{IIDM}})_{ij} = \lim_{\lambda\to 0} \Delta_{ij}(\lambda), \ (\hat{V}_n^{\text{FDIDM}})_{ij} = \Delta_{ij}(\lambda_n).$$

Theorem 3. *$n\hat{V}_n^{\text{IIDM}} \to_p V_0$ under the conditions of Theorem 1, and $n\hat{V}_n^{\text{FDIDM}} \to_p V_0$ under the conditions of Theorem 2, both as $K \times K$ matrices.*

Surprisingly, this shows one need only solve $K + 1$ (possibly) regularized MLEs to get the full $K \times K$ covariance. (See Alg. 3 in supplement.)

3.2 Handling Non-differentiable Evaluations and Evaluation Uncertainty

So far we have assumed that the evaluation function is a known and differentiable function. Both statements may be false when we are interested in evaluating average performance on a population but we only have a finite evaluation data set and unit performance is not differentiable.

Specifically, let $W_1, \ldots, W_m \sim W$ denote the evaluation data set (which may be the same as the training set or otherwise dependent or it may be an independent data set) and let $h(w; \theta)$ the unit

evaluation function. Consider the empirical evaluation map

$$\psi(\theta) = \frac{1}{m}\sum_{j=1}^{m} h(W_i;\theta).$$

If $h(W;\theta)$ is almost surely not differentiable in θ, then ψ is also almost surely not differentiable, which poses a challenge. We will show, however, that even though ψ is not differentiable (which would break the usual delta method), FDIDM actually remains valid, *without any changes to the method*, provided certain on-average-differentiability holds.

To motivate the challenge of nondifferentiability and the plausibility of on-average-differentiability, consider an example where $g_\theta(x)$ represents an order quantity to stock in context x and $w = (x, d)$ represents features and demand. If $h((x,d);\theta) = \max\{d - g_\theta(x), 0\}$ then $\psi(\theta)$ quantifies average unmet demand, but h is not differentiable. Other non-differentiable examples include evaluating regression and classification models' performance using non-differentiable utility functions. While h may not be differentiable and hence neither ψ, it may still be plausible that its expectation $\mathbb{E}[h(W;\theta)] = \mathbb{E}[\psi(\theta)]$ is differentiable. For example, if the distribution of demand conditioned on features is continuous, then the derivative of $\mathbb{E}[h(W;\theta)]$ in the example of average unmet demand will be the average of the conditional cumulative distribution function at $g_\theta(x)$ times $\nabla_\theta g_\theta(x)$, and the second derivative will involve the conditional density.

We next show FDIDM *still* works with non-differentiable ψ, given some on-average-differentiability.

Theorem 4. *Consider either $m \asymp n$ or $m \gg n$. Fix $\lambda_n = \lambda > 0$ constant. Suppose the assumptions of Theorem 2 hold, that Eq.* (2) *holds, and that $\mathbb{E}[h(W;\theta)]$ is twice continuously differentiable at θ_0. Suppose further that, for some functions $L(w)$ and $\dot{h}_0(w)$ with $\mathbb{E}[L^2(W)] < \infty$, we have that in a neighborhood of θ_0,*

$$|h(W;\theta_1) - h(W;\theta_2)| \le L(W)\|\theta_1 - \theta_2\| \quad \textit{almost surely},$$
$$\mathbb{E}[(h(W;\theta_1) - h(W;\theta_0) - \dot{h}_0(W)^\top(\theta_1 - \theta_0))^2] = o(\|\theta_1 - \theta_0\|^2).$$

Then

$$(\hat{V}_n^{\mathrm{FDIDM}})^{-1/2}(\hat{\psi}_n - \psi_0) \rightsquigarrow \mathcal{N}(0,1).$$

In the above example, $L(w)$ would be the Lipschitz constant of $g_\theta(x)$ in θ and $\dot{h}_0(w)$ would be the cumulative distribution function of demand at $g_{\theta_0}(X)$.

Although the asymptotic variance of $\hat{\psi}_n - \psi_0$ is now different (in particular both V_0 in Eq. (3) and $\hat{V}_n^{\mathrm{DeltaMethod}}$ in Eq. (5) are ill-defined), Theorem 4 shows that $\hat{V}_n^{\mathrm{FDIDM}}$ actually remains consistent for this new asymptotic variance. Thus, it provides a consistent estimate of standard errors and it still gives calibrated confidence intervals (note ψ_0 is now *random* but we can still have a confidence interval for it).

In some cases we may want to directly conduct inference on the population version of the evaluation, $\psi_0^* = \mathbb{E}[h(W;\theta_0)]$. To do this, all we have to do is simply also add the uncertainty due to finite evaluation data set. Under standard regularity conditions, we have

$$(\hat{V}_m^{\psi})^{-1/2}(\psi_0 - \psi_0^*) \rightsquigarrow \mathcal{N}(0,1), \quad \text{where} \quad \hat{V}_m^{\psi} = \frac{1}{(m-1)m}\sum_{j=1}^{m}(h(w;\hat{\theta}_n) - \psi(\hat{\theta}_n))^2.$$

Therefore, provided the training and evaluation data sets are independent,

$$(\hat{V}_n^{\mathrm{FDIDM}} + \hat{V}_m^{\psi})^{-1/2}(\psi(\hat{\theta}_n) - \psi^*(\theta_0)) \rightsquigarrow \mathcal{N}(0,1).$$

If not independent, then $\sqrt{\hat{V}_n^{\mathrm{FDIDM}}} + \sqrt{\hat{V}_m^{\psi}}$ provides a consistent upper bound on the standard error.

