

Article

DeepPaSTL: Spatio-Temporal Deep Learning Methods for Predicting Long-Term Pasture Terrains using Synthetic Datasets

Murtaza Rangwala ^{1*}, Jun Liu ^{1}, Kulbir Singh Ahluwalia ², Shayan Ghajar ³, Harnaik Singh Dhami ⁴, Benjamin F. Tracy¹, Pratap Tokekar⁴, Ryan K. Williams ¹

¹ Virginia Tech, Blacksburg; {murtazar, junliu, bftracy, rywilli1}@vt.edu

² University of Illinois, Urbana-Champaign; ksa5@illinois.edu

³ Oregon State University, Corvallis; ghajars@oregonstate.edu

⁴ University of Maryland, College Park; {dhami, tokekar}@umd.edu

* Correspondence: murtazar@vt.edu

1 **Abstract:** Effective management of dairy farms requires an accurate prediction of pasture biomass. **2** Generally, estimation of pasture biomass requires site-specific data, or often perfect world as- **3** sumptions to model prediction systems when field measurements or other sensory inputs are **4** unavailable. However, for small enterprises, regular measurements of site-specific data are often **5** inconceivable. In this study, we approach the estimation of pasture biomass by predicting sward **6** heights across the field. A convolution based sequential architecture is proposed for pasture **7** height predictions using deep learning. We develop a process to create synthetic datasets that **8** simulate the evolution of pasture growth over a period of 30 years. The deep learning based **9** pasture prediction model (DeepPaSTL) is trained on this dataset while learning the spatiotempo- **10** ral characteristics of pasture growth. The architecture purely learns from the trends in pasture **11** growth through available spatial measurements and is agnostic to any site-specific data, or climatic **12** conditions, such as temperature, precipitation, or soil condition. Our model performs within a **13** 12% error margin even during the periods with the largest pasture growth dynamics. The study **14** demonstrates the potential scalability of the architecture to predict any pasture size through a **15** quantization approach during prediction. Results suggest that the DeepPaSTL model represents a **16** useful tool for predicting pasture growth both for short and long horizon predictions, even with **17** missing or irregular historical measurements.

18 **Keywords:** Computer Vision; Convolution; Neural Network; Remote Sensing; Recurrent; Se- **19** quence; Biomass; Yield; Crop; Image Analysis, Agriculture

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Agronomy* **2021**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Submitted to *Agronomy* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

20 1. Introduction

21 Pasture lands provide an extensive ecosystem for grazing, maintaining plant and **22** animal biodiversity, and regulating soil erosion [1]. Furthermore, pasture lands are **23** arguably one of the primary and cheapest sources of livestock feed, particularly where **24** agricultural enterprises are not feasible [2]. The profitability of a pasture-dairy based **25** farm heavily depends on the maximum utilization of the on-farm grown pastures, where **26** feed availability for livestock can vary as widely as 50% [3,4]. The inherent temporal **27** and spatial dependencies of pasture growth lead to high uncertainty in estimates for **28** sward height data, especially when grasslands cannot be monitored with labor-intensive **29** traditional methods. This problem is essential as incorrect estimates result in wastage **30** in areas with high pasture availability and underfeeding of livestock at low pasture **31** availability [4]. Monitoring pasture growth through Unmanned Aerial Vehicles (UAVs) **32** [3] equipped with LIDAR or cameras can inform decisions for pasture feed allocation to **33** maximize profitability. However, the deployment of these remote sensing UAVs and the **34** subsequent time to process and interpret the data consumes valuable resources that may **35** hinder timely decision making for daily feed allocation.

Traditional numerical methods for prediction models of pastures have been proposed to help alleviate the problem of regular field measurements. They rely either on a perfect model of the site with extensive inputs such as soil conditions, crop physiology, and reproduction or rely on simplified measurements of site-specific data to generate yield predictions [5,6]. More significantly, even when site-specific data is available to either process-based models, it is an uphill battle to calibrate the models due to uncertainty in the parameters. Prior methods generally ignored the uncertainty in the data inputs and empirically calibrated their models with ground truth observations. However, when uncertainties in parameter values are considered, this uncertainty translated to large errors in scenarios where these parameters did not lie in the initial calibrated distribution [6].

In contrast, time series prediction techniques based on statistical models or machine learning are capable of learning not only through a generic set of model parameters or field measurements such as temperature changes, soil conditions or precipitation, but also capable of being agnostic to these data inputs by learning these features implicitly from historical pasture data [7,8], [9]. The flexibility offered by these algorithms opens up a tremendous opportunity to support decision making systems for agricultural prediction and planning tools even with sparse data and measurements. Statistical models generally rely on either time-series regression models, through spatial correlation, or through a combination of spatio-temporal variations. One advantage of statistical models is their inherent capability to assess model uncertainties, which machine learning models need to be adapted to specifically to capture these uncertainties. Despite the caveats to the added complexity from machine learning methods, they have limited reliance on site-specific data, and allow a transparent assessment of parameter uncertainties. For example, if a Bayesian Learning [10] is employed for a neural network based prediction model, the predictions would reflect a wider confidence interval if the model cannot adequately represent future pasture yield given its history and if available site-specific data. However, the current methods are generally focused on predicting pasture yields and cannot adequately address the issue of predicting pasture maps or specifically the individual sward heights across the complete fields of variable sizes, especially for long horizon predictions [6] or large pastures with variable size.

To address this issue, we utilize tools from recent advances in computer vision techniques, especially convolution neural networks (CNNs) [11,12] that have provided excellent results in long-term frame predictions for video sequences [13–16] and are also quite successfully used to capture intricate features of images or video frames [17–23]. The main advantage of deep learning models specifically based on CNNs is their capability to consider a map of historical sward heights in a field as an input sequence and predict the future map of sward heights of the pasture. With a well-designed neural network, and sufficient sward height data for training, the model has the capacity to provide useful insights on how to solve this complex and dynamic spatiotemporal problem. Encoder-Decoder models based on Convolutional Long Short-Term Memory (ConvLSTM) [20] models provide a general framework for spatiotemporal sequence-to-sequence learning problems. This is achieved by training connected ConvLSTMs that encode patterns within the historical observations and then unfold them to perform multi-step predictions of the future pasture terrains.

As a step towards the overall goal of predicting the pastureland environments, we propose a novel deep learning architecture, *Deep Pasture Spatio-Temporal Learning* (DeepPaSTL) that not only predicts the sward height data of pastures with high accuracy, but also provides a computationally efficient model of determining its prediction uncertainty. The proposed methodology reduces the burden of field measurements of the pasturelands by potentially reducing the frequency of measurements for areas that the DeepPasTL predicts with high certainty. For training, we create a new dataset that is generated from 30 years of historical data through a dynamic Gaussian mixture model (GMM), and evaluation is done both on a synthetic dataset derived from the simulated

90 data and also from 3D modeled grass pastures in Gazebo [24]. The aim of this paper is
 91 not just an evaluation of deep learning performance but to introduce a new direction for
 92 prediction-based systems on spatiotemporal evolution of pasture environments.

93 2. Materials and Methods

94 2.1. Problem Formulation

95 The goal of our study is to learn and predict the evolution of pasture growths
 96 through previously observed field measurements of sward heights. By applying a novel
 97 deep learning methodology to this problem, we forecast the future sward height maps
 98 of a variable length time horizon. Generally, in the real world, field measurements
 99 of pastures are performed every few days. Estimating the future of sward heights or,
 100 more generally, understanding how the pasture terrain evolves based on these historical
 101 measurements is of utmost importance to plan grazing activity or allocate resources for
 102 field measurements in the future, especially when predictions can be uncertain. This
 103 problem can be regarded as spatiotemporal sequence forecasting and can be solved
 104 through the sequence-to-sequence learning [25] within the domain of deep learning.

105 To enable training of the prediction network, we generate a synthetic dataset \mathcal{Z} of
 106 dynamic 2D maps of pastures simulating grass growth over time based on publicly avail-
 107 able historical pasture yield data, as described in Section 2.2. To this end, we consider the
 108 sward heights of pastures as an evolving 3D spatiotemporal process. Formally, we can
 109 now define the pasture terrain prediction as, given a periodically observed data $Z_{1,\dots,L_{in}}$,
 110 where $Z_i \in \mathcal{Z}$, denotes the sward height measurements of the field in an $N \times N$ grid,
 111 $Z_i \in \mathbb{R}^{N \times N}$, the goal is to predict the most likely L_{out} sequences, $Z_{L_{in}+1}, \dots, Z_{L_{in}+L_{out}}$,
 112 given the previous L_{in} sequences of sward heights,

$$Z_{L_{in}+1}, \dots, Z_{L_{in}+L_{out}} = \arg \max_{\hat{Z}_{L_{in}+1}, \dots, \hat{Z}_{L_{in}+L_{out}}} p(\hat{Z}_{L_{in}+1}, \dots, \hat{Z}_{L_{in}+L_{out}} | Z_1, \dots, Z_{L_{in}}). \quad (1)$$

113 Moreover, we also compare the accuracy of the results when the model training
 114 and inference are adapted with an Approximate Bayesian Learning with Markov Chain
 115 Monte Carlo (MCMC) [26] sampling to enable prediction of sward heights with uncer-
 116 tainty estimates as described in Section 2.6.

117 2.2. Simulated SpatioTemporal Dataset

118 We utilize the historical pasture data generated using Agri-cultural Production
 119 Systems sIMulator (APSIM) Next Generation's modules. Three sites in Iowa were
 120 selected in APSIM's Met module from 1979 to 2013 [27]. Site-specific parameters such as
 121 rain, temperature, day length, solar radiation, snowfall, and atmospheric pressure were
 122 considered from the dataset. We use mixed, fine loamy, superactive, mesic Hapludolls
 123 soil [27] available in APSIM's module and also common in Iowa to generate average
 124 pasture heights, and the SoilOM module was set to 1000 kg/ha initial surface residue.
 125 APSIM's tall fescue AgPasture module was used for modeling forage species [28] with
 126 the following parameters: initial values for belowground, aboveground biomass are
 127 set to 1000 kg/ha and 3000 kg/ha, with a rooting depth of 1m. NO₃-N was used for
 128 fertilizer application with a bi-yearly schedule of 84 kg N/ha on the first day of January
 129 and August. Since we simulate an ungrazed pasture, we disable APSIM's grazing
 130 module, and an average pasture height is generated through the above parameters as
 131 shown in Figure 1a.

