ISSN 0027-1349, Moscow University Physics Bulletin, 2024, Vol. 79, Suppl. 2, pp. S784–S797. © Allerton Press, Inc., 2024.

Approximation of Spatial and Temporal Variability of the Urban Heat Island in Moscow Using Machine Learning

M. I. Varentsov[1,2,3*], M. A. Krinitskiy[1,4,5], and V. M. Stepanenko[1,2]

[1] *Research Computing Center, Lomonosov Moscow State University, Moscow, Russia*
[2] *Faculty of Geography, Lomonosov Moscow State University, Moscow, Russia*
[3] *Obukhov Institute of Atmospheric Physics, Russian Academy of Sciences, Moscow, Russia*
[4] *Shirshov Institute of Oceanology, Russian Academy of Sciences, Moscow, Russia*
[5] *Moscow Institute of Physics and Technology, (National Research University), Dolgoprudny, Moscow oblast, Russia*

Received October 1, 2024; revised October 10, 2024; accepted October 20, 2024

Abstract—This study is devoted to the application of machine learning (ML) methods for statistical approximation of the urban-induced temperature anomaly, known as the urban heat island (UHI), and its spatiotemporal dynamics, using the example of the Moscow megacity. This task is considered as part of a more general problem of statistical downscaling of meteorological fields for urban conditions. Therefore, we aim to approximate a high-resolution field of urban temperature anomalies based on predictors characterizing low-resolution meteorological data and high-resolution surface properties. As the input data for training ML models, we use the results of high-resolution hydrodynamic simulations of the meteorological regime in the Moscow region conducted with the COSMO regional atmospheric model coupled with the TERRA_URB urban canopy parameterization. For the ML model, we use the gradient boosting method implemented by the CatBoost algorithm with GPU support. To account for nonlocal dependences between UHI and surface properties, we use an original quasi-local approach to define the feature vectors. This approach consists of using data localized at individual points (nodes of the computational mesh of the COSMO model) as feature descriptions and generating additional features based on the predictors' values for neighboring points using different types of convolution filters. As such filters, we use a moving average with a circular kernel of different radii and more advanced self-adjusting directional filters formed by taking into account large-scale data on wind speed and direction. We show that such nonlocal features are important for correctly reproducing the key patterns of the UHI spatial structure, in particular the smoother structure of seasonally-averaged temperature anomalies in comparison to surface properties, and the shift of temperature anomalies to the leeward side of the city for specific cases with different wind directions.

Keywords: urban climate, downscaling, urban boundary layer, convolution filter, gradient boosting, CatBoost

DOI: 10.3103/S0027134924702254

1. INTRODUCTION

Modern weather and climate modeling technologies rely on hydrodynamical models (HMs) that numerically solve the equations of atmospheric hydrothermodynamics. At the same time, machine learning (ML) methods are increasingly applied in this area. ML methods are used for refining and correcting HM-based forecasts at the post-processing stage [1]; for accelerating the solution of equations and replacing specific physical parameterizations in HMs [2, 3]; and even as alternatives to HMs [4, 5]. Another promising area for ML application is the problem of downscaling, i.e. increasing the resolution of weather and climate data, where ML methods serve as an alternative to the computationally expensive method of dynamic downscaling using regional HMs [6].

The task of downscaling is particularly relevant for urban areas due to their spatial heterogeneity and local meteorological anomalies, such as the

*E-mail: mikhail.varentsov@srcc.msu.ru

urban heat island [7]. Such anomalies significantly impact the population and economy, with some of the most notable effects being negative impacts on human thermal comfort and health during heat waves [8]. Global HMs, with a grid spacing of about 100 km for climate simulations and about 10 km for weather forecast models, typically do not account for urban meteorological effects. Therefore, regional mesoscale HMs, with grid spacings ranging from a few kilometers to a few hundred meters and supplemented with special urban parameterizations [9], are often used to downscale coarser meteorological data for cities within operational weather forecasting systems [10] or in regional weather and climate studies, including those focused on urban planning issues. However, all such models require significant computing resources, which imposes restrictions on their use.

Significant demands for computational resources are the main motivation for investigating the use of ML methods for high-resolution simulations of meteorological parameters in urban areas. Several studies have already demonstrated the potential of ML methods for high-resolution spatial modeling of temperature in cities, e.g., [11–13]. However, only a few studies, e.g., [14], address the full problem of downscaling low-resolution meteorological data for cities, while others approximate the spatial variability of meteorological quantities at fixed moments in time, rather than capturing their spatiotemporal dynamics.

Our study is devoted to the development of a ML-based statistical downscaling technique for low-resolution meteorological data for the Moscow megacity. As a first step, we focus on air temperature, specifically on its urban-induced anomaly, i.e., the UHI, and consider the problem of statistically approximating its spatiotemporal variability based on low-resolution predictors that characterize large-scale atmospheric processes and high-resolution predictors that describe surface properties. The purpose of this article is to present the formulation of the problem, the original architecture of the developed ML-based statistical model, and the initial results of its testing and evaluation.

2. DATA AND METHODS

2.1. Meteorological Data

The known problem in urban meteorology is a lack of representative observations. Weather stations operated by national hydrometeorological services are typically located outside built-up areas, e.g., at airports. To capture urban climate phenomena through observations, scientists deploy specialized urban meteorological networks [15, 16] or develop algorithms to collect and process data from nontraditional and opportunistic sensing technologies, such as citizen weather stations [17–19]. However, even the densest urban observational networks have limited and inhomogeneous spatial coverage, and their observations may contain gaps and be biased due to various uncertainties or hyperlocal microclimatic effects. Therefore, we begin our work on the ML-based statistical downscaling model using high-resolution meteorological data simulated by a regional mesoscale HM. Unlike observations, the results of hydrodynamic modeling are available on a regular grid with continuous coverage, which is especially convenient for developing and training ML models.

We used a dataset derived from long-term high-resolution simulations with the COSMO regional meteorological model for the Moscow region over ten summer seasons (2007–2016), described in detail in [20]. COSMO is a nonhydrostatic regional hydrodynamic model developed by the COSMO consortium and is widely used for regional climate studies, as well as for numerical weather forecasting in several countries, including Russia [10]. The model was applied for a three-step dynamic downscaling of ERA-Interim reanalysis data [21] (with a grid size of 0.75° longitude and latitude) to a finer domain covering the Moscow region with 1-km grid spacing and a size of 180×180 grid cells. The urban canopy parameterization known as TERRA_URB [22], coupled with COSMO, was used to account for features of the urban surface and simulate urban climate phenomena, including the UHI. TERRA_URB is a simple but computationally effective bulk parameterization that provides semiempirical corrections to surface parameters (roughness length, albedo, emissivity, heat capacity, etc.) used by the TERRA scheme, COSMO's land surface model, to account for the effects of the urban canopy on heat, moisture, and momentum surface-atmosphere fluxes. TERRA_URB was provided with a realistic description of Moscow city in terms of external parameters, including urban area fraction, anthropogenic heat flux, and geometric parameters of the urban environment, derived from OpenStreetMap data [20]. A distinctive feature of this meteorological dataset is the presence of paired numerical experiments, URB and noURB, in which the TERRA_URB scheme was activated in the first and deactivated in the second; in the latter, the city was replaced with natural land cover types (forest or cropland). This allows for the evaluation of urban-induced atmospheric effects based on differences between URB and noURB simulations. This approach has already been used to study the horizontal and vertical structure of Moscow's UHI, urban effects on wind speed [20], and on convective precipitation [23]. Figures 1a and 1b demonstrates how the simulation data appears for 2-m air temperature in the URB

