UrbanM2M User Manual V1.0

1 Software introduction

1 Software introduction

Urban expansion is a direct result of urbanization. Urban expansion has multiple effects, including ecological, economic and social effects, and has attracted attention in the fields of urban planning and ecology. The simulation of future urban expansion scenarios can help professionals to understand how urban expansion may occur in a region in the future, and thus promote the understanding of the future urban development and ecological protection of the region. Therefore, modeling future urban expansion scenarios is very important.

Cellular automata (CA) can well simulate future urban expansion scenarios. Previous CA models mostly simulate urban expansion scenarios in a pixel-by-pixel manner, and only utilize the urban maps of one or two years for simulation. This kind of simulation method lacks the ability to integrate spatial dynamics and temporal dependency for better simulation.

In this regard, we have developed the UrbanM2M software for more reliable simulation of future urban land expansion scenarios, which is based on the ConvLSTM (Convolutional Long Short-Term Memory) deep learning model, and uses time-series urban land tiles and spatial variables to simulate future urban land expansion.

UrbanM2M integrates data processing modules such as data normalization, accuracy assessment, etc. The front-end Web UI interface is based on Gradio, which is easy for users to get started. UrbanM2M also integrates data processing modules such as data normalization and accuracy evaluation, etc.

Except for using the Web UI for simulation, it is also possible to directly use the codes in our Github repository for model implementation. It is much more flexible, but rather hard to get started if the users are not experienced in deep learning.

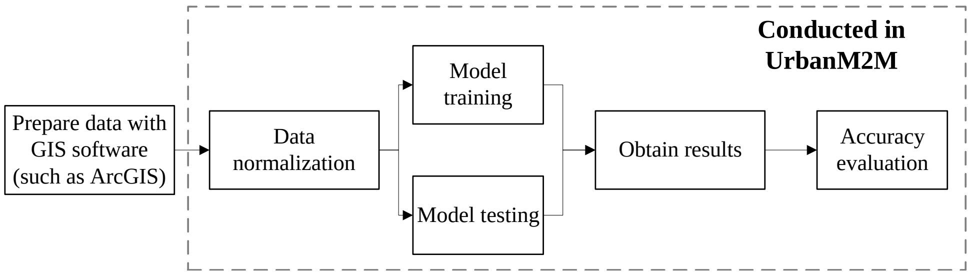


Figure 1-1 UrbanM2M simulation process

2 Software operating environment

2.1 hardware environment

UrbanM2M has high computer hardware requirements as follows:

software	minimum configuration	Recommended Configurations
Memory capacity	>=8GB	>=16GB
Remaining Hard Disk Capacity	Software ontology ≈ 7GB + data	>=20GB
GPU	NVIDIA, video memory >= 4GB. Driver version >=452.39 (AMD NOT SUPPORTED)	NVIDIA graphics card with >= 8GB of video memory. Driver version >= 452.39 (AMD NOT SUPPORTED)

2.2 System support

UrbanM2M supports running perfectly on Windows (including core algorithm package, Web GUI interface); the core algorithm can run perfectly on Linux, but the Web GUI interface doesn't support some functions (file and folder selection function) on Linux. It has not been tested on Mac system, so there is no guarantee that it can run successfully.

2.3 UrbanM2M ddownload

Download the project zip archive from Github or clone the UrbanM2M project

2.4 Preparation of the operating environment

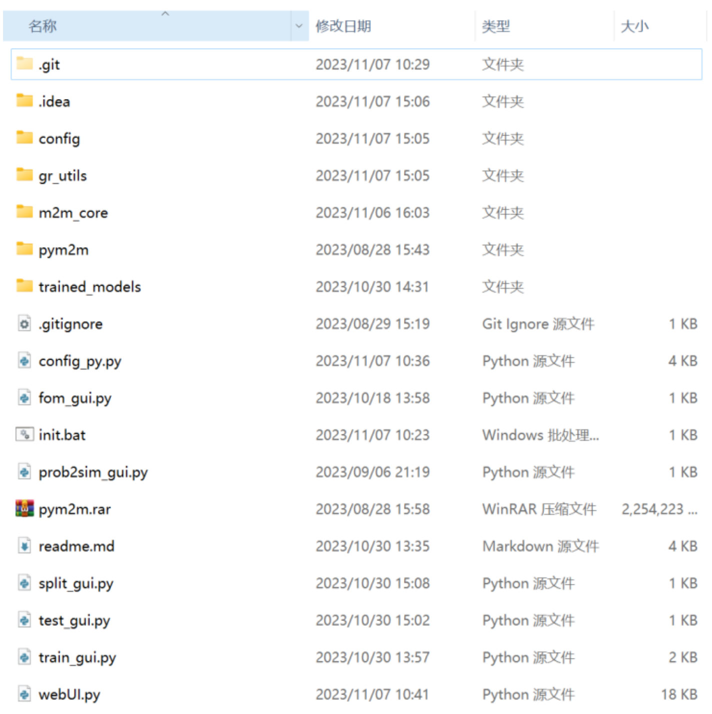
The environment required to run UrbanM2M is:

Python==3.10.x
PyTorch==1.10.0
GDAL==3.x
CUDA==11.3
cuDNN=8.2.1
gradio==3.x

If the user's device has a Python environment that meets the requirements and the CUDA/cuDNN version meets the requirements, the following steps can be skipped. Otherwise, it is recommended to download the environment zip.

