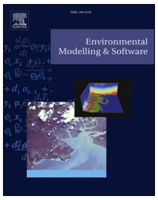
Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



Attention-based convolutional capsules for evapotranspiration estimation at scale

Samuel Armstrong [a,*], Paahuni Khandelwal [a], Dhruv Padalia [a], Gabriel Senay [f,b], Darin Schulte [c], Allan Andales [d], F. Jay Breidt [e], Shrideep Pallickara [a], Sangmi Lee Pallickara [a]

[a] *Department of Computer Science, Colorado State University, USA*
[b] *Department of Ecosystem Science and Sustainability, Colorado State University, USA*
[c] *Natural Resource Ecology Laboratory, Colorado State University, USA*
[d] *Department of Soil and Crop Science, Colorado State University, USA*
[e] *Department of Statistics, Colorado State University, USA*
[f] *U.S. Geological Survey Earth Resources Observation and Science Center, Fort Collins, Colorado, USA*

ARTICLE INFO

Keywords:
Evapotranspiration
Satellite imagery
Neural networks
Capsule networks
Residual learning
Attention-based learning

ABSTRACT

Evapotranspiration (ET) measures the amount of water lost from the Earth's surface to the atmosphere and is an integral metric for both agricultural and environmental sciences. Understanding and quantifying ET is critical for achieving effective management of freshwater and irrigation systems. However, current ET estimation models suffer from a trade-off between accuracy and spatial coverage. In this study, we introduce our model Quench, a neural network architecture that achieves highly-accurate ET estimates over large continuous spatial extents. Quench uses our novel Attention-Based Convolutional Capsule for its neural network layers to identify areas of focus and efficiently extract ET information from satellite imagery. Benchmarks that profile our model's performance show substantive improvements in accuracy, with up to 128% increase in accuracy compared to traditional convolutional-based and process-based models. Quench also demonstrates consistent model performance over high geospatial variability and a diverse array of regions, seasons, climates, and vegetations.

1. Introduction

Watering schedules for farms are informed by irrigation models. The precision of these models dictate the farm's ability to conserve water resources, while keeping their crops healthy. The key metric used by these irrigation models is evapotranspiration more commonly known as ET (Monteith, 1965). ET is the amount of water lost from the Earth's surface to the atmosphere and is measured in millimeters of water lost per day. ET is a combination of two separate processes: evaporation and transpiration. Evaporation is when water on the soil's surface changes from a liquid to a gas due to heat. Transpiration is when water inside of a plant is exhaled from the plants' stomata as a gas. If we can accurately estimate the rate of ET, we can then determine how much water an irrigation system needs at any given time.

There are various instruments and methods available for measuring ET. A Lysimeter is a high-precision (albeit expensive and cumbersome) equipment for measuring ET (Allan et al., 2018). Another high-precision ET measuring device, the Flux Tower, has been widely accepted and is a less-costly solution for measuring ET (Liang et al., 2012). However, both of these direct on-site measurements of ET are challenging and/or prohibitively expensive. Also due to the nature of on-site data, which only captures information from the surrounding area, instruments like Lysimeters and Flux Towers cannot measure ET across large continuous extents (Senay, 2008). As a result, a rich body of work focusing on modeling-based approaches for estimating ET at scale has been produced (Gowda et al., 2007; McShane and Driscoll Roy, 2017). Recent advances in computational resources, satellite imagery, and geospatial analysis, have allowed these modeling efforts to expand to large continuous extents (Casu et al., 2017). Machine learning models in particular have been able to achieve highly-accurate ET estimations at scale. These machine learning models, such as CNNs (Convolutional Neural Networks), achieve this by capturing dynamic relationships

[*] Corresponding author.
E-mail addresses: Sam.Armstrong@colostate.edu (S. Armstrong), Paahuni.Khandelwal@colostate.edu (P. Khandelwal), Dhruv.Padalia@colostate.edu (D. Padalia), Senay@usgs.gov (G. Senay), Darin.Schulte@colostate.edu (D. Schulte), Allan.Andales@colostate.edu (A. Andales), FJay.Breidt@colostate.edu (F.J. Breidt), Shrideep.Pallickara@colostate.edu (S. Pallickara), Sangmi.Pallickara@colostate.edu (S.L. Pallickara).

https://doi.org/10.1016/j.envsoft.2022.105366
Received 11 February 2022; Accepted 2 March 2022
Available online 25 April 2022

between remote sensing data and on-site ET measurements (Lecun et al., 2000). Remote sensing data is information captured from a distance (e.g., satellite imagery) that covers large contiguous extents. Using remote sensing data, machine learning models are able to estimate ET across large continuous areas with precision accuracy. However, machine learning models often encounter challenges in their ability to generalize and real-world applicability (Karpatne et al., 2017). Despite recent advances in machine learning and the large amount of available satellite imagery, there are still significant challenges in leveraging ET estimation at scale. This is mostly due to inconsistencies in these models' ability to generalize, especially over large continuous areas that have a high amount of geospatial variability.

To overcome the aforementioned challenges, we propose Quench, a model that provides highly-accurate ET estimates across a diverse range of regional, seasonal, climatic, and vegetative conditions. At a high-level, Quench is a neural network architecture that uses our novel neural network layer, the AttnConvCap (Attention-Based Convolutional Capsule). By leveraging the AttnConvCap, Quench accurately captures dynamic relationships between the surrounding area's environmental conditions and on-site ET measurements, resulting in highly-accurate ET estimates over large continuous extents. To address issues with generalization, Quench incorporates the process-based model SSEBop (operational Simplified Surface Energy Balance) (Gabriel et al., 2011) into its neural network architecture. This improves the model's ability to generalize and addresses the issues that stem from the geospatial sparsity of the on-site ET measurements in our dataset.

1.1. Research questions

In this study, we investigate the following research questions:

RQ1) ET-related data derived from remote sensing observations, have mismatching spatiotemporal resolutions and coverages. How can we integrate this data into a single dataset that captures the fine-grained environmental, meteorological, and agricultural conditions of the areas surrounding our ET measurement sites?

RQ2) Many of the existing machine learning ET models have encountered challenges with consistency in their model's performance stemming from a lack of geospatial variability. How can we construct a model that performs uniformly over large continuous extents that have high amounts of geospatial variability while achieving state-of-the-art accuracy?

RQ3) Reliable ET observations are a critical part of our training data. However, there are only highly-sparse ET measurements available across the continental U.S. How can we generate an ET estimation model that provides a consistent and uniform ability to generalize, despite the sparsity of these ET measurements in the training data?

1.2. Approach **overview**

Quench aims to generate highly-accurate ET estimates that are also resilient to the high amounts of geospatial variability observed over large geospatial extents. Our model factors in environmental, meteorological, and agricultural remote sensing data from satellite imagery (environmental data source), meteorological models, and agricultural databases. From the satellite imagery we use both the visible (e.g., *red*, *green*, and *blue*) and non-visible bands (e.g., *thermal infrared*) from the Landsat 8 Satellite (see Section III). Also, our model retrieves gridded meteorological data from GRIDMET (the University of Idaho Gridded Surface Meteorological Dataset) (Abatzoglou, 2011). The agricultural data we use includes the *Clim* (climate) and *Veg* (vegetation) type, and *Date* (temporal metadata).

Quench uses a Deep Capsule Network to encode and extract information from the satellite imagery. To achieve highly-accurate ET estimates across large continuous extents with significant geospatial variability, we propose an approach that uses our novel AttConvCap, as the neural network layer for Quench's Deep Capsule Network. Capsule-based computation is less susceptible to information loss than traditional convolutional layers and are more effective at identifying objects with varying positions and orientations (Sabour et al., 2017). By using capsule-based computation, Quench is able to capture the interactions between the surrounding area's environmental conditions and ET more effectively. In addition to the Deep Capsule Network, Quench utilizes the process-based model, SSEBop, to overcome the sparsity of the on-site ET measurements in our training data. By utilizing the SSEBop model's output in Quench's neural network architecture, we improve Quench's ability to generalize significantly, while also preserving the high accuracy of the Deep Capsule Network with its AttnConvCap layers.

We have evaluated Quench's performance using k-fold cross validation (Raschka, 2018). Our model demonstrates a MAE (Mean Absolute Error) of 0.4577 mm on average, which represents a 128% improvement in accuracy compared to both the CNN and SSEBop models. We performed a variogram analysis to evaluate Quench's performance over a diverse array of locations and regions. Quench demonstrated highly-consistent model performance across the continental U.S. The semivariance of Quench's accuracy was 0.06 mm^2 for up to 50° of latitude/longitude. In our model analyses (see Section V) we show that Quench also performs uniformly over a diverse array of seasons, climates, and vegetations, indicating promising results for Quench's ability to generalize over large spatial extents.

1.3. Paper contributions

In this study, we propose Quench, a highly-accurate ET estimation model with strong resiliency to geospatial variability. Our specific contributions include:

C1) We have designed and developed Quench, an ET estimation model that uses both a SSEBop model and Deep Capsule Network with AttnConvCap layers in its neural network architecture. Quench effectively captures interactions between environmental, meteorological, and agricultural conditions and local ET measurements to generate highly-accurate ET estimates at scale. Our model can generate ET estimates for any given location across the continental U.S. with high accuracy.

C2) Quench's neural network architecture incorporates the process-based SSEBop model, to address the sparsity of ET measurements sites in the training data. As evidenced by our benchmarks, utilizing the SSEBop model increases Quench's accuracy and ability to generalize.

C3) This study uses three years of remote sensing observations from across the continental U.S. that are sourced from the Landsat 8 Satellite, the AmeriFlux Network, and GRIDMET. After collecting the data we integrate and process it using interpolation, encoding, and normalization methods.

1.4. Paper organization

We present our related work in Section II. In Section III, we describe the data wrangling operations for our ET dataset. Section IV describes our model's methodology. The system benchmarks for Quench are presented in Section V. Lastly, we discuss our conclusions of Quench in Section VI.

2. Related work

There have been many different types of neural network models used to estimate ET in recent years. In the paper (Adeloye et al., 2012) the authors use SOMs (Self Organizing Maps), an unsupervised neural network model, to estimate ET. SOMs train their network weights to group inputted data points to neurons with similar feature values (e.g., humidity, temperature, ET). Once trained the SOM can then be used to group new, unseen data to neurons with similar features values. The SOM is able to group this new input data even if some of its feature values are missing. Once grouped to neurons with similar feature values, the new data replaces its missing values with the neurons' corresponding feature values. Because the SOM can fill in these missing feature values it can now nowcast ET as long as some of the other feature values are available. Quench's supervised learning differs from the SOM's unsupervised learning in that it has data grouped into input (e.g., humidity, temperature) and target (ET) data. The SOM on the other hand can estimate a value for any feature value that is missing from the data, not just the target data. This unsupervised learning approach works quite well in (Adeloye et al., 2012) and they are able to accurately estimate ET from 1-dimensional meteorological data.