To generate a 2D map of pasture environments, an evolving process of pastures
 is simulated through a Gaussian Mixture Model (GMM) [29,30]. The dynamic GMM
 process is defined as,

$$Z_t(x, y) = \sum_{j=1}^K w_j(t) g_j(x, y) = \mathbf{w}^T(t) \mathbf{g}(x, y),$$

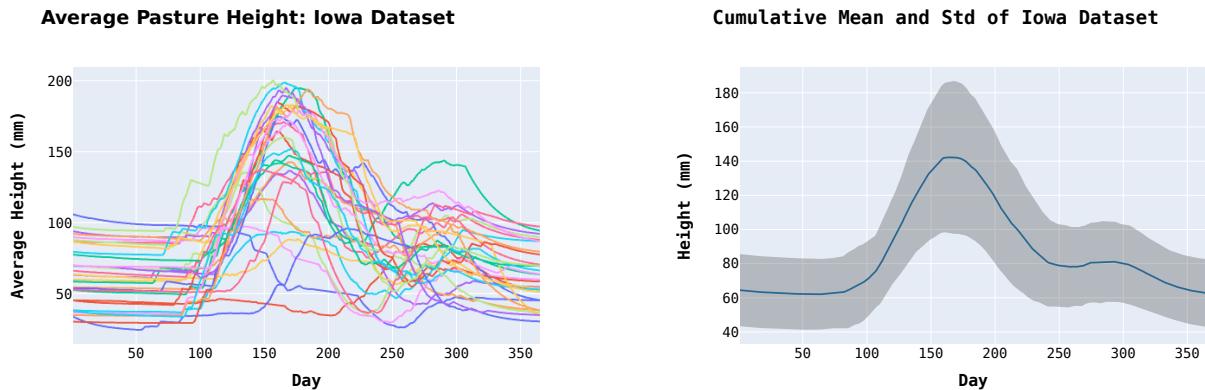


Figure 1. (a) Average height and (b) mean and standard deviation of the 30 years of historical data across 3 sites in Iowa from APSIMs Met module are shown for each day of the year.

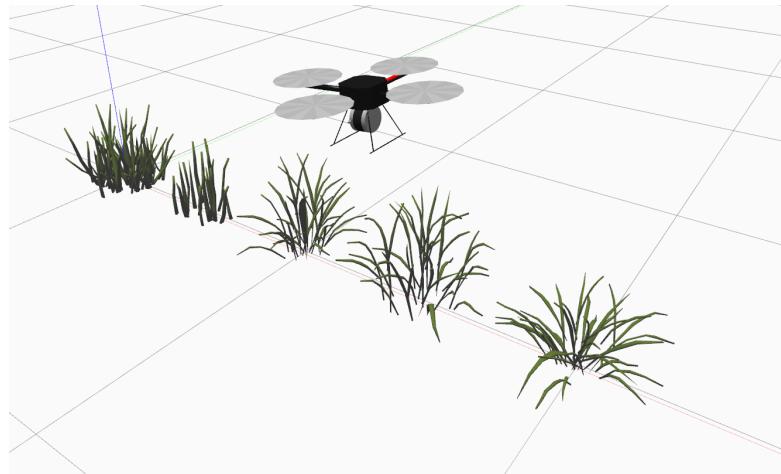


Figure 2. Grass models generated in Gazebo to populate the pasture terrain.

where $(x, y) \in \mathbb{R}^2$ is the 2D coordinates of the pasture, $w_j(t) \in \mathbb{R}^1$ is the weight associated with each basis function $g_j(x, y) \in \mathbb{R}^1$ for the corresponding location (x, y) and time t , and K is the number of basis functions, and $Z_t \in \mathbb{R}^2$ is the height of the pasture at location (x, y) at time t . The basis function is then defined as,

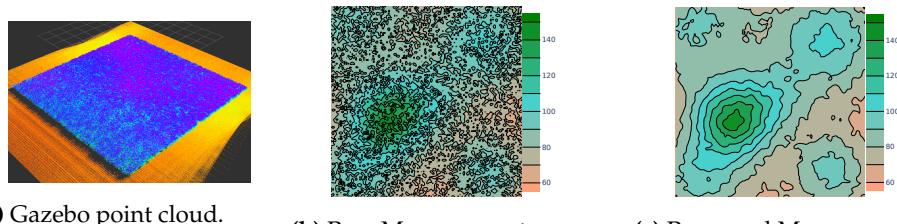
$$g_j(x, y) = \exp \left[-\frac{[(x, y) - (k_{x,j}, k_{y,j})]^2}{2l_j^2} \right],$$

132 where l_j is the length scale, and $(k_{x,j}, k_{y,j}) \in \mathbb{R}^2$ is the corresponding j th basis of the
 133 function $g_i(x, y)$. The dynamics of each weight w_j is modeled using random walk (1D)
 134 across different time steps t .

135 Finally, a pasture field is generated and mapped to a $10m \times 10m$ area. In order
 136 to match the rate of growth of sward heights from the historical data, Figure 1b, we
 137 add a bias to the results $Z_t(x, y) \leftarrow Z_t(x, y) + m_t - \bar{Z}_t$, where m_t , \bar{Z}_t is the mean of
 138 the historical and simulated pasture heights respectively. Additionally, a truncated
 139 Gaussian noise $\sigma(0, 1)$ is added to further match real-world measurements of sward
 140 heights. These steps are repeated for all days in 30 years of data and a synthetic dataset
 141 $\mathcal{Z} = \{Z_t | t = 0, \dots, T\} \in \mathbb{R}^{100 \times 100}$ of 2D pasture sward heights is generated which
 142 correlates to 100 point measurements per m^2 , and T is the total number of days in the
 143 historical dataset in 30 years from APSIM's Met module.

144 2.3. Pasture Construction for Evaluation

145 In order to reconstruct pasture environments similar to the real-world, [as part](#)
146 [of this study we developwe generate](#) five different types of 3D grass models [using](#)
147 [the Gazebo simulation and design tool, \[24\]](#), Figure 2. A $10m \times 10m$ patch is then
148 generated in Gazebo and populated with these [3D](#) grass models with a density of
149 250 grass models/ m^2 . To reduce computational requirements, we split the Gazebo
150 model in $2m \times 2m$ patches. Grass heights are modulated by re-scaling the model size to
151 fit the approximate heights in the simulated dataset given by \mathcal{Z} . In order to simulate
152 field measurements by UAV, we equip the standard hector quad-copter available in
153 Gazebo with LIDAR and measure the point clouds over the pasture Figure 3a. Standard
154 crop box filters in Gazebo are utilized to remove noise from the LIDAR measurements,
155 and the height of the sward heights are measured with respect to the ground plane of
156 the model, i.e., the perimeter of the pasture. Raw measurements Figure 3b of the point
157 cloud data are not particularly suited for neural networks due to a large noise floor for
158 each coordinate in the map. To ease the prediction for the neural network, we process
159 the raw point cloud through a median and flat convolution filter with a kernel size of
160 3×3 effectively smoothing the surface to a large degree Figure 3c. Due to the large
161 computational time required to generate simulated pastures in Gazebo, we limit our
162 3D pasture models to 30 samples of $100m \times 100m$ within the following time period:
163 04/01/2019 to 07/26/2019. The selected time period has the highest pasture growth in
164 our simulated dataset Figure 1b and is indicative of a difficult prediction problem for the
165 DeepPaSTL architecture due to its heavy fluctuations of the sward height measurements.



(a) Gazebo point cloud. **(b)** Raw Measurements **(c)** Processed Measurements

Figure 3. (a) Point cloud surface as measured in Gazebo with a hector quad-copter equipped with LIDAR. (b) Raw measurements on the Gazebo point cloud in (a) in a contour plot. (c) Filtered measurements adapted for neural network predictions.

166

167 2.4. Data processing for Training and Inference

168 First, in order to accommodate a truly scalable solution that is agnostic to the spatial
169 dimensions of the pasture prediction problem, we train our model to predict on
170 quantized patches of pastures and stitch the final prediction together. This methodology
171 allows the model to accommodate varying pasture sizes for long-term predicts. Ad-
172 ditionally, several other processing steps on the dataset are performed to improve the
173 performance of the prediction model as described below:

- 174** • The use of convolution neural networks in deep learning introduces an unintended
175 side effect popularly termed as boundary effects [31,32], where artifacts are intro-
176 duced at the boundaries of the image due to no spatial information [33,34] available
177 when CNN filters pass over boundaries of the image. We circumvent this issue
178 by enlarging each image with size $\delta \leq 100$, pixels through mirror padding [35] to
179 add spatial information on the boundaries of each pasture image in the dataset
180 $\mathcal{Z} \in \mathbb{R}^{100 \times 100}$ updating our new training dataset to $\mathcal{Z}_{new} \in \mathbb{R}^{100+\delta \times 100+\delta}$.
- 181** • Training and inference of the neural network on original dimensions of the training
182 dataset \mathcal{Z}_{new} may potentially increase accuracy. However, it severely limits the
183 capability of the neural network to adapt to variable input dimensions while also
184 increasing computational requirements as GPU memory is a limited resource,

