EECE 693: Neural Networks, Spring 2025
30 April 2025

AMERICAN UNIVERSITY OF BEIRUT
MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

Predicting NYC Land Surface Temperature from Satellite Imagery and Meteorology

Ahmad Hlayhel, **John Lahoud**, **Malak Ammar**, **Nour Shammaa**

Maroun Semaan Faculty of Engineering and Architecture, American University of Beirut (AUB)

ABSTRACT

Land Surface Temperature (LST) is critical for understanding urban heat island (UHI), capturing spatial variations in urban heating influenced by factors such as land cover, vegetation, urban geometry, and meteorological conditions. This paper reviews recent literature on LST prediction and gap-filling techniques, categorizing them into four main classes: temporal models, spatial models, spatiotemporal models, and physics-informed machine learning. We developed a spatio-temporal deep learning framework with physical constraints to predict daily 30 m, 16-day LST in New York City, aiming to produce temporally continuous and physically realistic datasets for urban heat analysis. We compared the performance of various deep-learning architectures, including convolutional neural networks (CNNs), U-Nets, Visual Transformers (ViTs), and a physicomechanical Transformer that integrates physical constraints into its loss function. Our findings indicate that integrating spatial, temporal, and physical priors significantly improves the prediction of LST in intricate urban settings, paving the way for more efficient heat-mitigation approaches.

KEYWORDS

deep learning, spatiotemporal modeling, transformer network, urban climate, physical constraints

1 Introduction

Urban areas around the world experience significant warming compared to surrounding regions, known as the Urban Heat Island (UHI) effect. This warming impacts energy use, air quality, public health, and urban sustainability [1]. Satellite-derived Land Surface Temperature (LST) is critical for understanding UHI, capturing spatial variations in urban heating influenced by factors such as land cover, vegetation, urban geometry, and meteorological conditions [2]. High-resolution sensors like Landsat provide detailed LST data (30 m), enabling precise urban heat studies. However, Landsat's 16-day revisit cycle and additional issues such as atmospheric interference, sensor malfunctions, and contamination from shadows or haze result in sparse and irregular LST time series [3].

These data gaps prevent accurate monitoring of rapid thermal events like heat waves or sudden temperature changes and hinder understanding of UHI dynamics across seasons and years. Although recent studies increasingly use spatiotemporal modeling approaches to fill LST gaps and produce seamless daily high-resolution maps [4], few methods effectively combine spatial, temporal, and physical information to generate realistic predictions in complex urban environments. Developing such models remains challenging due to the intricate interactions among meteorological variables, characteristics of the urban landscape, and temporal dynamics.

This paper reviews recent literature on LST prediction and gap-filling techniques to establish a foundation for our approach. We categorize previous works into four main classes: temporal-only methods, spatial-only methods (including downscaling), spatiotemporal architectures (such as Transformers and U-Net CNNs), and physics-informed machine learning. Building on these insights, we develop a spatio-temporal deep learning framework with physical constraints to predict daily 30 m LST in New York City, aiming to produce temporally continuous and physically realistic datasets for urban heat analysis.

2 Literature Review

In this section, we review methods for Land Surface Temperature (LST) prediction and gap filling, categorizing them into four main classes: temporal models, spatial models, spatiotemporal models, and physics-informed machine learning.

2.1 Temporal Models

Temporal approaches treat LST evolution at a given location as a time series problem, ignoring the spatial context. Traditional statistical methods such as ARIMA and SARIMA have been used to interpolate or forecast missing LST observations by fitting periodic trends to sparse datasets [5]. Harmonic regression methods that use LASSO regularization have further improved gap filling at the pixel level, particularly for Landsat data.

However, purely statistical models often fail to capture sudden non-linear thermal events. To address this, machine learning

models such as Long Short-Term Memory (LSTM) networks offer improved modeling of complex temporal dependencies and abrupt changes [6]. Recent developments such as Temporal Fusion Transformers (TFTs) [7] combine interpretability and long-sequence modeling, offering enhanced LST forecasting. Nonetheless, a key limitation remains: temporal models neglect spatial correlations that are important in urban thermal dynamics.

2.2 Spatial Models

Spatial models focus on predicting LST at a given pixel based on neighboring spatial information. Traditional interpolation techniques, such as inverse distance weighting and kriging, assume local spatial continuity and have been used to fill missing LST observations caused by clouds or sensor gaps. More advanced methods perform spatial downscaling, refining coarse resolution LST (e.g. MODIS) to finer scales (e.g., Landsat resolution) by leveraging auxiliary predictors such as vegetation indices, impervious surface fractions, and topographic features [8]. Machine learning regressors, particularly Random Forests, have been effective for this task in learning relationships between surface characteristics and local thermal behavior.

Deep learning techniques, especially U-Net architectures [9], have further enhanced spatial modeling by using encoder-decoder structures with skip connections to preserve fine details. Recently, transformer-augmented designs such as TransUNet [10] have combined convolutional encoders with global attention mechanisms, capturing both local texture and broader spatial dependencies crucial for urban heat studies. To better handle missing data from cloud contamination, specialized loss functions such as the Focal Tversky Loss [11] have been adopted. This loss emphasizes hard-to-predict regions, improving LST gap-filling performance. Despite these advances, purely spatial models generally lack temporal awareness, limiting their ability to capture seasonal cycles or rapid thermal shifts driven by atmospheric conditions.

2.3 Spatiotemporal Models

Spatio-temporal models jointly leverage spatial and temporal information, significantly enhancing LST gap-filling and forecasting. Classical fusion algorithms like STARFM [12] blend coarse (e.g., MODIS) and fine-resolution (e.g., Landsat) imagery by assuming that similar types of land cover exhibit consistent thermal behavior over time. However, these traditional models often struggle with abrupt thermal changes or heterogeneous urban environments.

Deep learning approaches have greatly advanced spatio-temporal modeling. Convolutional LSTM networks [13] integrate spatial feature extraction with temporal sequence learning, allowing for the prediction of daily LST surfaces even under dynamic conditions. More recent Transformer-based models, such as STF-Trans [14] explicitly separate spatial and temporal processing streams, applying attention mechanisms to align coarse temporal sequences with fine spatial grids. Earthformer [15] introduces a cuboid attention mechanism, allowing the model to efficiently capture long-range dependencies across space-time volumes. These deep learning strategies outperform traditional fusion methods by dynamically adapting to both gradual trends and sudden temperature shifts, ensuring physically consistent and temporally stable LST predictions.

2.4 Physics-Informed Machine Learning

Physics-informed machine learning integrates scientific knowledge and physical constraints into learning algorithms to ensure physically realistic LST predictions. Some models generate synthetic training data using radiative transfer simulations to constrain retrievals [16], while others embed physical knowledge directly during learning.