Another type of neural network used to estimate ET is the RBFN (Radial Basis Function Network) (Panda et al., 2019). Like Quench, the RBFN uses supervised training to train its ET model on remotely sensed satellite imagery. However, like the SOM, the RBFN has neurons that are made up of feature values that are representative of the training data. The RBFN's input data is first inputted into these RBF neurons and the similarity between the input data and RBF neuron's feature values are measured. This measured similarity is then inputted into a neural network layer, which outputs the estimated ET. The data used to train the SOM and RBFN models are point-based and estimate ET using 1D data for a singular point. Quench is an area-based model and extracts spatial patterns from 2D satellite imagery to estimate ET for an area consisting of many adjacent points.

Similar to Quench, the area-based U-Net model in the paper (Sadeghi et al., 2020) is used to nowcast precipitation (instead of evapotranspiration) using satellite imagery. This U-Net model is an image-to-image mapper and maps the inputted satellite imagery to an outputted precipitation map of the same size and shape. Because the U-Net model is analyzing an entire area, instead of a single point, it can utilize spatial patterns found in the surrounding area to inform its precipitation estimates. In the paper (Sadeghi et al., 2020), the U-Net model was better able to estimate precipitation, especially outlier severe weather events such as Hurricane Harvey, than traditional process-based models. We posit that our Quench model's hybrid neural network and process-based approach is also able to achieve highly-accurate estimates due to its neural network model, while also maintaining highly-consistent estimates due to its process-based model.

3. Dataset

Training Quench involves three different types of data: environmental, meteorological, and agricultural. We source this data from the following publicly available datasets: the Landsat 8 Satellite, the AmeriFlux Network, GRIDMET, and the Colorado State University Arkansas Valley Research Center.

3.1. AmeriFlux

The AmeriFlux Network is a network of research sites that measures changes in the atmosphere. Started in 1996, this relatively large network contains over one hundred sites that can be found throughout North and South America. These sites collect on-site data from the surrounding area, such as changes in the amount of water and carbon dioxide in the atmosphere. The metric we will be using from these sites is *Latent Heat Flux*, which is the measurement of heat associated with evaporation and transpiration lost from the Earth's surface to the atmosphere. By using the Eddy Covariance Model (Liang et al., 2012), *Latent Heat Flux* can be used to directly measure the amount of water lost from the Earth's surface to the atmosphere. Because it is directly measuring the amount of water lost, the Eddy Covariance Model is a highly-accurate way to measure ET.

Our ET dataset covers a three-year period from 2015 to 2017 in which only 54 AmeriFlux Network sites are available (Hobbie et al., 2020; Anderson, 2020; Anderson ., 2020, 2020; Hollinger and Richardson, 2019; Barron-Gafford, 2021; Brunsell, 2020a), (Brunsell, 2020b, 2021; Desai, 2021a, 2021b, 2021c, 2021d), (Perez-Ruiz and Vivoni, 2020; Green and Kelsey, 2020; Foster, 2021; Giblin, 2021; Ross Hinkle, 2019; NEON (National Ecological Observatory Network), 2021a), (NEON (National Ecological Observatory Network), 2021b; NEON (National Ecological Observatory Network), 2021c; NEON (National Ecological Observatory Network), 2021d; NEON (National Ecological Observatory Network), 2021e; NEON (National Ecological Observatory Network), 2021f), (NEON (National Ecological Observatory Network), 2021g; NEON (National Ecological Observatory Network), 2021h; NEON (National Ecological Observatory Network), 2021I; NEON (National Ecological Observatory Network), 2021J; NEON (National Ecological Observatory Network), 2021K), (NEON (National Ecological Observatory Network), 2021L; NEON (National Ecological Observatory Network), 2021M; NEON (National Ecological Observatory Network), 2021N; NEON (National Ecological Observatory Network), 2021O; NEON (National Ecological Observatory Network), 2021P), (NEON (National Ecological Observatory Network), 2021Q; NEON (National Ecological Observatory Network), 2021R; NEON (National Ecological Observatory Network), 2021S; NEON (National Ecological Observatory Network), 2021T; NEON (National Ecological Observatory Network), 2021U), (NEON (National Ecological Observatory Network), 2021V; NEON (National Ecological Observatory Network), 2021W; NEON (National Ecological Observatory Network), 2021X; Olson, 2021), (Scott, 2021; Russell, 2020, 2021a, 2021b, 2021c), (Silveira, 2021; Starr, 2021a, 2021b, 2021c; Craig, 2021; Vivoni, 2020)]. Each of these sites have a Flux Tower, which is an atmospheric measuring device that measures *Latent Heat Flux* from the surrounding area every 30 min. The Eddy Covariance Model takes all 48 of these *Latent Heat Flux* measurements taken throughout the day and converts them to *Daily ET*. The Eddy Covariance Model does this by first averaging these *Latent Heat Flux* measurements and then converting the average to millimeters of water lost throughout the day. *Latent Heat Flux* is measured in watts per square meter (W/m^2), where 28.356 W/m^2 is equal to 1 mm of water lost through ET (Senay, 2008). This process can be seen in Formula 1. The outputted *Daily ET* value is highly-accurate, but spatially sparse due to the nature of on-site data. A Flux Tower accurately measures approximately 500 m^2 of the area surrounding it.

$$Daily\ ET = \frac{1}{48}\sum_{n=1}^{48} Latent\ H\ eat\ Flux \times \frac{1}{28.356} \tag{1}$$

3.2. Lysimeter

Lysimeters are soil measurement devices comprised of large metal tanks that are buried in the ground and filled with soil. Underneath the tanks are sensors that measure changes in the soil's mass. Using these measurements a Lysimeter is able to calculate the amount of water gained or lost in the soil throughout the day. This is a highly-accurate way to measure ET because it is directly measuring the amount of water lost from the soil through evaporation and transpiration. Lysimeters measure the soil's mass every 15 min and calculate the *Daily ET* based on these 96 measurements (Allan et al., 2018). Like Flux Towers, Lysimeters suffer from a sparse spatial coverage due to the nature of on-site data. Also like Flux Towers, Lysimeters can accurately measure the ET for the 500 m^2 of area surrounding the site. However, unlike the AmeriFlux Network, there is not a robust data repository of Lysimeters that cover a large range of areas. Instead, Lysimeter data is confined to small areas and is usually gathered by independent groups and organizations, making collecting this data difficult (Allan et al., 2018). Homogenizing this data is also difficult due to the fact that these sites often use different variations of equipment and configurations for their Lysimeters. We have taken the ET measurements from the Colorado State University Arkansas Valley Research Center's Lysimeter site near Rocky Ford, Colorado. We use this Lysimeter's ET measurements alongside the ET measurements we gathered from the AmeriFlux Network's Flux Towers as our ET dataset's target data.

3.3. Landsat 8

The Landsat 8 Satellite was launched in a joint effort between NASA (National Aeronautics and Space Administration) and USGS (United States Geological Survey). It orbits the entire earth approximately every two weeks. Unfortunately this low temporal resolution can be further exacerbated by cloud coverage in the satellite imagery. Roughly two-thirds of the Earth's surface is covered with clouds at any given moment (King et al., 2013). Cloud coverage in satellite imagery can create a lot of noise, often to the point where it causes inaccurate results (especially in machine learning models). We collected the bi-weekly satellite imagery for each of our 54 AmeriFlux sites and Lysimeter site over a three year period (Jan. 1, 2015 to Dec. 31, 2017). We then removed any satellite images with cloud coverage obscuring a Flux Tower or Lysimeter site from our dataset. After doing this, we were left with 955 satellite images. Because these Landsat 8 images are remote sensing data, they have a large, fine-grained spatial coverage, unlike the on-site data from the Flux Tower and Lysimeter sites. Each point of the satellite imagery represents an area of 30 m^2. In order to match the Flux Tower and Lysimeters 500 m^2 spatial coverage our satellite images have a resolution of 16pixels^2, which represent an area of 480 m^2. These Landsat 8 images consist of a variety of different bands. Bands measure an image in different ranges of frequency along the electromagnetic spectrum. The eleven Landsat 8 bands we use are: *Coastal Aerosol*, *Red*, *Green*, *Blue*, *Near Infrared*, *Shortwave Infrared 1*, *Shortwave Infrared 2*, *Panchromatic*, *Cirrus*, *Thermal Infrared 1*, and *Thermal Infrared 2*. Also gathered from the Landsat 8's satellite imagery are the images corresponding *Lat* (Latitude), *Lon* (Longitude), and *Elev* (Elevation). These bands represent the surrounding area's environmental conditions, which are highly-influential to the area's ET. For example, temperature heavily influences ET and the *Thermal Infrared 1* and *Thermal Infrared 2* bands give us a detailed image of the *LST* (Land Surface Temperature) of the surrounding area.

3.4. GRIDMET

GRIDMET is a model that outputs gridded values of meteorological variables. The variables we use for training Quench are: *ETr* (Reference Evapotranspiration), *SpH* (Humidity), *SRad* (Solar Radiation), *TMin* (Minimum Daily Temperature), and *TMax* (Maximum Daily Temperature). The grids of these variables are calculated daily and cover the entire continental U.S. Because GRIDMET is remote sensing data, it has a large spatial coverage. Each point of GRIDMET covers an area of 4 Km^2, which is relatively coarse compared to the Landsat 8's 30 m^2 resolution. To compensate for this low-spatial resolution we use bilinear-interpolation, which increases the GRIDMET data's resolution to 30 m^2 to match the Landsat 8's resolution.

3.5. Agricultural data

In addition to the previously mentioned environmental and meteorologi-cal data, we have also collected agricultural data for training Quench. This data includes: *Clim*, *Veg*, and *Date*. Like *Lat*, *Lon*, and *Elev*, *Clim* and *Veg* are static and do not change over time. This makes gathering them relatively easy. There are multiple ways this can be done e.g., map APIs and agricultural databases. However, for our case both the Flux Tower and Lysimeter sites collect this agricultural data alongside their on-site data, so we have sourced this agricultural data from them. Once we have gathered all the data for our ET dataset, we must now encode and normalize it before we can begin training Quench.

3.6. Encoding input data (RQ1)

Encoding variables is necessary for machine learning model inputs when the variables are not accurately represented as is. The continuous variables from our dataset that require encoding are: *Lat*, *Lon*, and *Date*, while the categorical variables that require encoding are: *Clim* and *Veg*. We input *Lat* and *Lon* into Quench, because different coordinates will have different environmental factors that affect ET. One example of this is areas with a higher *Lat* (northern U.S.) will have colder environments that lose less water through ET than warmer areas, which have a lower *Lat* (southern U.S.). Another example is areas with a higher *Lon* (eastern U.S.) will have more humid climates and lose less water through ET compared to dryer areas with a lower *Lon* (western U.S.). To encode and normalize *Lat* and *Lon* we first convert them into spherical coordinates. We do this because 2D (x,y) coordinates do not properly represent the distance between two points on the surface of a sphere. We solve this by adding an extra dimension, converting *Lat* and *Lon* to 3D Cartesian coordinates (x,y,z). This process is known as cyclical feature encoding and can be seen in Formulas 2, 3, and 4 (Adams and Vamplew, 1998).

