XFORMER: A Multi-Stage Uncertainty-Guided Deep Learning Framework for Time Series Extreme Event Forecasting

Rupasree Dey[1], Andrei Bachinin[1], Tanjim Bin Faruk[1], Abdul Matin[1],
Mu Hong[2], Yao Zhang[2], Shrideep Pallickara[1], and Sangmi Lee Pallickara[1]
[1]Department of Computer Science, Colorado State University, USA
[2]Department of Soil and Crop Sciences, Colorado State University, USA
{rupasree.dey, andrei.bachinin, tanjim.faruk, abdul.matin,
mu.hong, yao.zhang, shrideep.pallickara, sangmi.pallickara}@colostate.edu

***Abstract*— Accurate prediction of extreme events remains difficult due to the scarcity of high-magnitude observations and the unreliability of conventional uncertainty estimates. We introduce XFORMER, a multi-stage ensemble framework that transforms uncertainty quantification from a passive diagnostic into an active driver of model improvement. Our method trains a diverse ensemble of forecasting models in two phases: first, by cultivating complementary predictors through varied initializations and learning dynamics; and second, by exploiting ensemble disagreement to locate and refine challenging regions of the data space. By coupling uncertainty-guided sampling with extreme-aware loss functions, we construct an adaptive curriculum that progressively prioritizes high-uncertainty and rare extreme events. This uncertainty-aware training paradigm shows that incorporating ensemble disagreement during training (rather than relying on it only at inference) yields more reliable and calibrated forecasts for rare but consequential events. We demonstrate XFormer's effectiveness across diverse domains—including ecosystem fluxes, air quality, and web traffic, achieving 91-98% detection rates for rare events (an 18-26 percentage point improvement over state-of-the-art models) while reducing prediction errors by 25-45%.**

deep ensembles, uncertainty quantification, time-series forecasting, calibrated uncertainty, extreme events

I. INTRODUCTION

Deep learning has been widely applied across diverse scientific domains, including chemistry, biology, forestry, and disaster monitoring. A central challenge in applying machine learning to the physical sciences lies in accurately predicting rare events within continuously evolving spatiotemporal systems [1]. What constitutes an extreme event varies across disciplines and methodologies, but such events (whether unusually high or unusually low) hold particular importance in developing predictive models. Their rarity leads to underrepresentation in datasets and results in severe class imbalance. Conventional machine learning models tend to emphasize majority classes and therefore perform poorly on minority events, which have few representative examples for training. Time series forecasting models face comparable difficulties, as rare or nonstandard temporal patterns are seldom encountered during training.

Extreme timeseries value forecasting problems have been extensively studied, particularly in domains such as weather forecasting and finance. Time series forecasting models aim to capture temporal dependencies among consecutive data samples and include architectures such as Recurrent Neural Networks (RNNs), the Long Short-Term Memory (LSTM) network, Gated Recurrent Units (GRU) [2], and Temporal Convolutional Network (TCN) [3]. Despite their success in general time series prediction tasks, these models are not inherently equipped to handle extreme value forecasting or class imbalance problems.

To alleviate imbalance within datasets, researchers have explored data augmentation techniques [4]. Common strategies such as oversampling and undersampling attempt to equalize class distributions but offer limited improvement in recognizing minority events. Synthetic oversampling techniques, including the Synthetic Minority Oversampling Technique (SMOTE) [5] and its variants, combine these strategies by generating artificial samples rather than replicating existing ones. Generative Adversarial Networks (GAN)-based augmentation [6] has also been investigated to enhance minority representation. These methods, however, depend on prior knowledge of data distributions and often fail to maintain robustness when applied to unseen or out-of-distribution conditions.

Statistical methods derived from extreme value theory provide another route for identifying extremes. Such approaches use a relevance function to assign each observation a score, with higher scores indicating greater extremity. Both the function and its associated threshold are typically specified by domain experts. Although recent work [7] seeks to generalize the relevance function for use within deep learning models, these methods still require detailed knowledge of dataset characteristics before training.

In this study, we propose a framework for refining time series forecasting models through uncertainty quantification to enhance the accuracy of extreme value predictions in multivariate, multi-task deep learning contexts. Concretely, we treat uncertainty as a training-time control signal rather than a post-hoc diagnostic: disagreement among ensemble members identifies regions where the model family has not converged on a consistent explanation of the data. This disagreement is primarily epistemic uncertainty (reducible with targeted data exposure), which is especially pronounced for rare regimes that are sparsely represented and therefore weakly constrained. We use this signal to (1) concentrate optimization on "hard" windows that induce unstable pre-

dictions, and (2) preferentially amplify gradients associated with tail dynamics. This yields an adaptive curriculum that reallocates model capacity toward high-impact events while preserving overall fidelity on typical behavior. We refer to this framework as XFORMER (for Extreme Forecasting Transformer), emphasizing its focus on uncertainty-guided learning of rare, high-impact events. Model uncertainty is estimated using an ensemble of predictors differentiated by stochastic variations in their training processes, allowing us to capture variability across the joint input–output space. These uncertainty estimates are combined with identified tail events to guide targeted model refinement. We introduce an extreme-aware dual-weighting loss function that adapts model parameters with particular sensitivity to extreme observations. To evaluate this approach, we employ an Informer-based multi-task Transformer architecture. Our specific contributions include:

- We have designed an ensemble-based uncertainty quantification methodology for time series forecasting models.
- We have designed a novel extreme-aware dual-weighing loss function that integrates uncertainty quantification into both model training and refinement, improving predictive fidelity for extreme events.
- Our framework is agnostic to the underlying model architecture. To demonstrate and evaluate its effectiveness, we develop a two-stage model refinement framework, built on top of the Informer Transformer architecture.
- We conducted extensive comparisons with both foundational and advanced frameworks, supported by ablation studies that isoloate the contribution of each component. We evaluated XFORMER across multiple real-world time series datasets to assess robustness and generalizability.

II. RELATED WORK

Incorporating Uncertainty during Model Training Traditional uncertainty quantification methods such as Deep Ensembles [8] and Monte Carlo Dropout [9] focus on post-hoc calibration with limited influence on model optimization. Recent approaches incorporate uncertainty through weighted losses that either down-weight uncertain samples for robustness [10] or up-weight them to emphasize hard examples [11]. However, these rely on static weighting schemes that do not adapt to evolving model confidence. Wen et al. [12] showed that naively combining ensembles with strong augmentations can harm calibration, underscoring the need for principled integration strategies. We address these limitations with a framework that actively leverages ensemble disagreement to regulate training. Our two-stage design (robust pre-training followed by uncertainty-guided refinement) ensures the model first learns stable representations before dynamically focusing on samples that are extreme or induce high ensemble disagreement. Unlike long-tailed learning methods [13], [14] designed for classification or quantile regression [15] requiring predefined quantiles, our approach dynamically identifies challenging regions in time-series forecasting without pre-specification, unifying adaptive focus with distributional awareness.

Knowledge Guided Machine Learning Several efforts have explored the use of knowledge guided machine learning (KGML) methods to model spatiotemporally evolving phenomena. These methods often work in tandem with mechanistic or process-based domain theoretic models, and also include custom multipart loss functions that leverage scientific knowledge. These include support for physical constraints grounded in soil hydrology, including the van Genuchten water retention equations and models (i.e., Richards' Equation) of hydraulic conductivity [16], [17]; vegetation indices [18]; evapotranspiration [19]; preserving graph properties such as betweenness centrality [20]; masking cloud occlusions in satellite imagery [21], accounting for human perceptual limits during visualization [22]; soil salinity [23], hyperspectral imagery [24], and accounting for correlations between soil spectroscopic properties [25].

III. METHODOLOGY

We propose an uncertainty-guided deep learning framework to improve time series forecasting accuracy for data with rare extreme values. As illustrated in Figure 1, our approach operates on multivariate time series datasets with aligned temporal steps and supports both single-task and multi-task learning. The framework integrates three interrelated strategies: (1) an ensemble-based uncertainty quantification methodology (§III-A); (2) an adaptive loss function that considers combined uncertainty (§III-B) and single target value distributions; and (3) a two-stage training strategy that progressively shifts the focus of model training toward extreme values (§III-C).

These strategies are general and apply to any deep learning forecasting architecture. We demonstrate their effectiveness by implementing them in XFORMER (Figure 2), a framework built on the Informer architecture [26]. Informer is well suited for this purpose because it efficiently handles long input sequences and mitigates the quadratic computational cost that typically constrains standard Transformer models.

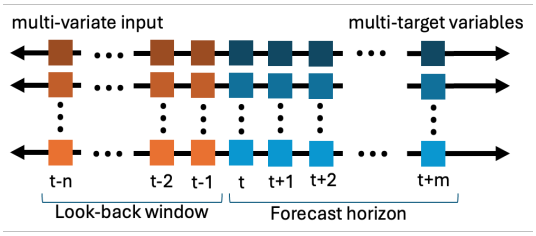

Fig. 1. Temporal aspects for multi-variate time series dataset with aligned time steps and multi target forecasting.

