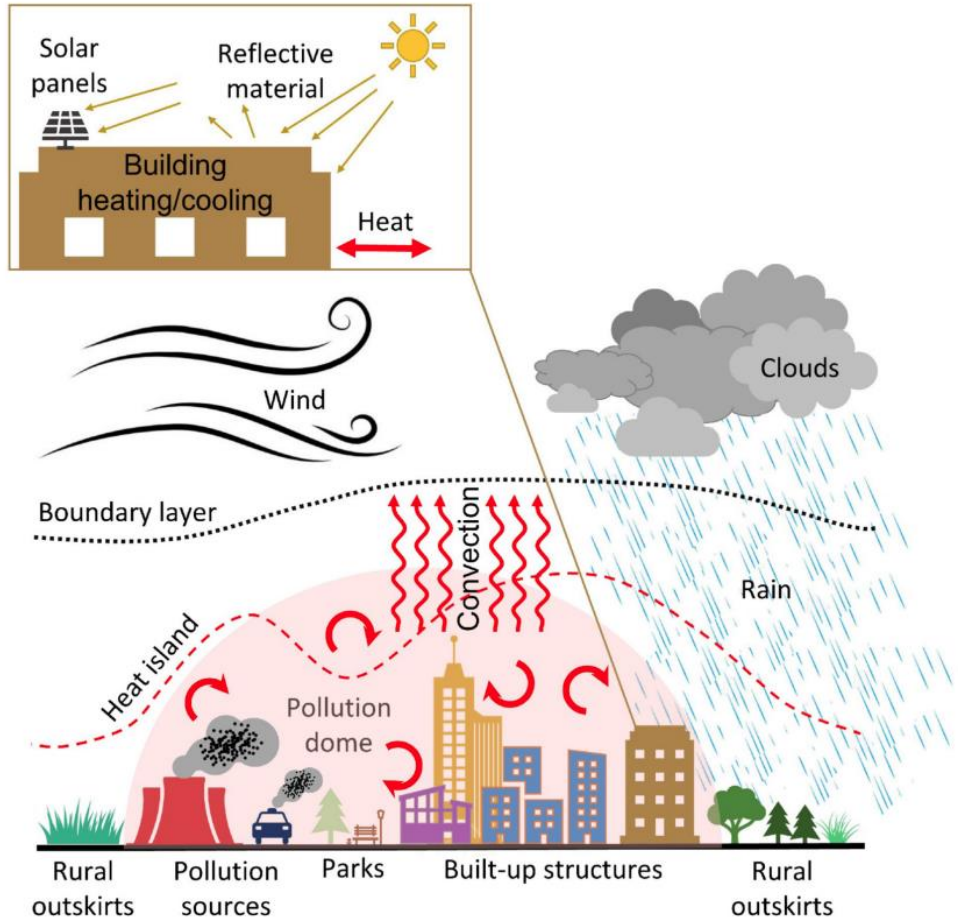
Project website

Integration and Execution of Community Land Model Urban (CLMU) in a Containerized Environment

Junjie Yu[a], Yuan Sun[a], Sarah Lindley[b], Caroline Jay[c], David O. Topping[a], Keith W. Oleson[d], Zhonghua Zheng[a]
a Department of Earth and Environmental Sciences, The University of Manchester, Manchester, M13 9PL, UK
b Department of Geography, The University of Manchester, Manchester, M13 9PL, UK
c Department of Computer Science, The University of Manchester, Manchester, M13 9PL, UK
d Climate and Global Dynamics Laboratory, NSF National Center for Atmospheric Research (NCAR), Boulder, CO 80307, USA

Urbanization Changes Local Climates

Temperature	Heavy rain	Dryness
Light rain	Sunlight	Wind



Data source:
Nat Cities 1, 686–694 (2024).
https://doi.org/10.1038/s44284-024-00120-x

Source: Y. Qian, T. C. Chakraborty, J. Li, et al. (2022).
https://doi.org/10.1007/s00376-021-1371-9

Urban Heat Mitigation

Strategies

(a) White infrastructure



(b) Green infrastructure



(c) Blue infrastructure

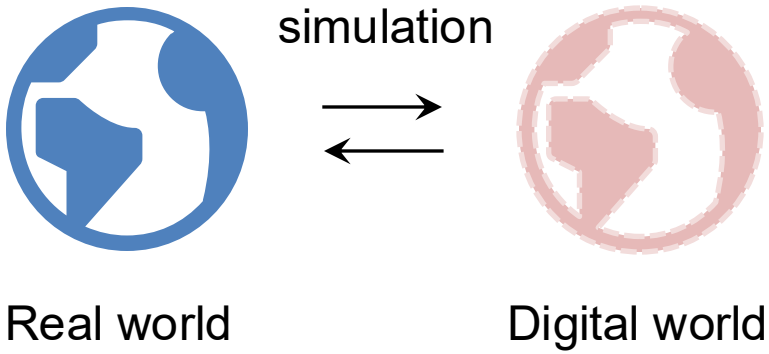


Wang Z. (2021).
https://doi.org/10.1016/j.scs.2021.103284

Strategies?

