UNCERTAINTY IN GRADIENT BOOSTING VIA ENSEMBLES

Anonymous authors
Paper under double-blind review

ABSTRACT

For many practical, high-risk applications, it is essential to quantify uncertainty in a model's predictions to avoid costly mistakes. While predictive uncertainty is widely studied for neural networks, the topic seems to be under-explored for models based on gradient boosting. However, gradient boosting often achieves state-of-the-art results on tabular data. This work examines a probabilistic ensemble-based framework for deriving uncertainty estimates in the predictions of gradient boosting classification and regression models. We conducted experiments on a range of synthetic and real datasets and investigated the applicability of ensemble approaches to gradient boosting models that are themselves ensembles of decision trees. Our analysis shows that ensembles of gradient boosting models successfully detect anomaly inputs while having limited ability to improve the predicted total uncertainty. Importantly, we also propose a concept of a *virtual* ensemble to get the benefits of an ensemble via only *one* gradient boosting model, which significantly reduces complexity.

1 INTRODUCTION

Gradient boosting (Friedman, 2001) is a widely used machine learning algorithm that achieves state-of-the-art results on tasks containing heterogeneous features, complex dependencies, and noisy data: web search, recommendation systems, weather forecasting, and many others (Burges, 2010; Caruana & Niculescu-Mizil, 2006; Richardson et al., 2007; Roe et al., 2005; Wu et al., 2010; Zhang & Haghani, 2015). Gradient boosting based on decision trees (GBDT) underlies such well-known libraries like XGBoost, LightGBM, and CatBoost. In this paper, we investigate the estimation of predictive uncertainty in GBDT models. Uncertainty estimation is crucial for avoiding costly mistakes in high-risk applications, such as autonomous driving, medical diagnostics, and financial forecasting. For example, in self-driving cars, it is necessary to know when the AI-pilot is confident in its ability to drive and when it is not to avoid a fatal collision. In financial forecasting and medical diagnostics, mistakes on the part of an AI forecasting or diagnostic system could either lead to large financial or reputational loss or to the loss of life. Crucially, both financial and medical data are often represented in heterogeneous tabular form — data on which GBDTs are typically applied, highlighting the relevance of our work on obtaining uncertainty estimates for GBDT models.

Approximate Bayesian approaches for uncertainty estimation have been extensively studied for neural network models (Gal, 2016; Malinin, 2019). Bayesian methods for tree-based models (Chipman et al., 2010; Linero, 2017) have also been widely studied in the literature. However, this research did not explicitly focus on studying *uncertainty estimation* and its applications. Some related work was done by Coulston et al. (2016); Shaker & Hüllermeier (2020), who examined quantifying predictive uncertainty for random forests. However, the area has been otherwise relatively under-explored, especially for GBDT models that are widely used in practice and known to outperform other approaches based on tree ensembles.

While for classification problems GDBT models already return a distribution over class labels, for regression tasks they typically yield only point predictions. Recently, this problem was addressed in the NGBoost algorithm (Duan et al., 2020), where a GBDT model is trained to return the mean and variance of a normal distribution over the target variable y for a given feature vector. However, such models only capture *data uncertainty* (Gal, 2016; Malinin, 2019), also known as *aleatoric uncertainty*, which arises due to inherent class overlap or noise in the data. However, this does not

quantify uncertainty due to the model's inherent lack of knowledge about inputs from regions either far from the training data or sparsely covered by it, known as *knowledge uncertainty*, or *epistemic uncertainty* (Gal, 2016; Malinin, 2019). One class of approaches for capturing *knowledge uncertainty* are Bayesian ensemble methods, which have recently become popular for estimating predictive uncertainty in neural networks (Depeweg et al., 2017; Gal & Ghahramani, 2016; Kendall et al., 2017; Lakshminarayanan et al., 2017; Maddox et al., 2019; Smith & Gal, 2018). A key feature of ensemble approaches is that they allow overall uncertainty to be decomposed into *data uncertainty* and *knowledge uncertainty* within an interpretable probabilistic framework (Depeweg et al., 2017; Gal, 2016; Malinin, 2019). Ensembles are also known to yield improvements in predictive performance.

This work examines ensemble-based uncertainty-estimation for GBDT models. The contributions are as follows. First, we consider generating ensembles using both classical Stochastic Gradient Boosting (SGB) as well as the recently proposed Stochastic Gradient Langevin Boosting (SGLB) (Ustimenko & Prokhorenkova, 2020). Importantly, SGLB allows us to guarantee that the models are asymptotically sampled from a true Bayesian posterior. Second, we show that using SGLB we can construct a *virtual* ensemble using only *one* gradient boosting model, significantly reducing the computational complexity. Third, to understand the attributes of using ensembles-based uncertainty estimation in GBDT models, we conduct extensive analysis on several synthetic datasets. Finally, we evaluate the proposed approach on a range of real regression and classification datasets. Our results show that this approach successfully enables the detection of anomalous out-of-domain inputs. Importantly, our solution is easy to combine with any implementation of GBDT.

2 PRELIMINARIES

Uncertainty Estimation via Bayesian Ensembles In this work we consider uncertainty estimation within the standard Bayesian ensemble-based framework (Gal, 2016; Malinin, 2019). Here, model parameters $\boldsymbol{\theta}$ are considered random variables and a prior $\mathtt{p}(\boldsymbol{\theta})$ is placed over them to compute a posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ via Bayes' rule:

$$\mathtt{p}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathtt{p}(\mathcal{D}|\boldsymbol{\theta})\mathtt{p}(\boldsymbol{\theta})}{\mathtt{p}(\mathcal{D})}\,. \tag{1}$$

where $\mathcal{D} = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ is the training dataset. Each set of parameters can be considered a hypothesis or explanation about how the world works. Samples from the posterior should yield explanations consistent with the observations of the world contained within the training data $\mathcal{D}$. However, on data far from $\mathcal{D}$ each set of parameters can yield different predictions. Therefore, estimates of *knowledge uncertainty* can be obtained by examining the diversity of predictions.

Consider an ensemble of probabilistic models $\{\mathtt{P}(y|\boldsymbol{x};\boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ sampled from the posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$. Each model $\mathtt{P}(y|\boldsymbol{x},\boldsymbol{\theta}^{(m)})$ yields a *different* estimate of *data uncertainty*, represented by the entropy of its predictive distribution (Malinin, 2019). Uncertainty in predictions due to *knowledge uncertainty* is expressed as the level of spread, or "disagreement", of models in the ensemble (Malinin, 2019). Note that exact Bayesian inference is often intractable, and it is common to consider either an explicit or implicit approximation $\mathtt{q}(\boldsymbol{\theta})$ to the true posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$. While a range of approximations has been explored for neural network models (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Maddox et al., 2019)[1], to the best of our knowledge, limited work has explored Bayesian inference for gradient-boosted trees. Given $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$, the *predictive posterior* of the ensemble is obtained by taking the expectation with respect to the models in the ensemble:

$$\mathtt{P}(y|\boldsymbol{x},\mathcal{D}) = \mathbb{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathtt{P}(y|\boldsymbol{x};\boldsymbol{\theta})\big] \approx \frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(y|\boldsymbol{x};\boldsymbol{\theta}^{(m)}),\ \boldsymbol{\theta}^{(m)} \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D})\,. \tag{2}$$

The entropy of the predictive posterior estimates *total uncertainty* in predictions:

$$\mathcal{H}\big[\mathtt{P}(y|\boldsymbol{x},\mathcal{D})\big] = \mathbb{E}_{\mathtt{P}(y|\boldsymbol{x},\mathcal{D})}\big[-\ln \mathtt{P}(y|\boldsymbol{x},\mathcal{D})\big]\,. \tag{3}$$