A. Uncertainty Quantification for Timeseries Forecasting

For deep learning–based time series forecasting models, uncertainty is often estimated by fitting a model to time-dependent observations and then simulating a large number of possible trajectories [27]. These simulations approximate the distribution of underlying measurements at each time step. In time series regression, both the look-back window and

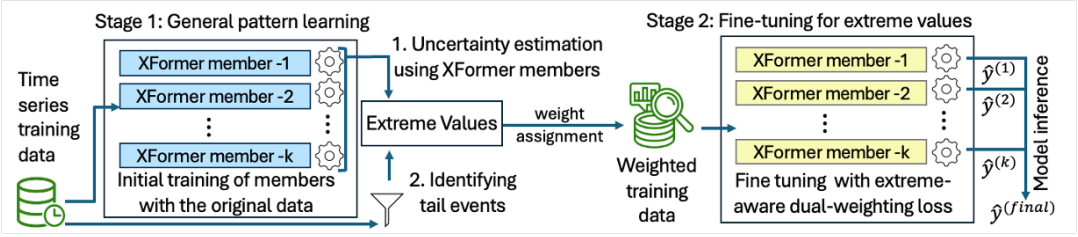

Fig. 2. Overview of the proposed framework with a two-stage training strategy. Stage 1: Train ensemble models with the Informer backbone on the original dataset, then generate weighted samples based on estimated uncertainty and extremity. Stage 2: Fine-tune the models with an extreme-aware loss using the selected samples. Final multi-target output is generated.

the forecast horizon(Figure 1), influence model performance significantly [28]. However, the relationships between these window lengths and predictive accuracy are neither linear nor independent of the characteristics of the underlying dataset.

To address this, we construct an ensemble of multiple models that share the same network architecture but differ in their training realizations. Diversity among ensemble members is driven by three primary sources of stochasticity: (1) random weight initializations, leading to distinct optimization paths; (2) stochastic mini-batch sampling in each epoch; and (3) dropout operations, which inject noise during forward passes. This process ensures architectural consistency while producing an ensemble of models with distinct predictive characteristics.

XFORMER leverages this ensemble in two ways: (1) to improve the overall forecast accuracy and (2) to quantify model uncertainty. Each member independently processes the input sequence to produce forecasts for all target variables, and the final ensemble prediction is obtained by averaging the outputs of all members.

$$\hat{y}_{\text{final}} = \frac{1}{K} \sum i = 1^{K} M_i(\mathbf{X}) \tag{1}$$

The ensemble prediction, $\hat{y}_{\text{final}}$, represents the mean output across all K member models. Each model, denoted by $M_i(\mathbf{X})$, processes the same input sequence but contributes a distinct perspective that is shaped by its training dynamics. Averaging these outputs produces a forecast that is both more stable and more representative of the model family's collective behavior.

Uncertainty is then computed for each data point, considering all input variables and the corresponding multivariate forecast results. Consequently, it captures the model's variability across the full joint distribution of inputs and outputs, rather than isolating any single target variable distribution. For each sample i, uncertainty u_i is the standard deviation across the K ensemble members:

$$u_i = \text{Std}(M_1(\mathbf{X}_i), M_2(\mathbf{X}_i), \ldots, M_K(\mathbf{X}_i)) \tag{2}$$

where $\mathbf{X}_i$ denotes the input features for sample i, $M_k(\cdot)$ represents the k-th ensemble model, and $\text{Std}(\cdot)$ computes the standard deviation across model predictions. The standard deviation is computed across model outputs and averaged over the prediction horizon and all target variables. We use standard deviation over variance for its stable threshold selection and more interpretable magnitude.

B. Extreme-aware Dual-weighting Loss Function

To adjust model training on extreme values while maintaining overall training effectiveness, we introduce a dual-weight loss function. This loss function assigns one weight to non-extreme values and a separate weight to extreme values. To apply the correct weight to each data point, the framework first identifies whether the value is extreme using both thresholds and uncertainty values.

Extreme values and model uncertainty are closely interconnected because modeling extreme events is inherently uncertain, due to limited data, inherent model limitations, and the complexity of extreme systems [29]. Our approach treats model uncertainty as an active signal for detecting extreme values, especially those from the joint distribution of multiple targets. Also, we account for tail event in the distribution of each individual target variable. A sample is labeled as a tail event if any target variable falls in the extreme range at any time step within the prediction horizon.

$$\text{TailEvent}_i = \bigcup k = 1^{n} \left(\max_{t=1}^{T} |y_i^{(k,t)}| \geq \tau_k \right) \tag{3}$$

where $y_i^{(k,t)}$ denotes the true value of target variable k for sample i at time step t, and T is the prediction horizon. The threshold τ_k defines the cutoff for identifying extreme values in variable k (e.g., its 90th percentile). The union operator $\bigcup$ indicates that a sample is labeled as a tail event if any of the n target variables exceed their respective thresholds. A sample is therefore classified as an Extreme Value when it satisfies either of two conditions: if it exhibits high model uncertainty or it contains tail event in any of its target variables:

$$\text{ExtremeValue}_i = (u_i \geq \tau_u) \cup \text{TailEvent}_i \tag{4}$$

where τ_u represents the percentile of ensemble standard deviation values to provide flexibility for various datasets. In this study we used 90% as the threshold. Once the extreme values are identified, model refinement is performed using the dual-weighting loss function that applies different weights for non-extreme values and extreme values. While standard mean squared error ($\mathcal{L}_{\text{standard}} = \frac{1}{N} \sum i = 1^{N} (\hat{y}^{(i)} - y^{(i)})^2$) treats all predictions equally, $\mathcal{L}_{\text{extreme}}$ introduces adaptive weighting:

$$\mathcal{L}_{\text{extreme}} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot (\hat{y}^{(i)} - y^{(i)})^2 \tag{5}$$

where weights are assigned based on target magnitude. Now, the dual-weighting loss is specified as:

$$w_i = \begin{cases} \lambda_{\text{extreme}}, & \text{if } y^{(i)} \geq \tau_q \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

Here τ_q represents the q-th percentile threshold (e.g., $q = 90\%$)) and applying λ_{extreme} amplifies the penalty for mispredictions of extreme values. This dual-weighting loss effectively corresponds to implicitly oversampling extreme-value samples by up-weighting their gradients. The specific values for these hyperparameters (τ_u, τ_k, τ_q, λ_{extreme}) are determined empirically through validation experiments (see Section IV).

C. Putting It All Together: Two-Stage Training Strategy

Our approach follows curriculum learning principles: Stage 1 learns stable patterns, Stage 2 focuses on hard examples via ensemble disagreement. Ensemble disagreement signals epistemic uncertainty (reducible model limitations) rather than noise, making it ideal for adaptive training [10]. Applying an extreme-focused loss from the start can destabilize training, as the model focuses on rare events before learning fundamental dynamics. We mitigate this with a two-stage approach, where general pattern learning is followed by extreme event specialization.

Stage 1: General Pattern Learning. The model is trained first to learn general temporal patterns and interdependencies among variables. During this general pattern learning phase, we train the model using a standard MSE loss, averaged across all target variables to capture dominant patterns. This conventional objective encourages the model to internalize broad behaviors (such as seasonal trends and diurnal cycles) without the destabilizing influence of heavily weighting rare events. Once the model has learned the dominant temporal dynamics, attention can shift toward the less frequent but more consequential events.

Inter-Stage: Weighted Sample Generation. At the end of Stage 1, the trained ensemble is used to identify the most challenging training samples. We perform weighted random sampling using both ensemble-based uncertainty (§III-A) and tail event identification (§III-B). This process oversamples difficult examples—cases where Stage 1 models show the greatest disagreement or where extreme values occur—increasing the model's exposure to high uncertainty and rare event behavior.

Stage 2: Extreme Event Specialization. After the general pattern learning stage and generating weighted samples, training continues for additional epochs with a reduced learning rate to fine-tune the model for predicting extreme values. We use the extremity-aware loss function (§III-B) to focus the model on predicting extreme values. For multi-task prediction with *n* number of tasks, the dual-weighting loss is averaged across all target tasks:

$$\mathcal{L}_{\text{multi-task}} = \frac{1}{n} \sum_{k=1}^{n} \mathcal{L}_{\text{extreme}}^{(k)} \tag{7}$$

This strategy, combined with the extreme-aware dual-weighting loss (governed by λ_{extreme}), allows the model to address both the rarity of extreme events in the training distribution and the inherent difficulty of predicting them accurately. The specific values for these hyperparameters are determined empirically and reported in the experimental section.

IV. Experimental Setup

Model Architecture and Implementation Details To facilitate and evaluate the effectiveness of the proposed approaches, we developed an Informer-based architecture [26] that takes input sequences with length of $T = 60$ months and prediction horizon $H = 60$ months has been used for training. The model uses dimensions $d_{\text{model}} = 128$, $n_{\text{heads}} = 8$ attention heads, $n_{\text{layers}} = 3$ transformer layers, feedforward dimension $d_{\text{ff}} = 512$, and dropout rate $p_{\text{dropout}} = 0.2$. To quantify model uncertainty, we construct an ensemble of $K = 7$ independently initialized models to provide robust uncertainty quantification. Each ensemble member consists of an input embedding layer with learned positional encodings, a stack of Transformer encoder layers, and separate output projection heads for each target flux variable. The use of separate output heads allows multi-task model while sharing the common temporal encoder.

Hyperparameter Selection The percentile thresholds (τ_u, τ_k, τ_q) and loss weight (λ_{extreme}) were determined through ablation studies on the validation set. We evaluated combinations of $\{70, 80, 90\}$th percentiles and extreme loss weights $\{5, 10, 15\}$, finding that P_{90} (90th percentile for all thresholds) with $\lambda_{\text{extreme}} = 5.0$ provided the best trade-off between peak event performance and overall accuracy across datasets.