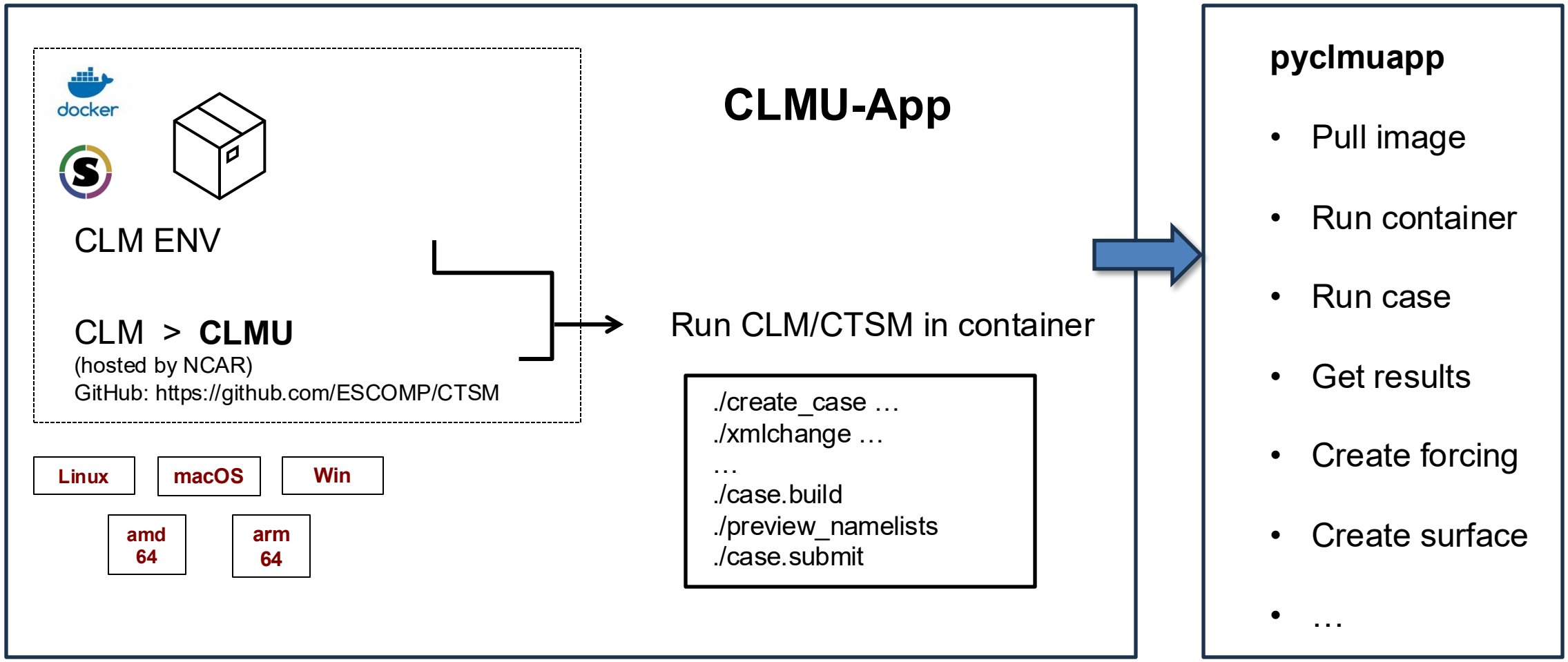
How to evaluate?