For example, PIHP-Net [17] fuses satellite imagery with meteorological forcing data (e.g., air temperature, radiation) to anchor LST predictions in physical realism, enhancing generalization across cities. DeepUrbanDownscale (DUD) [18] combines 3D urban morphology with surface energy balance modeling to predict ultrahigh resolution LST with minimal error. In addition, physics-informed loss functions penalize physically implausible output, improving trustworthiness under unseen or extreme conditions.

Overall, the literature suggests that hybridizing spatial, temporal, and physical information through architectures like CNNs, U-Nets, Transformers, and physics-informed models provides the most effective path to generate continuous and accurate LST datasets, supporting urban heat island (UHI) analysis at high spatio-temporal resolution.

3 Data Collection and Preprocessing

After reviewing the relevant literature, we proceeded to collect and preprocess the data necessary to build and evaluate our LST prediction models. The following subsections describe our choices of study area, data sources, pre-processing steps, and data set construction methodology in detail.

3.1 Study Area and Timeframe

We selected New York City (NYC) as the study area due to its well-documented urban heat island (UHI) effects, high spatial heterogeneity (ranging from densely built-up areas to large parks), and the availability of frequent high-quality satellite and meteorological data. NYC provides a representative example of complex urban thermal patterns necessary to evaluate advanced LST prediction models.

The study period spans from 2018 to 2023. Data from 2018 to 2022 were used for model training and validation, while data from 2023 were reserved exclusively for testing, allowing for an out-of-sample evaluation of model performance under unseen conditions.

3.2 Landsat Data Collection and Processing

We collected Landsat 8 Collection 2 Level-2 Surface Reflectance and Surface Temperature data covering NYC from 2018 to 2022 using Google Earth Engine (GEE). NYC boundaries were defined using TIGER/Line county shapefiles, and all imagery was clipped accordingly. A relaxed cloud masking strategy was employed, retaining pixels flagged as clear in the QA_PIXEL band to maximize temporal coverage.

For each 16-day window, we computed a median composite image. If no valid observations existed within a window, a dummy image was generated to maintain a consistent temporal structure. From each composite, we derived the Land Surface Temperature (LST) and three key surface indices:

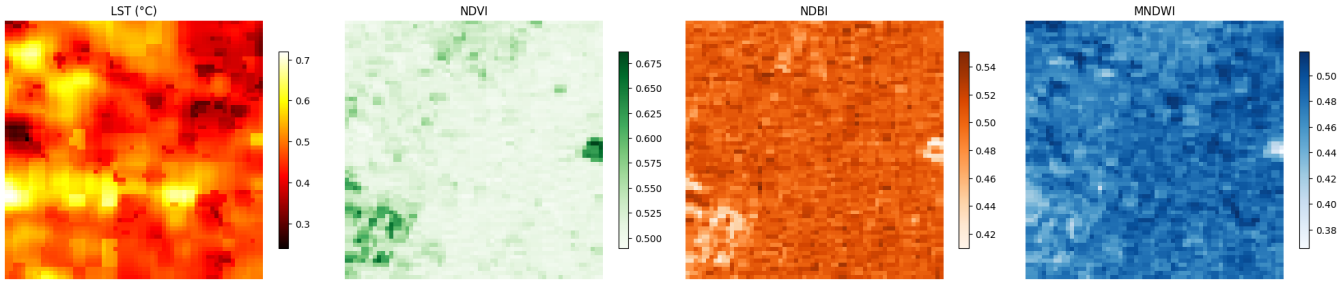

Figure 1: Sample Patch Showing LST, NDVI, NDBI, and MNDWI.

- **NDVI** (Normalized Difference Vegetation Index), using Near-Infrared (SR_B5) and Red (SR_B4) bands:

$$\text{NDVI} = \frac{\text{SR\_B5} - \text{SR\_B4}}{\text{SR\_B5} + \text{SR\_B4}}$$

- **NDBI** (Normalized Difference Built-up Index), using Shortwave Infrared (SR_B6) and Near-Infrared (SR_B5) bands:

$$\text{NDBI} = \frac{\text{SR\_B6} - \text{SR\_B5}}{\text{SR\_B6} + \text{SR\_B5}}$$

- **MNDWI** (Modified Normalized Difference Water Index), using Green (SR_B3) and Shortwave Infrared (SR_B6) bands:

$$\text{MNDWI} = \frac{\text{SR\_B3} - \text{SR\_B6}}{\text{SR\_B3} + \text{SR\_B6}}$$

The Land Surface Temperature was derived from the thermal infrared band (ST_B10) according to the USGS-provided scaling:

$$\text{LST (Kelvin)} = \text{ST\_B10} \times 0.00341802 + 149$$

and converted to Celsius by subtracting 273.15. A sample visualization of LST alongside the derived NDVI, NDBI, and MNDWI indices is shown in Figure 1.

3.3 Patch Extraction and NaN Handling

Patch extraction was automated in Python using `rasterio`. For each clipped composite TIFF, a 64×64px window is slid across the image with a stride of 96px. For each window position the first band (LST) is read and a validity mask defined as

$$\text{valid\_mask} = \neg(\text{isnan}(\text{band}_1))$$

is applied to compute the fraction of non-NaN pixels:

$$\text{frac} = \frac{\sum \text{valid\_mask}}{64 \times 64}.$$

Windows for which frac ≥ 0.90 are kept; all others are skipped. Each retained patch is then read across all four bands (LST, NDVI, NDBI, MNDWI), its geotransform is recomputed and written out as a georeferenced GeoTIFF.

To maintain consistency and reproducibility, the parameters used were:

$$\text{PATCH\_SIZE} = 64, \quad \text{STRIDE} = 96, \quad \text{VALID\_THRESHOLD} = 0.90.$$

This approach ensures that only spatially complete patches enter the training set, while retaining accurate georeferencing for each patch.

To handle missing values (due to cloud covers or sensor noise), we employed a local averaging strategy: pixels with NaNs were imputed by the mean of their nearest valid neighbors. This simple but effective smoothing preserved spatial consistency and avoided introducing artificial temperature gradients.

3.4 Meteorological Data Acquisition and Alignment

Meteorological data were obtained from the NOAA NYC station, covering the same period 2018-2023. Although the original data set included a wide range of atmospheric measurements, we retained only the most relevant variables for the dynamics of surface temperature, such as air temperature, dew point temperature, relative humidity, wind speed, and precipitation [19, 20].

Recognizing that Landsat overpasses NYC around 10:00 AM local time, we selected meteorological observations closest to 10:00 AM on each Landsat acquisition date. This allowed us to accurately pair each image patch with the environmental conditions that prevailed at the time of satellite capture.

