Delta Learning for Climate Prediction

Peter Bazianos
peterbaz@seas.upenn.edu

Cameron Davis
daviscam@wharton.upenn.edu

Saurabh Mallela
smallela@seas.upenn.edu

Grey Sarmiento
greysar@seas.upenn.edu

***Abstract*—Weather prediction remains one of science's most challenging puzzles, with even our best models struggling to forecast conditions just hours ahead. This study explores an innovative approach called delta learning that could improve scientists' ability predict weather patterns. Instead of trying to predict weather conditions directly, delta learning works by having machine learning models learn the difference between simplified theoretical predictions and the true measurements. Using climate data from September 2020, we tested this approach against traditional prediction methods, implementing various machine learning models to compare their performance. Our results reveal that delta learning consistently outperforms conventional approaches, especially when working with limited data and training time. These results suggest a more efficient path forward for weather prediction, as delta learning models can focus on learning the subtle patterns that theoretical methods miss rather than having to learn everything from scratch. These findings not only advance our understanding of machine learning in weather forecasting but also bring us closer to more accurate and reliable weather predictions that could benefit agriculture, urban planning, and climate science.**

I. Introduction

Across all fields of science, machine learning is increasingly a valuable tool for pattern recognition and predictive properties. Sharp improvement in model performance and data efficiency can be realized by coupling these architectures to underlying physical theory. This can be achieved through transfer learning, physics-informed loss functions, or delta machine learning, described in further detail in the following section. Under this framework, the parameters of the neural network are more efficiently allocated to learning patterns beyond what scientists already understand, saving training time and predictive capacity. Thus, these ”learn the difference” architectures prove especially valuable in the limit of low data.

In this work, we explore the value of delta learning for weather prediction. In this field, known climate trends as well as simple measurements such as current temperature can provide valuable aid to prediction of complex weather patterns, yet even the strongest models fail to predict weather just hours in advance. Climate data thus provides an exciting test ground to check whether the delta learning approach can be beneficial to the some of the most challenging predictive problems. Delta learning has hardly been explored, and its implementation presents notable risks that warrant careful consideration. While the approach can theoretically isolate and learn incremental changes, it may inadvertently propagate existing model errors when the base predictions are inaccurate. Furthermore, the decomposition of the prediction task into baseline and delta components could potentially mask important coupling effects between absolute weather states and their changes, leading to decreased model performance compared to traditional end-to-end learning approaches. This risk is particularly pronounced in chaotic systems like weather, where small errors in the baseline prediction might cascade into significant deviations in delta predictions.

We test multiple methods of model enhancement and explore the effect on accuracy of the model vs duration of training.

II. Background and Related Work

Delta machine learning refers to one way to combine theoretical and data driven methods to solve a wide variety of problems. Many problems in fields such as physics, chemistry, biology, etc can be solved approximately using methods that are based on theoretical models. Delta machine learning can aid in improving the accuracy or efficiency of these solutions. For instance, delta ML can help solve problems with high accuracy without the cost of high level methods by predicting the difference, or delta, between the output of a low level and high level method. Then, high precision results can be achieved by simply running the low level method and then feeding the result to the delta ML model. This technique has been successfully used to improve the accuracy of gas-phase electronic structure calculations [1]. Delta ML has also been shown to successfully correct errors in redox potentials calculated using density functional theory (DFT) and absorption energies calculated by time-dependent DFT when trained to predict the difference between the theoretical prediction and experimental result [2].

Delta ML methods as described above are not the only way to incorporate theory into data driven methods. For example, [3] provides a survey of works that have used “Theory Inspired Machine Learning” in a variety of ways. Some paradigms used in these works include theory-inspired feature engineering, theory-inspired model selection, and model regularization via theory. While these other approaches are interesting, this research focuses specifically on using delta ML to improve theoretical predictions.

III. Problem Setup

A. Dataset

The data used in this research comes from the ERA5 dataset, a comprehensive global climate and weather reanalysis dataset produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset provides a detailed record of atmospheric, oceanic, and land surface conditions dating back several decades. ERA5 is widely used for climate research, weather analysis, and environmental monitoring due to its high temporal and spatial resolution and extensive range of variables. ERA5 has an hourly temporal resolution, and a spatial resolution of 31 km.

The subset of ERA5 data used in this work is hourly data (ie: 24 data points per day) for $01 - 24$ September, 2020. The data points consist of features such as daily temperature, soil temperature, soil water content, and more (see Appendix: A for ERA5 data used).