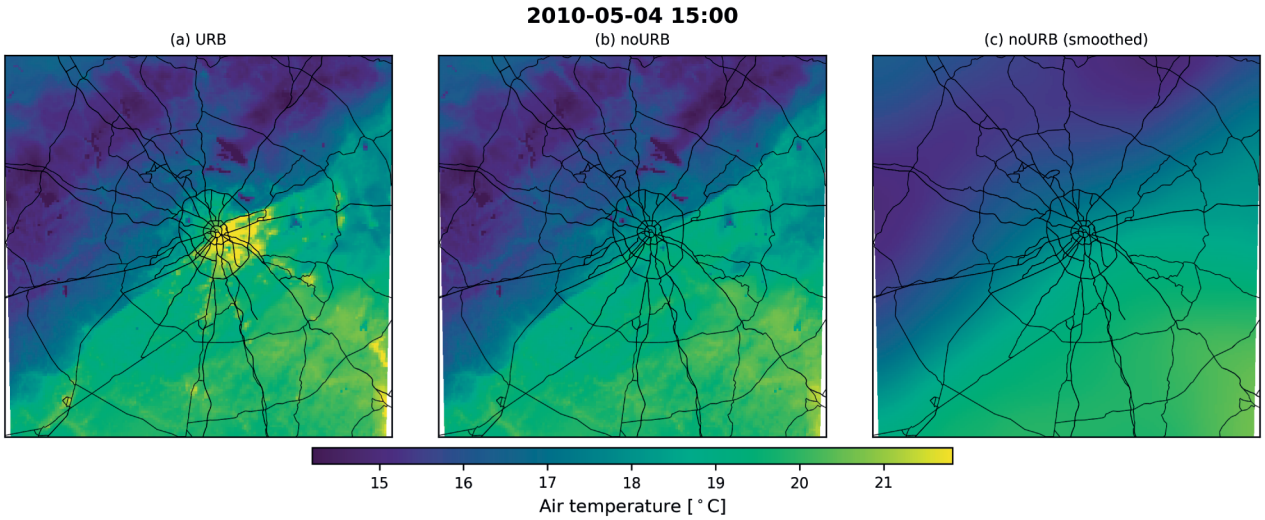

Fig. 1. Spatial patterns of the air temperature in the Moscow region according to results of URB (a) and noURB (b) numerical experiments with COSMO model, and according to noURB results smoothed with 50-km kernel (c) for a specific time moment. Black lines indicate the primary road network according to OpenStreetMap data.

and noURB numerical experiments. Model output is available at hourly intervals for both numerical experiments.

2.2. Statement of the Machine Learning Problem

We aim to approximate the temperature anomaly that occurs in the mesoscale HM due to urban-induced effects at the ith time moment and jth point in space, based on low-resolution predictors characterizing large-scale atmospheric processes (hereafter referred to as meteorological or weather-related predictors, denoted as W) and high-resolution predictors characterizing surface properties (hereafter referred to as surface predictors, denoted as S). Meteorological predictors $W_{1\ldots n_w,i,j}$ are defined for each point in time and space, yet their variations are expected to be smooth in space. Surface predictors $S_{1\ldots n_s,j}$ are assumed to be invariant over time. Available simulation data allows us to define urban temperature anomaly as the difference between the 2-m air temperature simulated in numerical experiments with (URB) and without (noURB) urban effects:

$$\Delta T_{i,j} = T_{URB,i,j} - T_{noURB,i,j}.$$

In other words, the target variable is the difference between the fields shown in Figs. 1a and 1b.

To build a statistical model capable of simultaneously approximating the variability of ΔT in space and time, we propose an original approach that originates from the problem of approximating the temporal dynamics of ΔT at a specific point, considered in our previous study [24]. In that study, observed UHI intensity in the center of Moscow was approximated based on a set of meteorological predictors derived from ERA5 reanalysis and/or observations at rural stations around Moscow. We further refer to the formulation of the ML problem used in [24] as local, implying that the target variable and predictors are defined for a given point and that the spatial structure of their fields is not taken into account.

Here, we extend the local problem statement from [24] into the spatial dimension. We assume that the thermal environment at a given point (a model grid cell) is determined by large-scale weather conditions, local surface properties at that point, and nonlocal surface properties that characterize the surroundings of this point at different scales. This hypothesis follows from general ideas about the structure of the atmospheric boundary layer over a nonuniform surface, where the oncoming airflow accumulates and mixes the impacts of individual elements of this surface (individual grid cells in the model). Meanwhile, such hypothesis is supported by previous observational and modeling studies for Moscow. In particular, it is expressed in the dependence between the UHI spatial structure and surface properties at different scales according to official and crowdsourced observations [19], the dependence between the size of urban parks and their cooling effect based on high-resolution COSMO simulations [25], in the existence of urban heat plumes [20].

Based on the abovementioned hypothesis, we propose an original approach to account for nonlocal dependences between the target variable and the predictors in both temporal and spatial dimensions with the framework of a ML problem. The target variable

$\Delta T_{i,j}$ is approximated based on the values of meteorological and surface predictors for a given point, as well as a set of additional features derived from the data for neighboring points in time and space. So-called temporal features (TFs) are used to account for delayed connections between ΔT and meteorological conditions. TFs are defined as $f(W_{k,i...i-m_t,j})$, meaning they are functions of the values of the predictor W_k for the previous m_t moments in time. We use two types of TFs: moving averages (MAs) and tendencies, which demonstrated significant importance in our previous study [24].

By analogy with TFs, we propose to account for nonlocal relationships between ΔT and surface predictors by generating additional spatial features (SFs) for the jth point using data on the surface properties of other points in its vicinity. This involves applying convolution functions of the form $g(S_{k,j...j-m_s})$, where the other m_s points are indexed by their distance from the jth point. The proposed formulation of the ML problem can be expressed as follows:

$$\Delta T_{i,j} = F\{W_{1,i,j}, f_1(W_{1,i...i-m_t,j}), ..., f_{l_t}(...),$$
$$...,$$
$$W_{n_w,i,j}, f_1(W_{n_w,i...i-m_t,j}), ..., f_{l_t}(...),$$
$$S_{1,j}, g_1(S_{1,j...j-m_s}), ..., g_{l_s}(...),$$
$$...,$$
$$S_{n_s,j}, g_1(S_{n_s,j...j-m_s}), ..., g_{l_s}(...)\}.$$

It is important to note that convolution functions $g(...)$ can also include meteorological predictions W_k, such as wind speed components, as additional arguments; however, we do not indicate this in the notation for clarity. The design of such convolution functions is described in detail in the following subsections. In either case, it is crucial that all features used for the ML-based approximation of $\Delta T_{i,j}$ are formally defined locally (for the ith moment in time and the jth point in space), even though their calculation may depend on data from other points. Therefore, we refer to the presented approach as quasi-local.

2.3. Meteorological Predictors

To characterize large-scale conditions and define meteorological predictors, we use simulation results from the noURB numerical experiment, smoothed with a running mean using a kernel of 50 by 50 km, which approximately corresponds to the size of the ERA-Interim reanalysis grid cell in the Moscow region. Figure 1c demonstrates how this smoothing works using the example of 2-m air temperature. We do not use meteorological data from the ERA-Interim reanalysis, which serves as forcing for COSMO simulations, because there may be significant differences between the regional model and the reanalysis in reproducing the large-scale synoptic situation, while our interest lies in the characteristics of the atmospheric processes that directly correspond to the simulated ΔT values.

