Deep Learning Time Series Prediction Strategies for Efficiently Emulating Noah Land Surface Model Soil Moisture Dynamics

Mitchell T. Dodson

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Atmospheric Science
in
The Department of Atmospheric and Earth Science
to
The Graduate School
of
The University of Alabama in Huntsville

May 2025

Approved by:

Dr. Christopher Hain, Research Advisor
Dr. Sundar Christopher, Committee Chair
Dr. Sean Freeman, Committee Member
Dr. Lawrence Carey, Department Chair
Dr. [College Dean Name], College Dean
Dr. Jon Hakkila, Graduate Dean

Abstract

Deep Learning Time Series Prediction Strategies for Efficiently Emulating Noah Land Surface Model Soil Moisture Dynamics

Mitchell T. Dodson

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Atmospheric Science

Atmospheric and Earth Science

The University of Alabama in Huntsville

May 2025

This work examines the ability of deep learning time series generative models to accurately and efficiently emulate the hourly temporal dynamics of the Noah Land Surface Model (Noah-LSM) out to a 2 week forecast horizon, given atmospheric forcings and static parameterization provided by the second phase North American Land Data Assimilation System (NLDAS-2) framework. Results from multiple neural network architectures are compared alongside variations in prediction target, loss function characteristics, and model properties. The most performant model types are subsequently evaluated with respect to forecast distance, daily and annual seasonality, and against a variety of regional scenarios, including several extreme event case studies. Ultimately, we present a software system for developing and testing neural networks that use time-varying and static data to estimate temporal dynamics, with the goal of providing a foundation for similar data-driven modeling techniques to be implemented within the upcoming third phase of the NLDAS data record.

Acknowledgements

You must pay your scholarly debts by thanking those who have provided intellectual guidance, facilities, or financial support for your project; thus, you thank those who have been significantly involved in your work. You must acknowledge any agency providing funding or other resources, and any individual or institution who has granted you permission to reprint material.

If you wish, you may also thank family or friends. You may conclude your acknowledgments with a dedication rather than using a separate dedication page. Your acknowledgments should be brief and consistent in tone with a formal publication.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Accurate characterization of the amount and distribution of water content within the soil column by land surface models is critical for governing land-atmosphere interaction in numerical weather prediction (NWP) (Brocca et al., 2010) (Koster et al., 2010), operational decision making preceding and during drought and flood events (Otkin et al., 2016), and for downstream datasets aiding assessment of vegetation health, crop yield prediction, and fire risk characterization (Case et al., 2023). In order to address these needs, the Noah-LSM was developed to serve as the land surface component coupled to NWP models including the Weather Research and Forecasting Model (WRF), the Global Forecast System (GFS) (Jin et al., 2010) (Mitchell et al., 2005), and climate models including the NCEP Climate Forecast System (CFSv2) (Saha et al., 2014). Noah-LSM also aids National Weather Service forecasts and US Drought Monitor designations within decision support frameworks like the Short-Term Research, Prediction, and Transition project's high-resolution implementation of the Land Information System (SPoRT-LIS) (Case et al., 2022) (Case and White, 2014), and facilitates research and derived product development by providing soil states for NASA Land Data Assimilation System (LDAS) datasets (Ek et al., 2003).

By applying observational and reanalysis data to Noah and other land surface models, NLDAS has provided the community with consistent and quality-controlled multi-model land surface states and associated forcings in a near real-time capacity since 1999 (Cosgrove et al., 2003), with phase 2 of the project also contributing a retrospective climatology extending back to 1979. The first and second generation data products are calculated on a 1/8 degree geodetic grid spanning land-dominated points in the conterminous United States (CONUS) from 25° to 53° North latitude and 125°-67° West longitude, and are released at an hourly frequency (Mitchell et al., 2004) (Xia et al., 2012b). The third phase of NLDAS is currently under development, and aims to implement a wealth of upgrades including new data assimilation techniques and physical parameterizations, an increase in the spatial resolution to $1km^2$, and the expansion of the domain to the full North American continent. As a consequence, the total number of valid land grid cells will increase dramatically from 76,088 in the first two phases to 27,245,580 with NLDAS-3 data products. In addition to the larger domain and updated physical processes used to develop the forcings and land surface states, the NLDAS-3 data suite will feature a variety of derived products. These products are anticipated include gridded climatological anomaly and segmented percentile data, stream routing and discharge estimates, and ensemble mean and spread information using forecast forcings (Kumar et al., 2024).

As the domain size and sophistication of data assimilation systems and land surface models like NLDAS and Noah-LSM continues to grow, a niche develops for methods that can generate reasonable estimates of the dynamics of

numerical models which require less compute time, simplify the runtime environment of the program, and which can be fitted to observational data and then generalized to broader domains without accruing significant additional complexity to the parameterization scheme. Data-driven modeling techniques like deep learning with artificial neural networks (ANNs) are addressing this need by introducing the ability to approximate the highly nonlinear and conditional relationships between arbitrary predictor and target datasets. This flexibility is accomplished by learning a sequence of transformations which are encoded as a composition of alternating high-dimensional matrix operations and element-wise nonlinear functions, and which serve as a mapping from the vector of predictors to a corresponding target vector (Hornik et al., 1989).

In the context of time series physical modeling, ANNs enable the development of a statistically optimal approximation of the relationship between past states, simultaneous covariate data variables, and unknown current or future states. This general principle has a wealth of use cases. Previous literature shows that ANNs are computationally efficient and reasonably accurate for modeling dynamical systems like Lorenz'95 by formulating the problem as a discrete-time estimator of an ordinary differential equation which isn't explicitly known by the model (Fablet et al., 2018). ANNs can also be structured to have useful properties like the ability to estimate the jacobian of the transfer mapping between inputs and future states, even if the system being emulated isn't differentiable (Nonnenmacher and Greenberg, 2021). The same strategy may be applied to forecasting the evolution of datasets like ECMWF Reanalysis v5 (ERA5) in a

local or global domain, however significant challenges emerge as (Dueben and Bauer, 2018) identify. As they describe, ANNs cannot be constrained by default to conserve quantities like energy and water, and unlike numerical models, their handling of the underlying physical processes as a "black-box" mean that identifying the causes of error within the model is difficult and often speculative. Furthermore, Earth system data tend to be highly regionally variable (ex. vegetation types), exhibit nonlinear autocorrelation between multiple variables (ex. temperature, dewpoint, and cloud cover), and are subject to rare but influential outliers (ex. snow and extreme precipitation). As such, although ANNs are adept at handling very nonlinear and conditional problem types, achieving the best performance and interpretability requires the utilization of application-specific knowledge when constructing and evaluating deep learning models.

Within the field of hydrologic modeling, most of the recent literature applying deep learning methods has focused on rainfall-runoff problems, where models forecast the hydrograph of a stream given time-varying atmospheric and land surface states as well as static properties. Inputs are typically considered within a spatial boundary drawn from a watershed outlet where a streamflow station provides the prediction target by directly observing the discharge. To that end, (Kratzert et al., 2018) applies a particular ANN architecture called Long Short-Term Memory (LSTM) networks to modeling discharge from the CAMELS dataset (Addor et al., 2017), which contains daily-resolution streamflow and meteorological forcings alongside parameters describing the topographic, land use, soil, and geologic properties of 671 catchments. They show that models trained on sin-

gle basins often outperform models trained using data from multiple basins within a region, and that subsequent "fine-tuning" of a generalized regional model on individual basins slightly improves model efficiency in many cases. Later, (Kratzert et al., 2019) improves on LSTM model performance by modifying the training strategy to optimize an objective function similar to Nash-Sutcliffe Efficiency (Nash and Sutcliffe, 1970), and by introducing a modification to the architecture that allows for static catchment parameters to be separately provided – and their influence separately investigated – from time-varying inputs. These experiments even out-performed several process-based models that were tuned specifically to the individual test basins. In spite of their black-box nature, (Lees et al., 2022) demonstrates that LSTMs used for daily-scale rainfall-runoff prediction maintain information correlated with physical properties of the catchment's hydrologic state including soil moisture and snow cover, which indicates that they preserve meaningfully interpretable data about their inputs. The general approach of employing LSTMs for discharge forecasting is already being utilized by stakeholders like the United States National Weather Service and River Forecast Center offices in an operational setting with the NASA SPoRT Streamflow-AI product, which uses near real-time Noah-LSM soil moisture estimates and outlooks as an input via the SPoRT-LIS data product (White et al., 2025), (Case et al., 2022).

Relatively few publications have applied deep learning techniques to estimate soil dynamics over a consistently spatially gridded domain, akin to the outputs of process-based models like Noah-LSM. In one instance, (Filipović et al., 2022) applied LSTMs to global daily-scale ERA5 data in order to predict the

3-day evolution of moisture content in an intermediate-depth soil layer. This is conceptually similar to emulating Noah-LSM using NLDAS forcings because ERA5 determines its soil moisture states using the ECMWF Scheme for Surface Exchanges over Land (Balsamo et al., 2009). Additionally, (O. and Orth, 2021) used an LSTM to assist in generalizing in-situ observations at 3 soil depth levels to a regional grid, also using daily ERA5 forcings data as an input, and adjusting predictions to match the pixel-wise gaussian parameters of the ERA5 soil moisture analysis. Both of these approaches use long lead times of 60 days or 1 year, respectively, and make predictions at only a few forecast horizons per execution of the model (3 days and 1 day, respectively).

This work seeks to apply a similar strategy of data-driven modeling for hourly-scale emulation of Noah-LSM over the full NLDAS-2 grid domain, with the goal of generating accurate and computationally reasonable forecasts out to a two-week horizon at three depth levels simultaneously. We will construct a few distinct neural network types suited to this problem structure, compare their results through a variety of bulk statistics and case studies using physical reasoning, discuss lessons learned regarding training methodology, and present a general free and open-source framework for developing time series dynamical estimators using deep learning for gridded physical datasets.

Chapter 2. Background

In this chapter we will elaborate on the history and relevant details of the implementation of NLDAS and Noah-LSM, frame the problem in terms of the difference between numerical modeling and data-driven modeling approaches, and describe the technical properties and pertinent considerations for neural networks intended for time series modeling.

2.1 NLDAS and Noah-LSM: History and Implementation

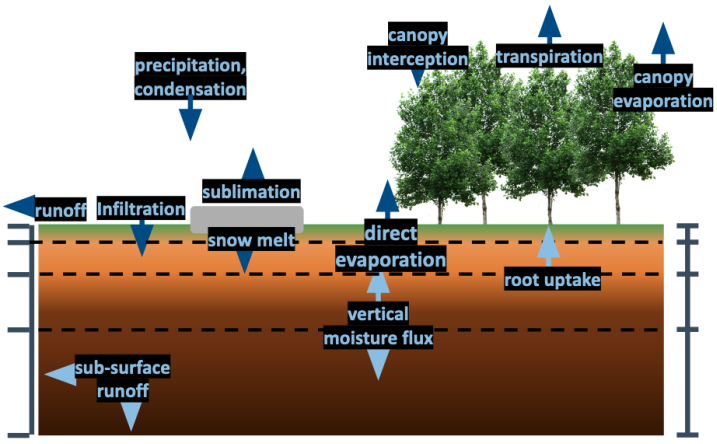

Figure 2.1: Schematic diagram of the feedbacks contributing to the evolution of soil moisture in Noah-LSM

The theoretical framework underpinning Noah-LSM was initially formulated in the 1980s as part of the OSU model, which characterizes boundary layer

moisture and energy fluxes as a 2-layer soil model subject to atmospheric forcings. The model expresses the infiltration and movement of water between the soil layers with the diffusive form of the Richards equation (Mahrt and Pan, 1984), direct evaporation using an analytic approximation of the Penman-Montieth relation in terms of atmospheric stability (Mahrt and Ek, 1984), and basic plant transpiration in terms of vegetation density and soil water content (Pan and Mahrt, 1987). These features form an interdependent system of differential equations that are numerically integrated using a combination of the Crank-Nicholson method and finite-differencing (Chen et al., 1997), which introduces the need for short time steps of 15 or 30 minutes in order for the system to remain numerically stable (Cartwright and Piro, 1992)(Mahrt and Pan, 1984).

The OSU model was later significantly improved, renamed to the first generation of Noah-LSM, and coupled with the NCEP Eta forecast model. Noah-LSM expanded the domain to four soil layers of increasing thicknesses (10cm, 30cm, 60cm, and 100cm), improved runoff dynamics by implementing Philip's equation for infiltration capacity (Schaake et al., 1996), and represented the influence of soil texture on moisture transport by introducing bounds on bare-soil potential evaporation that are determined by the soil composition (Betts et al., 1997) (Mahfouf and Noilhan, 1991). The model also features a significantly enhanced representation of vegetation including a more thorough treatment of canopy resistance via a "Jarvis-type" model of leaf stomatal control (Jarvis et al., 1976) (Jacquemin and Noilhan, 1990), which accounts for the dependence of transpiration on insolation, air temperature and dewpoint, soil moisture content, and vegetation density.

The vegetation effects are scaled by a monthly climatology of normalized difference vegetation index (NDVI) values observed by the NOAA-AVHRR satellite radiometer, which serve as a proxy for green vegetation fraction (GVF) (Gutman and Ignatov, 1998) (Chen et al., 1996), and the depth of root water uptake associated with plant transpiration is determined by a pixel's vegetation class as specified by the Simple Biosphere Model (Dorman and Sellers, 1989). Finally, the model's utility was greatly expanded with the addition of a frozen soil and snow pack parameterization incorporating the thermal and hydraulic properties of fractionally-frozen soil layers, the effects of state changes (Chen et al., 1996) (Koren et al., 1999), radiative feedbacks from partial snowpack coverage, and a snow density scheme (Ek et al., 2003).

Soon after the turn of the millennium, the first generation of NLDAS was under development as part of a multi-institution collaborative effort sponsored by the Global Energy and Water Cycle Experiment (GEWEX) Continental-scale International Projects (GCIP) team. The goal of the project was to incorporate long-term observations of land surface temperature, snow pack depth, and meteorological forcings from multiple sources (in-situ, satellite, radar) into a common framework used to independently evaluate land surface states and energy fluxes with four land surface models including Noah-LSM (Mitchell et al., 2004). Over a domain including the full conterminous United States (CONUS) at 0.125° resolution, the models were allowed to spin up over the course of a year, and soil states were recurrently used to initialize subsequent time steps rather than being "nudged" to correct for drift. Land cover and soil texture classification over the

domain was derived by coarsening the University of Maryland and STATSGO datasets, respectively, from their native 1km resolutions (Hansen et al., 2000), surface geometry and elevation is provided by the GTOPO30 dataset (of the Interior, 1997), and the parameter values for soil hydraulic properties were adapted from observations taken at the University of Virginia (Cosby et al., 1984).

Attention remained on Noah-LSM in the following years as it continued to support NLDAS and other data assimilation and forecasting systems, which led to a series of improvements introduced alongside the next phase of the NLDAS project. A seasonal effect was added to vegetation by scaling the leaf area index (LAI) by the GVF within bounds determined by the plant type, and transpiration was scaled by a root uptake efficiency factor determined by the proximity of soil temperature to an optimum growth temperature (298 K). Several parameters were adjusted including the influence of vapor pressure deficit on transpiration rate, the minimum stomatal resistance for several plant species, and hydraulic parameters for some soil textures. The aerodynamic conductance coefficient – an important factor in the strength of moisture and energy fluxes from the surface – was increased during daylight hours, and a basic anisotropy model was introduced by modifying the albedo of some surfaces in terms of the solar zenith angle (Wei et al., 2011). Snowpack physics were also modified to improve surface exchange coefficients, and to gradually diminish the snow albedo over the time since the last snowfall (Livneh et al., 2010) (Liang et al., 1994). These changes introduce new feedbacks and involve sensitive parameters like LAI which have a strong influence on the model's dynamics (Rosero et al., 2010). The retrospective NLDAS-2 data

Forcing	Unit	Source	Δt	Δx
Precipitation	kg m^{-2}	CPC Gauge observations WSR-88D retrievals	24h 1h	14km 4km
Temperature	K	NCEP NARR	3h	32km
Specific Humidity	kg kg^{-1}	NCEP NARR	3h	32km
Surface Pressure	Pa	NCEP NARR	3h	32km
Wind Velocity	m s^{-1}	NCEP NARR	3h	32km
Incident LW Flux	W m^{-2}	NCEP NARR	3h	32km
Incident SW Flux	W m^{-2}	GOES, NARR	3h, 1h	14km
Green Veg Fraction	%	AVHRR NDVI	Monthly	16km
Leaf Area Index	m^2 m^{-2}	UMD, AVHRR NDVI	Monthly	16km
Snow Water Equivalent	kg m^{-2}	Noah-LSM	14km	

Table 2.1: Atmospheric forcings and other time-varying parameters provided by NLDAS-2 at a 1-hourly resolution on the 0.125° CONUS grid. Data are resampled using spatial bilinear interpolation, then temporal disaggregation according to (Cosgrove et al., 2003). NLDAS forcing files also include values for convective available potential energy, the ratio of precipitation from convection, and surface potential evaporation (calculated as in Mahrt and Ek (1984)), but these three values aren't currently used as inputs to the models. Snow water equivalent estimates are an output of Noah-LSM by default, but are included as a predictor here under the assumption that they can be provided from a separate model or data assimilation source.

record generated after applying these modifications extends back to 1979, and continues to be updated in a near real-time capacity (Xia et al., 2012b).

The NLDAS-2 time-varying retrospective forcings listed in Table 2.1 will serve as the predictors used by the neural networks to forecast the Noah land surface model soil moisture states. Temperature, humidity, pressure, wind speed and heading, and longwave flux are derived exclusively from the National Centers for Environmental Prediction (NCEP) North American Regional Reanalysis (NARR) data product. As part of the downscaling procedure from their native 32km resolution to the 1/8° NLDAS domain, a lapse rate adjustement is applied to the temperature and humidity fields based on the elevation profile. Downward shortwave radiative flux is calculated using a blend of NARR and hourly Geostationary Operational Environmental Satellite (GOES) data, with a ratio-based bias correction based on (Berg et al., 2003) applied to account for a known positive bias in NARR-reported downward shortwave flux, and to mitigate discontinuities arising from the merger the two data sources (Pinker et al., 2003) (Xia et al., 2012a). Precipitation receives a special treatment in order to ensure sufficient spatial resolution and consistency; the Climate Prediction Center (CPC) daily gauge-based product (Chen et al., 2008) serves as the baseline, which is temporally disaggregated to 1 hour resolution using National Weather Service WSR-88D radar retrievals (Fulton et al., 1998). In regions lacking radar coverage, the disaggregation is completed using a weighted combination of the CPC's satellite-derived estimates from morphed passive-microwave and infrared observations (CMORPH) (Joyce et al., 2004), and the CPC Hourly Precipitation

Dataset (HPD), with NARR data as a final fallback (Baldwin and Mitchell, 1997). Although both the LAI and GVF vegetation parameters are based on multi-year monthly averages, they are disaggregated to an hourly resolution in order to be smoothly variable (Wei et al., 2011), and are thus treated like an atmospheric forcing in this work.

2.2 Distinctions in Modeling Techniques

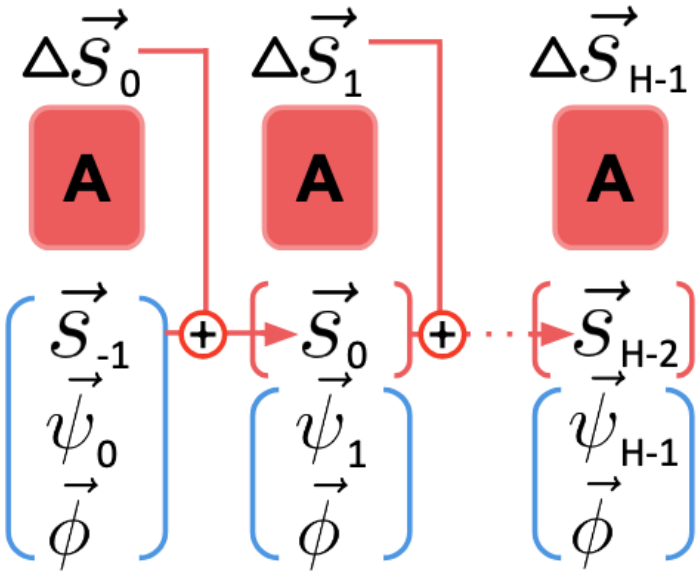

Figure 2.2: Diagram of a self-cycling discrete-time dynamical system with no hidden state. At each time, nonlinear operator **A** maps an initial state $\vec{s}_{k-1}$, exogenous forcing $\vec{\psi}_k$, and time-invariant parameters $\vec{\phi}$ to a new state $\Delta\vec{s}_k$, used to initialize the subsequent time step, and so forth until H predictions have been made.

2.2.1 Noah-LSM as a Discrete Dynamical System

At its core, Noah-LSM is a collection of coupled differential equations that express the total derivative of land surface states $\frac{d\vec{s}}{dt}$ in terms of the current state $\vec{s}$, forcing $\vec{\psi}$, and time-invariant properties of each grid cell $\vec{\phi}$. Since the model is implemented as an algorithmic procedure and is not continuously differentiable, we will use the notation $\frac{\Delta \vec{s}_k}{\Delta t} \approx \mathbf{A}(\vec{s}_{k-1}, \vec{\psi}_k, \vec{\phi}, \Delta t)$ to refer to the model as a transition function evaluated at a discrete time step Δt, such that the system evolves as a dynamical system described by Equation 2.1.

$$\vec{s}_k = \vec{s}_{k-1} + \mathbf{A}(\vec{s}_{k-1}, \vec{\psi}_k, \vec{\phi}_k, \Delta t) \cdot \Delta t \tag{2.1}$$

Here, $\vec{s}_k$ refers to the model's dynamic state variables like snow pack depth, soil moisture and temperature, and canopy storage, $\vec{\psi}_k$ encodes the covariate atmospheric variables from Table 2.1 (which are derived from weather forecasts or observations), and $\vec{\phi}$ includes time-invariant coefficients of the governing equations like vegetation type/fraction, soil texture, and slope/elevation. As a matter of convention, the time step $k = 0$ refers to the first perturbation of the initial state after applying the model's first increment change estimate, and includes the forcings that informed that perturbation, consistent with the relationships depicted by Figure 2.2.

To generate a time series, Noah-LSM numerically integrates the system of equations using Euler and Crank-Nicholson techniques, which explicitly evaluate the differential equations at several time intervals per computational time step

in order to estimate the nonlinear change in state, which evolves continuously in the real system being modeled. It is crucial that the increment change in time remains small between evaluations of the model (15min for NLDAS-Noah) to mitigate truncation error from the assumption of local linearity (Mitchell et al., 2004) (Cartwright and Piro, 1992). As Figure 2.2 suggests, the model does not retain any "hidden" internal information that is updated between timesteps; at each point, the information available to determine the new change in state is limited to the input domain of the model.

2.2.2 Process-based vs Data-driven Models

The process-based approach of numerical models like Noah-LSM is practically and epistemologically distinct from data-driven techniques like deep learning. In a process-based paradigm, the inductive biases that govern the model's behaviors can be explicitly understood since they are based on a characterization of the physical system which is derived from theoretical knowledge. Some uncertainty is introduced by the input data, and is shared among all model types; this includes uncertainty from noise and interpolation of forcing observations, discrete treatment of surface types, etc. Aside from that, model error in process-based models arises from sources including inadequacy of the theory for describing the system (that is, phenomena which are neglected or misrepresented, and become a source of unexplained variance), and truncation error accumulated from the approximation techniques used to solve the model's governing equations. Explicit understanding of the reasons for the model's behavior has the advantage of be-

ing interpretable, in the sense that particular systems within the model can be independently evaluated and blamed for contributing uncertainty. Additionally, granting the ability to impose absolute constraints within the model structure ensures the outputs fully adhere to some physical requirements, such as conservation of water and energy. Nonetheless, the onus falls entirely on the model developer to adjust many details of the implementation of the processes. The act of tuning a numerical model's parameters often implies postulating a source of uncertainty, addressing it by manually manipulating coefficients or introducing new systems within the governing equations, and then evaluating the impact of the changes using correlational analysis with a subset of the available data. This can be a laborious process, and typically results in the gradual accumulation of feedbacks and complexity within the model.

In contrast, many data-driven approaches to modeling physical systems – deep learning in particular – sacrifice the explainability and rigorous physicality of their estimates in exchange for developing a statistically optimal approximation of the relationship between the input and output domains by any means available. Although the overall algorithmic structure of the ANN is established by the developer – typically based on broad heuristics from past literature and experimentation – very little control can be asserted over the particular means by which predictions are determined from inputs. Instead, the effectiveness of the ANN's performance is characterized in terms of a differentiable loss function (also known as the objective or cost function) which may be defined by the developer,

and which is fundamentally (though indirectly) important for determining the solution developed in the training phase.