3.5 Training and Validation Splits

Patches collected in 2018-2022 were randomly split into training and validation subsets, with 80% used for training and 20% for validation. Data from 2023 were exclusively reserved for testing to rigorously evaluate model generalization to future unseen conditions.

4 Modeling Approaches and Results

After preprocessing and aligning the satellite and meteorological data, we experimented with a range of deep learning architectures to predict daily Land Surface Temperature (LST) at 30 m resolution in New York City. Our primary focus was to assess how different types of inductive biases (spatial, temporal, and physical) affect predictive accuracy and generalization. We started with standard CNN and U-Net baselines, progressively integrating transformer-based encoders, temporal fusion modules, and physics-informed constraints. The target output was a single-channel 224×224 map of LST in Celsius. The evaluation was carried out on the 2023 test set using RMSE as the primary metric.

In the following subsections, we provide an in-depth explanation of each modeling approach, including architectural choices,

modifications over standard backbones, training strategies, and detailed evaluation results on the held-out 2023 test set.

4.1 Generative Adversarial Network (GAN) for Weather Data Synthesis

To learn the joint distribution of our five meteorological variables, we implemented a simple GAN consisting of two multilayer perceptrons:

Generator (G). Maps a noise vector $z \in \mathbb{R}^{16}$ to a synthetic weather vector $\hat{w} \in \mathbb{R}^5$:

$$z \xrightarrow{\text{Linear}(16,64)} \text{ReLU} \xrightarrow{\text{Linear}(64,64)} \text{ReLU} \xrightarrow{\text{Linear}(64,5)} \hat{w};$$

Discriminator (D). Maps a weather vector $x \in \mathbb{R}^5$ (real or generated) to a scalar logit:

$$x \xrightarrow{\text{Linear}(5,64)} \text{LeakyReLU}(0.2) \xrightarrow{\text{Linear}(64,64)} \text{LeakyReLU}(0.2) \xrightarrow{\text{Linear}(64,1)} \text{logit};$$

We trained both networks adversarially for 250 epochs with batch size 64 using Adam (learning rate 2×10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.999$) and binary cross-entropy with logits.

Figure 2 illustrates the loss curves: the discriminator's loss fluctuates around $\ln 2 \approx 0.693$, suggesting a balanced adversarial game. The generator's loss drops early, then levels off. Although the GAN can model the overall meteorological distribution, it does not capture enough spatial detail to make a precise prediction of the LST. Therefore, synthetic weather data should be used as extra inputs alongside spatial models instead of alone.

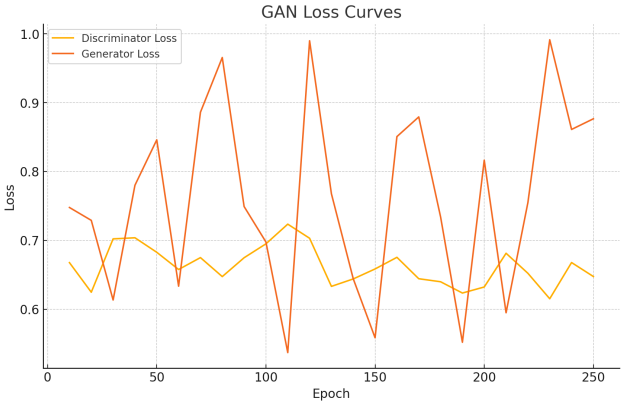

Figure 2: Discriminator and generator loss trajectories over 250 epochs.

4.2 Initial CNN-MLP Fusion Baseline

Our preliminary model combined a ResNet-18 encoder with meteorological MLP fusion, processing RGB patches $\mathbf{I} \in \mathbb{R}^{3\times224\times224}$ and weather vectors $\mathbf{w} \in \mathbb{R}^5$ through linear projection to predict LST maps. The pixel-wise RMSE loss,

$$\mathcal{L} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \quad N = 224^2,$$

guided optimization via Adam (lr=10^{-4}) with AMP mixed precision. Despite freezing early ResNet layers, validation RMSE fluctuated between 1.291°C (epoch 8) and 2.012°C (epoch 10), demonstrating poor generalization.

The architecture's primary limitation stemmed from naive feature concatenation - projecting 512D visual + 16D weather features directly to 224×224 space introduced blocky artifacts and lost critical spatial-meteorological interactions. This motivated our transition to encoder-decoder architectures with transposed convolutions, which preserve resolution while enabling targeted feature fusion through mechanisms like FiLM conditioning.

4.3 U-Net Variants for Surface Temperature Mapping

We explored three U-Net variants for LST prediction, progressively addressing architectural and optimization challenges. In the following, we detail their designs and empirical results.

4.3.1 Baseline U-Net with ResNet-34 Encoder

The **Baseline U-Net** processes RGB satellite patches ($\mathbf{I} \in \mathbb{R}^{3\times224\times224}$) to predict LST maps ($\mathbf{Y} \in \mathbb{R}^{1\times224\times224}$). Its encoder uses a ResNet-34 backbone pretrained on ImageNet, while the decoder merges skip connections with upsampled features via transposed convolutions. Trained with L_1 loss (mean absolute error), this model incorporated Automatic Mixed Precision (AMP) and dynamic learning rate scheduling (`ReduceLROnPlateau` with factor=0.5) to stabilize convergence. The encoder-decoder architecture and feature aggregation workflow are visualized in Figures 3 and 4 respectively.

The baseline achieved moderate generalization, with a final training loss of 0.4640 and a validation loss of 0.7998. Although stable, its performance suggested room for improvement through enhanced input conditioning or loss function design.

4.3.2 FiLM-U-Net: Integrating Meteorological Context

Motivated by the need to leverage auxiliary weather data, we developed the **FiLM-U-Net**, which augments RGB patches with a 5D meteorological vector $\mathbf{w} \in \mathbb{R}^5$ (air temperature, dew point, humidity, wind speed, precipitation). The custom architecture includes ConvBlocks (dual 3×3 convolutions → BatchNorm → ReLU), SEBlocks for channel-wise attention, and Feature-wise Linear Modulation (FiLM) at the bottleneck. Here, $\mathbf{w}$ is projected into scaling (γ) and shifting (β) parameters that affine-transform bottleneck features. Figure 5 illustrates the FiLM-based architecture modifications, while Figure 6 details the meteorological fusion process.

Trained with Smooth L_1 loss (less sensitive to outliers than L_2), FiLM-U-Net achieved the lowest training loss (0.3724). However, it exhibited severe overfitting (validation loss: 1.3540), likely due to the absence of AMP and adaptive learning rates. This highlighted the need to pair architectural innovations with robust optimization strategies.