As specific meteorological predictors W_k we use the following variables, which are directly available in COSMO output or are calculated based on their basis:

- 2-m air temperature (t2m);
- 2-m relative humidity (rh2m);
- 2-m specific humidity (qv2m);
- difference between surface temperature and 2-m air temperature (strat2m);
- 10-m wind speed (vel10m);
- sea level pressure (slp);
- total precipitation (tp);
- net longwave radiation (str);
- dowelling shortwave radiation (ssrd);
- weather factor, an empirical function from wind speed and cloudiness [24, 26] (WF).

Generally, the set of meteorological predictors is close to those one used in [24]. As noted earlier, in addition to the instantaneous values of W_k for the ith time moment, we use TFs derived from these values as $f(W_{k,i...i-m_t,j})$, specifically tendencies and moving averages (MAs). Initially, we planned to use the same set of TFs calculated for periods of 3, 6, and 12 h for all meteorological predictors. However, due to computational limitations, we were forced to significantly reduce the dataset's size. Therefore, for each predictor, the TFs were selected based on expert judgment, taking into account the results of the feature significance assessment from [24]. For slp and tp, we use only instantaneous values and do not include any TFs. For t2m and rh2m, we use 24-h MAs and 3, 6, and 12-h tendencies. For qv2m, we use only 24-h MAs. Moreover, we do not include instantaneous values for t2m, rh2m, and qv2m. For the remaining variables, we use instantaneous values and 3, 6, and 12-h MAs, but we do not use tendencies. We do not include horizontal wind speed components, u and v, as explicit meteorological predictors; however, they are used as additional arguments in convolution filters for surface predictors as described below.

2.4. Surface Predictors

For the surface predictors S_k, we utilize external city-descriptive parameters that characterize the urban-atmosphere interaction in the COSMO model. Among these parameters, the most significant is the fraction of urban area within model grid cells, where the urban parameterization TERRA_URB is applied (FR_URBAN). Other relevant parameters include building height (URB_H_BLD), the aspect ratio of street canyons (URB_H2W), and the total volume of buildings (URB_VOL_BLD), which is derived as the product of URB_H_BLD and the building area fraction. This total volume is used as a weight for the spatial redistribution of anthropogenic heat emissions [20]. Additionally, we incorporate surface height (HSURF). Each of these parameters is originally defined for 1-km grid cells of the COSMO model, and we refer to such values as local.

To account for nonlocal dependences between ΔT and surface properties within the framework of the proposed quasi-local approach, we generate three types of additional spatial features (SFs) by applying convolution functions to the fields of surface predictors. Firstly, we use static SFs, which are derived using a circular filter that smooths the original fields of surface properties over a specified radius. This filter is implemented using the *convolve* function from the *scipy.ndimage* Python library, utilizing an isotropic circular kernel. We apply this filter with a set of radii: 1, 2, 5, 10, and 15 km. Figures 2b and 2c illustrates the results of this transformation using urban fraction data (FR_URBAN predictor) as an example. Based on findings from [19], we anticipate that this approach will effectively account for diffusion associated with quasi-isotropic mixing in the atmospheric boundary layer.

A more challenging issue is to develop convolution functions capable of accounting for anisotropic mixing associated with horizontal heat transfer, specifically advection along the airflow. It is known that warmer air from the city center is transported to the leeward side of the city and further to rural areas; this effect is referred to as UHI advection [18, 27] or urban heat plumes [20, 28]. To address these effects, we propose two types of dynamic wind-dependent SFs. The SFs of the second type are defined as gradient-based advection terms, calculated for each moment in time based on a field of the given surface predictor S_k and the large-scale horizontal wind speed components u and v, as follows:

$$g_{\text{grad}X}(S_{k,j\ldots j-m_s}, u_{i,j}) = u_{i,j}\left(\frac{dS_k}{dx}\right)_j,$$

$$g_{\text{grad}Y}(S_{k,j\ldots j-m_s}, v_{i,j}) = v_{i,j}\left(\frac{dS_k}{dy}\right)_j,$$

where the spatial gradients of S_k are approximated numerically using *gradient* function from the *numpy* Python library. We calculate these gradient-based advection terms for the original fields of the surface predictors, as well as for the same fields smoothed with a 1-km radius, assuming that the resulting SFs will help approximate heat advection on the scale of neighboring grid cells.

SFs of the third type are proposed to account for heat advection on larger scales based on a dynamical self-adjusting directional filter. This filter depends on wind speed and direction, performing an averaging of surface properties in the upwind area relative to the jth point. Conceptually, this filter is somewhat similar to the concept of a footprint, which is widely used in micrometeorological studies [29]. It aggregates surface properties over an upwind area, serving as a "mesoscale footprint." We use two parameters to define the filter's kernel: the time scale τ (in hours) and the aspect ratio α (unitless).

The application of the filter for the ith time step consists of several steps. First, the horizontal wind speed components u and v are averaged over the preceding τ hours, providing $\overline{u}$ and $\overline{v}$ values. Next, we apply an elastic transformation to the original field of the surface predictor S_k, which involves interpolating the data from the original grid to new nodes defined for each point as follows:

$$x_{j,\text{new}} = x_{j,\text{old}} + \frac{N_s \overline{u}_j \tau}{2\Delta x},$$

$$y_{j,\text{new}} = y_{j,\text{old}} + \frac{N_s \overline{v}_j \tau}{2\Delta y},$$

where x and y are indices of the model grid, $\Delta x = \Delta y = 1000$ m are model grid steps, $N_s = 3600$ is the number of seconds in one hour. This transformation is implemented using *map_coordinates* function from the *scipy.ndimage* Python library. It shifts the predictor field in such a way that the region of length $L = N_s\tau\sqrt{\overline{u}_j^2 + \overline{v}_j^2}$ downwind from the jth point becomes centered at this point.

In the next step, the transformed field of the surface predictor is smoothed using an anisotropic convolution filter, the properties of which depend on wind speed and direction. We employ an elliptical filter with length L and width αL, rotated along the wind direction, with weights equal to 1 inside the ellipse and 0 outside. Initially, the intention was to determine the filter kernel for each point based on the local wind speed components $\overline{u}_j$ and $\overline{v}_j$. However, at this stage of the work, in order to simplify and expedite the computation, we use the *convole* function from the *scipy.ndimage* Python library, which assumes a constant kernel across the entire domain. To account for spatial variability in wind speed and direction, we

