Science-Guided Multi-Task Deep Learning for Emulating APSIM Simulations for Root-zone Soil Moisture Forecasting

Tanjim Bin Faruk, Abdul Matin, Rupasree Dey, Andrei Bachinin, Shrideep Pallickara, and Sangmi Lee Pallickara
Colorado State University, Fort Collins, CO, USA
{Tanjim.Faruk, Abdul.Matin, Rupasree.Dey, Andrei.Bachinin, Shrideep.Pallickara, Sangmi.Pallickara}@colostate.edu

Abstract

Soil moisture is a critical variable for water management, drought monitoring, and crop risk management, yet direct measurements below the surface are sparse and difficult to scale. We present a deep learning framework that forecasts root-zone soil moisture generated with a process-based agroecosystem simulation model, APSIM. Our approach forecasts multi-layer soil moisture profiles down to 1 m using static site descriptors and daily meteorological forcings. Our approach unifies (i) large-scale physics-based simulation data generation over diverse counterfactual soil properties, irrigation strategies, and weather information, (ii) a controlled benchmark spanning Temporal Convolutional Networks, and Mamba-style state space models, and (iii) a multi-task training objective that predicts both absolute moisture levels and step-wise changes (deltas). The deltas formulation anchors forecasts to the last observed state and focuses learning on day-to-day process rates, improving stability across depths and forecast steps. Experiments on a large APSIM-derived dataset with 11 depth layers evaluate accuracy under a standard held-out test split, spatial generalization to unseen stations, and temporal generalization to a future year. Across architectures, delta-aware training consistently improves forecasting performance relative to direct level prediction and simple baselines, with the strongest gains appearing under distribution shift.

Introduction

Soil moisture at the root zone governs plant-available water, influences evapotranspiration and crop stress, and directly affects irrigation demand and drought impacts. Despite its importance, reliable measurements of subsurface moisture remain geographically sparse: in-situ sensors are costly, and deeper-layer observations (tens of centimeters to meters) are particularly difficult to collect at scale.

In this study, we generate scientifically realistic datasets using process-based simulators with physically consistent soil–water scenarios across diverse conditions, and train machine learning (ML) models that can reliably reproduce these dynamics. The *Agricultural Production Systems Simulator (APSIM)* is a widely used agro-ecosystem model providing depth-resolved soil moisture estimation implementing Richards' equation. However, running APSIM exhaustively across large combinations of soils, locations, and irrigation strategies is computationally intensive and requires model calibrations.

Our deep learning framework, trained with APSIM output, forecasts root-zone moisture profiles up to 1 m depth. We construct a large training corpus via perturbing the soil texture variables (e.g., fraction of sand, silt, and clay) and sampled the training data points using stratified sampling over a multi-dimensional parameter space spanning soil texture, irrigation strategy, and station-level meteorological forcing. We then benchmark representative time-series modeling families—Temporal Convolutional Networks (Bai, Kolter, and Koltun 2018), and Mamba-style state space models (Gu and Dao 2023).

The key challenge is that soil moisture levels exhibit strong site- and soil-dependent offsets, while errors can accumulate significantly across forecast steps. To address this, we introduce a *delta-aware training mode* that incorporates prediction of the day-to-day changes and reconstructs the soil moisture level using both the amount of change and the soil moisture from the immediately previous time step. This reparameterization effectively captures the strong interdependencies of soil moisture across time steps compared to applying models directly to the measurements. This approach is agnostic to the underlying model architecture. We evaluate our approach with 11 depth layers (0–1000 mm) and test the robustness of our framework in capturing stable soil–water dynamics rather than memorizing scenario-specific solutions.

Our primary contributions include:

- An APSIM-supervised simulation-to-ML pipeline forecasting root-zone soil moisture profiles with strong generalizability across diverse soils, weather forcings, and irrigation strategies.
- *deltas* based training objectives with high fidelity and robustness under spatial and temporal distribution shift.
- A controlled benchmark across TCN, and Mamba-style state space models.