B. Task Setup

In order to explore delta ML and its benefits, we use various training setups to train machine learning models on the task of predicting soil temperature. In particular, we compare the following four approaches:

1. "Uninformed end-to-end model": The model is given past soil temperature and predicts current soil temperature.
2. "Privileged end-to-end model": The model is given past soil temperature and current air temperature and predicts current soil temperature.
3. "Privileged end-to-end model": The model is given past soil temperature and any other useful features (eg: current air temperature, humidity, etc) and predicts current soil temperature.
4. "Delta model with theory": The model is given past soil temperature as well as the estimate for current soil temperature (based on theoretical models) and predicts the delta between the theoretical and true value of current soil temperature.

C. Theoretical Models

For our theoretical baseline, we implemented two physics-based soil temperature models described by Zheng et al. (1993): a base model for bare ground conditions and an enhanced model that accounts for vegetation cover. These models were selected for several key reasons. First, they are based on well-established physical principles governing soil temperature dynamics, including heat capacity and energy balance. Second, they require only readily available meteorological data (air temperature, precipitation) as inputs, making them practical for widespread application. Third, they explicitly account for important physical factors like snow cover and vegetation effects, allowing us to test the delta learning approach across varied environmental conditions[4].

C.1 Base Model The base model calculates soil temperature $F(J)$ for a given day J using the following equation:

$$F(J) = [A(J) - A(J-1)] * M_1 + E(J)$$

where:

$$\begin{aligned} A(J) &= \text{current day's air temperature} \\ A(J-1) &= \text{previous day's air temperature} \\ M_1 &= \text{rate scaler (0.1 with snow cover, 0.25 without)} \\ E(J) &= \text{regional soil temperature estimate} \end{aligned}$$

This formulation captures the fundamental physics that soil temperature changes are driven by air temperature differences, with the rate of change moderated by surface conditions.

C.2 Vegetation Model The vegetation-influenced model builds upon this foundation by incorporating the shading and insulating effects of plant canopy cover:

$$T(J) = T(J-1) + [A(J) - T(J-1)] * M_2 * exp(-K * LAI)$$

where:

$$\begin{aligned} T(J-1) &= \text{previous day's soil temperature} \\ M_2 &= \text{rate scaler (typically 0.25)} \\ K &= \text{light extinction coefficient (often 0.5 for forests)} \\ \text{LAI} &= \text{leaf area index} \end{aligned}$$

This enhanced model accounts for how vegetation modifies soil temperature dynamics through canopy shading (represented by the exponential LAI term) while maintaining the core temperature difference driving force."

IV. Methodology

A. Model architecture

Our soil temperature prediction model adapts the ConvLSTM architecture introduced by Shi et al. [5]. The original model architecture presented in the paper is a twinned LSTM model that uses convolutional layers, where the outputs of and hidden states of one model (the encoder) feeds into the second model (the forecaster). The primary motivation of this architecture is to enable a "latent space" to be encoded between the encoder and forecaster.

Although we have implemented and tested ConvLSTM as seen [5], we ultimately found that due to the nature of our dataset we would benefit from a simpler architecture that would not suffer from vanishing gradients. Our final architecture is a convolutional LSTM that feeds N input channels (this could be previous soil temperature, current air temperature, and/or the theoretical prediction) through a one or two-layer CNN and outputs 4 channels which become the input, output, forget, and candidate memory gates of the LSTM.

The input features are structured as a 4D tensor with dimensions (batch, channels, height, width), where:

- Batch dimension feeds in b "uncorrelated" sequences of data in parallel

- Channels contain the various meteorological features (soil temperature, air temperature, etc.)
- Height and width correspond to the spatial dimensions of our grid

In addition to the ConvLSTM network, we implemented a CNN architecture to investigate the ability of an ML model to improve upon our theoretical models. To implement this model we use a 4 layer CNN with batch normalization and a kernel size of 3. This network takes as input a 4D tensor with dimensions (batch, channels, height, width) and outputs a 4D tensor with dimensions (batch, 1, height, width) (the channel dimension has length 1), which is the model's prediction for the soil temperature of the next time step for each input in the batch. The features the model needs to make a prediction (including the theoretical model's prediction and the soil temperature of the previous time step) are concatenated as additional channels to the input tensor.

Finally, we also tested two additional, simpler models to demonstrate the effectiveness of the delta learning approach.