Total uncertainty is due to both *data uncertainty* and *knowledge uncertainty*. However, in applications like active learning (Kirsch et al., 2019) and out-of-domain detection it is desirable to estimate

[1] A full overview is available in (Ashukha et al., 2020; Ovadia et al., 2019).

knowledge uncertainty separately. The sources of uncertainty can be decomposed by considering the *mutual information* between the parameters $\boldsymbol{\theta}$ and the prediction y (Depeweg et al., 2017):

$$\begin{aligned}\underbrace{\mathcal{I}\big[y, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{D}\big]}_{\text{Knowledge Uncertainty}} &= \underbrace{\mathcal{H}\big[\mathtt{P}(y|\boldsymbol{x}, \mathcal{D})\big]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathtt{P}(y|\boldsymbol{x}; \boldsymbol{\theta})]\big]}_{\text{Expected Data Uncertainty}} \\ &\approx \mathcal{H}\big[\frac{1}{M}\sum_{m=1}^{M} \mathtt{P}(y|\boldsymbol{x}; \boldsymbol{\theta}^{(m)})\big] - \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}[\mathtt{P}(y|\boldsymbol{x}; \boldsymbol{\theta}^{(m)})]\,.\end{aligned} \tag{4}$$

This is expressed as the difference between the entropy of the predictive posterior, a measure of *total uncertainty*, and the expected entropy of each model in the ensemble, a measure of *expected data uncertainty*. Their difference is a measure of ensemble diversity and estimates *knowledge uncertainty*.

Unfortunately, when considering ensembles of probabilistic *regression* models $\{\mathtt{p}(y|\boldsymbol{x}; \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ over continuous-valued target $y \in \mathbb{R}$, it is no longer possible to obtain tractable estimates of the (differential) entropy of the predictive posterior, and, by extension, mutual information. In this cases uncertainty estimates can instead derived via the law of total variation:

$$\underbrace{\mathbb{V}_{\mathtt{p}(y|\boldsymbol{x}, \mathcal{D})}[y]}_{\text{Total Uncertainty}} = \underbrace{\mathbb{V}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathbb{E}_{\mathtt{p}(y|\boldsymbol{x}, \boldsymbol{\theta})}[y]\big]}_{\text{Knowledge Uncertainty}} + \underbrace{\mathbb{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathbb{V}_{\mathtt{p}(y|\boldsymbol{x}, \boldsymbol{\theta})}[y]\big]}_{\text{Expected Data Uncertainty}}\,. \tag{5}$$

This is conceptually similar to the decomposition (4) obtained via mutual information. For an ensemble of probabilistic regression models which parameterize the normal distribution, and where each models yields a mean and standard-deviation, the total variance can be computed as follows:

$$\underbrace{\mathbb{V}_{\mathtt{p}(y|\boldsymbol{x}, \mathcal{D})}[y]}_{\text{Total Uncertainty}} \approx \underbrace{\frac{1}{M}\sum_{m=1}^{M}\Big[\Big(\sum_{m=1}^{M}\frac{\mu_m}{M}\Big) - \mu_m\Big]^2}_{\text{Knowledge Uncertainty}} + \underbrace{\frac{1}{M}\sum_{m=1}^{M}\sigma_m^2}_{\text{Expected Data Uncertainty}}, \{\mu_m, \sigma_m\} = f(\boldsymbol{x}; \boldsymbol{\theta}^{(m)}) \tag{6}$$

However, while these measures are tractable, they are based on only first and second moments, and may therefore miss high-order details in the uncertainty. They are also not scale-invariant, which can cause issues is the scale of prediction on in-domain and out-of-domain data is very different.

Gradient boosting is a powerful machine learning technique especially useful on tasks containing heterogeneous features. It iteratively combines weak models, such as decision trees, to obtain more accurate predictions. Formally, given a dataset $\mathcal{D}$ and a loss function $L : \mathbb{R}^2 \to \mathbb{R}$, the gradient boosting algorithm (Friedman, 2001) iteratively constructs a model $F : X \to \mathbb{R}$ to minimize the empirical risk $\mathcal{L}(F|\mathcal{D}) = \mathbb{E}_{\mathcal{D}}[L(F(\boldsymbol{x}), y)]$. At each iteration t the model is updated as:

$$F^{(t)}(\boldsymbol{x}) = F^{(t-1)}(\boldsymbol{x}) + \epsilon h^{(t)}(\boldsymbol{x})\,, \tag{7}$$

where $F^{(t-1)}$ is a model constructed at the previous iteration, $h^{(t)}(\boldsymbol{x}) \in \mathcal{H}$ is a weak learner chosen from some family of functions $\mathcal{H}$, and ϵ is learning rate. The weak learner $h^{(t)}$ is usually chosen to approximate the negative gradient $-g^{(t)}(\boldsymbol{x}, y) := -\frac{\partial L(y,s)}{\partial s}\big|_{s=F^{(t-1)}(\boldsymbol{x})}$:

$$h^{(t)} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\big[\big(-g^{(t)}(\boldsymbol{x}, y) - h(\boldsymbol{x})\big)^2\big]. \tag{8}$$

A weak learner $h^{(t)}$ is associated with parameters $\boldsymbol{\phi}^{(t)} \in \mathbb{R}^d$. We write $h^{(t)}(\boldsymbol{x}, \boldsymbol{\phi}^{(t)})$ to reflect this dependence. The set of weak learners $\mathcal{H}$ often consists of shallow decision trees, which are models that recursively partition the feature space into disjoint regions called leaves. Each leaf R_j of the tree is assigned to a value, which is the estimated response y in the corresponding region. We can write $h(\boldsymbol{x}, \boldsymbol{\phi}^{(t)}) = \sum_{j=1}^{d} \phi_j^{(t)} \mathbf{1}_{\{\boldsymbol{x} \in R_j\}}$, so the decision tree is a linear function of $\boldsymbol{\phi}^{(t)}$. The final GBDT model F is a sum of decision trees (7) and the parameters of the full model are denoted by $\boldsymbol{\theta}$.

For classification tasks, a model yields estimates *data uncertainty* if it is trained via negative log-likelihood and provides a distribution over class labels. However, classic GBDT regression models yield point predictions, and there has been little research devoted to estimating predictive uncertainty. Recently, this issue was addressed by Duan et al. (2020) via an algorithm called NGBoost (Natural

Gradient Boosting), which allows estimating *data uncertainty*. NGBoost simultaneously estimates the parameters of a conditional distribution $\mathtt{p}(y|\boldsymbol{x}, \boldsymbol{\theta})$ over the target y given the features $\boldsymbol{x}$, by optimizing a proper scoring rule. Typically, a normal distribution over y is assumed and negative log-likelihood is taken as a scoring rule. Formally, given an input $\boldsymbol{x}$, the model F predicts two parameters of normal distribution - the mean μ and the logarithm of the standard deviation $\log \sigma$. The loss function is the expected negative log-likelihood:[2]

$$\mathtt{p}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) = \mathcal{N}(y|\mu^{(t)}, \sigma^{(t)}), \quad \{\mu^{(t)}, \log \sigma^{(t)}\} = F^{(t)}(\boldsymbol{x})\,. \tag{9}$$

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \mathbb{E}_{\mathcal{D}}[-\log \mathtt{p}(y|\boldsymbol{x}, \boldsymbol{\theta})] = -\frac{1}{N}\sum_{i=1}^{N} \log \mathtt{p}(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{\theta})\,. \tag{10}$$

Note that $\boldsymbol{\theta}$ denotes the concatenation of two parameter vectors used to predict μ and $\log \sigma$.