After applying cyclical feature encoding to each pair of (*Lat*, *Lon*) coordinates, the outputted (x,y,z) coordinates are ranged from −1 to 1.

$$X = Sin((\pi / 2 - (Lat \times (\pi / 180))) \times Cos(Lon \times (\pi / 180)) \tag{2}$$

$$Y = Sin(\pi / 2 - (Lat \times (\pi / 180) \times Sin(Lon \times (\pi / 180)) \tag{3}$$

$$Z = Cos(\pi / 2 - (Lat \times (\pi / 180)) \tag{4}$$

We input *Date* into Quench because different times of the year will have different agricultural conditions that influence ET. For example, a *Date* in the summer will have a higher ET because vegetation consumes more water then than a *Date* in the winter when the vegetation has a lower rate of water consumption. Like *Lat* and *Lon*, *Date* also requires cyclical encoding, but instead of going from a 2D encoding to a 3D encoding, it goes from a 1D encoding to a 2D encoding. We do this because *Date* suffers from the same problem as *Lat* and *Lon*, where a single dimension does not accurately represent the distance between two points. For example, the *Date* pairs (Jul. 31, Aug. 1) and (Jan. 1, Dec. 31) both have a distance of one day. However, when calculating the Euclidean distance between (Jan. 1, Dec. 31), we have a distance of 364 days. Once we convert the *Date* to the *Day of the Year* (*N*/365), we then apply the 1D to 2D cyclical feature encoding as shown in Formulas 5 and 6.

After converting *Date* to the *Day of the Year* and applying the cyclical encoding we have 2D (x,y) coordinates accurately representing the distance between different *Date* variables. These outputted encoded values also range between −1 and 1.

$$X = Sin(2 \times \pi \times ((Day\ of\ the\ year - 1) / 364)) \tag{5}$$

$$Y = Cos(2 \times \pi \times ((Day\ of\ the\ year) / 364)) \tag{6}$$

For the categorical variables *Veg* and *Clim*, we use one-hot encoding. There are nine different *Veg* types in our ET dataset. One-hot encoding the third *Veg* type would result in an array of [0, 0, 1, 0, 0, 0, 0, 0, 0]. The nine *Veg* types we use are from the International Geosphere–Biosphere Programme (IGBP) land cover classification system. IGBP classifies areas of land based on the type and amount of vegetation present. The agricultural variable *Veg* heavily influences ET, because different types of vegetation consume and lose water at different rates.

Similar to *Veg*, different *Clim* types are associated with different agricultural conditions that affect ET. For the *Clim* types we use the Köppen Climate Classification, which classifies areas of land based on specific criteria unique to each classification. These criteria influence the type of vegetation present in the surrounding area and thus affect the rate of ET. Like *Veg*, *Clim* also uses one-hot encoding. After encoding the previously mentioned data, we now normalize the remaining environmental data by subtracting each value by the minimum value and then dividing that by the difference between the maximum and minimum values. We normalize each band of the satellite imagery as well as *Elev*.

4. Methodology

To estimate highly-accurate ET at any given location, we have designed a novel neural network architecture, Quench. Quench uses a Deep Capsule Networks that captures dynamic relations between local environmental conditions (e.g., *LST*) and ET in a hierarchical fashion. These conditions often encompass diverse spatial extents at different positions and orientations. Capsule Networks or CapsNets are advantageous in the sense that they do not require a pooling operation, which makes them less susceptible to information loss than traditional CNNs (Sabour et al., 2017). Also, CapsNets have demonstrable performance improvements when identifying objects with different positions or orientations (a.k.a. pose), a task that CNNs often struggle with. In capsule theory, a capsule is a representation of an object, where each object is arranged in a hierarchical fashion. For example, a low-level capsule could represent an ear or a nose, while a high-level capsule could represent a face. In CapsNets, these capsules are represented with activation vectors of an arbitrary length. Each element in these activation vectors represents an attribute of that object and its pose. These concepts are well aligned with our goals to identify and capture the relationships between fast-evolving environmental phenomena and ET over a relatively large and diverse geospatial extent.

4.1. Model architecture

Fig. 1 is a hierarchical representation of Quench, where Quench's inputs include environmental, meteorological, and agricultural data. The details about preprocessing this data are described in Section III-F. Quench is made up of two separate models the SSEBop model and Deep Capsule Network. The SSEBop model is inputted environmental, meteorological, and agricultural data, while the Deep Capsule Network is only inputted the environmental data. The Deep Capsule Network uses AttnConvCap layers, which are particularly designed for capturing relations between attributes in satellite images using both capsule-based computation and attention-based learning. Our SSEBop model implementation uses a CNN to extract information from the 2D data and a two-layer neural network to extract information from the 1D data. This information is then used to calculate ET using SSEBop's thermal index approach. Finally, the knowledge learned from the Deep Capsule Network and SSEBop models are combined and outputted as Quench's final ET estimate.

4.1.1. Satellite imagery encoder: an attention-based capsule network (RQ2)

As part of Quench, we propose a new neural network layer, the AttnConvCap, particularly tailored for capturing dynamic relations between surface observations from satellite imagery and ET. Quench's Deep Capsule Network model is a CapsNet that uses residual learning and is made up of AttnConvCap layers. These AttnConvCap layers aim to decrease the computational cost of the CapsNet by using a newer implementation of capsule-based computation, the convolutional capsule, which have a significantly lower computational footprint than previous CapsNet implementations. Also, the AttnConvCap increases the convolutional capsule's accuracy by guiding its capsule-based computation to focus on the areas of an image with a higher impact on the model's performance (i.e. attention-based learning). By combining the convolutional capsule with attention-based learning, Quench's AttnConvCap successfully addresses the computational challenges existing in the original CapsNet implementation without compromising model accuracy.

4.1.2. Building a capsule-based layer (RQ2)

Attention-based learning is a machine learning method that enables the model to focus on certain areas of the image and has been used successfully in both natural language processing and computer vision applications (Vaswani et al., 2017; Ramachandran et al., 2019). Quench proposes the novel AttConvCap as the neural network layer of the Deep Capsule Network used as Quench's satellite imagery encoder. As depicted in Fig. 1, a SAM (Spatial Attention Mapper) is paired with a convolutional capsule layer to generate a single channel attention map that is the same size as the inputted image. Quench's SAM is comprised of seven convolutional layers and eight RNNs (Recurrent Neural Networks). These can be divided up into an input layer, two directional layers, an attention layer, and an output layer. The idea behind the SAM's architecture is to move the area of attention in varying degrees in four different directions (up, down, left, and right), to find the area of attention that produces the best results. The input layer consists of a single convolutional layer that simply encodes the features of the inputted image. Each directional layer has four RNNs, one for each direction, and a proceeding convolutional layer to combine their four directional matrices into one. The attention layer uses three sequential convolutional layers to create four single channel attention maps, one for each direction. The outputted directional matrices from each of the directional layer's RNNs are first multiplied by their corresponding single channel attention maps from the attention layer, before being combined into a single directional matrix by the proceeding convolutional layer. By multiplying the attention maps with the directional matrices, we move the area of attention in the directions that provide the highest accuracy. We do this twice, using two sequential directional layers to further improve the accuracy of the outputted directional matrix. This directional matrix is then put through the output layer's single convolutional layer, which maps the directional matrix to a single channel attention map. This single channel attention map comprises of values between 0 and 1, where values closer to 1 require a higher amount of the model's attention and values closer to 0 require a lower amount.

Although the original implementation of CapsNets demonstrated a high accuracy on the MNIST (Modified National Institute of Standards and Technology) dataset (28pixels^2), it could not successfully scale to images with a resolution larger than 64pixels^2. The surface conditions of the surrounding area are critical to achieving a high-accuracy with

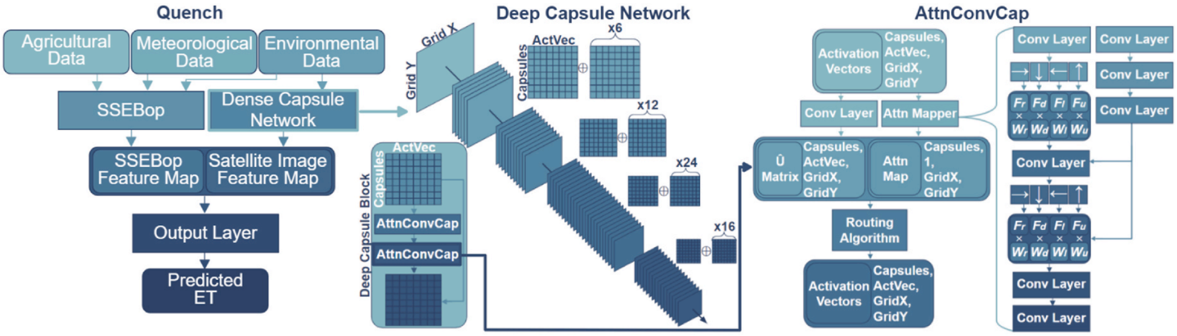


Fig. 1. Hierarchical architecture for Quench. The leftmost model represents Quench's neural network architecture. In the center is the satellite imagery encoder's Deep Capsule Network architecture. The rightmost model represents our novel AttnConvCap layer's architecture.

Quench. We designed the AttnConvCap with a convolutional capsule layer to address the high computational cost of CapsNet layers and a SAM to boost the layer's accuracy. In contrast to CapsNet layers, convolutional capsules use a convolutional layer in their routing algorithm to reduce the number of dimensions needed when calculating the routing matrix Û. The convolutional capsule's routing algorithm is known as locally-connected routing, while the original CapsNet layer's routing algorithm is known as dynamic routing.

Locally-connected routing can result in a slightly lower accuracy compared to dynamic routing. There are two key reasons for this. First, low-level capsules must be within a certain distance of a high-level capsule to be routed to it. This distance is equal to the size of the kernel used in the locally-connected routing's convolutional layer. This distance requirement is not the case with dynamic routing, where different tiered capsules can be at any distance from one another and still be routed. Second, each low-level capsule at different (x,y) locations must share a single transformation matrix in locally-connected routing, while in dynamic routing each low-level capsule has a unique transformation for each (x,y) location. However, in our scenario, the smaller computational costs (i.e. memory usage and training times) achieved by leveraging convolutional capsules layers, outweighs the slight loss of accuracy from the original CapsNet layer implementation. The SAM is inputted the same low-level capsule matrix that the convolutional layer is inputted and outputs an attention map for every low-level capsule. This attention map is then multiplied against Û, which routes low-level capsules to high-level capsules. This attention map guides the focus of routing low-level capsules towards more important areas of the image, while ignoring capsules in less important areas.