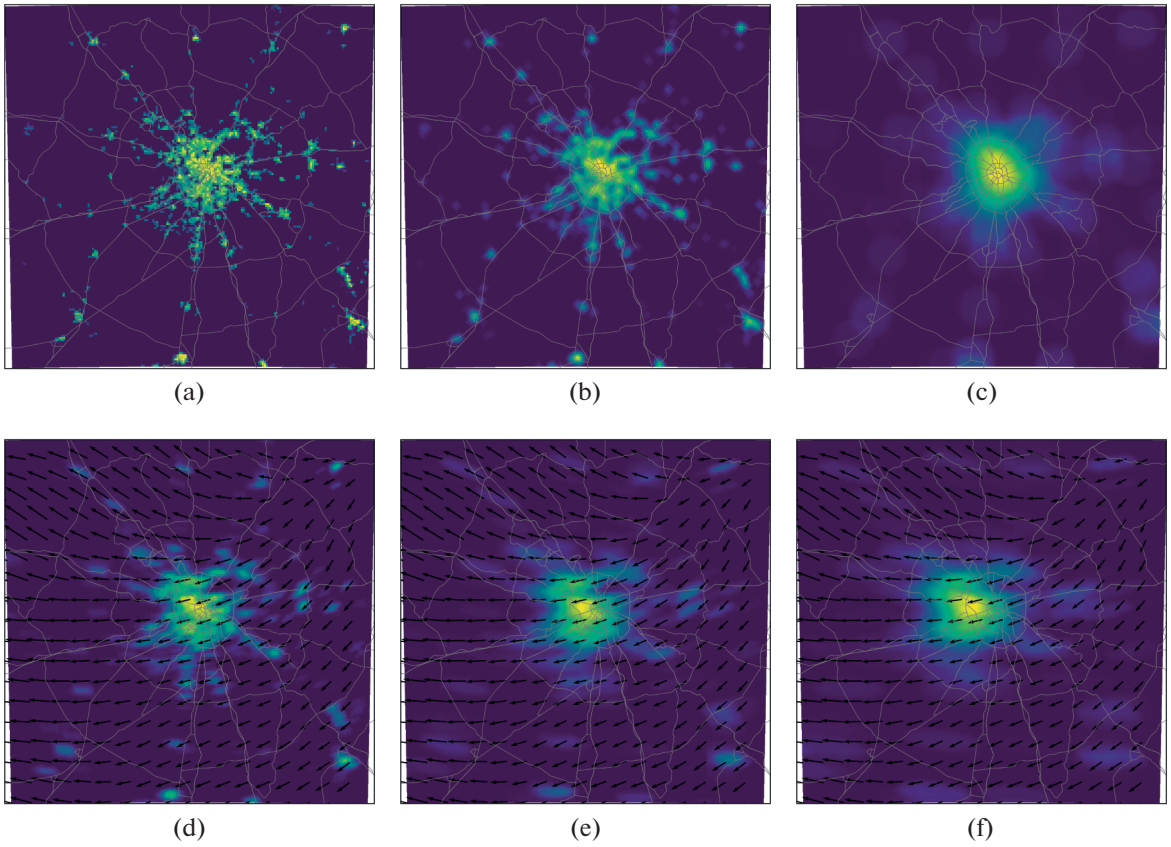

Fig. 2. The original field of a surface predictor FR_URBAN, i.e., the fraction of urban area in the model grid cells (a) and additional features generated on its basis using the static circular filter with a radius of 2 km (b) and 10 km (c), and using self-adjusting dynamic directional filter for a specific time moment (08.08.2010, 00 UTC) with τ of 2 (d), 4 (e), and 6 h (f) and $\alpha = \frac{1}{4}$. Gray lines indicate the primary road network according to OpenStreetMap data. Black arrows in (d–f) indicate wind vectors according to blurred data of noURB simulations, which defines the structure of dynamic directional filter.

divide the domain into several subdomains, for each of which the filter kernel is determined independently based on the subdomain-averaged wind speed components $\overline{\overline{u}}$ and $\overline{\overline{v}}$. Specifically, we split the full domain (180 × 180 grid cells) into 3 × 3 subdomains. In the marginal zones close to the subdomain borders (25% of the subdomain's grid cells from each side), a linear relaxation technique is applied to the convolution results to avoid artifacts associated with a sharp change in the convolution kernel. Using this proposed method, we generate a set of additional features corresponding to different time scales τ (1, 2, 3, 4, and 6 h) and different aspect ratio α ($\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$), assuming that the ML model should determine how to utilize this data on its own. Figures 2c–2d demonstrate examples of resulting SFs derived based on the URBAN_FR surface predictor for a specific case with prevailing eastern winds.

Due to the computational and memory limitations, we generate additional SFs only two surface predictors, FR_URBAN and URB_BLD_VOL. For the others, namely URB_H_BLD, URB_H2W and HSURF, we use only their local values as features.

2.5. Input Dataset

The construction of the final dataset used for ML model training and testing consists of several steps:

1. We prepare the target variable, meteorological predictors, and TFs as 3D (spatio-temporal) arrays.

2. We prepare surface predictors and static spatial features (SFs), derived using a circular convolution filter, as 2D arrays and repeat them along the third dimension so that the shape of the resulting 3D array coincides with that of the meteorological predictors.

3. We prepare dynamic SFs derived as gradient-based advection terms or using a dynamic self-adjusted directed convolution filter for each time moment and concatenate the resulting 2D arrays into 3D arrays.

4. We reshape 3D arrays of all features into 1D arrays and concatenate them together into a single 2D array, creating a tablelike dataset with the target variable and all considered features.

5. To reduce the number of features and resulting volume of the data array, we perform dimensionality reduction using principal component analysis (PCA) transformation from the *scikit-learn* Python library. We apply PCA to several groups of features, namely tendencies and moving averages (MAs) calculated over different periods, and retain the resulting components that together explain at least 99% of the variance. Typically, when applying PCA to the tendencies or MAs calculated over 3, 6, and 12 h, this transformation retains 2 features instead of 3. We also apply the PCA transformation to the features derived by smoothing the spatial predictors FR_URBAN and URB_BLD_VOL using a static circular filter. After applying PCA transformations, 75 features remain.

To compare the relevance of different groups of features, we prepare three subsets of features that differ in their approaches to accounting for surface predictors. The full set of 75 features, which includes all types of SFs derived to account for both isotropic diffusion and advection, is referred to as **SF_adv**. We also considered the subsets **SF_loc** and **SF_dif**, which contain fewer features. **SF_loc** includes only local values of the surface predictors, while **SF_dif** also includes static SFs derived using a static circular filter. The structure of these subsets is summarized in Table 1.

The full volume of available data from COSMO simulations includes 180×180 grid cells in the spatial dimension and 29 520 time steps (hourly data for periods from May to August over 10 years, 2007–2016). However, for this study, we use a significantly smaller test subset with data for a subdomain comprising 80×80 grid cells centered over Moscow and covering two years, 2010 and 2011, which represents about 5% of the full volume of available data. Nevertheless, after reshaping it into a tablelike format, this subset contains about 38×10^6 rows. With 76 columns (the target variable and 75 features), this results in about 2.8×10^9 values, which requires 11.2 GB of memory when using the float data type. Attempts to further increase the volume of the test subset were limited by the memory constraints of the personal computer used for data processing. We plan to increase the volume of data in the future by using more advanced computing platforms and/or implementing step-by-step training; however, in this study, we focus on the results obtained for the test subset.

2.6. Machine Learning Model

As an ML model, we use a gradient boosting regression method implemented in the form of the CatBoost algorithm, an open-source boosting model developed by Yandex [30] and available as the *catboost* Python library. CatBoost regression demonstrated the best quality metrics compared to several other ML models, including random forest regression, support vector regression, and multilayer perceptron, in our previous study focused on UHI's temporal dynamics at a fixed point [24]. Given the large volume of the dataset, an important advantage of using CatBoost is its support for parallelization on GPUs and its overall high training speed compared to other models. We use CatBoost regression with fixed hyperparameters: the number of ensemble members (*n_estimators*) is set to 1000, and all other parameters are set to their defaults. Hyperparameter tuning, as well as a comparison between different ML models, is planned as part of further work.

2.7. Model Evaluation

To evaluate model quality metrics, we use a blockwise cross-validation approach, similar to that in [24]. We split our dataset along the time dimension using the blocked k-fold method, with a training block size of 15 days and a test block size of 5 days, resulting in a train-to-test ratio of 3 : 1, and five iterations for rearranging these blocks ($K = 5$). In the first iteration, the first 15-day block of the dataset is included in the training subset, while the subsequent 5-day block is included in the testing subset, and so on. In each subsequent iteration, all blocks are shifted to the right by $(5 + 15)/5 = 4$ days. This procedure yields five instances of the trained ML model and five values for each quality metric estimated over the test subsets, allowing us to assess their uncertainty. As quality metrics, we use the root mean square error (RMSE), the mean error (ME), and the determination coefficient (R^2).