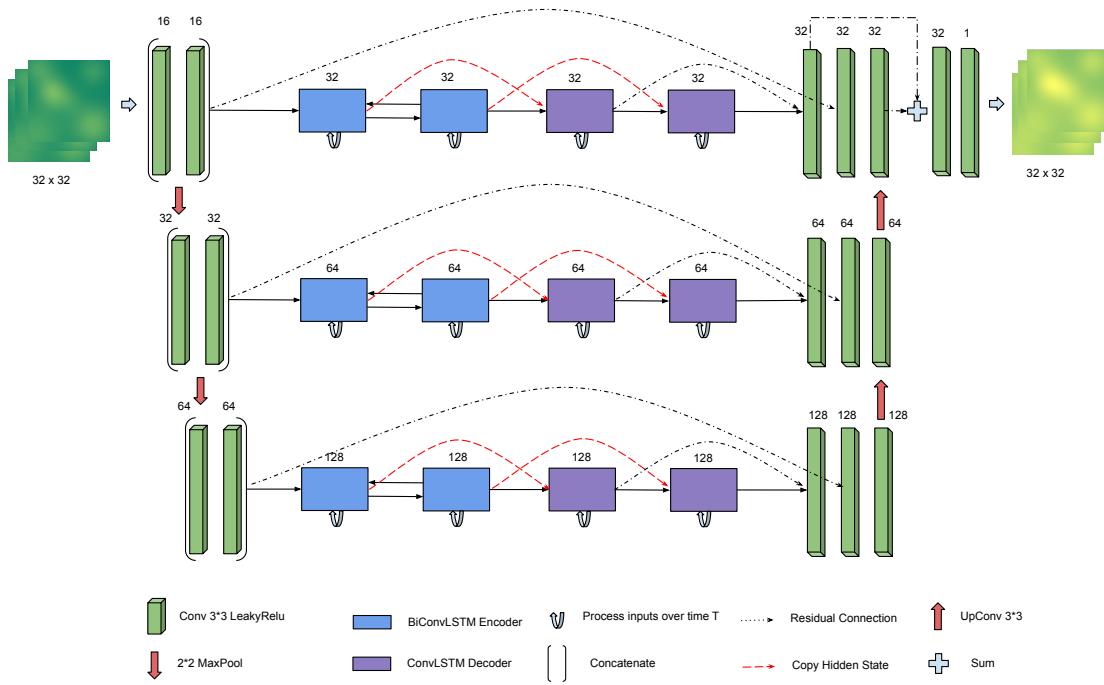


Figure 4. Encoder-decoder architecture with ConvLSTM and residual connections (example for 32×32 pixels in the lowest resolution). The encoder consists of the two initial 2D convolution layers that extract the initial features of the input. Subsequently, the BiConvLSTM encoders are deployed to learn the forward and backward correlations over these extracted features of the input sequence. The ConvLSTM decoder then recursively unwraps the hidden features encoded in the hidden state of the encoder, and the 2D convolution layers map it to output predictions. The number of feature maps of each CNN layer is denoted above their respective blocks.

185 specifically when training inputs with large dimensions. To this end, we quantize
 186 the training data $\mathcal{Z}_{new} \in \mathbb{R}^{100+\delta \times 100+\delta}$ into smaller sized patches of $\mathcal{Z}_q \in \mathbb{R}^{\delta \times \delta}$ with
 187 an overlap of 50% between them. The overlapping of the images and subsequent
 188 reconstruction of the image post inference through a weighted average allows us
 189 to mitigate boundary effects between each cropped frame, an undesirable artifact
 190 of CNN output that would occur if they were to be naively cropped without any
 191 overlaps. This methodology requires the neural network to only learn over small
 192 patches of the field and can be practically used to predict field sizes of any size
 193 $N \times N$, as long as the original image is appropriately processed to meet the input
 194 size of $\delta \times \delta$, where $N \geq \delta$.

- 195 • We fix the sequence length of the training inputs and output prediction to trajec-
 196 tories of time $L_{in}, L_{out} = 15$. The final input training set is then defined as input
 197 sequences of $\mathcal{Z}_{in} = \{\mathcal{Z}_{in}^i | i = 1, \dots, \tau - L_{in} - L_{out}\}$, where τ is the number of data
 198 points in the quantized dataset \mathcal{Z}_q . Each individual sequence for the backward
 199 propagation is $\mathcal{Z}_{in}^i = \{\mathcal{Z}_q^i, \dots, \mathcal{Z}_q^{i+L_{in}}\}$ where $\mathcal{Z}_{in}^i \in \mathbb{R}^{\delta \times \delta}$. Similarly, the target val-
 200 ues dataset $\mathcal{Y} = \{\mathcal{Y}^i | i = 1, \dots, \tau - L_{in} - L_{out}\}$ is created for training. Each input se-
 201 quence \mathcal{Z}_{in}^i has a corresponding target value $\mathcal{Y}_{out}^i = \{Y_q^{i+1+L_{in}}, \dots, Y_q^{i+1+L_{in}+L_{out}}\} \in$
 202 $\mathbb{R}^{\delta \times \delta}$, where $Y_q^j = Z_q^j$.

203 2.5. Deep Learning Model for Long-Term Prediction

The choice of our architecture Figure 4 is primarily motivated by our goal of spatiotemporal learning. Recently, ConvLSTMs [20] have shown remarkable progress in learning representations and future frame predictions of video sequences, precipitation nowcasting, and also for classification problems of deforestation. A ConvLSTM can be simply defined as a LSTM recurrent network [36], with convolution operations replacing

the matrix multiplication within a LSTM network as shown in Equation 2. LSTM networks are designed to process temporal dependencies by propagating its hidden state across time [19,25,36–42], or more simply, they transfer an aggregated history to allow future predictions to take advantage of the past. Similarly, the emergence of ConvLSTM is motivated by taking advantage of the temporal dependence of LSTMs and extending it as a spatiotemporal representation, making it an excellent choice for our application. The ConvLSTM architecture is defined as,

$$\begin{aligned} i_t &= \sigma(W_{xi} * U^t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i), \\ f_t &= \sigma(W_{xf} * U^t + W_{hf} * U_{t-1} + W_{cf} \circ C_{t-1} + b_f), \\ C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * U_t + W_{hc} * H_{t-1} + b_c), \\ o_t &= \sigma(W_{xo} * U^t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o), \\ H_t &= o_t \circ \tanh(C_t), \end{aligned} \quad (2)$$

where $U^t \in \mathbb{R}^{1 \times d \times d}$ is an input to the ConvLSTM layer, $(H_1, \dots, H_t) \in \mathbb{R}^{1 \times d \times d}$, $(C_1, \dots, C_t) \in \mathbb{R}^{1 \times d \times d}$ are the hidden and cell states of the ConvLSTM cell, and $i_t, f_t, o_t \in \mathbb{R}^{1 \times d \times d}$ are the interaction, forget, and output gates similar to an LSTM cell. The gates control the integration of information from the past and the present data to the next timestep. $*$ is the convolution operation, and \circ is the Hadamard Product [43].

In order to generate multi-step predictions, our architecture should be capable of identifying the underlying temporal patterns of available historical pasture growth L_{in} , and more so the spatial correlation within the pasture before generating predictions. To capture this spatiotemporal history, we introduce an encoder similar to the original ConvLSTM for precipitation nowcasting through radar data. However, we employ the use of Bi-ConvLSTM networks [44] similar to Bi-LSTMs [45], where we run two separate ConvLSTM networks each in the forward ($i \rightarrow i + L_{in}$) and reverse ($i + L_{in} \rightarrow i$) direction of the input sequence. By learning the bi-direction temporal dependencies of pasture growth, we enable our model to achieve a better representation of time-series data. The hidden states of the ConvLSTM networks are then merged with a CNN operation at each timestep before being fed to the subsequent networks, $H_t = f^{bi}(H_t^f, H_t^b)$, where f^{bi} is a CNN layer, H_t^f, H_t^b are the hidden states at time t of the ConvLSTM encoder in the forward and reverse direction respectively. The encoder recursively parses the spatiotemporal information in the input sequence and generates an aggregated hidden representation in the final step, which is then used as a basis for forecasting future growth. This approach allows our network to generate richer representations specifically for learning the trends in sward height growth by encoding the history of pasture dynamics through the encoder.

A decoder framework is then implemented to enable the reconstruction of future predictions based on the aggregated historical hidden representations of the encoder. Since we do not have information for future time steps, we only use ConvLSTM networks processing the output sequence in the forward direction. The decoders copy the last hidden state of the encoder networks as their own initial state. The decoder utilizes its own output states as an input for future timesteps along with the hidden states of the encoder to recursively generate predictions for pasture heights.

Finally, to increase the representational power of the DeepPaSTL architecture, we use CNNs to pre-encode the inputs before feeding the recurrent encoder-decoder networks. Similarly, the outputs of the encoder-decoder are also parsed through CNNs to generate the final prediction. We implement the encoder-decoder framework across three spatial resolutions by down-sampling them by a factor of 2 with MaxPool layers [46] to allow the network to learn dependencies at different spatial representations, akin to the ubiquitous U-Net CNN framework [47]. Similar to [20], we use two sets of BiConvLSTM for each encoder and similarly two ConvLSTM decoders to independently learn the concept of distance and correlation within its neighborhood [48]. The representations at

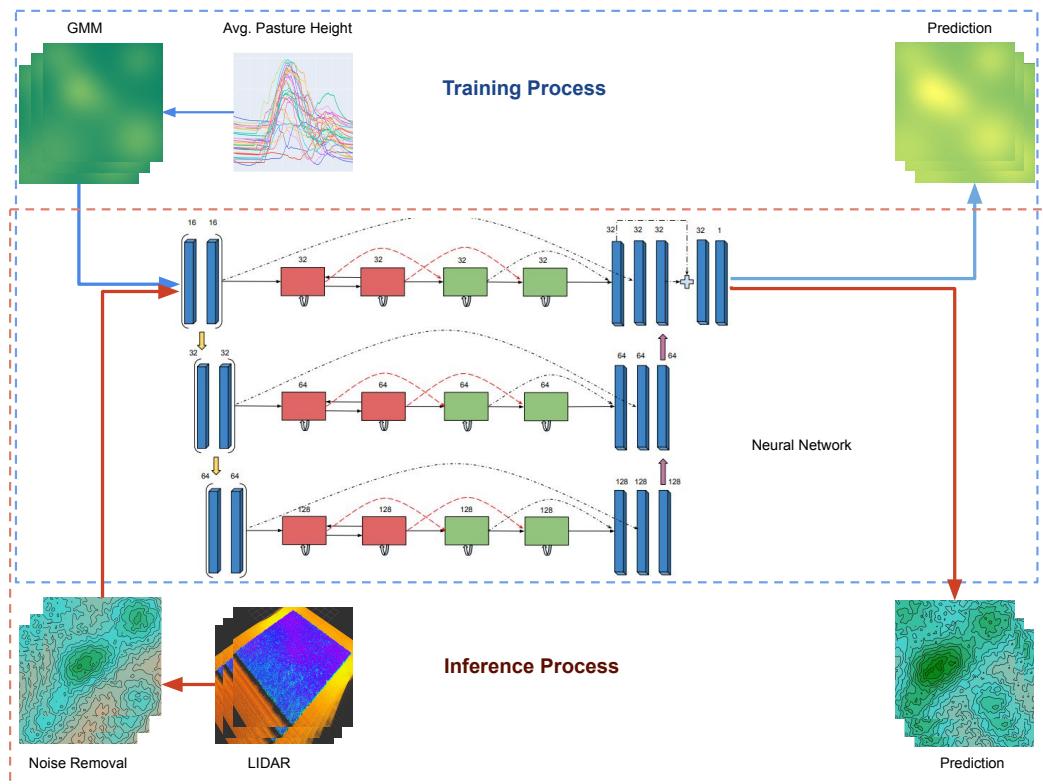


Figure 5. Process for training and inference of DeepPaSTL. Synthetic training datasets are created using GMM models based on the average pasture heights of Iowa sites. Real world field measurements can be obtained by using LIDAR point cloud measurements. The point cloud data is then processed to smooth out sensor noise and then DeepPaSTL is used to predict future pasture heights.