Benchmark Datasets and Data Preprocessing We evaluate our framework on benchmark datasets spanning multiple domains: Wikipedia Web Traffic Dataset [30] (daily pageview statistics), PM2.5 Urban Air Quality Dataset [31] (fine particulate matter concentrations with health and environmental significance), ETTh1 dataset [26], [32] (hourly Electricity Transformer Temperature measurements with complex temporal patterns), and DayCent® (version 279) ecosystem model outputs [33]. DayCent simulates cropland and grassland systems across multiple spatial scales and serves as the U.S. EPA's Tier-3 process-based model for national greenhouse-gas inventory. We use DayCent primarily for detailed analyses due to convenient data generation. For ecosystem predictions, we target four key variables: Gross Primary Productivity (GPP), Ecosystem Respiration (RECO), Net Ecosystem Exchange (NEE), and nitrous oxide (N_2O). To stabilize the variance of N_2O's highly skewed distribution (near-zero values with extreme spikes), we applied a 'log1p' transformation ($N_2O_{\text{log}} = \log(1 + N_2O_{\text{raw}})$). For carbon fluxes, we enforce ecological consistency (NEE = GPP − RECO) to impute missing values. Inputs are scaled using RobustScaler for robustness to outliers. The target variables were transformed with 1000 quantiles and Gaussian output distribution to map the original distribution to standard normal while preserving rank information for both typical values and extreme outliers effectively. We create

input-target pairs using a sliding window with input length (T)=60, prediction horizon (H)=60, and with stride of 1 month. Our data splitting ensures strict temporal integrity with chronological partitioning: training uses the first 80% of observations, validation the final 20%. Sliding windows (stride 1 month) are applied independently within each partition, ensuring no overlap between training and validation samples. A held-out test set from the most recent temporal segment provides final evaluation.

Baseline Models All baseline models are configured to ensure fair comparison. iTransformer [34] adopts an inverted architecture with dropout 0.1; Informer [26] uses the original encoder design with dropout 0.2; Transformer [35] employs standard encoder with global average pooling and dropout 0.1; bidirectional LSTM [36] has hidden size 256, 2 layers, and dropout 0.2; TimesNet [37] applies multi-scale temporal convolutions (kernel sizes [3, 5, 7]) with dropout 0.1. All transformer-based models share $d_{model} = 128$, 8 attention heads, 3 layers, and $d_{ff} = 512$. All baselines are trained for 50 epochs with learning rate 0.001, MSE loss, Adam optimizer, batch size 32, and early stopping (patience 10).

Evaluation Metrics Peak performance metrics focus on extreme events (values exceeding the 95th percentile): Peak NRMSE (Normalized Root Mean Squared Error), Peak NMAE (Normalized Mean Absolute Error), and Peak Detection Rate (percentage of true extremes correctly identified when predictions exceed the 95th percentile). These metrics directly measure the model's ability to capture rare, high-magnitude events critical for agricultural flux predictions. Overall performance is evaluated using R^2 Score. Because uncertainty is central to our training loop, we also assess whether uncertainty estimates are well calibrated (i.e., whether higher uncertainty corresponds to higher expected error). In addition to error–uncertainty correlation, we report reliability via coverage of prediction intervals derived from the ensemble and summarize miscalibration with a binned calibration error computed over forecast windows. For extreme-event detection, we explicitly guard against trivial "predict-everything-as-peak" behavior by pairing detection rate with precision (fraction of predicted peaks that are true peaks), ensuring improvements reflect sharper discrimination rather than systematic over-prediction.

All experiments are implemented in PyTorch 2.0 with CUDA 11.8 on NVIDIA A100 GPUs (40GB). Random seeds are fixed at 42 for reproducibility. Code and trained models will be released upon publication.

V. Results & Discussion

We evaluate our approach through a series of experiments examining the effects of ensemble size, ablation configurations, uncertainty estimation strategies, and alternative architectures using the DayCent ecosystem model outputs. Also, we conduct dataset-level comparisons across all benchmark datasets listed in Section IV to assess the cross-domain robustness and generalizability of the proposed approach.

A. Comparison among Uncertainty Estimation Methods

To evaluate the effectiveness of different uncertainty estimation strategies, we conducted experiments using the *DayCent* ecosystem model outputs. The results reveal that our ensemble-based approach consistently outperforms Monte Carlo (MC) Dropout across all major performance metrics. Specifically, the ensemble method achieved higher R^2 scores (0.90–0.98 vs 0.76–95 for MC Dropout) and reduced both NRMSE and NMAE by 30–51%. Beyond predictive accuracy, the ensemble approach produced more reliable uncertainty estimates. It had a substantially higher error–uncertainty correlation (∼0.5, compared to near zero for MC Dropout), indicating that its uncertainty values more faithfully reflected model confidence. MC Dropout tended to exhibit overconfidence, and inflated uncertainty despite slightly better nominal coverage. Given the ensemble method's superior predictive accuracy and uncertainty reliability, we adopt it as the core uncertainty estimation technique for all subsequent experiments.