4.1.3. Constructing a Deep Capsule Network (RQ2)

We implemented our AttConvCap neural network layer in a Deep Capsule Network with multiple layers of AttConvCap used in its residual learning strategy. Residual learning models such as the ResNet (Residual Neural Network) or DenseNet (Densely-Connected Convolutional Neural Network) are neural networks that input a feature map into a neural network layer and then adds (ResNet) or concatenates (DenseNet) the outputted feature map to the original inputted feature map (He et al., 2015; Huang and Liu Kilian, 2016). In contrast, non-residual learning models do not combine the inputted and outputted feature maps and simply use the latter as the encoded vector. Residual learning is designed to address the degradation problem, which arises when the accuracy of a model decreases as the number of layers in the model increases. This occurs because the update to weights in a neural network decreases exponentially the farther a neural network layer is from the final layer in the model. Also referred to as the vanishing gradients problem, these farther away layers' weights are then being updated by values that are virtually zero, resulting in the weights not changing during training, causing the model to perform poorly. Residual learning not only solves the degradation problem, but increases the model's overall accuracy (He et al., 2015).

To address the degradation problem and improve the model's overall accuracy, Quench stacks multiple AttnConvCap layers using the residual learning strategy. Unlike existing approaches in (Gugglberger David Peer and RodriguezSanchez, 2021; Bhamidi and El-Sharkawy, 2019, 2020), Quench uses a convolutional capsule-based layer (AttnConvCap) instead of the original CapsNet layer implementation. We have explored two different methods to stack these layers; adding (ResCaps) and concatenating (DenseCaps) the inputted and outputted capsule matrices, which are made up of activation vectors. In our implementations of these Deep Capsule Networks the initial lowest-level capsule matrix has eight capsules each with an activation vector of length eight. Because the ResCaps adds the inputted and outputted capsule matrices, the capsule matrices stay at that same 8x8 size throughout training. We chose this 8x8 configuration because it was large enough in size to maintain a high accuracy for our ResCaps model, while also small enough in size to initialize our DenseCaps capsule matrix without running out of memory on the NVIDIA Quadro P2200 GPUs we used for training our models. The DenseCaps' capsule matrices grow in size because it concatenates the inputted and outputted capsule matrices instead of adding them, increasing the outputted capsule matrix's activation vector length by 8 for each DenseCaps block. Both the ResCaps and DenseCaps blocks are made up of two AttnConvCap layers that output a capsule matrix, which is then combined with the inputted capsule matrix. For both the ResCaps and DenseCaps implementations, we group these blocks into four groups of (Anderson, 2020; Brunsell, 2020b; He et al., 2015; Desai, 2021c) Deep Capsule Blocks respectively. In between each of these four groups are transition layers that each reduce the length and height of the (x,y) grid of capsules by half. We have evaluated the performance for these two strategies (see Section V), and the ResCaps model outperforms the DenseCaps by 11.1%.

4.1.4. Improving model generalization (RQ3)

Training a model with highly-sparse training data may result in poor model generalization. Generalization refers to a model's ability to adapt effectively to new, previously unseen data. As described in Section III-A, there are approximately one hundred AmeriFlux sites that measure ET in North and South America, which is relatively sparse when training a model with reasonable generalization for the entire continental U.S. To address this challenge, we have integrated USGS's SSEBop model into Quench so that it can estimate ET values accurately, even when the target location has never been exposed to the model before. Quench uses the SSEBop model to capture relationships between environmental, meteorological, and agricultural observations and ET that are present at any geospatial location.

The USGS's SSEBop model estimates ET based on variables such as *LST* and *SpH*. This model is derived from the SSEBop approach in (Senay, 2008; Gabriel et al., 2011), but with tailored parameterizations for operational applications. It uses remote sensing data to generate ET fractions, which are then combined with *ETr* to approximate ET using the thermal index approach. The SSEBop model has demonstrated that it is capable of providing accurate ET estimations over large, continuous spatial extents.

Although there are several ways of implementing the SSEBop model, each is built around the same thermal index formula that outputs the estimated ET. This formula approximates ET based on the variables *Tmax*, *Cfactor*, *dT*, *LST*, and *ETr*, which can be seen in Formula 7 below.

$$ET = ((T_{m\ ax} \times C\,factor) + dT - LST)/dT \times ETr \tag{7}$$

Tmax represents the daily maximum temperature gathered from GRIDMET. *LST* represents the land surface temperature, which can be gathered from either the Landsat 8's *Thermal Infrared 1* or *2* bands, in our case we use the former. In order to have matching resolutions between the *Tmax* and *LST* images, we use bilinear-interpolation on the *Tmax* image. *ETr* estimates the water usage of a well-watered reference crop such as alfalfa or grass under a set of local weather conditions and is calculated using the Penman-Monteith equation. We retrieve the pre-calculated *ETr* values directly from GRIDMET, which also has the Penman-Monteith equation's parameters, if calculating *ETr* on the client-side is needed. The *dT* represents the vertical difference between the theoretical dry/bare surface temperatures and canopy-level air temperatures for each pixel. The *Cfactor* (a.k.a. temperature correction) is used to correspond the *Tmax* values with the cold/wet environmental conditions. The *Cfactor* can be calculated in a number of ways, but usually involves dividing the *LST* pixels by their corresponding *Tmax* pixels and then filtering out the pixels with low *NDVI* or *LST* values. *NDVI* measures the amount of live green vegetation in an area and can be calculated by taking the normalized difference between the *Near Infrared* and *Red* bands from the Landsat 8's satellite imagery.

For our implementation of SSEBop we estimate the *Cfactor* using a neural network to address the following challenges. First, other implementations used to calculate the *Cfactor* are often limited to images with high *NDVI* and *LST* values. This can be especially limiting in barren areas with little vegetation and/or cold temperatures. Second, other *Cfactor* implementations are usually gridded algorithms that are based on focal and zonal statistical methods that often require hand-tuning parameters (e.g., kernel sizes, number of layers), depending on the number and location of pixels with high *NDVI* and *LST* values. A neural network on the other hand has a single set of parameters for the entire dataset and requires no parameter adjustments between images or areas. This makes neural networks well-suited for calculating *Cfactor* over a large, diverse selection of images and areas. One downside to our neural network *Cfactor* implementation, is that it outputs a single value to represent the *Cfactor* for an entire area, while gridded algorithms output a grid of *Cfactor* values for the area, which is a more fine-grained representation. However, like ET values, *Cfactor* values in an area are quite similar and will have little variation over a small area like 480 m^2, likely making the difference in accuracy from using a 16x16 grid to represent the *Cfactor* instead of a single point very low. Our *Cfactor* model needs to be trained before training the Deep Capsule Network model can begin. The neural network for *Cfactor* estimation comprises a CNN for the 2D data, a two-layer neural network for the 1D data, and a final output layer for combining the 2D and 1D encoder's outputs into a single *Cfactor* value. Two channels are inputted into the CNN: the *LST* image divided by the *Tmax* image and the *NDVI* image. Six variables are inputted into the *Cfactor* neural network's 1D encoder: the *Clim*, *Veg*, *Date*, *Elev*, *Lat*, and *Lon* variables. The CNN and two-layer neural network's outputs are then flattened, concatenated, and inputted into the final neural network layer, which outputs the final *Cfactor* value.

5. Empirical benchmarks and performance evaluation

In order to evaluate how our AttnConvCap neural network layer performs we compare seven machine learning models used for Quench's satellite imagery encoder. We use a CNN, ConvCaps (Convolutional Capsule Network), AttnConvCaps (Convolutional Capsule Network with AttnConvCaps), DenseCaps, DenseAttnCaps (DenseCaps with AttnConvCaps), ResCaps, and ResAttnCaps (ResCaps with AttnConvCaps) for Quench's satellite imagery encoder and compare their results. In order to compare these machine learning models we use k-fold cross-validation splitting our dataset into five subsets of approximately the same size. We do this by first sorting the data by location and then sequentially binning each data into five bins (e.g., [1,2,3,4,5,1,2 …]). We then train each of these models on an 80/20 training/testing split, with the training data comprised of four of the five bins, and the testing data comprised of the single remaining bin. For each permutation of the bins (for a total of five different combinations) we train each of our seven models one hundred times. For each of these five permutations, one hundred trials, and seven model types we use random weight initializations, resulting in 3500 individually trained model instances. For consistency-sake, we only have five SSEBop models instances, one for each permutation of the binned data. Each of the 3500 individually trained model instances uses the same SSEBop model weights corresponding to their bin permutation, in order to preserve the training/testing split. For example, all the CNN, ConvCaps, AttnConvCaps, DenseCaps, DenseAttnCaps, ResCaps, and ResAttnCaps model instances with a bin permutation of 1,3,4,5/2 for the training/testing split uses the weights from the SSEBop model, which trained with the same 1,3,4,5/2 training/testing bin permutation. Once all the models are trained we record each model's ET estimation error for each image in the testing data.

5.1. Software implementation

Quench was implemented in PyTorch and uses an AdaGrad optimizer with a step-based learning rate decay scheduler. We used MSE (Mean Squared Error) as the loss function and the model was trained over three hundred epochs. Quench's hyperparameters, such as the number of layers, kernel size, etc., were chosen based on empirical observations combined with hand-tuning the model to ensure a high accuracy. Also factoring into the choosing of these hyperparameters was that each satellite imagery encoder should have a similar number of layers and nodes, in order to make a comparison between them fair, especially computation-wise. Our dataset creation involved selecting environmental, meteorological, and agricultural variables based on their model performance impact. Each time we removed a variable that did not boost the model's accuracy we retested each of the remaining variables.

5.2. ***Model*** *analysis*

The CNN model is used as the base machine learning model to measure the improvements we get from using capsule-based computation, attention-based learning, and/or residual learning in the other satellite imagery encoder models. Comparing the ConvCaps model to the CNN model gives a good comparison of the changes in both accuracy and computational costs when using convolutional capsule layers instead of convolutional layers. Comparing the DenseCaps and the ResCaps performance to the ConvCaps' allows us to quantify the increases in performance from using residual learning with convolutional capsules. An analysis of the DenseCaps and ResCaps performance, gives a clear indication of whether adding or concatenating the capsule matrices is more effective, especially in regards to the computational cost. Finally, by comparing the performance of the ConvCaps to the AttnConvCaps, the ResCaps to the ResAttnCaps, and the DenseCaps to the DenseAttnCaps, we are able to evaluate the effectiveness of using our AttnConvCap layer instead of a standard convolutional capsule layer.

CNN The CNN satellite imagery encoder consists of four convolutional layers, each followed by a LeakyReLU (Leaky Rectified Linear Unit) layer, a batch normalization layer, and an average pooling layer respectively. After these layers a single neural network layer is used to estimate ET from the outputted encoded vector. The CNN is the least complex of the machine learning models, but is a tried and true machine learning model.

ConvCaps and AttnConvCaps The ConvCaps' architecture is similar to the CNN's except that the last three convolutional layers are replaced with convolutional capsule layers and no average pooling layers are used. The number of capsules starts at 32 and is then halved after each layer (i.e. 16, 8, and 4). Each of these capsules has the same activation vector length of 32. Using convolutional capsule layers instead of convolutional layers improves the accuracy of Quench's satellite imagery encoder. This is due to the convolutional capsule layers' increased ability to identify objects with different poses and lower information loss. The AttnConvCaps model is identical to the ConvCaps model except instead of convolutional capsule layers it uses AttnConvCap layers.

