Predicting Urban Heat Island Hotspots in NYC

Shunsuke Akamatsu (sa4469) and Arnav Agarwal (aa5507)

May 12, 2025

Abstract

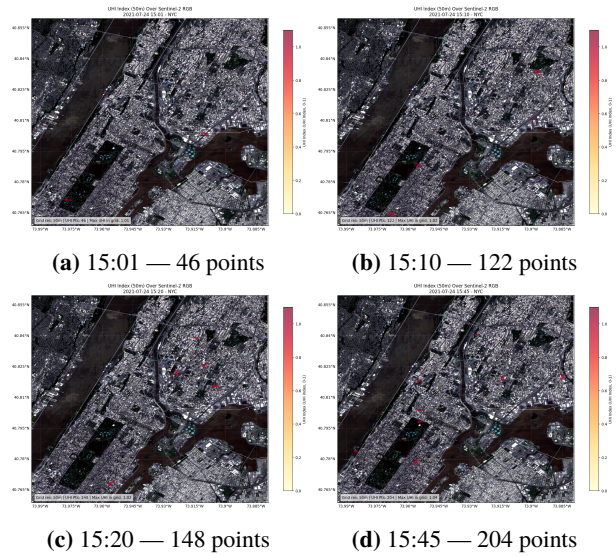
This project tackles Urban Heat Island (UHI) prediction in NYC using a multi-stage approach. Initially, traditional machine learning models (Random Forest, XGBoost) on a diverse dataset revealed key feature importances: air temperature surpassed LST, and building metrics proved more indicative than simple LULC indices. We then conducted a feature importance strategy to inform the development of deep learning models. A static feedforward model, tested with random data splits, showed the Clay Vision Transformer (validation $R^2 = 0.63$) outperforming LULC indices ($R^2 = 0.40$). Recognizing random splitting's limitations for heteroskedastic temporal data, sequential splitting was emphasized, under which the static model's performance significantly dropped (best validation $R^2 = 0.16$ using LST). Our main deep learning contribution, a branched temporal model using Clay for static features and ConvLSTM for dynamic weather sequences, achieved a validation $R^2 = 0.11$ on sequential data. This work highlights the challenges of UHI prediction with sparse data and proposes a deep learning pipeline with foundation models as a promising approach for spatiotemporal environmental forecasting.

1 Introduction

UHI phenomenon refers to elevated temperatures in urban areas compared to surrounding rural regions, driven by heat-retaining materials like concrete and asphalt, dense urban form, limited vegetation, and anthropogenic heat emissions. UHI intensifies energy demands and poses health risks [7]. As over 55% of the global population lives in cities – a figure projected to rise to 68% by 2050 – understanding and mitigating UHI has become essential for sustainable urban development. Accurate hotspot prediction enables targeted cooling interventions to improve livability. This project, developed in the context of the 2025 EY Open Science AI and Data Challenge [3], focuses on grid-level UHI prediction in New York City. We aim to build a scalable framework using both traditional and deep learning methods. Starting with feature correlation analysis and baseline models (Random Forest, XGBoost), we analyze feature importance to inform model design. Building on this, we explore deep learning strategies, including a static feedforward model and a branched temporal model using pretrained encoders (e.g., Clay [2]) and temporal weather grids. Architectures such as U-Net and ConvLSTM [9, 10] are employed to better capture spatial and temporal dynamics.