3.3 Implementation

FDIDM is given in pseudocode in Alg. 1. Given an objective function $\mathcal{L} := \sum_i^n \log f(Z_i;\theta)$, evaluation function ψ, and scalar width λ, FDIDM returns the estimated variance of $\psi(\hat{\theta}_n)$. The first step is to maximize the original objective w.r.t. θ. Usually, this task has already been solved as this is the trained predictive model. Then, maximize the ψ-regularized objective w.r.t. θ. Finally, return the estimated variance using the finite-difference method evaluated at $\lambda = 0$. See Appendix D for the corresponding algorithm when ψ is multivariate.

In practice, one can further reduce the computational cost of FDIDM due to the fact that the ψ-regularized objective can be made arbitrarily close to original objective by choosing λ small enough.

Algorithm 1: Finite-difference implicit delta method (FDIDM)

Input: Learning objective $\mathcal{L}$, evaluation ψ, scalar λ

Function `FDIDM(`$\mathcal{L}$, ψ, λ`):`

$\hat{\theta}_n \leftarrow \arg\max_\theta \mathcal{L}(\theta)$ `// optimize learning objective`

$\hat{\theta}_n(\lambda) \leftarrow \arg\max_\theta \mathcal{L}(\theta) + \lambda\psi(\theta)$ `// optimize` ψ`-regularized objective`

return $\frac{1}{\lambda}(\psi(\hat{\theta}_n(\lambda)) - \psi(\hat{\theta}_n))$ `// estimated variance of` $\psi(\hat{\theta}_n)$

Specifically, when using stochastic gradient ascent in FDIDM, once the optimum $\hat{\theta}_n$ has been found, only a small number of gradient updates may be required to also find $\hat{\theta}_n(\lambda)$.

FDIDM also admits non-gradient-based approaches. Consider the case that ψ is a simulator that takes a fitted model and returns a set of evaluations and no gradient. Then the ψ-regularized objective may optimized by gradient-free methods such as Nelder-Mead [22] and Bayesian optimization [11].

4 Alternatives for Uncertainty Quantification and Related Work

Uncertainty quantification in machine learning is a topic of major interest due to the need to make downstream inferences and decisions based on the predictions of large-scale networks trained on massive datasets in either a frequentist [13, 24, 27] or Bayesian fashion [2, 6, 12]. An exhaustive account of the literature is outside the scope of this paper. We highlight the principal ideas and points of contact with our work.

The Bootstrap The bootstrap simulates sampling from the true data generating distribution by resampling from the observed dataset (see, *e.g.*, [9] for an introduction and [18] for theory on when it works). The key advantage is that it enables general-purpose and easy-to-implement uncertainty quantification for estimators. It comes at a high computational burden because the estimator, which may comprise a model-fitting algorithm and prediction, needs to be executed many times. In the context of deep learning, many useful adaptations of the bootstrap and the related jackknife have been proposed to increase its computational efficiency [13, 26, 27]. Maintaining an ensemble of models as a representation of the variability of the evaluation is an appealing intuition that does not restrict one to local approximations, and may be combined with local approximations where necessary.

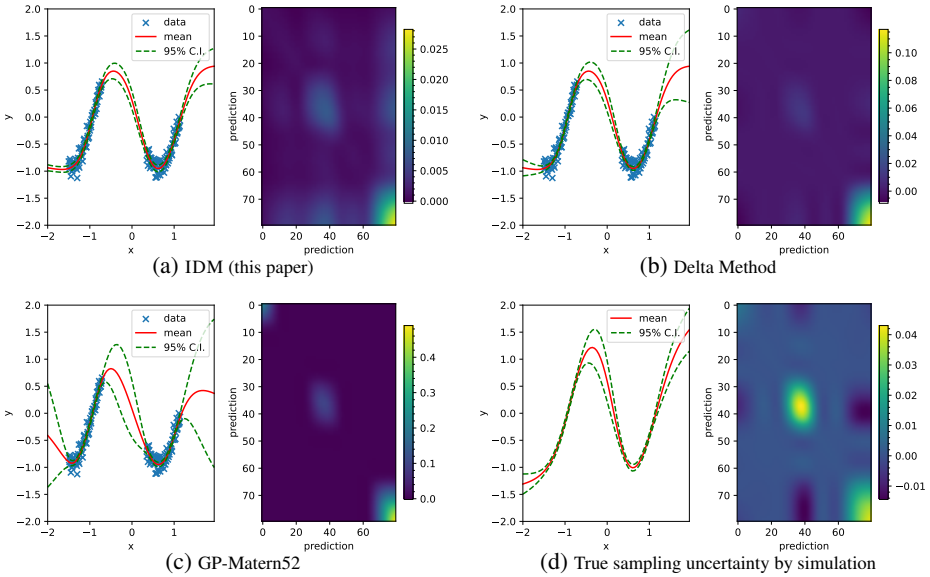
The Functional Delta Method The delta method [7] is a classic approach that is widely used with small models with an analytic Fisher information matrix (*e.g.*, linear regression) and, more recently, auto-differentiation unlocks the delta method for a larger class of models [24]. The bottleneck is the need to calculate then invert the Fisher information matrix, for which there are various approximations [21, 28]. The functional delta method extends the delta method to evaluations of infinite-dimensional parameters (see Ch. 12 [18]) but is usually restricted to analytically deriving influence functions in theory by differentiating the population estimand with respect to distributions and then approximating the influence function by plugging in estimates of unknown nuisances [3, 14].