CLMU: Community Land Model Urban



1. **Software/Operating System**
2. **Case configuration**
3. **Input data limitation**

A Universal Tool for Urban Climate Modeling



OS-independent **Personal computer** **Python interface**

A Universal Tool for Urban Climate Modeling

Python: workflow

1 Initialize

```
from pyclmuapp import usp_clmu
usp = usp_clmu()
```

2 Configuration and run

```
usp_res = usp.run(
        case_name = "usp",
        SURF= ”surfdata.nc",
        FORCING = ”forcing.nc",
        RUN_STARTDATE = "2012-01-01",
        STOP_OPTION = "nyears", STOP_N = "2", ...)
# usp_res will return a list of result files location.
```

A Universal Tool for Urban Climate Modeling

- **Generating forcing for model input from ERA5**

```
from pyclmuapp import get_forcing

get_forcing( lat=lat, lon=lon, zbot=zbot, start_year=start_year, end_year=end_year,
                start_month=start_month, end_month=end_month
               )
# return a file like: era5_forcing_51.5_0.12_30_2012_1_2012_2.nc
```

- **Generating surface data for model input**

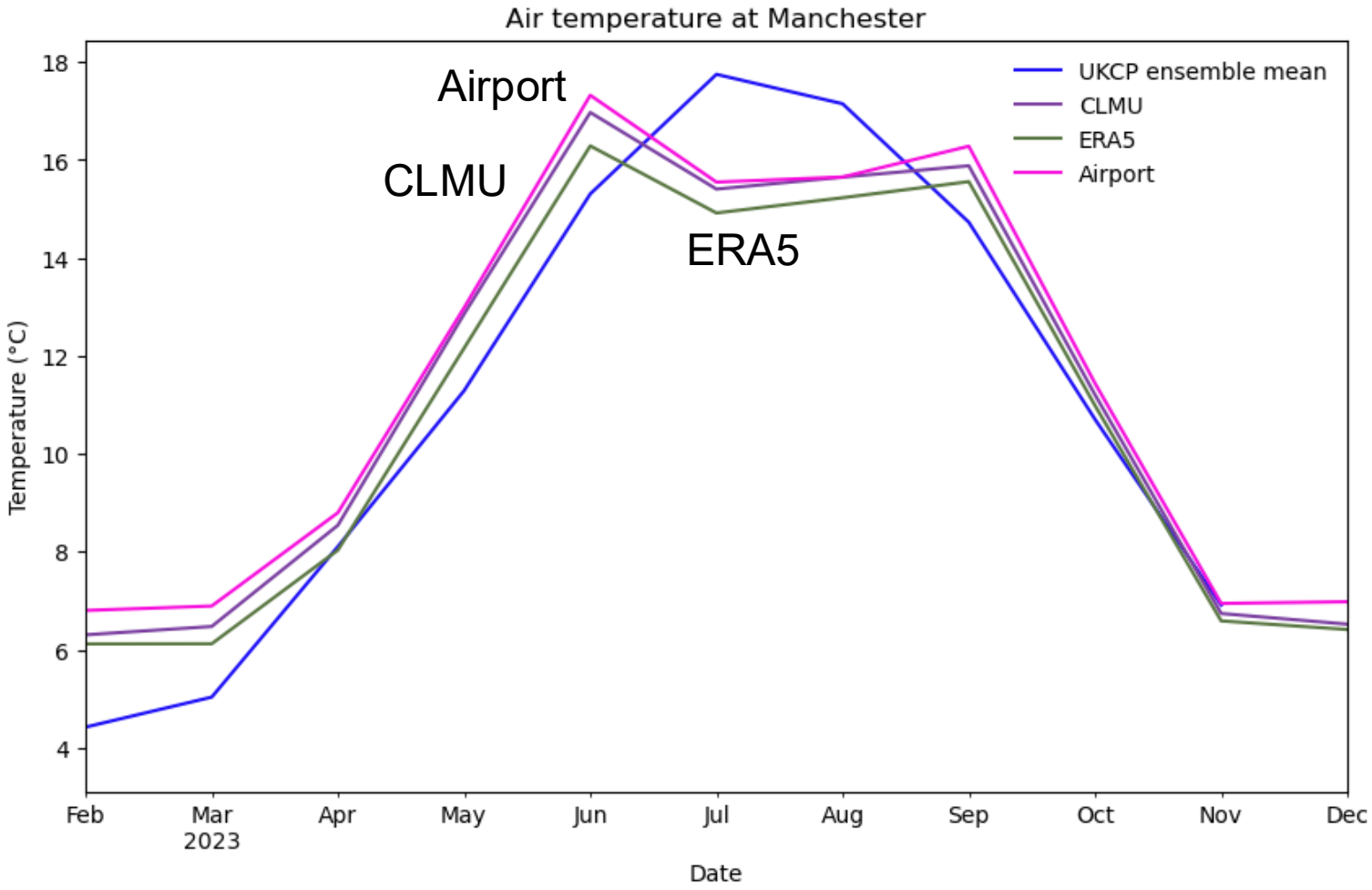
```
from pyclmuapp import get_urban_params

urban = get_urban_params(
            lat = 51.5116, lon = -0.1167,
            urban_ds='data/mksrf_urban_0.05x0.05_simyr2000.c170724.nc',
            soil_ds='data/mksrf_soitex.10level.c010119.nc',
            outputname='data/surfdata_london.nc'
            )
```