An ANN's learnable parameters refer to the real-valued elements of a series of arbitrarily-sized square matrices encoding affine transformations. Each "layer" of the ANN is comprised of one of these affine transformations followed by an element-wise nonlinear operation on its output, and the full ANN typically consists of multiple layers which are combined via composition or any kind of differentiable arithmetic operation. During training, all of the ANN's learnable parameters are iteratively adjusted by estimating the gradient of the loss function with respect to the parameters (given batched subsets of the predictor/target data pairs) then determining the direction and magnitude by which the parameters in each of the layers should be modified with an algorithm called backpropagation (Rumelhart et al., 1986). The general strategy of determining the sensitivity of the loss landscape to changes in the ANN's parameters, and tweaking them accordingly, is referred to as gradient descent. Given at least 2 layers (which constitutes the definition of a *deep* ANN), the network is theoretically capable of expressing an arbitrary decision boundary or multivariate function given a sufficient number of parameters (Hornik et al., 1989). This high level of expressivity enables the network to learn complex relationships and generalizations among high-dimensional parameters, given repeated exposure to instances of these relationships during training.

The principle of ANNs using high-dimensional nonlinear correlations rather than explicit processes to model the correspondence between two datasets is pow-

erful because they can approximate a highly nonlinear regression without numerically integrating a complex algorithm, and they can learn their parameterization based on a large volume of data without manual intervention. The quintessential drawback of relying on a black-box approach, though, is that models may perform poorly for no apparent reason, or may perform well for a fraught reason. For example, in one commonly-invoked anecdote described by (Lapuschkin et al., 2019), an image classifier over-performing but failing to generalize at identifying horses in a grassy field was found to actually rely on the presence of the watermark of a particular equestrian photographer whose work was a part of the training data. As such, only regarding bulk statistics and loss performance as indicative of a model's success is insufficient to consider it trustworthy. In a data-driven paradigm, then, the role of the ANN developer is to facilitate effective and reliable learning through careful training data curation, congnizant loss function and model architecture design, and thorough evaluation of the model's behavior in local and global scenarios throughout the input domain in order to ensure the effectiveness and consistency of predictions in a variety of inference settings.

Furthermore, it is important to note that in the current scope of this work, it is not possible for the ANNs to leverage their expressivity to out-perform the numerical model. Since the ANNs presented here are merely emulating the processes that are programmed into Noah-LSM, it isn't reasonable to expect the models to form a more accurate representation of real soil dynamics. Nonetheless, it is conceivable that future work could utilize a similar approach to (O. and Orth, 2021) in order to integrate observational data into the training domain alongside

model data, or to use it as a prediction target. The merger of these two data sources could aid in improving a data-driven model beyond the limitations of a numerical model's structure.

2.3 Deep Learning of Time Series

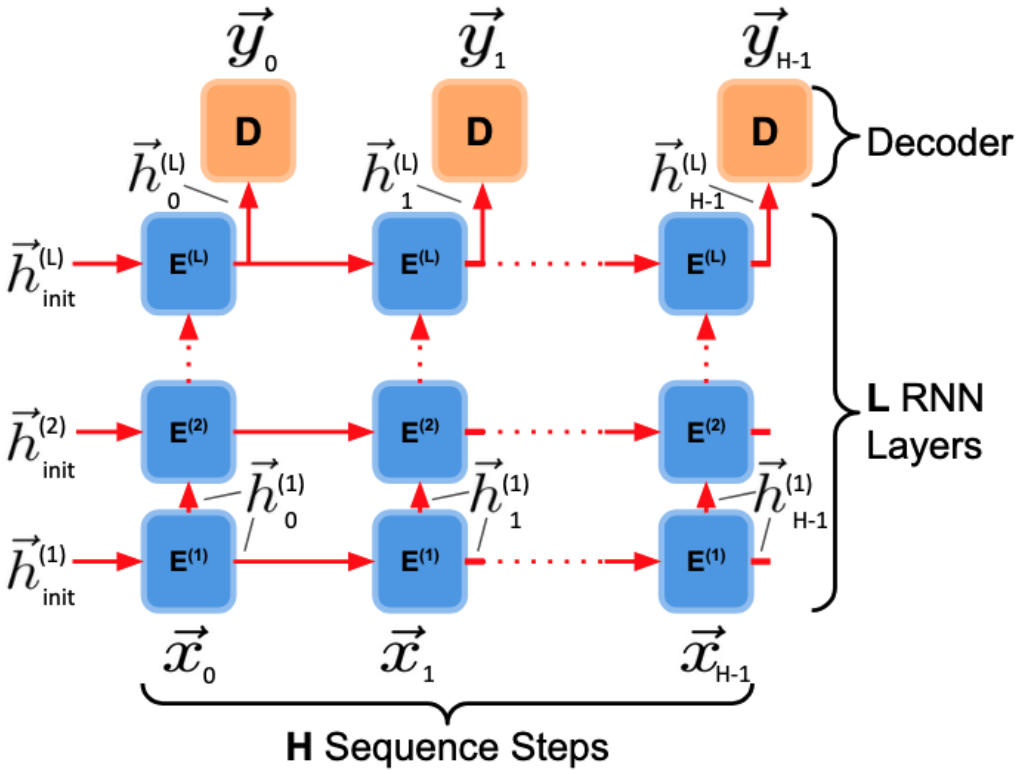

Figure 2.3: Schematic representation of an abstract sequence-to-sequence RNN with multiple layers.

The most simple variant of an ANN is referred to as a feed-forward neural network (FNN), which consists of a series of composed layers (see previous section) mapping the input vector directly to the output. Although a FNN is theoretically

capable of simulating Noah-LSM in the manner of Figure 2.2, inductive biases are commonly introduced in model architectures in order to promote efficiency, explainability, stability, and parsimony. For example, most neural networks used for sequence modeling like the recurrent neural networks (RNNs) in Figure 2.3 maintain one or more "hidden" latent parameters $\vec{h}$ with an arbitrary number of dimensions. This vector is modified and passed along by each subsequent iteration, giving the network the ability to make temporal generalizations and propagate information to future predictions. Although $\vec{h}$ is typically difficult to interpret directly, the gradient descent process incentivizes the network to preserve and consolidate information that is needed to accurately generate the full sequence of predictions. This construct is promising for improving the models' ability to estimate soil dynamics because characteristic drydown curves are known to exhibit hysteresis depending on their recent patterns of wetting and drying (Haines, 1930).

$$\begin{aligned} \vec{h}_k^{(1)} &= \mathbf{E^{(1)}}\left(\vec{h}_{k-1}^{(1)},\, \vec{x}_k\right) \\ \vec{h}_k^{(j)} &= \mathbf{E^{(j)}}\left(\vec{h}_{k-1}^{(j)},\, \vec{h}_k^{(j-1)}\right) \\ \vec{y}_k &= \mathbf{D}\left(h_k^{(L)}\right) \end{aligned} \tag{2.2}$$

The RNNs discussed here will follow the general structure described by Equation 2.2, which is consistent with the abstract architecture diagrammed in Figure 2.3, and follows the typical approach for constructing RNNs as outlined in (Russell and Norvig, 2020). The architecture consists of L encoder layers ($\mathbf{E}^{(1)}$-$\mathbf{E}^{(L)}$) which generate H top-layer latent vectors corresponding to each of

the prediction times, and 1 layer of decoder weights **D** which converts each of the top-layer latent vectors to a prediction for the increment change in state $\Delta\vec{s}$ between the current timestep and the next one. Parameters for **E** and **D** are shared across timesteps, however each of the L layers have distinct parameters. Here, $\vec{h}_k^{(1)}$ is a member of the time series of first-layer latent vectors given the arguments $\vec{x}_k$ and the latent vector from the previous first-layer timestep $\vec{h}_{k-1}^{(1)}$. Moving upward, $\vec{h}_k^{(j)}$ is an intermediate-layer latent vector such that $j \in [2, L]$, and when $k = 0$ (the first horizon timestep), $\vec{h}_{-1}^{(s)} = \vec{h}_{\text{init}}^{(s)}$ for any layer s. In many cases, the initial hidden states $\vec{h}_{init}^{(s)}$ are randomized or set to zero, however in this project we use a separate spin-up RNN to establish them, as will be discussed in the Methodology section.

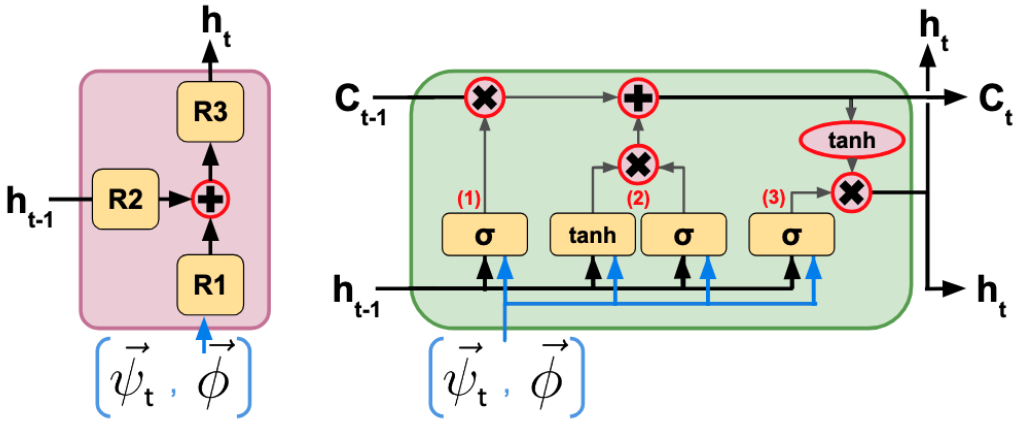

Figure 2.4: Schematic representation of individual RNN cells, the naïve RNN (left), and the LSTM (right)

The term RNN does not refer to a specific neural network architecture, but rather the category of architectures that share their weights across sequence steps. The sub-network unit of weights applied at each timestep (**G** or **E** in Figure 3.14)

is referred to as a RNN cell, and must contain internal structure for converting inputs from lower layers and the previous timestep into an output. Unlike traditional ANNs which use backpropagation to train on single input-output pairs, each cell must balance the penalty of its loss gradient between contributions from the input at each timestep with contributions from the latent vector of the previous timestep. To accommodate this difference, RNNs are trained using a special type of gradient descent called backpropagation through time, which balances the loss with respect to the two input sources, and accumulates it across all timesteps before updating each cell's parameters.

The most basic form of the RNN architecture (diagrammed on the left side of Figure 2.4) consists of 3 separate affine operations. As shown in the figure, R1 and R2 convert the lower-layer and previous-timestep vectors, respectively, into two uniformly-sized internal latent vectors. The sum of these internal vectors are transformed by a third affine operation, R3, into the cell's output, which is passed along to the higher layer and subsequent steps within the same layer.

This style of naïve RNN is vulnerable to the so-called vanishing or exploding gradient problem, which arises from the fact that the cell output is the direct product of learned matrix operations. Since during training each cell's weights are updated recurrently based on many sequence steps, the loss gradient with respect to each learned parameter can become incredibly sensitive to changes in each parameter. This can cause weights to become very close to zero, or to diverge to very large numbers, which halts the learning process (Mozer, 1995). Furthermore, since the latent state passed between sequence steps undergoes a

nonlinear transformation at each instance of the cell, it is more tenuous for the network to sustain information over a long context of past observations.

The LSTM architecture addresses these shortcomings by maintaining a separate hidden state $\vec{C}_t$ called the context vector. Rather than being generated by a matrix operation at each step, the context vector is only modified from the previous step's value using the output of a series of three "gates." These gates (numbered in Figure 2.4) include (1) the "forget gate", which uses a FNN to select a vector of values in the range (0,1). The forget vector is multiplied element-wise by $\vec{C}_{t-1}$ in order to selectively emphasize or diminish its activation. The "update gate" (2) transforms the inputs into a new coefficient vector in the range (-1,1), which is added to the context vector in order to retain information from the current time step. Finally, the "output gate" (3) generates a vector of multiplicative coefficients in the range (0,1) used to scale the new context vector $\vec{C}_t$ to the output latent state $\vec{h}_t$ (Hochreiter and Schmidhuber, 1997). The context vector remains stable compared to a hidden vector that is recurrently operated on by the same weight matrix, which facilitates the network to learn over a longer sequence interval.

The Transformer architecture has dominated many sequence modeling tasks in recent literature thanks to the key innovation of multi-head self-attention (MSA), which enables the architecture to learn complicated relationships between individual members of the input sequence regardless of their relative position. Additionally, transformers are efficient to train in parallel, unlike RNN architectures which rely on a chain of sequential operations, which makes them straightforward

to train on a massive scale (Vaswani et al., 2017). The results for natural language processing (NLP) (Devlin et al., 2019) and image classification (Dosovitskiy et al., 2021) tasks are impressive, however there are several key drawbacks that make them less appealing for a time series generation task like this one.

First, the memory cost of a transformer scales quadratically with sequence length since MSA learns parameters relating every possible combination of input steps. This is compounded with the fact that the full input sequence needs to be re-initialized with every step during inference. These properties are a direct trade-off with the Transformer's ability to train separate sequence steps in parallel. Furthermore, unlike RNNs and CNNs, Transformers don't have an inherent notion of order. In problems like NLP where sequence position conveys some information, a simple form of locality is introduced by adding a positional embedding vector directly to the inputs. Prior literature shows that transformers equipped with positional embedding still perform no better on basic time series forecasting tasks than a simple 2-layer FNN (Zeng et al., 2022). For these reasons, although MSA is a very powerful tool for many applications, we will be neglecting this very popular architecture in this work.

Chapter 3. Data and Methodology

3.1 Dataset Overview

This section includes a description of the storage of and framework used to interface with the data, insights on the value distributions and spatial variability of the input forcings from NLDAS-2, as well as a look at similar bulk properties of the target soil moisture states and governing processes within Noah-LSM. In this work, we define our valid domain to include all points falling within the conterminous United States, excluding those points within the NLDAS-2 domain falling with Canada and Mexico. We also exclude points that are classified as "water," "bedrock," or "other" in the STATSGO dataset, since they don't correspond to meaningful hydraulic properties, and have idle time series. What remains are 50,875 candidate grid cells within a 224x464 pixel domain.

3.1.1 Data Storage System

The data used in this project were acquired from the Goddard Earth Sciences Data and Information Services Center's Distributed Active Archive Center (GES DISC DAAC) in May of 2024. The DAAC archives the NLDAS-2 forcings and corresponding Noah-LSM model outputs as separate hourly files in a GRIB1 format, of which we downloaded the full 12-year time series from January 1, 2012

to December 31, 2023. This subset constitutes 210,384 files with a total size of just over 891.38 GB.

Since this project concerns developing 2-week time series of the forcings on a per-pixel basis, it would be inefficient to extract data from several hundred files for each sequence sample. Furthermore, it is widely recognized in deep learning that input/target pairs from heterogeneous datasets should be globally shuffled during the training process, as outlined by (Nguyen et al., 2022). This is because local subsets may have distribution characteristics that are distinct from the full dataset, so as the model trains on an unshuffled dataset, the loss gradients it experiences may encourage it to converge on a locally-optimal solution that does not generalize well to the overall task. Shuffling is especially salient for geoscience datasets like this one, which are highly spatially and temporally heterogeneous. With this in mind, the overhead from file I/O operations would be prohibitive for sporadically drawing samples from throughout the GRIB dataset during training or inference.

To address this problem while maintaining the spatiotemporal structure of the data, we develop a custom file standard using the HDF5 format – hereafter referred to as the `timegrid` – and extract the full 12-year NLDAS and Noah-LSM record as a collection of them. The HDF5 format offers a system for memory-mapped data chunking in multiple dimensions, which means the data therein can be sparsely buffered and accessed on a per-chunk basis without loading the entire file into memory: a considerable advantage for thoroughly shuffling or accessing subsets of contiguous data within large files. In practice, each timegrid contains

3 years of data covering 1/6 of the spatial domain, and stores the 4-dimensional time-varying data (time, latitude, longitude, data type), 3-dimensional static data (latitude, longitude, data type), and timestamps alongside a string-serialized attribute dictionary. The attributes contain information on abbreviated and full data type names, ordering, units, and sources, which are sufficient to inform a variety of accessor methods with a wealth of downstream use cases.

3.1.2 Regional Variance of Input Data

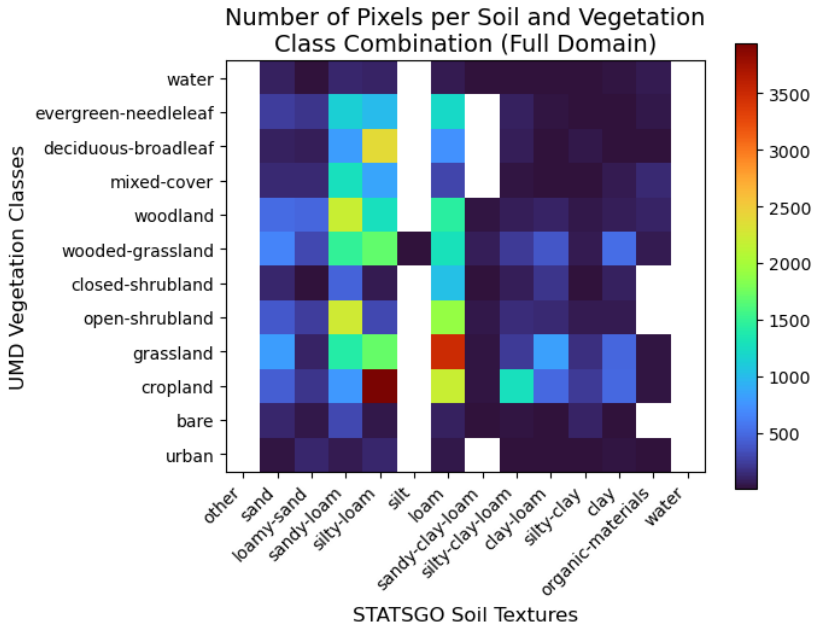

Figure 3.1: Full-domain combination matrix of vegetation and soil classes

One important aspect of the NLDAS-2/Noah-LSM datasets is the relationship between static and dynamic parameters, and the regional variance of both of these input types. Given any particular forcing time series, the subsequent

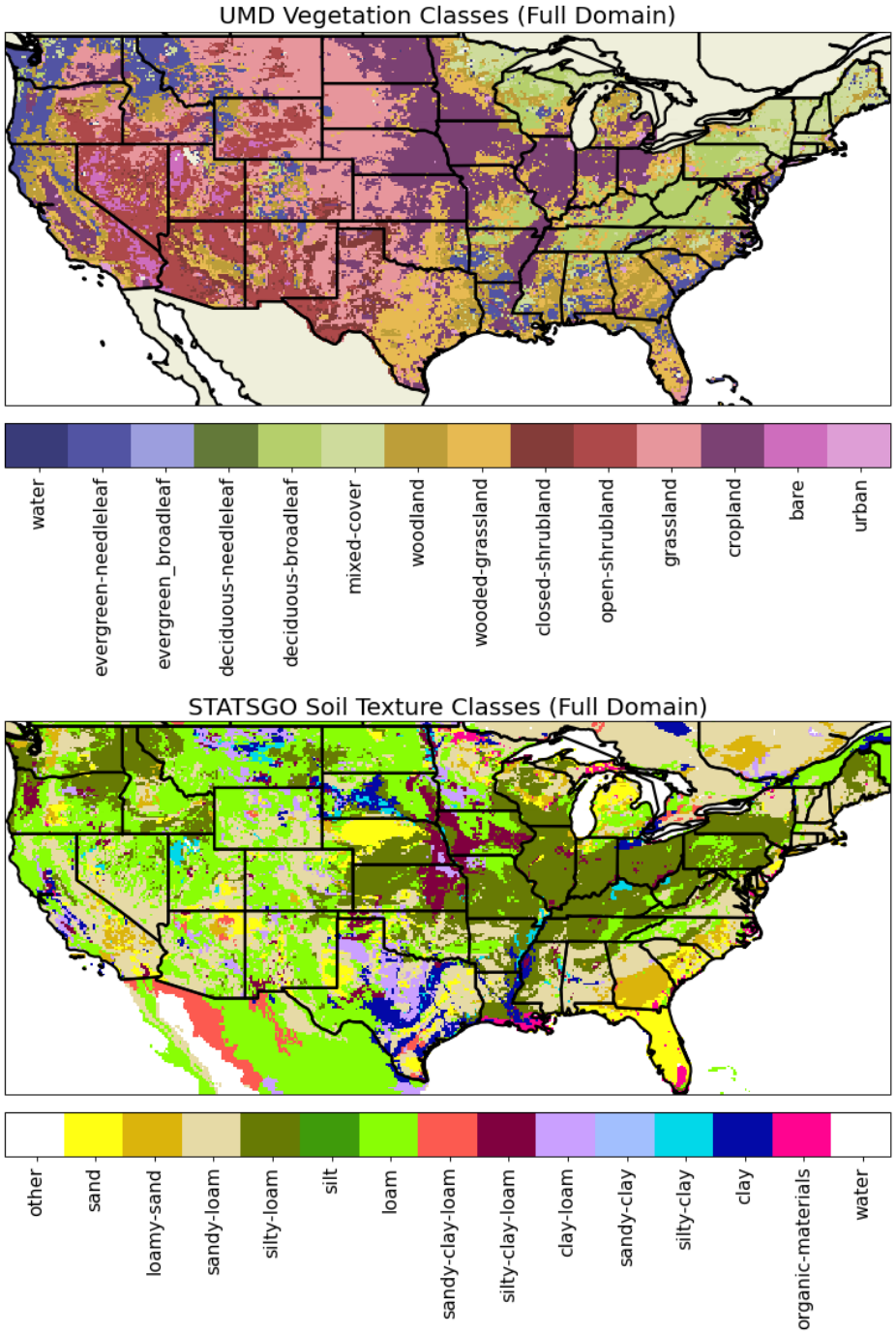

Figure 3.2: Spatial distribution of vegetation and soil classes

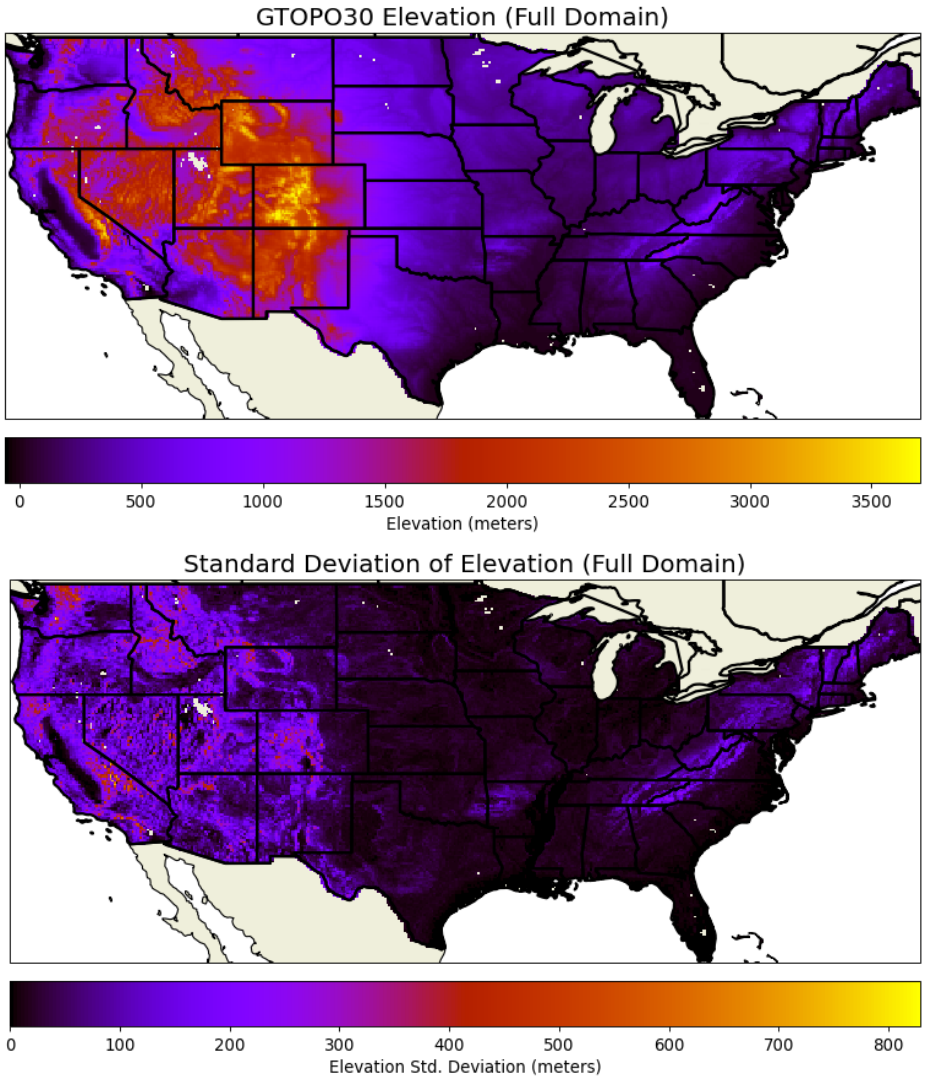

Figure 3.3: Elevation and standard deviation of elevation on the CONUS domain

land surface response is modulated by values for vegetation type, soil texture, elevation, standard deviation of elevation, slope, slope aspect, and a drainage parameter referred to as slope type, which are consistent per-pixel throughout the dataset. In this work, we will only train models informed by the first four. Slope and aspect were left out because within Noah-LSM they are only used within the snowpack parameterization (Barlage et al., 2010), and snow is generally not a target variables for the models presented here. Furthermore, slope correlates strongly with the standard deviation of elevation. Slope type isn't well-documented in the model, and wasn't considered during training, but had a noticable effect on some of the results. In retrospect, these inputs may have increased the information available to the models for estimating the transference of water from snow melt to the soil layers, especially in mountainous regions like the Rocky Mountain and Sierra Nevada ranges. An evaluation of the impact of these parameters on model results is left for future work. Elevation parameters aren't directly utilized within Noah-LSM, but it is used to perform orographic regressions on pressure, temperature, humidity, and precipitation while resampling forcing data. Thus it is included as a training input because it could be useful as a predictor that indicates processes relevant to mountain snowpack dynamics.