3 Generating ensembles of GDBT models

As discussed in Section 2, *knowledge uncertainty* can be estimated by considering the an ensemble of models $\{\mathtt{p}(y|\boldsymbol{x}; \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ sampled from the posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$. The level of diversity or "disagreement" between the models is an estimate of *knowledge uncertainty*. In this section we consider three approaches to generating an ensemble of GBDT models. We emphasize that this section discusses *ensembles of GBDT models*, where a *each* GBDT model is itself an *ensemble of trees*.

SGB ensembles One way to generate an ensemble is to consider several independent models generated via Stochastic Gradient Boosting (SGB). Stochasticity is added to GBDT models via random subsampling of the data at every iteration (Friedman, 2002). Specifically, at each iteration of (8) we select a subset of training objects $\mathcal{D}'$ (via bootstrap or uniformly without replacement), which is smaller than the original training dataset $\mathcal{D}$, and use $\mathcal{D}'$ to fit the next tree instead of $\mathcal{D}$. The fraction of chosen objects is called *sample rate*. This implicitly injects noise into the learning process, effectively inducing a distribution $\mathtt{q}(\boldsymbol{\theta})$ over such models. Thus, an *SGB ensemble* is an ensemble of independent models $\{\boldsymbol{\theta}^{(m)}\}_{m=1}^{M}$ built according to SGB with different random seeds for sub-sampling data. Unfortunately, there are no guarantees on how well the distribution $\mathtt{q}(\boldsymbol{\theta})$ estimates the true posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$.

SGLB ensembles Remarkably, there is a way to sample GBDT models from the true posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ via a recently proposed Stochastic Gradient Langevin Boosting (SGLB) algorithm (Ustimenko & Prokhorenkova, 2020). SGLB combines gradient boosting with stochastic gradient Langevin dynamics (Raginsky et al., 2017) in order to achieve convergence to the global optimum even for non-convex loss functions. The algorithm has two differences compared with SGB. First, Gaussian noise is explicitly injected into the gradients, so (8) is replaced by:

$$h^{(t)} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\left(-g^{(t)}(\boldsymbol{x}, y) - h(\boldsymbol{x}, \boldsymbol{\phi}) + \nu\right)^2\right], \nu \sim \mathcal{N}\left(0, \frac{2}{\beta\epsilon} I_{|\mathcal{D}|}\right)\,, \tag{11}$$

where β is the inverse diffusion temperature and $I_{|\mathcal{D}|}$ is an identity matrix. This random noise ν helps to explore the solution space in order to find the global optimum and the diffusion temperature controls the level of exploration. Second, the update (7) is modified as:

$$F^{(t)}(\boldsymbol{x}) = (1 - \gamma\epsilon)F^{(t-1)}(\boldsymbol{x}) + \epsilon h^{(t)}(\boldsymbol{x}, \boldsymbol{\phi}^{(t)})\,, \tag{12}$$

where γ is regularization parameter. If the number of all possible trees is finite (a natural assumption given that the training dataset is finite), then the SGLB parameters $\boldsymbol{\theta}^{(t)}$ at each iteration form a Markov chain that weakly converges to the stationary distribution, also called the invariant measure:

$$p^*_{\beta}(\boldsymbol{\theta}) \propto \exp(-\beta\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) - \beta\gamma\|\Gamma\boldsymbol{\theta}\|_2^2)\,, \tag{13}$$

where $\Gamma = \Gamma^T > 0$ is an implicitly defined regularization matrix which depends on a particular tree construction algorithm (Ustimenko & Prokhorenkova, 2020).

While Ustimenko & Prokhorenkova (2020) used the weak convergence to (13) to prove the global convergence, we apply this to enable sampling from the true posterior. For this purpose, we set

[2]Since GBDT model is determined by θ, we use notation $\mathcal{L}(F|\mathcal{D})$ and $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ interchangeably.

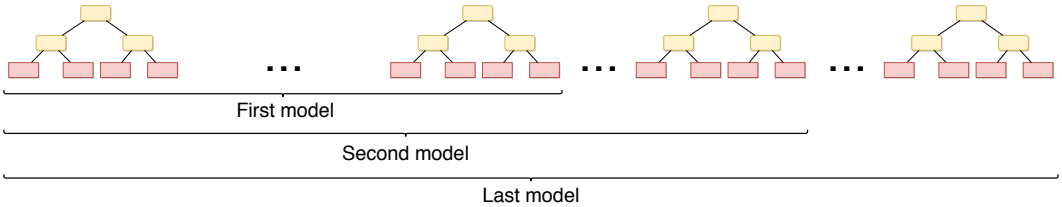


Figure 1: Virtual ensemble

$\beta = |\mathcal{D}|$ and $\gamma = \frac{1}{2|\mathcal{D}|}$. For the negative log-likelihood loss function (10) the invariant measure (13) can be expressed as:

$$p^*_\beta(\boldsymbol{\theta}) \propto \exp\left(\log \mathrm{p}(\mathcal{D}|\boldsymbol{\theta}) - \frac{1}{2}\|\Gamma\boldsymbol{\theta}\|_2^2\right) \propto \mathrm{p}(\mathcal{D}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})\,, \tag{14}$$

which is proportional to the true posterior distribution $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$ under Gaussian prior $\mathrm{p}(\boldsymbol{\theta}) = \mathcal{N}(0, \Gamma)$. Thus, an *SGLB ensemble* is an ensemble of independent models $\{\boldsymbol{\theta}^{(m)}\}_{m=1}^M$ generated according to the SGLB algorithm using different random seeds. In this case, asymptotically, models are sampled from the true posterior $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$.

Virtual SGLB ensembles While SGB and SGLB yield ensembles of independent models, their time and space complexity is M times larger than that of a single model, which is a significant overhead. Consequently, generating an ensemble requires either significantly increasing complexity or sacrificing the quality by reducing the number of training iterations. To address this, we introduce the concept of a *virtual ensemble* that enables generating an ensemble using *only one* model. This is possible since a GBDT model is itself an *ensemble of trees*. However, in contrast to random forests formed by independent trees (Shaker & Hüllermeier, 2020), the sequential nature of GBDT models implies that all trees are dependent and individual trees cannot be considered as separate models. Hence, we use "truncated" sub-models of a single GBDT model as elements of an ensemble, as illustrated in Figure 1. Notably, a virtual ensemble can be obtained using any already constructed GBDT model. Below we formally describe this procedure applied to SGLB models since in this case we can guarantee asymptotically sampling from the true posterior $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$.

Each "truncated" model is described by the vector of parameters $\boldsymbol{\theta}^{(t)}$. As the parameters $\boldsymbol{\theta}^{(t)}$ at each iteration of the SGLB algorithm form a Markov chain that weakly convergences to the stationary distribution (14), we can consider using them as an ensemble of models. However, unlike parameters taken from different SGLB trajectories, these will have a high degree of correlation, which adversely affects the ensemble's quality. This problem can be overcome by retaining only every K-th set of parameters. Formally, fix $K \geq 1$ and consider a set of models $\Theta_{T,K} = \{\boldsymbol{\theta}^{(Kt)}, \left[\frac{T}{2K}\right] \leq t \leq \left[\frac{T}{K}\right]\}$, i.e., we add to $\Theta_{T,K}$ every K-th model obtained while constructing *one* SGLB model using T iterations of gradient boosting. Choosing larger values of K allows us to reduce the correlation between samples from the SGLB Markov chain. Furthermore, we do not include to the ensemble the models $\boldsymbol{\theta}^{(t)}$ with $t < T/2$ as (14) holds only asymptotically. The set of $M = \left[\frac{T}{2K}\right]$ models $\Theta_{T,K}$ is called a *virtual ensemble*. Note that virtual ensembles behave similarly to true ensembles in the limit (for large K and T).