Bayesian Uncertainty Quantification Tractable methods for approximate Bayesian inference in neural networks, such as variational inference in feed-forward nets [2], autoencoders [17, 30], normalizing flows [29], dropout uncertainty [12], stochastic gradient Langevin dynamics [32] and related approaches, present an impressive range of options for uncertainty quantification. In cases where it is sufficient to only consider a single mode in the posterior, local methods can prove useful. In particular, an alternative interpretation of the delta method is as a special case of the Laplace approximation to Bayesian inference, where Eq. (5) arises in the posterior predictive distribution for a local multivariate Gaussian approximation around the maximum *a posteriori* estimate. Several recent works have investigated the potential of the Laplace approximation as a way to avoid having to characterizing the full posterior in deep networks [6, 15, 16]. IDM can provide another way to perform a Laplace approximation and may be orthogonally combined with the above methods.

5 Experiments

In this section, we evaluate finite-difference implicit delta method (FDIDM) on a range of tasks that require confidence intervals.[2] Our goal is to quantify the extent to which FDIDM applies in practice

[2] The source code for the experiments is available at `<ANONYMIZED>` [see supplement].



(a) IDM (this paper) (b) Delta Method

(c) GP-Matern52 (d) True sampling uncertainty by simulation

Figure 2: Fits along with uncertainty bounds and estimated prediction-covariance matrix for data generated from $y = -\sin(3x - \frac{3}{10}) + \frac{1}{10}\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$

and how it compares to alternative methods. We start with 1D synthetic data in Sec. 5.1 where we apply a neural net to recover known functions from small datasets. Then, in Sec. 5.2, we consider the task of inferring average utility under a neural net trained on a set of real-world benchmark datasets. Finally, in Sec. 5.3, we apply FDIDM to variational autoencoders and use the implicit delta perspective to understand the effect of KL down-weighting.

5.1 1D Synthetic Examples

We consider known quadratic and sinusoidal functions from which we draw a random dataset. Fig 2 gives the data generating stochastic function for a $\sin$ wave and the resulting fits for FDIDM, the classic delta method, a Gaussian process (GP) with Matern-52 kernel, as well as simulation from the true data generating function. (Appendices B and C provide the results on the quadratic function and further experimental details, respectively.) The quadratic example has evenly dispersed input data and there is close alignment between the methods. The $\sin$ wave is more challenging because it requires extrapolation – also known as "in-between" uncertainty in [10] – from outside the ranges of given inputs. As expected, IDM agrees most with the delta method while the GP overestimates uncertainty, particularly for extrapolation at the outer edges. However, it should be noted that the GP kernel[3] gives a different mean to IDM, DM, and simulation which are all using a neural net with 1 hidden layer of 50 `tanh` units.[4] The corresponding full covariance matrix of the predictions is also given in Fig. 2. All the methods recover the high level structure of covariance for both examples, though the scale factors differ considerably, especially for the $\sin$ wave.

5.2 Confidence in Predicted Cost Downstream of Classification

A set of classification tasks are solved with a neural net with one hidden layer and 50 `tanh` hidden units. In this setting, we wish to calculate confidence intervals over total cost in a downstream task under predictions from the network. An arbitrary cost function is set up, in this case, the average cross entropy of the observations on a held-out validation dataset, though in practice we could have a wide variety of cost functions relating to the task downstream of the classifier. It is challenging to form a confidence interval for even this simple cost function because it is a function of predictions from the network. Under this scenario, it is typical to make a bootstrapping estimate, requiring B times the cost of training the network (here, we use $B = 50$). FDIDM is also applicable in this setting. We show both methods on MNIST image classification [19] and a set of UCI benchmark datasets [8] in Fig. 3. We find that FDIDM has good coverage for a fraction of the cost of the bootstrap estimate.

[3]We also applied the $\arccos$ kernel in a GP which imitates a neural network [4] but found that the Matern-52 kernel inferred a mean that was closer in practice to the mean inferred by the neural net.

[4]This architecture is in line with [10], which also provided the basis for our $\sin$ example.

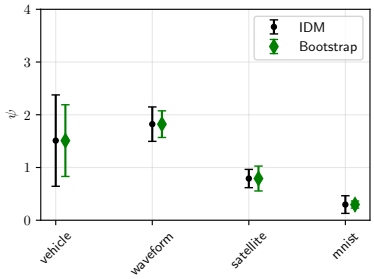


Figure 3: Predicted distribution of utility in classification on benchmark datasets. Datasets are shown in order of number of examples.

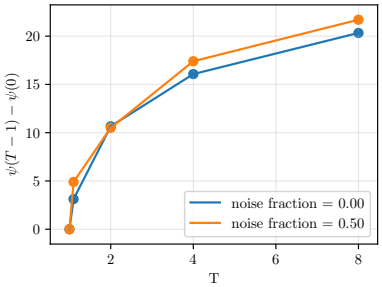


Figure 4: Improvement in reconstruction quality over the unweighted likelihood as a function of T.

5.3 Down-Weighting KL in Variational Autoencoders

The variational autoencoder (VAE) is a prominent example of approximate inference in deep generative models [17, 30]. In practice, it has been observed that down-weighting the KL term in the variational objective by a factor $\frac{1}{T}$, where $T > 1$, results in significantly better accuracy on held-out test data [1, 35]. This is closely related to other ways of reweighting the prior and likelihood terms in approximate inference such as data augmentation [25] and the cold posterior effect [33]. Various explanations for the benefit of setting $T > 1$ have been posited, *e.g.*, model mispecification. Here, we briefly evaluate a precise IDM-based interpretation of this phenomenon.