As described in the background, the vegetation classes encode the properties of the canopy relevant to precipitation interception and land surface shading, the efficiency of plant transpiration at removing water from the soil, and the number of layers from which water is drawn (that is, the rooting depth). Since the vegetation parameter is discretely categorical within the Noah-LSM algorithm,

we employ a special method of introducing them into the model called class embedding, which is elaborated upon in the next subsection. The soil texture class corresponds to a variety of hydraulic properties identified by (Cosby et al., 1984), which include field capacity, hydraulic conductivity, porosity, wilting point, matric potential, and Skempton's pore water pressure ("B" parameter). These describe physical characteristics of the soil-water system including the rate of downward percolation of water, the efficiency and limits of plant water uptake, the speed of infiltration, and the total amount of water that soil can contain per unit volume. The basic observable feature of soil that determines all of the hydraulic properties is the size distribution of its constituent particles, which is often articulated as the mass fraction of sand, silt, and clay components within the soil. In the interest of providing the models with real-valued inputs having relatively low dimensionality, these three texture components will serve as the representation of soil texture for the ANNs trained here.

The interplay between plant water uptake and soil water dynamics as governed by the static inputs represents a considerable source of complexity within Noah-LSM. Furthermore, as Figure 3.1 demonstrates, the distribution of combinations of soil and vegetation categories is extremely non-uniform, which makes it more difficult for ANNs to learn solutions that are general. Figure 3.2 shows the geographic locations of vegetation and soil classes. The most common class combination is silty-loam soil types juxtaposed with cropland, with 3,945 members found dominantly in the Midwest and lower Mississippi river basin, with some contribution from the Columbia Plateau in Washington and Eastern Nebraska.

Next most common are the 3,490 pixels in loamy grasslands, which are distributed widely throughout the West US including the high plains, Western Texas and New Mexico, Utah, and Idaho. The remaining combinations all have fewer than 2,500 members within the domain. Sand and clay dominated soils form the upper and lower extremes of soil particle size, respectively, and thus have rather different soil water characteristics. The sandiest soils are found in Southern Coastal Plains, Michigan, Texas, the Nebraska Sandhills, and the desert Southwest. Clay soils are relatively rare compared to silty and sandy soils, and considerably more spatially heterogeneous. They are mainly found in tight groupings around Central Texas, the Mississippi Alluvial Plain, Eerie Lake Plains, and the Missouri River Basin in South Dakota. Despite their infrequency, clay soils span the full range of surface classes.

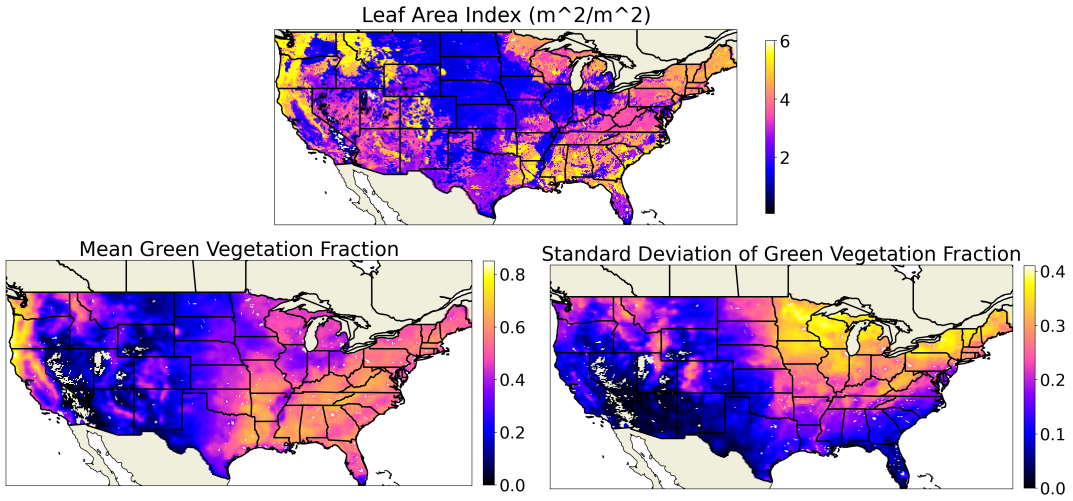

Figure 3.4: Gridded mean and standard deviation of vegetation input parameters (2012-2023)

Although they vary smoothly on an hourly basis, the LAI and GVF parameters are similar to static parameters in that they cycle consistently per-pixel on an annual basis (rather than dynamically changing based on variable atmospheric conditions), and modulate the soil water dynamics via through their effect on the vegetation parameterization. As Figure 3.4 indicates, the densest annual-averaged canopy cover corresponds to evergreen needleleaf surface types, and there is almost no canopy over croplands and grasslands of the Midwest, California Valley, and the Great Plains. The greenest satellite-derived vegetation covers the West Coast and Sierra ranges, followed by the South and Northeast. The standard deviation of GVF indicates the regions of most significant seasonal variability, which corresponds to deciduous-dominant locales.

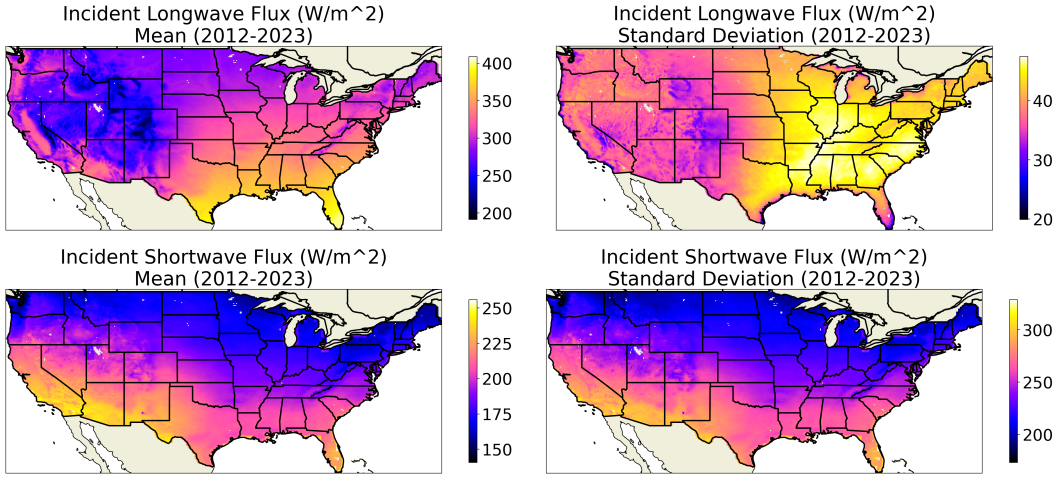

Figure 3.5: Gridded mean and standard deviation of radiative forcings (2012-2023)

In addition to the substantial regional variability of the static parameterization of Noah-LSM, there are considerable regional and seasonal differences in the NLDAS-2 atmospheric forcing time series. Figure 3.5 shows the mean and

standard deviation of radiative features over the full spatial and temporal domain, which demonstrates the distribution of annually-averaged downwelling radiation. Feedback from terrestrial emissions mean that longwave radiation is highest in regions that are generally cloudier, have warmer land surface temperatures, and are lower in elevation. The shortwave flux is highest at lower latitudes due to Earth's axial tilt, and in arid regions where there are fewer clouds.

Figure 3.6 displays the same statistics for each of the other input forcings. The precipitation dataset includes liquid rain and snow, and reflects that the highest average values fell in the Pacific Coastal, Cascade, and Sierra Nevada, followed by more moderate values in the broad expanse of the Southeast, with the deep south of Louisiana and Mississippi seeing the strongest variability in precipitation throughout the year. Pressure mainly correlates with altitude, and higher latitudes tend to see the most variance, likely due to the prolonged influence of Rossby waves. Humidity and temperature also strongly correlate with temperature and proximity to the warm gulf, where the highest humidity variance is in the southern states thanks to seasonal intrusions of warm, moist air. The highest temperature variance is found in the high plains where there are considerable intrusions of arctic air in the winter, and solar heating combined with warm air advection from the terrain-driven low level jet in the summer. The wind is locally strongest where there is channeling from mountain slopes, with a broad region of relatively high wind speeds in the plains due to the open terrain.

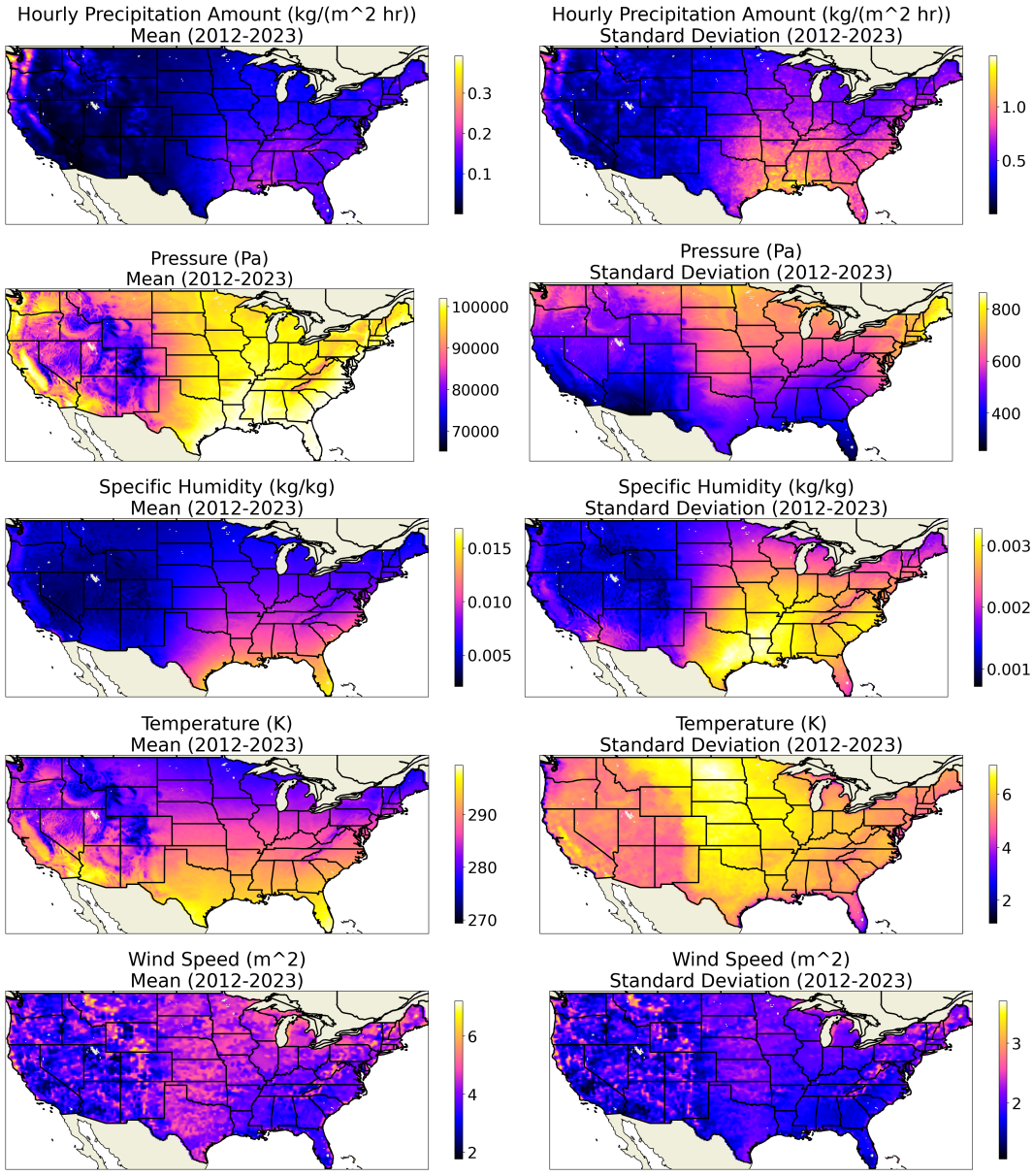

Figure 3.6: Gridded mean and standard deviation of input forcings (2012-2023)

3.1.3 Handling Snow

Figure 3.7 shows the mean and maximum snow accumulation throughout the dataset. Snow is particularly difficult to account for in the ANNs because it is relatively rare and highly regional, but has a profound influence on the soil dynamics. The presence of snow significantly modifies the surface albedo and roughness length, captures and stores precipitation as an additional state variable, and represents a new source of water for the soil column as it melts (Koren et al., 1999). The first exploratory models we trained treated snow as essentially an additional soil layer, and predicted the increment change in its value alongside the other soil states. Since snow is such a transient phenonmenon within the training dataset the ANNs would consistently predict close to zero change, even in snowy conditions, since doing so results in the lowest loss in most cases.

Subsequent models trained to exclusively target the increment change in snow water equivalent showed the same hesitancy to make non-zero predictions. In order to loosen the requirement that these models must always output zero change when there is no snow present, we modified the loss function used to train

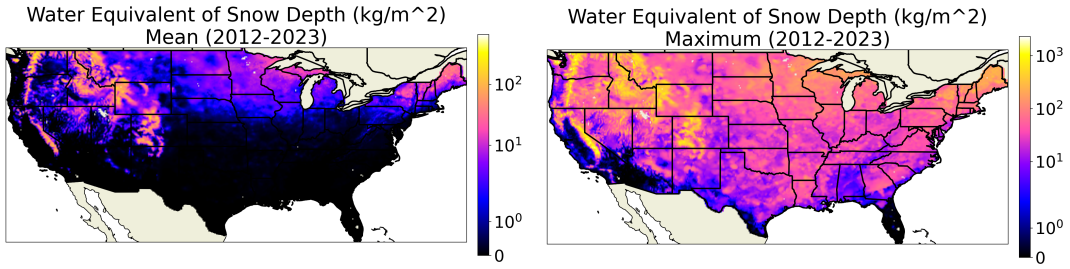

Figure 3.7: Mean and maximum accumulated snow amounts on a log scale (2012-2023)

the snow-only models such that predicting negative increment change values when there is no snow present will not be penalized. When combined with a post-processing step that truncates negative accumulated snow amounts at zero, this strategy focuses the gradient descent process exclusively on samples where the snow pack is relevant.

Figure 3.8 displays a sample from one of these loss-modified snow models, which captures the subtleties of extreme snow events, and maintains negative increment predictions when there is no snow present or accumulating. Curiously, the negative predictions of the model during the warm-season sample cycle on a diurnal basis, and may represent a hypothetical snow melt rate, which is an emergent property since the loss function wasn't applied in such scenarios. In any case, although these are encouraging preliminary results, further refinement of this strategy is out of the scope of this project. We will use the true snow amount as an input to the soil moisture ANNs presented here in order to prevent compounding error, and take the apparent validity of this approach as an

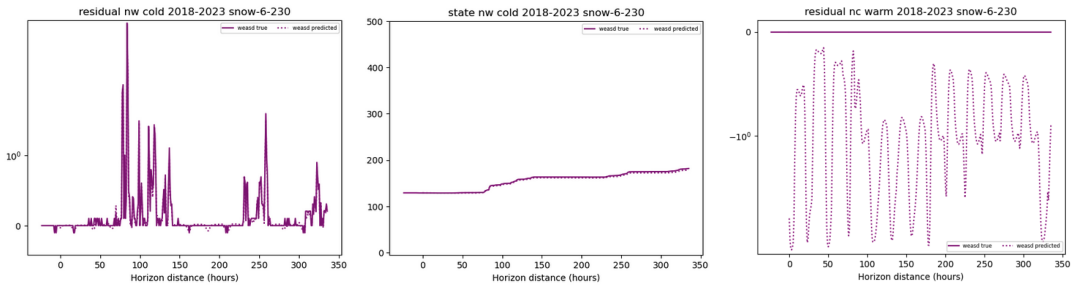

Figure 3.8: Samples of snow-only model predictions vs true values after loss function manipulation, including the increment change for a significant snowfall event (left), the accumulated state for the same sample (center), and an example of the increment outputs of a different warm-season sample (right).

indication that separate ANNs designed specifically for snow water equivalent estimation may be used to initialize soil moisture emulators in the future.

3.1.4 Input Data Value Distributions

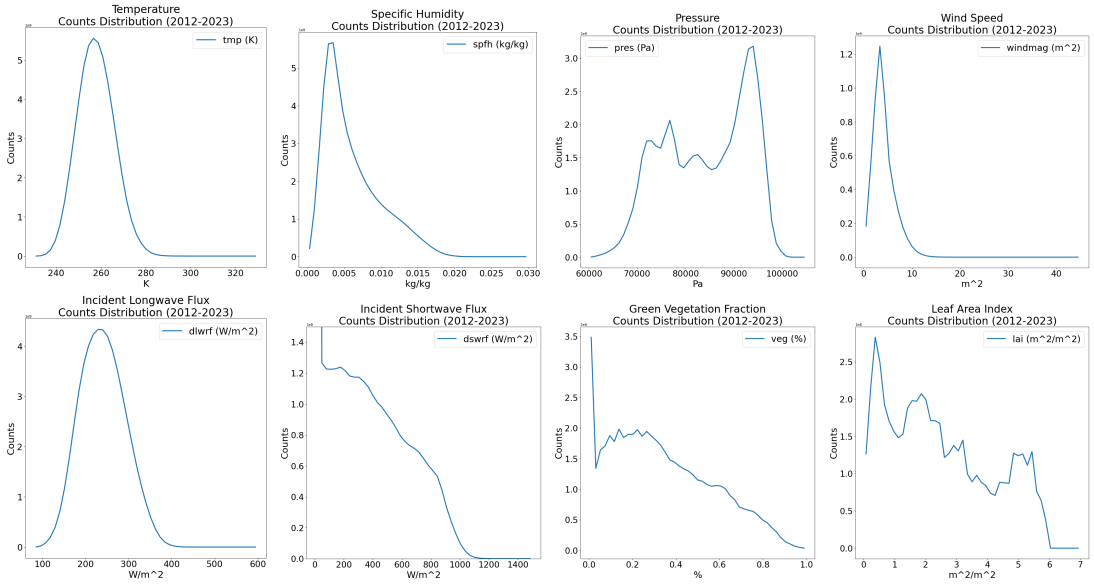

Figure 3.9: Overall distributions of dynamic model inputs (2012-2023)

The count distributions of dynamic inputs with the exception of precipitation are displayed by Figure 3.9. Of these only temperature and incident longwave flux follow generally gaussian distributions. Specific humidity is strongly skewed toward zero since its upper bound is limited by temperature. Pressure has a global peak just below sea level pressure, with relative maxima associated with the mountainous terrain of Appalachia and the West. Windspeed has a strong peak around 5 ms^{-1}, with a long tail of outliers. Shortwave flux is not entirely smooth, which is likely due in part to enhanced cloud cover from orographic ef-

fects, judging by the spatial distribution in Figure 3.5. The vegetation parameters are also strongly non-gaussian owing to their regional and seasonal heterogeneity, and the discrete differences in vegetation categories.

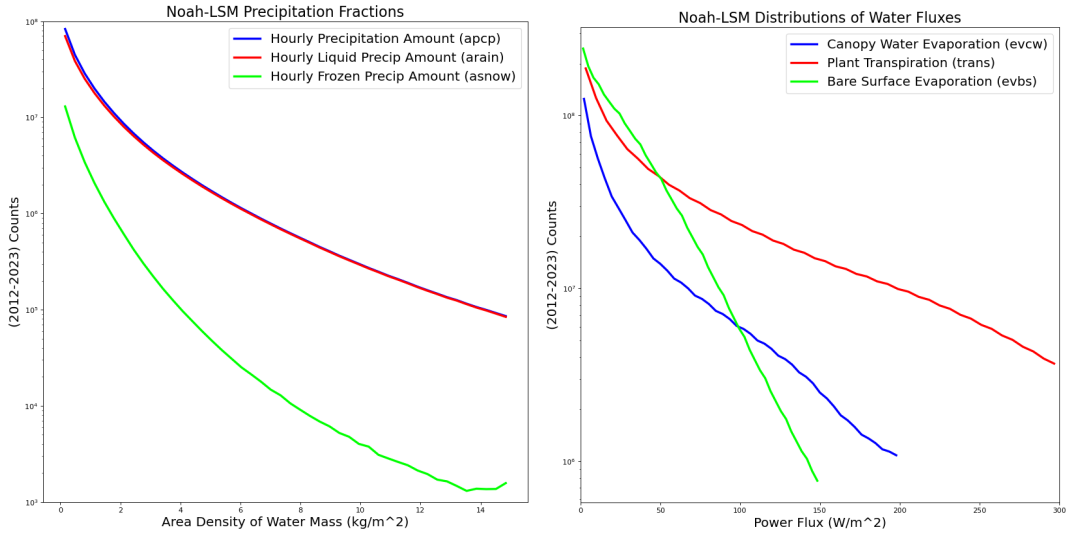

Figure 3.10: Overall distributions of precipitation types (left) and fluxes removing water from the surface system (right), both on a logarithmic axis (2012-2023).

Figure 3.10 compares the distributions of precipitation types, and those of the fluxes that remove water from the system. Liquid precipitation represents the vast majority of the total hourly precipitation amount, especially for strong precipitation events. Plant transpiration is the dominant sink for soil moisture content, with bare surface evaporation mainly limited to lower rates. Evaporation from the plant canopy can also be relevant in mitigating the amount of water that percolated downward into the soil column after rain events. Each of the distributions extend further with higher-value outliers, however these were truncated during the statistic collection process in order to emphasize the shape

of the more common lower-end values. Notice that all of these processes are plotted on a log axis in the figure for visual clarity, which belies the fact that these are extremely skewed distributions. Similar to snow as outlined in the previous subsection, although they are important processes within the model, the fluxes and precipitation – and especially their upper extremes – are ultimately rare in the context of the full dataset, which poses a challenge for statistical optimization techniques like deep learning.

3.1.5 Soil Moisture Distribution and Metrics

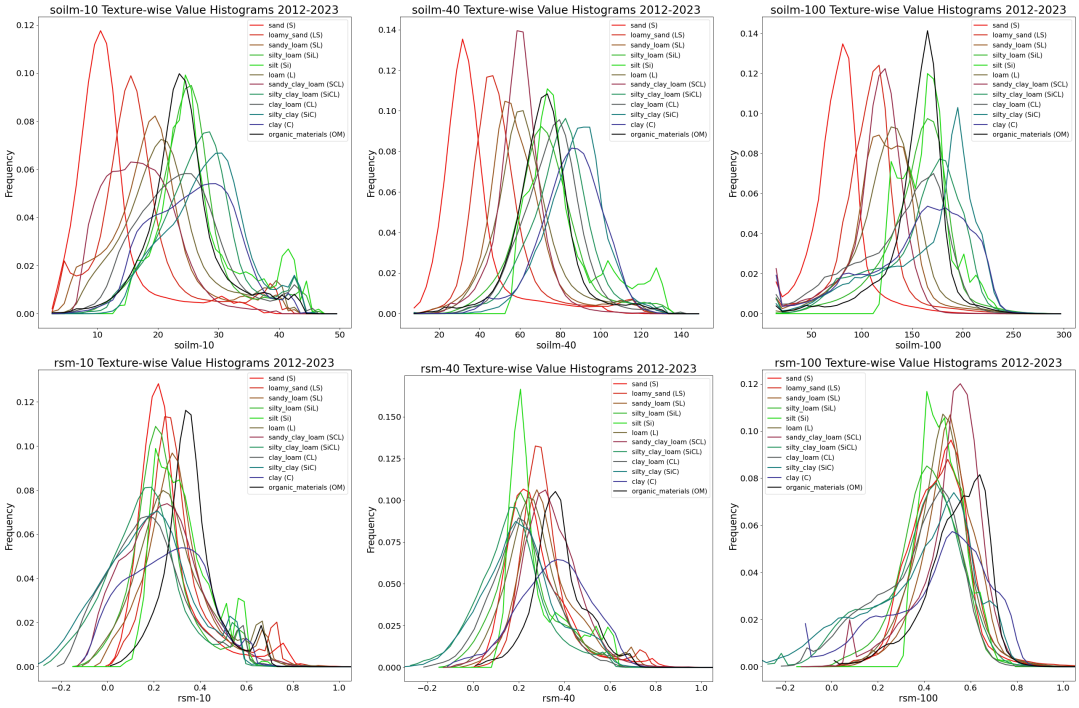

Figure 3.11: Distributions of relative soil moisture (top) compared to those of soil moisture area density (bottom) at the first three depth levels, separated by soil texture category. Red, green, and blue components of line colors correspond to the sand, silt, and clay composition of the soil textures, respectively.