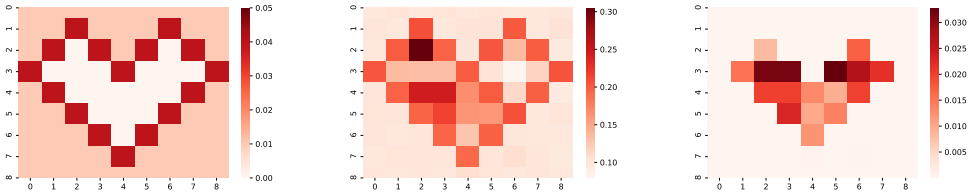
Importantly, we can compute the prediction of $\Theta_{T,K}$ with the same computation time as one $\boldsymbol{\theta}^{(T)}$. Indeed, when computing the prediction of one model, we have to sum up the predictions made by individual trees. To get the virtual ensemble, we only have to store the partial sums. For SGLB we also have to account for regularization (12), see Appendix A for a particular formula.

4 Analysis on Synthetic Data

In this section, we analyze how ensemble algorithms discussed in Section 3 perform on synthetic data. The aim is to understand the attributes of ensembles of GBDT models for estimating *data* and *knowledge uncertainty* in a controllable setting.

GBDT models are usually applied to tabular data, where features are often categorical. Hence, we first generate a dataset with each example described by two categorical features x_1, x_2 with 9 values each,

resulting in 81 possible combinations. The target depends on the features as $y = a(x_1, x_2) + \varepsilon(x_1, x_2)$, where $\varepsilon(x_1, x_2) \sim \mathcal{N}(0, b(x_1, x_2))$ and $a(x_1, x_2), b(x_1, x_2)$ are some deterministic functions. The values for $a(x_1, x_2)$ are randomly generated and the values for $b(x_1, x_2)$ are shown on Figure 2(a). We generate a heart-shaped dataset with this distribution: inside the heart (white region on Figure 2(a)) there are no training points, for the other cells we have 1000 examples.

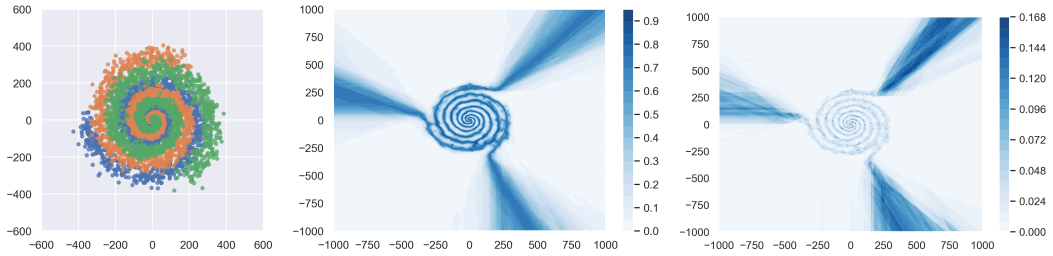


(a) True Data Uncertainty (b) Estimated Data Uncertainty (c) Knowledge Uncertainty

Figure 2: Uncertainty for synthetic regression dataset with two categorical features. Inside the heart (white region on the first figure) there are no training examples.

We train an ensemble of 10 SGLB models and observe the following effects. First, Figure 2(b) shows that the models correctly capture the data uncertainty in all cells containing training examples. At the same time, arbitrary values can be predicted inside the heart, as no training data constrain the models' behavior there. Second, Figure 2(c) shows that we can detect regions that are out-of-domain and are not covered by training data via estimates of knowledge uncertainty. Notably, the separation is perfect, as there is no trace of the original heart border.

To further analyze ensembles of GBDT models, we apply them to a classification task with continuous-valued features. This setting is much harder for gradient boosting trees.[3] We consider a 3-class spiral dataset shown on Figure 3(a). Figure 3(b) shows the total uncertainty estimated with SGLB ensemble, while Figure 3(c) demonstrates knowledge uncertainty. We observe several effects. First, total uncertainty correctly detects class boundaries and 'sectors' of input space outside the training dataset. Second, looking at these 'sectors' of high uncertainty, we can better understand how GBDT ensembles work: as decision trees are *discriminative functions* (Bishop, 2006), if features have values outside the training domain, then the prediction is the same as for the "closest" elements in the dataset. In other words, the models' behavior on the boundary of the dataset is further extended to the outer regions. Third, estimates of knowledge uncertainty allow discrimination between out-of-domain regions and class boundaries. However, we still can see traces of the class boundaries in Figure 3(c). A possible reason is the fact that for real-valued features, near the class borders, the splitting values may vary across all models in the ensemble, resulting in non-zero estimates of knowledge uncertainty due to decision-boundary 'jitter'.



(a) Spiral Dataset (b) Total Uncertainty (c) Knowledge Uncertainty

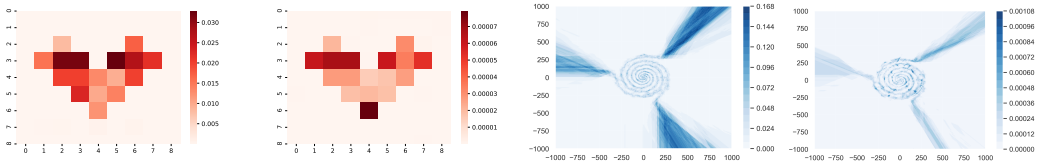
Figure 3: SGLB.

On both "heart" and "spiral" datasets, we observed that the absolute values of *knowledge uncertainty* are much smaller than of *data uncertainty* and therefore contribute very little to *total uncertainty*.

[3]To partially mitigate the difficulties, we use coordinates in rotated axes and radius as additional features.

Thus, we expect that while *knowledge uncertainty* is especially useful for detecting anomalous inputs, the proposed approaches will contribute little to error detection on top of estimates of data uncertainty provided by single models.

Finally, on Figure 4, we compare the performance of 'true' SGLB ensembles with the virtual SGLB ensembles (vSGBL) on both the "heart" and "spiral" datasets. The virtual ensemble is ten times cheaper to train and infer, but the ensemble members are strongly correlated. We observe that on the "heart" dataset, vSGLB perfectly detects regions not covered by training data. However, the absolute values of knowledge uncertainty in this region are much smaller than for SGLB, which can be explained by the correlations. The "spiral" dataset is more challenging for both SGLB and vSGLB. While having qualitatively similar behavior, virtual ensembles struggle to detect out-of-domain regions and separate them from class boundaries. In all cases, the absolute values of knowledge uncertainty are far lower than for 'true' SGLB ensembles. This shows that while vSGLB yields very cheap estimates of knowledge uncertainty by exploiting the 'ensemble of trees' structure of GBDT models, the quality of these estimates is inferior to ensembles of independent models.



(a) Heart SGLB KU (b) Heart vSGLB KU (c) Spiral SGLB KU (d) Spiral vSGLB KU

Figure 4: Comparison of SGLB and vSGLB knowledge uncertainty estimates

5 Experiments on classification and regression datasets

In this section, we evaluate the performance of ensembles of GBDT models on a range of classification and regression tasks, focusing on their ability to detect errors and out-of-domain inputs.

Experimental setup Our implementation of all GBDT models is based on the CatBoost library that is known to achieve state-of-the-art results in a variety of tasks (Prokhorenkova et al., 2018). Classification models yield a probability distribution over binary class labels, while the regression models yield the mean and variance of the normal distribution, as discussed in Section 2. All models are trained by optimizing the negative log-likelihood.[4] We consider SGB and SGBL single models as the baselines and examine all ensemble methods defined in Section 3. Ensembles of SGB and SGLB models consist of 10 independent (with different seeds) models with 1K trees each. The virtual ensemble vSGLB is obtained from one model with 1K trees, where each 50th model from the interval $[500, 1000]$ is added to the ensemble. Thus, vSGLB has the same computational and space complexity as just one SGB or SGLB model. Hyper-parameters are tuned by random-search, for details see Appendix B.2.