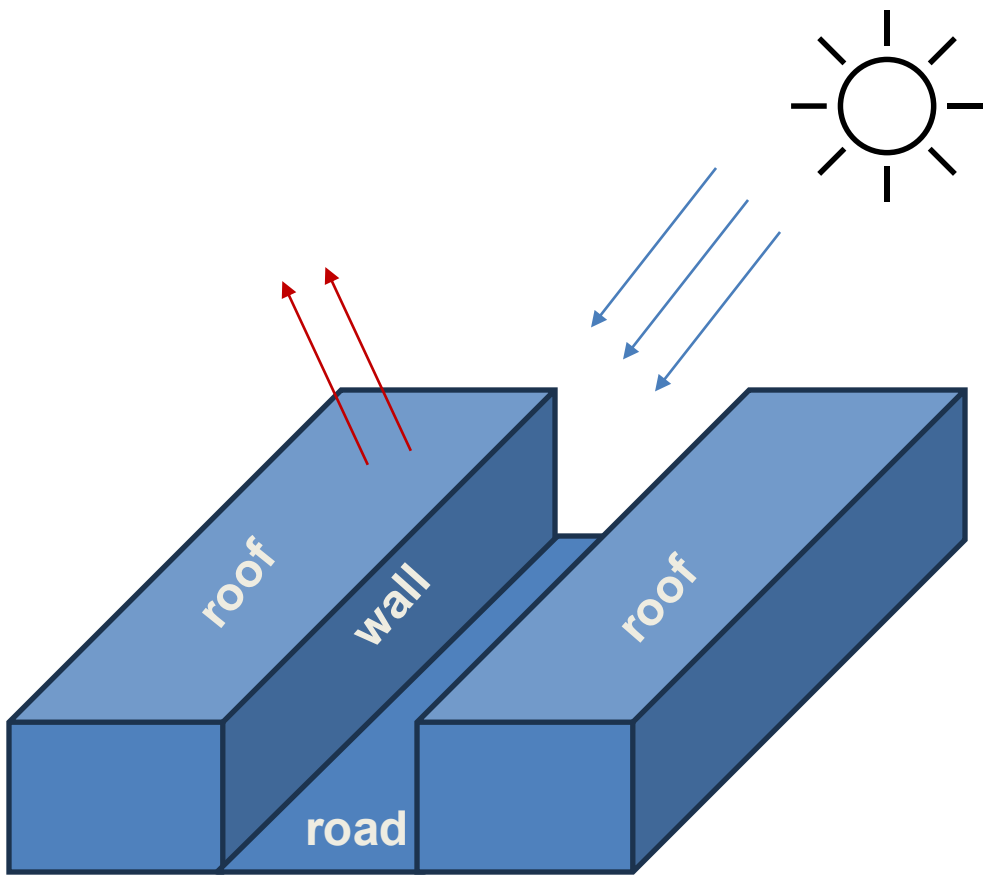
- **Modify the surface file**

```
usp.modify_surf(action={"ALB_ROOF_DIR":0.2}, mode="add")
```