The differences in conductivity, matric potential, and other hydraulic properties among soil textures causes their distributions to occupy distinct value ranges, and their dynamics to vary at rates that are characteristic to each specific class. As the top row of Figure 3.12 indicates, the distributions of soil moisture area density (in $kg\,m^{-2}$) tends to stratify such that coarser textures (sandy blends) have generally lower moisture content per unit volume, and vice-versa with silt and clay dominant soil textures. Although there is some regional influence, this is mainly owed to the faster percolation rates and lower porosity of the coarser soil.

Since the loss function calculations are directly dependent on the magnitude of soil moisture, we hypothesized that significant differences in these value ranges could diminish the ANNs' ability to learn general solutions that apply to all soil textures. For example, since clay has a slow conductivity and infiltration rate and thus typically smaller magnitudes of increment change, loss calculations based on the increment change would be de-emphasized compared to sand. For this reason, we adopt the relative soil moisture as a physically-interpretable metric for normalizing soil textures to values that occupy roughly the same scale.

$$\mathrm{RSM} = \frac{\frac{\mathrm{SOILM}}{d\,\rho_w} - \theta_{wp}}{\theta_s - \theta_{wp}} \tag{3.1}$$

Relative soil moisture (RSM) linearly scales the water content such that each texture's wilting point is at zero, and saturation point corresponds to one. In Equation 3.1, SOILM is the area density, ρ_w is the density of water, d is the depth of the soil layer, θ_s is the saturation point of a particular soil texture as a ratio of

the total volume, and θ_{wp} is the wilting point. In this manner, the RSM can never exceed 1, but its lower bound is defined in terms of the hydraulic suction head needed to uptake further water. As such, very dry soils can have RSM values below zero. The layerwise comparisons of SOILM to RSM in Figure 3.11 make clear how RSM normalization aligns the individual texture distributions, and the added benefit of uniting the separate soil layers to a similar range of state values rather than using mass quantities that scale with their unique depths.

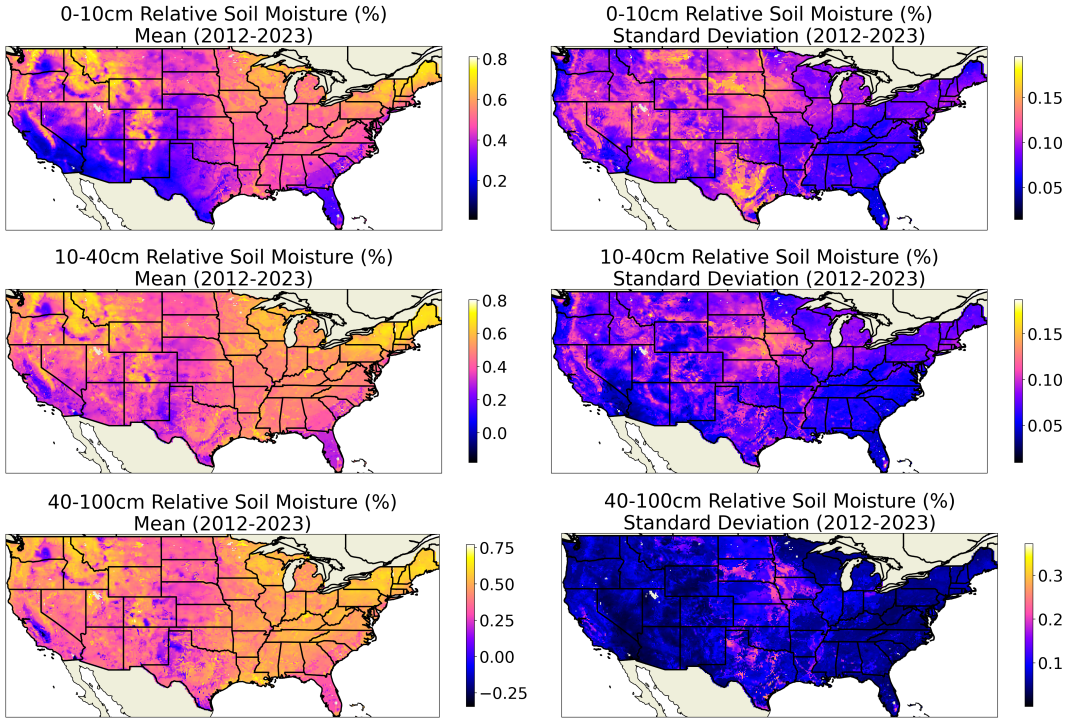

Figure 3.12: Gridded mean and standard deviation of relative soil moisture (2012-2023)

The spatial variation of soil moisture is depicted in Figure 3.11, which emphasizes several important regional distinctions. The mountainous regions of

the West remain rather saturated for much of the year – especially in the upper soil layers – owing to their high precipitation rates, the consistent presence of a snow pack, and limited drainage due to frozen soil. Sandy soils tend to have a lower average RSM than other soil types in a particular region due to their high hydraulic conductivity speedy drainage rates. The standard deviations were calculated using the overall per-pixel mean value; as such, clay-dominant soils generally have a higher standard deviation than other textures because their lower conductivity means they take more time to equilibrate after a rain event.

3.2 Model Architectures

The models that we tested fall into three broad architectural categories summarized in the background: fully-connected neural networks (FNNs), naïve RNNs, and LSTMs. In this subsection, we will elaborate on the particular implementation of these models for the unique problem structure of this project.

The basic fully-connected neural network variants we tested are structurally the most similar to Noah-LSM in the sense that the only information passed between time steps are the accumulated magnitudes of soil moisture at each layer given the prediction of the previous layer. As Figure 3.13 demonstrates, each timestep uses L ANN layers $\mathbf{A}^(1)$-$\mathbf{A}^(L)$ to transform the previous state, current forcing, and static parameter inputs into a prediction for the increment change in state. Only one initial state value $\vec{s}_{-1}$ is needed to run the model since it doesn't require a multi-step initialization window. The most performant

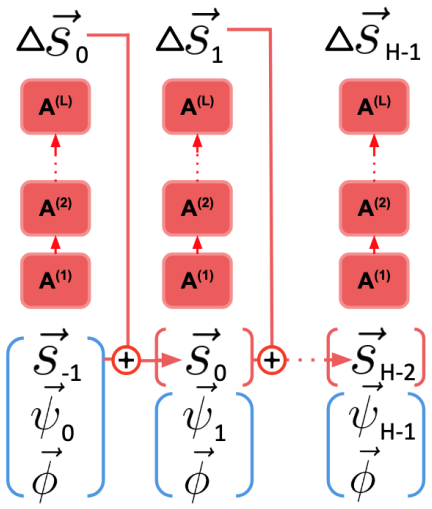

Figure 3.13: Schematic diagram of a multi-layered self-cycling fully connected neural network (FNN).

instance of this architecture will serve as a good baseline for understanding the benefit of introducing more structural complexity in the subsequent model types.

The naïve RNN and LSTM are distinct from the FNN in that they provide a hidden vector based on past timesteps to supplement the input at each subsequent timestep. In order to initialize the hidden vectors provided to first prediction timestep ($\vec{h}_{init}^{(l)}$ for $l \in [1, L]$) with values providing context about the recent history of soil states, we add a spin-up window ($\mathbf{G}^{(l)}$) that observes the W timesteps prior to the first prediction and produces a vector for each layer of the output sequence. This principle is diagrammed in Figure 3.14, and applies to both the naïve RNN and LSTM. In contrast to the prediction horizon sequence, the spin-up window explicitly receives surface states as inputs, and hidden states of each layer at the final timestep are the only values captured and passed along.

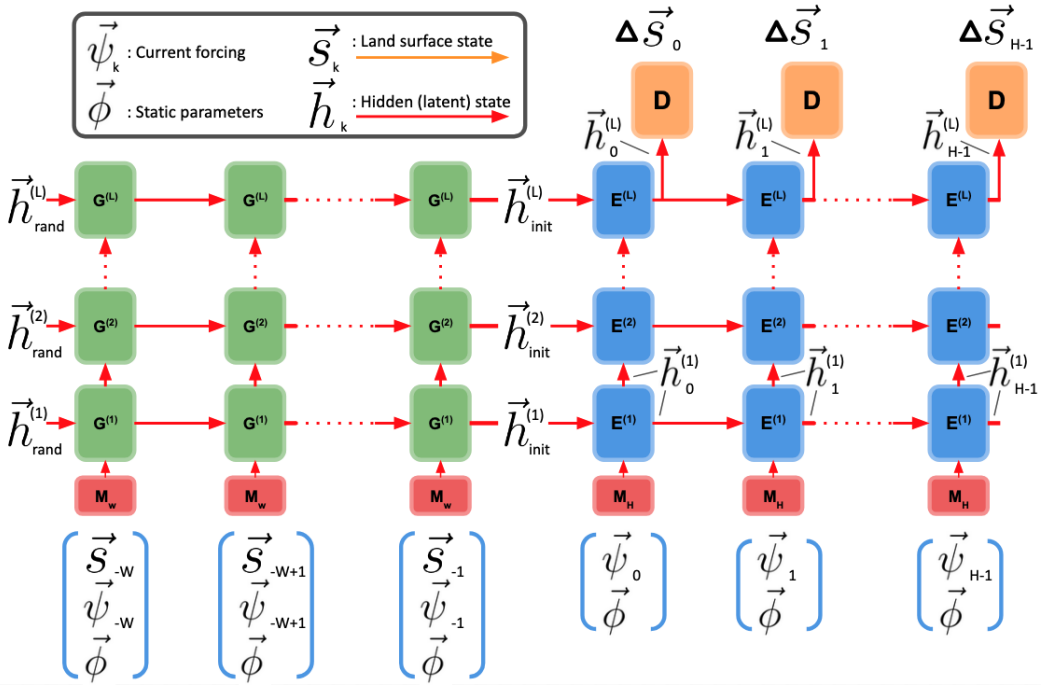

Figure 3.14: Sequence-to-sequence RNN architecture with initial projection layers **M**, spin-up window cells **G** for initializing first-step weights, prediction horizon cells **E**, and fully-connected decoder layer **D**.

For both of these reasons, the spin-up window sequence's weights are not shared with the forecast horizon sequence.

The next architecture we explored was the naïve RNN, which has multiple layers of cells that each produce an output based on 3 sets of weights applied to a latent vector from the previous timestep and an input vector from the layer below. The LSTM is similar except that each cell contains 4 sets of weights with outputs that are algebraically combined to produce the output rather than operated upon by a matrix transformation. For further details, refer back to Figure 2.4 and Equation 2.2.

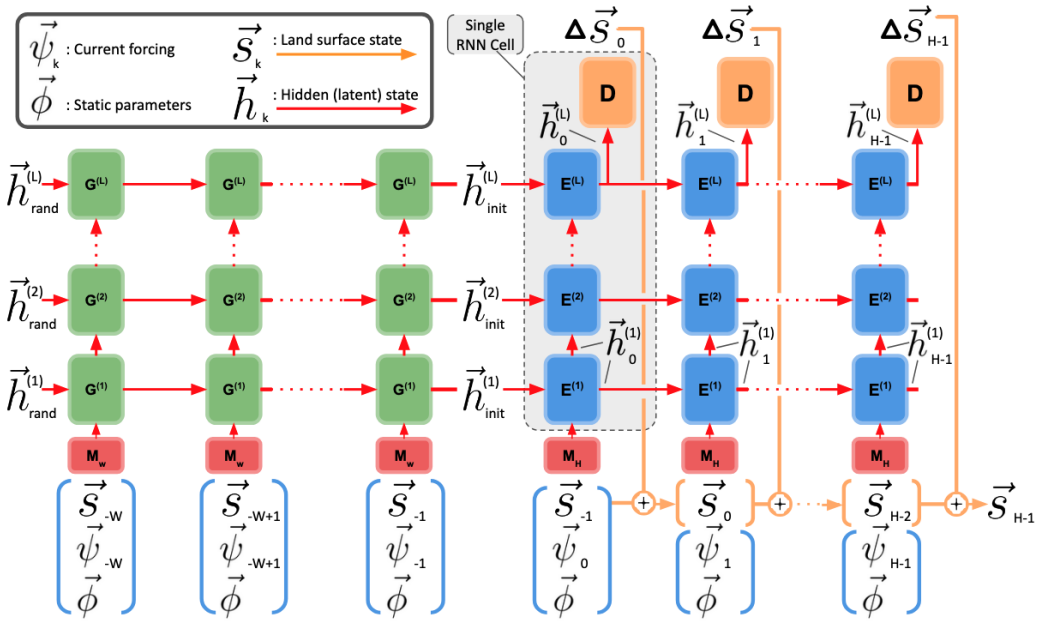

Figure 3.15: Sequence-to-sequence RNN with explicit output state accumulation.

When the inference algorithm is executed for a multi-layer RNN, every timestep of each layer is evaluated before the hidden states are passed along to a higher-level layer; there is no mechanic for information from the higher layers of previous timesteps to be passed along to the lower layers of subsequent steps. In contrast, the governing equations of Noah-LSM – including the Richardson equation, plant parameterization, and bare-surface evaporation – depend very strongly on the magnitude of soil moisture present at any given timestep. As such, we hypothesized that providing an explicit estimate of the soil moisture state as an input at each step would improve prediction skill. As Figure 3.15 demonstrates, we implemented this by encapsulating all the layers as a single RNN cell and adding the predicted increment change to the previous state after

each step. We will refer to naïve RNN and LSTM variants employing this strategy as "accumulators", abbreviated as AccRNN and AccLSTM.

All of the models presented here will predict the increment change in state rather than the state magnitude, consistent with the numerical model's estimation of the time derivative as in Equation 2.1. Early exploratory testing revealed that models which directly predict the soil state produced very erratic and physically unreasonable results, while predicting increment change led to smoother and more responsive estimates. We suspect that this is because the change in state correlates far more strongly with the current atmospheric forcings than the state magnitude, as many different environmental circumstances could lead to a particular soil state magnitude, but conditions like precipitation, humidity, and temperature have a direct influence on the immediate change in state.

Notice that unlike the numerical model described by Equation 2.1, which contains a variable increment change in time Δt, none of the neural network architectures here have a facility for determining the temporal resolution of predictions they generate. Instead, the coarseness of the timestep is implicit based on the training data, and cannot be changed after training; as such, the coarseness is an independent variable of the training configuration. Unless otherwise specified, the models presented here are all trained at the native 1 hour resolution of the forcing dataset (although the computational timestep of Noah-LSM is 15 minutes), however some models trained at 3 hour or 6 hour resolution showed very promising results. A more thorough analysis of the effect of courser time resolutions on prediction accuracy and model efficiency is left for future work.

3.3 Training Paradigm

One of the main challenges of developing ANNs is contending with the overwhelming number of parameters that can influence model behavior. The modeler must decide on a learning rate schedule, loss function characteristics, the number of layers within the model, the number of trainable weights within each layer, and many other parameters for which there are few reliable standards or heuristics that apply generally to different problem types. To address this, we initially trained 10-25 ANNs within each category while varying multiple parameters between generations based on intuition, and in order to capture a wide breadth of configurations. After these exploratory training runs, we selected the best models from each category and re-trained them while only perturbing one parameter at a time in both directions. For instance, increasing and decreasing the number of layers by one. Finally, the models in each architectural category having the best bulk statistical performance were selected for further more detailed evaluation.

All of the model training was executed on a CPU cluster with 32 cores, and training was allowed to proceed until the prediction skill stopped increasing for a validation dataset (withheld from training) for 48 epochs. In this context, the epoch refers to the number of updates to the model weights between performance reports, which we set to 256. In this environment, training would usually conclude in less than 36 hours. In order to logistically facilitate working with this large number of models, we developed a simple but robust system for storing, de-

ploying, and provisioning data for models based on a set of rigorous configuration standards.

We use functional generators to load, scale, and restructure data on-demand from the source `timegrid` files during training. This involves extracting sparsely-sampled chunks from multiple files, separating the data into arrays for the spin-up window, prediction horizon inputs, target values, and static data ordered as a set of uniform-length sequences, and linearly normalizing data values so that they approximate a unit gaussian. As mentioned in the beginning of this chapter, it is important to globally shuffle the data used for training in order to avoid over-fitting to a data distribution that only describes a subset of the full domain, so the sequence samples from separate chunks are randomly interleaved. The data generators also accommodate "derived" features that are calculated on-demand as a function of the stored features rather than being stored in parallel, which enables target values like RSM or inputs like wind magnitude and relative humidity to be used during training and evaluation without occupying extra disc space and forcing costly re-generation of the `timegrid` files.

One of the most important considerations for ANN training setup is the learning rate schedule, which governs the sensitivity of the model's trainable weights to the prediction cost given new data during training. If the learning rate is too high, training may not converge on an optimal solution that relies on fine-grained changes in weights, while if the learning rate is too low, the learning process may take too much time, and could prematurely halt after getting stuck in a local minimum of the loss environment which would require a larger update in

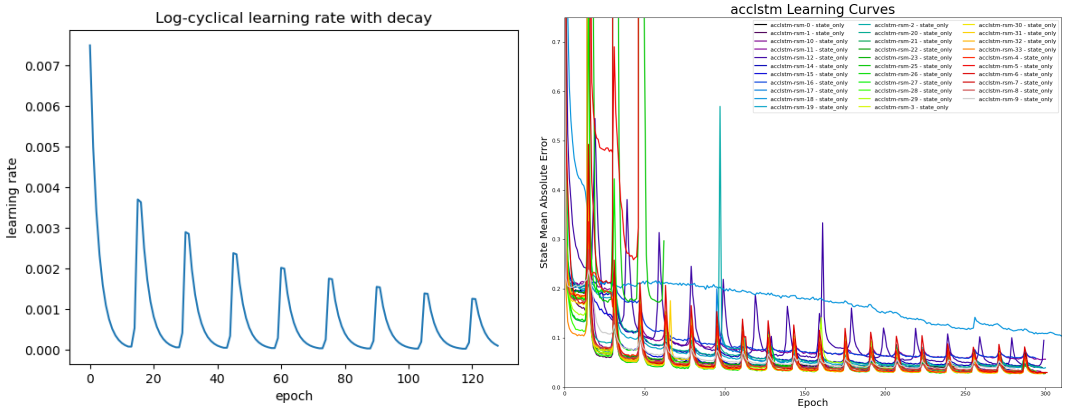

Figure 3.16: Learning rate schedule and subsequent learning curves for AccLSTM architectures.

model weights to resolve (Russell and Norvig, 2020). Practitioners generally start with a high learning rate early in training, and schedule it to gradually decrease as training proceeds (Ren et al., 2024), however a separate promising strategy is to cycle between lower and higher learning rates many times (Smith, 2017). We mainly utilized a combination of these two methods as shown in the left panel of Figure 3.16, which introduces new tunable parameters including the period of cycles, magnitude of decay, and initial minimum and maximum learning rate.

The loss function is even more critical in determining how the model ultimately behaves, given that it defines the standard for quantifying the skill of predictions. Many applications will simply employ mean absolute error (MAE) or mean squared error (MSE) as the loss metric without further changes, however as long as the functions remains differentiable, modifications can emphasize or de-emphasize certain outputs, balance multiple goals, or assign importance to individual samples based on properties of their data. As such, we tested the

effects of manipulating three characteristics of the loss function. First, the values normalized to a unit gaussian in the data generator include the inputs and the target soil moisture state magnitude, however the increment change in soil moisture has its own characteristic distribution that is unique to each soil layer. To address this, we implemented an optional second normalization of increment change values that is only applied within the loss function. In effect, this serves as a coefficient that scales the ratio of the loss associated with each of the output values. Second, the models should prioritize making the locally optimal prediction at each timestep, but they should also be incentivized to recover from error in previous step predictions and produce the globally optimal sequence. To incorporate both of these goals, we introduced a tunable ratio that combines the step-wise error with the integrated state error at each time step. Finally, accurate predictions at some timesteps is considerably more important for producing skillful sequences than many others. For example, a summertime thunderstorm that produces 12cm of precipitation in an hour followed by a two hour surface runoff event will often have a larger influence on the soil column than a subsequent week of drydown near equilibrium. The loss contribution of circumstances like these can be emphasized by adding an additional cost proportional to the true magnitude of change in each layer.

$$
\begin{aligned}
L(X',\, Y) &= \rho\, P(X', \mathrm{diff}(Y)) + (1-\rho)\, Q(\mathrm{acc}(X') + Y_{j=0}, Y) \\
P(X',\, Y') &= \frac{1}{S} \sum_{i=0}^{N} \sum_{j=0}^{S} (1 + \gamma \left|Y'_{i,j}\right|) \frac{\left|X'_{i,j} - Y'_{i,j}\right|}{\lambda_i} \\
Q(X,\, Y) &= \frac{1}{S} \sum_{i=0}^{N} \sum_{j=0}^{S} \frac{|X_{i,j} - Y_{i,j+1}|}{\lambda_i}
\end{aligned}
\tag{3.2}
$$

Equation 3.2 formalizes the loss function definition for a single pair of prediction (X) and target (Y) values with N output values (soil layers) and S sequence steps. The arguments to the top-level function L are the N predicted increment changes X', and the $L + 1$ true state values (which includes the land surface state immediately prior to the first prediction step). The "diff" function is the discrete forward difference along the sequence axis converting the true states to the true increment changes, and the "acc" function is the cumulative sum of the predicted changes along the sequence axis. Then $P(X', Y')$ handles the loss contribution of the increment changes, and $Q(X, Y)$ represents that of the state magnitude, which are balanced by the increment loss ratio ρ. The λ_i parameter serves as the pre-determined normalization coefficient for soil layer i, and the magnitude bias parameter γ scales the sensitivity of the loss in proportion with the absolute value of the increment change.

3.4 Evaluation Metrics

$$r(x,y) = \frac{\sum_{i=0}^{N}(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=0}^{N}(y_i - \bar{y})^2}} \tag{3.3}$$

$$MAE(x,y) = \frac{1}{N}\sum_{i=0}^{N}|y_i - x_i| \tag{3.4}$$

$$RMSE(x,y) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_i - x_i)^2} \tag{3.5}$$

The metrics we will use to evaluate the ANN's performance include the pearson correlation coefficient, mean absolute and mean squared error, error bias, as well as entropy-based metrics including information loss and fractional information. The pearson coefficient is a metric of the strength and direction of the linear relationship between variables, in this case between the target and predicted soil moisture content, or hourly increment change in soil moisture content. A value of would 1 indicate a perfect positive linear relationship, while zero suggests no correlation, and -1 a perfect inverse linear correlation. We calculate the pearson coefficient in accordance with Equation 3.3 independently for each 2-week prediction sequence, then average the results across pixels and initialization times to get the values presented in the following chapter.

Unlike the pearson coefficient, which is scale-invariant, the mean absolute and root mean squared error (RMSE) metrics present results using the same units as the subject data. MAE is straightforwardly the expected error for a randomly

selected target and prediction data pair (Equation 3.4, while RMSE utilizes the average squared difference in data pairs, which makes the metric more sensitive to high-error outliers. Since both of these metrics are differentiable, they can both be used as loss functions while training ANNs. In Equation 3.2 – and by default for most of the models presented here – we used MAE as the essential form of the loss function, however we will test models trained to optimize the squared error as well.