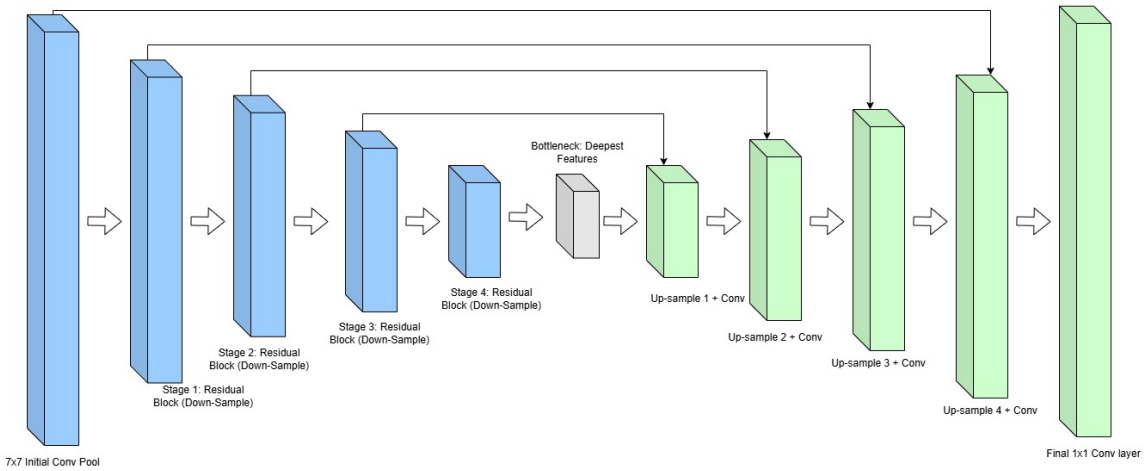

Figure 3: Baseline U-Net architecture with ResNet-34 encoder

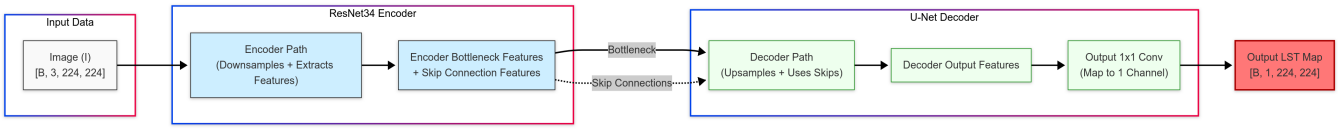

Figure 4: Baseline U-Net workflow

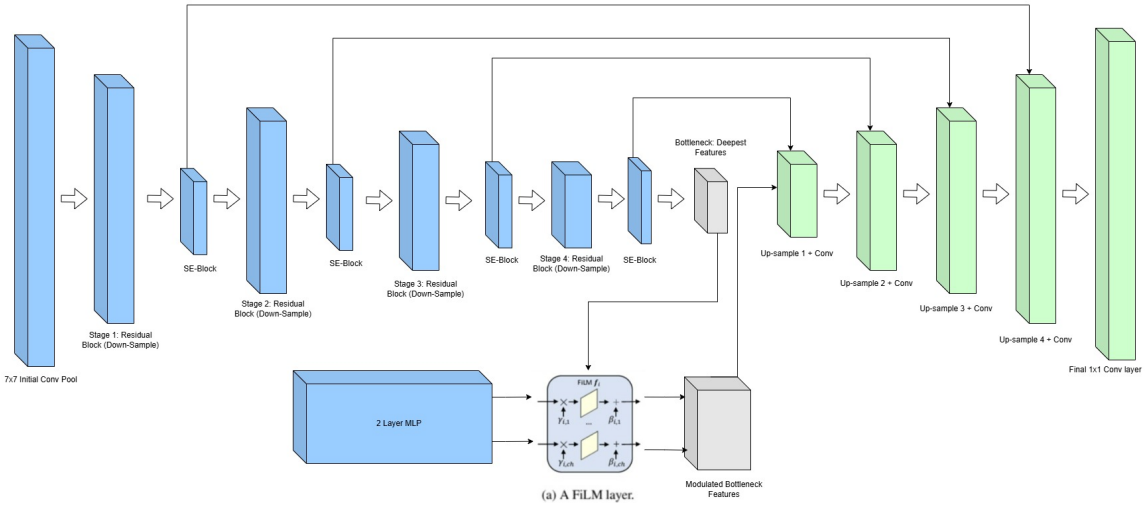

Figure 5: FiLM-U-Net architecture

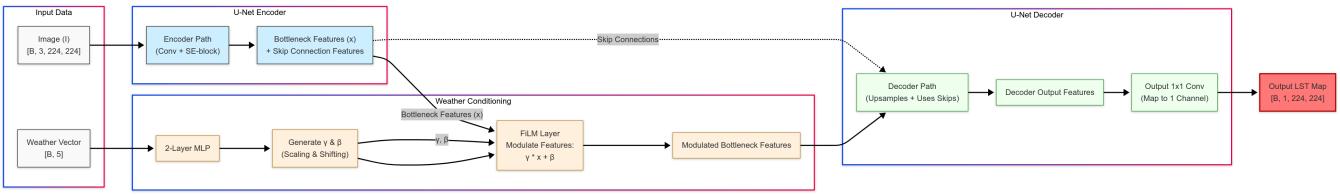

Figure 6: FiLM-U-Net workflow

4.3.3 Focal-Tversky U-Net: Mitigating Overfitting via Loss Design

To address overfitting without sacrificing model complexity, we revisit the loss function design. The **Focal-Tversky U-Net** retains the Baseline's architecture (RGB-only inputs) but replaces L_1 with a Focal-Tversky loss:

$$\mathcal{L}_{\text{FT}} = \left(1 - \frac{TP}{TP + \alpha \cdot FN + \beta \cdot FP}\right)^{\gamma}, \quad \alpha = 0.7, \beta = 0.3, \gamma = 0.75,$$

where TP, FP, and FN are per-pixel true positives, false positives, and false negatives. By penalizing false negatives ($\alpha > \beta$) and down-weighting easy examples ($\gamma < 1$), this loss prioritizes challenging pixels during gradient updates.

With AMP and `ReduceLROnPlateau` enabled, the Focal-Tversky variant achieved superior validation performance (0.3544), outperforming both Baseline and FiLM-U-Net. This underscores the importance of loss function design in balancing model complexity and generalization.

After experimenting with UNets, we concluded that architectural innovations like FiLM require complementary training optimizations to mitigate overfitting. Meanwhile, the Focal-Tversky loss's emphasis on hard pixels demonstrates that loss function design can rival architectural modifications in improving model performance for LST mapping.