We compare the algorithms on several classification and regression tasks (Gal & Ghahramani, 2016; Prokhorenkova et al., 2018), the description of which is available in Appendix B.3.

While not being the focus of the current research, Random Forest models are naturally suitable for ensemble approaches. Hence, we conduct additional experiments on comparing our results with these models in Appendix D.

Detection of errors and anomalous inputs We analyze whether measures of *total* and *knowledge uncertainty* can be used to detect errors and out-of-domain inputs. Error detection can be evaluated via the Prediction-Rejection Ratio (PRR) (Malinin, 2019; Malinin et al., 2020), which measures how well uncertainty estimates correlate with errors and rank-order them. The best value is 100, random is

[4] In Appendix B.1, we compare our implementation with the original NGBoost and Deep Ensembles in terms of NLL (negative log-likelihood) and RMSE. Our implementation has comparable performance to the existing methods and is especially good for NLL optimization.

0. Out-of-domain (OOD) detection is assessed via area under the ROC curve (AUR-ROC) (Hendrycks & Gimpel, 2016). For OOD detection, we need an OOD test-set. However, obtaining 'real' OOD examples for the datasets considered in this work is challenging, so we instead create synthetic OOD data as follows. For each dataset, we take its test set as the in-domain examples and sample an OOD dataset of the same size from the Naval dataset to get out-of-domain (OOD) ones. For the Naval dataset itself, we sample OOD from the horizontally stacked Kin8nm and Protein datasets. Categorical features are sampled from a uniform distribution. *Total* and *knowledge uncertainty* are estimated via entropy of the predictive posterior (3) and mutual information (4) for classification models and via *total variance* and *variance of the mean* (5) for regression ones.

Test errors can occur due to both noise and lack of knowledge, so we expect that ranking elements by *total uncertainty* would give better values of PRR. Table 1 shows that measures of *total uncertainty* consistently yield better PPR results across all datasets. This is consistent with results obtained for ensembles of neural network models (Lakshminarayanan et al., 2017; Malinin, 2019; Malinin & Gales, 2019; Malinin et al., 2020). However, ensembles do not outperform single models. We believe this occurs for two reasons. First, due to the additive nature of boosting, GDBT models are already ensembles. Second, as we have discussed in Section 4, for GBDT models, estimates of *knowledge uncertainty* obtained via the approaches considered here contribute little to estimates of *total uncertainty*.

Table 1: Detection of errors and OOD examples for regression and classification tasks

Dataset		Single		Ensemble			Single		Ensemble		
		SGB	SGLB	SGB	SGLB	vSGLB	SGB	SGLB	SGB	SGLB	vSGLB
		Classification % PRR (↑)					Classification % AUC-ROC (↑)				
Click	TU	**43**	**43**	**43**	**43**	**43**	40	39	40	40	39
	KU	—	—	25	25	10	—	—	**87**	**87**	78
Appetency	TU	70	**72**	71	71	71	15	19	16	19	17
	KU	—	—	81	82	67	—	—	53	72	**85**
Churn	TU	49	49	**50**	**50**	49	96	91	97	96	94
	KU	—	—	38	36	22	—	—	**99**	**99**	98
Upselling	TU	54	**56**	**56**	**56**	**56**	35	61	44	61	48
	KU	—	—	50	47	17	—	—	81	**92**	80
Internet	TU	77	76	76	76	**78**	77	76	76	77	71
	KU	—	—	70	70	43	—	—	**93**	**93**	84
Kick	TU	43	43	**44**	**44**	43	49	49	69	54	58
	KU	—	—	34	34	18	—	—	**99**	97	95
Dataset		Regression % PRR (↑)					Regression % AUC-ROC (↑)				
Protein	TU	49	47	**51**	**50**	48	83	87	**89**	**92**	**91**
	KU	—	—	30	29	2	—	—	**92**	**92**	80
Concrete	TU	**38**	**38**	**38**	**38**	30	83	84	84	84	78
	KU	—	—	29	28	25	—	—	91	**93**	65
Naval-p	TV	80	80	**85**	**84**	83	97	97	99	99	86
	KU	—	—	63	64	65	—	—	**99**	**99**	54
Energy	TU	**54**	**54**	**55**	**54**	**53**	67	66	83	84	60
	KU	—	—	31	31	39	—	—	**100**	**100**	60
Power-p	TU	28	**32**	30	**32**	**32**	37	42	43	46	47
	KU	—	—	8	12	14	—	—	**76**	73	68
Year	TU	61	61	62	**62**	61	61	59	61	62	60
	KU	—	—	29	29	20	—	—	63	**66**	52

In contrast, Table 1 shows that measures of *knowledge uncertainty* yield superior OOD detection performance compared to *total uncertainty* in terms of AUC-ROC, which is consistent with results for non-GBDT models (Malinin, 2019; Malinin & Gales, 2019; Malinin et al., 2020).[5] The results also show that SGB and SGLB ensembles performed almost equally well, with SGLB having marginally better results. At the same time, virtual ensembling (vSGLB) performed consistently worse (with one exception) than SGB/SGLB ensembles, which is explained by the presence of strong correlations between the models in a virtual ensemble. However, in all classification tasks, estimates of *knowledge*

[5]Note that single models do not allow distinguishing between the types of uncertainty.

uncertainty provided by vSGLB nevertheless outperform uncertainty estimates derived from *single* SGB and SGLB models. This shows that for classification tasks, useful measures of knowledge uncertainty *can* be derived from a *single* SGLB model by interpreting it as a virtual ensemble at *no additional computational or memory cost*.

6 Conclusion

This work examined principled, ensemble-based uncertainty-estimation for GBDT models. Two main approaches to generating ensembles of GDBT models, where each model is itself an ensemble of trees, were considered — Stochastic Gradient Boosting (SGB) and Stochastic Gradient Langevin Boosting (SGLB). Based on SGLB, we propose constructing a *virtual* ensemble (vSGLD) by exploiting the 'ensemble-of-trees' nature of GBDT models. Properties of the estimates of *total*, *data*, and *knowledge uncertainty* derived from these ensembles were first analyzed on synthetic data. It was shown that the proposed approach can successfully detect anomalous inputs and is especially successful on tabular data. On continuous data, detecting *knowledge uncertainty* is still possible, but it becomes harder to differentiate it with *data uncertainty* due to decision-boundary 'jitter'. Further experiments on a wide range of classification and regression datasets showed that while ensembles of GDBT models do not offer much advantage in terms of error detection, as each model is already an ensemble of trees, they do yield useful measures of *knowledge uncertainty*, which enables out-of-domain detection in both regression and classification tasks. Notably, measures of *knowledge uncertainty*, which can only be obtained via ensembles, achieve far better OOD detection performance than measures of *total uncertainty*. It is also shown that while there is little practical difference between SGB and SGLB ensembles, vSGLB performs noticeably worse. However, for classification tasks, vSGLB still yields useful measures of *knowledge uncertainty* at the computational time and space complexity of a *single* SGLB model. Thus, vSGLB allows us to derive the benefits of an ensemble at no additional computational and memory cost.

References

UCI datasets. https://github.com/yaringal/DropoutUncertaintyExps/tree/master/UCI_Datasets.

KDD internet usage data. https://www.kdd.org/kdd-cup/view/kdd-cup-2012-track-2, 1998.

Don't get kicked! https://www.kaggle.com/c/DontGetKicked, 1998.

Kdd cup 2009: Customer relationship prediction. https://www.kdd.org/kdd-cup/view/kdd-cup-2009/Data, 2009.