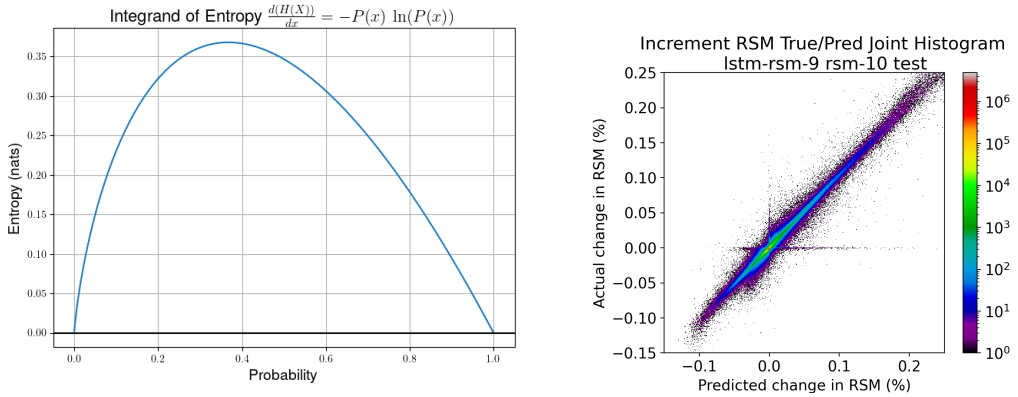

Figure 3.17: Entropy value curve for a single possible state of a system with respect to the probability of that state being occupied, and an example of a joint histogram validation curve from which the total entropy is calculated.

In addition to the metrics above, which all in some way characterize the deviation of the data pairs from a linear relationship, we will evaluate the ANNs in terms of the uncertainty contribution (also known as information loss) and fractional information according to the methodology that (Nearing et al., 2016) uses to the benchmark the uncertainty of Noah-LSM in terms of contributions from the input forcings, model parameters, and model structure. Unlike the correlation coefficient, MAE, and MSE, the fractional information and uncertainty contribu-

tion are used to characterize the extent to which using the ANN rather than the source model introduces ambiguity in the distribution of results by creating a sub-distribution of predictions corresponding to each target state.

$$H(z) = -\sum_{i=1}^{B} \frac{n_i^{(z)}}{N} \ln\left(\frac{n_i^{(z)}}{N}\right) \tag{3.6}$$

$$I(y^A, y^M) = H(y^M) + H(y^A) - H((y^A, y^M)) \tag{3.7}$$

The following discussion of the principles and mathematical formalisms from information theory are based on Chapter 2 of Elements of Information Theory (Schilling, 2005). The information metrics we will use are functions of the Shannon entropy $H(z)$ of the distributions of increment change in RSM, and are estimated in terms of the discrete joint histograms of the targeted and predicted values, such as the one pictured on the right of Figure 3.17. In Equation 3.6, $n^{(z)}$ represents a histogram of arbitrary variable z with B bins that count the integer number of occurrences of data within the value range described by each bin. Each of the bin counts are divided by the total number of observations N to yield the probability of a randomly selected sample occupying that position within the histogram. The integrand function plotted on the left-hand side of Figure 3.17 is applied to the probability associated with each of the bins in order to calculate their individual information contribution to the system, and the total entropy of the distribution is estimated as the sum of the contributions of the bins.

To better understand the properties of a distribution that entropy characterizes, consider a hypothetical distribution that describes a large number of low or zero-probability configurations, and a small number of high-probability configurations. As Figure 3.17 suggests, both the low and high-probability configurations correspond to a relatively small entropy contribution. In contrast, a different hypothetical distribution having probabilities that are spread uniformly across its possible configurations will have a relatively large entropy contribution from each of them, so the sum total entropy of the latter distribution will be higher than the former. In this sense, as described by (Schilling, 2005): "the entropy of a random variable is a measure of the uncertainty of the random variable; it is a measure of the amount of information required on the average to describe the random variable".

Entropy is a fundamental property of a distribution with units of information depending on the base of the logarithmic function used to calculate the integrand. In our case, we use the natural logarithm, which corresponds to "natural units" or nats. One of the main advantages of entropy is its chain rule property, which facilitates an additive relationship between the entropy of independent variables and that of their joint distribution. As a consequence, we can straightforwardly calculate the mutual information $I(y^A, y^M)$ of two variables, which is a commutative metric of the reduction in uncertainty of one given knowledge of the other. In Equation 3.7, y^A refers to the histogram of increment RSM values predicted by the ANN, y^M to the histogram of Noah-LSM values, and (y^A, y^M) to their joint histogram.

$$FI(y^A, y^M) = \frac{I(y^A, y^M)}{H((y^A, y^M))} \tag{3.8}$$

$$U(y^A; y^M) = H(y^A) - I(y^A, y^M) \tag{3.9}$$

In practice, we will use the entropy and mutual information values to calculate the derived metrics of fractional information ($FI(y^A, y^M)$, Equation 3.8) and uncertainty contribution ($U(y^A; y^M)$, Equation 3.9) to characterize the efficiency of the ANNs at emulating the information characteristics of Noah-LSM. Fractional information normalizes the mutual information by the total entropy of the joint distribution, which returns a more familiar ratio value since $H((y^A, y^M))$ is a strict upper bound on mutual information. The uncertainty contribution (or information loss) remains in the unit of nats, and explicitly articulates the additional entropy (that is, uncertainty) supplied by employing the ANN rather than the source model.

Chapter 4. Results

4.1 Exploratory Model Runs

As mentioned in the previous chapter, one of the main challenges encountered while training neural networks is the overwhelming number of independent variables that can affect model performance. These so-called "hyperparameters" may have complex interdependence, and there are few general best-practices. In order to identify a reasonable baseline for the performance of each of the model architectures, we started by training a wide variety of permutations on model properties including the number of layers, nodes per layer, activation function, and prediction coarseness, as well as training properties including loss function characteristics, batch normalization, and weight dropout rate. At this stage, developing a rigorous understanding of the effects of changes in model and training setup was not a priority; instead, we modified multiple hyperparameters simultaneously between training iterations with the goal of establishing rough rules of thumb and a sense for the approximate size of models needed for this task.

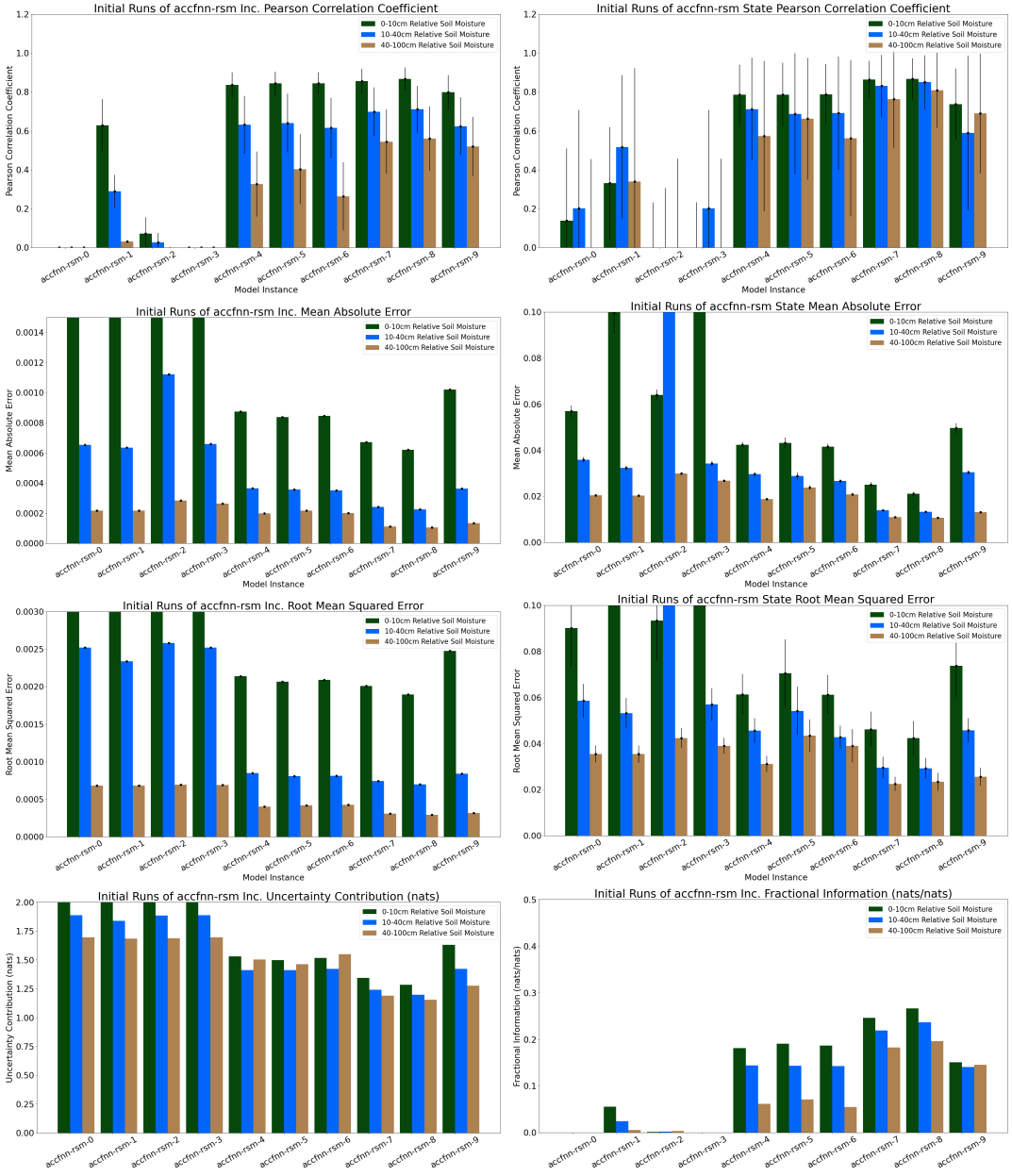

Figure 4.1: Bulk metrics for initial FNN training runs

Name	Desc	Weights	State MAE	State CC	Info Loss	Frac. Info
accfnn-rsm-0	First FNN	14203	0.057	0.139	2.196	0.000
			0.036	0.201	1.889	0.000
			0.020	-0.215	1.694	0.000
accfnn-rsm-1	Same setup as accmn-rsm-1 but only a FNN	14203	0.100	0.331	2.065	0.055
			0.032	0.517	1.840	0.024
			0.020	0.340	1.685	0.005
accfnn-rsm-2	Much wider and deeper, MSE loss, more dropout	85947	0.064	-0.139	2.193	0.001
			0.102	-0.201	1.885	0.002
			0.030	-0.214	1.688	0.003
accfnn-rsm-3	Same as fnn-2 but ignoring constant targets	85947	0.113	-0.139	2.196	0.000
			0.034	0.201	1.889	0.000
			0.027	-0.215	1.694	0.000
accfnn-rsm-4	Same as rsm-2 but no dropout, higher increment magnitude bias	85947	0.042	0.786	1.530	0.181
			0.030	0.711	1.412	0.144
			0.019	0.573	1.502	0.061
accfnn-rsm-5	Same as rsm-4 but ignoring constant targets	85947	0.043	0.785	1.499	0.191
			0.029	0.688	1.411	0.143
			0.024	0.663	1.463	0.071
accfnn-rsm-6	Same as rsm-4 but narrower and deeper	30843	0.042	0.788	1.516	0.187
			0.027	0.692	1.423	0.142
			0.021	0.563	1.550	0.055
accfnn-rsm-7	Same as rsm-6 but mae loss	30843	0.025	0.864	1.343	0.246
			0.014	0.831	1.242	0.219
			0.011	0.764	1.189	0.183
accfnn-rsm-8	Same as rsm-7 but ignoring constant targets	30843	0.021	0.867	1.283	0.266
			0.013	0.850	1.197	0.236
			0.011	0.809	1.154	0.196
accfnn-rsm-9	fnn 5 but actually using increment norm coefficients	85947	0.050	0.738	1.632	0.151
			0.030	0.589	1.423	0.140
			0.013	0.689	1.277	0.145

Table 4.1: Initial fully-connected neural network properties and bulk statistics.

All of the bulk statistics in this section are reported in terms of relative soil moisture. The bar charts of correlation coefficient, mean absolute error, and mean squared error separately display the error in hourly increment change and integrated soil state, while the entropy-based metrics (uncertainty contribution and fractional information) are calculated only for the increment change. Tables include metrics for each of the depth levels from top to bottom: 0-10cm, 40-100cm, and 100-200cm.

The validation loss of each of the ANN variants generally stopped decreasing after about 18 hours of training on a CPU, though the training time of individual models unsurprisingly depended most strongly on the size of the model and the learning rate parameters. The models shown here have a number of trainable parameters on the order of 100,000. The best-performing instances of the simplest architecture variant (accfnn-rsm-8) had only about 31,000 parameters and used mean absolute error as the base loss function. The FNN instances struggled to converge any time dropout was used during training, and the predictive skill of best model seemed to improve in the lower two layers in response to a loss function manipulation that ignored prediction cost associated with timesteps where the true magnitude of change in soil moisture state was close to zero. Furthermore, the best FNN did not consider error in state within the loss function (increment loss ratio $\rho = 1$), but was trained with a considerable increment magnitude bias of $\gamma = 60$. The network consisted of 8 fully-connected layers each with 64 nodes.

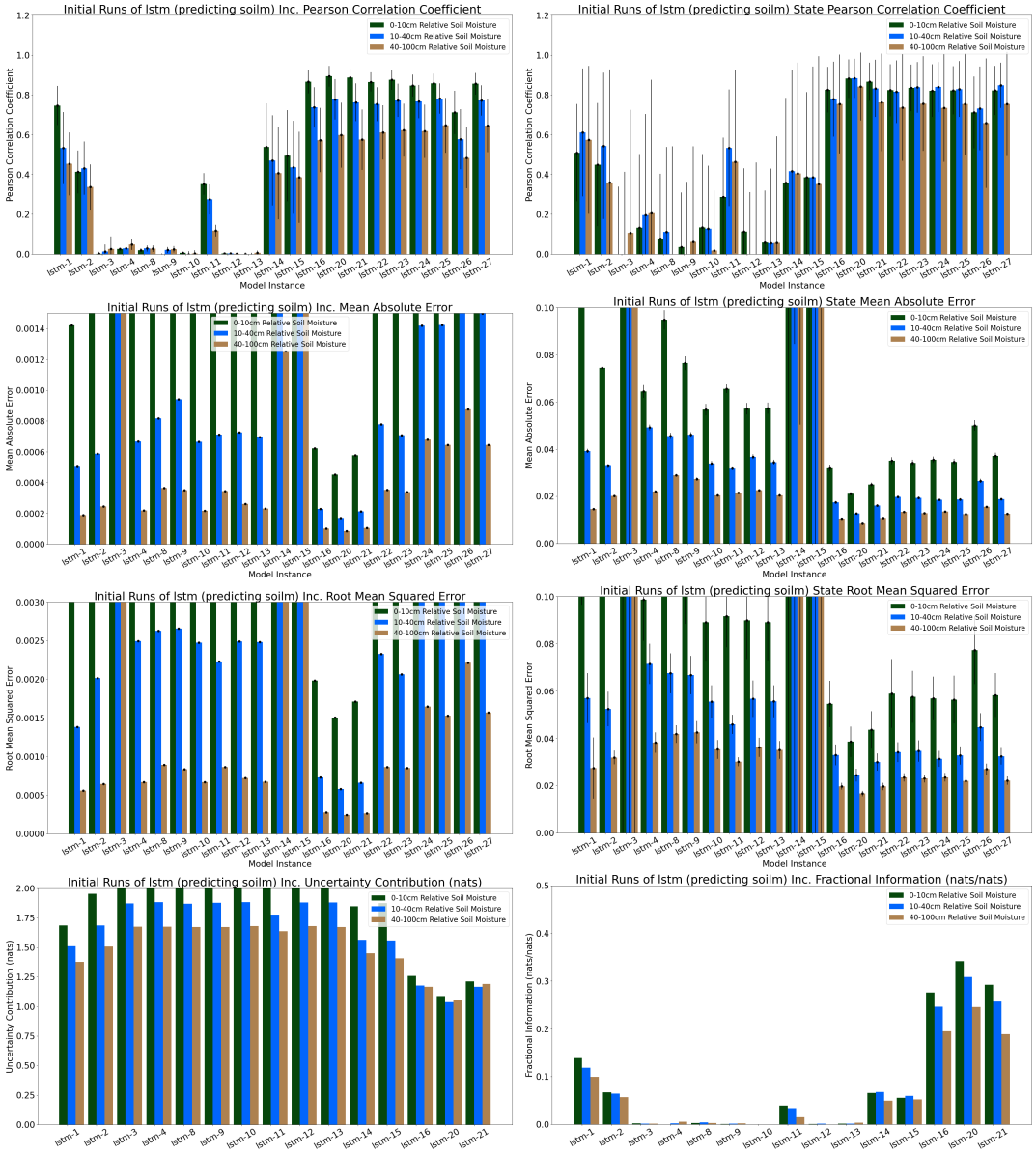

Figure 4.2: Bulk metrics for initial LSTM-VSM training runs

Name	Desc	Weights	State MAE	State CC	Info Loss	Frac. Info
lstm-1	4 layers, 64 wide. No batchnorm.	254397	0.107	0.509	1.686	0.138
			0.039	0.612	1.509	0.118
			0.015	0.574	1.378	0.099
lstm-2	batchnorm, higher learning rate	254397	0.074	0.450	1.954	0.067
			0.033	0.544	1.684	0.065
			0.020	0.360	1.507	0.057
lstm-3	Slower LR ; much 24 wide and 6 layers deep	64189	2.877	-0.029	2.183	0.002
			3.153	-0.084	1.873	0.002
			0.783	0.107	1.675	0.002
lstm-4	4 layers, 96 wide. faster LR, small loss from state magnitude	629021	0.064	0.132	2.198	0.000
			0.049	0.196	1.882	0.002
			0.022	0.205	1.675	0.006
lstm-8	4 layers, 64 wide. bigger batch size	288829	0.095	0.077	2.188	0.003
			0.046	0.111	1.870	0.004
			0.029	-0.034	1.671	0.003
lstm-9	4 layers, 16 wide. bigger batch size	20029	0.076	0.035	2.195	0.000
			0.046	-0.064	1.877	0.001
			0.027	0.061	1.673	0.002
lstm-10	6 layers, 24 wide; cyclical learning rate ; higher dropout	65445	0.057	0.133	2.197	0.000
			0.034	0.127	1.882	0.000
			0.020	0.016	1.680	0.000
lstm-11	2 layers, 256 wide; slower learning rate	1929597	0.066	0.286	2.058	0.039
			0.032	0.534	1.779	0.034
			0.021	0.463	1.636	0.015
lstm-12	4 layers, 32 wide; no batchnorm	74813	0.057	0.112	2.195	0.000
			0.037	-0.193	1.880	0.001
			0.023	-0.193	1.679	0.000
lstm-13	same as lstm-12 but increment ratio .8	74813	0.057	0.058	2.192	0.002
			0.034	0.055	1.879	0.001
			0.020	0.055	1.671	0.003

Table 4.2: Initial LSTM-VSM properties and bulk statistics (1).

Name	Desc	Weights	State MAE	State CC	Info Loss	Frac. Info
lstm-14	Heavy increment error ; decaying log-cyclical learning rate; batch norm	74813	0.726	0.357	1.847	0.065
			0.224	0.416	1.565	0.067
			0.147	0.405	1.451	0.050
lstm-15	No dropout, some increment magnitude bias	74813	1.556	0.385	1.874	0.055
			1.481	0.385	1.558	0.060
			0.831	0.352	1.407	0.052
lstm-16	Same as lstm-15 but smaller learning rate	74813	0.032	0.826	1.258	0.276
			0.017	0.779	1.176	0.246
			0.010	0.754	1.166	0.195
lstm-20	Stronger dependence on state, some increment magnitude bias, dropout	77117	0.021	0.882	1.088	0.342
			0.013	0.884	1.036	0.309
			0.008	0.842	1.058	0.246
lstm-21	lstm-20 but more dropout, more increment magnitude bias, some state loss	77117	0.025	0.866	1.213	0.292
			0.016	0.833	1.166	0.257
			0.011	0.762	1.191	0.189
lstm-22	4 layers, 32 wide, faster learning rate	80221	0.035	0.823		
			0.020	0.816		
			0.013	0.737		
lstm-23	much smaller encoder, heavier on increment, but less magnitude bias	43597	0.034	0.835		
			0.019	0.838		
			0.013	0.756		
lstm-24	64 nodes wide, 5 layers deep; weaker dependence on increment, strong increment magnitude bias	173165	0.035	0.821		
			0.018	0.840		
			0.013	0.735		
lstm-25	Same as lstm-24, but 128 nodes wide	364029	0.035	0.823		
			0.019	0.829		
			0.012	0.754		
lstm-26	Same as lstm-25, 6 layers deep	762413	0.050	0.712		
			0.026	0.732		
			0.015	0.658		
lstm-27	Same as lstm-24, more increment error	173165	0.037	0.822		
			0.019	0.848		
			0.012	0.754		

Table 4.3: Initial LSTM-VSM properties and bulk statistics (2).

Next, we present a variety of LSTM instances that predict the increment change in volumetric soil moisture (VSM; $\frac{kg}{m^2}$), which we will refer to as the LSTM-VSM group. The results reported here have been converted to relative soil moisture after-the-fact for consistency with the other architectures. In practice, these were the first generations of models we tested; those which appear in Table **??** were trained using a consistent learning rate and converged rapidly to fairly poor results. Even relatively large models with a variety of hyperparameter configurations didn't achieve a correlation coefficient higher than .65 for any of the layers. Introducing the log-cyclical learning rate schedule prolonged training and, combined with the loss function modifications, resulted in considerably better-performing LSTMs. Given its apparent success, we continued to use the log-cyclical learning rate strategy for the remainder of the models, changing only the rate of decay and the initial minima and maxima of the oscillations. The best LSTM-VSM models we trained had 77,117 trainable weights, a magnitude bias parameter of $\gamma = 10$, increment loss ratio $\rho = .999$, and did not use increment normalization within the loss function. Unlike the FNN architectures, the best LSTM-VSM variants tended to include a small weight dropout of 5% during training, however without further analysis it is difficult to draw conclusions on the actual impact of any of these changes in isolation.

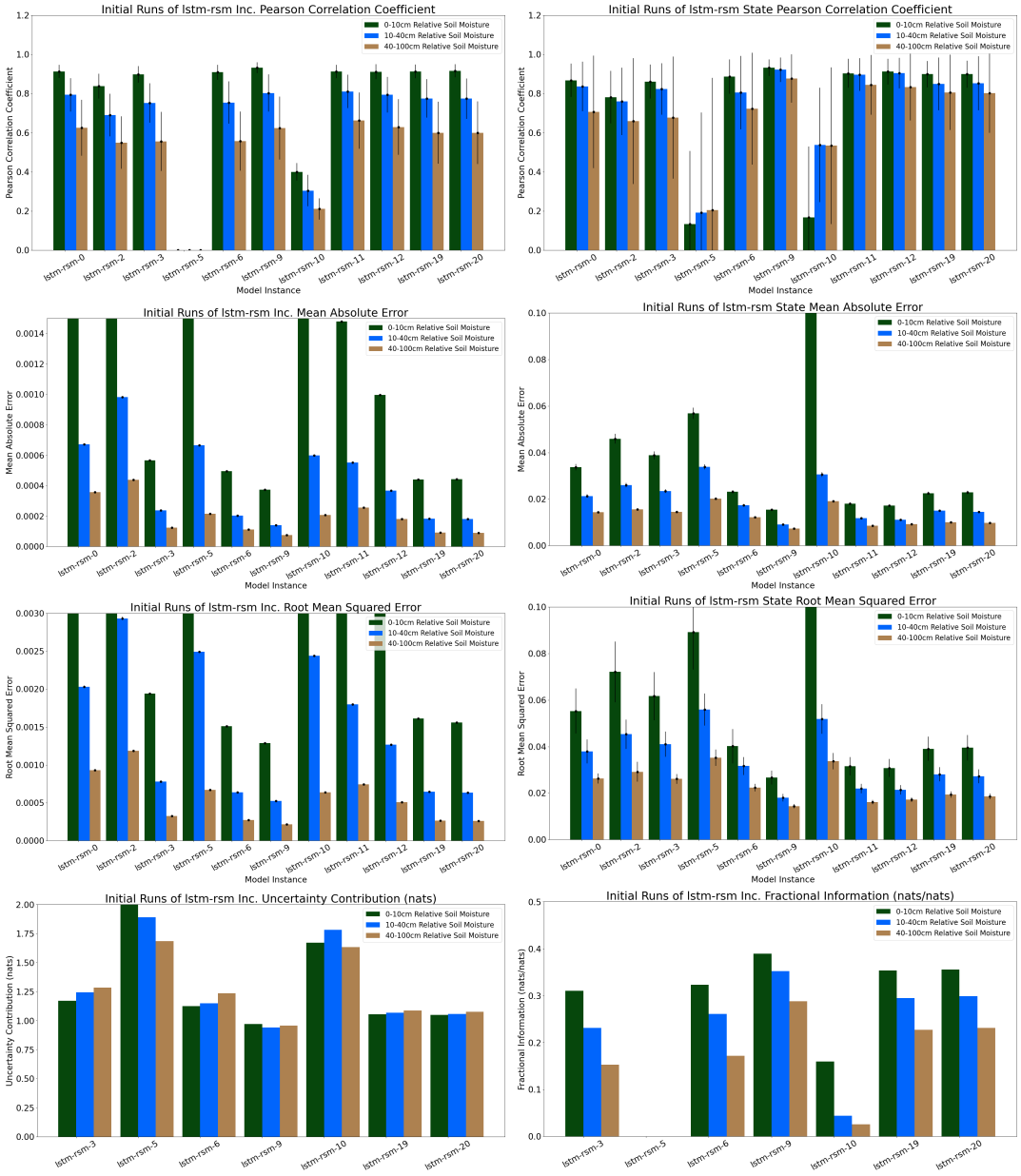

Figure 4.3: Bulk metrics for initial LSTM-RSM (relative soil moisture predictor) training runs

Name	Desc	Weights	State MAE	State CC	Info Loss	Frac. Info
lstm-rsm-0	Small magnitude bias. low dropout. small batch size. 64-wide 4-layer.	179483	0.034 0.021 0.014	0.868 0.837 0.707		
lstm-rsm-2	same as rsm-1, but some state error influence, magnitude bias 30	179483	0.046 0.026 0.016	0.782 0.761 0.660		
lstm-rsm-3	Low dropout, 4 deep, 64 wide decoder, and increment-only loss	176379	0.039 0.023 0.015	0.861 0.823 0.677	1.169 1.243 1.285	0.310 0.231 0.152
lstm-rsm-5	Same as lstm-rsm-3, but 10% state influence in loss function	176379	0.057 0.034 0.020	0.133 0.191 0.205	2.206 1.890 1.686	0.000 0.000 0.000
lstm-rsm-6	Small 3-layer predictor with 100 increment magnitude bias	48667	0.023 0.017 0.012	0.886 0.805 0.723	1.124 1.149 1.235	0.323 0.261 0.171
lstm-rsm-9	32 nodes wide, 4 layers deep, 10 increment magnitude bias	48667	0.015 0.009 0.007	0.932 0.922 0.877	0.970 0.941 0.958	0.389 0.352 0.288
lstm-rsm-10	256 nodes wide 5-layer model	2614907	0.138 0.031 0.019	0.168 0.537 0.534	1.671 1.783 1.632	0.159 0.044 0.025
lstm-rsm-12	4 layers deep, 32 nodes wide, steep learning rate decay	78651	0.017 0.011 0.009	0.912 0.904 0.833		
lstm-rsm-19	Single layer 256 nodes wide	725627	0.022 0.015 0.010	0.899 0.850 0.806	1.055 1.067 1.086	0.353 0.295 0.227
lstm-rsm-20	Same as rsm-19 except some influence of state error	725627	0.023 0.014 0.010	0.899 0.852 0.803	1.048 1.058 1.075	0.356 0.299 0.231

Table 4.4: Initial RSM-normalized LSTM properties and bulk statistics.