2 Related Work

UHI studies have traditionally relied on LST data obtained from satellites such as Landsat to map and analyze urban thermal patterns [1]. However, LST data alone provides limited UHI predictive power due to its coarse resolution and surface-level focus. Recent studies have called for integrated approaches that combine ground-based air temperature data with remote sensing and urban morphological features to improve UHI prediction [5]. Our work follows this direction by going beyond LST dependence and integrating diverse spatial and environmental datasets. In recent years, machine learning has gained popularity in UHI prediction, with models like Random Forest and XGBoost valued for handling nonlinear relationships and interactions among multiple predictors. Deep learning approaches are emerging as promising



(a) 15:01 — 46 points **(b)** 15:10 — 122 points

(c) 15:20 — 148 points **(d)** 15:45 — 204 points

Figure 1: UHI observation points (red) overlaid on Sentinel-2 imagery at different timestamps.

tools in environmental modeling, including UHI prediction. Particularly ConvLSTM models are increasingly applied to capture spatiotemporal dependencies in sequential environmental data [10]. Additionally, recent innovations in pretrained geospatial foundation models, such as Clay [2], Prithvi [4], and SatMAE [6], have enhanced data efficiency and representation learning, especially in settings with limited labeled data. These models learn rich representations from vast amounts of unlabeled satellite imagery that can be fine-tuned for specific downstream tasks. This project contributes by implementing a CNN-based pipeline with a Clay pretrained encoder and comparing its performance to traditional ML models, offering insights into their respective strengths and limitations.

3 Methods

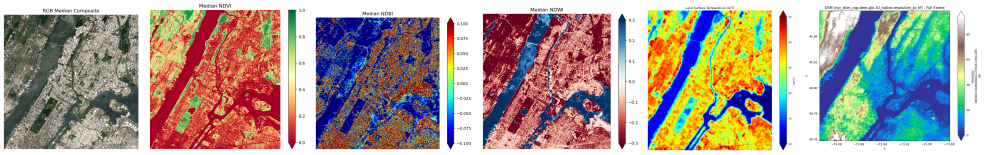
3.1 Datasets Utilized

UHI data: The UHI index dataset was collected on the afternoon of July 24, 2021, in Manhattan and the Bronx. Each observation includes latitude, longitude, and timestamp, with a spatial resolution of 50 meters. However, coverage is limited—both spatially and temporally—to specific neighborhoods and a single hour (15:00–16:00), posing challenges for building generalizable ML models. A temporal mismatch between satellite and ground observations further complicates data integration and time-series modeling. Figure 1 shows the spatial distribution of UHI points overlaid on Sentinel-2 imagery at several time slices.

Satellite Imagery: Sentinel-2 imagery provided three key land surface indices: Normalized Difference Vegetation Index (NDVI), Normalized Difference Built-up Index (NDBI), and Normalized Difference Water Index (NDWI), while Landsat-8 imagery supplied Land Surface Temperature (LST) data. Representative Sentinel-2 and Landsat-8 images illustrating land surface indices and temperature inputs are shown in Figure 2.

Weather Data: Intraday weather observations were collected from two meteorological stations (Bronx and Manhattan), including air temperature, relative humidity, wind speed and direction, and solar flux. To address spatial heterogeneity, we applied Inverse Distance Weighting (IDW) interpolation. Given the latitude and longitude (ϕ, λ) of a UHI grid cell, and the coordinates of the Bronx and Manhattan stations (ϕ_B, λ_B), (ϕ_M, λ_M), the squared distances are computed as:

$$d_B^2 = (\phi - \phi_B)^2 + (\lambda - \lambda_B)^2, \quad d_M^2 = (\phi - \phi_M)^2 + (\lambda - \lambda_M)^2 \tag{1}$$



(a) Original **(b)** NDVI **(c)** NDBI **(d)** NDWI **(e)** LST **(f)** DSM

Figure 2: Representative sample images from Sentinel-2, Landsat-8, and DSM.

The interpolated value V of any weather variable is then calculated as:

$$V = \frac{V_B/(d_B^2 + \varepsilon) + V_M/(d_M^2 + \varepsilon)}{1/(d_B^2 + \varepsilon) + 1/(d_M^2 + \varepsilon)} \tag{2}$$

where V_B and V_M are the observed values from Bronx and Manhattan respectively, and $\varepsilon = 10^{-6}$ is a small constant added to prevent division by zero. Wind direction (in degrees) was further transformed into two continuous components for model compatibility:

$$\text{wind\_dir\_sin} = \sin(\theta \cdot \pi/180), \quad \text{wind\_dir\_cos} = \cos(\theta \cdot \pi/180) \tag{3}$$

Building Data: Building footprint polygons were obtained from a KML dataset and processed via a geospatial pipeline. Coordinates were projected to a metric system to allow precise spatial calculations. To characterize the built environment around each UHI observation, we computed two metrics within a 100-meter buffer: (1) the number of intersecting building polygons and (2) the total built area in square meters. For each UHI point, we constructed a circular buffer, identified intersecting buildings, counted the polygons, and summed their geometric areas.