1. We looked simply at one positional coordinate in our data, and we used an LSTM architecture to predict the next soil temperature reading given the previous 12 readings.

2. We developed a fully connected neural network that used a small amount of time data, only looking at the past three hours of temperature, and used this to predict the soil temperature of the whole area of our land data.

In both of these approaches, we implemented two architecture options. The first was simply trained on soil data, and air temperature data was not incorporated at all. Then, the model was trained to learn the difference between the air temperature and the soil temperature, thus during prediction time, the model added the predicted difference to the known value of air temperature. It would be remiss for data scientists not to use the simple data values easily at hand when predicting soil temperature, and our model shows this very clearly.

B. Loss and Training Details

We train the CLSTM and CNN models using Mean Squared Error (MSE) loss to minimize the difference between predicted and actual soil temperatures. The models are trained using the Adam optimizer with an initial learning rate of 0.01. We implement a learning rate decay on training loss plateau, with a patience of 10 epochs and a factor of 0.5. We trained on 576 hours of data, with a coordinate map of size (51, 81).

For the delta learning approach, we modify the loss function to predict the difference between theoretical model estimates and actual values, rather than predicting absolute temperatures directly. As mentioned earlier, this allows the network to focus on learning the residual patterns not captured by physics-based models.

For CLSTM we found that the loss was rather sensitive to vanishing gradients and vanishing weights. To counter this, we introduced the following three auxiliary loss terms:

$$\ell(\hat{x}, y) = \ell_{\text{MSE}}(\hat{x}, y) + (\ell_{\text{spat}} + \ell_\sigma + \ell_\mu)$$

where ℓ_{spat} denotes spatial smoothness for $x \in \mathbb{R}^{m \times n}$, with δ denoting difference between neighbors (`torch.diff`) across rows and columns

$$\ell_{\text{spat}}(\hat{x}) = \frac{1}{m}\sum_{i=0}^{m} |\delta(\hat{x}_{i,*})| + \frac{1}{n}\sum_{i=0}^{n} |\delta(\hat{x}_{*,j})|$$

ℓ_σ punishes low variance / uniform output over x, with limiting ε for numerical stability

$$\ell_\sigma(\hat{x}) = \frac{1}{\sigma(\hat{x}) + \varepsilon}$$

and ℓ_μ is on the mean of the output compared to the target

$$\ell_\mu(\hat{x}, y) = |\mu(\hat{x}) - \mu(y)|$$

We found these loss terms to be ultimately necessary for reasonable performance of the model. Without them, the model would tend toward the average delta (which makes sense for MSE loss), which is about zero for this theoretical model.

V. Evaluation

The evaluation of our approach proceeds in three stages, (1) beginning with an assessment of the theoretical models' performance as a baseline for comparison, (2) followed by an exploration of pure ML models on predicting soil temperature , and (3) concluded with analyzing Delta-ML predictions to see if any accuracy is gained.

1. Assessing Theoretical Models We analyze the accuracy of both the base soil temperature model and the vegetation-influenced model using ERA5 data from September 2020, covering diverse geographical locations with varying environmental conditions.

To quantify model performance, we calculate the Root Mean Square Error (RMSE) between predicted and observed soil temperatures [See Figure 1].

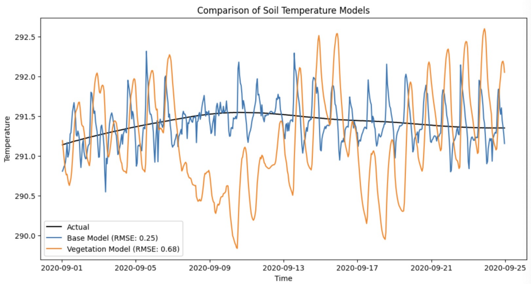


Fig. 1. Comparison of soil temperature models (Base Model and Vegetation Model) against actual measurements at 10 cm depth during September 2020, showing superior performance of the Base Model (RMSE: 0.25) compared to the Vegetation Model (RMSE: 0.68) in predicting daily temperature fluctuations and (seasonal trends).

The simpler formulation of the base model, which primarily relies on changes in air temperature and regional soil temperature estimates modulated by a binary rate scaler, demonstrates more stable predictions with an RMSE of 0.25K. In contrast, the vegetation model exhibits higher volatility (RMSE: 0.68K) due to its incorporation of multiple interacting environmental factors, including seasonal leaf area variations, light extinction coefficients, and soil temperature feedback. The exponential relationship between vegetation cover and soil temperature makes the model particularly sensitive to changes in canopy conditions, while its reliance on previous soil temperature calculations can lead to error propagation over time.