B. Ablation Studies

Table I reports performance improvements over the simple ensemble baseline with the DayCent ecosystem model outputs. Extreme-aware training (E+ET) improved peak detection rates by 15–21% across variables, demonstrating enhanced capture of rare extreme events. Adding uncertainty-guided sampling (SE+ET+US) further increased detection to 91–98%, with strong improvements for NEE (93.2%) and RECO (91.9%). The uncertainty-focused variant achieved NMAE reductions of 25–45%, most notably for N_2O (45.2%) and RECO (34.5%). This demonstrates the benefit of targeting difficult-to-predict regions using ensemble uncertainty during training. GPP showed modest NMAE gains due to its smoother dynamics and dependence on well-constrained meteorological drivers [38], exhibiting higher stability than other fluxes dominated by stochastic soil–atmosphere interactions. Nevertheless, our method achieved near-perfect peak detection (97.7%) for GPP, indicating robust performance across all flux types.

TABLE I

Ablation Experiments based on uncertainty sampling and extreme-aware training, using Agricultural flux dataset. SE: Simple Ensemble, E+ET: Ensemble + Extreme Training, SE+ET+US: Full proposed method (Ensemble + Extreme Training + Uncertainty Sampling). Detect: Peak Detection %. (Blue: Best, Green: Second Best)

Variable	SE	E+ET	SE+ET+US
	NMAE / Detect	NMAE / Detect	NMAE / Detect
NEE	0.0269 / 58.48	0.0184 / 79.83	0.0200 / 93.20
GPP	0.0517 / 84.25	0.0476 / 86.41	0.0620 / 97.70
RECO	0.0595 / 59.72	0.0450 / 74.81	0.0390 / 91.90
N_2O	0.1076 / 66.30	0.0705 / 81.37	0.0590 / 91.70

C. Comparison with State-of-the-Art Models

Table II compares XFormer against several state-of-the-art architectures. Our model achieves the highest peak detection rates across all flux variables, significantly outperforming

TABLE II

COMPARISON WITH SOTA MODELS ON AGRICULTURAL FLUX DATASET. D%: PEAK DETECTION RATE (↑), NR: NRMSE (↓), NM: NMAE (↓). (BLUE: BEST, GREEN: SECOND BEST)

	NEE			GPP			RECO			N2O		
Model	**D%↑**	**NR↓**	**NM↓**	**D%↑**	**NR↓**	**NM↓**	**D%↑**	**NR↓**	**NM↓**	**D%↑**	**NR↓**	**NM↓**
Informer	88.5	0.058	0.022	97.2	0.147	0.062	80.7	0.109	0.039	80.9	0.128	0.080
Transformer	79.6	0.065	0.022	91.6	0.133	0.046	77.2	0.126	0.045	75.2	0.134	0.082
LSTM	53.1	0.077	0.036	76.3	0.149	0.068	49.6	0.151	0.065	37.8	0.178	0.131
TimesNet	76.7	0.055	0.022	88.0	0.142	0.050	70.7	0.106	0.043	65.8	0.146	0.095
iTransformer	75.1	0.036	0.019	89.5	0.092	0.039	68.0	0.108	0.043	66.1	0.134	0.087
XFORMER	93.2	0.047	0.020	97.7	0.154	0.062	91.9	0.120	0.039	91.7	0.093	0.059

all baselines. For NEE, we achieve 93.2% detection rate versus iTransformer's 75.1% and Informer's 88.5%. For N_2O, we reach 91.7% compared to iTransformer's 66.1%, while reducing NMAE from 0.087 to 0.059. To ensure detection gains are not from over-prediction, we report precision: 0.87 (NEE), 0.92 (GPP), 0.84 (RECO), 0.79 (N_2O) for extreme events, confirming most predicted peaks are true extremes.

This highlights a fundamental trade-off. Although iTransformer achieves competitive error metrics (best NRMSE of 0.0358 for NEE) and the highest overall average R^2 (0.949), it systematically underperforms in detecting extreme events. Our XFORMER achieves a nearly identical average R^2 (0.941) but dramatically outperforms iTransformer on the critical task of peak detection. The results reveal that advanced single-model architectures optimize for overall accuracy at the cost of rare events, while our ensemble approach maintains strong overall performance while excelling at extreme detection, critical for environmental impact and decision-making.

D. Comparison of Ensemble Sizes

We evaluated ensemble sizes $K \in \{3, 5, 7, 10\}$ on a validation set. Performance improved substantially from K=3 to K=7 (peak detection: 85.2% → 91.4%), with diminishing returns beyond K=7 (91.4% → 91.8% at K=10). We selected K=7 as the optimal balance between accuracy and computational cost.