DenseCaps and DenseAttnCaps The DenseCaps model is composed of four groups with (Anderson, 2020; Brunsell, 2020b; He et al., 2015; Desai, 2021c) layers of DenseCaps Blocks that each increase the activation vector length by 8. Each of the transitional layers between these blocks reduces the activation vector length by half in addition to halving the capsule matrices' (x,y) axes length. Halving the activation vector length is done to prevent the DeepCaps' capsule matrices from becoming exceedingly large. The end result is the final outputted capsule matrix having 8 capsules each with an activation vector of length 255. This large increase in the activation vector length over training causes the DenseCaps to have a relatively large computational cost, similar to what we saw in (Sun et al., 2021). The DenseAttnCaps model is identical to the DenseCaps with the sole exception of each convolutional capsule layer being replaced by an AttnConvCap layer.

ResCaps and ResAttnCaps Unlike DenseCaps, our ResCaps model adds the inputted and outputted capsule matrices instead of concatenating them, keeping the activation vector length the same throughout training. Also, the ResCaps transition layers do not halve the length of the activation vectors, but still halve the length of the (x,y) axes. Halving the activation vector lengths is not needed for the ResCaps because the activation vector lengths are not growing exceedingly large due to concatenation of the capsule matrices. Due to the ResNet's residual learning strategy, the ResCaps maintains a relatively low-computational footprint, while also gaining the benefits of residual learning. This ResCaps model has been used with the original CapsNet implementation to achieve state-of-the-art results in (Gugglberger David Peer and RodriguezSanchez, 2021; Bhamidi and El-Sharkawy, 2019, 2020), but to our knowledge has not been used with multiple convolutional capsule layers (Gugglberger David Peer and RodriguezSanchez, 2021; Rajasegaran et al., 2019). Similar to the ConvCaps and AttnConvCaps and the DenseCaps and DenseAttnCaps, the sole difference between the ResCaps and ResAttnCaps is the use of AttnConvCap layers instead of convolutional capsule layers.

Model Accuracy After we trained the different machine learning models used for Quench's satellite imagery encoder, we average the error of the testing data for each model. The proposed model, ResAttnCaps, demonstrated the lowest MAE of 0.4577 mm on the testing data, which is a 56.75% decrease in the error compared to the lone SSEBop model, which achieved a MAE of 1.0562 mm. The next best performing model was the ResCaps model, which had a MAE of 0.4973 mm (53.01% decrease from SSEBop). Using the AttnConvCap in the ResCaps model resulted in a 7.96% decrease in the model's MAE, which is equivalent to 0.04 mm of ET. The third best performing model was the DenseAttnCaps, which had a MAE of 0.5149 mm (51.34% decrease from SSEBop). Unsurprisingly, the fourth best model was the DenseCaps, which had a MAE of 0.5250 mm (50.39% decrease from SSEBop). The difference between the error of the DenseAttnCaps and DenseCaps models was 1.92% or 0.01 mm of ET. Using our novel AttnConvCap improved the accuracy of both the ResCaps and DenseCaps models, but improved the ResCaps by four times more than the DenseCaps. The fifth and sixth best performing models were the AttnConvCaps followed by the ConvCaps model, which achieved a MAE of 0.9285 mm and 0.9395 mm (12.25% and 11.22% decrease from SSEBop) respectively. This again shows that using the AttnConvCap, instead of the standard convolutional capsule layer, improves the model's overall accuracy. Finally, the seventh best performing model was our baseline CNN model, which achieved a MAE of 1.0442 mm (1.32% decrease from SSEBop). Each of these models' MAE on the testing data can be seen in Fig. 2.

Computational Cost Let us now take a look at each satellite imagery encoder's cost (see Table 1). We measure this by taking the memory size of the model's saved weights, which are saved in a PT file (PyTorch tensor file). Unsurprisingly, the largest model is the DenseAttnCaps with a size of 255 MB, which had the third best MAE. The second largest model is theDenseCaps with a size of 79 MB, which had the fourth best MAE. The DenseCaps model is 35% the size of the DenseAttnCaps, indicating that the AttnConvCap increases the size of this model by almost three times its original size. The third largest and best performing model was the ResAttnCaps with a size of 6 MB. Interestingly, the fourth largest was the AttnConvCaps with a size of 3.9 MB, which had the fifth best model performance. The fifth and sixth largest models were the ResCaps (the second best performing model) followed by the ConvCaps (the sixth best performing model) each with a size of 3.5 MB. The ResCaps model is 58% the size of the ResAttnCaps, which means using the AttnConvCap layers in the ResCaps model is roughly doubling the size of the model. The smallest model was the CNN, which was the worst performing model. The key takeaway from these results is that although the DenseAttnCaps and DenseCaps have significantly larger footprints than the ResAttnCaps and ResCaps, they perform slightly worse. We posit that the smaller ResCaps' 8x8 capsule matrix is likely representing the dynamic relationships between the environmental data and ET more efficiently than the larger DenseCaps' 8x255 capsule matrix. This may be due to the DenseCaps' larger capsule matrix having more noise and unuseful elements in its activation vector, compared to the ResCaps' capsule matrix, which has a more concise, less noisy representation of the dynamic relationships between the environmental data and ET.

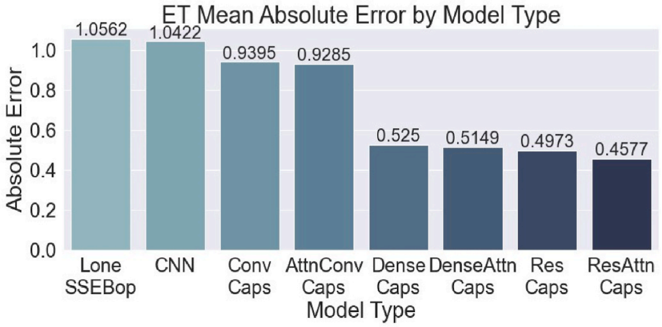


Fig. 2. Each satellite imagery encoder's MAE.

Table 1
Each satellite imagery encoder by MAE, Error Variance, Model Size, and Training Time.

Model Type	MAE	Error Variance	Model Size	Train Time
CNN	1.0442 mm	0.8238 mm^2	1,128 KB	2.6680s
ConvCaps	0.9395 mm	0.9477 mm^2	3,524 KB	8.1802s
AttnConvCaps	0.9285 mm	0.8877 mm^2	3,883 KB	11.4140s
DenseCaps	0.5250 mm	0.2639 mm^2	79,850 KB	20.1411s
DenseAttnCaps	0.5149 mm	0.2740 mm^2	225,356 KB	43.7888s
ResCaps	0.4973 mm	0.2510 mm^2	3,560 KB	15.8213s
ResAttnCaps	0.4577 mm	0.2431 mm^2	6,119 KB	37.0856s

As another measure of each satellite imagery encoder's model complexity we timed each of their epochs during training and averaged them. Each satellite imagery encoder's training time uses the same batch size of 32. The shortest training time is the CNN, which took 2.80s to train for each epoch. Following the CNN was the ConvCap, which took 8.18s to train each epoch, which confirms that the convolutional capsule layer-based model is more complex than the convolutional layer-based model in both memory usage and training time. The model with the third shortest training time was the AttnConvCap, which took 11.41s for each epoch to train. The model with the fourth longest training time was the ResCaps, which took 15.82s to train each epoch. The ResCaps is followed by the DenseCaps, which had an average epoch time of 20.14s. The sixth and seventh shortest epoch times were the ResAttnCaps at 37.08s followed by the DenseAttnCaps at 43.78s. The DenseCaps model takes roughly 27% more time to train than the ResCaps, even though the ResCaps model achieved a higher accuracy. This is also true with the DenseAttnCaps, which takes 18% longer than the ResAttnCaps, but has a lower accuracy. One interesting take away from these training times is that using AttnConvCap layers in the ConvCaps model instead of convolutional capsule layers increases the training time by only 28%. Compare this to using the AttnConvCap layers in the ResCaps and DenseCaps models, which significantly increases the training time, more than doubling both of them. This indicates that although using AttnConvCap layers instead of standard convolutional capsule layers increases the models' accuracy, they increase both the computational footprint and training time of the models. One key difference between the memory usage and the training times is that the ResAttnCaps and ResCaps have a relatively low memory cost compared to their more expensive training times. This is because the matrices used in locally-connected routing are not saved in the PT file, while the convolutional layers' weights are. The training times, unlike the memory usage, do get increased when capsule-based computation is used because they accurately measure the use of the locally-connected routing's matrices. Overall the ResAttnCaps and ResCaps models seem to provide better results than the corresponding DenseAttnCaps and DenseCaps models, while also costing less in both memory and training time.

5.3. *Spatial analysis*

Now that we know that the ResAttnCaps provides the best accuracy of our seven satellite imagery encoders, we can now use it in our final

Quench model for our remaining analyses. Let us first take a look at the performance of this Quench model relative to each Flux Tower and Lysimeter site's geospatial location. The Flux Tower and Lysimeter sites are distributed across the continental U.S. and can be seen in Fig. 3.

Although these sites are spaced out, some of them tend to be grouped into smaller clusters (e.g., central California, southeastern Arizona), resulting in some of the 54 site markers in Fig. 3 having some overlap. However, the visible markers allow us to see which areas perform better than others. There is only one marker that has a shade of dark blue, showing a MAE higher than 1.0, which is located in eastern Kansas. However, on either side of this site are markers with significantly better results, indicating that the poorly performing marker is not the product of its geospatial positioning. Aside from that single poorly performing marker the other markers seem to be evenly distributed and in the range of 0.0–0.8 mm MAE. Let us now look at the variogram of our final Quench model in Fig. 4.

Variograms are used to measure spatial semivariance among multidimensional coordinates. The x-axis represents the distance between coordinates in respect to degrees of *Lat* and *Lon*. The semivariance metric on the y-axis represents the amount of variance from the MAE. What really stands out in this variogram is the shortness of the range (the height of the plotted line), which is approximately 0.06 mm^2, which is relatively low.

Having a small range indicates that Quench's absolute error is not dependent on the location, while having a large range indicates that Quench's absolute error is heavily influenced by its geospatial location. A variogram will always have some amount of range, due to the random variability found in statistical data. However, having a range of 0.06 mm^2 indicates that Quench performs uniformly across our ET dataset's AmeriFlux and Lysimeter sites, with the location of sites having no significant correlation to Quench's absolute error. We can confirm this with the nugget of the variogram (the starting height of the plotted line), which is virtually zero. The nugget represents the small-scale variability of our model's error. Both the low range and low nugget value of the variogram suggests that Quench's absolute error is uniform across each of the geospatial locations in our ET dataset at both a local and regional scale. Now that we have analyzed Quench's performance in relation to its geospatial coordinates, let us now take a look and see if Quench's performance is affected by seasonal, climatic, and vegetative factors.

