Carnegie Mellon University

Department of Statistics & Data Science

Thesis Proposal

Draft as of September 2, 2025

Uncertainty quantification and inference using online methods and neural networks

Selina Carter

Date of Proposal: September 2, 2025

Committee:
Arun Kumar Kuchibhotla, Advisor
Jeff Schneider, Carnegie Mellon University
Larry Wasserman, Carnegie Mellon University
Cosma Shalizi, Carnegie Mellon University
Pratik Patil, University of Texas at Austin

Abstract

Uncertainty quantification (UQ) is essential for performing statistical inference on parameters or outcomes of interest. This thesis compares methods for constructing confidence and prediction intervals using online algorithms and neural networks. There are three main areas:

Inference using online algorithms. Many data sources arrive in a streaming, "online" fashion or are too large for traditional estimation tools. Stochastic gradient descent (SGD) is a popular estimation technique in these cases. This thesis evaluates recent methods for conducting inference in this setting, specifically: the Averaged SGD (ASGD) plug-in variance estimator, t-statistic methods, and the Hull-based Confidence Interval (HulC). Through a simulation study on simple inference tasks—linear and logistic regression in both low and high dimensions—we find that the HulC method performs comparably to the t-statistic method and significantly outperforms the plug-in ASGD variance estimator in terms of attaining the desired coverage and minimizing confidence interval width. This result is possibly explained by the high sensitivity of ASGD plug-in method to tuning parameters such as the learning rate, whereas the HulC benefits from fewer assumptions and greater robustness to hyperparameter choices.

We seek to run additional experiments using other online algorithms such as implicit SGD (Toulis and Airoldi, 2017), ROOT-SGD (Li et al., 2022), gradient-free SGD (Chen et al., 2024), and truncated SGD (Zhou et al., 2021).

Neural network ensembles for UQ. Existing work on ensembles of neural networks largely overlooks the utility of bootstrapping data as a UQ technique. Bootstrapping is a non-parametric statistical technique for quantifying uncertainty when the estimator's theoretical distribution is unknown. However, it relies on key assumptions—one sufficient condition being the Hadamard differentiability of the estimator (or functional). We investigate the conditions under which neural network estimators of the underlying function $f(x)$, given fixed input x, satisfy Hadamard differentiability, enabling bootstrapping as a pointwise UQ method for inferring $f(x)$ up to some pointwise bias. Our simulations on simple problems demonstrate that by bootstrapping training data in multi-layer perceptrons (MLPs) using relu or tanh activation functions, we can construct pointwise confidence intervals that achieve the target uncertainty level for $f(x)$ if the neural network architecture is sufficiently flexible. Since bootstrapped ensembles are computationally expensive, we additionally explore new techniques that reduce this computational burden, such as importance sampling, t-statistics, and HulC-based approaches. As an application in physics,

we examine the challenge of inferring temporal state-to-state dynamics of plasma in a Tokamak—a magnetic confinement device central to thermonuclear fusion research. We evaluate the effectiveness of various ensembles for accurately inferring these dynamics through a simulation study using generated Tokamak ground-truth data.

Prediction intervals for streaming time series using prior finite-horizon data. Given access to multiple (finite-horizon) sequences, i.e., $\{(x_{i,t}, y_{i,t})_{t=1}^{T_i}\}_{i=1}^{N}$, where $x_{i,t} \in \mathbb{R}^d$ are covariates and $y_{i,t} \in \mathbb{R}^p$ is the state vector we are trying to predict, we want to predict on a new sequence $(x_{N+1,t}, y_{N+1,t})_{t=1}^{T_0}$ that is streaming online and is right-censored (i.e., we don't know when the trajectory will end, so we only observe up to time T_0). Our aim is two-fold: first, using a black-box prediction method (such as recurrent neural networks or transformers) we seek to forecast s-steps-ahead states $\hat{y}_{N+1,T_0+1}, \dots \hat{y}_{N+1,T_0+s}$; second, we want to learn prediction intervals that have tight widths, primarily focusing on conformal methods. We will first assume the case of iid or exchangeable sequences and then relax these assumptions. There are three main outputs: (1) we will develop a new algorithm that incorporates the previous bank of sequences to predict s-step-ahead states and prediction bounds; (2) we will show analytically that this algorithm reduces the prediction interval width compared to baseline algorithms in the literature, while also maintaining correct theoretical coverage; (3) in a simulation study, we will test the proposed algorithm against baseline techniques. As a use case, I will primarily focus on Tokamak plasma dynamics, a challenging problem in nuclear energy research.

This idea is inspired principally by recent work (Angelopoulos et al., 2023) that models non-conformity scores in an online setting: they assume a single (potentially adversarial) time series consisting of covariates ($x_t \in \mathcal{X}$) and responses ($y_t \in \mathcal{Y}$) for $t \in \mathbb{N}$, and their aim is to construct a prediction set C_t that does not require the assumption of exchangeable data as in standard conformal prediction. They develop a method ("conformal PID control") that achieves long-run coverage in the time horizon T and sharp prediction sets even under distribution shift. However, they do not consider the framework in which multiple (finite-horizon) sequences are available (i.e., $\{(x_{i,t}, y_{i,t})_{t=1}^{T_i}\}_{i=1}^{N}$) prior to predicting on a new sequence that is streaming. We seek to explore if, under additional assumptions on the additional sequence data (such relevancy to the new unknown trajectory), we can analytically reduce widths of the prediction intervals than the method proposed by Angelopoulos et al. (2023).

Introduction

Uncertainty quantification (UQ) is essential for performing statistical inference on parameters or outcomes of interest. This thesis compares methods for constructing confidence and prediction intervals using online algorithms and neural networks.

There are three main areas of this thesis. The first is **inference using online algorithms** (1), which focuses on constructing confidence intervals with streaming data. The second is **neural network ensembles for UQ** (2), which focuses on constructing pointwise confidence intervals using neural networks in a computationally efficient manner. The third is **prediction intervals for streaming time series using prior finite-horizon data** (3), which explores black-box prediction methods to forecast future states in temporally dependent sequences, given a bank of existing trajectories.

Following, I will individually detail each project including prior literature, proposed work, and next steps. Section 4 describes the overall thesis timeline across all projects.

1 Inference using online algorithms

Many data sources arrive in a streaming, "online" fashion or are too large for traditional estimation tools. Stochastic gradient descent (SGD) is a popular estimation technique in these cases. However, statistical inference for online algorithms is a difficult problem because estimation of asymptotic variance can inflate the computational cost: estimators obtained from online/sequential algorithms forces one to consider the computational aspects of the inference problem, i.e., one cannot access all of the data as many times as needed. Previous works have proposed online estimation of the covariance matrix as well as batching methods to construct confidence intervals. In this work, we propose the use of the recently developed HulC (hull-based confidence) procedure for uncertainty quantification in the online setting. The highlights of this procedure include: no inflation in the computational cost; no estimation of the asymptotic variance; and asymptotically exact coverage.

We compare the performance of the HulC procedure with those of previous works in the context of linear and logistic regression over a wide range of covariance settings and dimension-aspect ratios (Carter and Kuchibhotla, 2025). Our main finding is that we get comparable or better coverage properties compared to the methods that estimate the asymptotic variance.

We also find that although SGD is the most commonly mentioned method in machine learning – and while its cousin, averaged SGD (ASGD), has desirable asymptotic properties for inference purposes – our implementation and simulations show that the practical performance of ASGD is highly sensitive to the choice of tuning parameters of the algorithm. We could not find a simple remedy that improves performance and also makes the asymptotic properties manageable. This was at least surprising. It is unclear if this is a well-known observation in the literature on online algorithms, and we hope that this work acts as a word of caution to anyone using online algorithms blindly.

1.1 Motivation and prior literature

Suppose we have data $Z_1, \ldots, Z_T$ that are generated independently from a common distribution P. The analyst is interested in a summary functional $\theta_\infty(P) \in \mathbb{R}^d$ defined by the optimization problem:

$$\theta_\infty \equiv \theta_\infty(P) \ := \ \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \mathbb{E}_P[\ell(Z;\theta)],$$

for some loss function $\ell(\cdot;\cdot)$. The objective function is $\mathbb{E}_P[\ell(Z;\theta)]$. We (implicitly) assume that θ_∞ is uniquely defined. Based on the data, a natural estimator (referred to as the *M*-estimator) of θ_∞ is based on the empirical loss:

$$\widehat{\theta}_T := \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \frac{1}{T} \sum_{i=1}^{T} \ell(Z_i;\theta). \tag{1}$$

How do we perform inference on θ_∞ in an online fashion? Construction of confidence intervals for functionals based on asymptotically *normal* estimators (i.e., Wald inference procedures) is a classical topic in statistical inference. In the online setting, the ground-breaking work by Robbins and Monro (1951) introduced the first formal stochastic approximation procedure for finding the optimum of a regression function, becoming a cornerstone of statistical optimization. Several works have modified stochastic approximation algorithms with better finite-sample or asymptotic performance. Stochastic gradient descent with averaging by Ruppert (1988) and Polyak and Juditsky (1992), otherwise known as averaged SGD (ASGD), is one of the first such refinements. They establish that, given the ASGD estimator $\bar{\theta}_T$ and under key assumptions (a fixed dimension d, a *strongly* convex objective function J, a Lipschitz gradient $\nabla_\theta \mathbb{E}_P[\ell(Z;\theta)]$, and step sizes η_t that diminish sufficiently slowly to 0), they prove that

$$\sqrt{T}(\bar{\theta}_T - \theta_\infty) \xrightarrow{d} N(0, J^{-1}VJ^{-1}),$$

where $J := \nabla^2_\theta \mathbb{E}_P[\ell(Z; \theta_\infty)]$ is the Hessian matrix of the objective function at $\theta = \theta_\infty$ and V is the covariance matrix of $\nabla_\theta \ell(Z; \theta)_\infty$.