Observing that any objective does not change its critical points under (non-zero) rescaling, it holds that down-weighting the KL term yields the same optimization problem as up-weighting the reconstruction error by T. We consider this setting and define ψ to be the negative reconstruction error under a VAE. Rearranging the terms in Eq. (8), letting $\lambda_n := T - 1$, we have

$$\hat{\psi}_n(\lambda_n) = (T-1)\hat{V}_n^{\mathrm{FDIDM}} + \hat{\psi}_n(0). \tag{10}$$

The two immediate implications of Eq. (10) for fixed $T > 1$ of $o(n)$ are that,

1. the reconstruction quality for the objective implied by variational Bayes is upper bounded by that of the objective with up-weighted likelihood; and,
2. the higher the variance in the reconstruction quality, as determined by the dataset and model, the greater the benefit of up-weighting the likelihood term in variational Bayes.

Since (1) is already supported by existing empirical work, we focus on evaluating whether (2) also holds in practice. To do this, we artificially increase the variance of the reconstruction quality by perturbing a proportion of the dataset and compare $\hat{\psi}_n$ for various T. Fig. 4 shows the results for applying a VAE to the MNIST dataset after adding i.i.d. noise $\epsilon \sim \mathrm{Uniform}(0, \frac{1}{20})$ to each pixel in a randomly chosen fraction of the images. The figure indicates that, as predicted by Eq. (10), the gap in the reconstruction quality for $T > 1$ relative to $T = 1$ increases as more variance is introduced. Higher values of T do indeed result in better reconstruction quality and this advantage grows with the amount of variance. In sum, these findings are consistent with the hypothesis that the advantage from KL down-weighting may be explained as the residual between the ψ-regularized variational objective and the objective implied by the evidence lower bound, though further studies are required.

6 Discussion, Limitation, and Conclusions

In this paper we develop the implicit delta method for forming calibrated confidence intervals via a careful regularization of the model objective. Like the delta method, the method requires the MLE to be at a stable optimum, where perturbations around the optimum reliably capture sampling uncertainty. If this is not the case, *e.g.*, the parameter has failed to converge or the objective itself is changing, it yields unreliable results. For this reason, IDM – like the delta method and the bootstrap – may be misleading for small data, and indeed uncertainty quantification with small data is fundamentally difficult. The most appealing feature of IDM is that it does not require the variance of the parameters to be made explicit, which also suggests future research in exploring the compatibility of nonparametric models with IDM. There is also the potential to explore IDM in constrained MLE.

References

[1] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International conference on learning representations*, 2020.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[3] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

[4] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

[5] Russell Davidson and James G MacKinnon. *Estimation and inference in econometrics*, volume 63. Oxford New York, 1993.

[6] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless Bayesian deep learning. *Advances in neural information processing systems*, 34, 2021.

[7] J. L. Doob. The limiting distributions of certain statistics. *The annals of mathematical statistics*, 6(3):160 – 169, 1935. doi: 10.1214/aoms/1177732594.

[8] Dheeru Dua and Casey Graff. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of Information and Computer Science*, 25:27, 2019.

[9] Bradley Efron and Trevor Hastie. *Computer Age Statistical Inference*, volume 6. Cambridge University Press, 2021.

[10] Andrew YK Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. 'in-between' uncertainty in Bayesian neural networks. *arXiv preprint arXiv:1906.11537*, 2019.

[11] Peter I Frazier. Bayesian optimization. In *Recent advances in optimization and modeling of contemporary problems*, pages 255–278. Informs, 2018.

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, pages 1050–1059. PMLR, 2016.

[13] Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A Swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019.

[14] Hidehiko Ichimura and Whitney K Newey. The influence function of semiparametric estimators. *Quantitative Economics*, 13(1):29–61, 2022.

[15] Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th international conference on artificial intelligence and statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR, 13–15 Apr 2021.

[16] Mohammad Emtiyaz Khan, Alexander Immer, Ehsan Abedi, and Maciej Korzepa. Approximate inference turns deep networks into Gaussian processes. In *Advances in neural information processing systems*, volume 32. Curran Associates, Inc., 2019.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Michael R Kosorok. *Introduction to empirical processes and semiparametric inference.* Springer, 2008.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

[21] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.

[22] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[23] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[24] Geir K Nilsen, Antonella Z Munthe-Kaas, Hans J Skaug, and Morten Brun. Epistemic uncertainty quantification in deep learning classification by the delta method. *Neural networks*, 145: 164–176, 2022.

[25] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with Bayesian principles. *Advances in neural information processing systems*, 32, 2019.

[26] Ian Osband and Benjamin Van Roy. Bootstrapped Thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

[27] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, 29:4026–4034, 2016.

[28] Barak A Pearlmutter. Fast exact multiplication by the Hessian. *Neural computation*, 6(1): 147–160, 1994.

[29] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[30] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[31] Larry Wasserman. *All of Statistics*. Springer, 2004.

[32] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning*, pages 681–688. Citeseer, 2011.

[33] Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

[34] Jeffrey M Wooldridge. *Econometric analysis of cross section and panel data*. MIT press, 2010.

[35] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International conference on machine learning*, pages 5852–5861. PMLR, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Proof of consistency in Sec. 3; connections discussed in Sec. 4; evaluation in Sec. 5
   (b) Did you describe the limitations of your work? [Yes] See Sec. 6

(c) Did you discuss any potential negative societal impacts of your work? [N/A] To the best of the authors' knowledge, our work does not have negative social impacts

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See Sec. 3 and Appendix A

(b) Did you include complete proofs of all theoretical results? [Yes] See Sec. 3 and Appendix A

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In supplement; will also be posted on github outside of the anonymous review process

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 5, Appendix B and C, and source code

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Source code

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

Supplementary Materials

Outline of Supplementary Materials

- Proofs of results stated in the main text are provided in Appendix A.
- Additional experimental results, including coverage plots, are provided in Appendix B.
- Additional details for all experiments, including specifics on the implementation of the optimization, are provided in Appendix C.
- The finite-difference IDM algorithm for multivariate-valued evaluations are detailed in Appendix D.
- Replication code is provided in an attached zip file.