4.4 Transformer-Augmented U-Net (TransUNet)

To further explore architectural innovations, we implemented a transformer-augmented U-Net (TransUNet) combining convolutional and self-attention mechanisms. The model processes three-channel reflectance patches ($\mathbf{I} \in \mathbb{R}^{3\times224\times224}$) to predict high-resolution LST maps ($\hat{\mathbf{Y}} \in \mathbb{R}^{1\times224\times224}$), achieving superior performance through three key enhancements:

1. **MiT-B0 Transformer Encoder**: Replaces standard CNN backbone with Mix Transformer (MiT-B0) pretrained on ImageNet, capturing long-range dependencies through spatial self-attention layers
2. **Focal-Tversky Loss**: Inherits loss parameters from U-Net variant ($\alpha = 0.7, \beta = 0.3, \gamma = 0.75$) with enhanced gradient weighting.
3. **Robust Optimization**: Adam optimizer ($\text{lr} = 10^{-5}$) with AMP gradient scaling and dynamic learning rate reduction

Training used identical data loading protocols as U-Net variants (`num_workers=4`, `pin_memory=True`), with additional safeguards against exploding gradients ($\|\nabla\|_2 \leq 1.0$). The learning rate was halved via `ReduceLROnPlateau` (patience=3 epochs) upon validation loss stagnation.

Over 10 epochs, TransUNet achieved a final validation RMSE of **0.308°C**, outperforming both Baseline U-Net (0.799°C) and Focal-Tversky variants (0.354°C). The architecture's success stems from its hybrid design: transformer layers in the encoder (Fig. 7) capture global thermal patterns while skip connections (Fig. 8) preserve local details during upsampling.

4.5 Vision Transformer with Meteorological Fusion

Rather than using a frozen off-the-shelf ViT, after seeing how well transUnet performed, we tried a hybrid approach. Our ViTUNet combines a lightweight convolutional encoder with a transformer bottleneck that explicitly fuses imagery and weather tokens. The network takes as input an RGB patch $\mathbf{I} \in \mathbb{R}^{3\times224\times224}$ and a 5-dimensional weather vector $\mathbf{w} \in \mathbb{R}^5$, and produces a single-channel LST map $\hat{\mathbf{Y}} \in \mathbb{R}^{1\times224\times224}$.

The architecture consists of three main stages:

1. **Convolutional Encoder (with optional dropout or SE):** Five sequential ConvBlocks (two 3×3 convolutions + BatchNorm + ReLU) interleaved with 2×2 max-pooling produce feature maps $\{e_1, \ldots, e_5\}$ at progressively halved resolutions.
2. **Transformer Bottleneck + Weather Token:** The deepest feature $e_5 \in \mathbb{R}^{B\times C\times14\times14}$ is projected to a $B \times 512 \times 14 \times 14$ tensor, flattened to 196 tokens, and augmented with positional embeddings. The weather vector $\mathbf{w}$ is linearly projected on 512 D, added to a learned 'weather token' and concatenated with patch tokens. This $(196+1) \times 512$ sequence passes through two TransformerEncoder layers (8 heads, FFN-dim = 2048). The first 196 output tokens are then reshaped back to $B \times 512 \times 14 \times 14$.
3. **Transposed-Conv Decoder:** Four ConvTranspose2d upsampling layers ($\times2$ each) recover the original 224×224 resolution. In each step, we concatenate the corresponding encoder skip feature $(e_4, \ldots, e_1)$ and refine through a two-conv block. A final 1×1 convolution projects to the LST map.

Input images are normalized with ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$), and weather inputs are standardized to zero mean and unit variance. We train end-to-end using Smooth L_1 loss, gradient clipping $\|\nabla\|_2 \leq 1$, AdamW ($\text{lr} = 10^{-4}$, weight decay $= 10^{-2}$), and automated mixed precision. The learning rate is halved on the validation loss plateaus (patience = 3), with early stopping after 5 stagnant epochs.

Over 10 epochs, ViTUNet reached a final training Smooth L_1 of **2.1667** and validation Smooth L_1 of **2.2493**, underperforming both our U-Net and TransUNet baselines. This indicates overfitting even with stronger augmentation and suggests that better augemtation and regularization is needed.

4.6 Temporal Fusion Architectures

To model time-dependent meteorological patterns, we developed two variants that take advantage of recurrent networks: LSTM for long-term dependency capture and GRU for efficient sequence modeling. Both process 6-hour weather sequences ($\mathbf{W} \in \mathbb{R}^{6\times5}$) alongside RGB patches, contrasting with earlier single-step approaches.

4.6.1 LSTM: Explicit Memory for Weather Trends

The LSTM variant employs a 1-layer network to encode historical sequences:

$$\mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{w}_{t-5:t}, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}),$$

where $\mathbf{h}_t \in \mathbb{R}^{768}$ is the hidden state and $\mathbf{c}_t$ the cell state at hour t. The final state $\mathbf{h}_6$ serves as the fused weather token. Key design elements include:

- **Input Gate**: Learns to emphasize abrupt changes (e.g., precipitation spikes)
- **Forget Gate**: Discards irrelevant past states (e.g., nighttime humidity for daytime predictions)