Elevation Data: A Digital Surface Model (DSM) from the Copernicus DEM GLO-30 collection (30m resolution), accessed via Planetary Computer, was used. Initial explorations with NASADEM revealed it was not a bare-earth model for this urban area, as it demonstrated high values corresponding to the buildings in downtown manhattan. While a bare-earth Digital Elevation Model (DEM) would be ideal for capturing terrain height, a suitable source was not readily found on Planetary Computer. Future work could explore integrating the DEM provided by the New York State GIS Program Office [8]. Figure 2f visualizes the DSM we used.

Metadata: Each UHI observation included temporal metadata (hour, minute) and spatial metadata (grid coordinates). At each timestep for the deep learning model, this was fed into the pretrained encoder for its spatial and temporal embeddings. We used the latitude and longitude at the center of the observation grid, normalised into sine and cosine components, as the spatial information for the Clay encoder and the full datetime for as temporal input.

3.2 Data Preprocessing and Feature Analysis

For traditional tabular models, this involved loading UHI data, integrating and interpolating weather observations using IDW, calculating satellite-derived indices for UHI point locations, computing building metrics within a 100m radius from KML footprints, and encoding temporal features. For deep learning, the inputs were structured as gridded tensors centered on UHI locations. These included Sentinel-2 image patches, 2D weather maps, temporal encodings, and DSM layers. In the branched temporal model, sequences of these grids, together with an autoregressive UHI history channel, served as the input. Data loading, normalization, and batching were handled by custom PyTorch Dataset classes, and sparse supervision was addressed using masked loss functions. The final integrated feature set for tabular models provided a rich basis for predictive modeling. A feature correlation analysis (Figure 3) revealed that NDVI, NDWI, and NDBI were highly correlated. Air temperature and relative humidity exhibited a strong negative correlation, consistent with atmospheric dynamics. Notably, NDVI showed a

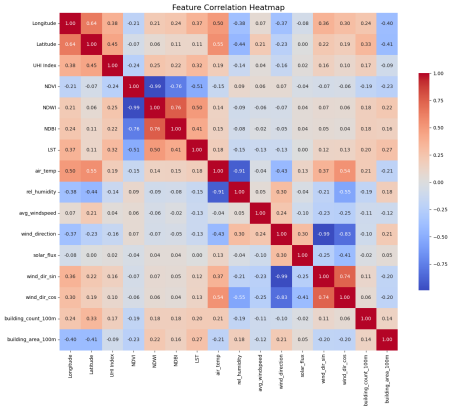


Figure 3: Feature correlation matrix summarizing the relationships among key variables in the tabular dataset.

negative correlation with the UHI Index ($r = -0.24$), suggesting vegetation's cooling effect, while building count had only a weak positive correlation ($r \approx 0.17$), indicating built density alone is not a strong UHI predictor in this study area.

3.3 Tabular Machine Learning Approach

As a baseline and primarily to conduct feature importance analysis, we implemented two classical machine learning models: Random Forest Regressor and XGBoost Regressor. Both models were trained to predict the continuous UHI index using the comprehensive tabular feature set created in the preprocessing stage. This feature set included NDVI, NDBI, NDWI, LST, interpolated air temperature, relative humidity, wind speed, wind direction (sine and cosine), solar flux, building count, building area, DSM, and temporal sine/cosine embeddings. Data for these models was split into training and testing sets. We explored both random splitting and sequential (temporal) splitting. Given that UHI data has a temporal component, sequential splitting provides a more realistic evaluation of model generalization. Feature importance, assessed using permutation importance and SHAP (SHapley Additive exPlanations) analysis, was mainly derived from models trained with a random split to identify strong feature signals, which subsequently informed our deep learning design choices.