Relatively recently, efficient online inference for θ_∞ has garnered renewed research interest. Early contributions include Pelletier (2000) and Gahbiche and Pelletier (2000) both of which seemingly went unnoticed by the more recent work Chen et al. (2020). See Carter and Kuchibhotla (2025) for an elaborate discussion of recent literature. All these works propose different online estimators $\widetilde{\theta}_T$ such that

$$r_T(\widetilde{\theta}_T - \theta_\infty) \xrightarrow{d} N(0, \Gamma),$$

for some rate of convergence r_T and some covariance matrix Γ. The key is to be able to estimate Γ in an online fashion such that confidence intervals can be constructed for each new sample Z_i. For example, Chen et al. define an online plug-in estimator for $\Gamma = J^{-1}VJ^{-1}$ as well as a batch-means estimator.

In addition, there are some alternatives to this online Wald inference procedure, including bootstrap-based methods, the functional central limit theorem (CLT), or the *t*-statistic; see Fang et al. (2018), Zhong et al. (2023), Lam and Wang (2023), Lee et al. (2022), and Zhu et al. (2024). Except for the method of Zhu et al. (2024) (based on Ibragimov and Müller (2010)), all other existing methods require additional computations or memory compared to the original online algorithm. For example, the variance estimators of Gahbiche and Pelletier (2000) and Chen et al. (2020) require storing the intermediate iterations of the SGD, and the bootstrap method of Fang et al. (2018) requires running a large number of SGDs parallel to the original SGD. In addition, all existing methods (except Zhu et al. (2024)) require some additional structure on the online algorithm in addition to asymptotic normality. Moreover, even relying on asymptotic normality is restrictive from the point of view of finite-sample performance. In the following section, we propose the application of HulC (Kuchibhotla et al., 2024) for computationally efficient, rate-optimal, and asymptotically valid confidence regions for θ_∞.

1.2 Objectives and Contributions

In Carter and Kuchibhotla (2025), we propose computationally efficient, rate-optimal, and asymptotically valid confidence regions based on the output of online algorithms *without* estimating the asymptotic variance. As a special case, this implies inference from any algorithm that yields an asymptotically normal estimator. We focus our efforts on the ASGD estimator, $\bar{\theta}_T$. The online HulC confidence interval (CI) is described in Section 5.1 of the Appendix.

Kuchibhotla et al. (2024) provide a finite-sample validity guarantee for this confidence interval, which we restate in the following result:

Theorem 1. *Suppose $Z_1, Z_2, \ldots, Z_T$ are independent random variables. Then for $1 \leq k \leq d$,*

$$\mathbb{P}\left(e_k^\top \theta_\infty \notin \widehat{\mathrm{CI}}_{T,\alpha}^{(k)}\right) \leq \alpha\left(1 + 2(B_\alpha \Delta_T)^2 e^{2B_\alpha \Delta_T}\right),$$

where

$$\Delta_T := \max_{1 \leq j \leq B_\alpha}\left(\frac{1}{2} - \min_{\gamma \in \{\pm 1\}} \mathbb{P}(\gamma(e_k^\top \bar{\theta}_T^{(j)} - e_k^\top \theta_\infty) \leq 0)\right)_+,$$

represents the maximum median bias of the estimators.

Proof. The result follows from Theorem 2 of Kuchibhotla et al. (2024). □

1.2.1 Simulation study

In a simulation study, we assess the utility of HulC by comparing confidence regions for $\theta_\infty \in \mathbb{R}^d$ on two simple cases: linear regression and logistic regression. Mimicking the simulation settings from Chen et al. (2020), we generate T iid samples (Y_i, X_i), $1 \leq i \leq T$ (for further details, see section 4 of Carter and Kuchibhotla, 2025). Throughout, we aim to cover $e_k^\top \theta_\infty$ with a 95% confidence for each coordinate $k \in \{1, 2, \ldots, d\}$. We take non-random scalar step sizes of the form $\eta_t = ct^{-0.505}$, with a grid of c values.

We compare four different inference methods: **(1)** the Wald interval (an offline method used as a baseline); **(2)** the ASGD plug-in estimator by Chen et al. (2020); **(3)** ASGD t-stat (Ibragimov and Müller, 2010; Zhu et al., 2024); and **(4)** the online HulC CI based on ASGD estimators. The definitions are in Appendix 5.2.

1.3 Findings

Figure 1 shows the “typical behavior” of the inference methods in the case of linear regression, Toeplitz covariance, and a high dimension ($d = 100$). Our findings at a high level are as follows:

- **ASGD is highly sensitive to hyperparameter c**: There is a "Goldilocks zone" of c (depending on the model parameters) in which the ASGD estimator $\bar{\theta}_T$ converges to θ_∞, which can be quite narrow depending on the problem. We do not know of any theoretical results supporting these empirical observations.

- **The ASGD plug-in confidence interval undercovers θ_∞**: This undercoverage might stem from two sources: (i) slow rate of convergence to asymptotic normality; and (ii) slow rate of consistency of the ASGD plug-in variance estimation. (See Figures 5, 6, and 7 in the Appendix).

- **Both the t-stat and HulC intervals achieve the correct coverage for an appropriately chosen c**: This suggests that these online methods are practical alternatives to the ASGD plug-in estimator.

- **HulC confidence intervals are "comparable" in width**: Typically, HulC confidence intervals are wider than all other methods and this is in line with the theoretical width analysis of Kuchibhotla et al. (2024). Our numerical study suggests that the HulC intervals are only slightly wider than those of the t-stat for most choices of c, T, and covariance schemes.

1.3.1 Directions for future research

We seek to run additional experiments using other online algorithms such as implicit SGD (Toulis and Airoldi, 2017), ROOT-SGD (Li et al., 2022), gradient-free SGD (Chen et al., 2024), and truncated SGD (Zhou et al., 2021).

2 Neural network ensembles for UQ

Existing works on ensembles of neural networks largely overlook the utility of bootstrapping data as a UQ method. Bootstrapping is a non-parametric statistical technique for quantifying uncertainty when the estimator's theoretical distribution is unknown. However, recent findings suggest that bootstrapping neural networks is unnecessary; rather, random weight initialization combined with varying mini-batch sequences in stochastic gradient descent provides sufficient diversity among ensemble members to capture model uncertainty for inference (see Nixon et al. (2020); Lakshminarayanan et al. (2017)). In this work, we challenge this current consensus by constructing

Figure 1: Linear regression, Covariance = Toeplitz, d = 100

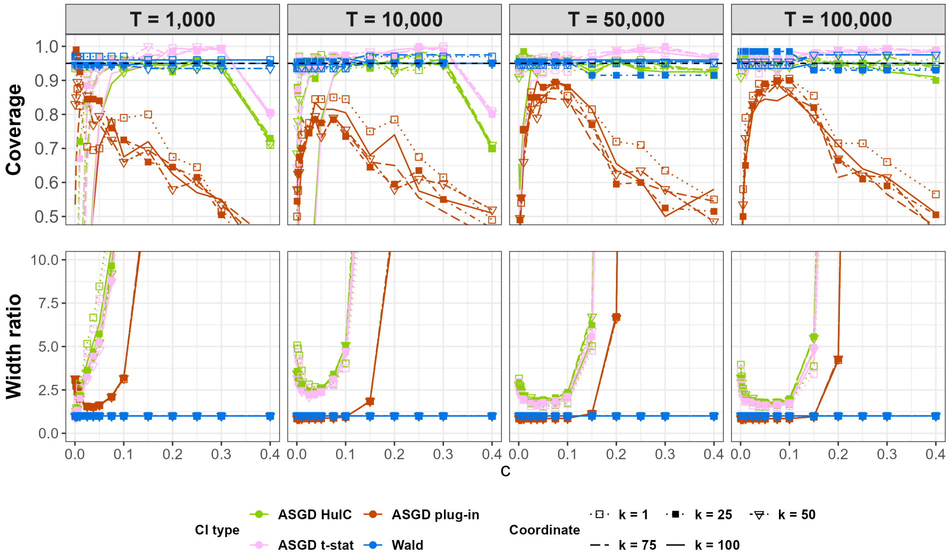


Figure 2: Comparison of Wald, AGSD plug-in, HulC, and t-stat methods in the linear regression setting with a Toeplitz covariance structure and dimension $d = 100$. The ASGD plug-in confidence interval consistently demonstrates poor coverage, while both the HulC and the t-stat methods generally produce correct coverage for appropriately chosen c. Meanwhile, the width ratios for t-stat and HulC are not excessively large when c is appropriately chosen; as the sample size T increases, the ratios decrease.

pointwise confidence intervals of the (true) underlying function $m(\mathbf{x})$ with bootstrapping compared to fixed-data (non-bootstrapped) ensembles. We demonstrate that large fixed-data ensembles relying *only* on different initial weights and mini-batch gradient descent typically fail to cover the target parameter at the desired $(1-\alpha)\cdot 100\%$ rate of uncertainty. Instead, by bootstrapping data for each neural network member of an ensemble, we can construct pointwise confidence intervals that achieve the target uncertainty level for $m(\mathbf{x})$ if the neural network architecture is sufficiently flexible. Since bootstrapped ensembles are computationally expensive, we additionally explore three techniques that reduce this computational burden: the "cheap bootstrap" (Lam, 2022), HulC (Kuchibhotla et al., 2024), and the robust t-statistic (Zhu et al., 2024).

2.1 Prior literature

Neural networks are a flexible class of estimators which are able to approximate a large class of functions, even in the presence of noise (Raghu et al., 2017). Neural network ensembles – commonly referred to as deep ensembles – are widely used across many fields due to their strong empirical performance. By training B separate models and aggregating their predictions, ensembles often achieve better results than any individual model alone. The simplest type of ensemble is to average the predictions of the B models. If the B models are trained on independent data sets, such a procedure can lead to a reduction in bias of the final prediction (Bishop and Nasrabadi, 2006). When given a single dataset, aggregation of B models using bootstrapped samples (*bagging* see for example Breiman, 1996), which reduces the expected error of the ensemble prediction over individual models.