243 different spatial resolutions are finally merged together by up-sampling through CNNs.
 244 In order to improve training time, performance and negate the problem of vanishing
 245 gradients during training, we employ the use of residual connection [49–52], and batch
 246 normalization [53]. Residual connections from the pre-encoding to the post-encoding
 247 layers also help the network recreate the spatial context of the original images.

248 2.6. Uncertainty Estimation of the Model

249 Standard deep learning models that are trained through supervised learning do
 250 not estimate the uncertainty in its prediction. However, the paradigm of Bayesian
 251 Neural Networks (BNN) [10] enables the neural networks to estimate uncertainty in
 252 their outputs by evaluating the posterior distribution over its network weights. However,
 253 to model a large BNN, especially with the representational power required to forecast
 254 pasture growth, makes them computationally prohibitive. This is due to the fact that
 255 a full posterior distribution over the parameters of the neural network needs to be
 256 computed for each forward and backward pass. Recently, a computationally efficient
 257 method of approximating Bayesian inference [10] with the use of dropouts [26] was
 258 proposed. The key idea was to perform Markov Chain Monte Carlo (MCMC) sampling
 259 of the network parameters to generate stochastic inference of the network only in the
 260 forward pass. Dropouts in deep learning are more popularly used only during training to
 261 remove randomly sampled nodes from each layer l with a fixed probability p^l , to reduce
 262 overfitting and increase the robustness of the network by allowing each node to learn
 263 redundant and independent representations. However, [26] shows that introducing
 264 dropouts during inference enables the model to estimate uncertainty in its output.
 265 We utilize the approximate Bayesian inference in our model by introducing dropouts

²⁶⁶ between each layer preceding the final output layer with $p^l = 0.4$, and generate 500
²⁶⁷ samples of stochastic inference before estimating the average for the final prediction.

²⁶⁸ 2.7. Experiment Details

²⁶⁹ A brief comparison of different input of patch sizes, δ is used to compare the
²⁷⁰ accuracy of the architecture as input size increases. The main limitation of the patch size
²⁷¹ is attributed to the limited GPU memory (VRAM) available for training. The input sizes
²⁷² can be increased as large as the available system capacity allows, although through our
²⁷³ empirical evaluations, we observe that lower input sizes had better performance. Since it
²⁷⁴ is quite unlikely that field measurements of pastures are available for every consecutive
²⁷⁵ day, we compare results when the input observations are split apart every few days,
²⁷⁶ i.e., $s = \{1, 2, 4\}$ time intervals between each input in the sequence. Additionally,
²⁷⁷ having a larger s increases the effective time horizon of prediction, for example, for
²⁷⁸ $s = 4$ and $L_{out} = 15$, the model predicts 60 days into the future where every step is
²⁷⁹ a progression of 4 days. We also perform comparisons to identify the architecture's
²⁸⁰ adaptability to missing data by performing imputation, wherein mean data is added
²⁸¹ between missing observations in the case of $s = \{2, 4\}$. This helps simulate cases where
²⁸² field measurements might not be available due to severe weather conditions or resource
²⁸³ constraints, and observe that the prediction model performs sufficiently well under
²⁸⁴ these cases. To verify the effectiveness of the DeepPaSTL architecture, we evaluate our
²⁸⁵ trained model on the simulated dataset in Section 2.2 and on 3D simulation of pasture
²⁸⁶ environments in Gazebo with point cloud measurements as described in Section 2.3.

²⁸⁷ 2.8. Model Training and Evaluation

²⁸⁸ Training and inference are performed for an input and output sequence of 15 steps,
²⁸⁹ each using back-propagation through time (BPTT). However, it is to be noted that due
²⁹⁰ to the dynamic encoder-decoder framework, the architecture can use a variable se-
²⁹¹ quence length during inference. [The complete training and evaluation process is shown](#)
²⁹² [in Figure 5](#).

²⁹³ All models are trained with mean square error loss (MSE). Training is stopped
²⁹⁴ when the validation loss does not improve for 10 consecutive epochs. Learning rates
²⁹⁵ were individually tuned for each network by calculating the steepest gradient on a
²⁹⁶ small sample dataset, although they usually were set to 3.5×10^{-4} . To test the model
²⁹⁷ performance, we use the last two years of data (2008, 2009) for all evaluations in this
²⁹⁸ study. The models are trained on 2x AMD Epyc 7742 CPUs and 8x Nvidia RTX 6000
²⁹⁹ GPUs with PyTorch as its back-end. Training time is generally 15 to 20 hours for 30
³⁰⁰ epochs.

We evaluate the performance of our architecture with the following metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and average standard deviation of all predictions (aSt. Dev.), defined as,

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{B} \times \sum_{i=1}^B (Y_i - \hat{Y}_i)^2}, \\ \text{MAE} &= \frac{1}{B} \times \sum_{i=1}^B |Y_i - \hat{Y}_i|, \\ \text{MAPE}(\%) &= \frac{100}{B} \times \sum_{i=1}^B \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|, \\ \text{aSt. Dev.} &= \sqrt{\frac{1}{B} \sum_{i=1}^B \text{var}(\hat{Y}_i)}, \end{aligned}$$

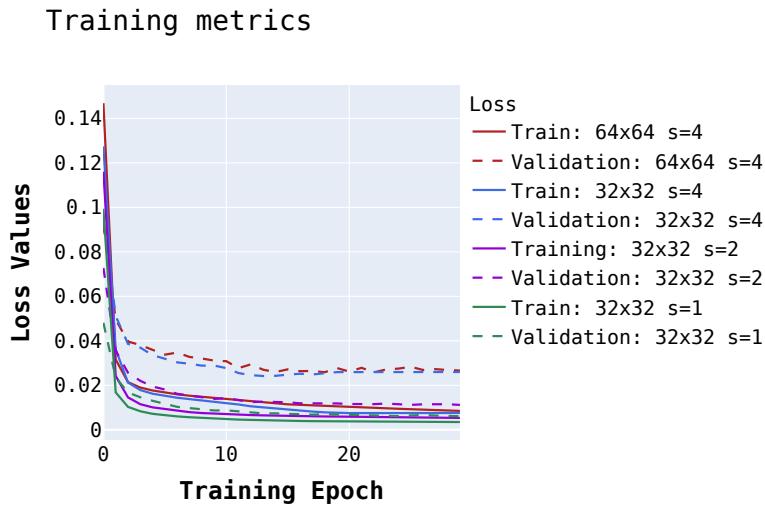


Figure 6. Training and Validation (MSE) Losses for 30 epochs for models trained with $\delta = 64, s = 4$ and $\delta = 32, s = \{1, 2, 4\}$. We observe that the loss rates are correlated to the observation intervals for input sequence. This is attributed to the fact that the architecture's prediction performance is heavily dependent on recognizing temporal patterns in pasture growth due to the highly dynamic nature of pasture evolution.

301 where B is the number of output sequences, Y_i is the ground truth, and \hat{Y}_i is the final
 302 prediction of the neural network after post-processing as described in Section 2.4.

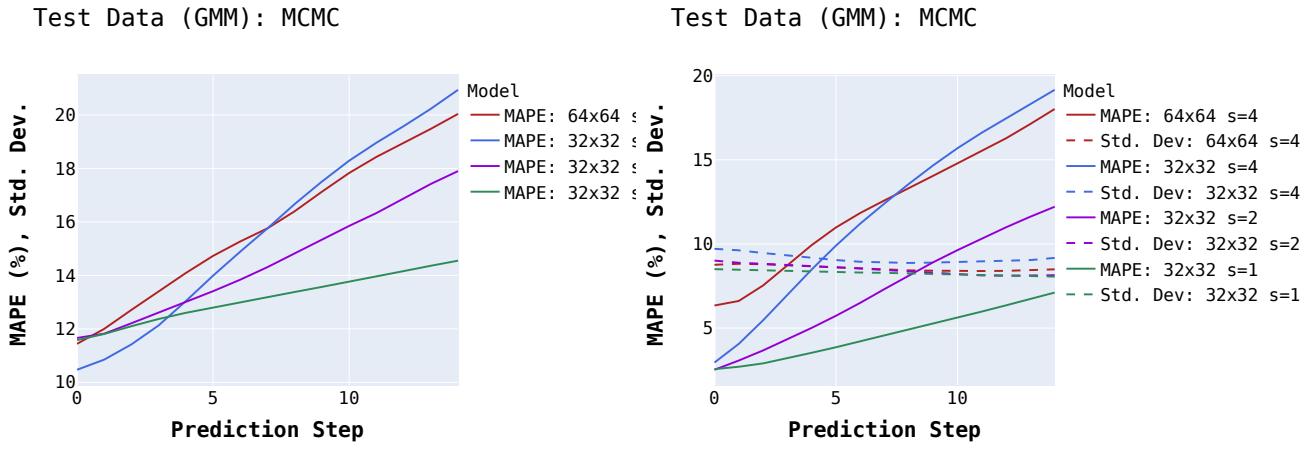
303 3. Results

304 A comparison of the DeepPaSTL architecture over different spatial input sizes is
 305 performed on 3D pastures generated in Gazebo to understand the impact of quantization
 306 and spatial learning of the architecture. We then run our model for different observation
 307 or input intervals, s , to evaluate temporal dependencies. Additionally, we study how
 308 the imputation of missing data can impact the accuracy of the architecture when field
 309 measurements of the pastures are not available on a daily basis. Through our experimen-
 310 tal results conducted both on the simulated data from GMM and the 3D pastures from
 311 Gazebo, we observe the following:

- 312 • DeepPaSTL predictions perform within a 15% error rate for long horizon predictions
 313 up to 60 days in the future, and approximately with a 5% error rate for predictions
 314 closer to its historical data.
- 315 • Allowing the model to have regular observations, i.e., with smaller intervals, is
 316 essential for capturing large dynamic changes in the pasture growth.
- 317 • DeepPaSTL prediction uncertainty increases as the volatility in pasture growth
 318 increases.
- 319 • We show that DeepPaSTL has the capacity to predict and generate future pasture
 320 terrains that replicate the growth and surface characteristics of ground truth data.