3.4 Deep Learning Architectures

To capture complex spatial and temporal patterns, we developed two primary deep learning architectures. These models leverage gridded representations of satellite imagery, weather data, and elevation.

Our initial model, a static, feedforward model, was used to validate the feature processing pipeline and conduct ablations, such as comparing direct LULC indices with embeddings from the pretrained Clay Vision Transformer [2]. The model takes multiple gridded inputs: weather features, optional static features (e.g. LST, DSM), and optionally Sentinel-2 imagery processed by a Clay-based feature extractor with BatchNorm and a 1×1 convolutional projection. These feature maps are concatenated and passed to a configurable "feature head." For initial testing, we used a simple stack of 2D convolutions, but a U-Net-like decoder [9] with skip connections and upsampling performed better and was adopted for further experiments. Finally, a 1×1 convolution projects the decoder's output to a single-channel UHI prediction map.

To capture the spatiotemporal nature of UHI, we developed the Branched Temporal Model, which processes static and dynamic inputs separately. The static branch encodes Sentinel-2 imagery using the Clay ViT extractor with BatchNorm and a 1×1 projection, while other static grids like DSM are projected separately. The dynamic branch takes sequences of gridded weather data and the previous timestep's UHI, passing them through a multi-layer ConvLSTM [10]. A temporal attention mechanism aggregates hidden states via a softmax-weighted sum,

Table 1: Performance of ML Models under Different Data Splitting Strategies and Feature Configurations

Split Type	Features	Model	R^2	RMSE	MAE
Random	No Lag	Random Forest	0.9486	0.0037	0.0025
Random	No Lag	XGBoost	0.9260	0.0044	0.0033
Sequential	No Lag	Random Forest	0.1763	0.0138	0.0110
Sequential	No Lag	XGBoost	0.2101	0.0136	0.0111
Sequential	With Lag	Random Forest	0.2496	0.0132	0.0103
Sequential	With Lag	XGBoost	0.1766	0.0138	0.0110

highlighting informative timesteps. The resulting feature map is projected and concatenated with static features, then passed to a U-Net decoder [9] with bicubic upsampling and a final 1×1 convolution to produce the UHI prediction. All models used masked loss and the AdamW optimizer.

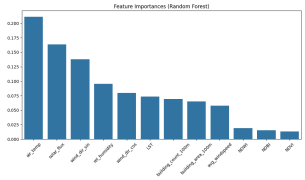
4 Results and Discussion

4.1 Tabular Machine Learning Models

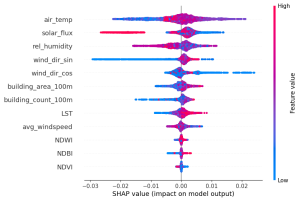
Performance under Different Temporal Splitting Strategies We first evaluated the models using a standard random train-test split (80-20, shuffled). As shown in Table 1, both Random Forest and XGBoost achieved high R^2 scores (> 0.92) with low RMSE and MAE values, indicating strong performance when temporally similar data points appear in both the training and test sets. However, this setup risks overestimating generalization due to temporal leakage, as the model may encounter test samples that are closely correlated in time with those used for training. To better assess forecasting ability, we adopted a sequential splitting strategy, where the first 80% of timestamps were used for training and the remaining 20% for testing. Under this more realistic setting, model performance dropped significantly (e.g., R^2 from $\sim$0.95 to $\sim$0.2), reflecting the difficulty of predicting temporally unseen data. This suggests that the high spatiotemporal variability of UHI, along with unobserved microclimatic factors and human activity shifts, challenges short-term forecasting with the current features. To address this, we also tested whether including the UHI value from the previous timestamp as an additional input could help. As shown in Table 1, this led to a modest improvement for the Random Forest model (R^2 increased from 0.1763 to 0.2496), indicating that recent UHI history provides some useful context. However, the overall improvement remained limited, highlighting the need for richer temporal or spatial representations.