Related Work

Emulating Process-Based Systems Process-based models like APSIM capture soil–crop dynamics but are computationally expensive at scale (Kheir et al. 2023). Hybrid simulation–ML frameworks have been proposed to accelerate them (Shahhosseini et al. 2021), yet purely data-driven emulators can drift under out-of-distribution conditions such as unseen weather regimes (Kashinath et al. 2021). DeepSoil (Khandelwal, Pallickara, and Pallickara 2024) combines deep learning with HydroBlocks to generate soil moisture maps. Building emulators that are both fast and physically reliable remains an open challenge.

Multi-Task Learning Joint prediction of auxiliary targets alongside a primary variable strengthens the learned representations of dynamic systems and improves long-term stability (Yuan et al. 2022). Our approach follows this intuition: co-predicting step-wise deltas and absolute moisture stabilizes forecasts across depths and horizons.

Efficient Sequence Modeling Temporal Convolutional Networks (TCNs) stack dilated causal convolutions whose receptive field grows exponentially with network depth while keeping parameter count fixed (Bai, Kolter, and Koltun 2018). Patch-based designs further reduce cost by aggregating the local temporal context (Nie et al. 2023). State-space models such as Mamba use input-dependent selective gating to process sequences in linear time, capturing long-range dependencies without the quadratic overhead of standard attention (Gu and Dao 2023). These efficient architectures are well-suited to the large-scale emulation setting targeted in this work.

Dataset and Problem Setup

We use APSIM output as our data source due to its physically consistent soil moisture dynamics. APSIM solves Richards' equation for water movement through the soil and accounts for crop uptake, evapotranspiration, and irrigation, ensuring the training data respects the soil physics.

Parameter Space Design We design simulation runs to span a multi-dimensional parameter space $\Theta = \Theta_{\text{soil}} \times \Theta_{\text{irrig}} \times \Theta_{\text{weather}}$ representing the realistic range of conditions a deployed model might encounter:

- **Soil texture space** Θ_{soil}: discretized combinations of percentages of sand, silt, and clay that satisfy sand + silt + clay $= 100\%$. Each texture determines water retention characteristics via the Rosetta pedotransfer functions.
- **Irrigation management space** Θ_{irrig}: Parameterized by seasonal water allocation a, soil moisture deficit trigger d, and minimum inter-irrigation interval m.
- **Weather forcing space** Θ_{weather}: Observed meteorological data from multiple stations across diverse geographic regions and seasons.

Stratified Sampling Strategy Exhaustive simulation across all $|\Theta|$ combinations is computationally prohibitive. We apply multi-dimensional stratified sampling to select a representative subset $\mathcal{S} \subset \Theta$ while preserving diversity across critical dimensions that affect soil moisture behavior. We define strata based on geographic region, soil drainage class, irrigation intensity, and deficit sensitivity, sample proportionally, and enforce at least one sample per stratum to preserve coverage.

Temporal Windowing and Leakage Prevention Each simulation produces a daily time series of length D days. We extract training samples using a sliding window: H days of history predict a forecast horizon of T days. Sliding windows create temporal dependencies within each simulation, so we enforce strict separation by partitioning splits at the simulation level, never at the individual sample level, preventing temporal leakage.

Forecasting Task, Features, and Evaluation

We forecast a fixed multi-day horizon of soil moisture across $L = 11$ depth layers spanning 0–1000 mm, using a fixed-length historical window of meteorological forcings and static site descriptors. (Specific history length and forecast horizon are provided in the experimental setup.)

Features Our input representation combines (i) static descriptors that encode location, terrain, soil texture, climate regime, and irrigation strategy, and (ii) daily hydroclimate forcings and management signals over a historical window.

- **Static features (12 dimensions)**: unit-sphere coordinates derived from latitude/longitude (3), elevation (1), soil texture fractions (sand and silt; 2), irrigation strategy parameters (seasonal allocation, deficit trigger threshold, minimum inter-irrigation interval; 3), and one-hot Köppen-style climate regime indicators (3).
- **Dynamic features (13 per day)**: daily rainfall, applied irrigation amount, solar radiation, maximum and minimum temperature, vapor pressure, wind speed, cyclical day-of-year encoding (sine/cosine; 2), days since planting, days since last irrigation, an extreme rainfall indicator, and crop phenology stage.
- **Targets**: soil moisture levels $y_{t,\ell}$ at $L = 11$ layers and auxiliary step changes $\Delta y_{t,\ell}$ (used in deltas mode).