(Link: https://pan.baidu.com/s/1YzYKjb3hXWjbVtwn0Vpesw?pwd=0td9)

Extract pym2m.tar.gz to Urbanm2m. The unpacked folder structure is as follows.



名称	修改日期	类型	大小
.git	2023/11/07 10:29	文件夹	
.idea	2023/11/07 15:06	文件夹	
config	2023/11/07 15:05	文件夹	
gr_utils	2023/11/07 15:05	文件夹	
m2m_core	2023/11/06 16:03	文件夹	
pym2m	2023/08/28 15:43	文件夹	
trained_models	2023/10/30 14:31	文件夹	
.gitignore	2023/08/29 15:19	Git Ignore 源文件	1 KB
config_py.py	2023/11/07 10:36	Python 源文件	4 KB
fom_gui.py	2023/10/18 13:58	Python 源文件	1 KB
init.bat	2023/11/07 10:23	Windows 批处理...	1 KB
prob2sim_gui.py	2023/09/06 21:19	Python 源文件	1 KB
pym2m.rar	2023/08/28 15:58	WinRAR 压缩文件	2,254,223 ...
readme.md	2023/10/30 13:35	Markdown 源文件	4 KB
split_gui.py	2023/10/30 15:08	Python 源文件	1 KB
test_gui.py	2023/10/30 15:02	Python 源文件	1 KB
train_gui.py	2023/10/30 13:57	Python 源文件	2 KB
webUI.py	2023/11/07 10:41	Python 源文件	18 KB

Figure 2-1 Urbanm2m folder structure

3 Data preparation

3.1 Introduction

Unlike most CA models, UrbanM2M uses **time-series** urban land rasters and spatial variables to predict future urban land expansion scenarios, which requires the user to prepare urban land rasters for consecutive years. Also, UrbanM2M predicts transition probabilities in terms of **raster tiles** (rather than image elements, as used in most models). This requires the user to prepare urban land rasters, spatial variables, etc. slightly larger than the study area to ensure the completeness of the raster tiles in the edge areas of the study area. Therefore, UrbanM2M requires high spatial consistency of data.

Chapters 3 and 4 of this document will introduce **the process of data preparation and the process of using UrbanM2M** by taking the simulation of urban sprawl in the Taihu Lake Basin of the Yangtze River Delta as an example. Among them, this chapter introduces the process of preparing UrbanM2M input data in GIS software.

Users can also download prepared example data:

https://pan.baidu.com/s/17Rj-qi28VVWd3HTqhqwheQ?pwd=81g9

If the users want to use their own data, it is recommended for them to use open-source time-series urban land dataset (or land use dataset) such as GISA, GAIA as the input urban land data.

3.2 Basic data requirements

（1） All data coordinate systems are consistent.

（2） It is highly recommended that all data be converted to a projected coordinate system (such as WGS 1984 Web Mercator).

（3） The number of rows and columns is the same for all raster data.

3.3 make ready

Before you start, you need to create a folder to store the data. And create two empty folders, vars and year. once created, the folder structure is as follows.

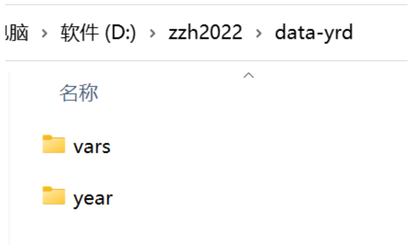


Figure 3-1 Data root directory structure preparation

In the following steps, we will use the Yangtze River Delta as an example.

3.4 Create study area buffer vector

In this step, the user needs to prepare the following data:

Table 1 Creation of study area buffer vectors - data preparation

Data	Note
Study area vector file	/

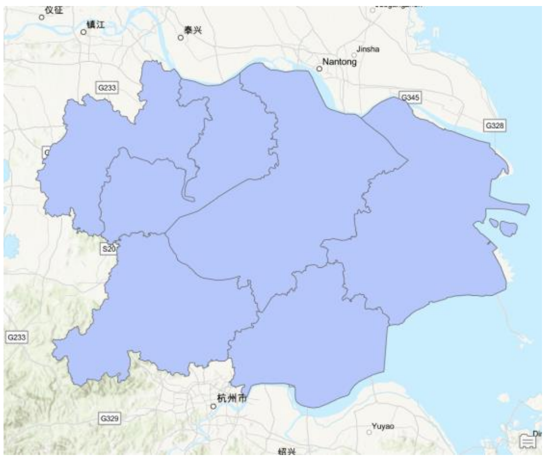


Figure 3-2 Vector file of the study area

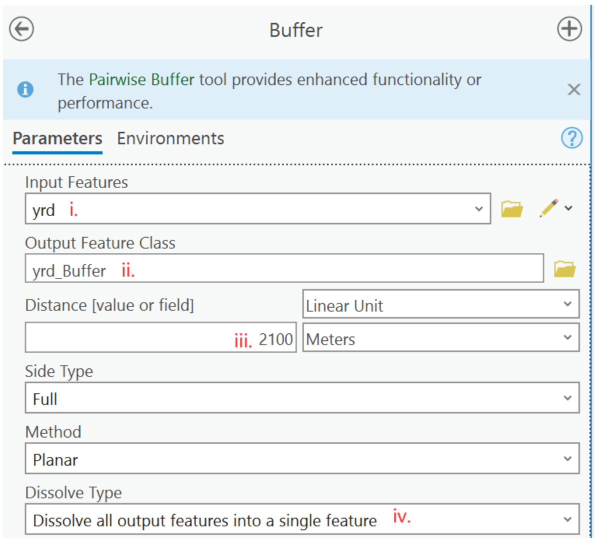
Buffered study area vectors (hereafter referred to as **buffer vectors**) are created using the [Buffer] tool of GIS software. Buffer vectors are used to extract a circle of land larger than the study area, variable rasters, etc. In most GIS software, the parameters of the buffer are as follows:

i. **Input data.** Please make a determination based on the user's data.

ii. **Output data.** Can be determined at your discretion.

iii. **Buffer radius units and values.** Please ensure that the units are in meters (Meters). Please ensure that the buffer radius is at least 70 times the resolution of the urban land raster that the user will be using. For example, when using a 30-meter resolution raster, set it to at least 2100 meters; when using a 100-meter resolution raster, set it to at least 7000 meters.

iv. **Fusion Type.** Select Fusion as an element.


Buffer

The Pairwise Buffer tool provides enhanced functionality or performance.

Input Features: yrd i.

Output Feature Class: yrd_Buffer ii.

Distance [value or field]: iii. 2100

Linear Unit: Meters

Side Type: Full

Method: Planar

Dissolve Type: Dissolve all output features into a single feature iv.

Figure 3-3 Buffer creation parameters

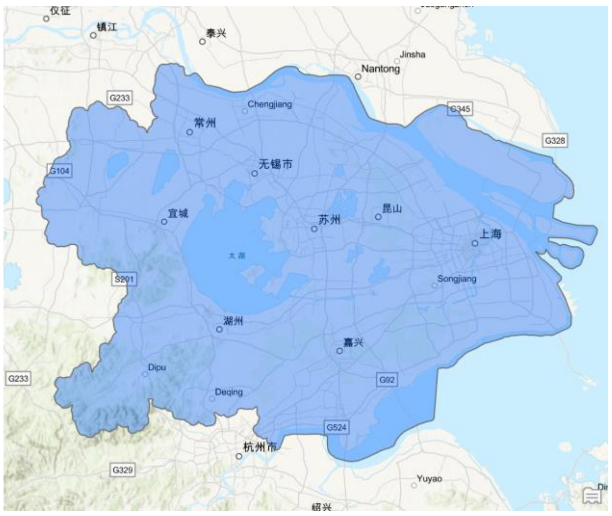


Figure 3-4 Buffer creation results

3.5 Preparing the urban land data raster

The data that the user needs to use for this step is as follows:

Table 2 Preparing the Land Raster - Data Preparation

Data	Note
Time-series urban land use raster (larger scale)	/
The buffer vector obtained in 3.4	/

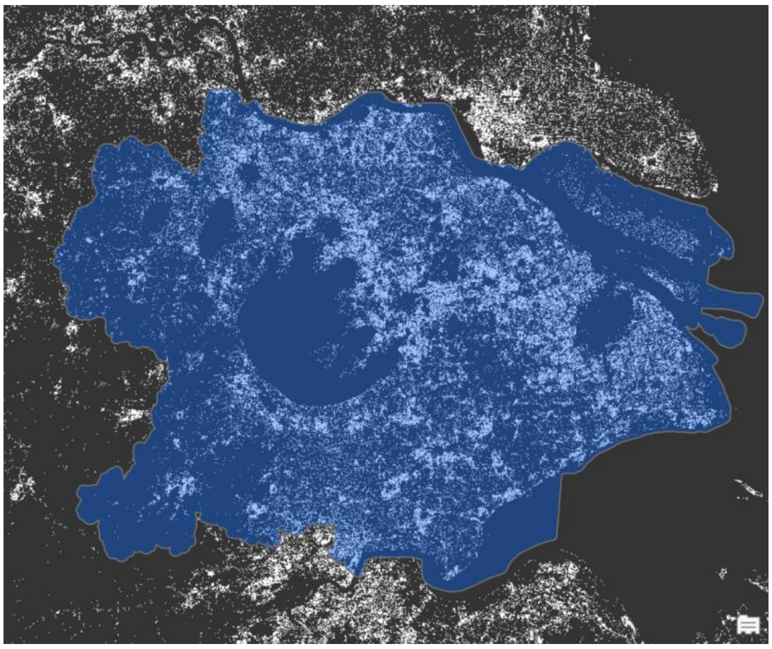


Figure 3-5 Schematic representation of the urban land raster in the study area

Before starting this step, make sure that the extent of the user's land raster is at least as large as the buffer vector obtained in the previous step (**not as large as the study area**).

This step uses the buffer vector to get a raster of urban land with multiple time series within the buffer. The user needs to repeat the following steps several times.