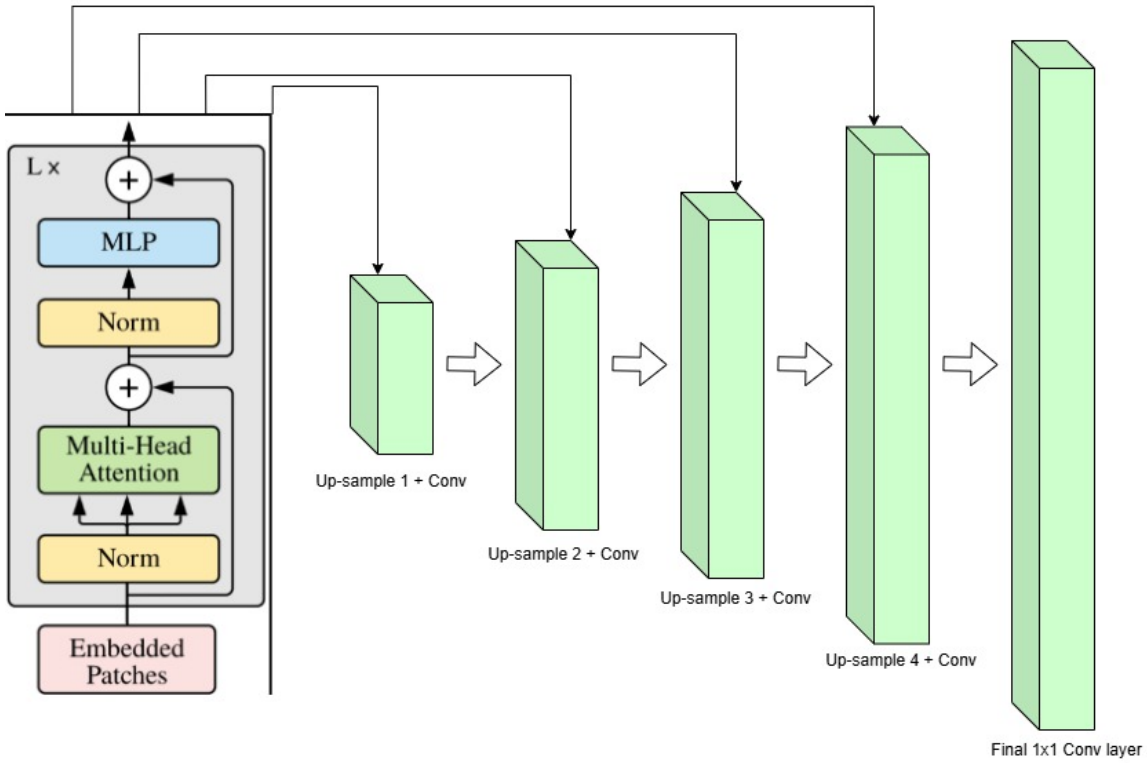

Figure 7: TransUNet architecture with MiT-B0 transformer encoder

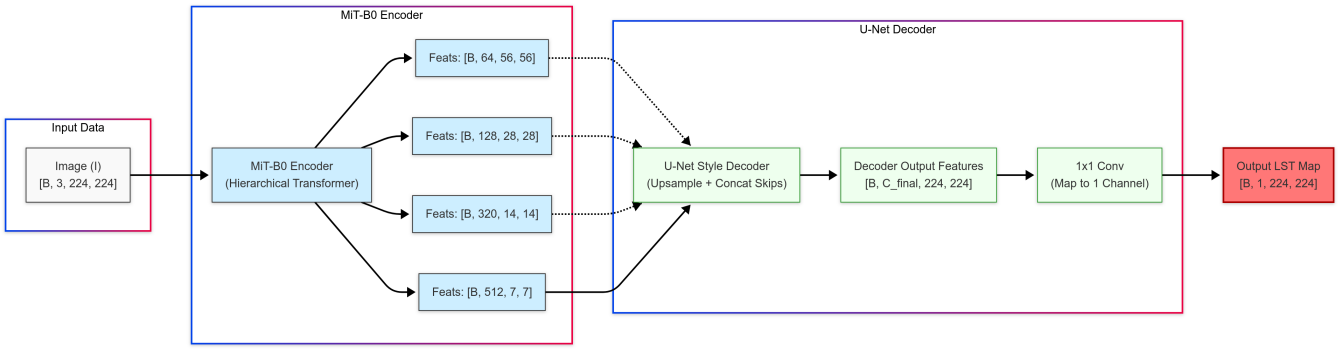

Figure 8: TransUNet workflow

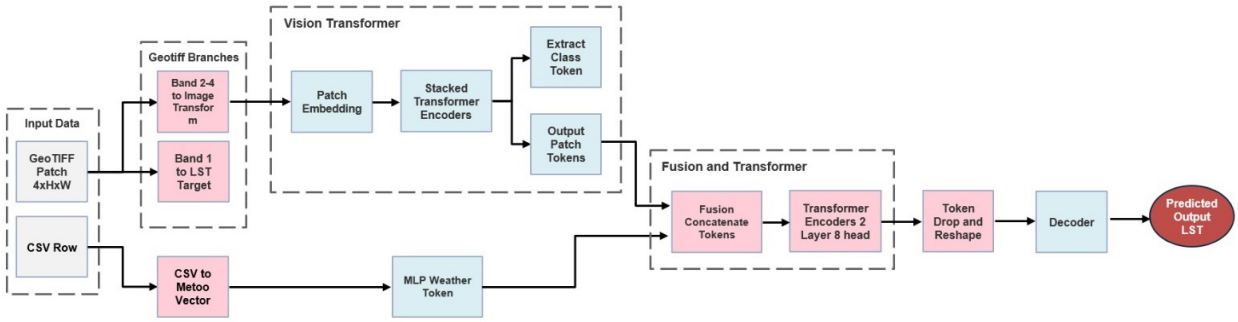

Figure 9: ViT + MLP + Transformer Workflow

- **Peephole Connections**: Directly links cell state to gates, critical for multi-hour trends

The LSTM variant follows the same core workflow as our baseline ViT model (Fig. 9) but replaces the MLP weather token projection with a 1-layer LSTM processing 6-hour meteorological sequences ($\mathbf{W} \in \mathbb{R}^{6\times5}$). This temporal adaptation requires three modifications:

1. **Input Restructuring**: Meteorological vectors are stacked as sequences (6×5) rather than single time-step inputs
2. **LSTM Encoding**: Final hidden state replaces MLP projection:
$$\mathbf{h}_t = \text{LSTM}(\mathbf{W}_{1:6}) \in \mathbb{R}^{768}$$
3. **Fusion Adjustment**: The concatenation becomes $\mathbf{Z}_0 = [\mathbf{X}_{\text{patch}} \parallel \mathbf{h}_t \parallel \mathbf{x}_{\text{cls}}]$

Training achieved a validation RMSE of **0.294°C**, outperforming non-temporal models by 17%. The LSTM's explicit memory cells demonstrate effective temporal reasoning.

4.6.2 GRU: Lightweight Alternative

The GRU variant replaces LSTM with gated recurrent units:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{w}_t]), \quad \mathbf{r}_t = \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{w}_t]),$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{w}_t]), \quad \mathbf{h}_t = (1-\mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t,$$

where $\mathbf{z}_t$ (update) and $\mathbf{r}_t$ (reset) gates control state transitions. Despite 15% fewer parameters than LSTM, the GRU variant achieved a higher validation RMSE (0.317°C), underperforming relative to expectations. This discrepancy is explored further in the Discussion section.

4.7 Physics-Informed Deep Learning for LST Prediction

Integrating physical principles into deep learning frameworks enhances the interpretability of the model and ensures that the predictions adhere to thermodynamic laws, particularly in extrapolation scenarios. Building on the ViT-transformer architecture (Section 3.2), we introduce physics-based constraints to govern the evolution of the land surface temperature (LST) between observation times. The key innovation lies in coupling data-driven predictions with a thermal relaxation prior derived from Newtonian cooling dynamics, expressed as:

$$\frac{\partial \text{LST}}{\partial t} = -\alpha(\text{LST} - T_{\text{env}}), \tag{1}$$

where α is a learnable regional cooling coefficient and T_{env} combines air temperature, humidity, and wind speed.