TCN and Mamba are trained and evaluated under a shared protocol: the same feature construction, normalization (train-only statistics), data splits, training objective semantics (levels vs. deltas), and metric computation are applied across all models.

Methods

We compare two representative time-series models, Temporal Convolusional Network (TCN) and Mamba-style State Space Model (SSM) as shown in Figure 1, under an identical forecasting interface: static descriptors and H days of dynamic forcings are mapped to T-day forecasts for L soil layers (shape $T \times L$).

TCNs use 1D temporal convolutions with dilation to capture multi-scale dependencies efficiently. Their inductive bias favors local-to-medium range temporal patterns and stable training in long sequence settings.

Selective state space models (SSMs) provide linear-time sequence processing while retaining the ability to model long-range dependencies. We use a Mamba-style SSM backbone with the same input/output heads as other models to isolate architectural effects.

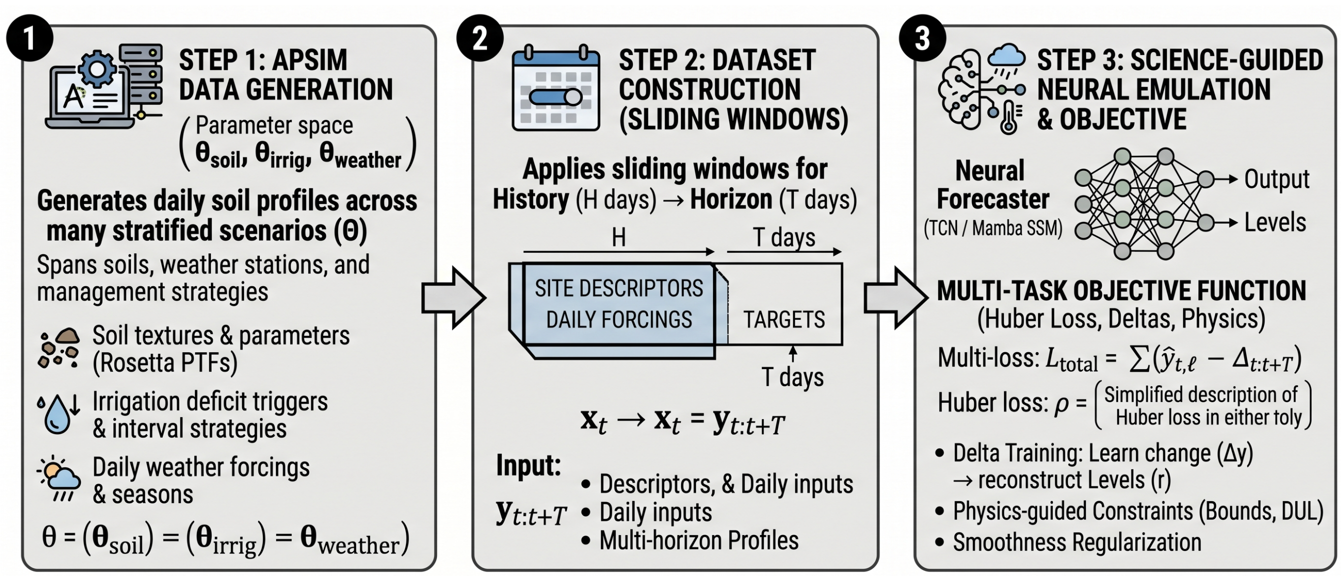

Figure 1: Overview of the APSIM-driven soil moisture emulation framework: A unified simulation-driven data generation, architecture-agnostic levels+deltas multitask neural emulator with physics constraints.