2. Assessing Pure ML Models

When training the CLSTM on the sequenced temperature data and the sequenced temperature data with current air temperature, the model had considerable trouble capturing both the smoothly varying regions of the temperature landscape and the overall distribution (mean and variance) of the data itself. Results from training, even with the auxiliary loss terms mentioned in section (IV-B), we still did not see reasonable results. We also tried learning this task with a much larger model capacity (up to five ConvLSTM units with up to 256 output feature channels), as well as with the full Encode-Forecast architecture proposed in [5]. Although we won't go into detail on this here, these efforts were not fruitful. Results from this task can be seen in figure 4.

3. Assessing Delta-ML Models We first used the simple CNN architecture to predict soil temperature at time $t+1$ given soil temperature at time t, the theoretical prediction for soil temperature at time $t+1$, and the air temperature at time $t+1$. The results of the CNN are shown in figures 5 and 6. These results demonstrate that the simple CNN architecture was able to provide an improvement over the theoretical model's predictions in terms of the L1 Loss.

The results from the fully connected network show striking value of the delta learning approach. In comparison to the results from solely learning on soil temperature data, the learning on temperature difference achieved far lower error in fewer epochs, stemming from the strong relationship between soil and air temperature. Had climate scientists ignored their ability to incorporate air temperature and solely attenmpted to learn soil patterns, valuable computer time, model tunability, and predictive capacity would have been lost.

Training CLSTM on the delta temperature data saw some success as well. Reflecting on our observations of the quality of the theoretical model fit to the true soil temperature data, we can see that the theoretical predictions fluctuate at a relatively regular frequency and amplitude about the true temperature. Figure 2 visualizes the predictions of our model, which is in the form of expected gap between the theoretical model and the true temperature. Figure 3 visualizes what these predictions mean in practice by adding them to the theoretical model.

We also tried including air temperature data and the base theoretical predicted values as additional input channels to the CLSTM. However, these versions of the training suffered from misleading noise (air temperature) and correlation between dimensions (theoretical values). Training with air temperature led to considerably more noisy predictions, and training with the actual theoretical values led to vanishing gradients within the first few weight updates.

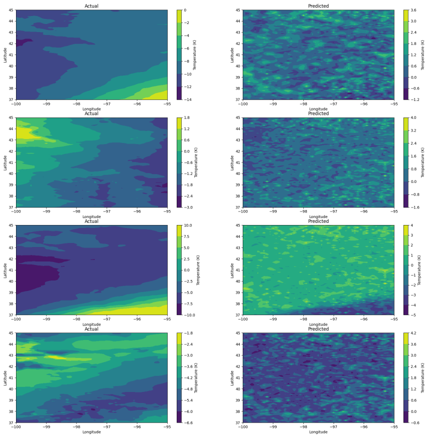


Fig. 2. Delta predictions of the CLSTM trained on our dataset. Although the data is not as smooth as the label, we observed that some general trends were captured.

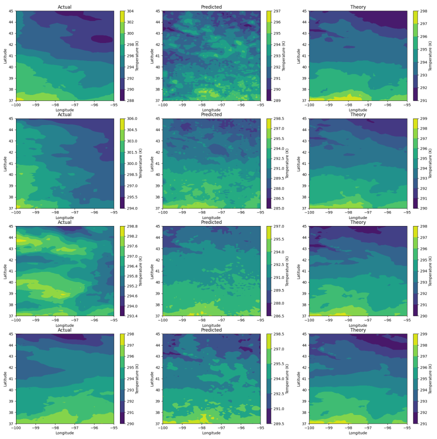


Fig. 3. Full predictions of the delta-CLSTM. Left column is actual data, middle is predicted, and right is theoretical data. There is still considerable noise.

The results of the difference learning model show striking improvement over that of the temperature learning directly, by offering a simpler pattern to learn. In the limit of low epochs, the difference learning consistently performs far better, hinting that the model's parameters are more efficiently used in capturing the soil patterns and not wasting training time on learning daily temperature fluctuations.

VI. Conclusion

Our model provides insight into more efficient data training to a variety of scientific inquiries with machine learning. There are often excellent cheap data sources to aid in more complicated prediction.