A Omitted Proofs

A.1 Proof of Theorem 1

Proof. Let $\theta_0 \in \mathcal{N} \subset \Theta$ be the interior neighborhood where the assumed differentiable hold. Since $\hat{\theta}_n \to_p \theta_0$, we have that $\mathbb{P}\Big(\hat{\theta}_n \in \mathcal{N}\Big) \to 1$.

Define $L_n(\theta) = \sum_{i=1}^n \log f(Z_i;\theta)$ on $\mathcal{N}$. By assumptions of continuity and existence of envelope, lemma 4.3 of Newey and McFadden [4] guarantees that $-\frac{1}{n}\nabla^2 L_n(\hat{\theta}_n) \to_p I(\theta_0)$. Since $I(\theta_0) \succ 0$, we have by continuous mapping that $\mathbb{P}\Big(\nabla^2 L_n(\hat{\theta}_n) \prec 0\Big) \to 1$.

Consider the event $\hat{\theta}_n \in \mathcal{N}$ and $\nabla^2 L_n(\hat{\theta}_n) \prec 0$, which has probability tending to 1. Define $\mathcal{L}(\lambda,\theta) = \nabla L_n(\theta) + \lambda \nabla\psi(\theta)$ on $\mathbb{R}\times\mathcal{N}$. Note that $\mathcal{L}(0,\hat{\theta}_n) = 0$ and that the Jacobian of $\mathcal{L}$ with respect to θ at $(0,\hat{\theta}_n)$ is $\nabla^2 L_n(\hat{\theta}_n) \prec 0$. Therefore, there exists $\tilde{\theta}_n(\lambda)$ with $\mathcal{L}(\lambda,\tilde{\theta}_n(\lambda)) = 0$ and $\frac{\partial\tilde{\theta}_n}{\partial\lambda}(\lambda) = -(\nabla^2 L_n(\hat{\theta}_n(\lambda;\psi)))^{-1}\nabla\psi(\hat{\theta}_n(\lambda;\psi))$ for $\lambda \in (-u,u)$ for some $u > 0$. Moreover, since $\nabla^2 L_n(\hat{\theta}_n) \prec 0$ and by continuous derivatives, there is a $0 < u' \le u$ such that $\tilde{\theta}_n(\lambda)$ uniquely solves $\mathcal{L}(\lambda,\theta) = 0$ for $\lambda \in (-u',u')$ and therefore $\tilde{\theta}_n(\lambda) = \hat{\theta}_n(\lambda;\psi)$ for such λ. Finally, chain rule applied to $\psi(\hat{\theta}_n(\lambda;\psi))$ for $\lambda \in (-u',u')$ gives $\frac{\partial\hat{\psi}_n}{\partial\lambda}(\lambda) = \nabla\psi(\hat{\theta}_n(\lambda;\psi))^\top \frac{\partial\hat{\theta}_n}{\partial\lambda}(\lambda)$.

Applying this at $\lambda = 0$, we conclude that with probability tending to 1,

$$n\hat{V}_n^{\mathrm{IIDM}} = \nabla\psi(\hat{\theta}_n)^\top \left(-\frac{1}{n}\sum_{i=1}^n \nabla^2 \log f(Z_i;\hat{\theta}_n)\right)^{-1} \nabla\psi(\hat{\theta}_n).$$

By continuity, $\nabla\psi(\hat{\theta}_n) \to_p \nabla\psi(\theta_0)$. We have also already shown that $-\frac{1}{n}\nabla^2 L_n(\hat{\theta}_n) \to_p I(\theta_0) \succ 0$. Therefore, the proof is completed by continuous mapping. □

A.2 Proof of Theorem 2

Proof. The assumptions imply that $\hat{\psi}_n''(0)$ exists and $n^2\hat{\psi}_n''(0)$ converges to a constant in probability. Applying Taylor's theorem using the Lagrange form of the remainder, we have that, for some random $\tilde{\lambda}_n \in [0,\lambda_n]$,

$$\hat{\psi}_n(\lambda_n) = \hat{\psi}_n(0) + \hat{\psi}_n'(0)\lambda_n + \hat{\psi}_n''(0)\tilde{\lambda}_n^2.$$

Rearranging yields

$$n\hat{V}_n^{\mathrm{FDIDM}} = n\hat{V}_n^{\mathrm{IIDM}} + n\hat{\psi}_n''(0)\tilde{\lambda}_n^2/\lambda_n.$$

By Theorem 1, $n\hat{V}_n^{\mathrm{IIDM}} \to_p V_0$.

Considering the remainder term, we have

$$\left|n\hat{\psi}_n''(0)\tilde{\lambda}_n^2/\lambda_n\right| \le \frac{\lambda_n}{n}\left|n^2\hat{\psi}_n''(0)\right| = o(1)O_p(1) = o_p(1),$$

completing the proof. □

A.3 Proof of Theorem 3

Proof. From the proof of Eq. (7), we know that, with probability tending to 1, $\frac{\partial}{\partial\lambda}\hat{\theta}_n(\lambda;\psi^{(j)}) = -(\nabla^2 L_n(\hat{\theta}_n(\lambda;\psi^{(j)})))^{-1}\nabla\psi^{(j)}(\hat{\theta}_n(\lambda;\psi^{(j)}))$ (from the implicit function argument) and $\frac{\partial}{\partial\lambda}\psi^{(i)}(\hat{\theta}_n(\lambda;\psi^{(j)})) = \nabla\psi^{(i)}(\hat{\theta}_n(\lambda;\psi^{(j)}))^\top\frac{\partial}{\partial\lambda}\hat{\theta}_n(\lambda;\psi^{(j)})$ (from chain rule argument). Combining and applying for each entry i, j yields the first statement. The second statement follows exactly as in Theorem 2 by a Taylor expansion. □