5.4. Seasonal analysis

First let us take a look at how our final Quench model performs in each month of the year. In the test data we have a fairly uniform distribution of data in each month. The percentage of our ET dataset for each month ranges from 6% to 10%, which can be seen in Fig. 5. Above in Fig. 6 we can see a clear pattern that the model performs better in the colder seasons than the hotter ones. The winter months perform the best with a MAE of 0.33 mm, followed by the fall months with a MAE of 0.41 mm, the spring months with a MAE of 0.49 mm, and finally the summer months with an MAE of 0.62 mm. We are able to confirm this by taking the Kendall correlation coefficient of each season and their corresponding error, which results in (winter, −0.1216), (fall, −0.0275), (spring, 0.0285), and (summer, 0.1256). Although each of these correlation coefficients have a relatively weak relationship (greater than −0.2 and less than 0.2), we see that the winter and fall months have a negative correlation, while the spring and summer months have a positive correlation. This is likely due to the fact that ET changes more in the hotter months making it harder to estimate. We can confirm this by taking the same Kendall correlation coefficients for the SSEBop model, which are (winter, −0.0258), (fall, −0.0477), (spring, 0.0045), and (summer, 0.0733), which follow the same pattern of the hotter seasons having a positive correlation and the colder seasons having a negative correlation. Overall our Quench model performs relatively uniform across the different months, which we can see in the weak correlation coefficients and small changes in seasonal MAE, although Quench does appear to perform slightly better in the colder seasons.

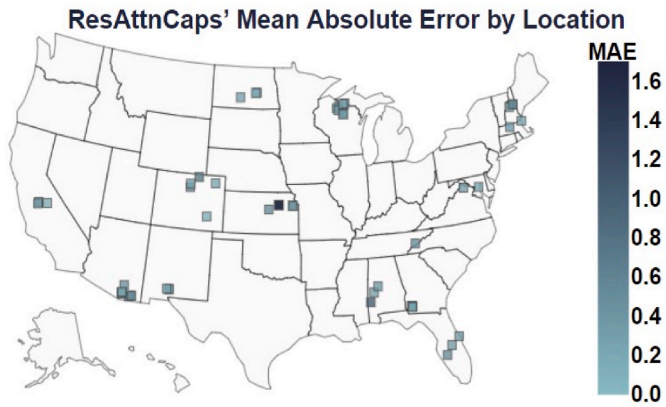


Fig. 3. Quench's MAE by location.

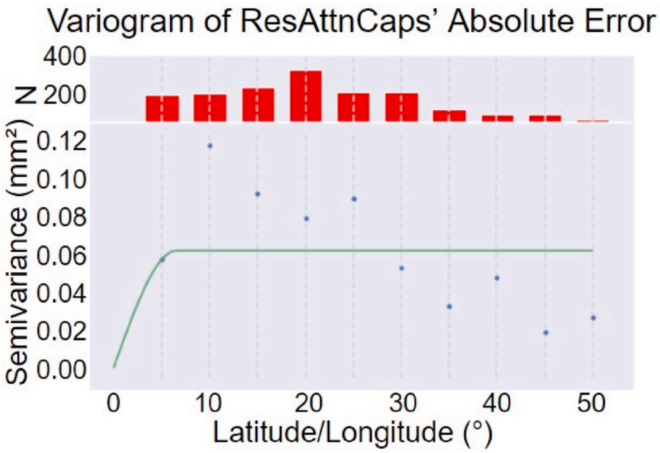


Fig. 4. A variogram of Quench's absolute error.

5.5. Climate analysis

There are a total of ten different *Clim* types in our dataset, which in ascending order of MAE are: Bsh (Hot Semi-Arid), Csa (Hot Summer Mediterranean), Dfa (Hot-Summer Humid Continental), Bwk (Cold Desert), Bsk (Cold Semi-Arid), Cwa (Hurricane-Influenced Humid Subtropical), Dfb (Warm-Summer Humid Continental), Dwb (Hurricane-Influenced Warm-Summer Humid Continental), Dfc (Subarctic), and Cfa (Humid Subtropical). Each of their performances can be seen in Fig. 7.

The *Clim* types of our ET dataset are less well-distributed than the *Veg* type. Each *Clim* type's percentage of our ET dataset ranges from 0.2% to 33%, which can be seen in Fig. 5.

The range of MAE by *Clim* types seems similar to the range of MAE by season, where each *Clim* type and season's MAE ranges between 0.2 mm and 0.7 mm. The exceptions to this are the Bsh and Csa *Clim* types, which each make up less than 1.0% of the total data, likely resulting in their lower MAEs. To confirm this we take the Kendall correlation coefficient for each *Clim* type, which are: (Bsh, −0.0423), (Csa, −0.0558), (Dfa, −0.0883), (Bwk, −0.1050), (Bsk, −0.0991), (Cwa, −0.0340), (Dfb, −0.0502), (Dwb, 0.0136), (Dfc, 0.0339), and (Cfa, 0.2340). Excluding the Cfa *Clim* type each of these correlation coefficients have relatively weak correlations, indicating that Quench performs uniformly on each of them. The Cfa (Humid Subtropical) *Clim* type on the other hand, barely has a medium strength correlation (0.2–0.4) and the highest MAE of 0.68 mm. The Cfa *Clim* type has a relatively high temperature and more complex plant and water ecosystems than the other *Clim* types,

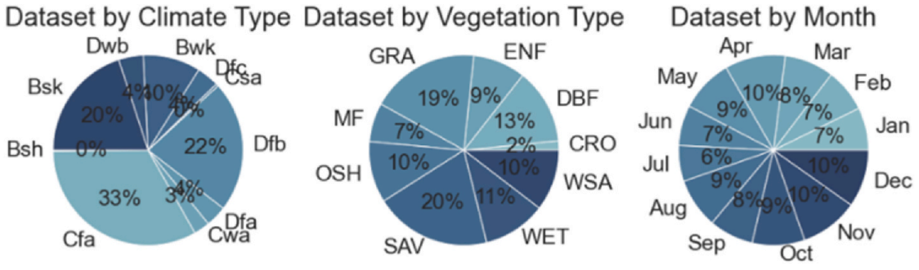


Fig. 5. ET dataset by *Clim* type, *Veg* type, and month respectively.

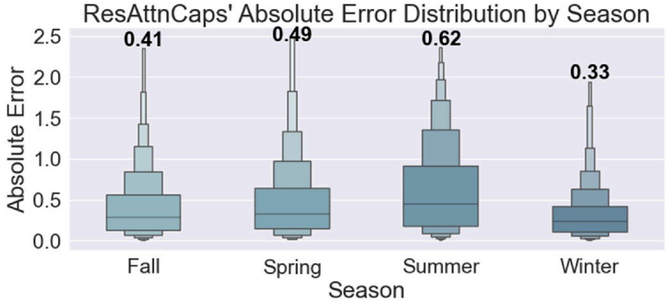


Fig. 6. Quench's absolute error distribution by season.

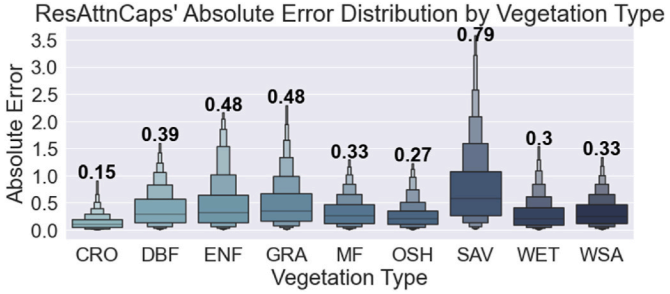


Fig. 8. Quench's absolute error distribution by *Veg* type.

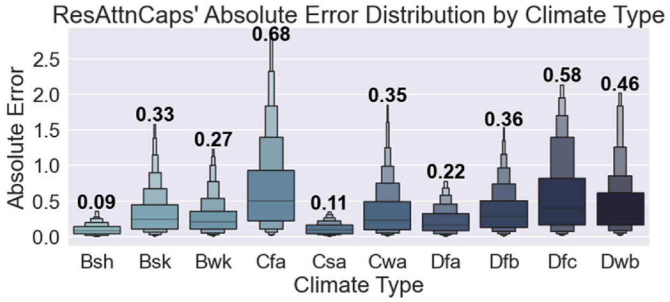


Fig. 7. Quench's absolute error distribution by *Clim* type.

which is likely the reason that Quench performs slightly worse on the Cfa *Clim* type. Another possible reason is that the Cfa *Clim* type makes up a third of the total data, resulting in a larger distribution of absolute error. However, the Cfa error still has a relatively low correlation coefficient, just making it into the medium strength correlation range. Also the Cfa's MAE of 0.68 mm is not that much larger than the MAE that Quench achieved on the remaining *Clim* types (0.45 mm), indicating Quench performs uniformly on each *Clim* type, with some minor variations in its performance on the Cfa *Clim* type.

5.6. Vegetative analysis

Our dataset contains nine different *Veg* types, which in ascending order of MAE are: CRO (Croplands), OSH (Open Shrublands), WET (Permanent Wetlands), WSA (Woody Savannas), MF (Mixed Forests), DBF (Deciduous Broadleaf Forests), ENF (Evergreen Needleleaf Forests), GRA (Grasslands) and SAV (Savannas). Each of these *Veg* types' performance can be seen in Fig. 8.

The *Veg* types are more evenly distributed than *Clim* types, but less uniform than the months, with a range of 7–20%, excluding the CRO, which makes up less than 2% of the data. When pairing the *Veg* types with their corresponding Kendall correlation coefficient, we get: (CRO, −0.0921), (OSH, −0.1090), (WET, −0.0955), (WSA, 0.0552), (MF, −0.0390), (DBF, −0.0173), (ENF, 0.0081), (GRA, 0.03981), and (SAV, 0.2227). Each of these again have a weak correlation except for the SAV *Veg* type. The SAV *Veg* type makes up approximately one fifth of the data and has a medium correlation coefficient strength. Likely the reason for this poor performance is that the SAV *Veg* type has a higher variety in its tree coverage (10–30%), than the other *Veg* types, which are either predominantly forested or canopy-free areas. This diverse mix of herbaceous and other understory systems that the SAV *Veg* type has, could make estimating accurate ET values more difficult for Quench, compared to the other *Veg* types. Despite the higher MAE of SAV (0.79 mm), a correlation coefficient of 0.2227 is still fairly low, and just reaches the medium strength correlation threshold. Although Quench does appear to struggle more on the SAV *Veg*, it performs remarkably well on the remaining 80% of the data, where Quench achieved a MAE of less than 0.5 mm on each remaining *Veg* type. Overall, Quench is able to achieve a high overall accuracy, while maintaining a high-quality, consistent ability to generalize over a diverse set of regions, seasons, climates, and vegetations, with some minor effects to its performance in the hotter months, the Cfa *Clim* type, and the SAV *Veg* type.

5.7. Effectiveness of integrating outputs from the SSEBop model

To evaluate the effectiveness of using the SSEBop model in Quench, we trained Quench with and without the SSEBop model and compared their performance. We found that using the SSEBop model increased Quench's overall accuracy by 11.5%. We also found that using the SSEBop model reduces the number of error outliers in the testing data by 16.2%. This indicates that by combining the SSEBop and Deep Capsule Network models into Quench's architecture, we increase Quench's accuracy and ability to generalize.