The next group of models we trained will be referred to as LSTM-RSM models, which have the same structure as the previous set, but target the increment change in RSM rather than volumetric soil moisture. Like the LSTM-VSM models, these seemed to show diminishing returns with model sizes beyond 100,000 weights. The best model we identified has only 48,667 trainable weights, and is 4 layers in depth. Curiously, while the loss function manipulations appeared to have a positive impact on the FNN and LSTM-VSM variants, LSTM-RSM models generally underperformed when a higher magnitude bias and a stronger contribution from the state were used. The best model had a relatively small increment magnitude bias $\gamma = 10$, and no contribution from state error ($\rho = 1$).

4.2 Best Models' Bulk Statistics Comparison

As Table **??** and Figure 4.4 display, the best LSTM-RSM model performed better than the other categories at all of the depth levels, and for each of the evaluation metrics, followed by the LSTM-VSM, and finally the FNN architecture. During evaluation on the full 2018-2023 test dataset, we recorded the execution speed of each of the models as they were applied to multiple subdomains with varying numbers of valid pixels, then fitted a linear regression to the relationship between subdomain size and the time it took to generate a 2-week prediction for each pixel. Evaluation was done with a single CPU thread on a shared high-performance computing cluster. The results in Table **??** indicate that the initial spin-up time of each model is between 3 and 4 seconds, and execution time for each subsequent pixel is between .1 and 1 millisecond, and is directly proportional

to the number of trainable weights in each of the models. When applied to the full domain of 50,875 pixels, the FNN was by far the fastest model at only around 9.5 seconds, while the LSTM variants clocked in at a bit over 40 seconds.

Name	Slope (ms)	Intercept ($\frac{s}{px}$)	R^2	Full Domain (s)
accfnn-rsm-8	.1262	3.088	.852	9.508
lstm-20	.8233	3.510	.984	45.397
lstm-rsm-9	.7544	3.246	.953	41.627

Table 4.5: Linear regression of execution speed for each of the best models.

Name	Weights	State MAE	State CC	Info Loss	Frac. Info
accfnn-rsm-8	30843	0.021	0.867	1.283	0.266
		0.013	0.850	1.197	0.236
		0.011	0.809	1.154	0.196
lstm-20	77117	0.021	0.882	1.088	0.342
		0.013	0.884	1.036	0.309
		0.008	0.842	1.058	0.246
lstm-rsm-9	48667	0.015	0.932	0.970	0.389
		0.009	0.922	0.941	0.352
		0.007	0.877	0.958	0.288

Table 4.6: Size and bulk statistic values of the best models from each category.

The highest error rates in terms of RSM for each of were associated with the 0-10cm surface layer, which is unsurprising considering that a larger number of processes within Noah-LSM affect the soil dynamics within the topmost layer, which is uniquely influenced by bare-surface evaporation and infiltration from rain and canopy percolation. Furthermore, since the soil layers increase in thickness within Noah-LSM from 10cm at the surface layer to 30cm and 60cm for the deeper layers, each kilogram of actual water mass corresponds to a smaller increment of RSM as it drains downward.

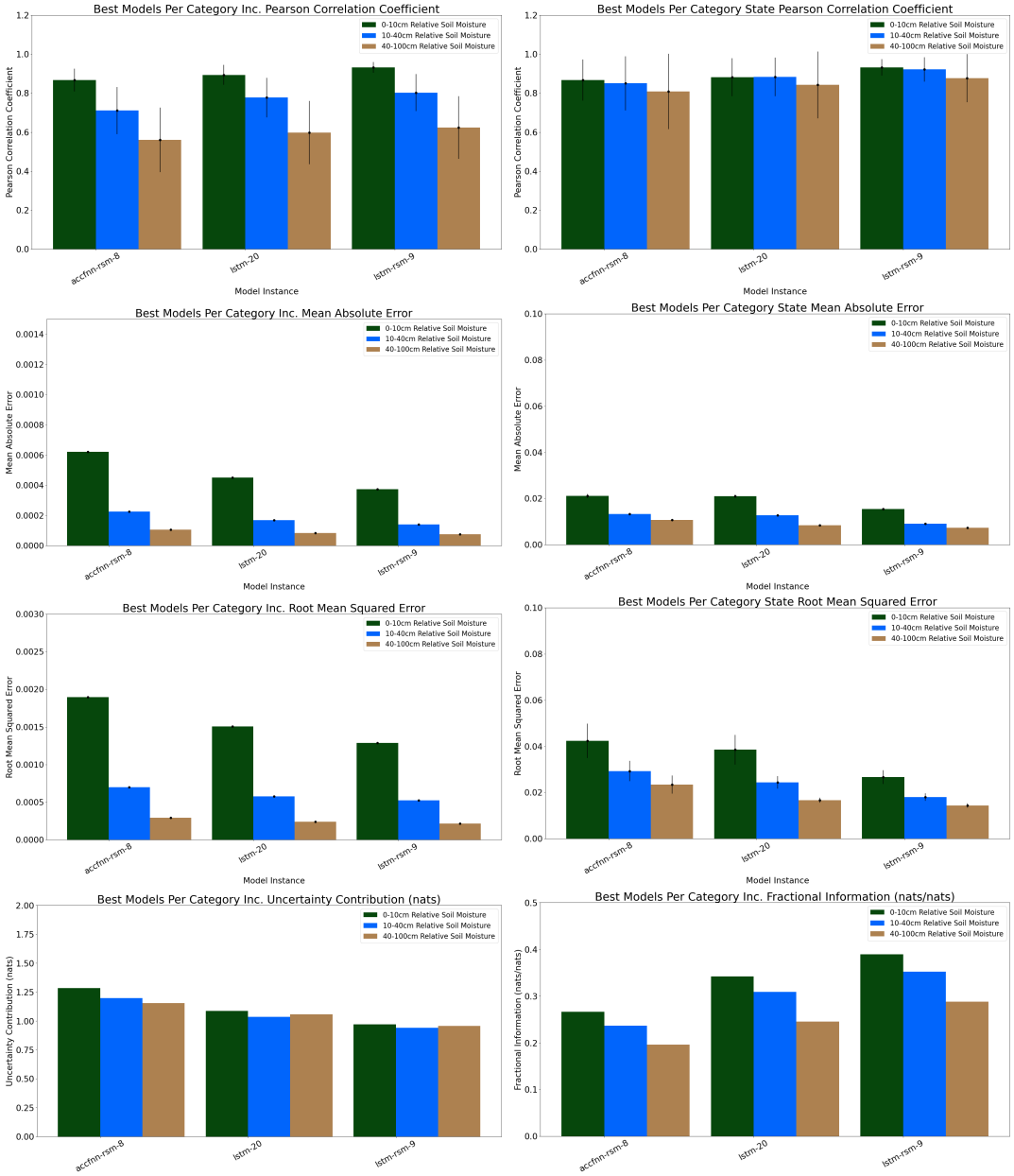

Figure 4.4: Bulk metrics comparing the best exploratory models from each category

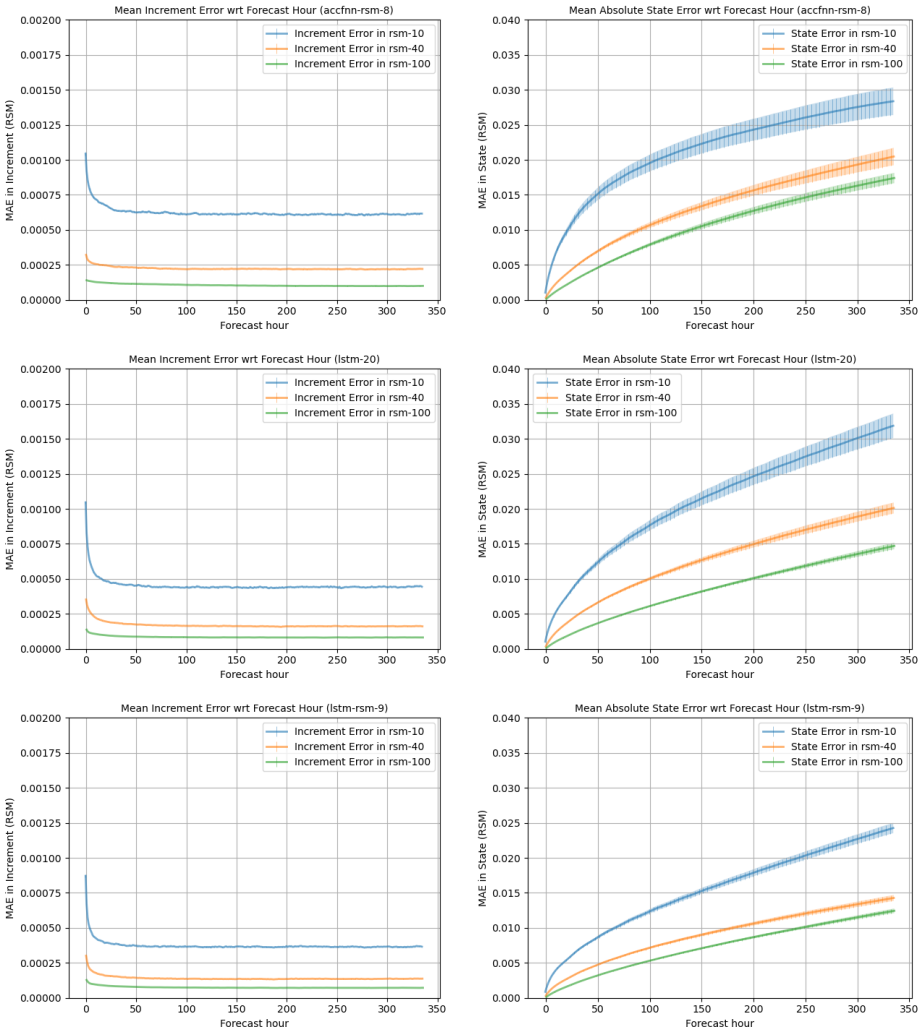

Figure 4.5: Mean absolute error of each model type with respect to forecast horizon increment (left) and state (right)

Figure 4.5 shows the mean absolute error in RSM for each of the best models with respect to the forecst horizon distance in hours from initialization time. One surprising result that appears universal among the models we trained is that the first few prediction steps consistently have the highest average error in the increment change. For the LSTMs, this could be an indication that the 24 hour spin-up encoder used to initialize the RNN latent vectors prior to the first prediction step have room for improvement, however the FNN has no such encoder. The cause of this spike in error is presently unclear, however in practice, this issue could be mitigated by initializing prediction a few hours prior to the actual initial unknown forecast step, and discarding the first troublesome predictions of increment change.

Figures 4.6, 4.7, and 4.8 show the overall spatial distribution of mean absolute error and model bias at the 0-10cm, 10-40cm, and 40-100cm depth levels, respectively. We will use the information that these graphics provide to identify areas of anomalous model behavior for further investigation in the following section.

The LSTM-RSM shows a widespread slight dry bias in the surface layer with the exception of urban areas, mountainous regions, and some sandy areas in Texas, New Mexico, and the Northern Midwest. The LSTM-VSM (lstm-20) largely follows suit, though it also showed regions of wet bias where the LSTM-RSM was more neutral or dry including silty regions corresponding to the Ouachita mountains, central Kentucky and the Cumberland plateu, the Northern Rockies in Idaho and Montana, and Florida. The FNN model had significantly

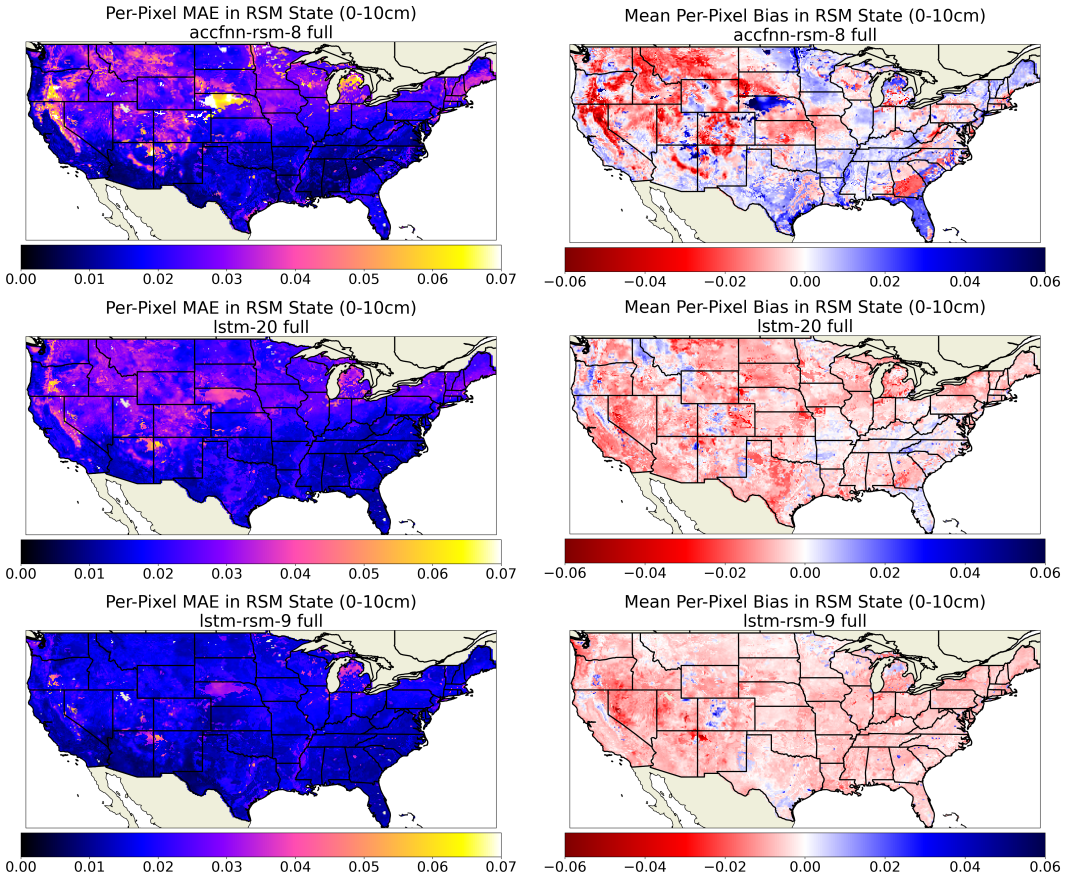

Figure 4.6: Gridded MAE and bias in state for each of the best models at the 0-10cm depth level, evaluated on full test set (2018-2023)

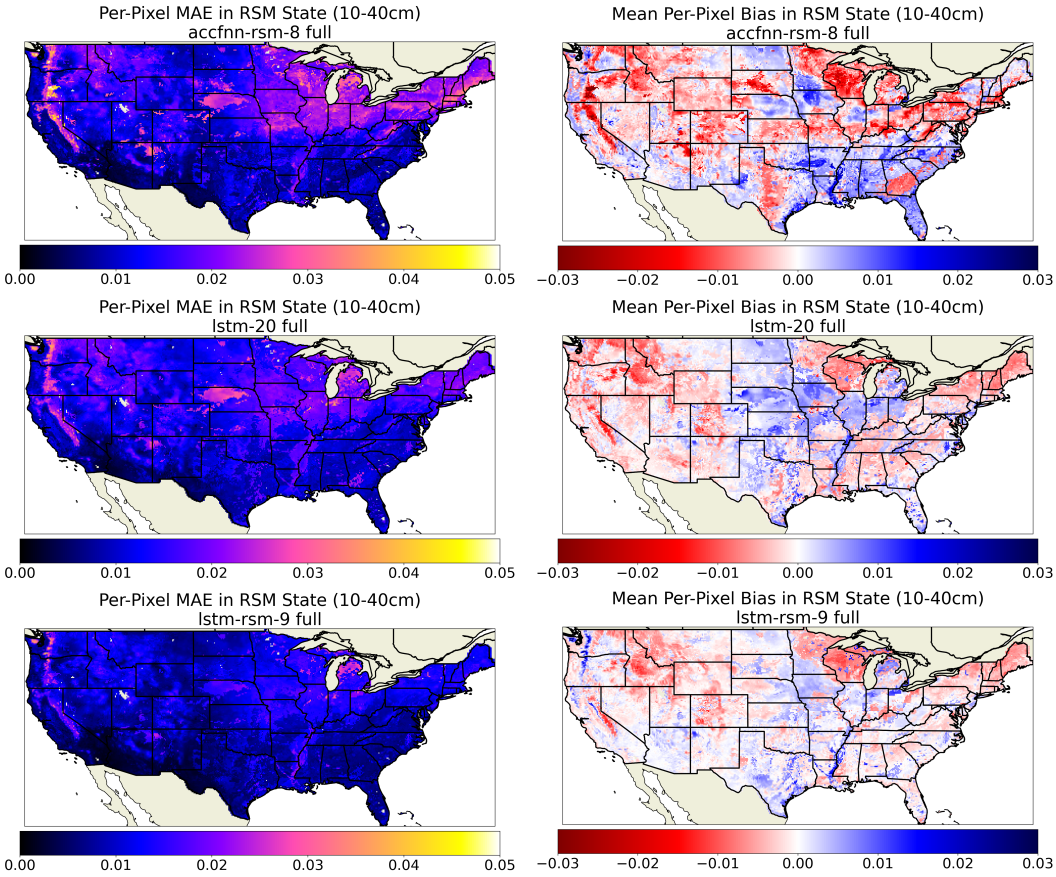

Figure 4.7: Gridded MAE and bias in state for each of the best models at the 10-40cm depth level, evaluated on full test set (2018-2023)

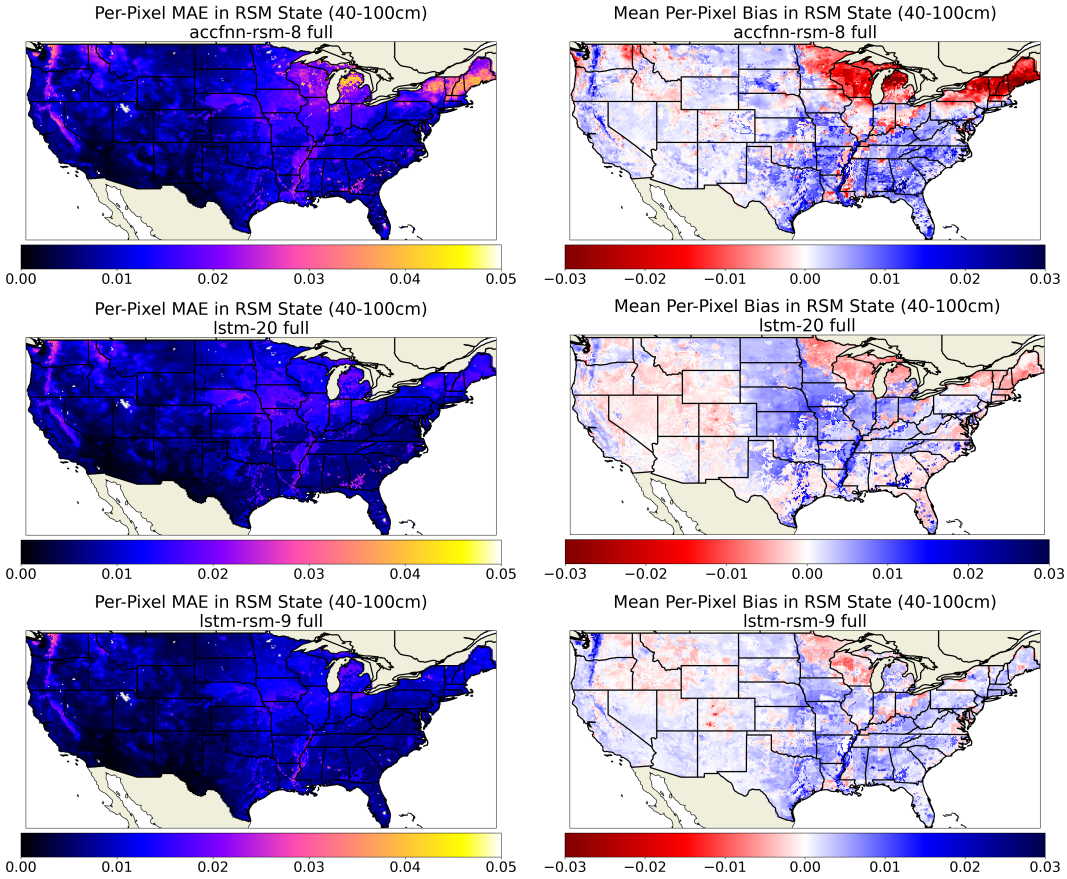

Figure 4.8: Gridded MAE and bias in state for each of the best models at the 40-100cm depth level, evaluated on full test set (2018-2023)

more of a wet bias in the surface layer for some regions, which were largely distinguished discretely by their soil texture boundaries.

All three models had more of a mixture of wet and dry biases in the 10-40cm layer. In the Midwest, Northern Wisconson and Minnesota were dry except for isolated sandy pixels, while the cropland in the Ohio and Missouri River basins to the South and West of that region were notably too wet: a pattern that was further emphasized in the 40-100cm layer. The wet bias associated with the cropland surface type was also apparent in Mississippi Alluvial Plains of Eastern Arkansas and the Missouri Bootheel, discrete regions of East Texas, the Florida Penninsula, and East-central South Carolina. Furthermore, each of the models also demonstrated a wet bias associated with clay-dominant pixels in the Centeral Texas and surrounding the Southern extent of the Mississippi River. In addition to the Northeast and Northern Midwest, the regions with the most significant dry bias in the lower two layers seemed to be associated with higher elevation, especially the Sierra Nevada mountains, Colorado Rockies, and Central Idaho, although the Cascades seem to be an exception with a stark wet bias for the LSTM-RSM model.