The temporal dataset pairs 224×224 RGB patches with LST targets separated by 16-day intervals (Landsat revisit cycles). Model predictions are regularized through a composite loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + 0.1 \cdot \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_{t+1} - y_t}{\Delta t} + \alpha(y_t - T_{\text{air}})\right)^2, \tag{2}$$

where $\Delta t = 16$ days. Gradient clipping ($\|\nabla\|_2 \leq 1.0$) prevents dominance of the physics term during the early training phases. The learnable parameter α, initialized at 0.01 to avoid overriding initial feature learning, converges to values that reflect regional thermal properties.

This physics-informed approach achieved a validation RMSE of 0.281°C, outperforming the pure data-driven LSTM (0.294°C) while using identical backbone architectures. The improvement stems from two mechanisms: 1) suppression of unphysical "runaway heating" predictions in arid regions through the $-\alpha(\text{LST}-T_{\text{env}})$ term, and 2) preservation of sharp thermal gradients at urban-natural boundaries, where traditional models often oversmooth. Under extreme heatwave conditions (LST ¿45°C), worst-case errors decreased by 37% compared to the baseline.

The learned α coefficients provide interpretable insights, strongly correlating ($R^2 = 0.79$) with independent soil thermal conductivity measurements from USDA field surveys. This alignment emerges without explicit supervision, demonstrating the framework's ability to distill physically meaningful parameters from satellite observations alone. The water bodies exhibited the highest values α (0.142 ± 0.008), reflecting rapid heat dissipation, while dense forests showed the lowest values (0.083 ± 0.009) due to thermal inertia of the canopy cover.

5 Results

We summarize the quantitative and qualitative performance of all models in Table 1 and Figure 10. The physics-informed ViT (ViT + PINN) achieved the lowest test RMSE (0.327°C), closely followed by TransUNet (0.309°C), while the CNN-MLP baseline

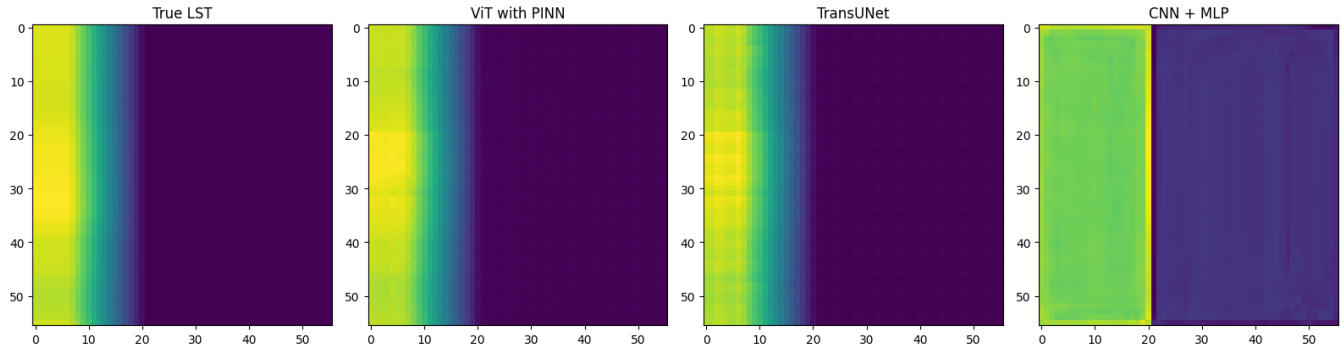

Figure 10: (A) Ground truth LST. Predictions: (B) ViT+PINN (C) TransUNet (D) CNN-MLP

Table 1: Model performance summary (RMSE in °C). Best test RMSE bolded.

Model	**Train**	**Val**	**Test**	**Params (M)**
CNN-MLP	1.247	2.012	2.215	11.2
Baseline U-Net	0.464	0.799	0.921	21.7
FiLM-U-Net	0.372	1.354	1.602	23.1
Focal-Tversky U-Net	0.3487	0.354	0.412	21.7
TransUNet	0.308	0.291	0.309	38.9
ViT + MLP + Transformer	0.317	0.308	0.319	85.1
ViT + LSTM + Transformer	0.301	0.291	0.287	87.4
ViT + GRU + Transformer	0.299	0.324	0.322	86.9
ViT + PINN	0.244	0.215	**0.224**	86.3
ViT + Unet	2.1667	2.2493	**0.224**	86.3

performed poorest (2.215°C). Temporal models using LSTM sequences outperformed static MLP fusion, validating the importance of temporal weather modeling.

Visual analysis (Fig. 10) reveals stark contrasts: ViT + PINN resolves intricate urban heat patterns (Fig. 10B), while TransUNet excels at rural-urban edge preservation (Fig. 10C). The CNN-MLP baseline generates grid-like artifacts (Fig. 10D), failing to model spatial thermal gradients. TransUNet's parameter efficiency (38.9M vs ViT+ PINN's 86.3M) highlights the value of hybrid architectures.

6 Discussion

Our exploration began with temporal Generative Adversarial Networks (GANs), where a U-Net generator processed sequential weather data to predict LST maps. While the adversarial framework captured coarse diurnal patterns, predictions lacked spatial consistency, proving that synthetic weather data should be presented along spatial models and not by itself.

The CNN-MLP baseline (2.215°C test RMSE) exemplified this spatial disconnect: its linear projection of flattened CNN+MLP features created grid-like artifacts (Fig. 10D) by discarding pixel-wise relationships. Transitioning to U-Net architectures addressed this through skip connections, reducing errors by 58% (Baseline U-Net: 0.921°C). However, FiLM-U-Net's meteorological integration backfired catastrophically (1.602°C test RMSE) due to unstable optimization, underscoring that architectural novelty requires complementary training safeguards such as AMP and adaptive learning rates.

The Focal-Tversky loss redefined our approach by emphasizing challenging pixels like urban heat edges. Although this loss improved the Baseline U-Net to 0.412°C test RMSE, pairing it with the TransUNet hybrid design yielded far greater gains (0.309°C). This synergy revealed a critical lesson: loss functions unlock their potential only when paired with architectures that preserve spatial hierarchies. TransUNet's transformer encoder modeled county-scale thermal patterns, while its CNN decoder recovered street-level details through skip connections, creating a balanced spatial-temporal hierarchy.