321 3.1. Effect of input quantization

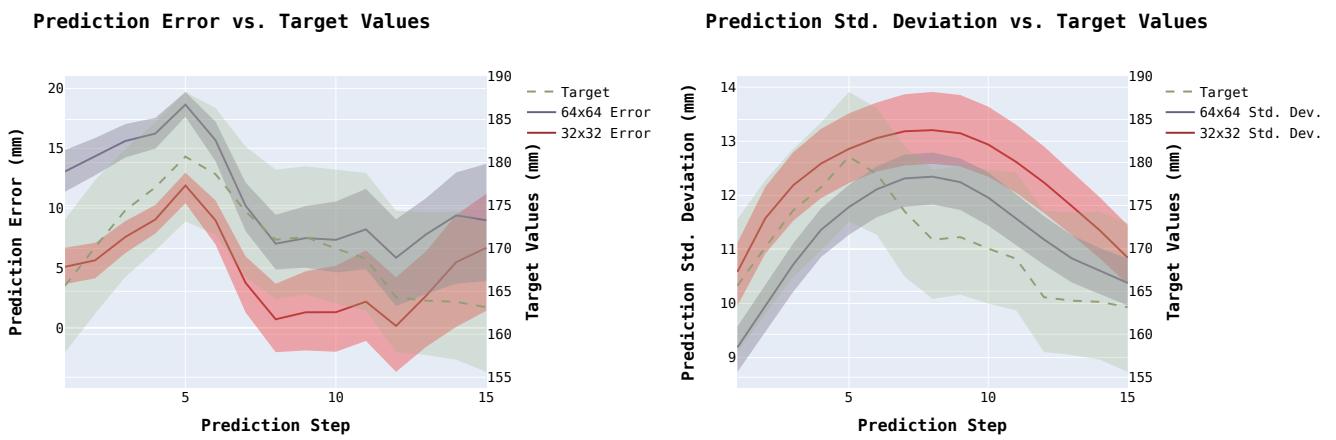
322 We first compare the effect of the input quantization for interval $s = 4$ and $\delta =$
 323 $\{32, 64\}$ with uncertainty estimates as described in Section 2.6. The predicted sward
 324 heights for models trained with $\delta = 64$ showed a slightly lower variability as compared
 325 to the smaller spatial size of the inputs with $\delta = 32$. This larger variance in uncertainty for
 326 lower quantization is to be expected as the model has access to fewer spatial information.
 327 However, we do observe that for the initial time horizon, the lower quantization $\delta = 32$
 328 significantly outperforms the larger $\delta = 64$, Figure 7, while as the number of steps in
 329 the output prediction increases, the error rates for $\delta = \{64, 32\}$ are relatively similar.



(a) Models without MCMC sampling.

(b) Models with MCMC sampling.

Figure 7. Mean absolute percentage error across all the data points consisting of 2 years in the test set (GMM) for (a) models without MCMC sampling, and (b) models with MCMC sampling. MAPE and standard deviation are averaged over all the coordinates of the pasture, and the prediction step for different models. We observe that as s increases the errors increase over the prediction horizon. $s = 4, 2, 1$ effectively correlate to 60, 30, and 15 day prediction horizons.



(a) Error vs Prediction Step

(b) Std. Dev. vs Prediction Step

Figure 8. Prediction (a) error (mm) and (b) standard deviation (mm) bands for 50%, 75%, 25% quantile range for $\delta = \{32, 64\}$, $s = 4$ for a $10m \times 10m$ pasture for $L_{out} = \{1, 6, 11, 15\}$ respectively and ground truth prediction from the 3D Pasture generated in Gazebo depicting the rate of change of pastures over 60 days. We observe that the lower quantization $\delta = 32$ outperforms the larger input quantization over the complete predicted time period.

Table 1: Cumulative accuracy scores averaged for the time period ($L_{in} = L_{out} = 15$). Following models were tested ($\delta = 64, s = 4$), and ($\delta = 32, s = \{1, 2, 4\}$) with and without approximate Bayesian inference (MCMC) with 500 samples and $p^l = 0.4$. Accuracy's are calculated both for the test set from 2008 to 2009, and the 30 days sequence of point cloud measurements from 3D pasture simulated in Gazebo. All values are reported in mm.

Model	Test Dataset (GMM)				3D Pasture (Gazebo)			
	RMSE	MAE	MAPE	aSt. Dev.	RMSE	MAE	MAPE	aSt. Dev.
($\delta = 64, s = 4$) + MCMC	20.02	14.54	12.25	8.55	12.37	11.21	6.49	11.15
($\delta = 32, s = 4$) + MCMC	19.11	13.36	11.79	9.14	7.37	6.33	3.61	12.3
($\delta = 32, s = 2$) + MCMC	11.52	8.13	7.33	8.48	—	—	—	—
($\delta = 32, s = 1$) + MCMC	6.85	5.05	4.63	8.28	—	—	—	—
($\delta = 64, s = 4$)	26.35	20.04	15.84	—	24.91	24.03	14.02	—
($\delta = 32, s = 4$)	24.76	18.81	15.65	—	19.41	18.13	10.6	—
($\delta = 32, s = 2$)	21.74	16.7	14.49	—	—	—	—	—
($\delta = 32, s = 1$)	18.66	14.40	13.15	—	—	—	—	—

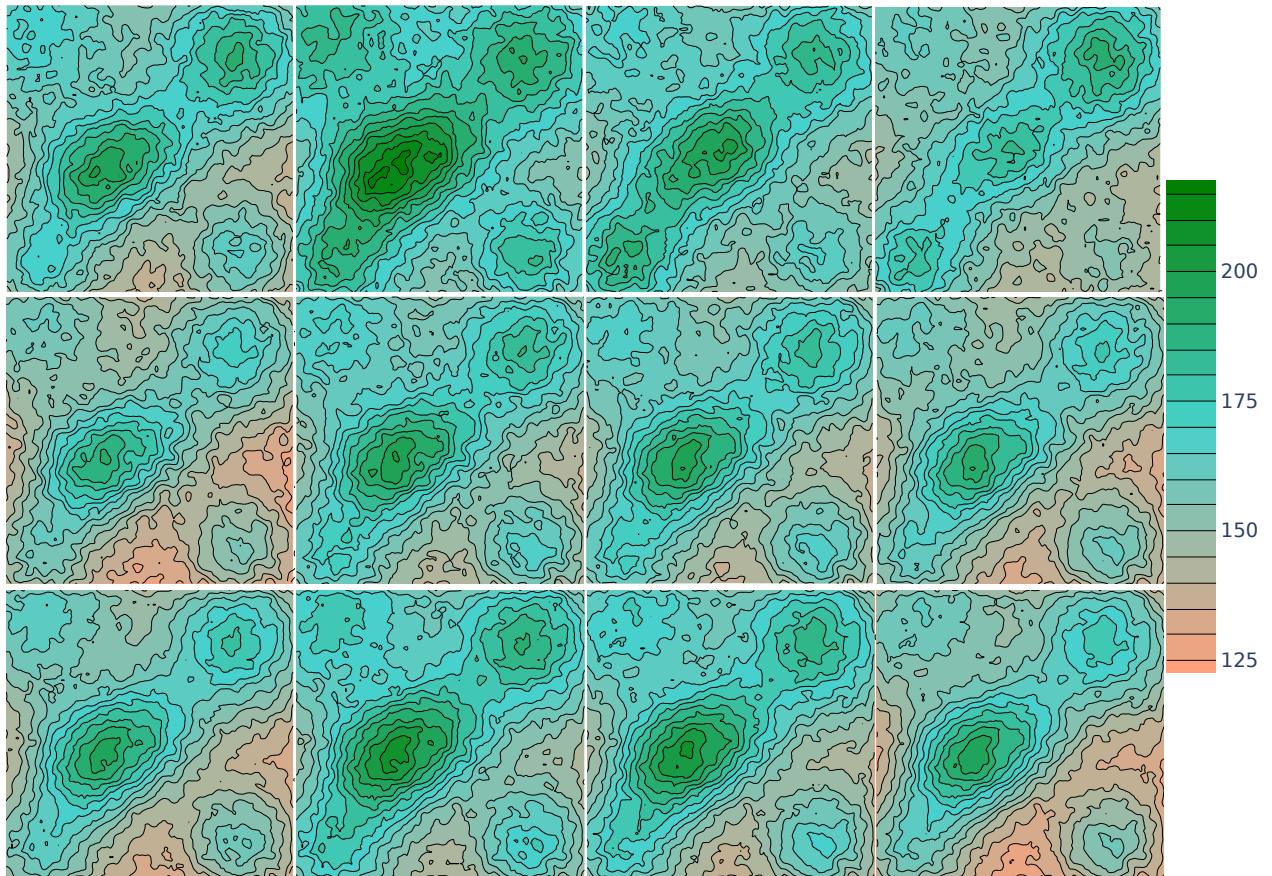


Figure 9. Prediction average height (mm) using approximate Bayesian inference for $\delta = \{32, 64\}, s = 4$ for a $10m \times 10m$ pasture for $L_{out} = \{1, 6, 11, 15\}$ respectively. All values reported are in (mm). Input sequence to the DeepPaSTL network $L_{in} = 15$ is 04/01/2019 to 05/27/2019 and output prediction $L_{out} = 15$ every 4 days. The predictions are 1, 24, 44, 60 days in the future during the peak pasture growth time of 05/31/2019 to 07/26/2019. **Top:** Target values acquired from point cloud measurements with LIDAR of 3D pasture generated in Gazebo. **Middle:** $\delta = 64$ generally underestimates the growth of the pasture resulting in larger errors, however it generally has a better tracking for lower lying areas, or receding pasture heights especially for the longer horizon. **Bottom:** $\delta = 32$ The lower quantization tracks the peaks and troughs of the sward height measurements quite accurately for the near horizon.