Feature Importance and Insights The primary purpose of the traditional ML models was to conduct feature importance analysis. Based on permutation importance and SHAP analyses from models trained with random splitting (Figure 4), several key insights emerged. First, dynamic weather conditions, particularly air temperature and solar flux, were identified as the most influential predictors. Among these, air temperature consistently showed higher importance than land surface temperature, emphasizing the value of real-time meteorological measurements. In addition, engineered building features, such as building count and building area within a 100 m radius, exhibited greater predictive power than raw land-use and land-cover indices like NDVI, NDWI, and NDBI. This suggests that higher-level representations of urban structure contribute more meaningfully to UHI prediction than spectral indices alone. Finally, the relatively lower importance of LULC indices led us to adopt more expressive image encoders in the deep learning models. In particular, we leveraged pretrained models like Clay to extract rich spatial features from satellite imagery, going beyond simple index-based representations.

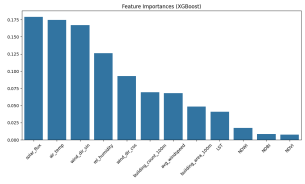
These findings shaped our deep learning strategy, emphasizing dynamic weather inputs and detailed representations of land cover and the built environment. The sharp performance drop in sequential splits further highlighted the need for models capable of learning spatiotemporal dependencies, which deep learning architectures are well suited to address.



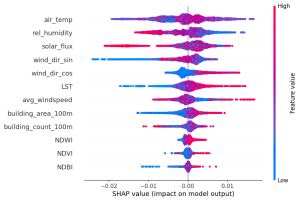
(a) RF - Permutation importance



(b) RF - SHAP analysis



(c) XGB - Permutation importance



(d) XGB - SHAP analysis

Figure 4: Feature importance results from Random Forest (RF) and XGBoost (XGB), trained with random split.

4.2 Deep Learning-Based Models

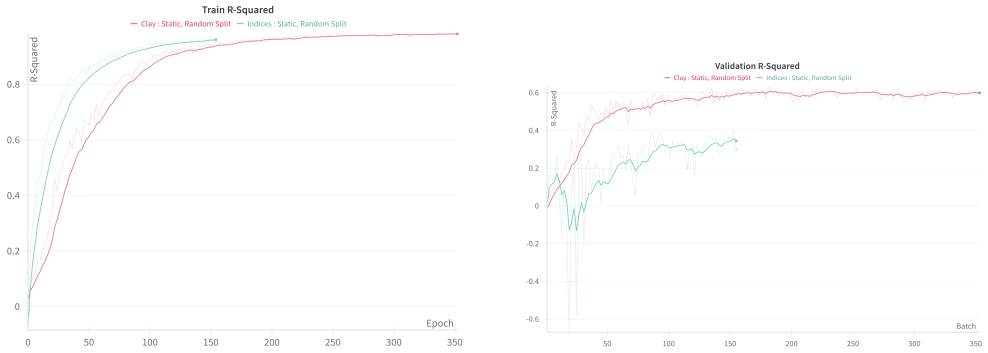
Our deep learning models aimed to capture more complex patterns than traditional methods, with performance summarized in Table 2. The Static Feedforward Model, on a **random split**, showed the Clay pretrained encoder (Validation $R^2 = 0.63$) outperforming LULC indices (Validation $R^2 = 0.40$), supporting the utility of foundation models despite some overfitting (Train R^2 0.98 and 0.96 respectively). Example learning curves are in Figure 5.