Category	Model / Objective	Standard Test Split		Held-out Station Split	
		MAE	R^2	MAE	R^2
Deterministic	Persistence	0.0107	0.8527	0.0093	0.8693
Learned	LightGBM (Deltas)	0.0094±0.000013	0.9031±0.00055	0.0085±0.000003	0.9205±0.00091
Neural Emulators	TCN (Levels)	0.0187±0.00022	0.8135±0.0021	0.0184±0.00019	0.8151±0.0018
	TCN (Deltas)	0.0086±0.000095	**0.9066±0.00078**	0.0071±0.00014	**0.9231±0.0011**
	Mamba (Levels)	0.0106±0.00018	0.8830±0.0040	0.0093±0.00022	0.9059±0.0058
	Mamba (Deltas)	**0.0083±0.000065**	0.8981±0.0010	**0.0070±0.00026**	0.9192±0.0037

Table 1: Main APSIM emulation results on the 2022→2023 benchmark (averaged over 7 days and 11 depths). Neural models report mean±std over 3 seeds; deterministic baselines have no std. LightGBM is a learned tabular baseline (mean±std). Bold marks the best deltas-mode neural emulator for MAE and R^2.

Training Objectives: Levels vs. Deltas

Let $y_{t,\ell}$ denote soil moisture at forecast step $t \in \{1, \dots, T\}$ and layer $\ell \in \{1, \dots, L\}$, and let $y_{0,\ell}$ denote the anchor level (last observed day in the history window). We define step changes (deltas) as $\Delta y_{t,\ell} = y_{t,\ell} - y_{t-1,\ell}$ with $y_{0,\ell} = y_{0,\ell}$.

Levels mode. In levels mode, the model predicts absolute soil moisture levels $\hat{y}_{t,\ell} = f_\theta(x)_{t,\ell}$ and is trained by minimizing a robust regression loss (ρ) over all steps and layers:

$$\mathcal{L}_{\text{levels}} = \frac{1}{TL} \sum_{t=1}^{T} \sum_{\ell=1}^{L} \rho(\hat{y}_{t,\ell} - y_{t,\ell}) \,. \tag{1}$$

Deltas mode. In deltas mode, the model predicts step changes $\widehat{\Delta y}_{t,\ell} = f_\theta(x)_{t,\ell}$ and soil moisture levels are reconstructed by accumulation:

$$r_{t,\ell} = y_{0,\ell} + \sum_{i=1}^{t} \widehat{\Delta y}_{i,\ell}. \tag{2}$$

We train with a delta-dominant objective that also includes a reconstruction-consistency term to mitigate horizon drift from accumulated delta errors:

$$\mathcal{L}_{\text{deltas}} = (1-\alpha)\, \mathcal{L}_{\text{recon}} + \alpha\, \mathcal{L}_{\Delta}, \tag{3}$$

where

$$\mathcal{L}_{\Delta} = \frac{1}{TL} \sum_{t,\ell} \rho\Big(\widehat{\Delta y}_{t,\ell} - \Delta y_{t,\ell}\Big)\,, \tag{4}$$

and

$$\mathcal{L}_{\text{recon}} = \frac{1}{TL} \sum_{t,\ell} \rho(r_{t,\ell} - y_{t,\ell})\,. \tag{5}$$

We fix α across all architectures to avoid architecture-specific tuning confounds and report a sensitivity analysis for α in the ablations.

Robust Regression Loss

We instantiate $\rho(\cdot)$ as the Huber loss, which behaves quadratically near zero and linearly for large residuals:

$$\rho(e) = \begin{cases} \frac{1}{2}e^2, & |e| \leq \delta_h \\ \delta_h \left(|e| - \frac{1}{2}\delta_h\right), & |e| > \delta_h \end{cases} \tag{6}$$

where δ_h is the Huber transition parameter. We use the same loss and δ_h for all models for a fair comparison.

Experiments

Dataset and Evaluation Regimes

We evaluate our approach over the APSIM benchmark dataset that we constructed from stratified simulation sampling and sliding temporal windows. We report performance under two regimes designed to assess both in-distribution accuracy and generalization:

1. **Standard test split:** a held-out test set where train/validation/test are partitioned at the *simulation* level (not at the sliding-window sample level), preventing leakage from overlapping windows of the same underlying scenario.
2. **Held-out station split:** a spatial generalization setting where entire weather stations are withheld from training and used only for evaluation, testing whether the emulator transfers to unseen forcing distributions and locations.

All model evaluations use the above splits, normalization statistics computed from training data only, and the same training/evaluation protocol to isolate architectural and objective differences.