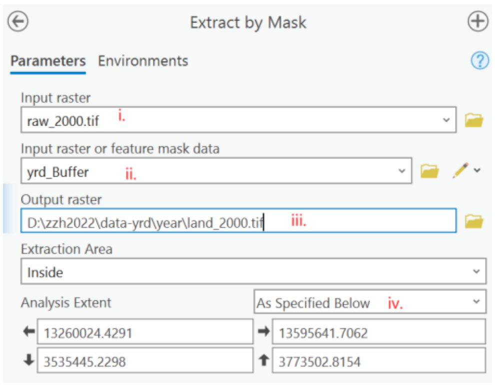
Use the [Extract by Mask] Tool and set the following parameters to extract the urban land raster for each year one by one. (This can also be batch processed using methods such as ArcPy)

i. **Input raster**: urban land raster for the current year

ii. **Mask**: buffer vector.

iii. **Output**: must be saved in the year folder of the data root directory and must be named land_{year}.tif. e.g. name the vector for 2011 as land_2011.tif. **If it is not named correctly, the subsequent programs will not recognize the content.**

iv. **Processing range**: selected as buffer vector.



Extract by Mask

Parameters Environments

Input raster: raw_2000.tif i.

Input raster or feature mask data: yrd_Buffer ii.

Output raster: D:\zzh2022\data-yrd\year\land_2000.tif iii.

Extraction Area: Inside

Analysis Extent: As Specified Below iv.

13260024.4291	13595641.7062
3535445.2298	3773502.8154

Figure 3-6 Preparing the urban land raster-parameter settings

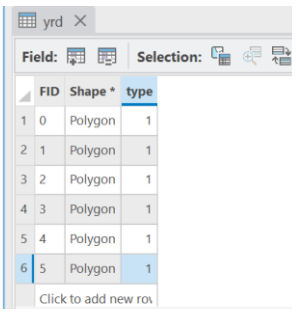
3.6 Prepare study area extent raster named range.tif

The data to be prepared for this step is as follows:

Table 3 Preparation of Study Area Extent Raster - Data Preparation

Data	Note
Urban land raster in the study area for any given year	As a template grid
Study area vector	/
Buffer vector	/

(1) Add a long-integer field to the study area vector, using the field calculator and setting its value to 1.

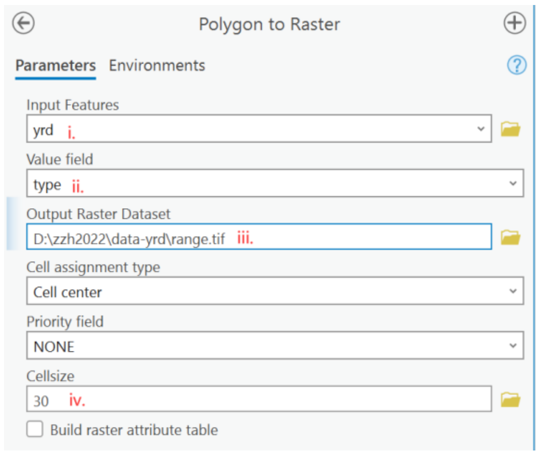


	FID	Shape *	type
1	0	Polygon	1
2	1	Polygon	1
3	2	Polygon	1
4	3	Polygon	1
5	4	Polygon	1
6	5	Polygon	1

Figure 3-7 Adding fields to the study area vector

(2) Use the [Polygon to Raster] tool to convert the study area vector to a raster with value 1. The parameters and environment settings are as follows:

i. **Input**: study area vector (not buffer vector).
ii. **Value field**: the field that was set to 1 in the previous sub-step.
iii. Output: **must be saved in the root directory of the data folder created in 3.3, naming must be range.tif.**
iv. Raster size: consistent with the user's site raster.
v. Environmental Setting - Coordinate System: consistent with the user's land use raster.
vi. Environmental Setting - Extent: set to any year's urban land raster for the study area.
vii. Environmental Setting – Snap Raster: the same setting to Processing Extent.
viii.


Polygon to Raster

Parameters Environments

Input Features: yrd i.

Value field: type ii.

Output Raster Dataset: D:\zzh2022\data-yrd\range.tif iii.

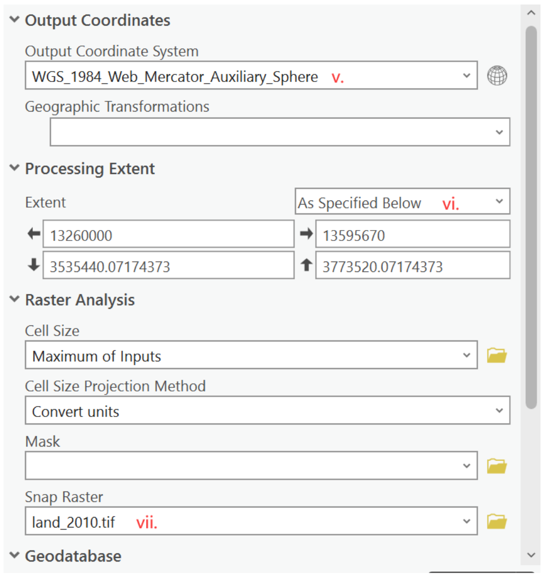
Cell assignment type: Cell center

Priority field: NONE

Cellsize: 30 iv.

☐ Build raster attribute table

Figure 3-8 Preparing the study area range raster - parameter settings


Output Coordinates

Output Coordinate System: WGS_1984_Web_Mercator_Auxiliary_Sphere v.