As seen in the delta models, the theoretical models commonly available often have a patterned or predictable variance from the true data that can be approximated via a neural network. Oftentimes learning the data trajectory relative to the prediction can therefore be a simpler task.

Reflecting on our implementation, the CLSTM could benefit from a more direct estimation of parameters such as overall temperature range of the delta-image. This parameter seemed to fluctuate somewhat independently of the spatial variance of the sequence, and predicting them separately may save model capacity. It is also possible that if we had access to data that was discretized more finely over time (every 15 minutes, every 30 minutes) the CLSTM hidden states could be of more help to the output predictions.

Related to this, both the CNN and the fully-connected networks that we implemented also showed some success on the dataset, signalling that the previous hour or two may be enough to predict the next hour for the soil temperature task we were interested in.

Because the theoretical data remains roughly centered on the true temperature, a truer theoretical model could take the average over a short time window of the original theoretical model to provide the trajectory of the centered path of the oscillation. It would be useful to see how the models then perform on this delta instead.

As is often the case, much of the work of this project was in the “last 10 percent” of training, retraining, visualizing results, and tweaking the loss. This project was a reminder of the power of auxiliary loss terms in guiding learning away from suboptimal minima: specifically, the variance penalty on the CLSTM loss helped “kick” the model out of predicting zero-everywhere, and the inverse relation also helped do this in a scaled way. There is a balance between realizing a model architecture doesn’t work and believing the loss and training params just need to be adjusted. Sanity checks on small datasets or on reduced sizes can often help answer this question and was helpful in our learning as well.

VII. Appendix

A. *ERA5 Data Used*

Soil Temperature Data

1. **stl1 - Surface soil temperature (level 1)**
   - Units: Kelvin
   - Represents: Temperature at soil surface (0.0m depth)
2. **stl4 - Deep soil temperature (level 4)**
   - Units: Kelvin
   - Represents: Temperature at 100cm below surface

Surface Conditions

1. **lai_lv - Leaf Area Index for Low Vegetation**
   - Units: m^2/m^2
   - Represents: Ratio of leaf area to ground area for low vegetation
2. **lai_hv - Leaf Area Index for High Vegetation**
   - Units: m^2/m^2
   - Represents: Ratio of leaf area to ground area for high vegetation
3. **t2m - 2-meter Temperature**
   - Units: Kelvin
   - Represents: Air temperature at 2 meters above ground
4. **tp - Total Precipitation**
   - Units: meters
   - Type: Accumulated value
   - Represents: Total precipitation amount
5. **skt - Skin Temperature**
   - Units: Kelvin
   - Represents: Temperature at Earth’s surface/skin layer

Snow Coverage

1. **snowc - Snow Cover**
   - Units: Percentage (%)
   - Represents: Percentage of ground covered by snow

B. Figures

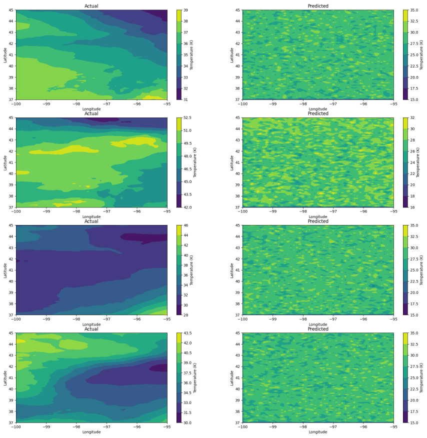


Fig. 4. Example of results for training on the raw temperature sequences. Similar results occurred for training with concatenated air temperature.