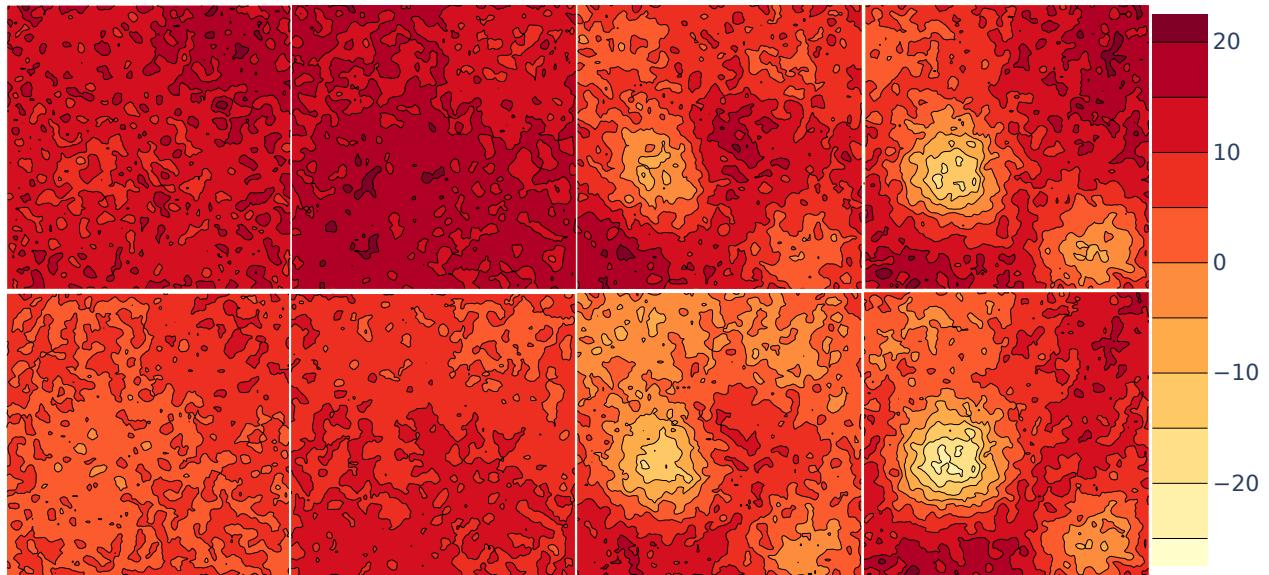


Figure 10. Prediction errors (mm) for $\delta = \{32, 64\}$, $s = 4$ for a $10m \times 10m$ pasture for $L_{out} = \{1, 6, 11, 15\}$ respectively. All values reported are in (mm). Input sequence to the DeepPaSTL network $L_{in} = 15$ is 04/01/2019 to 05/27/2019 and output prediction $L_{out} = 15$ every 4 days. These error maps effectively correlate to 1, 24, 44, 60 days in the future during the peak pasture growth time of 05/31/2019 to 07/26/2019. **Bottom:** $\delta = 32$ We observe the lower quantization of the pasture has a distinct advantage through reduced prediction errors as compared to **Top:** $\delta = 64$ for the same set of inputs.

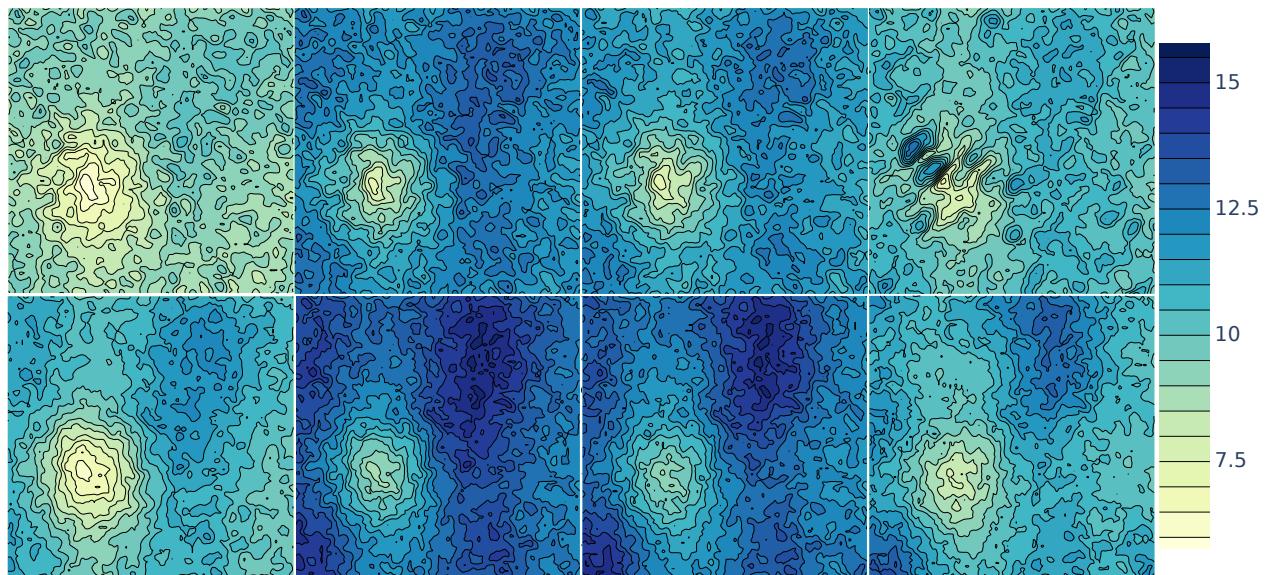


Figure 11. Standard deviation (mm) of predictions for $\delta = \{32, 64\}$, $s = 4$ for a $10m \times 10m$ pasture for $L_{out} = \{1, 6, 11, 15\}$ respectively. All values reported are in (mm). We observe lower uncertainties at the peaks of the pasture due to its lower variability over time. Input sequence to the DeepPaSTL network $L_{in} = 15$ is 04/01/2019 to 05/27/2019 and output prediction $L_{out} = 15$ every 4 days. The uncertainty estimates effectively correlate to 1, 24, 44, 60 days in the future during the peak pasture growth time of 05/31/2019 to 07/26/2019. **Bottom:** $\delta = 32$ We observe the lower quantization of the pasture has higher uncertainty in its prediction when less spatial information is available for processing especially when pasture dynamics are high. **Top:** $\delta = 64$ has smaller prediction uncertainties for the same set of inputs.

330 This is mainly attributed to the fact that the model with large spatial representations
331 has an inherent advantage to perform better in a time period with fast-moving pasture
332 dynamics, due to its extended capacity to learn spatial correlations of the evolving
333 field. However, increasing the spatial size of the architecture makes it harder to train
334 the network effectively to predict changes in the pasture. Pasture maps for the error
335 $Y_i - \hat{Y}_i$ and uncertainty in its prediction are shown in Figure 7, 9, 10, 11. Through
336 our empirical evaluations, we observe that the lower quantization of spatial inputs
337 significantly outperformed, Table 1 and Figure 7, larger spatial input sizes, especially
338 during the first half of the prediction horizon. This can be also be observed in the 3D
339 Gazebo point cloud predictions where pasture growth rates were the highest for the
340 initial time horizon, Figure 8a. Therefore, we use $\delta = 32$ for all future experiments, as
341 the model can always update its predictions over time with new field measurements.

342 3.2. Effect of intervals between observations

343 We evaluate our architecture on varying input and output interval sizes of $s =$
344 $\{1, 2, 4\}$, with a prediction horizon of $\{15, 30, 60\}$ days respectively, and we observe
345 the accuracy of the architecture decreases as the number of intervals between each
346 observation is increased. Training and validation loss for each model are shown in Figure
347 6. Despite the accuracy loss, our model performs with a cumulative 88% accuracy even
348 in the most difficult pasture growth timelines for a 60-day prediction horizon. Trends in
349 pasture growth exhibit a complicated pattern where there exists strong non-linearity in
350 growth pattern and large fluctuations over time. We observe that the accuracy across
351 the prediction horizon averaged over the complete two-year testing dataset decreases
352 drastically when the interval length is increased from 2 to 4, as shown in Table 1 and
353 Figure 7. Moreover, we observe that the error rates follow the dynamic growth pattern
354 of the pasture, where large growth in short periods of time, Figure 8a.