Transitioning to a sequential split, the Static Feedforward Model's performance dropped significantly. The Clay encoder yielded a validation R^2 of 0.06, while LULC indices achieved 0.08. An ablation using LST instead of air temperature (with Clay) provided the best static sequential result (Validation $R^2 = 0.16$), suggesting LST's stability might be advantageous for non-recurrent architectures in temporal forecasting. Severe overfitting persisted (Figure 6).

The Branched Temporal Model, designed for explicit spatiotemporal modeling, also used a sequential split. The Clay encoder configuration (Validation $R^2 = 0.11$) outperformed LULC indices (Validation $R^2 = 0.08$). This result, while modest, surpassed the Static Feedforward Model with similar inputs ($R^2 = 0.06$), indicating benefits from the temporal architecture (Figure 7). Across all deep learning experiments on sequential splits, a notable gap between high training R^2 and low validation R^2 values highlighted significant overfitting, a common issue with limited, variable time-series data.

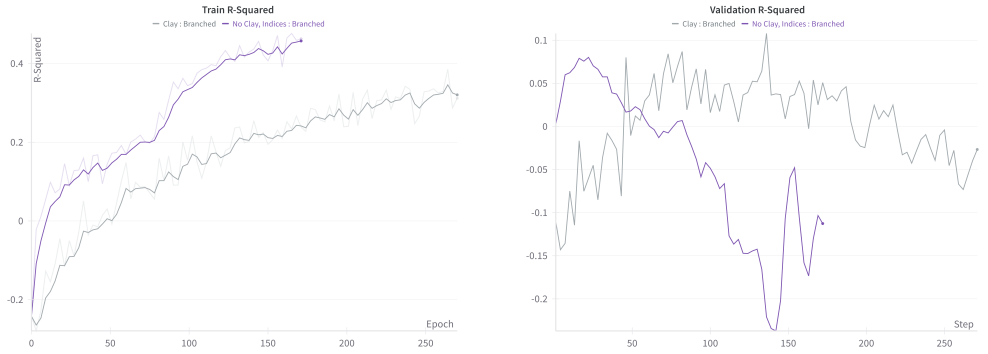
Table 2: Summary of Deep Learning Model Performance

Model Configuration	Split Type	Train R^2	Validation R^2	Validation RMSE
Static- LULC Indices	Random	0.96	0.40	0.0118
Static- Clay Encoder	Random	0.98	0.63	0.0092
Static- LST (no AirTemp)	Sequential	0.34	0.16	0.0141
Static- LULC Indices	Sequential	0.94	0.08	0.0147
Static- Clay Encoder	Sequential	0.16	0.06	0.0150
Branched- LULC Indices	Sequential	0.46	0.08	0.0148
Branched- Clay Encoder	Sequential	0.39	0.11	0.0145



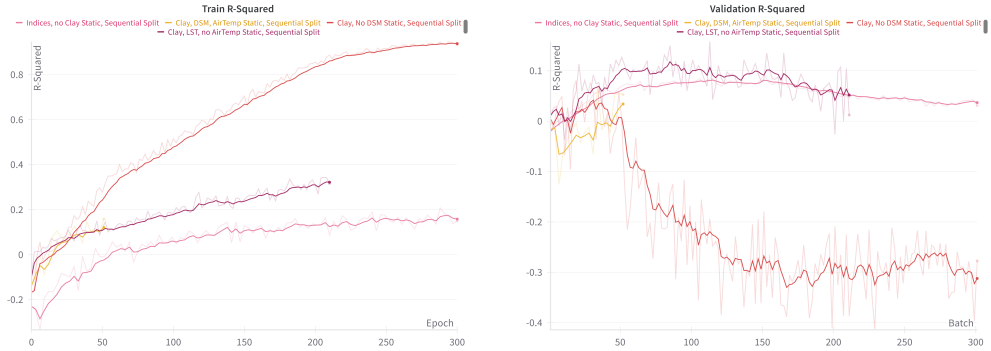
(a) Train R^2 (Random, Clay) **(b)** Validation R² (Random, LULC)

Figure 5: Example R² learning curves for Static Feedforward Model on random split.