Used in this way, deep ensembles aim to enhance the stability and unbiasedness of point estimates; hence, they do not directly address uncertainty quantification, such as confidence intervals. Bootstrapping (Efron, 1979), on the other hand, has been used in a variety of inferential and UQ tasks, including linear and nonlinear regression (Heng and Lange, 2025), non-linear non-parametric regression (Chernozhukov et al., 2022), and even time series models (Xu and Xie, 2021a). As a resampling-based method, bootstrapping offers a way to approximate the sampling distribution of an estimator, enabling the construction of confidence intervals, which is particularly useful when the theoretical distribution is unknown or difficult to derive – such is the case for neural networks.

However, the role of bootstrapping in training deep ensembles for uncertainty quantification has been largely overlooked. Notably, Lakshminarayanan et al. (2017) established a widely adopted practice of using only random weight initialization and varied mini-batch sequences across ensemble members without applying bootstrapping. Further, Nixon et al. (2020) argued that bootstrapped deep ensembles offer no performance benefit. However, their evaluation focused on predictive “accuracy" (i.e., reducing the variance of the estimator), rather than producing a confidence interval, which is the goal of bootstrapping and central to uncertainty quantification more broadly. Additionally, their experiments were limited to classification tasks rather than regression.

2.2 Objectives

We seek to challenge the current consensus that maintains bootstrapping is an unnecessary procedure for UQ using deep ensembles. We primarily explore this through simulations. We conduct extensive experiments to test if bootstrapping enables superior **pointwise** $(1-\alpha)\cdot 100\%$ **confidence intervals** in regression problems using deep ensembles, as opposed to relying solely on different initialization and fixed (non-bootstrapped) data for each ensemble member. In addition, since quantile-based UQ with bootstrapping requires a large ensemble size and is computationally expensive, we apply alternative approaches requiring only a small ensemble size, namely: the “cheap bootstrap” (Lam, 2022), HulC (Kuchibhotla et al., 2024), and the robust t-statistic (Zhu et al., 2024).

2.2.1 Preliminary

We consider a training dataset $\mathcal{D}_n := \{\mathbf{X}_i, Y_i\}_{i=1}^{n}$, with input $\mathbf{X}_i \in \mathbb{R}^d$ and output $Y_i \in \mathbb{R}$. Note that the results also apply to a vector-output $\mathbf{Y}_i$ by applying the UQ procedure separately to each coordinate. Our goal is to construct a $(1-\alpha)\cdot 100\%$ **pointwise confidence interval** over true conditional expectation of Y_i given $\mathbf{X}_i = \mathbf{x}_i$, i.e.,

$$m(\mathbf{x}_i) := \mathbb{E}[Y_i | \mathbf{X}_i = \mathbf{x}_i],$$

and we assume the residuals $\varepsilon_i := Y_i - m(\mathbf{x}_i)$ are independent random variables with mean 0 and finite conditional variance $\sigma_\varepsilon^2(\mathbf{x}_i) < \infty$. Our approximating function for m is a fully-connected feedforward neural network with L hidden layers and H hidden units per layer. Specifically, if $L = 1$, then the output of the neural network is

$$f_H(\mathbf{x}, \boldsymbol{\theta}) := \nu_0 + \sum_{h=1}^{H} \nu_h \psi(\tilde{\mathbf{x}}^\top \boldsymbol{\omega}_h),$$

where $\boldsymbol{\theta} := [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_H, \nu_0, \ldots, \nu_H]^\top$ is the vector of network weights, the activation function ψ is applied to each hidden unit, and $\tilde{\mathbf{x}} := [1, \mathbf{x}^\top]^\top \in \mathbb{R}^{d+1}$ is the input vector with the intercept (or "bias") component. As in Franke and Neumann (2000), we assume the network function is uniquely parameterized by restricting the parameter space to a fundamental compact domain $\boldsymbol{\Theta}_H$, which excludes symmetry operations of the weight vector, making $\boldsymbol{\theta}$ identifiable. The training procedure minimizes the objective function J under squared error loss:

$$J(\boldsymbol{\theta}) := \mathbb{E}[(Y_i - f_H(\mathbf{X}_i, \boldsymbol{\theta}))^2]$$

Using mini-batch gradient descent and random weight initialization over a fixed number of epochs (without early stopping), we train the network to obtain an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$, producing $\hat{f}_H(\mathbf{x}) := f_H(\mathbf{x}, \hat{\boldsymbol{\theta}})$, namely,

$$\hat{\boldsymbol{\theta}}_n := \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}_H}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f_H(\mathbf{X}_i, \boldsymbol{\theta}))^2$$

For simplicity, we will write f_H and $\hat{f}_H$ to denote a fully connected neural network given a particular fixed number of hidden layers $L \geq 1$ and hidden neurons H.

2.2.2 Theoretical sketch

Here I provide a sketch of a theoretical direction that underpins the utility of bootstrapping a neural network to construct a $(1 - \alpha) \cdot 100\%$ pointwise confidence interval over $m(\mathbf{x})$; this section needs more rigor.

If $m(\mathbf{x}) = f_H(\mathbf{x}, \boldsymbol{\theta}_0)$ for some $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}_H$ (i.e., the neural network architecture is correctly specified), then a non-linear regression central limit theorem for $\hat{\boldsymbol{\theta}}_n$ enables inference on $\boldsymbol{\theta}_0$ (see Franke and Neumann (2000), p. 1931). Furthermore, using the delta method, one can perform inference on $m(\mathbf{x})$.

In the misspecified case where $m(\mathbf{x}) \neq f_H(\mathbf{x}, \boldsymbol{\theta}_0)$, then $\hat{\boldsymbol{\theta}}_n$ converges to the parameter of the best network function approximator for $m(\mathbf{x})$ (denoted by $\tilde{\boldsymbol{\theta}}_0 := \operatorname{argmin}_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_H} J(\boldsymbol{\theta})$) if **(A1)** the activation function ψ is bounded and twice continuously differentiable with bounded derivatives, and m is bounded; **(A2)** $J(\boldsymbol{\theta})$ has a unique global minimum at $\tilde{\boldsymbol{\theta}}_0$ lying in the interior of $\boldsymbol{\Theta}_H$ and $\nabla^2 J(\tilde{\boldsymbol{\theta}}_0)$ is positive definite (Franke and Neumann, 2000; White, 1989). As stated by Franke and Neumann (2000), the central limit theorem results on one-hidden-layer ($L = 1$) networks can also be applied to multi-layer networks ($L > 1$).

The bootstrap provides an alternative approximation for the distribution of $\hat{\boldsymbol{\theta}}_n - \tilde{\boldsymbol{\theta}}_0$ if we have access to a uniformly consistent estimator $\hat{m}$ for m

(Franke and Neumann, 2000), such as "connectionist sieve estimator" (White, 1990), where the complexity of the network is allowed to grow with n. (Note that Franke and Neumann (2000) offer a non-standard bootstrap procedure that we should explore.) However, inference requires a central limit theorem for this case; otherwise, if the dimension of the network is fixed, then there is unknown bias (error) in the approximation of m.

2.2.3 Methods

Through experiments on simulated data, we use five methods that produce $(1-\alpha)\cdot 100\%$ pointwise confidence intervals over the true conditional function $m(\mathbf{x})$ across a grid of $\mathbf{x}$ values, where $m(\mathbf{x})$ is approximated by a neural network $\hat{f}_H(\mathbf{x})$ with a fixed architecture (L, H) trained using mini-batch gradient descent and random weight initialization over a fixed number of epochs (without early stopping). Ensemble members are all trained in the same way for each method.

(Method 1) Fixed-data ensemble: We train an ensemble of $B = 200$ neural networks, each using the same dataset $\mathcal{D}_n$. Each ensemble member is initialized with a different set of weights as well as a unique set of mini-batches. We construct confidence intervals using a quantile-based method: for a given input $\mathbf{x}$, each ensemble member produces a prediction, yielding a set of estimates $\{\hat{f}_H(\mathbf{x})^{(1)}, \cdots, \hat{f}_H(\mathbf{x})^{(B)}\}$. We define the pointwise $(1-\alpha)100\%$ confidence interval $\text{CI}_{\text{quantile}}(\alpha, \mathbf{x})$ as the inner $(1-\alpha)$100th-quantiles of the B estimates.

(Method 2) Bootstrapped ensemble: We train an ensemble of $B = 200$ neural networks, each using a bootstrapped dataset $\mathcal{D}_n^*$, that is: the dataset used to train ensemble member b, i.e., $\mathcal{D}_n^b$, is a random sample with replacement from $\mathcal{D}_n$ of size n. We construct confidence intervals using a quantile-based method: for a given input $\mathbf{x}$, each ensemble member produces a prediction, yielding a set of estimates $\{\hat{f}_H(\mathbf{x})^{(1)^*}, \cdots, \hat{f}_H(\mathbf{x})^{(B)^*}\}$. We define the pointwise $(1-\alpha)100\%$ confidence interval $\text{CI}_{\text{quantile}}(\alpha, \mathbf{x})$ as the inner $(1-\alpha)$100th-quantiles of the B estimates.

(Method 3) Cheap bootstrap (Lam, 2022): To test if a smaller ensemble can achieve the desired target, we set $B \in \{5, 6, 10, 15, 20, 30\}$ with bootstrapping to produce a t-statistic-based confidence interval.

(Method 4) Hull-based confidence region (HulC): HulC (Kuchibhotla et al., 2024), described in 1.2, outputs a confidence interval that has coverage of at least $(1-\alpha)\cdot 100\%$ if the median bias of the estimator is at most $\Delta \in [0, 0.5]$. The computational advantage is that HulC requires no more than 11 ensemble members for $\alpha \geq 0.001$. As a starting point for the simulations, we assume $\Delta = 0$; we also test $\Delta \in \{.1, .2, .25, .3\}$ for cases of obvious bias in the estimator $\hat{f}_H$.