355 3.3. Uncertainty over pasture dynamics

356 Due to the volatile nature of pasture growth, it is imperative for prediction models
357 to be capable of estimating uncertainty. Through stochastic inference by approximate
358 Bayesian methods [26], we observe that the DeepPaSTL architecture has a higher un-
359 certainty in its prediction at regions in the pasture with large growth dynamics Fig 8.
360 The model learns to predict regions with high sward heights quite accurately, as the
361 model inherently captures these strong features within its spatial representations, and
362 consequently we observe very low uncertainty in its prediction at high grass regions
363 within the pasture. This is also partially attributed to the fact that peak pasture growths
364 have a lower growth rate compared to pasture heights that are shorter. However, in the
365 case of $s = 4$ as we move forward in time towards the last prediction step at $i = 8, 10$ i.e.,
366 on the 32nd, and 40th day in the future, we observe that the confidence of the model
367 drops as the time horizon increases, due to heavy pasture growth, and a sparse historical
368 data Figure 8b. It can be clearly observed that the average uncertainty increases, which
369 is further exaggerated by the increased volatility in pasture growth. Moreover, under
370 the approximate Bayesian inference due to repeated sampling and inference, the perfor-
371 mance of the Bayesian DeepPaSTL model substantially outperforms the deterministic
372 single pass inference that is used in standard deep learning methods Figure 7. The
373 MCMC sampling method allows the model to have a 3x improvement over standard
374 single forward pass inferences over the short prediction horizon. It is to be noted that
375 MCMC sampling with $s = \{1, 2\}$ on average has an accuracy that is twice as good as the
376 single forward pass methods. We attribute this improvement to the DeepPaSTLs capacity
377 to accurately model the stochastic dynamics of the pasture by allowing different nodes
378 in the network to dominate in each forward pass. We hypothesize that by prioritizing
379 on each individual node through stochastic sampling allows the model to regenerate
380 precise dynamics of pasture growth by focusing on different factors and representa-
381 tions of the historical observations. However, for long horizon predictions of $s = 4$ the

³⁸² difference in accuracy reduces as prediction steps get close to L_{out} , Figure 7b which is
³⁸³ attributed to lack of observational data. We show the results for prediction performance
³⁸⁴ with and without stochastic inference in Table 1, and Figure 7. Mean predictions for a
³⁸⁵ 60-day horizon, mapped as a 3D field is shown in Figure 12, where the example if an
³⁸⁶ out-of-domain problem, that has not be included in the training data.

³⁸⁷ 3.4. *Imputation of Missing Data*

³⁸⁸ In order to improve the accuracy of the model for long horizon prediction near
³⁸⁹ the 40+ day mark and to address real-world applications that allow a reduction in the
³⁹⁰ frequency of field measurements, we test the accuracy of the network when imputation
³⁹¹ is performed for missing data in the input sequence. We evaluate the performance of
³⁹² the models when under the following conditions: a) When data is missing every other
³⁹³ day, where an observation sequence of interval size $s = 2$ is modified to fit a $s = 1$
³⁹⁴ prediction model using an interpolation of the average growth between the missing data,
³⁹⁵ b) and similarly data available in 4 day intervals has 3 values inserted to predict with
³⁹⁶ $s = 1$ models. We then compare the results to a perfect model where data is available
³⁹⁷ every day for $s = 1$. We show that the network is robust to these events by adapting to
³⁹⁸ the imputed data and manages to predict the future pasture sward heights to high
³⁹⁹ accuracy, Table 2. We observe a modest improvement in the performance of model as the
⁴⁰⁰ prediction network adapts to a gradual change of pasture growth from the interpolated
⁴⁰¹ data. This reinforces our assumption on the robustness of DeepPaSTL architecture and
⁴⁰² allows farm owners and enterprises to expend less resources on daily field measurements,
 saving valuable time and reducing the cost of operations of dairy-farms.

Table 2: We measure the cumulative accuracy of the prediction model under missing observations over 2 years of the test data. We denote x_t as the input that is missing, while Z_t as the available observation. x_t is calculated by fitting a linear curve between the available observations within its interval. Evaluation is done for ($L_{in} = L_{out} = 15$) with ($\delta = 32, s = 1$) with MCMC inference, using 500 samples and $p^l = 0.4$. Accuracy's are calculated for the test set generated with GMM from 2008 to 2009.

⁴⁰³ Imputation	^{(\delta = 32, s = 1) + MCMC}			
	⁴⁰⁴ RMSE	⁴⁰⁵ MAE	⁴⁰⁶ MAPE	⁴⁰⁷ aSt. Dev.
Z_1, Z_2, \dots, Z_{15}	6.85	5.05	4.63	8.28
$Z_1, x_2, Z_3, x_4, \dots, Z_{15}$	5.98	4.29	4.24	8.57
$Z_1, x_2, x_3, Z_4, x_5, x_6, \dots, Z_{15}$	6.04	4.44	4.29	8.73

⁴⁰⁸

⁴⁰⁹ 4. Discussion

⁴¹⁰ This study demonstrates that the DeepPaSTL architecture accurately predicts pre-
⁴¹¹ grazing pasture growths with an average error below 12%, using only the sward height
⁴¹² measurements as its input. The experimental evaluations of this study highlight the
⁴¹³ capability of the DeepPaSTL architecture to implicitly learn the biological dependencies
⁴¹⁴ of pasture growths on climate variables such as precipitation, temperature, soil types,
⁴¹⁵ and pasture management processes among others. DeepPaSTL introduces a novel
⁴¹⁶ direction in pasture predictions by treating spatial measurements as the sole observation
⁴¹⁷ data for forecasting future pasture growths. The advantage of using this approach
⁴¹⁸ enables pasture farms to accurately predict future pasture evolution, even if they are not
⁴¹⁹ equipped to monitor fields on a regular schedule. Our results highlight the practical
⁴²⁰ applicability of our method by depending only on high-resolution spatial mappings
⁴²¹ that can be generated through remote sensing, satellite imagery or UAVs. The proposed
⁴²² methodology in this study also provides a highly scalable prediction methodology that
⁴²³ is adaptable to both small and large pastures.

⁴²⁴ Our results provide several insights on DeepPaSTL's capability of predicting a
⁴²⁵ highly dynamic spatiotemporal pasture over long horizons. Our approach exhibits

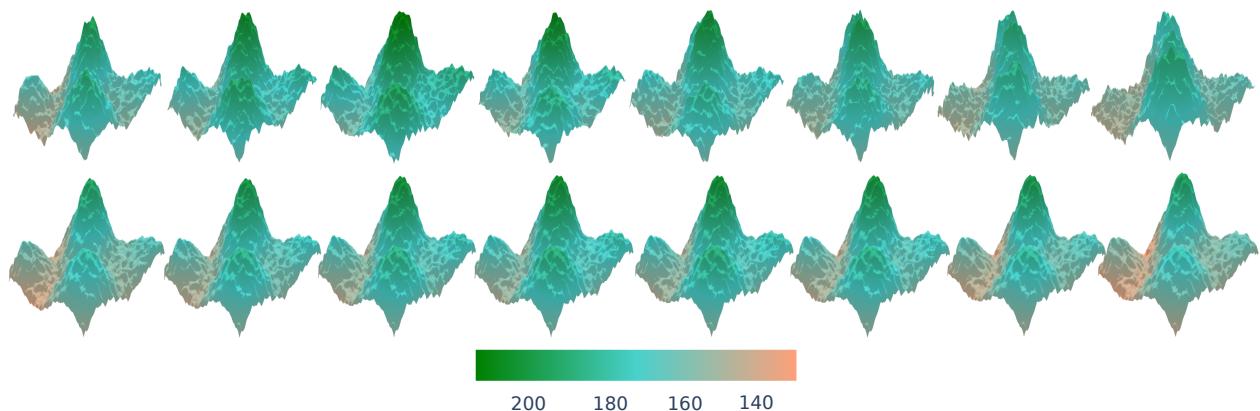


Figure 12. An example of 3D predicted pastures from the Gazebo simulated point cloud measurements for $s = 4$. **Top:** Ground truth measured sward heights (mm) of the pasture. **Left to right** for a 60-day prediction horizon shown for every 8th day. **Bottom** Prediction by a $\delta = 32, s = 4$ model.

421 excellent accuracy where mean errors were within 5% for shorter time intervals s . For
 422 example, mean errors for $s = 1$ were within 5% across 2 years of the testing set, which is
 423 a substantial improvement over larger sequence intervals of $s = 4$ with a cumulative
 424 accuracy of 12%, and a short horizon accuracy for the 20th day to be within 10%.
 425 Moreover, allowing the architecture to perform spatiotemporal predictions over smaller
 426 quantizations eases the prediction and learning burden of the network, further improving
 427 the accuracy. Lower quantizations of the model do not necessarily impact the prediction
 428 process, since inference times are negligible (usually less than an hour) when compared
 429 to pasture growth changes.

430 Bayesian inference, which combines the MCMC sampling to simulate a stochastic
 431 inference of the network, proved to be more robust than standard deep learning inference
 432 methods without uncertainty measurements. Inference through approximate Bayesian
 433 methods enabled the model to predict pasture growths with lower error, and more
 434 importantly, a strong correlation was observed between large errors and uncertainty
 435 in the predictions. The findings were indicative of correlations between the model's
 436 capacity to understand the influence of spatiotemporal evolution through its observed
 437 data and its confidence in predicting large pasture height swings in a short period of
 438 time. The uncertainty is more pronounced when data is sparse especially, as s becomes
 439 larger.

440 The performance of the DeepPaSTL model may also be affected by several other
 441 factors that are not considered in this study due to the lack of available data. We simulate
 442 noise in LIDAR measurements of sward heights in the pasture through a Gaussian
 443 noise. Moreover, we also perform processing over these point cloud measurements
 444 to adapt the data to the neural network. These processes introduce bias and errors
 445 in the final prediction. Our synthetic dataset assumed five different varieties of grass,
 446 however, these might differ across the spatial field and other climatic and local factors,
 447 that would change the dynamics of the input observations. This assumes that owners
 448 and enterprises are capable of adapting and controlling the variety of grass species in
 449 their environment to mitigate the issue of large divergences between training data and
 450 real-world measurements. However, the neural network can always be fine-tuned with
 451 newer observations and datasets to adapt to new pasture environments, and we expect
 452 the impact on the accuracy of the model to be modest.

453 Overall our prediction results from the DeepPaSTL architecture emphasize several
 454 important directions that prediction and planning tools can consider for integration
 455 and future development. First, the DeepPaSTL encoder-decoder architecture presents a
 456 highly flexible tool for predicting pasture heights across varying spatial sizes and tempo-
 457 ral observations. Second, we empirically show that synthetic datasets that are modeled

458 appropriately can be a useful tool to generate training data for deep learning prediction
459 models for pasture growths. Third, the accuracy of the predictions is correlated to the
460 frequency of observations. However, the lack of intermediate field measurements can be
461 mostly mitigated through apt use of data imputation. Finally, to allow deeper insights
462 and increase the generalization power of the architecture, we hope to extend our work
463 to a broader range of applications by including site-specific measurements and other
464 climatic conditions, if available, as part of DeepPaSTL.