Geographic Transformations:

Processing Extent

Extent: As Specified Below vi.

← 13260000 → 13595670

↓ 3535440.07174373 ↑ 3773520.07174373

Raster Analysis

Cell Size: Maximum of Inputs

Cell Size Projection Method: Convert units

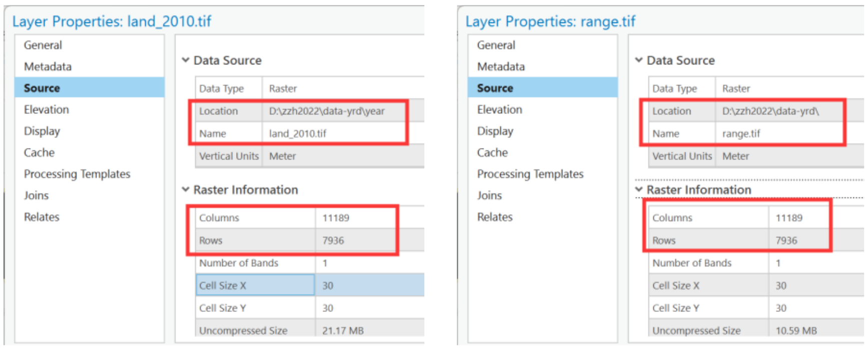
Mask:

Snap Raster: land_2010.tif vii.

Geodatabase

Figure 3-9 Preparing the study area range raster - Environmental Settings

If created correctly, the urban land raster created in 3.5 and the range raster created in this step should have the same number of rows and columns.



Layer Properties: land_2010.tif

Data Source

Data Type	Raster
Location	D:\zzh2022\data-yrd\year
Name	land_2010.tif
Vertical Units	Meter

Raster Information

Columns	11189
Rows	7936
Number of Bands	1
Cell Size X	30
Cell Size Y	30
Uncompressed Size	21.17 MB

Layer Properties: range.tif

Data Source

Data Type	Raster
Location	D:\zzh2022\data-yrd\
Name	range.tif
Vertical Units	Meter

Raster Information

Columns	11189
Rows	7936
Number of Bands	1
Cell Size X	30
Cell Size Y	30
Uncompressed Size	10.59 MB

Figure 3-10 Successful creation

3.7 Preparing the restriction raster restriction.tif

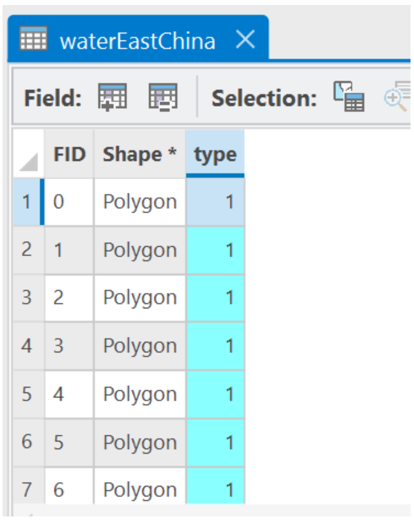
The restriction raster is used to identify areas that are not available for urban development. A value of 1 in the restriction rsater means no development, and other values mean no constraints. **If the user does not intend to constrain, this step can be skipped (recommended for users first time to use UrbanM2M). In subsequent steps, if restriction.tif is absent, the program automatically generates an all-zero raster to indicate that no constraints will be made.**

The data to be prepared for this step is as follows:

Table 4 Preparing the restriction raster- data preparation

Data	Note
Restricted area vector	e.g., water bodies, protected areas, etc.; The spatial coordinate system is the same as the urban land raster

After preparing the restricted area vector, first add a long-integer field to the vector, using the Field Calculator, and set its value to 1.



	FID	Shape *	type
1	0	Polygon	1
2	1	Polygon	1
3	2	Polygon	1
4	3	Polygon	1
5	4	Polygon	1
6	5	Polygon	1
7	6	Polygon	1

Figure 3-11 Adding a long plastic field

Subsequently use the [Polygon to Raster] tool to get a restriction raster with the following parameters:

i. **Input**: Restricted area vector.

ii. **Value field**: the field that was set to 1 in the previous sub-step.

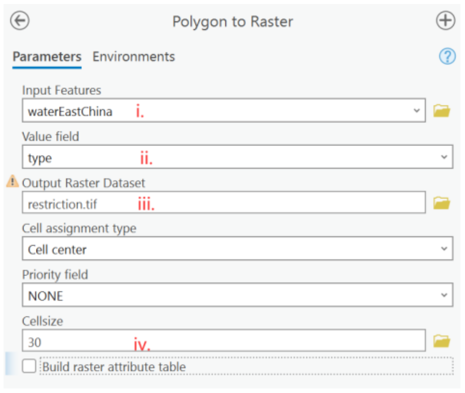
iii. **Output**: **must be saved in the data root directory created in 3.3, named must be restriction.tif.**

iv. **Raster size**: consistent with the user's urban land raster.

v. Environment Setting - Coordinate System: consistent with the user's land use raster.

vi. Environment Setting – Processing Extent: set to any year's urban land raster for the study area.

vii. Environment Setting – Snap Raster: the same setting to Processing Extent.


Polygon to Raster

Parameters Environments

Input Features: waterEastChina i.

Value field: type ii.