(Method 5) **Robust t-statistic ("tstatH"):** This method is an alternative to HulC that computes pointwise t-statistics and corresponding confidence intervals using the B^*_α estimators $\{\hat{f}_H(\mathbf{x})^{(1)}, \cdots, \hat{f}_H(\mathbf{x})^{(B^*_\alpha)}\}$ where B^*_α is defined in Algorithm 5.1. It has the same coverage guarantees as HulC.

2.2.4 Experiment setup

We test three data-generating functions according to Table 1, where $y = m(\mathbf{x}) + \varepsilon$, $\mathbf{x} \in \mathbb{R}^d$, $y \in \mathbb{R}$, and $\varepsilon \sim \mathcal{N}(0, 1)$ is a standard Gaussian noise. See Table 2 in the Appendix for hyperparameter settings. For each setting, we run 100 simulations: each has a new dataset $\mathcal{D}_n = \{Y_i, \mathbf{X}_i\}_{i=1}^n$. Given a particular setting combination $\mathbf{s}$, for each experiment $j \in \{1, \ldots, 100\}$, we employ one of the five ensemble methods to produce B MLPs. After each ensemble member b has finished training, we produce pointwise predictions $\hat{f}_H(\mathbf{x}_k) = \hat{\mathbb{E}}[Y_k|\mathbf{X}_k, \mathbf{s}]^{(b)}$ for $k \in \{1, \ldots, K\}$, evaluated over a fixed, equally spaced grid of input points $\mathbf{X}_k$ within the domain $\otimes_{i=1}^d [-5, 5]$. We set $K = 100$ for $d = 1$ and $K = 125$ for $d = 3$. We then produce the $(1-\alpha) \cdot 100\%$ pointwise confidence interval for each ensemble method.

Using 100 experiments, we calculate the empirical pointwise coverage rate as $\mathrm{C}^{\mathbf{s}} \in \mathbb{R}^K$, that is, the average number of times the true function values $m(\mathbf{x}_k) = \mathbb{E}[Y_k|\mathbf{X}_k]$ have been covered by the pointwise confidence intervals produced by each method. Setting $\alpha = .05$, the target pointwise coverage is 95%. To summarize results to a single metric, we then average the pointwise coverage rates to an **average coverage**, i.e. $\bar{\mathrm{C}}^{\mathbf{s}} := \frac{1}{K} \sum_{k=1}^K \mathrm{C}_k^{\mathbf{s}} \in [0, 100]$, which should also ideally be 95%. We also compare **confidence interval widths** of the five methods. Smaller widths are preferred so that uncertainty is minimal.

As an example, Figure 8 (Appendix) shows an example ensemble ($B = 100$) using Method 1 on the $\sin(x)$ function.

2.3 Findings

Figure 3 compares all five methods (assuming a median bias $\Delta = 0$). Further plots are available in Appendix 5.6. We note the following:

- **Fixed-data ensembles (method 1) rarely achieve the desired pointwise coverage for** $m(\mathbf{x})$: except for a tight range of the step

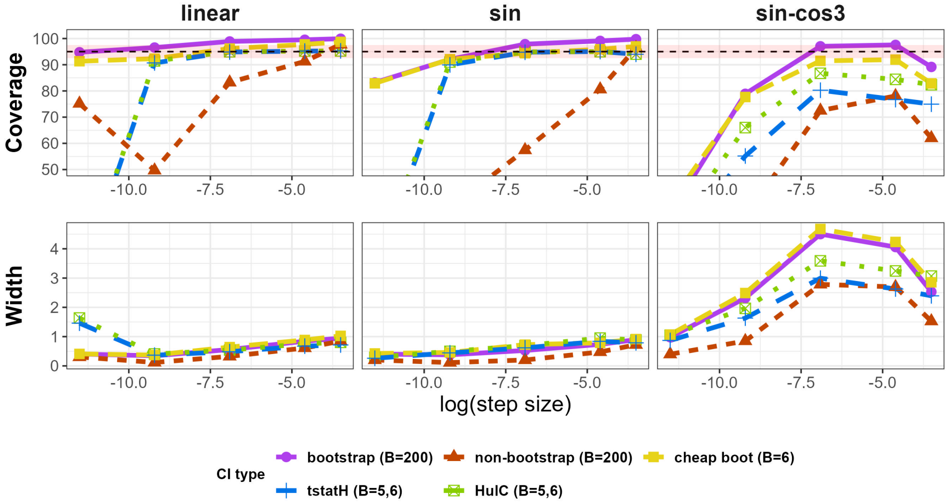


Figure 3: Over 100 experiments, we plot average coverage for all five methods on all three functions, with $n_{bs} = 32$, $n = 1000$, ψ =relu, $n_e = 1000$, and varying step sizes. “Coverage” refers to average pointwise coverage across all grid points.

size hyperparameter, which is problem-specific, fixed-data ensembles are inadequate for UQ on the target function $m(\mathbf{x})$.

- **Bootstrapping (method 2) achieves desired pointwise coverage for $m(\mathbf{x})$ if the step size is appropriate given the problem**: if the step size is too small, bootstrapping undercovers the target; if too large, overcoverage can occur. If the step size is in the correct mid-sized range (around 0.001 in most of our examples), then correct coverage is achieved and interval widths are smallest.

- **However, over-coverage for bootstrapping (method 2) is often a problem**: Given a small (mini-batch) size in combination with a large step size, even if the MLP training has ostensibly "converged", bootstrapping often achieves 100% pointwise coverage of $m(\mathbf{x})$.

- **Like the bootstrap, the cheap bootstrap (method 3) achieves the target coverage if the tuning parameters are in the right range** (see Figure 11 in the Appendix).

- **HulC (method 4) and the robust t-statistic (method 5) achieve the desired coverage in the case of the 1-dimensional functions and $\Delta = 0$.** The widths are all comparable, suggesting that the these two methods do not compensate the smaller ensemble with more uncertainty.

- **The sin-cos3 function poses greater challenges for HulC (method 4) and the robust t-statistic (method 5), likely due to median bias**. We discover that by setting $\Delta = 0.25$ and $n_e \geq 400$, HulC achieves the correct coverage on the sin-cos3 function (see Figure 4). Widths are not egregiously higher.

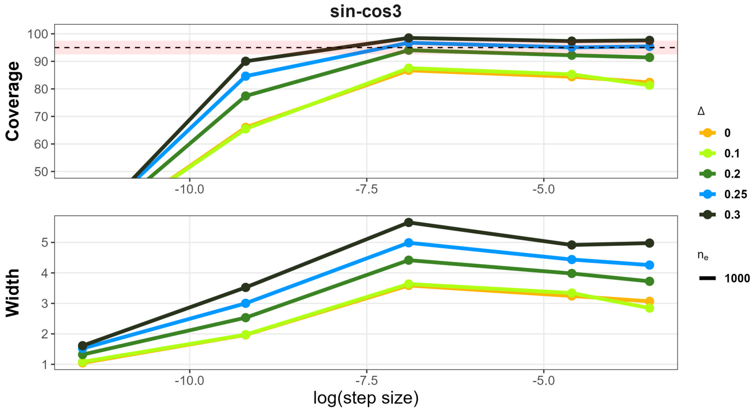


Figure 4: Over 100 experiments, we plot average coverage for HulC with median-bias upper bound Δ on all three functions, with $n_{bs} = 32$, $n = 1000$, $\psi =$relu, $n_e = 1000$, and varying step sizes on the sin-cos3 function. Contrasted with the case of $\Delta = 0$, we can see that coverage improves overall as Δ increases. When the step size $\eta \geq .01 \approx \exp(-4.6)$, coverage is correct for $\Delta = 0.25$. The widths get larger as Δ increases—a "price" of median bias. For $\eta = .01$, the width for $\Delta = .25$ is 5.22, while the width for $\Delta = 0$ is 3.91, a 33% increase.

2.4 Directions for future research

Next steps:

- Flesh out the theory of our procedure (i.e., elaborate on Section 2.2.2).
- We should consider adding the non-standard bootstrap procedure to infer the weight vector $\boldsymbol{\theta}_0$ presented by Franke and Neumann (2000), then using a delta method (or other) technique to infer $m(\mathbf{x})$.
- Since we don't know the median bias upper bound (Δ) a priori on complicated datasets (such as sin-cos3 and the Tokamak data), we

could run HulC with $\Delta = 0.5$ (its maximum value; see Algorithm 1 of Kuchibhotla et al. (2024)) or run Adaptive Hulc (see Algorithm 2 of Kuchibhotla et al. (2024)).

- As an application in physics, we will examine the challenge of inferring temporal state-to-state dynamics of plasma in a Tokamak—a magnetic confinement device central to thermonuclear fusion research. We evaluate the effectiveness of various ensembles for accurately inferring these dynamics through a simulation study using generated Tokamak ground-truth data with a pre-specificed neural network artechture.
  - Currently, we have the code for the ground-truth data. However, each neural network takes many hours to run. I need to create a csv file that records information from each experiment on separate cluster nodes, instead of doing all 100 experiments on a single node.
- We should also create more interesting theoretical datasets in higher dimensions, as well as figure out a way to use "real" data baselines in the classification context (CIFAR for example).

3 Prediction intervals for streaming time series using prior finite-horizon data

In this project, I ambitiously seek to develop a new method. The inspiration comes from my passion for time series (or sequential) data that I had previously encountered in my work at the Inter-American Development Bank. As an application in physics, we will examine the challenge of inferring temporal state-to-state dynamics of plasma in a Tokamak—a magnetic confinement device central to thermonuclear fusion research. We will first explore existing black-box prediction methods to forecast future states in temporally dependent sequences, such as transformers. The next step is to add a key innovation to existing conformal prediction methods on non-exchangeable data: we assume access to a bank of existing (finite) trajectories, which we expect to improve upon existing forecasting/conformal prediction methods that assume a single time series. Indeed, the presence of a bank of existing sequences is applicable to the Tokamak dataset, as well as other potential use cases in finance, robotics, and health.