465 **5. Conclusion**

466 We prove the capabilities of modern deep learning techniques and algorithms for
467 predicting pre-grazed pasture terrains for both long and short horizon. Through our
468 proposed techniques, we aim to provide an important first step towards applying high
469 resolution prediction methodologies over complete pasture terrains. Our DeepPaSTL
470 modelling is capable of predicting large pastures over long horizons with an adequate
471 degree of accuracy across both short and large pasture forms. As part of future work,
472 we believe DeepPaSTL can be adapted to predict pasture regression due to grazing
473 activities with minimal modifications. Since, DeepPaSTL learns general trends in pasture
474 growth rates, it can be directly applied to learn and predict growth of pastures recovering
475 from grazing. A dual prediction model for recovery and regression of pastures due to
476 grazing can be incorporated as part of planning systems substantially reduced time and
477 resources spent on field measurements. Adoption of these techniques can be accelerated
478 by appropriate modeling of growth patterns of individual sites to generate synthetic
479 historical datasets for DeepPaSTL to perform effectively across varied locations. Since,
480 DeepPaSTL can learn with new data accumulated over months, the model has an inherent
481 capacity to effectively adapt to varying climatic and environmental conditions.

482 **Author Contributions:** Conceptualization, formal analysis, methodology, investigation, software,
483 visualization, writing-original draft, validation, M.R.; Data curation, formal analysis, conceptual-
484 ization, writing - review and editing, J.L.; data curation, formal analysis, visualization, Software,
485 K.S.A.; data curation, S.G.; data curation, H.S.D.; Supervision, project administration, validation,
486 writing - review and editing, funding acquisition, R.K.W.; Supervision, project administration,
487 funding acquisition, B.F.T.; Supervision, project administration, funding acquisition, P.T. [All](#)
488 [authors have read and agreed to the published version of the manuscript.](#)

489 **Funding:** This work was supported by the National Institute of Food and Agriculture under Grant
490 2018-67007-28380.

491 **Institutional Review Board Statement:** Not applicable.

492 **Informed Consent Statement:** Not applicable.

493 **Computer Code and Software:** Our DeepPaSTL code is open sourced and available at:
494 <https://github.com/caslab-vt/DeepPasTL>

495 **Conflicts of Interest:** The authors declare no conflict of interest.

496 **Abbreviations**

497 The following abbreviations are used in this manuscript:

498

APSIM	Agricultural Production Systems sIMulator
BiConvLSTM	Bidirectional Convolutional Long Short Term Memory
BPTT	Back Propagation Through Time
CNN	Convolution Neural Network
ConvLSTM	Convolutional Long Short Term Memory
DeepPaSTL	Deep Pasture SpatioTemporal Learning
DOAJ	Directory of Open Access Journals
GMM	Gaussian Mixture Model
LIDAR	Light Detection and Ranging
⁴⁹⁹ LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MaxPool	Maximum Pooling
MCMC	Markov Chain Monte Carlo
MDPI	Multidisciplinary Digital Publishing Institute
MSE	Mean Squared Error
UAV	Unmanned Aerial Vehicle
VRAM	Video Random Access Memory

References

1. Garcia, S.C.; Clark, C.E.; Kerrisk, K.L.; Islam, M.R.; Fariña, S.; Evans, J. Gaps and variability in pasture utilisation in Australian pasture-based dairy systems **2020**.
2. Sala, O.E.; Paruelo, J.M.; others. Ecosystem services in grasslands. *Nature's services: Societal dependence on natural ecosystems* **1997**, pp. 237–251.
3. Insua, J.R.; Utsumi, S.A.; Basso, B. Estimation of spatial and temporal variability of pasture growth and digestibility in grazing rotations coupling unmanned aerial vehicle (UAV) with crop simulation models. *PloS one* **2019**, *14*, e0212773.
4. Fulkerson, W.; McKean, K.; Nandra, K.; Barchia, I. Benefits of accurately allocating feed on a daily basis to dairy cows grazing pasture. *Australian Journal of Experimental Agriculture* **2005**, *45*, 331–336.
5. Tao, F.; Yokozawa, M.; Zhang, Z. Modelling the impacts of weather and climate variability on crop productivity over a large area: a new process-based model development, optimization, and uncertainties analysis. *agricultural and forest meteorology* **2009**, *149*, 831–850.
6. Iizumi, T.; Yokozawa, M.; Nishimori, M. Parameter estimation and uncertainty analysis of a large-scale crop model for paddy rice: Application of a Bayesian approach. *Agricultural and forest meteorology* **2009**, *149*, 333–348.
7. Lobell, D.B.; Burke, M.B. On the use of statistical models to predict crop yield responses to climate change. *Agricultural and forest meteorology* **2010**, *150*, 1443–1452.
8. De Rosa, D.; Basso, B.; Fasiolo, M.; Friedl, J.; Fulkerson, B.; Grace, P.R.; Rowlings, D.W. Predicting pasture biomass using a statistical model and machine learning algorithm implemented with remotely sensed imagery. *Computers and Electronics in Agriculture* **2021**, *180*, 105880.
9. Chen, Y.; Guerschman, J.; Shendryk, Y.; Henry, D.; Harrison, M.T. Estimating Pasture Biomass Using Sentinel-2 Imagery and Machine Learning. *Remote Sensing* **2021**, *13*, 603.
10. Neal, R.M. *Bayesian learning for neural networks*; Vol. 118, Springer Science & Business Media, 2012.
11. Fukushima, K.; Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*; Springer, 1982; pp. 267–285.
12. LeCun, Y.; Haffner, P.; Bottou, L.; Bengio, Y. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*; Springer, 1999; pp. 319–345.
13. Lin, Z.; Li, M.; Zheng, Z.; Cheng, Y.; Yuan, C. Self-attention convlstm for spatiotemporal prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 11531–11538.
14. Xu, Y.; Gao, L.; Tian, K.; Zhou, S.; Sun, H. Non-local convlstm for video compression artifact reduction. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7043–7052.
15. Azad, R.; Asadi-Aghbolaghi, M.; Fathy, M.; Escalera, S. Bi-directional ConvLSTM U-Net with densely connected convolutions. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.
16. Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; Huang, T. Youtube-vos: Sequence-to-sequence video object segmentation. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 585–601.
17. Oliu, M.; Selva, J.; Escalera, S. Folded recurrent neural networks for future video prediction. Proceedings of the European Conference on Computer Vision, 2018, pp. 716–731.
18. Michalski, V.; Memisevic, R.; Konda, K. Modeling deep temporal dependencies with recurrent grammar cells. 2014, Vol. 27, pp. 1925–1933.
19. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using lstms. Proceedings of the International conference on machine learning. PMLR, 2015, pp. 843–852.

20. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Proceedings of the Advances in neural information processing systems, 2015, pp. 802–810.
21. Lotter, W.; Kreiman, G.; Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104* **2016**.
22. Wang, Y.; Long, M.; Wang, J.; Gao, Z.; Yu, P.S. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. Proceedings of the International Conference on Neural Information Processing Systems, 2017, pp. 879–888.
23. Wang, Y.; Gao, Z.; Long, M.; Wang, J.; Philip, S.Y. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. Proceedings of the International Conference on Machine Learning. PMLR, 2018, pp. 5123–5132.
24. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). IEEE, Vol. 3, pp. 2149–2154.
25. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. Proceedings of the Advances in neural information processing systems, 2014, pp. 3104–3112.
26. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. Proceedings of the International conference on machine learning. PMLR, 2016, pp. 1050–1059.
27. Archontoulis, S.V.; Miguez, F.E.; Moore, K.J. Evaluating APSIM Maize, Soil Water, Soil Nitrogen, Manure, and Soil Temperature Modules in the Midwestern United States. *Agronomy Journal* **2014**, *106*, 1025–1040.
28. Li, F.; Newton, P.; Lieffering, M. Testing simulations of intra- and inter-annual variation in the plant production response to elevated CO₂ against measurements from an 11-year FACE experiment on grazed pasture. *Global change biology* **2013**, *20*.
29. Liu, J.; Williams, R.K. Submodular optimization for coupled task allocation and intermittent deployment problems. *IEEE Robotics and Automation Letters* **2019**, *4*, 3169–3176.
30. Liu, J.; Williams, R.K. Monitoring over the long term: Intermittent deployment and sensing strategies for multi-robot teams. Proceedings of the IEEE International Conference on Robotics and Automation, 2020, pp. 7733–7739.
31. Innamorati, C.; Ritschel, T.; Weyrich, T.; Mitra, N.J. Learning on the edge: Explicit boundary handling in cnns. *arXiv preprint arXiv:1805.03106* **2018**.
32. Innamorati, C.; Ritschel, T.; Weyrich, T.; Mitra, N.J. Learning on the edge: Investigating boundary filters in cnns. *International Journal of Computer Vision* **2019**, pp. 1–10.
33. Hashemi, M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. *Journal of Big Data* **2019**, *6*, 1–13.
34. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. Proceedings of the International Conference on Engineering and Technology, 2017, pp. 1–6.
35. Tang, H.; Ortis, A.; Battiatto, S. The impact of padding on image classification by using pre-trained convolutional neural networks. Proceedings of the International Conference on Image Analysis and Processing. Springer, 2019, pp. 337–344.
36. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
37. Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* **2013**.
38. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* **2014**.
39. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.
40. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.
41. Ranzato, M.; Szlam, A.; Bruna, J.; Mathieu, M.; Collobert, R.; Chopra, S. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604* **2014**.
42. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. Proceedings of the International conference on machine learning. PMLR, 2015, pp. 2048–2057.
43. Horn, R.A. The hadamard product. Proc. Symp. Appl. Math, 1990, Vol. 40, pp. 87–169.
44. Song, H.; Wang, W.; Zhao, S.; Shen, J.; Lam, K.M. Pyramid dilated deeper convlstm for video salient object detection. Proceedings of the European conference on computer vision (ECCV), 2018, pp. 715–731.
45. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* **2005**, *18*, 602–610.
46. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
47. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. Proceedings of the International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
48. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-based prediction model for spatio-temporal data. Proceedings of the International conference on advances in geographic information systems, 2016, pp. 1–4.
49. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. *arXiv preprint arXiv:1507.06228* **2015**.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

51. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. Proceedings of the European conference on computer vision. Springer, 2016, pp. 630–645.
52. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
53. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the International conference on machine learning. PMLR, 2015, pp. 448–456.