E. Cross-Dataset Robustness Evaluation

To assess generalization across domains, XFORMER was evaluated on multiple benchmark datasets (Table III). The results show that the uncertainty-guided ensemble training approach delivers consistent and substantial improvements in extreme value detection. On the Wikipedia web traffic dataset, detection rates improved by 24.77%, while the Walmart retail dataset showed a 27.72% increase in detection capability alongside substantial error reduction (NMAE: -0.0443).

The PM2.5 urban air quality dataset clearly highlights the trade-off central to our approach: XFORMER's peak detection rate jumped from 50.4% to 83.4% (a +32.98 point gain), while slashing Peak NMAE by 36% (from 0.1753 to 0.1116). This specialization in high-impact events resulted in a minor, calculated drop in overall R^2 (from 0.712 to 0.668), demonstrating a successful reallocation of model capacity from general performance to critical rare event forecasting. Even on the challenging ETTh1 electricity transformer temperature dataset, consistent improvements were observed. The consistent gains in peak detection across diverse datasets confirm the broad generalizability of the uncertainty-guided ensemble approach.

TABLE III

EXTREME VALUE PREDICTION PERFORMANCE ACROSS SELECTED DATASETS. DETECT: PEAK DETECTION %

	Baseline (Informer)		Ours (XFORMER)	
Dataset	Detect (%)	NMAE	Detect (%)	NMAE
Wikipedia	30.20	0.0082	**54.97**	**0.0081**
Walmart	13.76	0.1354	**41.48**	**0.0911**
PM2.5	50.40	0.1753	**83.38**	**0.1116**
ETTh1	46.43	0.1168	**51.70**	**0.0971**

VI. CONCLUSION AND FUTURE WORK

We present an uncertainty-guided ensemble framework that transforms uncertainty quantification from passive inference into an active driver of model learning. By decoupling foundational pattern learning from extreme event specialization through a two-stage training, our approach achieves 90%+ peak detection rates and 14-36% error reduction compared to single-stage baselines while maintaining overall accuracy. The key innovation is in leveraging ensemble disagreement not merely as a confidence measure but as a signal for dynamic capacity allocation during training. Comprehensive evaluations show consistent advantages over established architectures, with cross-dataset validation confirming transferability beyond environmental systems. Ablation studies show uncertainty-guided sampling provides additional gains over progressive refinement alone. One promising direction is to replace fixed hyperparameters (e.g., the uncertainty threshold and the relative emphasis on tail events) with closed-loop schedules driven by training diagnostics. For example, the sampling temperature or loss weights can be adapted based on the stability of ensemble disagreement over time, the marginal gain in peak detection, and calibration statistics (e.g., coverage drift). This would yield a self-tuning curriculum that increases focus on extreme/uncertain regions only when the model has sufficiently learned the background dynamics, reducing the risk of over-specializing early or allocating capacity to noise. Future work will explore dynamic weight adjustment, out-of-distribution generalization, and heterogeneous ensemble

architectures for broader applicability across domains where rare events dominate practical importance.

VII. Acknowledgment

This research was supported by the National Science Foundation (1931363, 2312319), the National Institute of Food Agriculture (COL014021223, 2025-77039-45531), and an NSF/NIFA Artificial Intelligence Institutes AI-LEAF Award [2023-03616].

References

[1] E. Pickering, S. Guth, G. E. Karniadakis, and T. P. Sapsis, "Discovering and forecasting extreme events via active learning in neural operators," *Nature Computational Science*, vol. 2, no. 12, pp. 823–833, 2022.

[2] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[3] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[4] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[6] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in neural information processing systems*, vol. 32, 2019.

[7] R. P. Ribeiro and N. Moniz, "Imbalanced regression and extreme value prediction," *Machine Learning*, vol. 109, no. 9, pp. 1803–1835, 2020.

[8] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[9] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[10] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[11] S. Mindermann, J. M. Brauner, M. T. Razzak, M. Sharma, A. Kirsch, W. Xu, B. Höltgen, A. N. Gomez, A. Morisot, S. Farquhar *et al.*, "Prioritized training on points that are learnable, worth learning, and not yet learnt," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 630–15 649.

[12] Y. Wen, G. Jerfel, R. Muller, M. W. Dusenberry, J. Snoek, B. Lakshminarayanan, and D. Tran, "Combining ensembles and data augmentation can harm your calibration," *arXiv preprint arXiv:2010.09875*, 2020.

[13] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 9, pp. 10 795–10 816, 2023.

[14] X. Xiang, Z. Zhang, and X. Chen, "Curricular-balanced long-tailed learning," *Neurocomputing*, vol. 571, p. 127121, 2024.

[15] J. W. Taylor, "Forecasting value at risk and expected shortfall using a semiparametric approach based on the asymmetric laplace distribution," *Journal of Business & Economic Statistics*, vol. 37, no. 1, pp. 121–133, 2019.