In the lower two layers, there is also an anomalous tiling artifact present in the model biases that is especially noticable in the Ohio river valley, Montana, and the central Great Plains. The emerging squares are 8 pixels in length and width, forming a 1° grid. This was initially perplexing since no similar pattern appears in the forcings, however close inspection of the mean Noah-LSM soil moisture content in 40-100cm layer of Figure 3.12 also reveals subtle edges that correspond

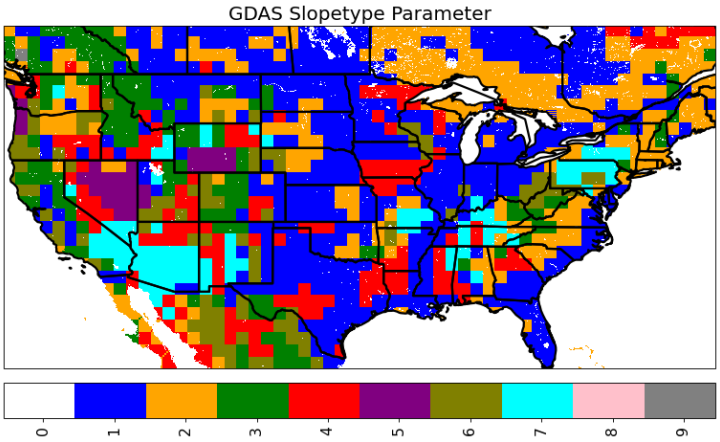

Figure 4.9: SLOPETYPE classes defining bottom-layer drainage.

to the bias tiles. Upon further investigation, we found that the square boundaries closely align with an input parameter in Noah-LSM called SLOPETYPE which controls the rate at which water drains out of the lowest layer of the model by applying one of 9 discrete classes (Mitchell, 2005), which are displayed by Figure 4.9. This parameter is distinct from the surface slope (which is defined in terms of degrees from vertical), and wasn't provided as an input to the ANNs during training, so the patterns in model bias are emergent from the unexplained variance introduced by leaving out these values.

Despite the inconsistencies between the models' biases, several regions stand out as especially troublesome in terms of all of the models' mean absolute error. Sandy areas in the Midwest, Nebraska, Oregon, and the Four Corners region of the Desert Southwest have high error rates in the first two layers. In the 40-100cm layer, clay pixels in Mississippi, Louisiana, Alabama, and North-

ern Ohio suffer, and the Sierra Nevada and Cascade mountains proved especially difficult for the models to characterize in the lower two soil layers.

4.3 Spatial, Temporal, and Situational Evaluation

From this point forward in our analysis of the results we will focus exclusively on the best overall model, "lstm-rsm-9", with the goal of better understanding its performance in a variety of regional and meteorological scenarios.

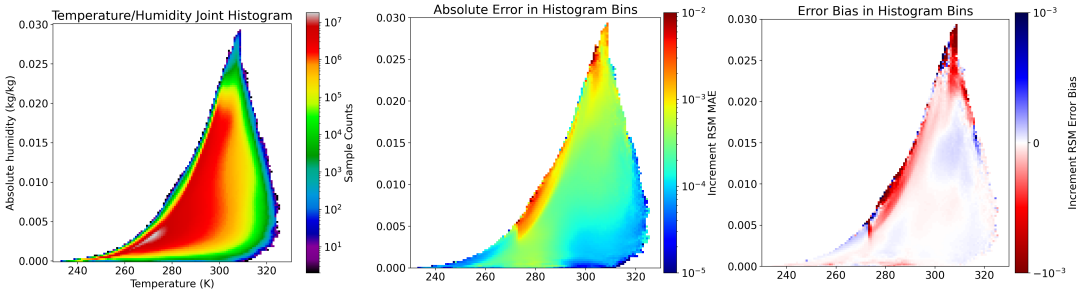

Figure 4.10: Joint histogram between temperature and absolute humidity (left) with increment mean absolute error (center) and bias (right) in corresponding bins (lstm-rsm-9; 2018-2023).

4.4 Parameter Variations

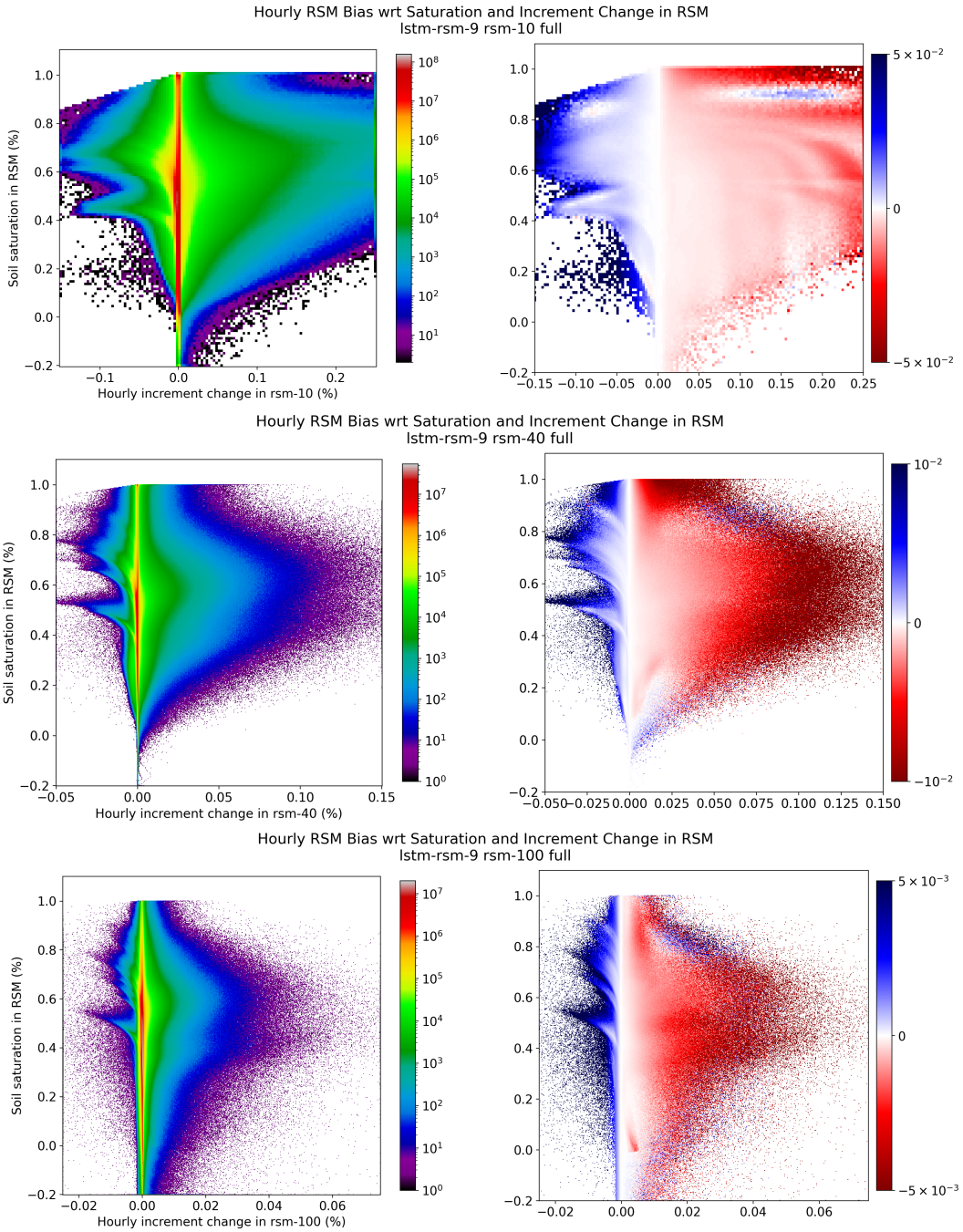

Figure 4.11: Joint histogram between target increment change in RSM and RSM magnitude (left) with increment bias (right) in corresponding bins at each depth level (lstm-rsm-9; 2018-2023).

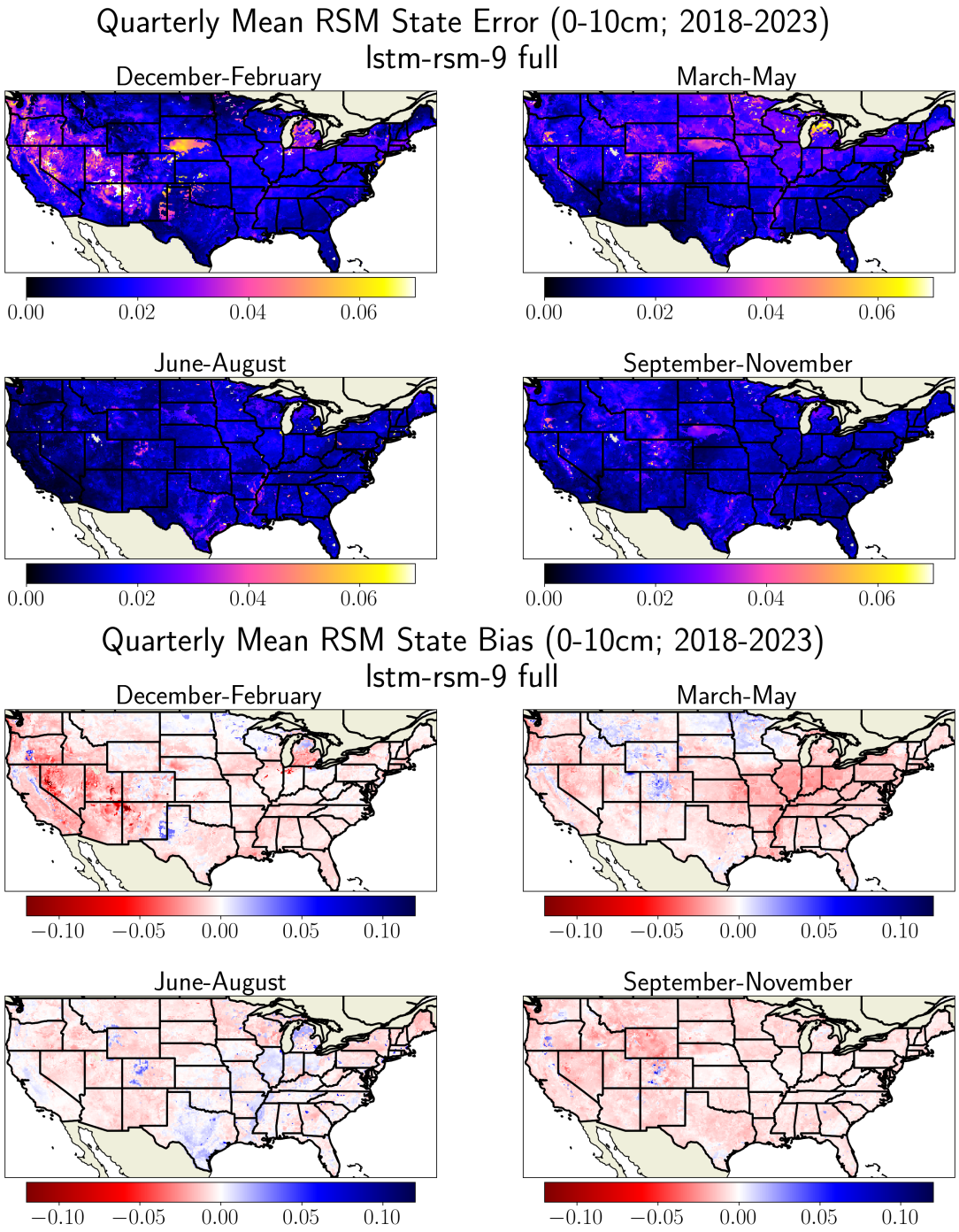

Figure 4.12: Quarterly mean absolute error (top) and bias (bottom) for the 0-10cm layer (lstm-rsm-9; 2018-2023).

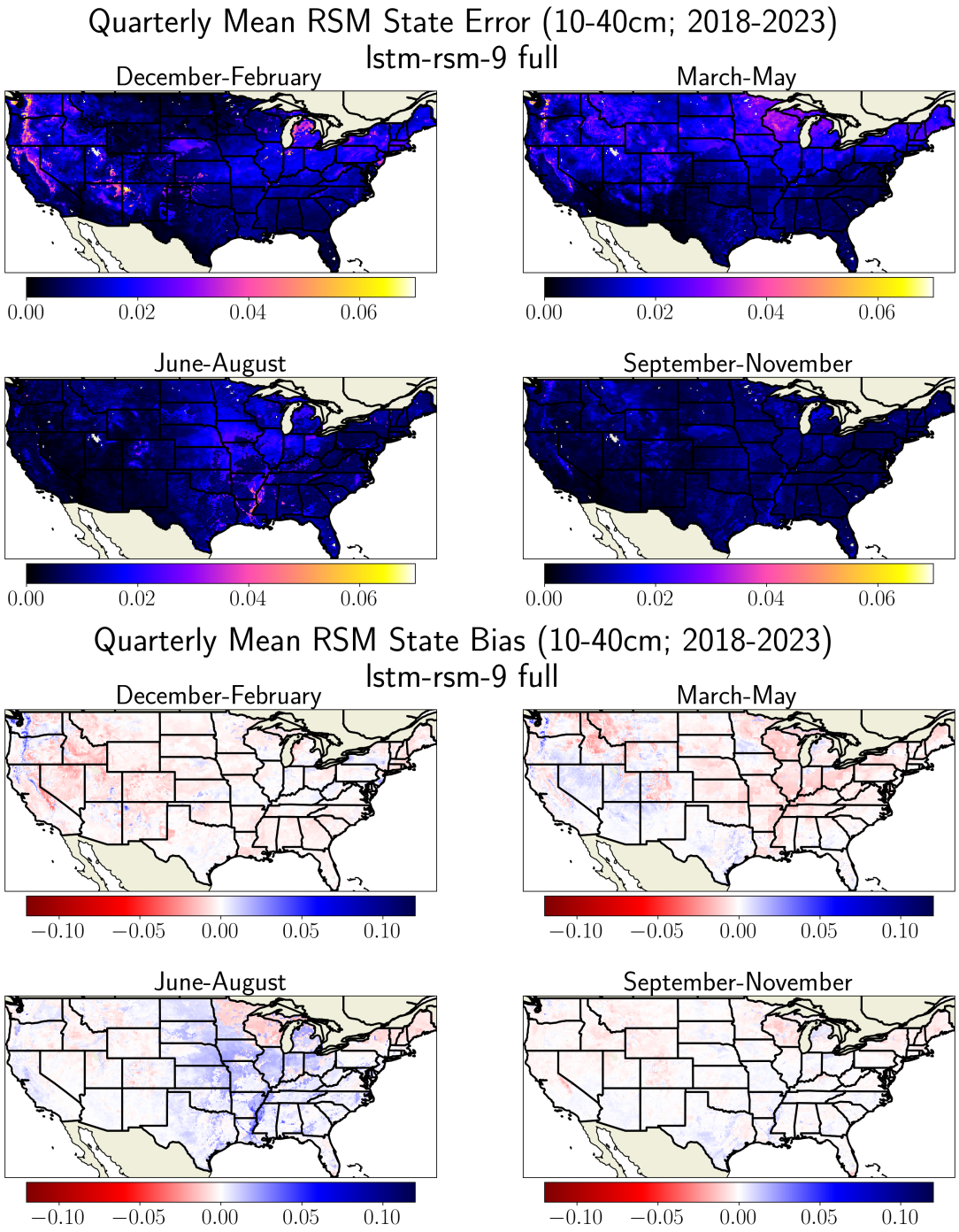

Figure 4.13: Quarterly mean absolute error (top) and bias (bottom) for the 10-40cm layer (lstm-rsm-9; 2018-2023).

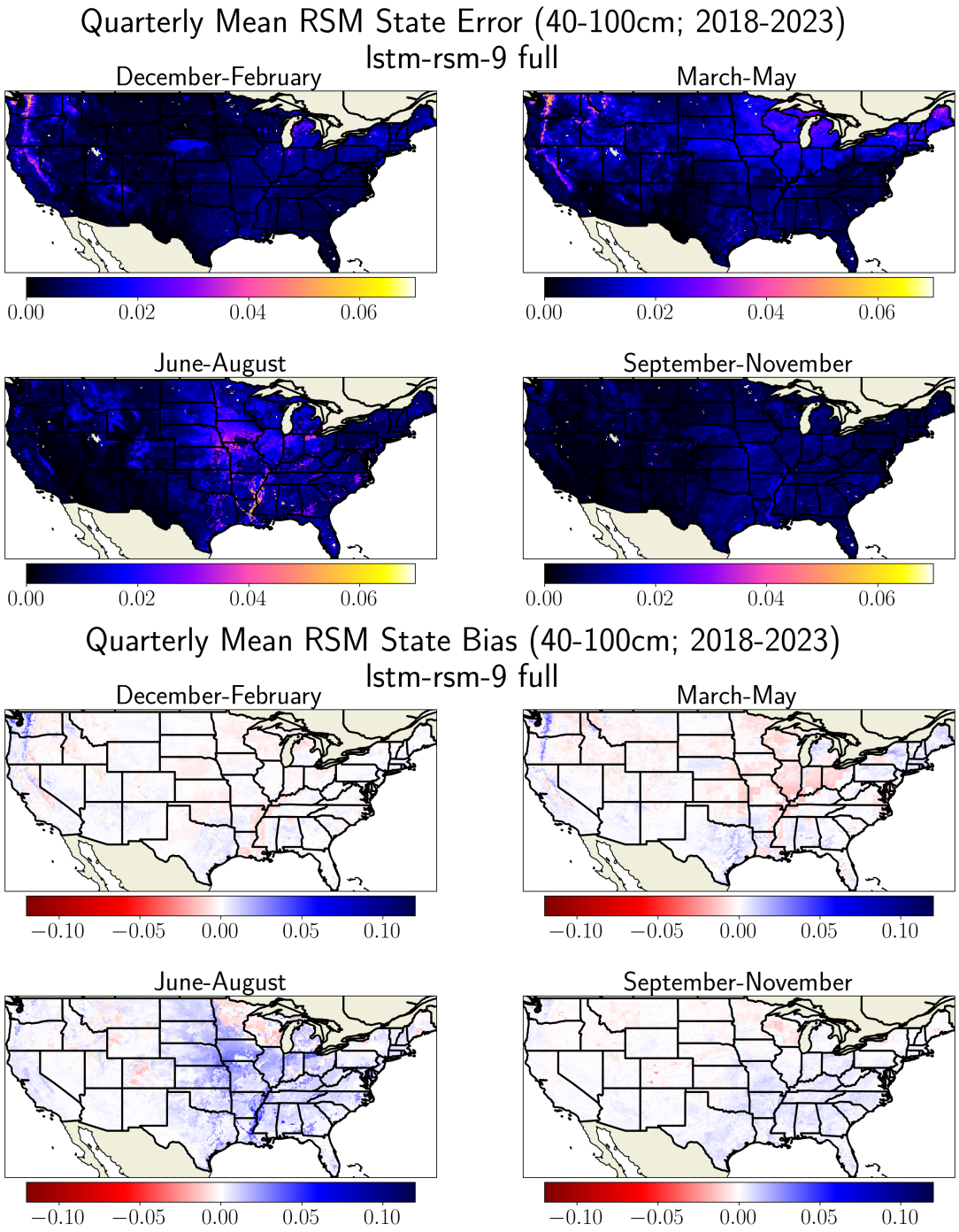

Figure 4.14: Quarterly mean absolute error (top) and bias (bottom) for the 40-100cm layer (lstm-rsm-9; 2018-2023).

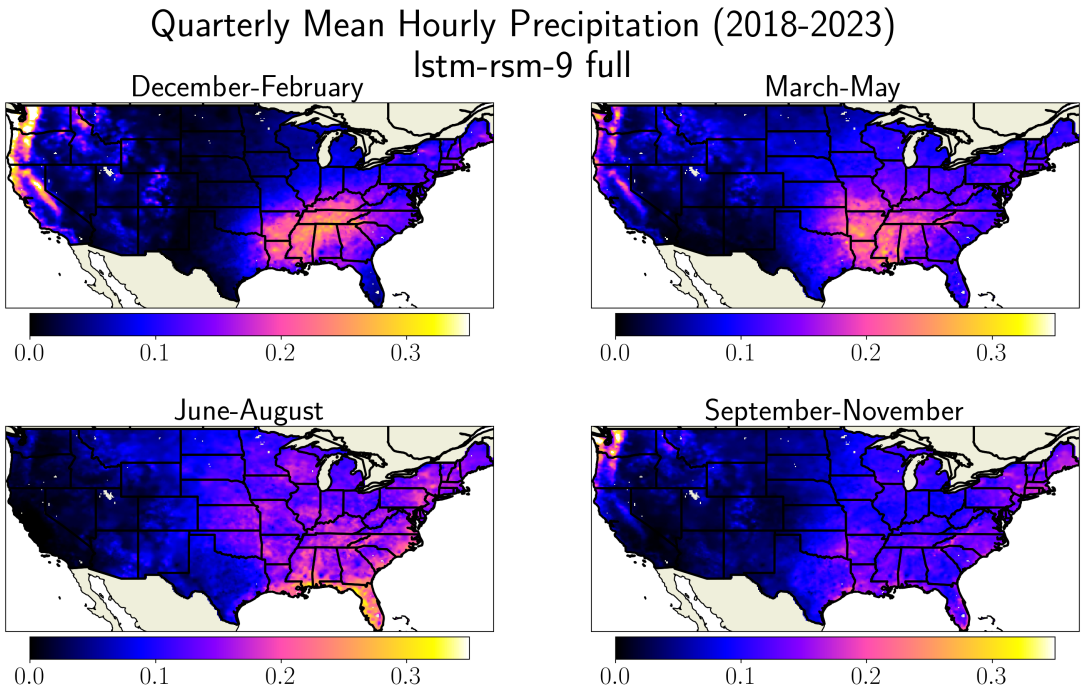

Figure 4.15: Quarterly mean precipitation distribution during the test period ($\frac{cm}{hr}$; 2018-2023).