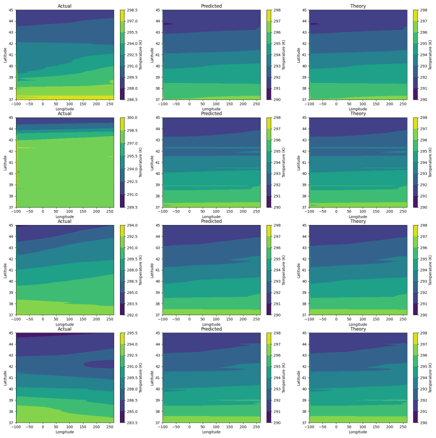
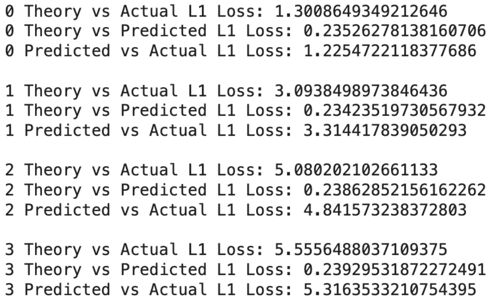
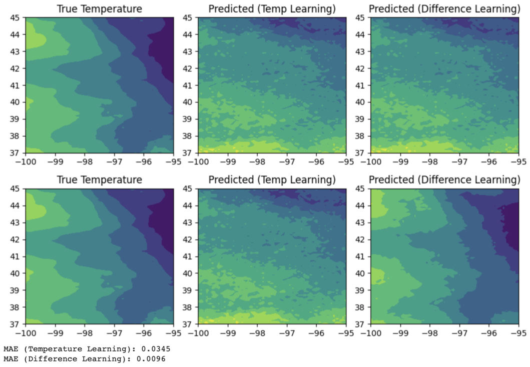


Fig. 5. Comparison of actual, theoretical, and CNN prediction of the soil temperature for 4 randomly chosen time points.



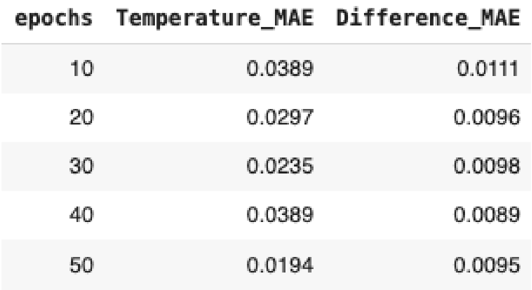
```
0 Theory vs Actual L1 Loss: 1.3008649349212646
0 Theory vs Predicted L1 Loss: 0.23526278138160706
0 Predicted vs Actual L1 Loss: 1.2254722118377686

1 Theory vs Actual L1 Loss: 3.0938498973846436
1 Theory vs Predicted L1 Loss: 0.23423519730567932
1 Predicted vs Actual L1 Loss: 3.314417839050293

2 Theory vs Actual L1 Loss: 5.080202102661133
2 Theory vs Predicted L1 Loss: 0.23862852156162262
2 Predicted vs Actual L1 Loss: 4.841573238372803

3 Theory vs Actual L1 Loss: 5.5556488037109375
3 Theory vs Predicted L1 Loss: 0.23929531872272491
3 Predicted vs Actual L1 Loss: 5.3163533210754395
```

Fig. 6. Quantitative comparison of the theoretical and CNN prediction errors relative to the actual soil temperature. Shows the L1 loss between these various quantities for 4 randomly chosen time points (same time points as in Figure 2).



```
MAE (Temperature Learning): 0.0345
MAE (Difference Learning): 0.0096
```

Fig. 7. Performance of the fully connected network on soil temperature learning and temperature difference learning. The difference learning model shows greater alignment with the true temperature and avoids unrealistic noisy temperature patterns.



epochs	Temperature_MAE	Difference_MAE
10	0.0389	0.0111
20	0.0297	0.0096
30	0.0235	0.0098
40	0.0389	0.0089
50	0.0194	0.0095

Fig. 8. Performance Comparison of Temperature Prediction Models Across Training Epochs: Direct Temperature vs. Difference Learning Approaches

References

[1] R. Ramakrishnan, M. Rupp, P. O. Dral, and O. A. Lilienfeld, ”Big Data meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach,” *arXiv preprint arXiv:1503.04987*, 2015.

[2] X. Chen, P. Li, E. Hruska, and F. Liu, ”Δ-Machine learning for quantum chemistry prediction of solution-phase molecular properties at the ground and excited states,” *Phys. Chem. Chem. Phys., 2023,25, 13417-13428*, 2023.

[3] J. Hoffer, A. B. Ofner, F. M. Rohrhofer, M. Lovrić, R. Kern, S. Lindstaedt, and B. C. Geiger, ”Theory-inspired machine learning—towards a synergy between knowledge and data,” *Weld World 66, 1291–1304 (2022). https://doi.org/10.1007/s40194-022-01270-z*, 2022.

[4] D. Zheng, E. R. Hunt Jr, S. W. Running, “A daily soil temperature model based on air temperature and precipitation for continental applications,” *Clim. Res.* **2**, 183-191 (1993). https://www.int-res.com/articles/cr/2/c002p183.pdf

[5] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” arXiv:1506.04214v2 [cs.CV], (2015).