3.1 Preliminary

We suppose we have access to multiple (finite-horizon) sequences, i.e., $\{(x_{i,t}, y_{i,t})_{t=1}^{T_i}\}_{i=1}^{N}$, where $x_{i,t} \in \mathbb{R}^d$ are covariates and $y_{i,t} \in \mathbb{R}^p$ is the state vector we are trying to predict. We want to predict on a new sequence $(x_{N+1,t}, y_{N+1,t})_{t=1}^{T_0}$ that is streaming online and is right-censored (i.e., we don't know when the trajectory will end, so we only observe up to time T_0).

3.1.1 Objectives

Our aim is two-fold: first, using a black-box prediction method (such as recurrent neural networks or transformers) we seek to forecast s-steps-ahead states $\hat{y}_{N+1,T_0+1}, \ldots \hat{y}_{N+1,T_0+s}$; second, we want to learn prediction intervals that have tight widths, primarily focusing on conformal methods. We will first assume the case of iid or exchangeable sequences and then relax these assumptions.

3.1.2 Outputs

There are three main outputs: (1) we will develop a new algorithm that incorporates the previous bank of sequences to predict s-step-ahead states and prediction bounds; (2) we will show analytically that this algorithm reduces the prediction interval width compared to baseline algorithms in the literature, while also maintaining correct theoretical coverage; (3) in a simulation study, we will test the proposed algorithm against baseline techniques. As a use case, I will primarily focus on Tokamak plasma dynamics, a challenging problem in nuclear energy research.

3.2 Prior literature

This idea is inspired principally by recent work (Angelopoulos et al., 2023) that models non-conformity scores in an online setting: they assume a single (potentially adversarial) time series consisting of covariates ($x_t \in \mathcal{X}$) and responses ($y_t \in \mathcal{Y}$) for $t \in \mathbb{N}$, and their aim is to construct a prediction set C_t that does not require the assumption of exchangeable data as in standard conformal prediction. They develop a method ("conformal PID control") that achieves long-run coverage in the time horizon T and sharp prediction sets even under distribution shift. However, they do not consider the framework in which multiple (finite-horizon) sequences are available (i.e., $\{(x_{i,t}, y_{i,t})_{t=1}^{T_i}\}_{i=1}^{N}$) prior to predicting on a new sequence that is streaming. We seek to explore if, under additional assumptions on the additional sequence data (such relevancy to the new unknown trajectory), we can analytically reduce widths of the prediction intervals than the method proposed by Angelopoulos et al. (2023).

We identify the same gap in related work on conformal inference for non-exchangeable (or time series) data: for example, under adaptive conformal inference (Gibbs and Candes, 2021), a single time series $\{(x_t, y_t)\}_{t\in\mathbb{N}}$ is assumed; likewise, other works focus on guarantees for a single stream of dependent data (Oliveira et al., 2024; Xu and Xie, 2021b, 2023; Zaffran et al., 2022; Stankeviciute et al., 2021; Barber et al., 2023; Chernozhukov et al., 2018, 2021; Feldman et al., 2023; Jensen et al., 2022; Gibbs and Candès, 2024)

Barber et al. (2020)outlines the mathematical limitations of producing conditional (as opposed to marginal) predictive coverage guarantees for a single time series, but they do not consider the case of multiple time sequences that could improve knowledge of the future.

3.3 Next steps

- Provide a 1-hour lecture on transformers for time series (for discussion with Arun; all are invited to join)

- Review literature on black box predictors for time series (ex. Ekambaram et al. (2023); Katz (2025); Razzhigaev et al. (2024); Upadhya (2023); Zeng et al. (2022))

- Discuss ideas on how to improve work by Angelopoulos et al. (2023) when a bank of trajectories is available.

4 Timeline

Project 1	Project 2	Project 3	Time window
Investigate and write code for additional online CI's (implicit SGD, root SGD, gradient-free SGD, truncated SGD)	Make code for running Tokamak simulations on separate clusters	Deliver transformer (for time series) lecture for Committee	September 2025
	Prepare poster for STAMPS workshop (October 4)		Before Oct 4, 2025
Run code for additional online CI's and write up results		Write code base for transformer for time series	October 2025
		Initial idea stage on theory for new conformal prediction	November-December 2025
Completion of additional simulations	Completion of additional simulations		January-February 2026
		Theoretical exploration and design simulations	March 2026
		Simulations	May 2026
Write results into dissertation	Write results into dissertation	Simulations	Summer 2026
		Write results into dissertation	Fall 2026
Defense	Defense	Defense	December 2026

5 Appendix

5.1 Online HulC algorithm

Online HulC confidence interval

Suppose we have an algorithm $\mathcal{A}$ that takes a stream of data and returns an estimator of $\theta_\infty \in \mathbb{R}^d$, and we want to construct a confidence interval for each of the coordinates of θ_∞. The HulC procedure (Kuchibhotla et al., 2024) to construct a $(1-\alpha)$-confidence interval works as follows:

Step 1: For $\alpha \in (0,1]$, set $B_\alpha = \lceil \log_2(2/\alpha) \rceil$. With a standard uniform random variable U, define

$$B_\alpha^* = \begin{cases} B_\alpha, & \text{if } U > 2^{B_\alpha}(\alpha/2) - 1, \\ \lfloor \log_2(2/\alpha) \rfloor, & \text{otherwise.} \end{cases}$$

The choice of B_α is so that $\mathbb{E}[2^{1-B_\alpha^*}] = 1 - \alpha$. If $\log_2(2/\alpha)$ is an integer, then $B_\alpha^* = \log_2(2/\alpha)$.

Step 2: Compute

$$\bar{\theta}_T^{(j)} = \mathcal{A}(Z_{j+B_\alpha^*\lfloor (T-j)/B_\alpha^* \rfloor}, \ldots, Z_j; \theta^{(0,j)}), \quad \text{for} \quad 1 \le j \le B_\alpha^*.$$

This means that $\bar{\theta}_T^{(j)}$ is computed based on the data $Z_j, Z_{B_\alpha+j}, Z_{2B_\alpha+j}, \ldots$. (Note that this is streaming the data $Z_1, \ldots, Z_T$ into B_α^* buckets without the knowledge of the time horizon T.) Note that we allow each estimator to be constructed starting from a different initial value $\theta^{(0,j)}$.

Step 3: For $1 \le k \le d$, construct the confidence interval

$$\widehat{\mathrm{CI}}_{T,\alpha}^{(k)} := \left[\min_{1 \le j \le B_\alpha^*} e_k^\top \bar{\theta}_T^{(j)}, \max_{1 \le j \le B_\alpha^*} e_k^\top \bar{\theta}_T^{(j)} \right]. \tag{2}$$

We refer to the confidence interval (2) as the HulC CI.

5.2 Existing Online Inference Methods

All the existing inference methods for ASGD rely on the expansion and asymptotic normality of $\bar{\theta}_T$. In the following, we briefly summarize the existing methods that are used for comparison with the HulC confidence intervals in Section 5.1. Our comparison is not exhaustive, given the numerous methods in existence.

Wald interval (offline method — baseline): We use the Wald interval as a baseline method. This uses the global minimizer of the empirical loss function as defined in (1). The Wald interval is

$$\widehat{\mathrm{CI}}_{T,\alpha}^{(k)} := \left[e_k^\top \widehat{\theta}_T \pm \frac{z_{\alpha/2}}{T^{1/2}}(e_k^\top \widehat{J}_T^{-1} \widehat{V}_T \widehat{J}_T^{-1} e_k)^{1/2}\right],$$

where

$$\widehat{J}_T := \frac{1}{T}\sum_{i=1}^{T} \nabla^2 \ell(Z_i; \widehat{\theta}_T) \quad \text{and} \quad \widehat{V}_T := \frac{1}{T}\sum_{i=1}^{T} (\nabla \ell(Z_i; \widehat{\theta}_T))(\nabla \ell(Z_i; \widehat{\theta}_T))^\top.$$

ASGD Plug-in (Chen et al., 2020): This is possibly the first general inference method using ASGD. The confidence interval is given by

$$\widehat{\mathrm{CI}}_{T,\alpha}^{(k)} := \left[e_k^\top \bar{\theta}_T \pm \frac{z_{\alpha/2}}{T^{1/2}}(e_k^\top \widetilde{J}_T^{-1} \widetilde{V}_T \widetilde{J}_T^{-1} e_k)^{1/2}\right],$$

where

$$\widetilde{J}_T := \frac{1}{T}\sum_{t=1}^{T} \nabla^2 \ell(Z_t; \theta^{(t-1)}) \quad \text{and} \quad \widetilde{V}_T := \frac{1}{T}\sum_{t=1}^{T} (\nabla \ell(Z_t; \theta^{(t-1)}))(\nabla \ell(Z_t; \theta^{(t-1)}))^\top.$$

In contrast to our asymptotic normality result, the construction of this confidence interval requires the loss function to be twice differentiable. Additionally, the results of Chen et al. (2020) imply a slow rate of convergence of $\widetilde{J}_T^{-1}\widetilde{V}_T\widetilde{J}_T^{-1}$ compared to the offline estimator $\widehat{J}_T^{-1}\widehat{V}_T\widehat{J}_T^{-1}$. This directly impacts the coverage of the ASGD plug-in interval; see Kauermann and Carroll (2001, Theorem 1). More recent studies offer alternative online variance estimators without matrix inversion, for instance, Zhu et al. (2023).