A.4 Proof of Theorem 4

Proof. Define $\psi^*(\theta) = \mathbb{E}[h(W;\theta)]$. Following example 3.2.22 of van der Vaart and Wellner [5], our conditions imply that

$$\psi(\hat{\theta}_n) = \psi(\theta_0) + \nabla\psi^*(\theta_0)^\top(\hat{\theta}_n - \theta_0) + o_p(1/\sqrt{n}).$$

Therefore,

$$\sqrt{n}(\psi(\hat{\theta}_n) - \psi(\theta_0)) \rightsquigarrow \mathcal{N}(0,\, \nabla\psi^*(\theta_0)^\top I^{-1}\nabla\psi^*(\theta_0)).$$

Then if, hypothetically, we apply FDIDM with the unknown $\psi^*(\theta)$ to obtain $\tilde{V}_n^{\mathrm{FDIDM}} = (\psi^*(\hat{\theta}_n(\lambda;\psi^*)) - \psi^*(\hat{\theta}_n))/\lambda$, then, by Theorem 2, we have that

$$\frac{\psi(\hat{\theta}_n) - \psi(\theta_0)}{\sqrt{\tilde{V}_n^{\mathrm{FDIDM}}}} \rightsquigarrow \mathcal{N}(0,1).$$

To complete the proof, we next argue that $\hat{V}_n^{\mathrm{FDIDM}} - \tilde{V}_n^{\mathrm{FDIDM}} \to_p 0$, that is, the difference vanishes between doing hypothetical FDIDM with the unknown ψ^* and actual FDIDM with the known ψ. Let $\theta_0 \in \mathcal{N} \subset \mathrm{Interior}(\Theta)$ denote the interior neighborhood where all desirable differentiabilities hold. Our assumptions yield $\mathbb{P}(\hat{\theta}_n \in \mathcal{N}, \hat{\theta}_n(\lambda;\psi) \in \mathcal{N}, \hat{\theta}_n(\lambda;\psi^*) \in \mathcal{N}) \to 1$ and $\sup_{\theta\in\mathcal{N}}|\psi(\theta) - \psi^*(\theta)| \to_p 0$ per lemma 4.3 of Newey and McFadden [4]. Then, on the event that $\hat{\theta}_n \in \mathcal{N}, \hat{\theta}_n(\lambda;\psi) \in \mathcal{N}, \hat{\theta}_n(\lambda;\psi^*) \in \mathcal{N}$, we have

$$\begin{aligned}\left|\hat{V}_n^{\mathrm{FDIDM}} - \tilde{V}_n^{\mathrm{FDIDM}}\right| &= \lambda^{-1}\left|\psi(\hat{\theta}_n(\lambda;\psi)) - \psi(\hat{\theta}_n) - \psi^*(\hat{\theta}_n(\lambda;\psi^*)) + \psi^*(\hat{\theta}_n)\right| \\ &\le 2\lambda^{-1}\sup_{\theta\in\mathcal{N}}|\psi(\theta) - \psi^*(\theta)|,\end{aligned}$$

yielding the statement. □

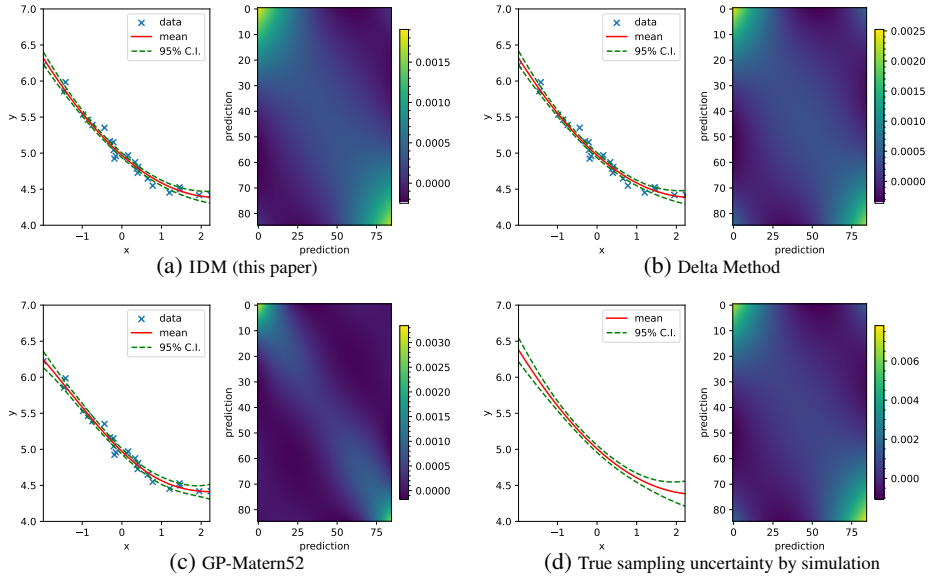
B Further Experiments

Fig. 5 shows the fitted mean and covariance on a single draw of the quadratic dataset. With the inputs more dispersed (than the sin example) we see that all methods are in closer agreement. Fig. 6 gives the 95% coverage of the methods as a function of input. The quadratic function has close to 95% coverage across the inputs for all methods. The sin example restricts the set of available inputs (even on resample) and uses a neural network on a small dataset (N=160), both of which make it particularly challenging for coverage. There is some agreement among the methods that use the neural network (IDM and delta method). The GP-Matern52 comprises a different model of the mean and covariance and is included for reference. All of the methods are at least 20 percentage points away from the true coverage across the inputs.