Forecast Setup and Training Protocol

Unless otherwise noted, models use a history window of $H = 30$ days to predict a forecast horizon of $T = 7$ days for all $L = 11$ depth layers (0, 50, 100, 200, ... , 1000 mm). We train all neural models with identical splits, train-only normalization, and a shared optimization protocol, and we report mean$\pm$std over three random seeds.

Baselines

We include both non-learning (persistence) and learning-based (LightGBM) baselines that reflect common operational heuristics and strong tabular competitors.

Persistence. The persistence baseline carries forward the anchor state for all forecast steps,

$$\hat{y}^{\text{pers}}_{t,\ell} = y_{0,\ell} \quad \forall t, \ell, \tag{7}$$

and provides a strong reference when soil moisture evolves slowly and short-horizon dynamics are dominated by inertia. Beating persistence indicates the model learns event-driven deviations from zero-change behavior (e.g., wetting and dry-down responses).

LightGBM (tabular). We train gradient-boosted decision trees on summarized history features (e.g., recent precipitation/irrigation aggregates, seasonal encodings, and static descriptors), providing a strong non-deep-learning baseline that performs competitively in structured forecasting tasks.

Accuracy Metrics

We report forecasting quality using complementary error and association metrics computed on predicted levels $\hat{y}$ against ground truth y, aggregated over all forecast steps and depth layers unless otherwise noted. Let i index evaluation points (across samples, steps, and layers), with predictions $\hat{y}_i$ and targets y_i, and let $\bar{y}$ and $\bar{\hat{y}}$ denote sample means.

We report mean absolute error (MAE) as a robust measure of typical error magnitude. Error metrics do not fully describe whether a model captures temporal variability and co-movement. We report the coefficient of determination:

Results

Table 1 summarizes performance on the standard test split and the held-out station split. We report both error metric (MAE) and association metric (R^2) to characterize not only average accuracy but also how well models capture variability. Across architectures, deltas mode substantially improves over levels mode. In particular, deltas-mode models reduce MAE by roughly $\times 2$ relative to the levels-mode TCN and substantially outperform a seasonal climatology baseline.

Discussion and Limitations

Our results support a simple conclusion: deltas mode provides a strong stabilization mechanism for APSIM emulation by focusing learning on day-to-day process rates while maintaining consistency with the last observed state. This benefit is architecture-agnostic and complements the architecture comparison across TCN, and Mamba-style SSMs.

Limitations and sim-to-real gap. Our evaluation criteria focuses how accurately the models emulate APSIM's estimations, where soil properties and management are known and observation noise is absent. In real-world deployment, sim-to-real gaps will arise due to limitation of APSIM's performance, sensor error, unmodeled management actions, and imperfect soil parameterization. This study focuses on developing models that capture physically consistent dynamics, which can potentially be transferred to a model with in-situ observations.

Conclusion

We present an APSIM-to-ML emulation framework for forecasting multi-layer root-zone soil moisture profiles (0–1 m) under a unified protocol across sequence-modeling families. A key finding from our experiments is that deltas mode can substantially improve forecasting accuracy and stability relative to direct level prediction.

Acknowledgments

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 AI Institutes AI-LEAF Award [2023-03616].

References

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

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Kashinath, K.; Mustafa, M.; Albert, A.; Wu, J.; Jiang, C.; Esmaeilzadeh, S.; Azizzadenesheli, K.; Wang, R.; Chattopadhyay, A.; Singh, A.; et al. 2021. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194): 20200093.

Khandelwal, P.; Pallickara, S. L.; and Pallickara, S. 2024. 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*, 233–246.

Kheir, A. M.; Mkuhlani, S.; Mugo, J. W.; Elnashar, A.; and Nangia, V. 2023. Integrating APSIM model with machine learning to predict wheat yield spatial distribution. *Agronomy Journal*, 115(6): 3188–3196.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*.

Shahhosseini, M.; Martinez-Feria, R. A.; Hu, G.; and Archontoulis, S. V. 2021. Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt. *Nature Communications*, 12(1): 1–15.

Yuan, Y.; Zhang, Y.; Wang, Z.; et al. 2022. Probabilistic physics-informed machine learning for dynamic systems. *Mechanical Systems and Signal Processing*, 181: 109506.