ASGD *t*-stat (Ibragimov and Müller, 2010; Zhu et al., 2024): The t-statistic for ASGD, proposed by Zhu et al. (2024) and building on the general method envisioned by Ibragimov and Müller (2010), also relies on bucketing the data into B buckets, similar to HulC. The method is as follows: Fix any $B \geq 2$, and compute

$$\bar{\theta}_T^{(j)} = \mathcal{A}(Z_{j+B\lfloor (T-j)/B \rfloor}, \ldots, Z_j; \theta^{(0,j)}), \quad \text{for} \quad 1 \leq j \leq B.$$

Compute

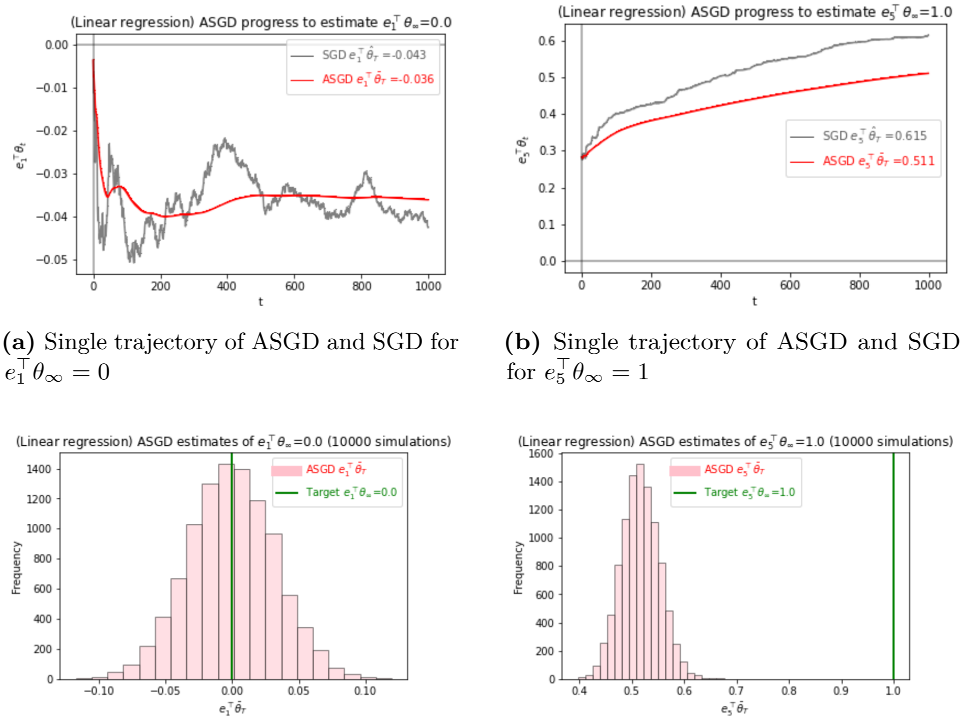
$$\widetilde{\theta}_T = \frac{1}{B}\sum_{j=1}^{B} \bar{\theta}_T^{(j)} \quad \text{and} \quad \widetilde{\sigma}_{k,T}^2 = \frac{1}{B-1}\sum_{j=1}^{B}(e_k^\top \bar{\theta}_T^{(j)} - e_k^\top \widetilde{\theta}_T)^2.$$

Report

$$\widehat{\mathrm{CI}}_{T,\alpha}^{(k)} := \left[e_k^\top \widetilde{\theta}_T \pm t_{B-1,\alpha/2} \widetilde{\sigma}_{k,T} \right],$$

where $t_{B-1,\alpha/2}$ is the $(1-\alpha/2)$-th quantile of t distribution with $B-1$ degrees of freedom. The results of Ibragimov and Müller (2010) and Zhu et al. (2024) imply that this is an asymptotically valid $(1-\alpha)$ confidence interval for any $B \geq 2$. Because there is no straightforward method to choose B, we choose $B = B^*$ from the HulC confidence interval. This means that for a 95% confidence interval, we use approximately 5 buckets.

5.3 ASGD figures



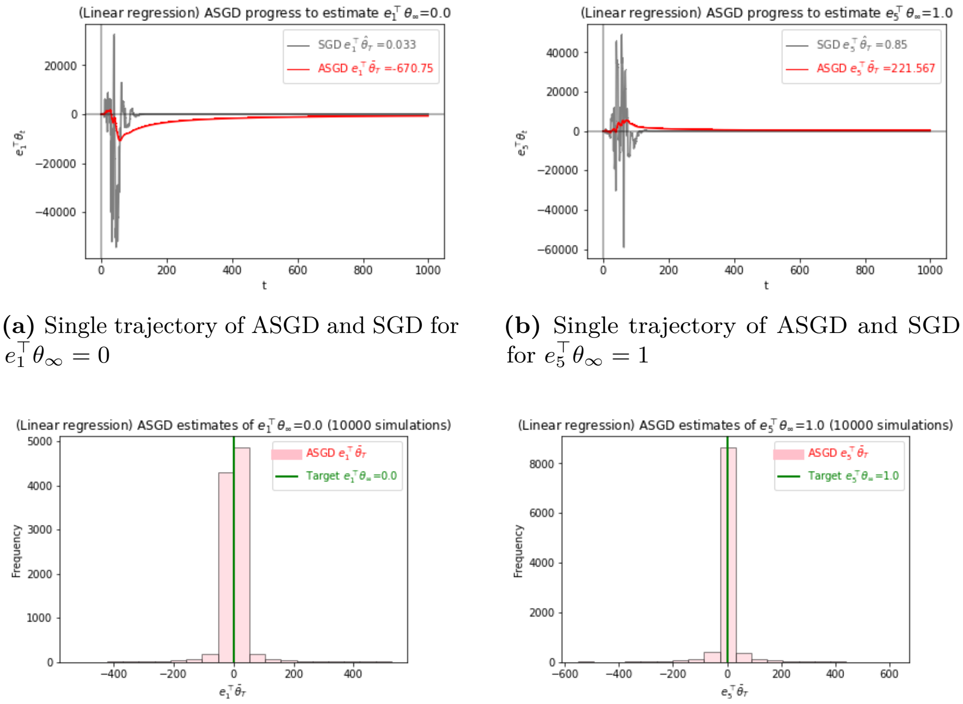
(a) Single trajectory of ASGD and SGD for $e_1^\top \theta_\infty = 0$

(b) Single trajectory of ASGD and SGD for $e_5^\top \theta_\infty = 1$

(c) Histogram of ASGD estimates for $e_1^\top \theta_\infty = 0$
(10^4 replications)

(d) Histogram of ASGD estimates for $e_5^\top \theta_\infty = 1$
(10^4 replications)

Figure 5: For the linear regression task with identity covariance, $d = 5$, $T = 10^3$, and a **small step size hyperparameter** $c = 0.01$, both the ASGD and SGD estimators for θ_∞ fail to converge. The distance from the target is worse for the larger coordinate, $e_5^\top \theta_\infty = 1$, compared to the first coordinate, $e_1^\top \theta_\infty = 0$, likely due to the initialization procedure, which favors smaller coordinates. In plot (c), which presents a histogram of $10,000$ repetitions of ASGD, there is no systematic bias: the mean ASGD estimates for $e_1^\top \theta_\infty = 0$ is -0.0006. However, there is systematic bias for $e_5^\top \theta_\infty$ (as well as all larger coordinates), as seen in (d): the mean ASGD estimate for $e_5^\top \theta_\infty = 1$ is 0.516.



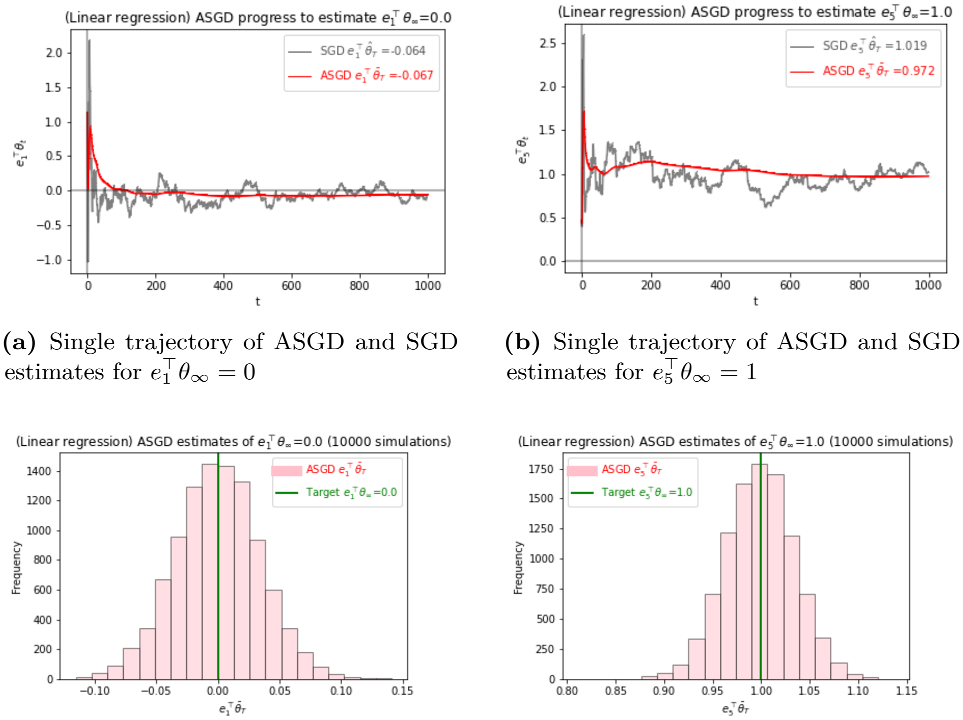
(a) Single trajectory of ASGD and SGD for $e_1^\top \theta_\infty = 0$

(b) Single trajectory of ASGD and SGD for $e_5^\top \theta_\infty = 1$

(c) Histogram of ASGD estimates for $e_1^\top \theta_\infty = 0$
(10^4 replications)

(d) Histogram of ASGD estimates for $e_5^\top \theta_\infty = 1$
(10^4 replications)

Figure 6: For the linear regression task with identity covariance, $d = 5$, $T = 10^3$, and a **large step size hyperparameter** $c = 2$, the SGD estimator tends to converge, but not the ASGD estimator, due to the large initial "wrong SGD points" at the start of the trajectory. The ASGD estimator is off by a large margin from the target parameter for both $e_1^\top \theta_\infty = 0$ (a) and $e_5^\top \theta_\infty = 1$ (b). In addition, like the case when c is small (Figure 5), there is systematic bias: in the histograms of 10^4 replications, shown in plots (c) and (d), which both exclude the largest 1% outliers, the mean ASGD estimates for $e_1^\top \theta_\infty = 0$ and $e_5^\top \theta_\infty = 1$ are respectively 4.296 and 4.687.