3. RESULTS

3.1. Model Training Performance

Since the ML problem in our study is solved using a substantial amount of data (2.8×10^9 values in the SF_adv subset), the performance of ML algorithms during the training phase becomes critical. Initially, model training was performed using a high-end laptop (Intel Core i7-11800H, 24 GB RAM, GeForce

Table 1. Summary of the feature subsets used to approximate spatiotemporal dynamics of the UHI

Subset name	Meteorological predictors and derived TFs	Surface predictors and derived SFs				Number of features
		Local values	Static SFs	Gradient-based advection terms	SFs based on the dynamical directional filter	
SF_loc	+	+	−	−	−	33
SF_dif	+	+	+	−	−	37
SF_adv	+	+	+	+	+	75

RTX 3050 Ti Laptop GPU), and the size of the dataset was configured to be close to the limits of the laptop's capabilities. The key limitation was memory size; however, the learning time was also significant. For the GPU version of the CatBoost algorithm ($task\_type = 'GPU'$), training for one fold using 75% of the full dataset required about 20–30 min, resulting in approximately 6 h to train ML models across all five folds and three feature subsets. The acceleration achieved by using the GPU-based version of the ML model reached up to 2–3 times compared to the CPU-based version. We have also performed preliminary tests using a high-performance computing (HPC) system, namely, the MSU-270 supercomputer of the Lomonosov Moscow State University. We used one computational node of the supercomputer, equipped with 8 high-end GPU devices, 128 CPUs and 2 TB RAM. Not surprisingly, we achieved much higher training performance here, with training taking only about 35 s per fold for the GPU-based version of the ML model, resulting in a speedup of about 50 times compared to the laptop.

3.2. UHI Spatial Patterns

The primary novelty of the methods used in our study lies in a new approach to approximating the spatial structure of local meteorological anomalies, exemplified by the UHI, through the generation of additional features (SFs) designed to account for nonlocal dependences with surface properties. Consequently, the analysis of the results is primarily aimed at investigating and demonstrating the added value of these additional SFs. To achieve this, we consider the results of the statistical approximation of the UHI, characterized by the value ΔT, by ML models based on three sets of features: SF_loc, SF_dif and SF_adv (Table 1). First, we examine the spatial patterns of the seasonally averaged urban anomalies of daily mean, daytime, and nighttime air temperature (Fig. 3). It is important to note that for the ML-based models, averaging is performed only over their predictions for the test subsets. As in our previous study [24], the ML-based models accurately reproduce the diurnal cycle of ΔT with nocturnal maximum and daytime minimum (Figs. 3b and 3c) and its synoptic-scale variations; however, the latter is not shown.

A comparison of the results of ML-based statistical approximation using different feature subsets demonstrates similar pattern for all three types of seasonal averaging. ΔT fields from COSMO simulations and predictions of the ML model appear very similar when using the SF_dif and SF_adv feature subsets, both of which include nonlocal SFs. In contrast, there are noticeable differences in the ΔT fields predicted based on the SF_loc subset. For example, urban parks are cooler than the surrounding built-up areas but still warmer than rural areas according to COSMO simulations, which is also confirmed by observations, e.g., in [19]. However, ML model predicts there as low ΔT values as for rural areas in case of SF_loc subset, resulting in much sharper temperature gradients between parks and surrounding built-up areas. This is not surprising, as the ML model, forced by the SF_loc subset, does not have any information about the fact that urban parks are surrounded by built-up areas and therefore cannot predict the associated warming. In contrast, the SF_dif and SF_adv subsets provide this information through additional nonlocal SFs. This example visually demonstrates the added value of the nonlocal dependences between ΔT and surface properties, represented by the SFs.

The difference between the predictions of the ML model based on three different feature subsets becomes even more pronounced when considering specific cases, particularly those where the UHI is shifted by the airwlow to the leeward side of the city, creating urban heat plumes. Figure 4 presents the spatial patterns of ΔT for three such cases with different wind directions, where the UHI is shifted in the southern, southwestern, and northeastern directions according to COSMO simulations. Not surprisingly, ML models supplied only with local surface properties (SF_loc subset) or with static SFs (SF_dif

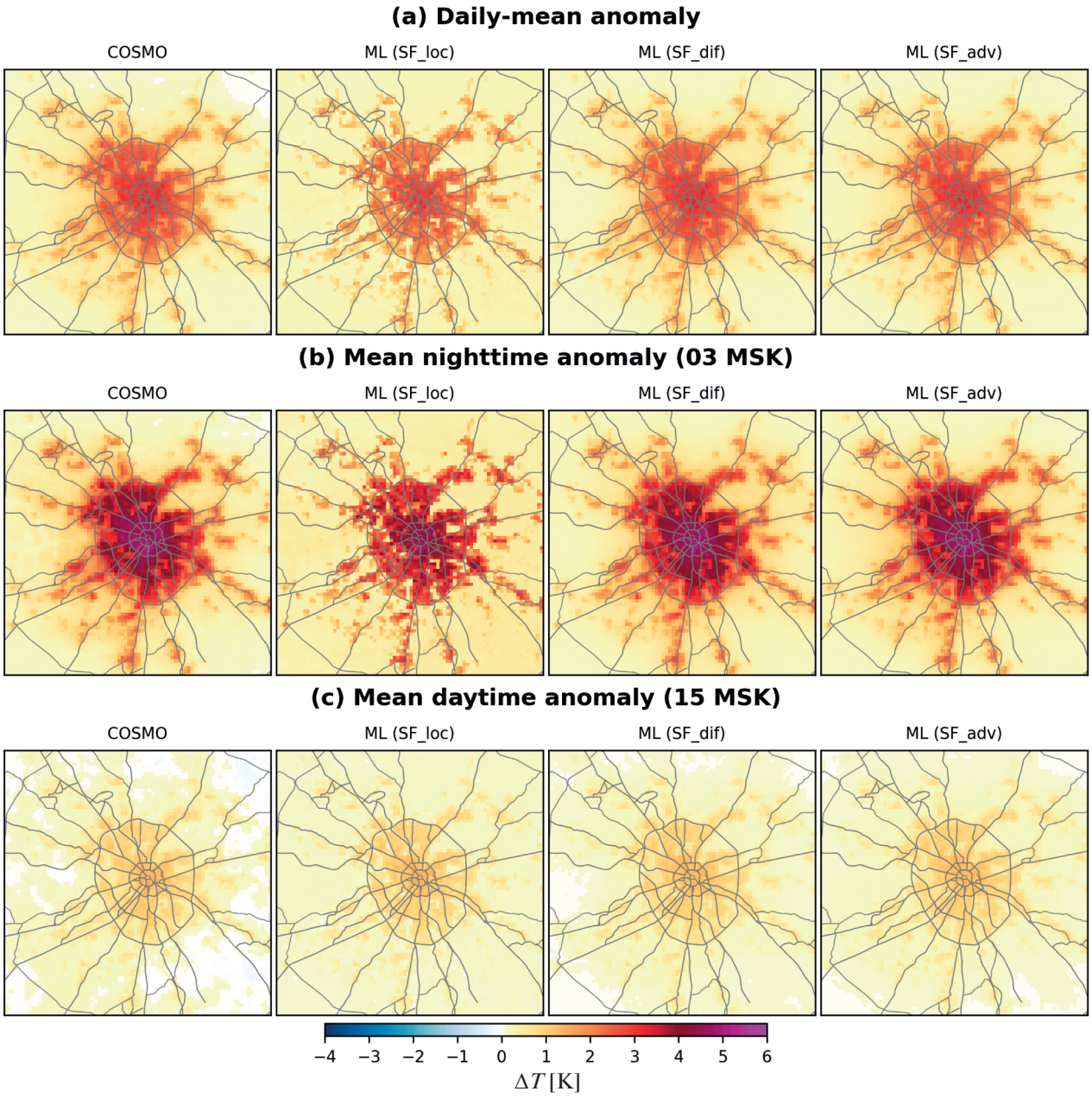

Fig. 3. Spatial patterns of the (a) daily mean, (b) mean nighttime, and (c) mean daytime values of ΔT averaged over two warm seasons according to COSMO simulations and predictions of ML model based on different feature subsets.

subset) do not reproduce these effects. In contrast, a closer correspondence with COSMO simulations is achieved by the ML model supplied with the SF_adv feature subset, which includes SFs derived using a self-adjusted dynamic directional filter. In the case of the SF_adv subset, the ML model reproduces the shift of the UHI to the leeward side of the city and further downwind to rural areas. Moreover, for these cases, the SF_adv subset allows for a more accurate reproduction the location of UHI hotspot with maximum values of ΔT, which are found downwind from the city center.