(a) Train R² (Sequential, Clay) **(b)** Validation R² (Sequential, Clay)

Figure 7: Example R² learning curves for the Branched Temporal Model (with Clay encoder) on sequential split.



(a) Train R² (Sequential, Clay) **(b)** Validation R² (Sequential, Clay)

Figure 6: Example R² learning curves for Static Feedforward Model (Clay Encoder) on sequential split.

5 Conclusion

This project developed and evaluated a range of models for predicting UHI intensity in NYC using diverse spatial and temporal data. Traditional machine learning models (Random Forest, XGBoost) performed well under random splits ($R^2 > 0.92$), identifying key predictors like air temperature and solar flux, but their performance dropped sharply on sequential splits

($R^2 \approx 0.2$), highlighting the challenges of generalization with limited and sparse UHI data. Deep learning models, including a static feedforward model and a branched temporal model with ConvLSTM and attention, showed modest improvements in sequential settings, especially when incorporating pretrained encoders like Clay. However, overfitting remained a concern, and overall performance was constrained by the dataset's narrow temporal and spatial scope. Future work should focus on acquiring longer-term, citywide UHI data, integrating additional dynamic inputs such as traffic and building energy usage, and exploring advanced architectures and pretrained models. Improved evaluation strategies, regularization, and interpretability tools will also be crucial for developing robust and actionable UHI forecasting systems.

6 Limitations and Future Work

This study faced several limitations, primarily due to the restricted spatiotemporal coverage of the UHI dataset, constraining generalisability and reducing model performance under sequential splits. Although a DSM was used, the absence of a bare-earth DEM hindered the model's capacity to separate true elevation from surface structures. We conducted exploratory data analysis of the UHI values themselves well into our model development journey. This is a takeaway for future work, as doing so early would have guided us towards more sample-efficient models. Furthermore, only a limited set of hyperparameters was explored, we chose configurations with generally improving performance. Future work should prioritise acquiring broader, long-term UHI datasets across cities to support generalisable models, while integrating high-resolution DEMs and dynamic data sources. Investigating advanced other pretrained encoders, spatiotemporal architectures such as GNNs, and attention-based models is worthwhile, alongside more comprehensive hyperparameter optimisation. Alternative modelling strategies—like extracting features from the current encoder and feeding them into a simpler MLP for pointwise prediction—could be promising. Further improvements could stem from spatiotemporal cross-validation, interpretability methods (e.g., SHAP, attention maps), incorporating more dynamic features (such as traffic data) and uncertainty quantification to enhance real-world applicability and robustness.

References

[1] Cátia Rodrigues de Almeida et al. Study of the urban heat island (uhi) using remote sensing data/techniques: A systematic review. *Environments*, 8(10):105, 2021.

[2] Clay Foundation. Clay: An open source ai model for earth. https://clay-foundation.github.io/model/, 2023. Accessed: May 11, 2025.

[3] Ernst & Young Global Limited. The 2025 ey open science ai and data challenge: Cooling urban heat islands, 2025. Available at https://challenge.ey.com/challenges/.

[4] Ritwik Mukherjee et al. Prithvi-100m: A foundation model for remote sensing. 2023.

[5] SangHyeok Lee et al. Multidisciplinary understanding of the urban heating problem and mitigation: a conceptual framework for urban planning. *International Journal of Environmental Research and Public Health*, 19(16):10249, 2022.

[6] Yezhen Cong et al. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *Advances in Neural Information Processing Systems*, pages 16418–16433, 2022.

[7] Jeremy S Hoffman et. al. The effects of historical housing policies on resident exposure to intra-urban heat: a study of 108 us urban areas. *Climate*, 8(1):12, 2020.

[8] New York State GIS Program Office. New york state DEM data. https://gis.ny.gov/elevation-data/, 2023. Accessed: May 11, 2025.

[9] Olaf Ronneberger et. al. U-net: Convolutional networks for biomedical image segmentation.

[10] Xingjian Shi et. al. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.