[16] P. Khandelwal, S. L. Pallickara, and S. Pallickara, "Deepsoil: A science-guided framework for generating high precision soil moisture maps by reconciling measurement profiles across in-situ and remote sensing data," in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 2024, pp. 233–246.

[17] P. Khandelwal, J. D. Niemann, D. J. Mulla, S. Pallickara, and S. L. Pallickara, "Subterra: Estimating soil moisture at root zone depths using science-guided learning," in *2025 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2025, pp. 328–335.

[18] K. Bruhwiler, P. Khandelwal, D. Rammer, S. Armstrong, S. L. Pallickara, and S. Pallickara, "Lightweight, embeddings based storage and model construction over satellite data collections," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 246–255.

[19] S. Armstrong, P. Khandelwal, D. Padalia, G. Senay, D. Schulte, A. Andales, F. J. Breidt, S. Pallickara, and S. L. Pallickara, "Attention-based convolutional capsules for evapotranspiration estimation at scale," *Environmental Modelling & Software*, vol. 152, p. 105366, 2022.

[20] A. Matin, S. Armstrong, S. Mitra, S. Pallickara, and S. L. Pallickara, "Rapid betweenness centrality estimates for transportation networks using capsule networks," in *2022 Fourth International Conference on Transdisciplinary AI (TransAI)*. IEEE, 2022, pp. 89–96.

[21] P. Khandelwal, S. Armstrong, A. Matin, S. Pallickara, and S. L. Pallickara, "Cloudnet: A deep learning approach for mitigating occlusions in landsat-8 imagery using data coalescence," in *2022 IEEE 18th International Conference on e-Science (e-Science)*. IEEE, 2022, pp. 117–127.

[22] S. Mitra, D. Rammer, S. Pallickara, and S. L. Pallickara, "Glance: A generative approach to interactive visualization of voluminous satellite imagery," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 359–367.

[23] R. Dey, A. Matin, E. Lewark, T. B. Faruk, A. Bachinin, S. Leuthold, M. F. Cotrufo, S. Pallickara, and S. L. Pallickara, "Deepsalt: Bridging laboratory and satellite spectra through domain adaptation and knowledge distillation for large-scale soil salinity estimation," 2025.

[24] A. Matin, R. Dey, T. B. Faruk, S. Pallickara, and S. L. Pallickara, "Knowledge-guided masked autoencoder with linear spectral mixing and spectral-angle–aware reconstruction," 2026.

[25] A. Bachinin, R. Dey, P. Khandelwal, S. Leuthold, M. F. Cotrufo, S. Pallickara, and S. L. Pallickara, "Science-informed multitask transformer for soil property prediction from ftir spectroscopy," in *2025 IEEE International Conference on eScience (eScience)*. IEEE, 2025, pp. 48–57.

[26] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[27] M. A. Silva, C. Amado, Á. Ribeiro, and D. Loureiro, "Uncertainty evaluation in time-dependent measurements," *Measurement*, vol. 196, p. 111196, 2022.

[28] D. Effrosynidis, E. Spiliotis, G. Sylaios, and A. Arampatzis, "Time series and regression methods for univariate environmental forecasting: An empirical evaluation," *Science of The Total Environment*, vol. 875, p. 162580, 2023.

[29] M. D. Risser and C. Tebaldi, "Uncertainty and extremes," in *Uncertainty in climate change research: An integrated approach*. Springer, 2025, pp. 217–228.

[30] S. Bhat, "Wikipedia Web Traffic 2018–19," https://www.kaggle.com/datasets/sandeshbhat/wikipedia-web-traffic-201819, 2018, kaggle Dataset.

[31] S. M. T. Hasan, "Global Urban Air Quality Index Dataset (2015–2025)," https://www.kaggle.com/datasets/syedmtalhahasan/global-urban-air-quality-index-dataset-2015-2025, 2025, kaggle Dataset.

[32] H. Zhou, "Electricity Transformer Dataset (ET-Dataset) — ETT-small (ETTh1/ETTh2, ETTm1/ETTm2)," https://github.com/zhouhaoyi/ETDataset, 2021, gitHub Dataset.

[33] W. J. Parton, M. Hartman, D. Ojima, and D. Schimel, "DAYCENT and its land surface submodel: description and testing," *Global and planetary Change*, vol. 19, no. 1-4, pp. 35–48, 1998.

[34] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[35] V. Ashish, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, p. I, 2017.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[38] X. Xu, F. Jiao, J. Liu, J. Ma, D. Lin, H. Gong, Y. Yang, N. Lin, Q. Wu, Y. Zhu *et al.*, "Stability of gross primary productivity and its sensitivity to climate variability in China," *Frontiers in Plant Science*, vol. 15, p. 1440993, 2024.