C Details for Experiments

In this section we provide details for the experimental setup used in the paper.

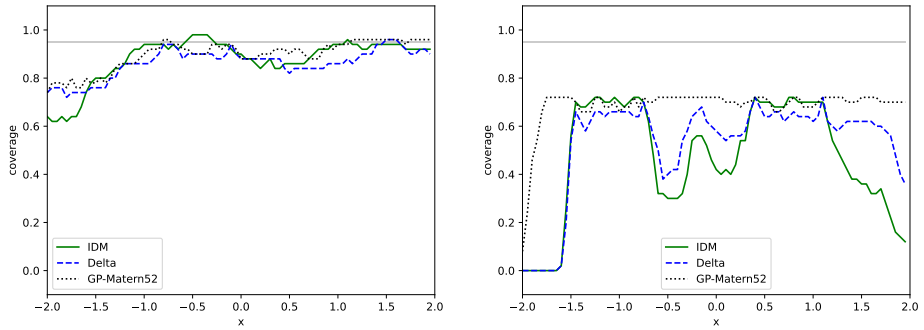
Stochastic Gradient Finite-Difference IDM The algorithm we propose for combining stochastic gradient ascent with finite-difference IDM is given in Alg. 2. This algorithm is used in all experiments. It estimates β-confidence intervals given a stochastic objective $\tilde{\mathcal{L}}$ that is an unbiased estimate of the true objective $\mathbb{E}[\tilde{\mathcal{L}}] = \mathcal{L}$ and evaluated each iteration i on a new minibatch of the dataset. The other inputs are the target evaluation ψ, scalar width λ, learning rate schedule η, sample size S, and desired



(a) IDM (this paper) (b) Delta Method

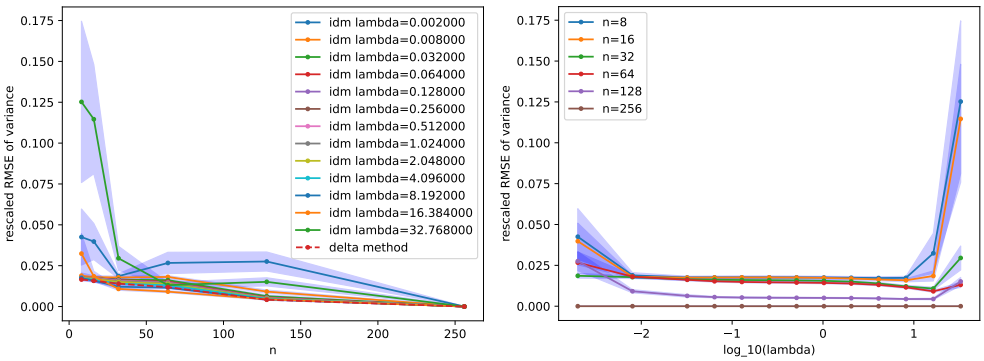
(c) GP-Matern52 (d) True sampling uncertainty by simulation

Figure 5: Fits along with uncertainty bounds and estimated prediction-covariance matrix for data generated from $y = \frac{1}{10}x^2 - \frac{1}{2}x + 5 + \frac{1}{10}\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$



(a) **quadratic** task (b) **sin** task

Figure 6: Coverage of 95% confidence intervals pointwise for prediction at each input.



(a) Number of data points as dependent variable. (b) λ as dependent variable.

Figure 7: Convergence of IDM in **quadratic** task for different values of n and λ. There is a wide dynamic range of acceptable λ.

confidence β. In practice, we fixed λ to be 1% of the training objective at the MLE, though it is also possible to average the IDM output across variable λ.

Algorithm 2: Stochastic gradient finite-difference implicit delta method

Input: Stochastic objective $\tilde{\mathcal{L}}$, evaluation ψ, scalar λ, schedule η, sample size S, confidence β

Function `SG-FDIDM(`$\tilde{\mathcal{L}}$, ψ, λ, η, S, β`):`

```
// randomly initialize parameters
```

$\hat{\theta}_n \sim \text{Uniform}(-0.1, 0.1)$

```
// maximize ℒ with stochastic gradient ascent
```

$i \leftarrow 0$

while $\hat{\theta}_n$ *not converged* **do**

  $\hat{\theta}_n \leftarrow \hat{\theta}_n + \eta(i)\nabla_\theta \tilde{\mathcal{L}}_i$

  $i \leftarrow i + 1$

end

```
// collect S samples for estimating ψ(θ̂_n)
```

forall $s \in [0, S)$ **do**

  $\psi_{0,s} \leftarrow \psi(\hat{\theta}_n)$

  $\hat{\theta}_n \leftarrow \hat{\theta}_n + \eta(i)\nabla_\theta \tilde{\mathcal{L}}_i$

  $i \leftarrow i + 1$

end

```
// maximize ℒ + λψ with stochastic gradient ascent
```

while $\hat{\theta}_n$ *not converged* **do**

  $\hat{\theta}_n \leftarrow \hat{\theta}_n + \eta(i)\nabla_\theta (\tilde{\mathcal{L}}_i + \lambda\psi)$

  $i \leftarrow i + 1$

end

```
// collect S samples for estimating ψ(θ̂_n(λ))
```

forall $s \in [0, S)$ **do**

  $\psi_{\lambda,s} \leftarrow \psi(\hat{\theta}_n)$

  $\hat{\theta}_n \leftarrow \hat{\theta}_n + \eta(i)\nabla_\theta (\tilde{\mathcal{L}}_i + \lambda\psi)$

  $i \leftarrow i + 1$

end

```
// take average of samples
```

$\bar{\psi}_0 \leftarrow \frac{1}{S}\sum_{s\in[0,S)} \psi_{0,s}$

$\bar{\psi}_\lambda \leftarrow \frac{1}{S}\sum_{s\in[0,S)} \psi_{\lambda,s}$

return $\bar{\psi}_0 \pm \Phi^{-1}((1+\beta)/2)\sqrt{\frac{1}{\lambda}(\bar{\psi}_\lambda - \bar{\psi}_0)}$ `// confidence interval for` $\psi(\hat{\theta}_n)$

We use TensorFlow v2.1.0 `tensorflow.compat.v1` to compute the gradients of the models in all experiments [1]. In experiments with a Gaussian process (GP), we applied the GPFlow package and optimized the marginal likelihood w.r.t. the kernel parameters (type II maximum likelihood) [3].