6. Conclusion

We described our model, Quench, as accurately estimating ET over large geospatial extents. Quench addresses the model performance challenges stemming from geospatial variability with its uniquely designed neural network architecture that incorporates a SSEBop model and Deep Capsule Network comprised of our novel AttnConvCap layers.

RQ1) We integrate environmental, meteorological, and agricultural datasets and align these based on their spatiotemporal characteristics. Data collected over three years are encoded (e.g., cyclical encoding and one-hot encoding) and normalized in preparation for training Quench and analyzing its performance.

RQ2) Quench's Deep Capsule Network captures the dynamic relationships between local ancillary conditions regardless of its orientation or spatial positioning. Quench uses AttConvCap layers to construct its Deep Capsule Network. The novel AttConvCap effectively determines the area that Quench focuses on, while efficiently extracting ET information from satellite imagery.

RQ3) Quench improves its generalization by leveraging outputs from the process-based SSEBop model. This addresses the input data's sparsity issues, which stem from the low number of available ET measurement sites.

Overall, Quench demonstrates highly-accurate results and consistent model performance that is suitable for estimating ET over large geospatial extents. Quench accomplishes an MAE of 0.4577 mm, significantly outperforming the other base models such as the CNN and lone SSEBop model. Quench has demonstrably consistent model performance across our ET dataset, which encompasses the continental U.S. and a large variety of regions, seasons, climates, and vegetations.

Software and data availability

All the code for the CNN, ConvCaps, AttnConvCaps, DenseCaps, DenseAttnCaps, ResCaps, and ResAttnCaps models are freely and publicly available on GitHub at https://github.com/Samarmy/Attention-Based- Convolutional-Capsules-for-Evapotranspiration-Estimation-at-Scale as of September 01, 2022. This code was developed and is maintained by Samuel Armstrong, who can be contacted at Sam.Armstrong@colostate.edu. Python version 3.6.8 was the programming language used for data processing and modeling. The neural network library used for developing and training our models was PyTorch version 1.6.0. We recommend using hardware with a CUDA compatible GPU when running our code, although this is not necessary. The total size of the model files is 256 KB. The data used to train these models is also available on the same GitHub repository. This data is made up of NumPy files, which have a collective size of 42 MB. This dataset is comprised of publicly available data from the AmeriFlux data portal (https://ameriflux.lbl.gov/data/download-data/), USGS Earth Explorer (https://earthexplorer.usgs.gov/), and GridMET website (https://www.climatologylab.org/gridmet.html). The Lysimeter data used in this manuscript was provided by the Colorado State University Arkansas Valley Research Center and can be found on our GitHub repository at https://github.com/Samarmy/Attention-Based- Convolutional-Capsules-for-Evapotranspiration-Estimation-at-Scale.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the National Science Foundation [OAC-1931363, ACI-1553685], the National Institute of Food and Agriculture [COL0-FACT-2019], and a Cochran Family Professorship. Funding for the AmeriFlux data portal was provided by the U.S. Department of Energy Office of Science. Landsat-8 images were courtesy of the U.S. Geological Survey. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

References

Abatzoglou, John T., Dec. 2011. Development of gridded surface meteorological data for ecological applications and modelling. Int. J. Climatol. 33,1, 121–131. https://doi.org/10.1002/joc.3413, 10.1002/joc.3413.

Adams, Anthony, Vamplew, Peter, 1998. Encoding and decoding cyclic data. Commonwealth Q. 16, 54–58.

Adeloye, Adebayo J., Rustum, Rabee, Kariyama, Ibrahim D., 2012. Neural computing modeling of the reference crop evapotranspiration. Environ. Model. Software 29,1, 61–73. https://doi.org/10.1016/j.envsoft.2011.10.012. https://www.sciencedirect.com/science/article/pii/S1364815211002234, 1364–8152.

Allan, Andales, et al., Jan. 2018. Design and construction of a precision weighing lysimeter in southeast Colorado. Transactions of the ASABE 61, 509–521. https://doi.org/10.13031/trans.12282.

(Dataset) Anderson, Ray G., 2020. *AmeriFlux BASE US-ASL USSL San Joaquin Valley Almond Low Salinity,* Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617706. https://www.osti.gov/dataexplorer/biblio/dataset/1617706.

(Dataset) Anderson, Ray G., 2020. *AmeriFlux BASE US-PSL USSL San Joaquin Valley Pistachio Low*, Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617720. https://www.osti.gov/dataexplorer/biblio/dataset/1617720.

(Dataset) Barron-Gafford, Greg, 2021. AmeriFlux BASE US-MtB Mt Bigelow, Ver, vols. 3–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579717. https://www.osti.gov/dataexplorer/biblio/dataset/1579717.

Bhamidi, Sree Bala Shruthi, El-Sharkawy, Mohamed, 2019. Residual capsule network. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference. UEMCON. https://doi.org/10.1109/EMCON47517.019.8993019, 0557–0560.

Bhamidi, Sree Bala Shruthi, El-Sharkawy, Mohamed, 2020. 3 level residual capsule network for complex datasets. In: 2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS), pp. 1–4. https://doi.org/10.1109/LASCAS45839.2020.9068990.

(Dataset) Brunsell, Nathaniel, 2020a. AmeriFlux BASE US-KFS Kansas Field Station, Ver, vols. 7–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246132. https://www.osti.gov/dataexplorer/biblio/dataset/1246132.

(Dataset) Brunsell, Nathaniel, 2020b. AmeriFlux BASE US-Kon Konza Prairie LTER (KNZ), Ver. 5-5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246068. https://www.osti.gov/dataexplorer/biblio/dataset/1246068.

(Dataset) Brunsell, Nathaniel, 2021. AmeriFlux BASE US-KLS Kansas Land Institute, Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1498745. https://www.osti.gov/dataexplorer/biblio/dataset/1498745.

Casu, F., et al., Dec. 2017. Big remotely sensed data: tools, applications and experiences, 10.1016/j.rse.2017.09.013. https://doi.org/10.1016/j.rse.2017. 09.013, 202,1-2.

(Dataset) Craig, Tweedie, 2021. AmeriFlux BASE US-Jo1 Jornada Experimental Range Bajada Site, Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1767833. https://www.osti.gov/dataexplorer/biblio/dataset/1767833.

(Dataset) Desai, Ankur, 2021a. AmeriFlux BASE US-Los Lost Creek, Ver, vol. 195. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246071. https://www.osti.gov/dataexplorer/biblio/dataset/1246071.

(Dataset) Desai, Ankur, 2021b. AmeriFlux BASE US-PFa Park Falls/WLEF, Ver., vols. 19–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246090. https://www.osti.gov/dataexplorer/biblio/dataset/1246090.

(Dataset) Desai, Ankur, 2021c. AmeriFlux BASE US-Syv Sylvania Wilderness Area, Ver, vols. 18–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246106. https://www.osti.gov/dataexplorer/biblio/dataset/1246106.

(Dataset) Desai, Ankur, 2021d. AmeriFlux BASE US-WCr Willow Creek, Ver., vols. 21–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246111. https://www.osti.gov/dataexplorer/biblio/dataset/1246111.

(Dataset) Foster, Tammy E., 2021. AmeriFlux BASE US-KS4 Kennedy Space Center (Spartina Marsh), Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1767834. https://www.osti.gov/dataexplorer/biblio/dataset/1767834.

Gabriel, Senay, et al., Dec. 2011. Estimating basin scale evapotranspiration (ET) by water balance and remote sensing methods. Hydrol. Process. 25, 4037–4049. https://doi.org/10.1002/hyp.8379.

(Dataset) Giblin, Anne, 2021. AmeriFlux BASE US-PHM Plum Island High Marsh, Ver, vols. 3–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1543377. https://www.osti.gov/dataexplorer/biblio/dataset/1543377.

Gowda, Prasanna H., et al., 2007. In: ET mapping for agricultural water management: present status and challenges, 3, pp. 223–237. https://doi.org/10.1007/s00271-007-0088-6, 26.

(Dataset) Green, Mark, Kelsey, Eric, 2020. AmeriFlux BASE US-HBK Hubbard Brook Experimental Forest, Ver. vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1634881. https://www.osti.gov/dataexplorer/biblio/dataset/1634881.

Gugglberger, Josef, David Peer, RodriguezSanchez, Antonio, 2021. Training Deep Capsule Networks with Residual Connections arXiv: 2104.07393 [cs.CV].

He, Kaiming, et al., 2015. Deep residual learning for image recognition. In: *CoRR* abs/1512.03385 arXiv: 1512. 03385. http://arxiv.org/abs/1512.03385.

(Dataset) Hobbie, John, Rocha, Adrian, Shaver, Gaius, 2020. AmeriFlux BASE US-An1 Anaktuvuk River Severe Burn, Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246142. https://www.osti.gov/dataexplorer/biblio/dataset/1246142.

(Dataset) Hollinger, David, Richardson, Andrew, 2019. AmeriFlux BASE USBar Bartlett Experimental Forest, Ver, vols. 5–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246030. https://www.osti.gov/dataexplorer/biblio/dataset/1246030.

Huang, Gao, Liu, Zhuang, Kilian, Q., 2016. Weinberger. "Densely connected convolutional networks". In: *CoRR* Abs/1608, 06993 arXiv: 1608.06993. http://arxiv.org/abs/1608.06993.

Karpatne, Anuj, et al., 2017. Theory-guided data science: a new paradigm for scientific discovery from data. In: IEEE Transactions on Knowledge and Data Engineering 29, vol. 10, pp. 2318–2331. https://doi.org/10.1109/TKDE.2017.2720168, 10.1109/tkde.2017.2720168, 1041–4347.

King, Michael D., et al., 2013. Spatial and temporal distribution of clouds observed by MODIS onboard the terra and aqua satellites. IEEE Trans. Geosci. Rem. Sens. 51 (7), 3826–3852. https://doi.org/10.1109/TGRS.2012.2227333.

Lecun, Yann, Haffner, Patrick, Bengio, Y., Aug. 2000. Object Recognition with Gradient-Based Learning.

McShane, Ryan R., Driscoll, Katelyn P., Roy, Sando, 2017. A Review of Surface Energy Balance Models for Estimating Actual Evapotranspiration with Remote Sensing at High Spatiotemporal Resolution over Large Extents. https://doi.org/10.3133/sir20175087, 10.3133/sir20175087.

Monteith, John L., 1965. Evaporation and environment. In: Symposia of the Society for Experimental Biology, vol. 19. Cambridge University Press (CUP) Cambridge, pp. 205–234.

(Dataset) NEON (National Ecological Observatory Network), 2021I. AmeriFlux BASE US-xJR NEON Jornada LTER (JORN), Ver, vol. 45. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617731. https://www.osti.gov/dataexplorer/biblio/dataset/1617731.

(Dataset) NEON (National Ecological Observatory Network)., 2021J. AmeriFlux BASE US-xKA NEON Konza Prairie Biological Station - Relocatable (KONA), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579722. https://www.osti.gov/dataexplorer/biblio/dataset/1579722.