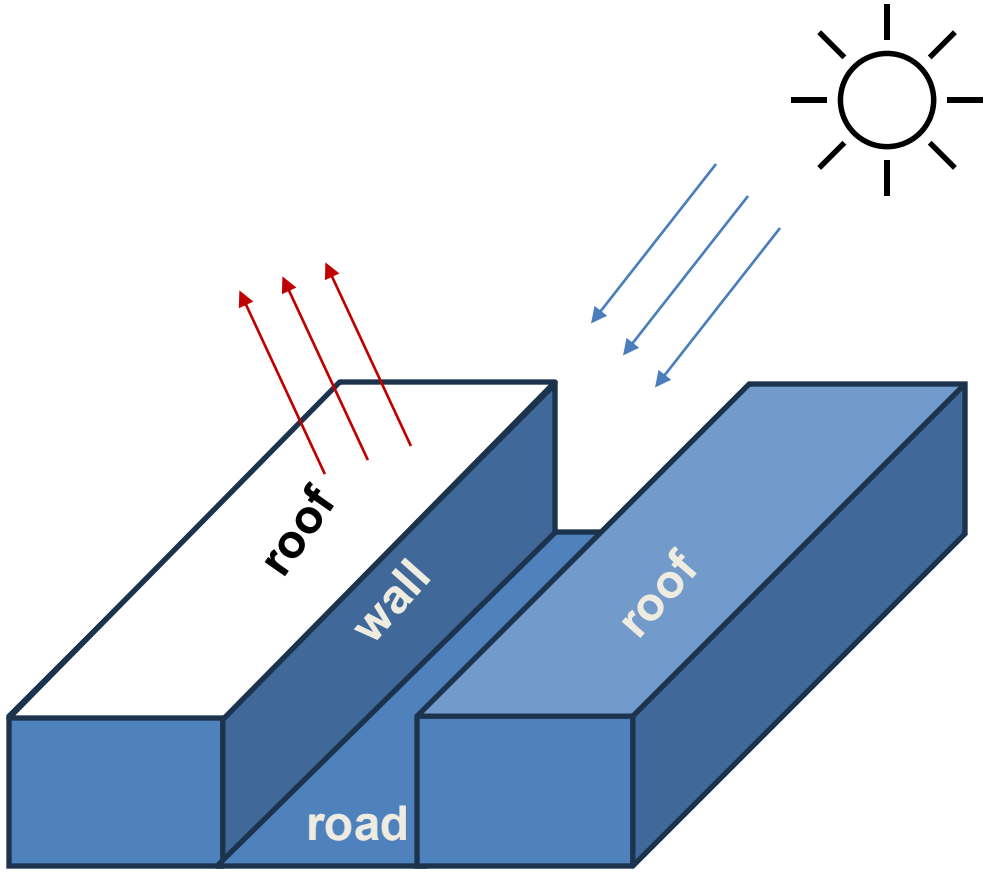
CLMU Provides Better Urban Temperature Estimates