3.3. Model Quality Metrics

Improvements in the reproduction of ΔT, achieved by considering its nonlocal with surface properties in SF_dif and SF_adv feature subsets, are confirmed not only by visual comparisons, as shown in Figs. 3 and 4, but also by formal quality metrics evaluated through cross-validation. Figure 5 presents the results of cross-validation in the form of comparisons

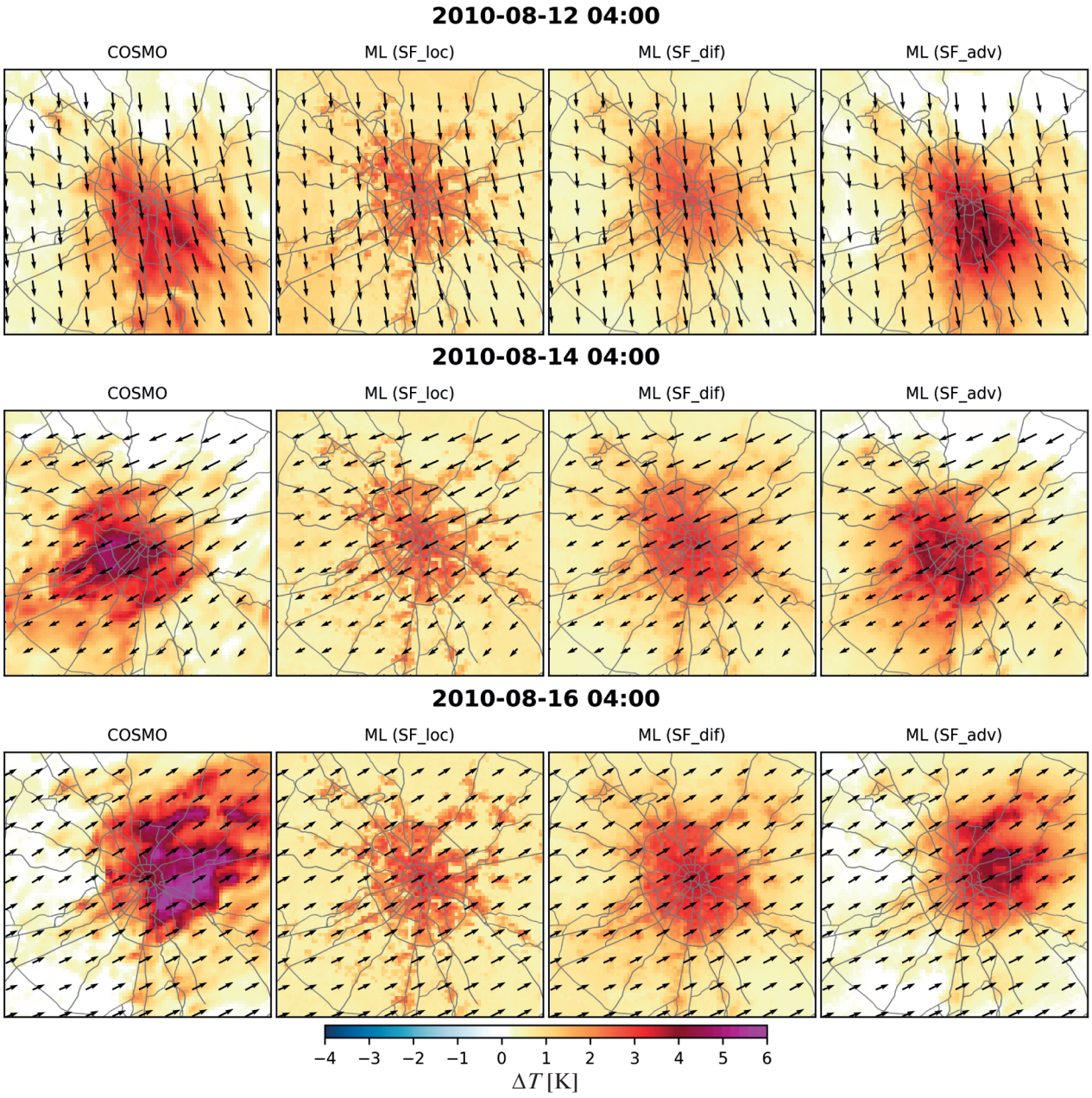

Fig. 4. Spatial patterns of ΔT according to COSMO simulations and predictions of ML model based on different feature subsets for three cases with a clear UHI advection effect and different prevailing wind direction (wind vectors are shown by black arrows).

between the ground truth (COSMO simulations) and ML-based predictions, illustrated as hexagonal binning diagrams for the full data sampling, i.e., all time moments with an hourly interval over the May–August period for the years 2010 and 2011 (a), and separately for nocturnal hours (0–6 MSK) (b), when the diurnal maximum of the UHI magnitude is typically reached. For both samplings, there are clear improvements achieved with the SF_dif subset compared to the SF_loc subset, and further improvements for SF_adv against SF_dif. These enhancements are reflected in the shape of the data cloud, which becomes more compact and closer to the one-to-one line, as well as in the quality metrics, namely RMSE and R^2. The RMSE decreases from 0.72 K for SF_loc to 0.53 K for SF_adv, i.e., by 28% for sampling with all time moments, and from 0.86 to 0.54, i.e. by 37% for nocturnal sampling. A similar order of improvement is observed in the R^2 metric. It is important to note that only nonlocal surface features allow for the reproduction of extreme values