(Dataset) NEON (National Ecological Observatory Network), 2021K. AmeriFlux BASE US-xKZ NEON Konza Prairie Biological Station (KONZ), Ver, vols. 5–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1562392. https://www.osti.gov/dataexplorer/biblio/dataset/1562392.

(Dataset) NEON (National Ecological Observatory Network), 2021L. AmeriFlux BASE US-xLE NEON Lenoir Landing (LENO), Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1773398. https://www.osti.gov/dataexplorer/biblio/dataset/1773398.

(Dataset) NEON (National Ecological Observatory Network), 2021M. AmeriFlux BASE US-xNG NEON Northern Great Plains Research Laboratory (NOGP), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617732. https://www.osti.gov/dataexplorer/biblio/dataset/1617732.

(Dataset) NEON (National Ecological Observatory Network), 2021N. AmeriFlux BASE US-xRM NEON Rocky Mountain National Park, CASTNET (RMNP), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579723. https://www.osti.gov/dataexplorer/biblio/dataset/1579723.

(Dataset) NEON (National Ecological Observatory Network), 2021O. AmeriFlux BASE US-xSE NEON Smithsonian Environmental Research Center (SERC), Ver, vols. 4–5. AmeriFlux AMP. https://www.osti.gov/dataexplorer/biblio/dataset/1617734. https://doi.org/10.17190/AMF/1617734.

(Dataset) NEON (National Ecological Observatory Network), 2021P. AmeriFlux BASE US-xSL NEON North Sterling, CO (STER), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617735. https://www.osti.gov/dataexplorer/biblio/dataset/1617735.

(Dataset) NEON (National Ecological Observatory Network), 2021Q. AmeriFlux BASE US-xSP NEON Soaproot Saddle (SOAP), Ver., vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617736. https://www.osti.gov/dataexplorer/biblio/dataset/1617736.

(Dataset) NEON (National Ecological Observatory Network), 2021R. AmeriFlux BASE US-xSR NEON Santa Rita Experimental Range (SRER), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579543. https://www.osti.gov/dataexplorer/biblio/dataset/1579543.

(Dataset) NEON (National Ecological Observatory Network), 2021S. AmeriFlux BASE US-xST NEON Steigerwaldt Land Services (STEI), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617737. https://www.osti.gov/dataexplorer/biblio/dataset/1617737.

(Dataset) NEON (National Ecological Observatory Network), 2021T. AmeriFlux BASE US-xTA NEON Talladega National Forest (TALL), Ver, vols. 3–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1671902. https://www.osti.gov/dataexplorer/biblio/dataset/1671902.

(Dataset) NEON (National Ecological Observatory Network), 2021U. AmeriFlux BASE US-xTR NEON Treehaven (TREE), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1634886. https://www.osti.gov/dataexplorer/biblio/dataset/1634886.

(Dataset) NEON (National Ecological Observatory Network), 2021V. AmeriFlux BASE US-xUK NEON the University of Kansas Field Station (UKFS), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617740. https://www.osti.gov/dataexplorer/biblio/dataset/1617740.

(Dataset) NEON (National Ecological Observatory Network), 2021W. AmeriFlux BASE US-xUN NEON University of Notre Dame Environmental Research Center (UNDE), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617741. https://www.osti.gov/dataexplorer/biblio/dataset/1617741.

(Dataset) NEON (National Ecological Observatory Network), 2021X. AmeriFlux BASE US-xWD NEON Woodworth (WOOD), Ver, vol. 45. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579724. https://www.osti.gov/dataexplorer/biblio/dataset/1579724.

(Dataset) NEON (National Ecological Observatory Network), 2021a. AmeriFlux BASE US-xBL NEON Blandy Experimental Farm (BLAN), Ver, vols. 3–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1671893. https://www.osti.gov/dataexplorer/biblio/dataset/1671893.

(Dataset) NEON (National Ecological Observatory Network), 2021b. AmeriFlux BASE US-xBR NEON Bartlett Experimental Forest (BART), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579542. https://www.osti.gov/dataexplorer/biblio/dataset/1579542.

(Dataset) NEON (National Ecological Observatory Network), 2021c. AmeriFlux BASE US-xCP NEON Central Plains Experimental Range (CPER), Ver., vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579720. https://www.osti.gov/dataexplorer/biblio/dataset/1579720.

(Dataset) NEON (National Ecological Observatory Network), 2021d. AmeriFlux BASE US-xDC NEON Dakota Coteau Field School (DCFS), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617728. https://www.osti.gov/dataexplorer/biblio/dataset/1617728.

(Dataset) NEON (National Ecological Observatory Network), 2021e. AmeriFlux BASE US-xDL NEON Dead Lake (DELA), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1579721. https://www.osti.gov/dataexplorer/biblio/dataset/1579721.

(Dataset) NEON (National Ecological Observatory Network), 2021f. AmeriFlux BASE US-xGR NEON Great Smoky Mountains National Park, Twin Creeks (GRSM), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1634885. https://www.osti.gov/dataexplorer/biblio/dataset/1634885.

(Dataset) NEON (National Ecological Observatory Network), 2021g. AmeriFlux BASE US-xHA NEON Harvard Forest (HARV), Ver, vols. 5–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1562391. https://www.osti.gov/dataexplorer/biblio/dataset/1562391.

(Dataset) NEON (National Ecological Observatory Network), 2021h. AmeriFlux BASE US-xJE NEON Jones Ecological Research Center (JERC), Ver, vols. 4–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617730. https://www.osti.gov/dataexplorer/biblio/dataset/1617730.

(Dataset) Olson, Brent, 2021. AmeriFlux BASE US-ALQ Allequash Creek Site, Ver, vols. 7–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1480323. https://www.osti.gov/dataexplorer/biblio/dataset/1480323.

Panda, Sudhanshu Sekhar, et al., 2019. Estimation of evapotranspiration and its parameters for pine, switchgrass, and intercropping with remotely-sensed images based geospatial modeling. Environ. Model. Software 121, 104487. https://doi.org/10.1016/j.envsoft.2019.07.012. https://www.sciencedirect.com/science/article/pii/S1364815219302701, 1364–8152.

(Dataset) Perez-Ruiz, Eli R., Vivoni, Enrique R., 2020. *AmeriFlux BASE US- Jo2 Jornada Experimental Range Mixed Shrubland, Ver*, vol. 15. AmeriFlux AMP. https://doi.org/10.17190/AMF/1617696. https://www.osti.gov/dataexplorer/biblio/dataset/1617696.

Rajasegaran, Jathushan, et al., 2019. DeepCaps: Going Deeper with Capsule Networks arXiv: 1904.09546.

Ramachandran, Prajit, et al., 2019. Stand-alone self-attention in vision models. In: *CoRR* abs/1906.05909 arXiv: 1906.05909. http://arxiv.org/abs/1906.05909.

Raschka, Sebastian, 2018. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. eprint arXiv:1811.12808.

(Dataset) Ross Hinkle, Charless, 2019. AmeriFlux BASE US-DPW Disney Wilderness Preserve Wetland, Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1562387. https://www.osti.gov/dataexplorer/biblio/dataset/1562387.

(Dataset) Russell, Scott, 2020. AmeriFlux BASE US-CMW Charleston Mesquite Woodland, Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1660339. https://www.osti.gov/dataexplorer/biblio/dataset/1660339.

(Dataset) Russell, Scott, 2021a. AmeriFlux BASE US-SRG Santa Rita Grassland, Ver., vols. 10–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246154. https://www.osti.gov/dataexplorer/biblio/dataset/1246154.

(Dataset) Russell, Scott, 2021b. AmeriFlux BASE US-SRM Santa Rita Mesquite, Ver., vols. 22–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246104. https://www.osti.gov/dataexplorer/biblio/dataset/1246104.

(Dataset) Russell, Scott, 2021c. AmeriFlux BASE US-Wkg Walnut Gulch Kendall Grasslands, Ver, vols. 17–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246112. https://www.osti.gov/dataexplorer/biblio/dataset/1246112.

Sabour, Sara, Frosst, Nicholas, Hinton, Geoffrey E., 2017. Dynamic routing between capsules. In: *CoRR* abs/1710.09829 arXiv: 1710.09829. http://arxiv.org/abs/1710.09829.

Sadeghi, Mojtaba, et al., 2020. Improving near real-time precipitation estimation using a U-Net convolutional neural network and geographical information. Environ. Model. Software 134, 104856. https://doi.org/10.1016/j.envsoft.2020.104856. https://www.sciencedirect.com/science/article/pii/S1364815220309130, 1364–8152.

(Dataset) Scott, Russ, 2021. AmeriFlux BASE US-Whs Walnut Gulch Lucky Hills Shrub, Ver., vols. 17–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1246113. https://www.osti.gov/dataexplorer/biblio/dataset/1246113.

Senay, Gabriel B., 2008. Modeling landscape evapotranspiration by integrating land surface phenology and a water balance algorithm. Algorithms 1-2, 52–68. https://doi.org/10.3390/a1020052. URL:1999–4893. https://www.mdpi.com/1999-4893/1/2/52.

(Dataset) Silveira, Maria, 2021. AmeriFlux BASE US-ONA Florida Pine Flatwoods, Ver, vols. 2–5. AmeriFlux AMP. https://www.osti.gov/dataexplorer/biblio/dataset/1660350. https://doi.org/10.17190/AMF/660350.

(Dataset) Starr, Gregory, 2021a. AmeriFlux BASE US-LL1 Longleaf Pine Baker (Mesic Site), Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1773395. https://www.osti.gov/dataexplorer/biblio/dataset/1773395.

(Dataset) Starr, Gregory, 2021b. AmeriFlux BASE US-LL2 Longleaf Pine Dubignion (Intermediate Site), Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1773396. https://www.osti.gov/dataexplorer/biblio/dataset/1773396.

(Dataset) Starr, Gregory, 2021c. AmeriFlux BASE US-LL3 Longleaf Pine - Red Dirt (Xeric Site), Ver, vols. 1–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1773397. https://www.osti.gov/dataexplorer/biblio/dataset/1773397.

Sun, Kun, et al., Apr. 2021. Dense capsule networks with fewer parameters. Soft Comput. 25, 6927–6945. https://doi.org/10.1007/s0050 0-021-05774-6, 10.

Vaswani, Ashish, et al., 2017. Attention is all you need. In: *CoRR* Abs/1706, 03762 arXiv: 1706.03762. http://arxiv.org/abs/1706.03762.

(Dataset) Vivoni, Enrique R., 2020. AmeriFlux BASE US-SRS Santa Rita Experimental Range Mesquite Savanna, Ver, vols. 2–5. AmeriFlux AMP. https://doi.org/10.17190/AMF/1660351. https://www.osti.gov/dataexplorer/biblio/dataset/1660351.

Chapter 16 - vegetation production in terrestrial ecosystems. In: Liang, Shunlin, Li, Xiaowen, Wang, Jindi (Eds.), 2012. Advanced Remote Sensing. Academic Press, Boston, ISBN 978-0-12-385954-9, pp. 501–531. https://doi.org/10.1016/B978-0-12-385954-9.00016-2. https://www.sciencedirect.com/science/article/pii/B9780123 859549000162.