Urban Climate Adaptation of CLMU



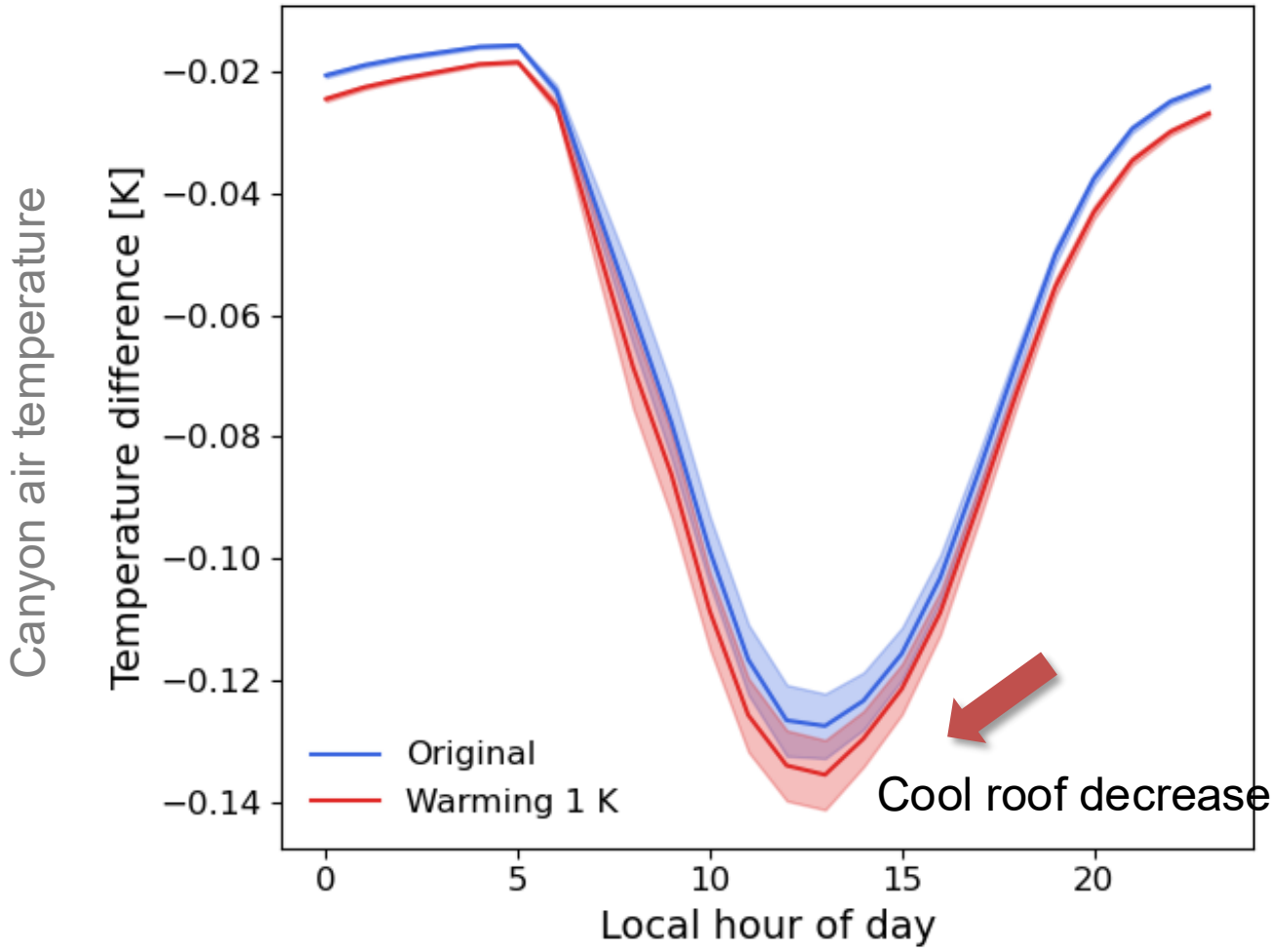
Urban canyon



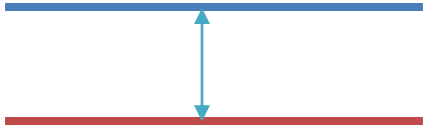
Adaptation

Cool Roofs Provide Greater Benefits at Noon and Under Warming Scenario

Temperature difference = cool roof experiment - default



Cool roof decrease more heat in noon



Cool roof benefit more under global warming

cool roof: roof albedo added by 0.2

Potential engineering adaptations

New materials

- Albedo
- Heat capacity
- Conductivity
- Solar panel
- …

Blue infrastructures

- Water irrigation
- Road / **roof sprinkling**
- Water misting
- …

Green infrastructures

- Trees
- Grass
- Green roofs
- Park
- …

Anthropogenic heat control

- **Heating**
- **Air conditioning**
- Transportation
- …

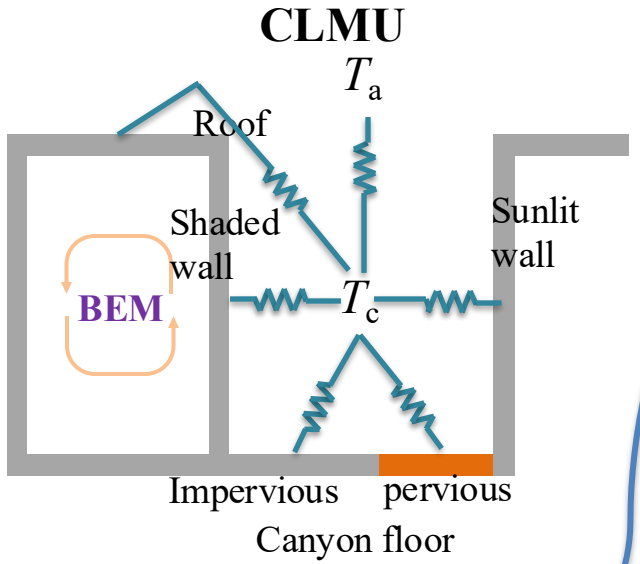
Making a better urban environment

Large Simulations for Optimization and Sensitivity Test

a

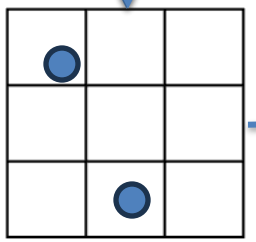
Parameters:

1. Height
2. Canyon_HWR
3. Roof fraction
4. Pervious road fraction
5. Albedo (different facet)
6. Emissivity (different facet)