Name	Feature Negated	State MAE	State CC	Info Loss	Frac. Info
lstm-rsm-9	Best LSTM-RSM	0.015 0.009 0.007	0.932 0.922 0.877	0.970 0.941 0.958	0.389 0.352 0.288
lstm-rsm-34	Leaf Area Index	0.013 0.008 0.007	0.942 0.915 0.888	0.927 0.916 0.946	0.406 0.364 0.294
lstm-rsm-35	Vegetation	0.016 0.009 0.008	0.931 0.918 0.855	1.000 0.954 1.003	0.372 0.346 0.268
lstm-rsm-36	Temperature	0.016 0.010 0.007	0.925 0.912 0.880	1.007 0.964 0.970	0.372 0.339 0.280
lstm-rsm-37	Humidity	0.013 0.009 0.007	0.938 0.919 0.884	0.968 0.942 0.991	0.389 0.352 0.273
lstm-rsm-38	Pressure	0.014 0.008 0.007	0.936 0.919 0.860	0.938 0.925 0.917	0.402 0.361 0.306
lstm-rsm-39	Wind Magnitude	0.014 0.009 0.007	0.940 0.922 0.882	0.942 0.948 0.973	0.401 0.352 0.280
lstm-rsm-40	Downward LW Radiation	0.015 0.009 0.007	0.925 0.916 0.890	1.012 0.955 0.949	0.371 0.343 0.289
lstm-rsm-41	Downward SW Radiation	0.016 0.009 0.007	0.923 0.914 0.877	1.072 0.967 0.962	0.347 0.338 0.284

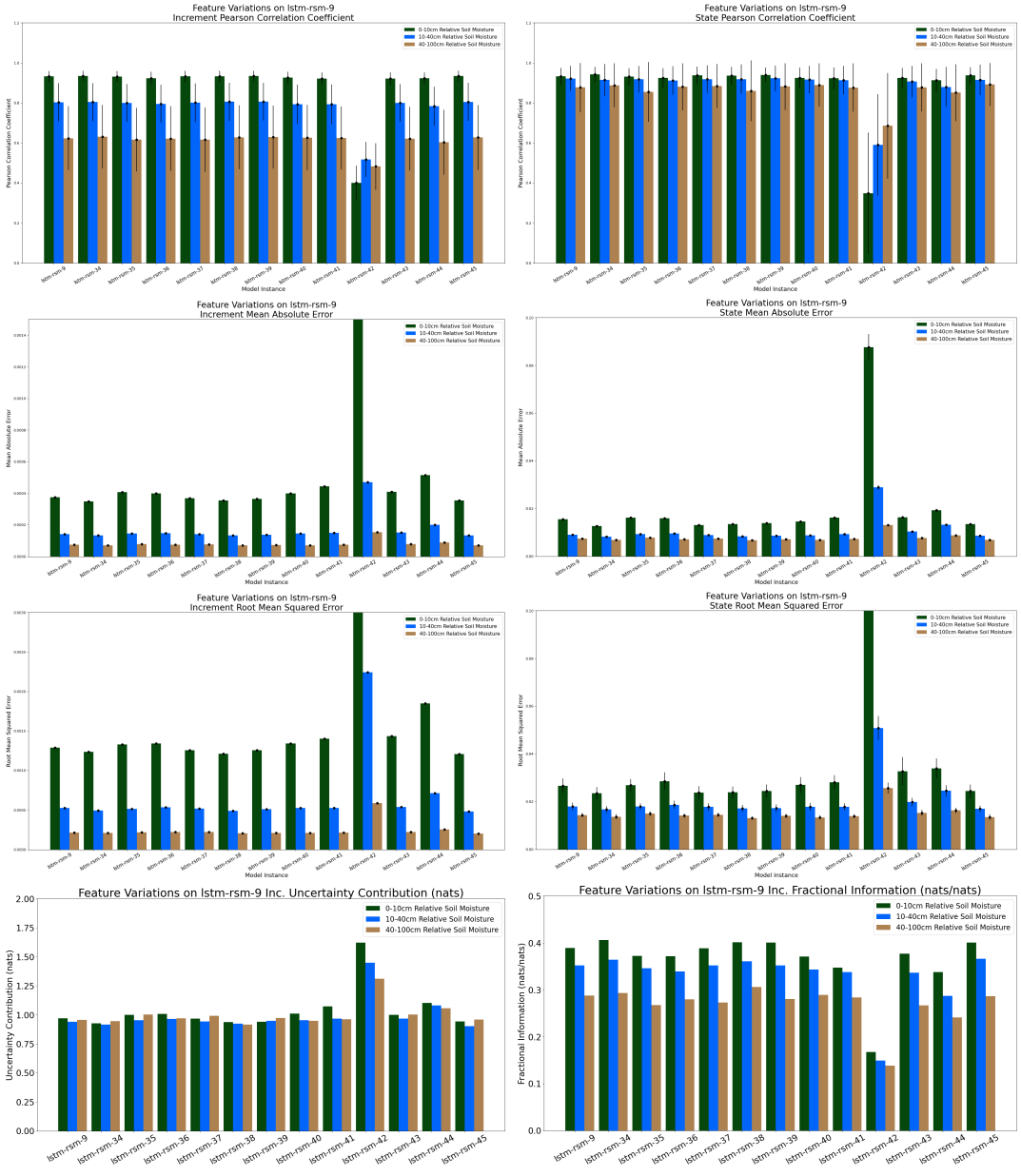

Figure 4.16: Bulk metrics for initial FNN training runs

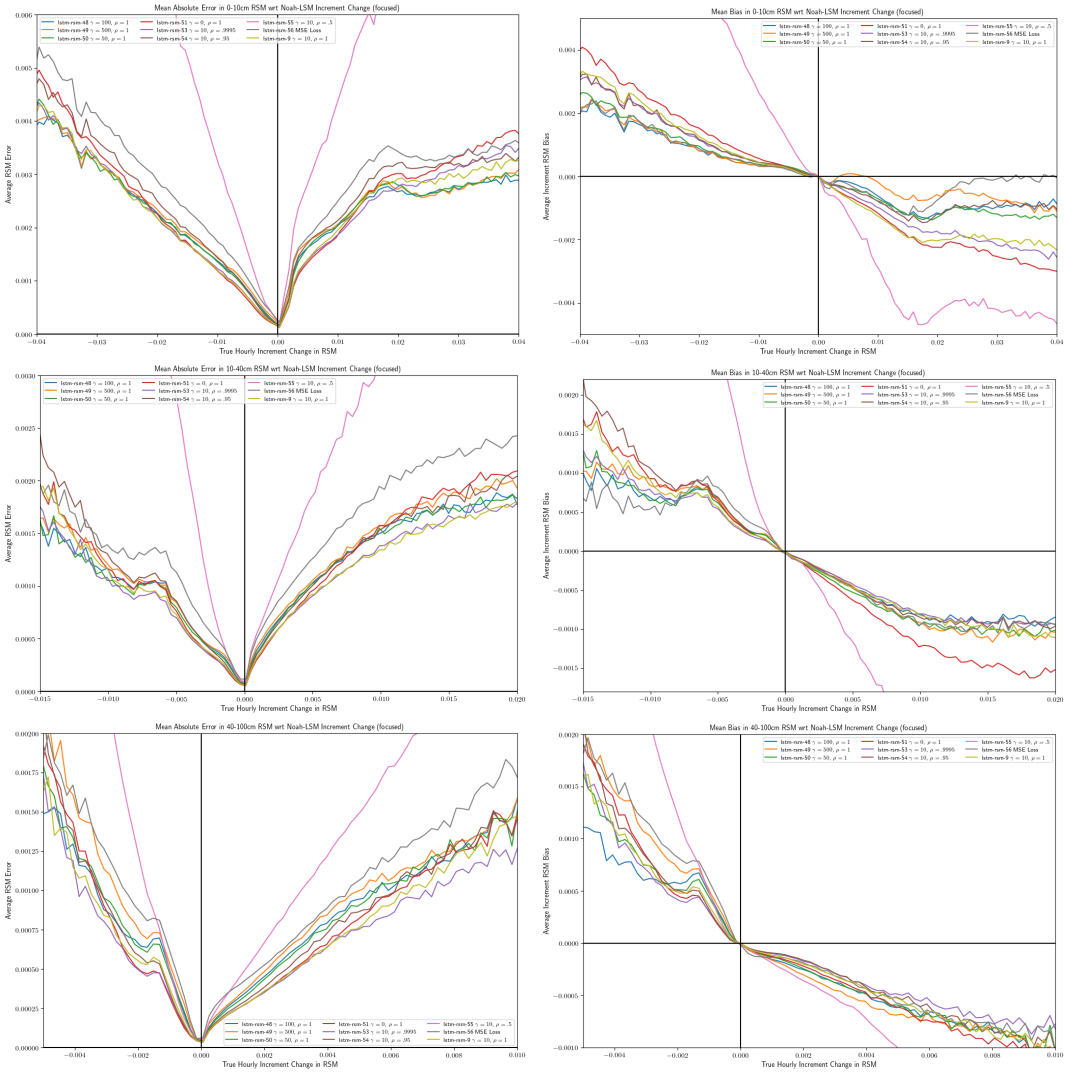

Figure 4.17: Mean absolute error (left) and bias (right) with respect to actual hourly increment change in RSM for models trained with loss function manipulations.

Name	Loss	γ	ρ	State MAE	State CC	Info Loss	Frac. Info
lstm-rsm-9	MAE	10	1.	0.015 0.009 0.007	0.932 0.923 0.876	0.970 0.943 0.960	0.388 0.350 0.287
lstm-rsm-51	MAE	0	1.	0.015 0.009 0.007	0.936 0.913 0.883	0.925 0.904 0.933	0.408 0.368 0.298
lstm-rsm-50	MAE	50	1.	0.016 0.012 0.009	0.926 0.875 0.842	1.031 1.013 1.085	0.362 0.316 0.233
lstm-rsm-48	MAE	100	1.	0.017 0.013 0.010	0.921 0.889 0.779	1.062 1.057 1.119	0.348 0.297 0.217
lstm-rsm-49	MAE	500	1.	0.021 0.015 0.012	0.904 0.856 0.741	1.117 1.125 1.220	0.325 0.271 0.176
lstm-rsm-53	MAE	10	.9995	0.014 0.009 0.007	0.940 0.923 0.866	0.959 0.931 0.969	0.392 0.355 0.281
lstm-rsm-54	MAE	10	.95	0.012 0.008 0.007	0.942 0.929 0.899	1.019 0.983 0.990	0.366 0.330 0.270
lstm-rsm-55	MAE	10	.5	0.016 0.010 0.007	0.911 0.903 0.858	1.364 1.225 1.148	0.243 0.228 0.201
lstm-rsm-56	MSE	10	1.	0.024 0.018 0.014	0.893 0.791 0.640	1.194 1.176 1.344	0.295 0.243 0.132

Table 4.8: Bulk statistic results for lstm-rsm-9 variants trained with isolated changes in loss function modifications.

4.5 Case Studies

References

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P. (2017). The CAMELS data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences*, 21(10):5293–5313. Publisher: Copernicus GmbH.

Baldwin, M. and Mitchell, K. (1997). The NCEP hourly multisensor US precipitation analysis for operations and GCIP research. *American Meteorological Society*, 54(Preprints, 13th Conference on Hydrology, Long Beach, CA):55.

Balsamo, G., Beljaars, A., Scipal, K., Viterbo, P., Hurk, B. v. d., Hirschi, M., and Betts, A. K. (2009). A Revised Hydrology for the ECMWF Model: Verification from Field Site to Terrestrial Water Storage and Impact in the Integrated Forecast System. *Journal of Hydrometeorology*, 10(3):623–643. Publisher: American Meteorological Society Section: Journal of Hydrometeorology.

Barlage, M., Chen, F., Tewari, M., Ikeda, K., Gochis, D., Dudhia, J., Rasmussen, R., Livneh, B., Ek, M., and Mitchell, K. (2010). Noah land surface model modifications to improve snowpack prediction in the Colorado Rocky Mountains. *Journal of Geophysical Research: Atmospheres*, 115(D22). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2009JD013470.

Berg, A. A., Famiglietti, J. S., Walker, J. P., and Houser, P. R. (2003). Impact of bias correction to reanalysis products on simulations of North American soil moisture and hydrological fluxes. *Journal of Geophysical Research: Atmospheres*, 108(D16). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2002JD003334.

Betts, A. K., Chen, F., Mitchell, K. E., and Janjić, Z. I. (1997). Assessment of the Land Surface and Boundary Layer Models in Two Operational Versions of the NCEP Eta Model Using FIFE Data. *Monthly Weather Review*, 125(11):2896–2916. Publisher: American Meteorological Society Section: Monthly Weather Review.

Brocca, L., Melone, F., Moramarco, T., and Morbidelli, R. (2010). Spatial-temporal variability of soil moisture and its estima-

tion across scales. *Water Resources Research*, 46(2). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2009WR008016.

Cartwright, J. H. E. and Piro, O. (1992). THE DYNAMICS OF RUNGE–KUTTA METHODS. *International Journal of Bifurcation and Chaos*, 02(03):427–449.

Case, J. L., Schultz, C. J., and Hain, C. R. (2023). Role of Antecedent Soil Moisture and Vegetation Stress in Lightning-Initiated Wildfires. NTRS Author Affiliations: Ensco (United States), Marshall Space Flight Center NTRS Meeting Information: 103rd American Meteorological Society Annual Meeting; 2023-01-08 to 2023-01-12; undefined NTRS Document ID: 20230000164 NTRS Research Center: Marshall Space Flight Center (MSFC).

Case, J. L. and White, K. D. (2014). Assessment of the 3-km SPoRT Land Information System for Drought Monitoring and Hydrologic Forecasting. Technical report, NASA Martial Space Flight Center.

Case, J. L., White, K. D., Fuell, K. K., and Hain, C. R. (2022). NASA SPoRT Land Information System Products for Soil Moisture Analysis.

Chen, F., Janjic, Z., and Mitchell, K. (1997). Impact of Atmospheric Surface-layer Parameterizations in the new Land-surface Scheme of the NCEP Mesoscale Eta Model. *Boundary Layer Meteorology*, 85(3):391–421. Num Pages: 391-421 Place: Dordrecht, Netherlands Publisher: Springer Nature B.V.

Chen, F., Mitchell, K., Schaake, J., Xue, Y., Pan, H.-L., Koren, V., Duan, Q. Y., Ek, M., and Betts, A. (1996). Modeling of land surface evaporation by four schemes and comparison with FIFE observations. *Journal of Geophysical Research: Atmospheres*, 101(D3):7251–7268. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/95JD02165.

Chen, M., Shi, W., Xie, P., Silva, V. B. S., Kousky, V. E., Wayne Higgins, R., and Janowiak, J. E. (2008). Assessing objective techniques for gauge-based analyses of global daily precipitation. *Journal of Geophysical Research: Atmospheres*, 113(D4). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2007JD009132.

Cosby, B. J., Hornberger, G. M., Clapp, R. B., and Ginn, T. R. (1984). A Statistical Exploration of the Relationships of Soil Moisture Characteristics to

the Physical Properties of Soils. *Water Resources Research*, 20(6):682–690. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/WR020i006p00682.

Cosgrove, B. A., Lohmann, D., Mitchell, K. E., Houser, P. R., Wood, E. F., Schaake, J. C., Robock, A., Marshall, C., Sheffield, J., Duan, Q., Luo, L., Higgins, R. W., Pinker, R. T., Tarpley, J. D., and Meng, J. (2003). Real-time and retrospective forcing in the North American Land Data Assimilation System (NLDAS) project. *Journal of Geophysical Research: Atmospheres*, 108(D22). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2002JD003118.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs].

Dorman, J. L. and Sellers, P. J. (1989). A Global Climatology of Albedo, Roughness Length and Stomatal Resistance for Atmospheric General Circulation Models as Represented by the Simple Biosphere Model (SiB). *Journal of Applied Meteorology and Climatology*, 28(9):833–855. Publisher: American Meteorological Society Section: Journal of Applied Meteorology and Climatology.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929 [cs].

Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009.

Ek, M. B., Mitchell, K. E., Lin, Y., Rogers, E., Grunmann, P., Koren, V., Gayno, G., and Tarpley, J. D. (2003). Implementation of Noah land surface model advances in the National Centers for Environmental Prediction operational mesoscale Eta model. *Journal of Geophysical Research: Atmospheres*, 108(D22). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2002JD003296.

Fablet, R., Ouala, S., and Herzet, C. (2018). Bilinear Residual Neural Network for the Identification and Forecasting of Geophysical Dynamics. In *2018 26th*

European Signal Processing Conference (EUSIPCO), pages 1477–1481. ISSN: 2076-1465.

Filipović, N., Brdar, S., Mimić, G., Marko, O., and Crnojević, V. (2022). Regional soil moisture prediction system based on Long Short-Term Memory network. *Biosystems Engineering*, 213:30–38.

Fulton, R. A., Breidenbach, J. P., Seo, D.-J., Miller, D. A., and O'Bannon, T. (1998). The WSR-88D Rainfall Algorithm. *Weather and Forecasting*, 13(2):377–395. Publisher: American Meteorological Society Section: Weather and Forecasting.

Gutman, G. and Ignatov, A. (1998). The derivation of the green vegetation fraction from NOAA/AVHRR data for use in numerical weather prediction models. *International Journal of Remote Sensing*, 19(8):1533–1543. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/014311698215333.

Haines, W. B. (1930). Studies in the physical properties of soil. V. The hysteresis effect in capillary properties, and the modes of moisture distribution associated therewith. *The Journal of Agricultural Science*, 20(1):97–116.

Hansen, M. C., Defries, R. S., Townshend, J. R. G., and Sohlberg, R. (2000). Global land cover classification at 1 km spatial resolution using a classification tree approach. *International Journal of Remote Sensing*, 21(6-7):1331–1364. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/014311600210209.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780. Conference Name: Neural Computation.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Jacquemin, B. and Noilhan, J. (1990). Sensitivity study and validation of a land surface parameterization using the HAPEX-MOBILHY data set. *Boundary-Layer Meteorology*, 52(1):93–134.

Jarvis, P. G., Monteith, J. L., and Weatherley, P. E. (1976). The interpretation of the variations in leaf water potential and stomatal conductance found in

canopies in the field. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 273(927):593–610. Publisher: Royal Society.

Jin, J., Miller, N. L., and Schlegel, N. (2010). Sensitivity Study of Four Land Surface Schemes in the WRF Model. *Advances in Meteorology*, 2010(1):167436. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2010/167436.

Joyce, R. J., Janowiak, J. E., Arkin, P. A., and Xie, P. (2004). CMORPH: A Method that Produces Global Precipitation Estimates from Passive Microwave and Infrared Data at High Spatial and Temporal Resolution. *Journal of Hydrometeorology*, 5(3):487–503. Publisher: American Meteorological Society Section: Journal of Hydrometeorology.

Koren, V., Schaake, J., Mitchell, K., Duan, Q.-Y., Chen, F., and Baker, J. M. (1999). A parameterization of snowpack and frozen ground intended for NCEP weather and climate models. *Journal of Geophysical Research: Atmospheres*, 104(D16):19569–19585. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/1999JD900232.

Koster, R. D., Mahanama, S. P. P., Yamada, T. J., Balsamo, G., Berg, A. A., Boisserie, M., Dirmeyer, P. A., Doblas-Reyes, F. J., Drewitt, G., Gordon, C. T., Guo, Z., Jeong, J.-H., Lawrence, D. M., Lee, W.-S., Li, Z., Luo, L., Malyshev, S., Merryfield, W. J., Seneviratne, S. I., Stanelle, T., van den Hurk, B. J. J. M., Vitart, F., and Wood, E. F. (2010). Contribution of land surface initialization to subseasonal forecast skill: First results from a multi-model experiment. *Geophysical Research Letters*, 37(2). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2009GL041677.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M. (2018). Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110. Publisher: Copernicus GmbH.

Kumar, S. V., Mocko, D. M., Fadji, M., Hain, C. R., Fuchs, B., and Wade, R. (2024). The North American Land Data Assimilation System (NLDAS-3) Phase 3: Modeling Water Availability over North and Central America.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096. Publisher: Nature Publishing Group.

Lees, T., Reece, S., Kratzert, F., Klotz, D., Gauch, M., De Bruijn, J., Kumar Sahu, R., Greve, P., Slater, L., and Dadson, S. J. (2022). Hydrological concept formation inside long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences*, 26(12):3079–3101.

Liang, X., Lettenmaier, D. P., Wood, E. F., and Burges, S. J. (1994). A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres*, 99(D7):14415–14428. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/94JD00483.

Livneh, B., Xia, Y., Mitchell, K. E., Ek, M. B., and Lettenmaier, D. P. (2010). Noah LSM Snow Model Diagnostics and Enhancements. *Journal of Hydrometeorology*, 11(3):721–738. Publisher: American Meteorological Society Section: Journal of Hydrometeorology.

Mahfouf, J. F. and Noilhan, J. (1991). Comparative Study of Various Formulations of Evaporations from Bare Soil Using In Situ Data. *Journal of Applied Meteorology and Climatology*, 30(9):1354–1365. Publisher: American Meteorological Society Section: Journal of Applied Meteorology and Climatology.

Mahrt, L. and Ek, M. (1984). The Influence of Atmospheric Stability on Potential Evaporation. *Journal of Climate and Applied Meteorology*, 23(2):222–234.

Mahrt, L. and Pan, H. (1984). A two-layer model of soil hydrology. *Boundary-Layer Meteorology*, 29(1):1–20.

Mitchell, K. (2005). The Community Noah Land Surface Model Users' Guide Version 2.7.1.

Mitchell, K., Helin, W., Lu, S., Gayno, G., and Meng, J. (2005). NCEP implements major upgrade to its medium-range global forecast system, including land-surface component. *GEWEX News*, 15(4):8–9.

Mitchell, K. E., Lohmann, D., Houser, P. R., Wood, E. F., Schaake, J. C., Robock, A., Cosgrove, B. A., Sheffield, J., Duan, Q., Luo, L., Higgins, R. W., Pinker, R. T., Tarpley, J. D., Lettenmaier, D. P., Marshall, C. H., Entin, J. K., Pan, M., Shi, W., Koren, V., Meng, J., Ramsay, B. H., and Bailey, A. A. (2004). The multi-institution North American Land Data Assimilation System (NLDAS): Utilizing multiple GCIP products and partners in a continental distributed hydrological modeling system. *Journal of Geophysical Research: Atmospheres*, 109(D7). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2003JD003823.

Mozer, M. C. (1995). A focused backpropagation algorithm for temporal pattern recognition. In *Backpropagation: theory, architectures, and applications*, pages 137–169. L. Erlbaum Associates Inc., USA.

Nash, J. E. and Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10(3):282–290.

Nearing, G. S., Mocko, D. M., Peters-Lidard, C. D., Kumar, S. V., and Xia, Y. (2016). Benchmarking NLDAS-2 Soil Moisture and Evapotranspiration to Separate Uncertainty Contributions. *Journal of Hydrometeorology*, 17(3):745–759. Publisher: American Meteorological Society Section: Journal of Hydrometeorology.

Nguyen, T. T., Trahay, F., Domke, J., Drozd, A., Vatai, E., Liao, J., Wahib, M., and Gerofi, B. (2022). Why Globally Re-shuffle? Revisiting Data Shuffling in Large Scale Deep Learning. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1085–1096. ISSN: 1530-2075.

Nonnenmacher, M. and Greenberg, D. S. (2021). Deep Emulators for Differentiation, Forecasting, and Parametrization in Earth Science Simulators. *Journal of Advances in Modeling Earth Systems*, 13(7):e2021MS002554. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2021MS002554.

O., S. and Orth, R. (2021). Global soil moisture data derived through machine learning trained with in-situ measurements. *Scientific Data*, 8(1):170. Publisher: Nature Publishing Group.

of the Interior, E. R. O. a. S. C. S. G. S. S. D. (1997). USGS 30 ARC-second Global Elevation Data, GTOPO30.

Otkin, J. A., Anderson, M. C., Hain, C., Svoboda, M., Johnson, D., Mueller, R., Tadesse, T., Wardlow, B., and Brown, J. (2016). Assessing the evolution of soil moisture and vegetation conditions during the 2012 United States flash drought. *Agricultural and Forest Meteorology*, 218-219:230–242.

Pan, H.-L. and Mahrt, L. (1987). Interaction between soil hydrology and boundary-layer development. *Boundary-Layer Meteorology*, 38(1):185–202.

Pinker, R. T., Tarpley, J. D., Laszlo, I., Mitchell, K. E., Houser, P. R., Wood, E. F., Schaake, J. C., Robock, A., Lohmann, D., Cosgrove, B. A., Sheffield, J., Duan, Q., Luo, L., and Higgins, R. W. (2003). Surface radiation budgets in support of the GEWEX Continental-Scale International Project (GCIP) and the GEWEX Americas Prediction Project (GAPP), including the North American Land Data Assimilation System (NLDAS) project. *Journal of Geophysical Research: Atmospheres*, 108(D22). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2002JD003301.

Ren, Y., Ma, C., and Ying, L. (2024). Understanding the Generalization Benefits of Late Learning Rate Decay. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pages 4465–4473. PMLR. ISSN: 2640-3498.

Rosero, E., Yang, Z.-L., Wagener, T., Gulden, L. E., Yatheendradas, S., and Niu, G.-Y. (2010). Quantifying parameter sensitivity, interaction, and transferability in hydrologically enhanced versions of the Noah land surface model over transition zones during the warm season. *Journal of Geophysical Research: Atmospheres*, 115(D3). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2009JD012035.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536. Publisher: Nature Publishing Group.

Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H.-y., Iredell, M., Ek, M., Meng, J., Yang, R., Mendez, M. P., Dool, H. v. d., Zhang, Q., Wang, W., Chen, M., and Becker, E. (2014). The NCEP Climate Forecast System Version 2. *Journal of Climate*, 27(6):2185–2208. Publisher: American Meteorological Society Section: Journal of Climate.

Schaake, J. C., Koren, V. I., Duan, Q.-Y., Mitchell, K., and Chen, F. (1996). Simple water balance model for estimating runoff at different spatial and temporal scales. *Journal of Geophysical Research: Atmospheres*, 101(D3):7461–7475. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/95JD02892.

Schilling, D. (2005). Entropy, Relative Entropy, and Mutual Information. In *Elements of Information Theory*, pages 13–55. John Wiley & Sons, Ltd, 2 edition. Section: 2 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/047174882X.ch2.

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. arXiv:1706.03762 [cs].

Wei, H., Xia, Y., Mitchell, K. E., and Ek, M. B. (2011). Improvement of the Noah land surface model for warm season processes: evaluation of water and energy flux simulation. *Hydrological Processes*, 27(2):297–303. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hyp.9214.

White, A. T., White, K. D., Hain, C. R., Antia, M., Fuell, K., and Case, J. L. (2025). NASA SPoRT's Streamflow-AI: Updates and Advancements. NTRS Author Affiliations: University of Alabama in Huntsville, NOAA National Weather Service, Marshall Space Flight Center, Amentum, Ensco (United States) NTRS Meeting Information: 105th American Meteorological Society (AMS) Annual Meeting; 2025-01-12 to 2025-01-16; undefined NTRS Docu-

ment ID: 20250000353 NTRS Research Center: Marshall Space Flight Center (MSFC).

Xia, Y., Mitchell, K., Ek, M., Cosgrove, B., Sheffield, J., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Duan, Q., and Lohmann, D. (2012a). Continental-scale water and energy flux analysis and validation for North American Land Data Assimilation System project phase 2 (NLDAS-2): 2. Validation of model-simulated streamflow. *Journal of Geophysical Research: Atmospheres*, 117(D3). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2011JD016051.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Lettenmaier, D., Koren, V., Duan, Q., Mo, K., Fan, Y., and Mocko, D. (2012b). Continental-scale water and energy flux analysis and validation for North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products. *Journal of Geophysical Research: Atmospheres*, 117(D3). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2011JD016048.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2022). Are Transformers Effective for Time Series Forecasting? arXiv:2205.13504 [cs].