1D Synthetic Experiments The setup for the synthetic experiments is as follows,

- **quadratic**: draw 25 data points $x \sim \mathcal{N}(0, 1)$ and let $y = \frac{1}{10}x^2 - \frac{1}{2}x + 5 + \frac{1}{10}\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. Use constant learning rate $\eta(\cdot) = \frac{1}{100}$.
- **sin**: fix 160 data points evenly spaced in ranges $[-1.5, -0.7)$ and $[0.35, 1.15)$ and let $y = -\sin(3x - \frac{3}{10}) + \frac{1}{10}\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$. Use constant learning rate $\eta(\cdot) = \frac{1}{200}$ for MLE objective and $\frac{1}{2000}$ for ψ-regularized objective.

In each case, the objective is evaluated over the entire dataset, such that $\tilde{\mathcal{L}} = \mathcal{L}$ and $S = 1$. In accordance with a univariate Gaussian density over the prediction, the network is trained to minimize half the sum of squared errors and the resulting interval is divided by the root mean squared error after convergence. The visualized covariance matrices were projecting to ensure positive semi-definiteness. This was done by averaging the matrix with its transpose, finding the eigendecomposition of the resulting matrix, then reconstructing the matrix with only the eigenvectors corresponding to positive eigenvalues.

Algorithm 3: Multivariate finite-difference implicit delta method

Input: Learning objective $\mathcal{L}$, multivariate evaluation ψ, scalar λ

Function `MV-FDIDM(`$\mathcal{L}$, ψ, λ`):`

  $\hat{\theta}_n \leftarrow \arg\max_\theta \mathcal{L}(\theta)$ `// optimize learning objective`

  $K \leftarrow \dim(\psi(\theta))$

  forall $k = 1, \ldots, K$ **do**

    $\hat{\theta}_n^{(k)}(\lambda) \leftarrow \arg\max_\theta \mathcal{L}(\theta) + \lambda\psi_k(\theta)$ `// optimize` ψ_k`-regularized objective`

  end

  return $\left[\frac{1}{\lambda}(\psi_j(\hat{\theta}_n^{(k)}(\lambda)) - \psi_j(\hat{\theta}_n))\right]_{j=1:K,k=1:K}$ `//` $K \times K$ `covariance matrix of` $\hat{\psi}_n$

Figure Figure 7 shows the convergence of the root mean squared error of IDM w.r.t. the true variance as determined by 50 resamples from the data generating distribution in the quadratic task. The squared errors are rescaled by n^2 to account for decreasing scale ($\frac{1}{n}$) of the true the variance as n grows. Shaded error bars indicate one standard error. Convergence for the standard delta method is also shown for reference. We find there is a wide dynamic range of acceptable values of λ. Small values of $\lambda < 0.01$ perform poorly, likely due to numerical instability, but performance improves for larger λ. The setting $\lambda = 0.512$ even outperforms the delta method. These findings support the implication of Theorem 2 indicating that convergence holds as long as λ grows sublinearly to n.

Benchmark Classification Experiments For all classification datasets, the stochastic objective $\tilde{\mathcal{L}}$ is defined over minibatches of size $B = 128$. $\tilde{\mathcal{L}}_i$ is selected by stepping repeatedly through the randomly shuffled dataset in blocks of size B. RMSProp with an initial learning rate of 0.01 is used for optimization.[5] ψ is evaluated over a held-out validation dataset, a randomly selected 20% subset of the data for the benchmarks except `mnist` which is evaluated on the standard 10k test split of the 60k-10k partition.

KL Down-Weighting Experiment We use the standard 60k `mnist` train dataset. ψ is defined as the total negative reconstruction loss on a random 1k subset of the training data. The Adam [2] optimizer is applied to the (possibly) regularized VAE objective with initial learning rate 0.001. The experiment adapted the VAE implementation in TensorFlow available at `https://github.com/wiseodd/generative-models`.

D Finite-Difference IDM for Multivariate ψ

In this section we present the multivariate algorithm for finite-difference IDM (FDIDM). In Alg. 3, a multivariate evaluation function ψ is taken as input, along with a learning objective $\mathcal{L}$ and scalar width λ (as in the univariate case). The algorithm optimizes the original MLE objective, then optimizes K regularized MLE objectives, each with a different dimension of ψ. To obtain the estimated $K \times K$ covariance matrix $\hat{\Sigma}$ of ψ, finite differences are calculated by crossing the output dimension of ψ corresponding to the row of $\hat{\Sigma}$ with the optimized ψ-regularized objective corresponding to the column of $\hat{\Sigma}$.

Supplementary Material References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`

[3] Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40): 1–6, apr 2017. URL `http://jmlr.org/papers/v18/16-537.html`.

[4] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245, 1994.

[5] Aad van der Vaart and Jon Wellner. Weak convergence and empirical processes. *Springer*, 1996.