Inputs ⟷ F ⟷ Climate response

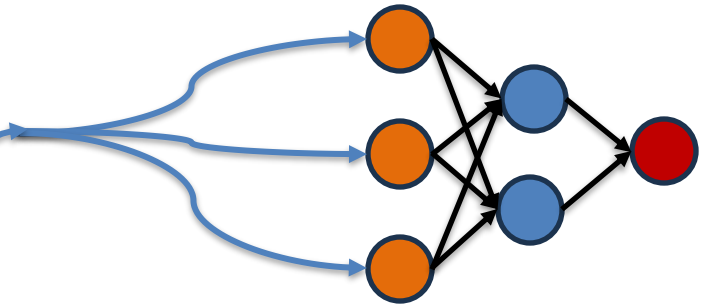
b



LHS Sampling

Easily parallelize simulations with different parameters using Python

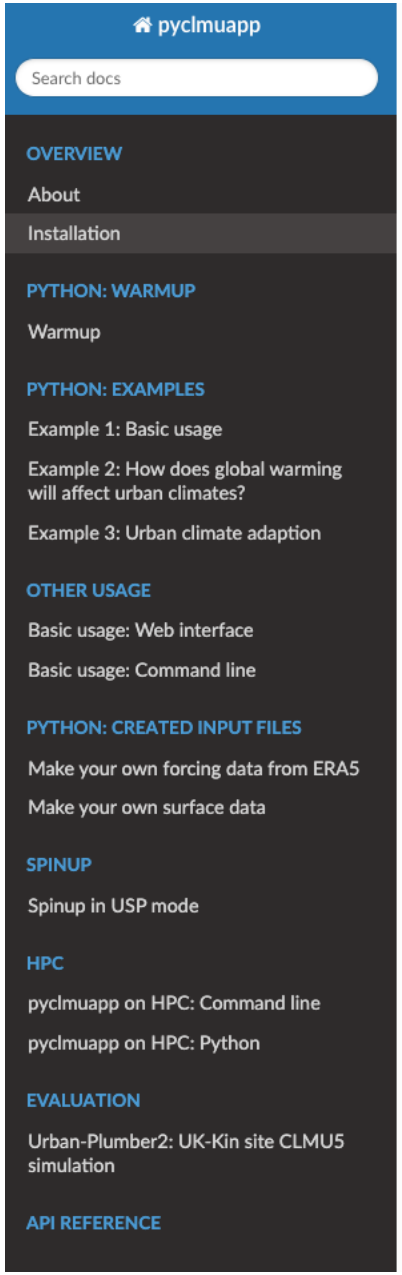
pyclmuapp

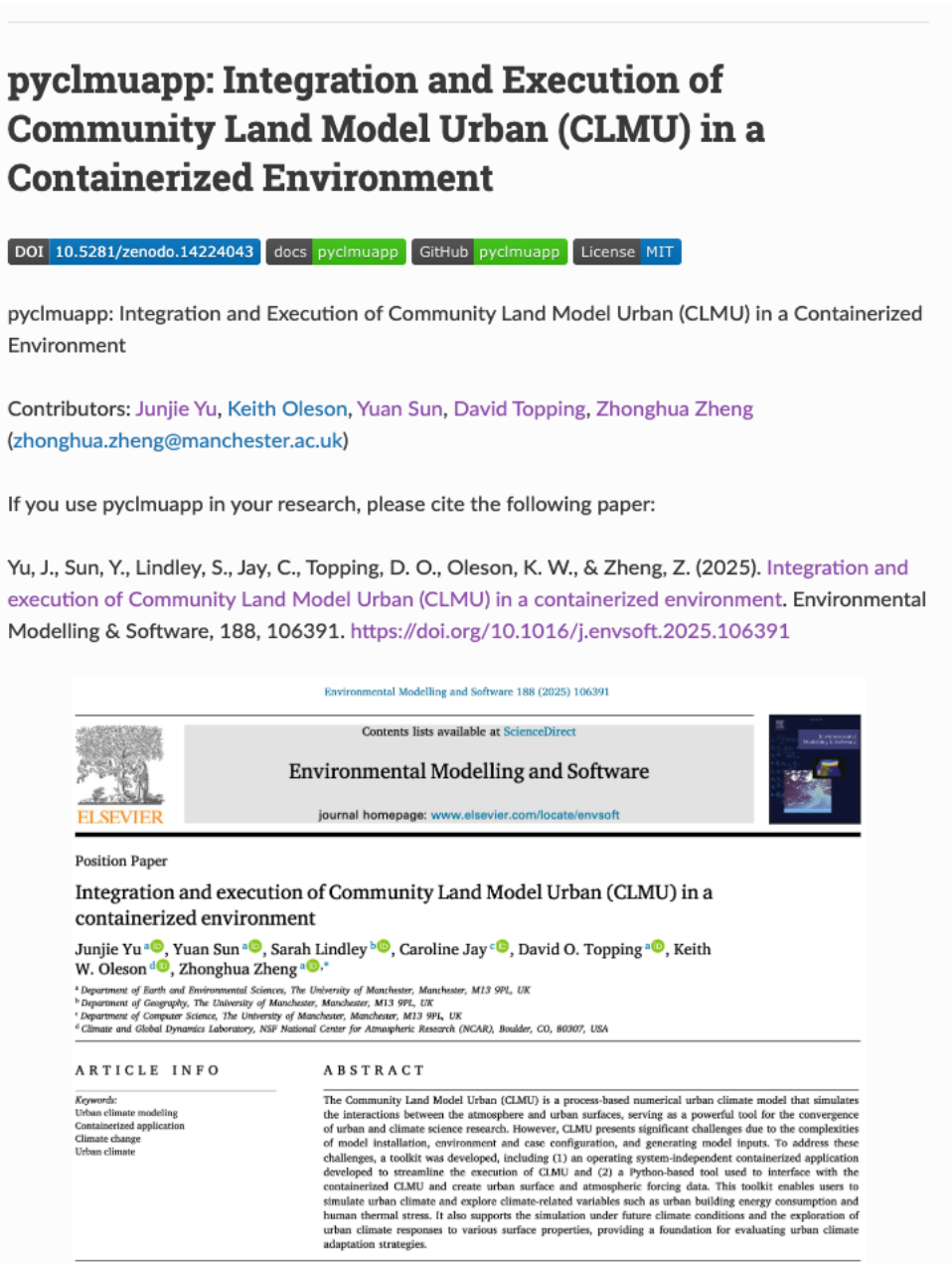


F' Surrogate Modeling

- Optimizing urban adaptation planning
- Test urban surface parameter sensitivity

pyclmuapp: tutorials and paper





pyclmuapp: Integration and Execution of Community Land Model Urban (CLMU) in a Containerized Environment

DOI 10.5281/zenodo.14224043 | docs pyclmuapp | GitHub pyclmuapp | License MIT

pyclmuapp: Integration and Execution of Community Land Model Urban (CLMU) in a Containerized Environment

Contributors: Junjie Yu, Keith Oleson, Yuan Sun, David Topping, Zhonghua Zheng (zhonghua.zheng@manchester.ac.uk)

If you use pyclmuapp in your research, please cite the following paper:

Yu, J., Sun, Y., Lindley, S., Jay, C., Topping, D. O., Oleson, K. W., & Zheng, Z. (2025). Integration and execution of Community Land Model Urban (CLMU) in a containerized environment. Environmental Modelling & Software, 188, 106391. https://doi.org/10.1016/j.envsoft.2025.106391

Environmental Modelling and Software 188 (2025) 106391

Contents lists available at ScienceDirect

Environmental Modelling and Software

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

Position Paper

Integration and execution of Community Land Model Urban (CLMU) in a containerized environment

Junjie Yu [a], Yuan Sun [a], Sarah Lindley [b], Caroline Jay [c], David O. Topping [a], Keith W. Oleson [d], Zhonghua Zheng [a,*]