Kdd cup 2012 (track 2): Predict the click-through rate of ads given the query and user information. https://www.kdd.org/kdd-cup/view/kdd-cup-2012-track-2, 2012.

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. URL https://openreview.net/forum?id=BJxI5gHKDr.

Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11 (23-581):81, 2010.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168. ACM, 2006.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

John W Coulston, Christine E Blinn, Valerie A Thomas, and Randolph H Wynne. Approximating prediction uncertainty for random forest regression models. *Photogrammetric Engineering & Remote Sensing*, 82(3):189–197, 2016.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems. *stat*, 1050:11, 2017.

Tony Duan, Anand Avati, Daisy Yi Ding, Sanjay Basu, Andrew Y Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. *Proc. 37th International Conference on Machine Learning (ICML)*, 2020.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4): 367–378, 2002.

Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. 33rd International Conference on Machine Learning (ICML-16)*, 2016.

Dan Hendrycks and Kevin Gimpel. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. `http://arxiv.org/abs/1610.02136`, 2016. arXiv:1610.02136.

Kaggle. Amazon dataset. `https://www.kaggle.com/bittlingmayer/amazonreviews`, 2017.

A. Kendall, Y. Gal, and R. Cipolla. Multi-Task Learning Using Uncertainty to Weight Losses for Scene Geometry and Semantics. In *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2017.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, 2019.

Ronny Kohavi and Barry Becker. Adult dataset. `https://archive.ics.uci.edu/ml/datasets/Adult`, 1996.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2017.

Antonio R Linero. A review of tree-based bayesian methods. *Communications for Statistical Applications and Methods*, 24(6), 2017.

Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *arXiv preprint arXiv:1902.02476*, 2019.

Andrey Malinin. *Uncertainty Estimation in Deep Learning with application to Spoken Language Assessment*. PhD thesis, University of Cambridge, 2019.

Andrey Malinin and Mark JF Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. 2019.

Andrey Malinin, Bruno Mlodozeniec, and Mark JF Gales. Ensemble distribution distillation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BygSP6Vtvr`.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pp. 6638–6648, 2018.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *CoRR*, abs/1702.03849, 2017. URL `http://arxiv.org/abs/1702.03849`.

Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pp. 521–530. ACM, 2007.

Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2):577–584, 2005.

Mohammad Hossein Shaker and Eyke Hüllermeier. Aleatoric and epistemic uncertainty with random forests. In *International Symposium on Intelligent Data Analysis*, pp. 444–456. Springer, 2020.

L. Smith and Y. Gal. Understanding Measures of Uncertainty for Adversarial Example Detection. In *UAI*, 2018.

Aleksei Ustimenko and Liudmila Prokhorenkova. SGLB: Stochastic Gradient Langevin Boosting. *arXiv e-prints*, art. arXiv:2001.07248, 2020.

Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015.

A INFERENCE IN VIRTUAL SGLB

We can compute $\Theta_{T,K}$ with the same computation time as one $\boldsymbol{\theta}^{(T)}$. According to (12), for SGLB we have $\boldsymbol{\theta}^{(T)} = \sum_{i=1}^{T} \epsilon(1-\gamma\epsilon)^{T-i}\boldsymbol{\phi}^{(i)}$, where $(1-\gamma\epsilon)^{T-i}$ appears due to shrinkage. While computing $\boldsymbol{\theta}^{(T)}$ we store the partial sums $\boldsymbol{\theta}^{(T)}_{\leq t} = \sum_{i=1}^{t} \epsilon(1-\gamma\epsilon)^{T-i}\boldsymbol{\phi}^{(i)}$. Then, any model $\boldsymbol{\theta}^{(t)}$ from $\Theta_{T,K}$ can easily be obtained from the stored values:

$$\boldsymbol{\theta}^{(t)} = \sum_{i=1}^{t} \epsilon(1-\gamma\epsilon)^{t-i}\boldsymbol{\phi}^{(i)} = (1-\gamma\epsilon)^{t-T}\boldsymbol{\theta}^{(T)}_{\leq t}\,. \tag{15}$$

B EXPERIMENTAL SETUP

B.1 OUR IMPLEMENTATION OF DATA UNCERTAINTY

As discussed in Section 2.2 of the main text, for regression we simultaneously predict the parameters μ and $\log\sigma$ of the Normal distribution. Similarly to NGBoost, we use the natural gradients. For our loss and parameterization, the natural gradient is:

$$g^{(t)}(\boldsymbol{x}, y) = \left(\mu^{t-1} - y, \frac{1}{2} - \frac{1}{2}\left(\frac{y-\mu^{(t-1)}}{\sigma^{(t-1)}}\right)^2\right). \tag{16}$$

At each step of the gradient boosting procedure, we construct one tree predicting both the components of g^t, similarly to the MultiRMSE regime of CatBoost.[6]

Recall that for classification we optimize the logistic loss.

In Table 2, we compare our implementation with NGBoots Duan et al. (2020) and Deep Ensembles Lakshminarayanan et al. (2017) on regression datasets. The best results are highlighted.

Table 2: Comparison of our implementation with existing methods

Dataset	RMSE			NLL		
	Deep. Ens.	NGBoost	CatBoost	Deep. Ens.	NGBoost	CatBoost
Boston	3.28 ± 1.00	$\mathbf{2.94 \pm 0.53}$	$3.18_{\pm 0.72}$	2.41 ± 0.25	$\mathbf{2.43 \pm 0.15}$	$2.56_{\pm 0.22}$
Concrete	6.03 ± 0.58	$\mathbf{5.06 \pm 0.61}$	$5.12_{\pm 0.54}$	3.06 ± 0.18	$\mathbf{3.04 \pm 0.17}$	$\mathbf{3.04}_{\pm 0.13}$
Energy	2.09 ± 0.29	$\mathbf{0.46 \pm 0.06}$	$0.52_{\pm 0.05}$	1.38 ± 0.22	0.60 ± 0.45	$\mathbf{0.52}_{\pm 0.13}$
Kin8nm	$\mathbf{0.09 \pm 0.00}$	0.16 ± 0.00	$0.15_{\pm 0.0}$	$\mathbf{-1.20 \pm 0.02}$	-0.49 ± 0.02	$-0.55_{\pm 0.02}$
Naval	$\mathbf{0.00 \pm 0.00}$	0.00 ± 0.00	$0.0_{\pm 0.0}$	$\mathbf{-5.63 \pm 0.05}$	-5.34 ± 0.04	$-5.46_{\pm 0.04}$
Power	4.11 ± 0.17	3.79 ± 0.18	$\mathbf{3.58}_{\pm 0.24}$	2.79 ± 0.04	2.79 ± 0.11	$\mathbf{2.61}_{\pm 0.06}$
Protein	4.71 ± 0.06	4.33 ± 0.03	$\mathbf{3.91}_{\pm 0.02}$	2.83 ± 0.02	2.81 ± 0.03	$\mathbf{2.59}_{\pm 0.02}$
Wine	0.64 ± 0.04	$\mathbf{0.63 \pm 0.04}$	$0.64_{\pm 0.06}$	0.94 ± 0.12	$\mathbf{0.91 \pm 0.06}$	$0.94_{\pm 0.11}$
Yacht	1.58 ± 0.48	$\mathbf{0.50 \pm 0.20}$	$0.63_{\pm 0.31}$	1.18 ± 0.21	0.20 ± 0.26	$\mathbf{0.13}_{\pm 0.25}$
Year MSD	$\mathbf{8.89 \pm NA}$	$8.94 \pm NA$	$8.97 \pm NA$	$\mathbf{3.35 \pm NA}$	$3.43 \pm NA$	$3.41 \pm NA$

B.2 Parameter tuning

For all approaches, we use random search to tune *learning-rate* parameter in $\{0.1, 0.01, 0.001\}$, tree *depth* in $\{3, 4, 5, 6\}$, *subsample* rate in $\{0.25, 0.5, 0.75\}$. For SGLB-based approaches, we set *diffusion-temperature* $= N$ and *model-shrink-rate* $= \frac{1}{2N}$.