Output Raster Dataset: restriction.tif iii.

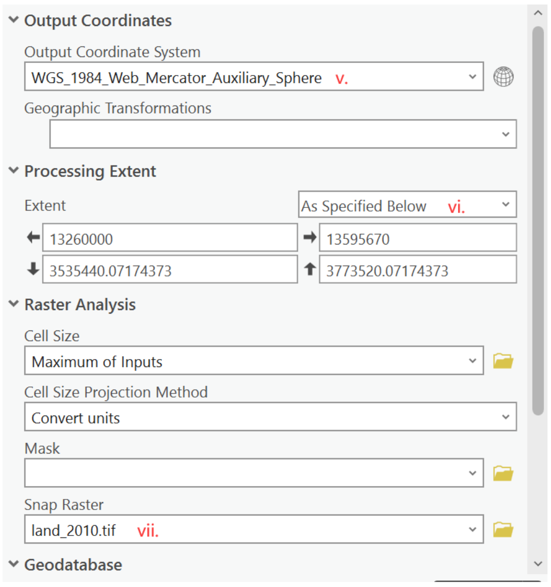
Cell assignment type: Cell center

Priority field: NONE

Cellsize: 30 iv.

Build raster attribute table

Figure 3-12 Preparing the restriction raster - parameter settings


Output Coordinates

Output Coordinate System: WGS_1984_Web_Mercator_Auxiliary_Sphere v.

Geographic Transformations:

Processing Extent

Extent: As Specified Below vi.

← 13260000 → 13595670

↓ 3535440.07174373 ↑ 3773520.07174373

Raster Analysis

Cell Size: Maximum of Inputs

Cell Size Projection Method: Convert units

Mask:

Snap Raster: land_2010.tif vii.

Geodatabase

Figure 3-13 Preparing the restriction raster - environment settings

The raster obtained after processing is schematically shown below:

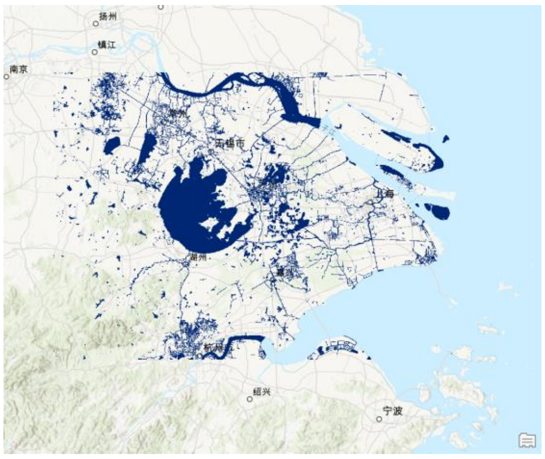


Figure 3-14 Schematic of the constraint grid

The number of rows and columns of a successfully created constraint raster should also match the site raster.

3.8 Preparation of spatial variable rasters

The data to be prepared for this step is as follows:

Table 5 Preparing the Spatial Variable Raster - Data Preparation

Data	Note
Data used to generate spatial variables	e.g. DEM, city center point
Buffer vector previously created	/

Users can choose the spatial variables according to their needs. However, if the user is going to use the trained model provided by us (Not recommend for oversea users), the spatial variables that the user needs to prepare must be: slope, distance from the center of the county (county city, district) and distance from the center of the town.

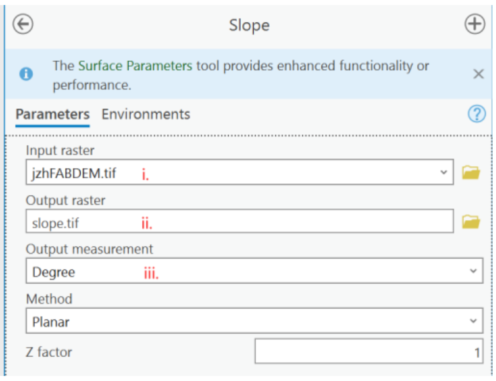
The following is an example of the process of preparing the slope and distance from the town center to illustrate the spatial variable raster preparation process.

3.8.1 Preparing the slope rasters

First prepare the DEM file. Users can download the DEM file (Such as SRTM, FABDEM) with suitable resolution from the internet.

Use the [Slope] tool in the GIS software with the following parameter settings:

i. **Input**: DEM prepared by the user.
ii. **Output**: set it freely, just don't put it in the vars folder.
iii. **Output** measurement unit: DEGREE.
iv. Environment Setting - Coordinate System: consistent with the user's land use raster.
v. Environment Setting - Processing Extent: set to any year's urban land raster for the study area.
vi. Environment Setting - Cell Size: the same setting to Processing Extent.
vii. Environment Setting - Mask: the same setting to Processing Extent.
viii. Environment Setting - Snap Raster: the same setting to Processing Extent.


Slope

The Surface Parameters tool provides enhanced functionality or performance.

Parameters Environments

Input raster: jzhFABDEM.tif i.