(a) Single trajectory of ASGD and SGD estimates for $e_1^\top \theta_\infty = 0$

(b) Single trajectory of ASGD and SGD estimates for $e_5^\top \theta_\infty = 1$

(c) Histogram of ASGD estimates for $e_1^\top \theta_\infty = 0$
(10^4 replications)

(d) Histogram of ASGD estimates for $e_5^\top \theta_\infty = 1$
(10^4 replications)

Figure 7: For the linear regression task with identity covariance, $d = 5$, $T = 10^3$, and a **"mid-range" step size hyperparameter** $c = 0.5$, both the SGD and ASGD estimators tend to converge. The ASGD estimator appears unbiased: in the histograms of 10^4 replications in plots (c) and (d), which both exclude the largest 1% outliers, the mean ASGD estimates for $e_1^\top \theta_\infty = 0$ and $e_5^\top \theta_\infty = 1$ are respectively -0.0005 and 0.998.

Table 1: Functions employed for the experiments

Function type	$m(x)$	d	$L \times H$
linear	$1 + 2x$	1	1×32
sine	$\sin(x)$	1	2×32
sine-cosine	$\sin(x_1)\cos(5x_2) + 0.2x_3^2$	3	2×64

5.4 HulC applied to neural network

Lemma 1. *Fixing* $\mathbf{x}$*, if* $\hat{f}_H(\mathbf{x})^{(b)}, 1 \le b \le B$ *are independent random variables and*

$$\Delta \ := \ \max_{1 \le b \le B} \text{Med-Bias}_{m(x)}(\hat{f}_H(\mathbf{x})^{(b)}) \ \in \ [0, 1/2],$$

$$\mathbb{P}\left(m(\mathbf{x}) \notin \left[\min_{1 \le b \le B} \hat{f}_H(\mathbf{x})^{(b)}, \ \max_{1 \le b \le B} \hat{f}_H(\mathbf{x})^{(b)}\right]\right) \le \left(\frac{1}{2} - \Delta\right)^B + \left(\frac{1}{2} + \Delta\right)^B.$$

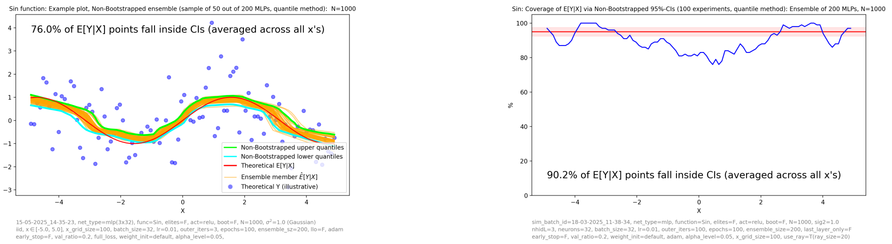
5.5 Neural network ensemble experiment settings

For each setting, we run 100 simulations: each has a new dataset $\mathcal{D}_n = \{Y_i, \mathbf{X}_i\}_{i=1}^n$. Given a particular setting combination $\mathbf{s}$, for each experiment $j \in \{1, \ldots, 100\}$, we employ one of the five ensemble methods to produce B MLPs. After each ensemble member b has finished training, we produce pointwise predictions $\hat{f}_H(\mathbf{x}_k) = \hat{\mathbb{E}}[Y_k|\mathbf{X}_k, \mathbf{s}]^{(b)}$ for $k \in \{1, \ldots, K\}$, evaluated over a fixed, equally spaced grid of input points $\mathbf{X}_k$ within the domain $\otimes_{i=1}^d[-5, 5]$. We set $K = 100$ for $d = 1$ and $K = 125$ for $d = 3$. We then produce the $(1 - \alpha) \cdot 100\%$ pointwise confidence interval for each ensemble method.

Using 100 experiments, we calculate the empirical pointwise coverage rate as $\mathrm{C}^{\mathbf{s}} \in \mathbb{R}^K$, that is, the average number of times the true function values $m(\mathbf{x}_k) = \mathbb{E}[Y_k|\mathbf{X}_k]$ have been covered by the pointwise confidence intervals produced by each method. Setting $\alpha = .05$, the target pointwise coverage is 95%. To summarize results to a single metric, we then average the pointwise coverage rates to an **average coverage**, i.e. $\bar{\mathrm{C}}^{\mathbf{s}} := \frac{1}{K}\sum_{k=1}^K \mathrm{C}_k^{\mathbf{s}} \in [0, 100]$, which should also ideally be 95%. Char et al. (2024)

Table 2: Settings employed for the experiments

Setting	**Values**
Variable settings:	
Function type	$\{\text{linear}, \text{sin}, \text{sin-cos3}\}$
Method	$\{\text{non-bootstrap}, \text{bootstrap}, \text{cheap bootstrap},$
	$\text{HulC}, \text{robust t-stat}\}$
Ensemble size B	$B = 200$ if Method $\in \{\text{non-bootstrap}, \text{bootstrap}\}$
	$B \in \{6, 10, 15, 20, 30\}$ if Method $=$ cheap bootstrap
	$B \in \{5, 6\}$ if Method $\in \{\text{HulC}, \text{robust t-stat}\}$
Sample size n	$\{100, 1000, 5000\}$
Activation function ψ	$\{\text{linear}, \text{relu}, \text{tanh}\}$
Number of epochs n_e	$\{100, 400, 1000\}$
Batch size n_{bs}	$\{32, n\}$
Learning rate η	$\{3 \cdot 10^{-2}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$
Fixed settings:	
Network type	Multi-layer perceptron (MLP) with single head
Network size	L hidden layers and H neurons per layer
Loss type	mean squared error (MSE)
Training points of $\mathbf{X} \in \mathbb{R}^d$	$\mathbf{X} \sim \text{Unif}(\otimes_{j=1}^{d}[-5, 5])$
Optimizer	Adam (learning rate $= \eta$)
Validation ratio (to estimate loss)	$0.2n$
Type I error rate α	0.05 (to produce 95% pointwise confidence intervals)



(a) Example ensemble

(b) Pointwise coverage over 100 experiments (red region is target $= 95 \pm 2.5\%$)

Figure 8: In **(a)**, we plot an example of a non-bootstrapped ensemble ($B = 200$) on the sine function, with $\eta = 0.01$, $n_{bs} = 32$, $\psi =$ relu, $n_e = 100$, and $n = 1000$ to train a 2×32 MLP. Over 100 experiments, the pointwise coverage in **(b)** averages to 90.2%, slightly below the target 95%.

5.6 Neural network ensemble additional plots

5.6.1 Non-bootstrapped Ensemble with Mini-batch Gradient Descent

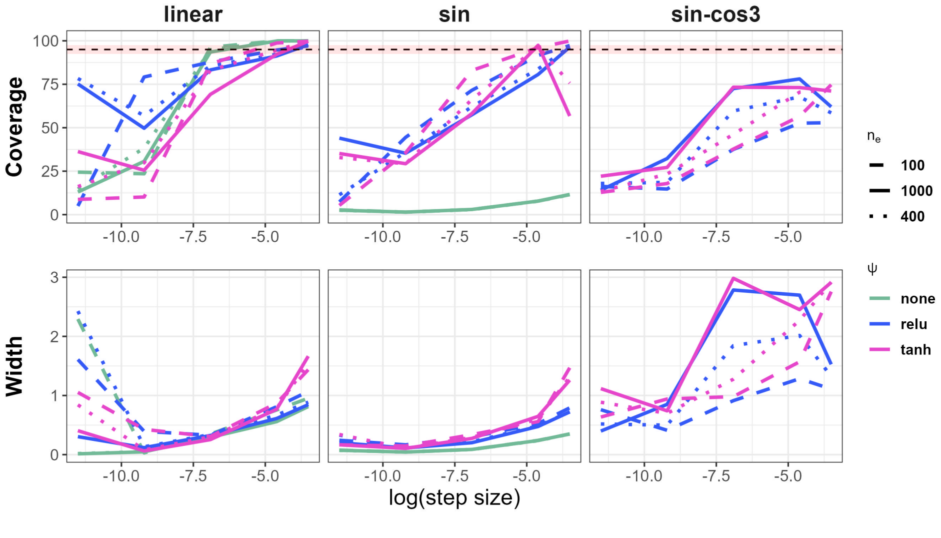


Figure 9: Over 100 experiments, we plot average coverage for non-bootstrapped ensembles ($B = 200$) on all three functions, setting $n_{bs} = 32$ and $n = 1000$, with varying step sizes, activation functions (ψ), and training time (n_e). Coverage generally falls short of the target but can depend on step size and training time. Overcoverage is often accompanied by interval widths that are excessively large.

5.6.2 Bootstrapped Ensemble with Mini-batch Gradient Descent

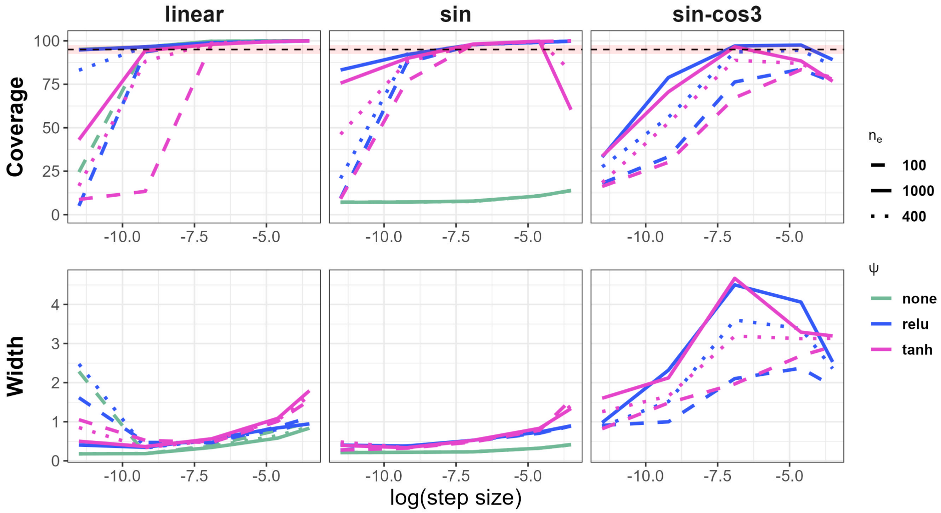


Figure 10: Over 100 experiments, we plot average coverage for non-bootstrapped ensembles ($B = 200$) on all three functions, setting $n_{bs} = 32$ and $n = 1000$, with varying step sizes, activation functions (ψ), and training time (n_e). The target coverage is achieved within a "Goldilocks zone" of step size, which is problem-specific but appears to lie on the interval around $\eta = 0.001 \approx \exp(-6.9)$.