B.3 Datasets

The datasets are described in Table 3.

B.4 Statistical significance

For regression, we use standard train/validation/test splits UCI and perform cross-validation to estimate statistical significance with paired t-test. In the corresponding tables, we highlight the approaches that are insignificantly different from the best one (p-value > 0.05).

For classification (and the Years regression dataset) we split the datasets in proportion 65/15/20 in train, validation, and test sets. In this case, we measure statistical significance for NLL and error/RMSE on the test test and in the corresponding tables the approaches that are insignificantly different from the best one are highlighted. For PRR and AUC-ROC results (for classification and Years) we highlight the best value.

C Additional experimental results

In Table 4, we compare ensemble approaches with single models in terms of NLL and error-rate for classification and in terms of NLL and RMSE for regression tasks. Results for NLL demonstrate an advantage of ensembling approaches compared to single models. However, in many cases the difference is not significant, which can be explained by the additive nature of boosting: averaging several tree ensembles gives another (larger) tree ensemble. Thus, improved NLL can result from the increased complexity of ensemble models. We can make similar conclusion from the results for RMSE and error rate: there is ensembling yields an advantage, but it is not so significant. Surprisingly, for NLL optimization virtual ensembling performed better than single models indicating that SGLB can benefit from using a proper re-weighting of trees which simulates an ensemble.

Tables 5 extends the corresponding table from the main text to all considered datasets.

[6] `https://catboost.ai/docs/concepts/loss-functions-multiregression.html`

Table 3: Datasets description

Dataset	# Examples	# Features
Appetency kdd (2009)	50000	231
Churn kdd (2009)	50000	231
Upselling kdd (2009)	50000	231
Adult Kohavi & Becker (1996)	48842	15
Amazon Kaggle (2017)	32769	9
Click kdd (2012)	399482	12
Internet int (1998)	10108	69
Kick kic (1998)	72983	36
Boston UCI	506	13
Concrete UCI	1030	8
Energy UCI	768	8
Kin8nm UCI	8192	8
Naval UCI	11934	16
Power UCI	9568	4
Protein UCI	45730	9
Wine UCI	1599	11
Yacht UCI	308	6
Year MSD UCI	515345	90

Table 4: NLL and RMSE/Error rate for regression and classification

Dataset	Single		Ensemble			Single		Ensemble		
	SGB	SGLB	SGB	SGLB	vSGLB	SGB	SGLB	SGB	SGLB	vSGLB
	Classification NLL (↓)					Classification % Error (↓)				
Adult	**0.275**	**0.276**	**0.275**	**0.275**	**0.276**	**12.8**	**12.6**	**12.6**	**12.7**	**12.6**
Amazon	0.145	**0.142**	0.144	0.143	**0.141**	**4.4**	**4.5**	**4.3**	**4.4**	**4.5**
Click	0.392	0.392	**0.392**	**0.392**	0.393	**15.6**	**15.6**	**15.6**	**15.6**	**15.6**
Appetency	0.074	**0.073**	**0.073**	**0.073**	**0.073**	**1.8**	**1.8**	**1.8**	**1.8**	**1.7**
Churn	0.229	0.229	**0.227**	**0.228**	0.229	**7.1**	**7.1**	**7.1**	**7.1**	**7.1**
Upselling	0.167	**0.163**	**0.163**	**0.163**	**0.163**	4.7	**4.6**	**4.6**	**4.6**	**4.6**
Internet	0.228	0.229	0.228	0.228	**0.222**	10.3	**10.0**	**10.1**	**9.9**	**10.1**
Kick	0.286	0.288	**0.285**	**0.285**	0.287	**9.5**	9.5	**9.5**	**9.5**	**9.5**
Dataset	Regression NLL (↓)					Regression RMSE (↓)				
BostonH	2.55	2.54	2.54	2.54	**2.46**	**3.18**	**3.19**	**3.17**	**3.18**	3.40
Kin8nm	-0.55	-0.55	-0.55	-0.55	**-0.59**	0.15	0.15	0.15	0.15	**0.14**
Protein	2.59	2.61	**2.54**	**2.57**	2.62	3.91	**3.97**	**3.89**	**3.95**	4.08
Concrete	**3.04**	**3.03**	**3.03**	**3.02**	**3.03**	**5.12**	**5.06**	**5.11**	**5.06**	5.63
Naval-p	-5.45	-5.46	**-5.65**	**-5.65**	-5.22	0.00	0.00	**0.00**	**0.00**	0.00
Wine-qu	0.93	0.94	0.93	0.94	**0.90**	**0.64**	**0.64**	**0.64**	**0.64**	**0.64**
Energy	**0.51**	**0.53**	**0.51**	**0.52**	**0.50**	**0.52**	**0.53**	**0.52**	**0.53**	**0.56**
Power-p	2.61	2.61	**2.59**	**2.59**	**2.58**	**3.58**	**3.58**	**3.57**	**3.57**	**3.56**
Yacht	**0.13**	**0.17**	**0.12**	**0.10**	**0.16**	**0.63**	**0.61**	**0.63**	**0.61**	0.95
Year	**3.41**	**3.41**	**3.41**	**3.41**	**3.41**	8.97	8.98	**8.96**	**8.96**	9.03

Table 5: Detection of errors and OOD examples for regression and classification tasks

Dataset		Single		Ensemble			Single		Ensemble		
		SGB	SGLB	SGB	SGLB	vSGLB	SGB	SGLB	SGB	SGLB	vSGLB
		Classification % PRR (↑)					Classification % AUC-ROC (↑)				
Adult	TU	**72**	71	71	**72**	71	19	14	19	21	21
	KU	—	—	52	52	26	—	—	66	**72**	44
Amazon	TU	65	67	65	67	**69**	82	**84**	82	**84**	**84**
	KU	—	—	53	59	51	—	—	58	65	79
Click	TU	**43**	**43**	**43**	**43**	**43**	40	39	40	40	39
	KU	—	—	25	25	10	—	—	**87**	**87**	78
Appetency	TU	70	72	71	71	71	15	19	16	19	17
	KU	—	—	81	**82**	67	—	—	53	72	**85**
Churn	TU	49	49	**50**	**50**	49	96	91	97	96	94
	KU	—	—	38	36	22	—	—	**99**	**99**	98
Upselling	TU	54	**56**	**56**	**56**	**56**	35	61	44	61	48
	KU	—	—	50	47	17	—	—	81	**92**	80
Internet	TU	77	76	76	76	**78**	77	76	76	77	71
	KU	—	—	70	70	43	—	—	**93**	**93**	84
Kick	TU	43	43	**44**	**44**	43	49	49	69	54	58
	KU	—	—	34	34	18	—	—	**99**	97	95
Dataset		Regression % PRR (↑)					Regression % AUC-ROC (↑)				
BostonH	TU	45	46	45	46	**50**	**71**	**71**	**71**	**71**	57
	KU	—	—	42	45	47	—	—	**72**	**71**	36
Kin8nm	TU	**59**	**59**	**59**	**59**	**58**	**50**	**50**	**51**	**50**	47
	KU	—	—	18	19	29	—	—	**54**	**49**	37
Protein	TU	49	47	**51**	**50**	48	83	87	**89**	**92**	**91**
	KU	—	—	30	29	2	—	—	**92**	**92**	80
Concrete	TU	**38**	**38**	**38**	**38**	30	83	84	84	84	78
	KU	—	—	29	28	25	—	—	91	**93**	65
Naval-p	TV	80	80	**85**	**84**	83	97	97	99	99	86
	KU	—	—	63	64	65	—	—	**99**	**99**	54
Wine-qu	TV	33	**33**	**33**	**33**	**34**	56	58	56	58	59
	VoE	—	—	22	17	2	—	—	**65**	**64**	32
Energy	TU	**54**	**54**	**55**	**54**	**53**	67	66	83	84	60
	KU	—	—	31	31	39	—	—	**100**	**100**	60
Power-p	TU	28	**32**	30	**32**	**32**	37	42	43	46	47
	KU	—	—	8	12	14	—	—	**76**	73	68
Yacht	TU	**89**	**89**	**89**	**89**	**87**	47	44	48	44	47
	KU	—	—	81	80	69	—	—	55	**57**	52
Year	TU	61	61	62	**62**	61	61	59	61	62	60
	KU	—	—	29	29	20	—	—	63	**66**	52