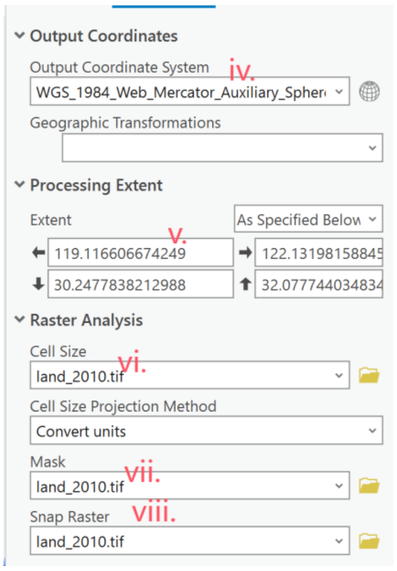
Output raster: slope.tif ii.

Output measurement: Degree iii.

Method: Planar

Z factor: 1

Figure 3-15 Preparing the Slope - Parameter Settings


Output Coordinates

Output Coordinate System iv.
WGS_1984_Web_Mercator_Auxiliary_Sphere

Geographic Transformations

Processing Extent

Extent: As Specified Below v.
119.116606674249
122.13198158845
30.2477838212988
32.077744034834

Raster Analysis

Cell Size vi.
land_2010.tif

Cell Size Projection Method
Convert units

Mask vii.
land_2010.tif

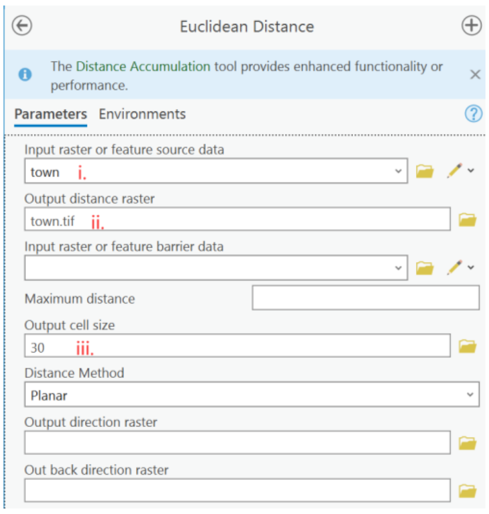
Snap Raster viii.
land_2010.tif

Figure 3-16 Preparing the Slope - Environment Setting

3.8.2 Preparation Distance to town center raster

An example of preparing a distance to town center raster is illustrated. The steps for preparing other distance rasters are similar.

First prepare the town center point vector file. Then use the [Euclidean Distance Tool] in the GIS software to calculate the distance to town center with the following parameters and environment settings:


Euclidean Distance

The Distance Accumulation tool provides enhanced functionality or performance.

Parameters Environments

Input raster or feature source data
town i.

Output distance raster
town.tif ii.

Input raster or feature barrier data

Maximum distance

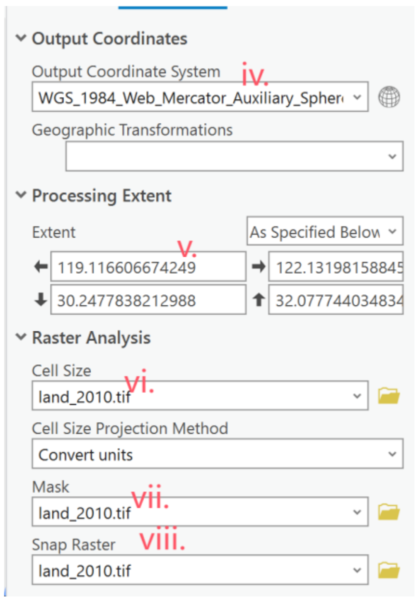
Output cell size
30 iii.

Distance Method
Planar

Output direction raster

Out back direction raster

Figure 3-17 Preparing distance variables - environment settings


Output Coordinates

Output Coordinate System iv.

WGS_1984_Web_Mercator_Auxiliary_Spher

Geographic Transformations

Processing Extent

Extent: As Specified Below

v.

119.116606674249 | 122.13198158845

30.2477838212988 | 32.077744034834

Raster Analysis

Cell Size vi.

land_2010.tif

Cell Size Projection Method

Convert units

Mask vii.

land_2010.tif

Snap Raster viii.

land_2010.tif

Figure 3-18 Preparing distance variables - environment settings

i. Input: point vector prepared by the user.

ii. Output: set it freely, just don't put it in the vars folder.

iii. Output raster size: consistent with the site raster.

iv. Environment Setting - Coordinate System: consistent with the user's land use raster.

v. Environment Setting – Processing Extent: set to any year's urban land raster for the study area.

vi. Environment Setting – Cell Size: the same setting to Processing Extent.

vii. Environment Setting – Mask: the same setting to Processing Extent.

viii. Environment Setting – Snap Raster: the same setting to Processing Extent.

3.9 Data folder structure

After completing the above steps, there should be two folders vars/year and two rasters range.tif and restriction.tif (optional) in the data root directory. the year folder contains the land rasters, and the vars folder is empty (it will be used to normalize the results in the subsequent steps).

4 Launching the UrbanM2M WebUI

4.1 Use the pym2m environment:

If you choose the Python environment provided with the download (pym2m), right-click on init.bat and run it as administrator. Visit https://127.0.0.1/7860 in your browser.

4.2 Use the your environment:

If you choose to use the included Python environment:
(1) Right-click init.bat and edit.
(2) Replace line 4 of . /pym2m/python.exe to the user's python.exe path.
(3) Saving.
Right-click on init.bat and run it as administrator. Visit https://127.0.0.1/7860 in your browser.