[a] Department of Earth and Environmental Sciences, The University of Manchester, Manchester, M13 9PL, UK
[b] Department of Geography, The University of Manchester, Manchester, M13 9PL, UK
[c] Department of Computer Science, The University of Manchester, Manchester, M13 9PL, UK
[d] Climate and Global Dynamics Laboratory, NSF National Center for Atmospheric Research (NCAR), Boulder, CO, 80307, USA

ARTICLE INFO

Keywords:
Urban climate modeling
Containerized application
Climate change
Urban climate

ABSTRACT

The Community Land Model Urban (CLMU) is a process-based numerical urban climate model that simulates the interactions between the atmosphere and urban surfaces, serving as a powerful tool for the convergence of urban and climate science research. However, CLMU presents significant challenges due to the complexities of model installation, environment and case configuration, and generating model inputs. To address these challenges, a toolkit was developed, including (1) an operating system-independent containerized application developed to streamline the execution of CLMU and (2) a Python-based tool used to interface with the containerized CLMU and create urban surface and atmospheric forcing data. This toolkit enables users to simulate urban climate and explore climate-related variables such as urban building energy consumption and human thermal stress. It also supports the simulation under future climate conditions and the exploration of urban climate responses to various surface properties, providing a foundation for evaluating urban climate adaptation strategies.