5.6.3 Cheap Bootstrapped with Mini-batch Gradient Descent

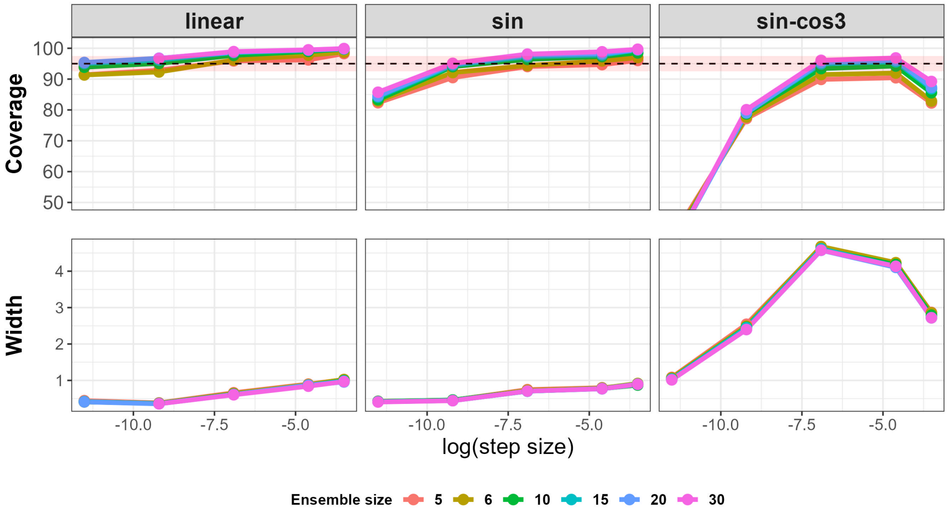


Figure 11: Over 100 experiments, we plot average coverage for cheap bootstrapped ensembles ($B \in \{6, 10, 15, 20, 30\}$) on all three functions, with $n_{bs} = 32$, $n = 1000$, ψ =relu, and $n_e = 1000$. Coverage generally reaches the target given the right step size, except in the case of sin-cos3, which requires an ensemble size $B \geq 10$.

References

Angelopoulos, A., Candes, E., and Tibshirani, R. J. (2023). Conformal PID control for time series prediction. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 23047–23074. Curran Associates, Inc.

Barber, R., Candès, E., Ramdas, A., and Tibshirani, R. (2020). The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10.

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24:123–140.

Carter, S. and Kuchibhotla, A. K. (2025). Statistical inference for online algorithms.

Char, I., Chung, Y., Abbate, J., Kolemen, E., and Schneider, J. (2024). Full shot predictions for the diii-d tokamak via deep recurrent networks.

Chen, X., Lai, Z., Li, H., and Zhang, Y. (2024). Online statistical inference for stochastic optimization via Kiefer-Wolfowitz methods. *Journal of the American Statistical Association*, 119(548):2972–2982.

Chen, X., Lee, J. D., Tong, X. T., and Zhang, Y. (2020). Statistical inference for model parameters in stochastic gradient descent. *The Annals of Statistics*, 48(1):251–273.

Chernozhukov, V., Fernández-Val, I., and Melly, B. (2022). Fast algorithms for the quantile regression process. *Empirical economics*, pages 1–27.

Chernozhukov, V., Wüthrich, K., and Yinchu, Z. (2018). Exact and robust conformal inference methods for predictive machine learning with dependent data. In Bubeck, S., Perchet, V., and Rigollet, P., editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 732–749. PMLR.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. (2021). Distributional conformal prediction. *Proceedings of the National Academy of Sciences*, 118(48):e2107794118.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26.

Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., and Kalagnanam, J. (2023). Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 459–469. ACM.

Fang, Y., Xu, J., and Yang, L. (2018). Online bootstrap confidence intervals for the stochastic gradient descent estimator. *Journal of Machine Learning Research*, 19(78):1–21.

Feldman, S., Ringel, L., Bates, S., and Romano, Y. (2023). Achieving risk control in online learning settings. *Transactions on Machine Learning Research*.

Franke, J. and Neumann, M. H. (2000). Bootstrapping neural networks. *Neural computation*, 12(8):1929–1949.

Gahbiche, M. and Pelletier, M. (2000). On the estimation of the asymptotic covariance matrix for the averaged Robbins–Monro algorithm. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 331(3):255–260.

Gibbs, I. and Candes, E. (2021). Adaptive conformal inference under distribution shift. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1660–1672. Curran Associates, Inc.

Gibbs, I. and Candès, E. J. (2024). Conformal inference for online prediction with arbitrary distribution shifts. *Journal of Machine Learning Research*, 25(162):1–36.

Heng, Q. and Lange, K. (2025). Bootstrap estimation of the proportion of outliers in robust regression. *Statistics and Computing*, 35(1):3.

Ibragimov, R. and Müller, U. K. (2010). t-statistic based correlation and heterogeneity robust inference. *Journal of Business & Economic Statistics*, 28(4):453–468.

Jensen, V., Bianchi, F. M., and Anfinsen, S. (2022). Ensemble conformalized quantile regression for probabilistic time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–12.

Katz, H. (2025). Forecasting the u.s. renewable-energy mix with an alr-bdarma compositional time-series framework.

Kauermann, G. and Carroll, R. J. (2001). A note on the efficiency of sandwich covariance matrix estimation. *Journal of the American Statistical Association*, 96(456):1387–1396.

Kuchibhotla, A. K., Balakrishnan, S., and Wasserman, L. (2024). The HulC: confidence regions from convex hulls. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(3):586–622.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413.

Lam, H. (2022). A cheap bootstrap method for fast inference. *arXiv preprint arXiv:2202.00090*.

Lam, H. and Wang, Z. (2023). Resampling stochastic gradient descent cheaply for efficient uncertainty quantification. *arXiv preprint arXiv:2310.11065*.

Lee, S., Liao, Y., Seo, M. H., and Shin, Y. (2022). Fast and robust online inference with stochastic gradient descent via random scaling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7381–7389.

Li, C. J., Mou, W., Wainwright, M., and Jordan, M. (2022). Root-SGD: Sharp nonasymptotics and asymptotic efficiency in a single algorithm. In *Conference on Learning Theory*, pages 909–981. PMLR.

Nixon, J., Lakshminarayanan, B., and Tran, D. (2020). Why are bootstrapped deep ensembles not better? In *"I Can't Believe It's Not Better!"NeurIPS 2020 workshop*.

Oliveira, R. I., Orenstein, P., Ramos, T., and Romano, J. V. (2024). Split conformal prediction and non-exchangeable data. *Journal of Machine Learning Research*, 25(225):1–38.

Pelletier, M. (2000). Asymptotic almost sure efficiency of averaged stochastic algorithms. *SIAM Journal on Control and Optimization*, 39(1):49–72.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In *International Conference on Machine Learning*, volume 70, pages 2847–2854. PMLR.

Razzhigaev, A., Mikhalchuk, M., Goncharova, E., Gerasimenko, N., Oseledets, I., Dimitrov, D., and Kuznetsov, A. (2024). Your transformer is secretly linear.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Ruppert, D. (1988). Efficient estimations from a slowly convergent Robbins-Monroe process. Technical report, Cornell University Operations Research and Industrial Engineering.

Stankeviciute, K., M. Alaa, A., and van der Schaar, M. (2021). Conformal time-series forecasting. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6216–6228. Curran Associates, Inc.

Toulis, P. and Airoldi, E. M. (2017). Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics*, 45(4):1694–1727.

Upadhya, N. (2023). Do transformers lose to linear models? Towards Data Science (Medium). Accessed: 2025-08-27.

White, H. (1989). Some asymptotic results for learning in single hidden-layer feedforward network models. *Journal of the American Statistical Association*, 84(408):1003–1013.

White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3(5):535–549.

Xu, C. and Xie, Y. (2021a). Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pages 11559–11569. PMLR.

Xu, C. and Xie, Y. (2021b). Conformal prediction interval for dynamic time-series. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11559–11569. PMLR.

Xu, C. and Xie, Y. (2023). Conformal prediction for time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11575–11587.

Zaffran, M., Feron, O., Goude, Y., Josse, J., and Dieuleveut, A. (2022). Adaptive conformal predictions for time series. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25834–25866. PMLR.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2022). Are transformers effective for time series forecasting?

Zhong, Y., Kuffner, T., and Lahiri, S. (2023). Online bootstrap inference with nonconvex stochastic gradient descent estimator. *arXiv preprint arXiv:2306.02205*.

Zhou, Y., Li, X., and Banerjee, A. (2021). Noisy truncated sgd: Optimization and generalization.

Zhu, W., Chen, X., and Wu, W. B. (2023). Online covariance matrix estimation in stochastic gradient descent. *Journal of the American Statistical Association*, 118(541):393–404.

Zhu, W., Lou, Z., Wei, Z., and Wu, W. B. (2024). High confidence level inference is almost free using parallel stochastic optimization. *arXiv preprint arXiv:2401.09346*.