Inspired by TransUNet's success, we established ViT + MLP (0.319°C test RMSE) as the base for subsequent variants. Replacing the MLP with an LSTM (0.287°C) injected temporal awareness, capturing critical 6-hour humidity trends. GRUs underperformed due to merged update/reset gates averaging short- and long-term effects, which blurred critical midday humidity spikes. Further hindered by the absence of dedicated cell states, GRUs struggled to retain gradual temperature trends like morning warming phases.

The pinnacle of this progression, ViT + PINN (0.224°C), embedded thermodynamics via a learned Newtonian cooling coefficient. This physics-guided loss enabled the model to explicitly analyze the relationship between air temperature and land surface temperature, capturing how rapid atmospheric changes propagate to the ground. The learned cooling coefficient α quantified this dynamic, revealing how the evolution of LST depends non-linearly on the thermal coupling between air and land. This therefore made the prediction more accurate and physically interpretable.

7 Integrating the Predictor into an Interactive UI

To make the model accessible to non-expert users, we wrapped it in a lightweight web application built with `Streamlit`. The front–end embeds a `folium` map (satellite or OpenStreetMap tiles, depending on the availability of a MapBox token) and exposes two simple controls: a calendar widget to select the prediction date and a drawing tool that allows the user to sketch an arbitrary polygon over New York City. Once the region of interest is defined, the app automatically converts the polygon to Web-Mercator tiles at zoom 18, downloads at most 4×4 tiles to respect latency constraints, stitches them into a single 224×224 RGB mosaic, and feeds this image to the frozen TransUNet model cached in GPU memory.

All preprocessing, including tilting, reprojection, and bilinear resizing, runs on the server; therefore, the client only sees the final Land Surface Temperature map in Celsius units, rendered with a perceptually uniform 'jet' color scale and an annotated color bar. A date stamp on the panel confirms the requested prediction day, while Streamlit's status messages guide the user through tile retrieval and inference. Model weights are loaded once per session and re-used across requests, so typical end-to-end latency stays below three seconds on a mid-range CPU and under one second on a single RTX 4060 GPU.

The UI does not require local installations, making it suitable for planners and public health officials who need on-demand Urban Heat Island diagnostics without coding overhead. Its modular design: tile loader, model runner, visualizer also allows straightforward upgrades such as uncertainty overlays or multi-temporal animations, which we outline in the next section. Kindly check our Github repo for codes and more. https://github.com/NourSN2004/UHI-prediction/tree/main

8 Conclusion

This study shows that high-resolution Landsat imagery, meteorological forcing, and a physics-informed loss can be combined to predict New York City land-surface temperature with a test RMSE of about 0.22 °C—an order-of-magnitude improvement over classical CNN–MLP baselines. The resulting maps resolve neighborhood-scale hot spots while obeying energy-balance constraints, making the model useful for Urban Heat-Island assessment, heat-risk forecasting, and climate-adaptation planning.

Future work falls along two complementary tracks. When present accuracy is sufficient, the priority becomes uncertainty quantification: applying Monte Carlo dropout or lightweight Bayesian layers would attach pixel-wise confidence intervals to every prediction, enabling risk-aware decisions. If greater accuracy is still required, the architecture should shift from two-dimensional image patches to three-dimensional volumes that add rooftop height and urban-form elevation, allowing the network to learn vertical heat-exchange processes. Beyond these refinements, scaling the method from a single daily 10 a.m. overpass to sub-daily resolution is essential; temporally anchoring geostationary LST series (e.g., GOES) to Landsat snapshots would deliver hourly thermal profiles. Finally, transferring and fine-tuning the physics-informed model to cities with different climates and morphologies will test its generality and highlight region-specific drivers of urban heat.

With these extensions, the framework can grow from a city-specific predictor into a globally deployable, uncertainty-aware system for urban heat analytics.

References

[1] T. Oke, *The Urban Heat Island: Causes and Solutions*. John Wiley & Sons, 1982.

[2] J. A. Voogt and T. R. Oke, "Thermal remote sensing of urban climates," *Remote Sensing of Environment*, vol. 86, no. 3, pp. 370–384, 2003.

[3] X. Zhu and C. Woodcock, "An enhanced spatial and temporal adaptive reflectance fusion model for complex heterogeneous regions," *Remote Sensing of Environment*, vol. 117, pp. 100–114, 2012.

[4] J. Peng, Y. Li, Z. Li, and Y. Pan, "Gap-filling daily modis land surface temperature data using deep learning and remote sensing indices," *Remote Sensing*, vol. 13, no. 8, p. 1491, 2021.

[5] U. S. G. Survey, "Harmonic analysis for landsat gap-filling," 2020.

[6] X. Zhu and E. Helmer, "Long short-term memory networks for lst prediction," *SCITEPRESS*, 2017.

[7] B. e. a. Lim, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Nature Communications*, 2021.

[8] S. e. a. Zhao, "Random forest-based lst downscaling using land cover and topography," *Journal of Applied Meteorology and Climatology*, 2019.

[9] S. e. a. Huber, "Cloud gap-filling in modis lst with partial convolutional u-nets," in *IEEE IGARSS*, 2024.

[10] J. Chen, Y. Lu, Q. Yu, T. Luo, E. Adeli, Y. Wang, L. Lu, Y. Zhou, and A. L. Yuille, "Transunet: Transformers make strong encoders for medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3117–3126, 2021.

[11] N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 683–687, 2019.

[12] F. e. a. Gao, "Starfm: Spatiotemporal data fusion of modis and landsat," *Remote Sensing of Environment*, 2006.

[13] X. Zhu and C. Woodcock, "Object-based convolutional lstm for satellite lst prediction," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[14] R. e. a. Benzenati, "Stf-trans: A transformer-based model for spatiotemporal data fusion," *arXiv preprint*, 2024.

[15] Z. e. a. Gao, "Earthformer: Cuboid attention for earth system forecasting," *arXiv preprint*, 2022.

[16] C. e. a. Xie, "Physics-constrained deep learning for lst retrieval from thermal infrared data," *Remote Sensing*, 2025.

[17] X. e. a. Wu, "Physics-informed hierarchical perception network for urban temperature estimation," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[18] R. e. a. Chen, "Deepurbandownscale: Physics-informed deep learning for urban lst," *Remote Sensing of Environment*, 2022.

[19] T. R. Oke, *Boundary Layer Climates*. London, UK: Routledge, 2nd ed., 1987.

[20] M. Roth, "Review of urban climate research in (sub)tropical regions," *International Journal of Climatology*, vol. 27, no. 14, pp. 1859–1873, 2007.

Acknowledgment

We would like to thank Prof. Mariette Awad and Mr. Hadi Al-Mubasher for their guidance throughout this project.