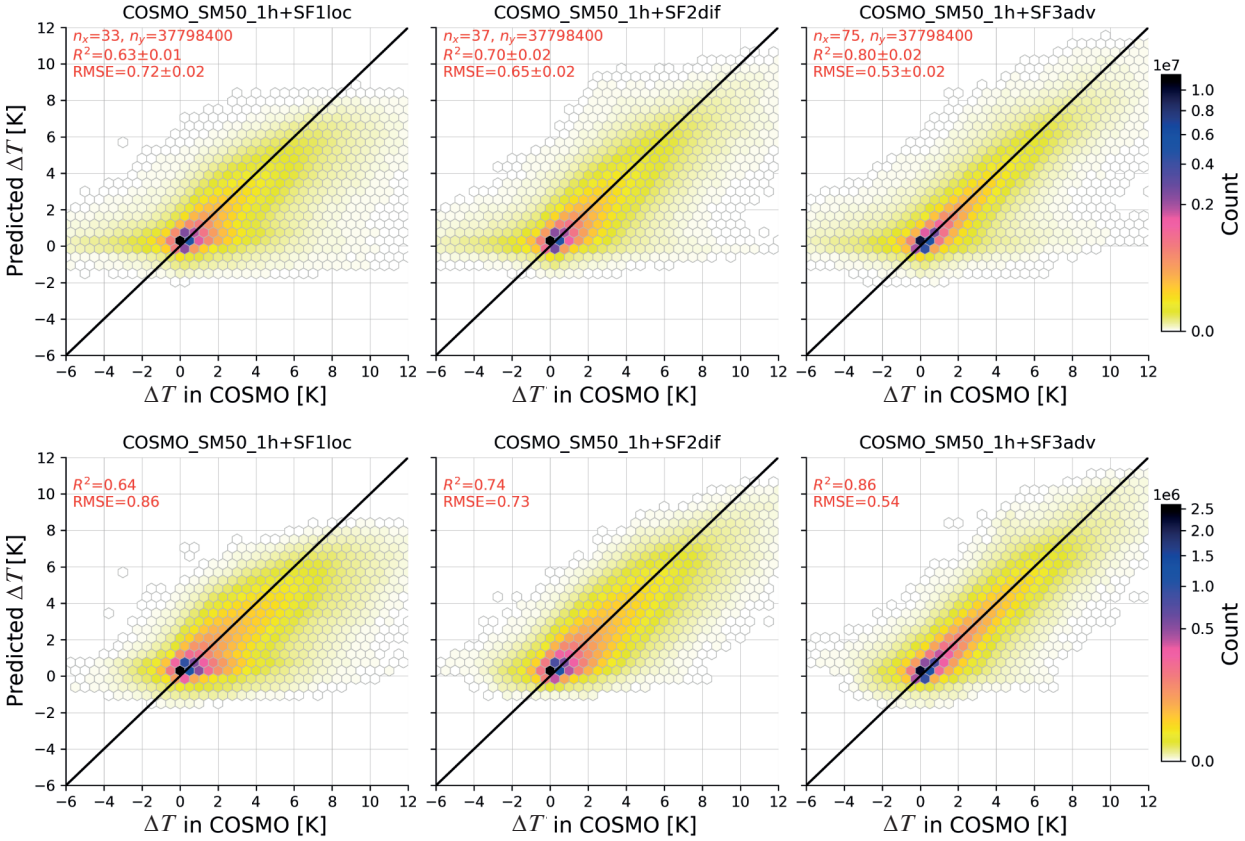

Fig. 5. Hexbin plots showing comparison between ΔT values simulated by COSMO model, which are considered as a ground truth, and predicted by ML model based on different SF_loc (left column), SF_dif (center column) and SF_adv (right column) feature subsets, for all available time moments (upper row) and only for nocturnal hours, 0–6 MSK (lower row). Each subplot contains the values of quality metrics RMSE (°C) and R^2, as well as their uncertainty, characterized by standard deviation according to blocked k-fold method, upper plots also contain the size of the dataset, i.e., number of features (n_x) and rows (n_y).

of $\Delta T > 8$ K, which remain underestimated by the ML model when using the SF_loc feature subset.

4. CONCLUSIONS AND OUTLOOK

In this study, we employed a data-driven machine learning (ML) model to simulate the spatiotemporal dynamics of the urban-induced temperature anomaly, known as the urban heat island (UHI) in the Moscow megacity. This task was framed within the context of a statistical downscaling problem, which involves reconstructing a high-resolution field of the target meteorological variable based on low-resolution meteorological data and high-resolution surface properties. For training the ML models, we utilized results from high-resolution (1-km grid spacing) hydrodynamic simulations of the meteorological regime in the Moscow region, conducted using the COSMO regional atmospheric model coupled with the TERRA_URB urban canopy parameterization. The current results are based on simulations for the warm seasons (May–August) of two years, 2010 and 2011. The target variable, ΔT, was defined as the difference between fields of 2-m air temperature simulated by COSMO in numerical experiments with (URB) and without (noURB) urban effects taken into account. For meteorological predictors, we used a set of COSMO output variables from the noURB numerical experiment, smoothed with a running mean using a kernel size of 50 by 50 km. For surface predictors, we utlilized a set of external parameters utilized by the TERRA_URB scheme in COSMO, with urban area fraction being the most significant. We employed a gradient boosting method in form of the CatBoost algorithm with GPU support as ML model.

The distinctive feature of our study is an original quasi-local approach to defining the feature vectors, proposed to account for nonlocal connections between the target variable, ΔT, and surface properties. This approach involves localizing the variables of the feature description at individual points (nodes

of the computational grid of the COSMO model) while incorporating additional spatial features (SFs) calculated based on the values of predictors at neighboring points within a specified area using convolution filters. We employ asimple circular filters with varying radii, as well as more advanced self-adjusting directional filters that consider large-scale data on wind speed and direction. The latter include gradient-based advection terms and a directional isotropic filter, which combines elastic transformation and convolution with an elliptic kernel. To demonstrate the added value of different types of additional SFs, we consider three distinct subsets of features: SF_loc, SF_dif and SF_adv. Here, SF_loc contains only the local values of spatial predictors, SF_dif includes the SFs derived from the circular filter, and SF_adv encompasses all discussed features.

The results currently available are largely preliminary due to the use of only a small portion of the available data, a lack of hyperparameter tuning, and certain expert restrictions in the formation of the feature set aimed at reducing memory consumption. Nevertheless, even these preliminary results allow us to draw important conclusions. Firstly, we demonstrated the fundamental possibility of simultaneously approximating the spatial and temporal variability of ΔT with a ML model. The trained ML model was able to reproduce key patterns of this variability, including the diurnal cycle, dependence on synoptic conditions, and differences between the city center and periphery, as well as between built-up areas and urban parks.

Secondly, we highlighted the importance of nonlocal interactions between predictors for accurately reproducing the urban heat island (UHI) spatial structure. The nonlocal spatial features (SFs) derived using a static circular filter in the SF_dif subset allowed for a smoother representation of seasonally averaged ΔT fields compared to surface properties, including the positive ΔT in urban parks, while in case of SF_loc subset there are as low ΔT values as in rural areas. The nonlocal SFs derived using self-adjusting directional filters allowed to reproduce the dependence of the ΔT field on wind speed in direction, including the shift of UHI hotspot to the leeward side of the city and heat plumes over the downwind rural areas. Qualitative improvements in reproducing UHI spatial features were accompanied by enhancements in quantitative cross-validation scores. The use of the SF_adv subset reduced the root mean square error (RMSE) by 28% compared to the SF_loc subset. An even larger reduction of RMSE by 37% was observed during nocturnal hours, when UHI is most pronounced.

Presented results allow to outline several research directions for further studies. The priority task is to expand the size of the dataset using the entire available volume of COSMO simulation data from [20] for 10 summer seasons, instead of the currently utilized data for just 2 seasons and a smaller subdomain. To get around the dataset size limitations we encountered, we plan to leverage HPC infrastructure, including the MSU-270 supercomputer, and explore opportunities for batch training available for CatBoost.

The successful application of the proposed approach for ML-based approximation of the urban-induced temperature anomaly provides strong motivation for testing it with other types of local meteorological anomalies, such as associated with orography or water bodies, as well as with other meteorological variables like near-surface humidity or wind speed. If successful, the combination of ML-based statistical models for approximating various local meteorological anomalies could serve as a computationally efficient technique for the statistical downscaling of coarse meteorological data, such as global weather forecasts or climate projections.