D Comparison with Random Forest

In our paper, we specifically focus on uncertainty estimation in Gradient Boosted Decision Tree (GBDT) models. However, some related work was done for quantifying uncertainty in random forests (Coulston et al., 2016; Shaker & Hüllermeier, 2020), which are also ensembles of decision trees. Thus, for completeness, we additionally compare our uncertainty estimation results on GBDTs to random forests and ensembles of random forests. In these experiments, we used the scikit-learn implementation of random forests (Pedregosa et al., 2011).

Unlike GBDT, where trees are added to correct the previous model's mistakes, random forests (RF) consist of decision trees that are *independently* trained on bootstrapped sub-samples of the dataset. Hence, in contrast to GBDT, each individual decision tree model in a random forest can be considered an element of an ensemble for the purpose of estimating *knowledge uncertainty*. Drawing a parallel to virtual SGLB, we call this approach vRF (virtual RF) since it allows estimating *knowledge uncertainty* using only one trained random forest model. Similarly, one can also construct an ensemble of several independently trained random forest models, which is expected to be a stronger baseline. However, we expect a small difference between vRF and an ensemble of random forests, as there are, a-priori, no correlations between trees both in a single model and across multiple RF models.

In Tables 6 and 7, we compare the predictive performance of random forests (both individual models and explicit ensembles of multiple models) to SGLB individual and ensemble models on classification and regression tasks. The results show that generally, though with exceptions, GBDT models outperform random forest models in terms of classification error rate and NLL. Similarly, for regressions tasks, GBDT models outperform random forests in terms of RMSE. Note that unlike GBDT models, where trees' maximum depth is limited to 6 in our experiments, the maximum depth for RF is unlimited. It is known that RFs require deeper trees to achieve performance comparable to GBDTs, but this makes these models more complex in terms of memory and inference time (for the same number of trees). Note that we cannot calculate NLL for RF regression models as they are not naturally probabilistic (do not yield a predicted variance). As a result, they are unable to estimate *data uncertainty*, and therefore we can only obtain estimates of *knowledge uncertainty*.

Table 6: Comparison with random forests: NLL and Error rate for classification

Dataset	Single		Ensemble		Single		Ensemble	
	SGLB	RF	SGLB	RF	SGLB	RF	SGLB	RF
	Classification NLL (↓)				Classification % Error (↓)			
Adult	**0.276**	0.288	**0.275**	0.287	**12.6**	13.5	**12.7**	13.4
Amazon	**0.142**	0.162	0.143	0.162	**4.5**	5.2	**4.4**	5.2
Click	0.392	0.414	**0.392**	0.412	**15.6**	15.7	**15.6**	15.7
Appetency	**0.073**	0.077	**0.073**	0.076	**1.8**	2.2	**1.8**	2.2
Churn	0.229	**0.226**	0.228	**0.226**	7.1	**6.9**	7.1	**6.9**
Upselling	**0.163**	0.171	**0.163**	0.171	**4.6**	5.2	**4.6**	5.2
Internet	0.229	0.242	0.228	0.242	**10.0**	10.6	**9.9**	10.6
Kick	0.288	0.301	**0.285**	0.301	9.5	10.3	**9.5**	10.3

Table 7: Comparison with random forests: RMSE for regression

Dataset	Single		Ensemble	
	SGLB	RF	SGLB	RF
Boston	3.19	**2.96**	3.18	**2.96**
Kin8nm	**0.15**	0.16	**0.15**	0.16
Protein	3.97	**3.5**	3.95	**3.5**
Concrete	**5.06**	5.22	**5.06**	5.21
Naval	0.00	0.01	**0.00**	0.01
Wine	**0.64**	0.76	**0.64**	0.76
Energy	**0.53**	0.72	**0.53**	0.72
Power	**3.58**	3.72	**3.57**	3.72
Yacht	**0.61**	0.82	**0.61**	0.82

Table 8 shows that RF models are generally inferior in their estimates of uncertainty relative to GBDT models both in terms of error detection (PRR) and in terms of out-of-domain input detection (ROC-AUC). One reason for this may be that bootstrap simply fails to generate sufficient diversity in fully independent models. Furthermore, error detection performance, in terms of PRR, is especially poor for RF regression models. This is because we are only able to obtain estimates of *knowledge uncertainty*, which tend to yield worse performance than estimates of *total uncertainty*.

Table 8: Detection of errors and OOD examples for regression and classification tasks

Dataset		vSGLB	vRF	SGLB	RF	vSGLB	vRF	SGLB	RF
		Classification % PRR (↑)				Classification % AUC-ROC (↑)			
Adult	TU	71	31	**72**	44	21	50	21	51
	KU	26	42	52	43	44	50	**72**	51
Amazon	TU	**69**	41	67	47	**84**	61	**84**	61
	KU	51	47	59	47	79	61	65	61
Click	TU	**43**	19	**43**	26	39	72	40	74
	KU	10	24	25	24	78	72	**87**	74
Appetency	TU	71	40	71	48	17	66	19	71
	KU	67	44	**82**	46	**85**	66	72	71
Churn	TU	49	29	**50**	31	94	91	96	93
	KU	22	30	36	30	98	91	**99**	93
Upselling	TU	**56**	20	**56**	22	48	60	61	59
	KU	17	22	47	21	80	60	**92**	59
Internet	TU	**78**	63	76	68	71	81	77	80
	KU	43	66	70	66	84	81	**93**	80
Kick	TU	43	29	**44**	37	58	92	54	93
	KU	18	32	34	33	**95**	92	97	93
Dataset		Regression % PRR (↑)				Regression % AUC-ROC (↑)			
Boston	TU	**50**	—	46	—	57	—	**71**	—
	KU	47	0	45	0	36	60	**71**	61
Kin8nm	TU	**58**	—	**59**	—	47	—	**50**	—
	KU	29	12	19	13	37	41	**49**	39
Protein	TU	48	—	**50**	—	**91**	—	**92**	—
	KU	2	5	29	5	80	50	**92**	50
Concrete	TU	30	—	**38**	—	78	—	84	—
	KU	25	-4	28	-9	65	68	**93**	69
Naval	TV	83	—	**84**	—	86	—	99	—
	KU	65	-1	64	-3	54	77	**99**	82
Wine	TV	**34**	—	**33**	—	59	—	58	**—**
	VoE	2	0	17	1	32	62	64	**67**
Energy	TU	**53**	—	**54**	—	60	—	84	**—**
	KU	39	3	31	9	60	99	**100**	**100**
Power	TU	**32**	—	**32**	—	47	—	46	—
	KU	14	0	12	0	68	42	**73**	44
Yacht	TU	**87**	—	**89**	—	47	—	44	—
	KU	69	-3	80	-10	52	50	**57**	50