From a methodological perspective, it is interesting to compare the proposed quasi-local approach with convolutional neural networks (CNNs), which are increasingly popular for statistical downscaling [6]. Using the terminology introduced in this study, the CNN-based approach can be called as nonlocal, since the CNNs take two-dimensional meteorological fields as input features. Both approaches have their advantages. The current approach rely on the expertly-defined convolution functions, while in CNNs, they are optimized during training. Given the demonstrated significant contribution of nonlocal connections between the target variable and predictors, one might expect better approximation quality from CNNs. On the other hand, the proposed quasi-local approach may exhibit greater generalization ability due to its lack of rigid dependence on domain configuration, as well as better interpretability and flexibility, in particular due to the possibility of using classical ML models, such as the currently employed gradient boosting method.

From the perspective of atmospheric physics, it is important to further develop the concept of a “mesoscale footprint” that justifies the use of dynamic directional filters for generating additional SFs. Currently, these filters are determined solely by wind speed and direction, although it seems evident that their configuration should also depend on the properties of the atmospheric boundary layer, such as height and intensity of turbulent mixing. Another open question is the distribution of weights within the filter kernel. The currently employed kernel with uniform weights appears to be overly simplified. It can be assumed that adapting the footprint theory [29] to

the mesoscale will enable the development of more physically-based convolutional filters.

ACKNOWLEDGMENTS

The research was carried out using the MSU-270 supercomputer of the Lomonosov Moscow State University.

FUNDING

The work in field of machine learning was supported by the Noncommercial Foundation for the Advancement of Science and Education, INTELLECT. The work on development of the methodology for constructing dynamic directional filters within a concept of "mesoscale footprint" was supported by the Russian Science Foundation, project no. 24-17-00155.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. L. Chen, B. Han, X. Wang, J. Zhao, W. Yang, and Zh. Yang, Appl. Sci. **13**, 12019 (2023). https://doi.org/10.3390/app132112019
2. C. O. De Burgh-Day and T. Leeuwenburg, Geosci. Model. Dev. **16**, 6433 (2023). https://doi.org/10.5194/gmd-16-6433-2023
3. M. A. Krinitskiy, V. M. Stepanenko, A. O. Malkhanov, and M. E. Smorkalov, Supercomputing Frontiers and Innovations **9**, 19 (2022). https://doi.org/10.14529/jsfi220302
4. R. Keisler, arXiv Preprint (2022). https://doi.org/10.48550/arXiv.2202.07575
5. R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, Sh. Mohamed, and P. Battaglia, Science **382**, 1416 (2023). https://doi.org/10.1126/science.adi2336
6. Yo. Sun, K. Deng, K. Ren, J. Liu, Ch. Deng, and Yo. Jin, ISPRS J. Photogramm. Remote Sens. **208**, 14 (2024). https://doi.org/10.1016/j.isprsjprs.2023.12.011
7. T. R. Oke, G. Mills, A. Christen, and J. A. Voogt, *Urban Climates* (Cambridge University Press, Cambridge, 2017). https://doi.org/10.1017/9781139016476
8. K. V. Wong, A. Paddon, and A. Jimenez, Journal of Energy Resources Technology **135**, 022101 (2013). https://doi.org/10.1115/1.4023176
9. M. A. Tarasova, M. I. Varentsov, and V. M. Stepanenko, Izv., Atmos. Ocean Phys. **59**, 111 (2023). https://doi.org/10.1134/S0001433823020068
10. G. S. Rivin, I. A. Rozinkina, R. M. Vil'fand, D. B. Kiktev, K. O. Tudrii, D. V. Blinov, M. I. Varentsov, D. I. Zakharchenko, T. E. Samsonov, I. A. Repina, and A. Yu. Artamonov, Russ. Meteorol. Hydrol. **45**, 455 (2020). https://doi.org/10.3103/s1068373920070018
11. T. Gardes, R. Schoetter, J. Hidalgo, N. Long, E. Marquès, and V. Masson, Sci. Total Environ. **737**, 139253 (2020). https://doi.org/10.1016/j.scitotenv.2020.139253
12. Z. S. Venter, O. Brousse, I. Esau, and F. Meier, Remote Sens. Environ. **242**, 111791 (2020). https://doi.org/10.1016/j.rse.2020.111791
13. A. Oliveira, A. Lopes, S. Niza, and A. Soares, Sci. Total Environ. **805**, 150130 (2022). https://doi.org/10.1016/j.scitotenv.2021.150130
14. Ch. Yi, Yi. Shin, and J.-W. Roh, Atmosphere **9**, 164 (2018). https://doi.org/10.3390/atmos9050164
15. C. L. Muller, L. Chapman, C. S. B. Grimmond, D. T. Young, and X. Cai, Int. J. Climatol. **33**, 1585 (2013). https://doi.org/10.1002/joc.3678
16. D. Milošević, S. Savić, M. Kresoja, Z. Lužanin, I. Šećerov, D. Arsenović, J. Dunjić, and A. Matzarakis, Int. J. Biometeorol. **66**, 371 (2022). https://doi.org/10.1007/s00484-020-02058-w
17. F. Meier, D. Fenner, T. Grassmann, M. Otto, and D. Scherer, Urban Clim. **19**, 170 (2017). https://doi.org/10.1016/j.uclim.2017.01.006
18. O. Brousse, C. Simpson, N. Walker, D. Fenner, F. Meier, J. Taylor, and C. Heaviside, Environ. Res. Lett. **17**, 044041 (2022). https://doi.org/10.1088/1748-9326/ac5c0f
19. M. Varentsov, D. Fenner, F. Meier, T. Samsonov, and M. Demuzere, Front. Environ. Sci. **9**, 7169681 (2021). https://doi.org/10.3389/fenvs.2021.716968
20. M. Varentsov, H. Wouters, V. Platonov, and P. Konstantinov, Atmosphere **9**, 50 (2018). https://doi.org/10.3390/atmos9020050
21. D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart, Q. J. R. Meteorol. Soc. **137**, 553 (2011). https://doi.org/10.1002/qj.828
22. H. Wouters, M. Demuzere, U. Blahak, K. Fortuniak, B. Maiheu, J. Camps, D. Tielemans, and N. P. M. Van Lipzig, Geosci. Model. Dev. **9**, 3027 (2016). https://doi.org/10.5194/gmd-9-3027-2016

23. V. S. Platonov, M. I. Varentsov, Yu. I. Yarinich, A. N. Shikhov, and A. V. Chernokulsky, Urban Clim. **54**, 101837 (2024).
https://doi.org/10.1016/j.uclim.2024.101837
24. M. Varentsov, M. Krinitskiy, and V. Stepanenko, Climate **11**, 200 (2023).
https://doi.org/10.3390/cli11100200
25. M. Varentsov, V. Vasenev, Yu. Dvornikov, T. Samsonov, and O. Klimanova, Sci. Total Environ. **902**, 165966 (2023).
https://doi.org/10.1016/j.scitotenv.2023.165966
26. K. E. Runnalls and T. R. Oke, Phys. Geogr. **21**, 283 (2000).
https://doi.org/10.1080/02723646.2000.10642711
27. R. Bassett, X. Cai, L. Chapman, C. Heaviside, J. E. Thornes, C. L. Muller, D. T. Young, and E. L. Warren, Q. J. R. Meteorol. Soc. **142**, 2434 (2016).
https://doi.org/10.1002/qj.2836
28. A. Cosgrove and M. Berkelhammer, npj Clim. Atmos. Sci. **1**, 46 (2018).
https://doi.org/10.1038/s41612-018-0055-3
29. N. Kljun, P. Calanca, M. W. Rotach, and H. P. Schmid, Geosci. Model. Dev. **8**, 3695 (2015).
https://doi.org/10.5194/gmd-8-3695-2015
30. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, in *Advances in Neural Information Processing Systems, Montreal, 2018*, Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, 2018), Vol. 31, p. 6638.
https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

AI tools may have been used in the translation or editing of this article.