4.3 UrbanM2M WebUI Interface introduction

UrbanM2M WebUI consists of three parts: data dir setup, main processing flow tab and auxiliary tools. After startup, Data dir must be set. After that, users can choose the functions in the process flow tab according to their needs.

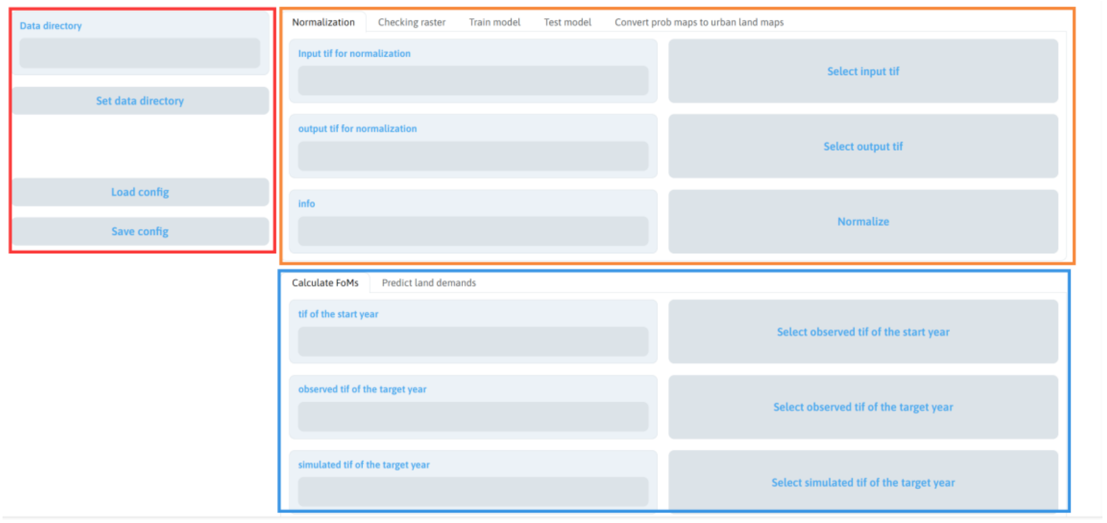

Data directory
Set data directory
Load config
Save config
Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Input tif for normalization
Select input tif
output tif for normalization
Select output tif
info
Normalize
Calculate FoMs
Predict land demands
tif of the start year
Select observed tif of the start year
observed tif of the target year
Select observed tif of the target year
simulated tif of the target year
Select simulated tif of the target year

Figure 4-1 UrbanM2M WebUI interface

5 UrbanM2M Core Process

5.1 Notes on the use of UrbanM2M

（1） Please make sure that all rasters generated in Chapter 3 have the same spatial extent, number of rows and columns, and coordinate system. Otherwise, unexpected erroneous results may occur.

(2) If cross-region model migration is desired, the extent of the training model region and the migration target region should not differ too much, or the results may not be very satisfactory.

(3) Too large urban land demand may yield more confusing results when cross-regional model migration, in case the region for training model is much larger than the target region.

5.2 Setting the data root directory

Just enter the user's data root directory in the Data directory text box (or click Set data directory to set it). If you do not set the data directory, almost all functions will report an error, so be sure to set the data directory before using other functions.

In the following sections, **$data_dir** is used to denote the data root directory.

5.3 Data Normalization (Normalization)

5.3.1 Note

Data normalization is used to ensure that the input raster meets the input requirements of ConvLSTM. The spatial variables input to ConvLSTM should be in the range 0-1.

In the [Normalization] tab, select the raster to be normalized in [input tif], and select the path to be saved in [output tif].

Please note, **make sure that the saved results are located in the $data_dir/vars folder.** This will ensure that the program reads the spatial variables correctly.

If the user wishes to use the pre-trained models located in urbanm2m/trained_models after unpacking, they should normalize the slope, distance from town center, and distance from county center variables and name them slope.tif, town.tif, and county.tif.

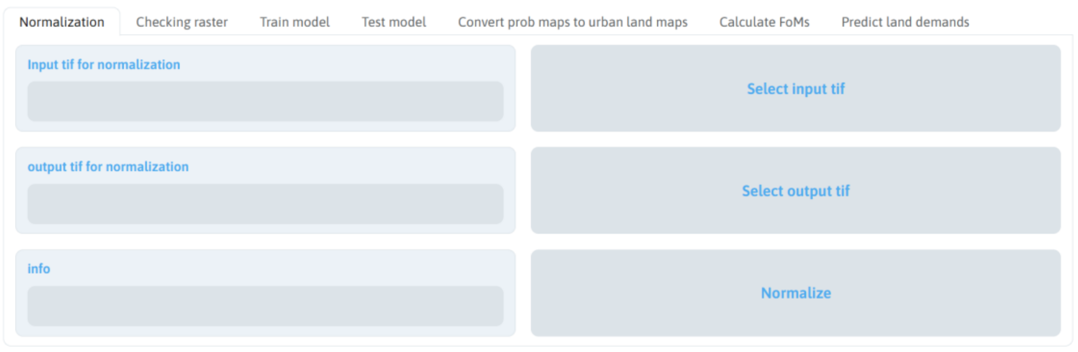
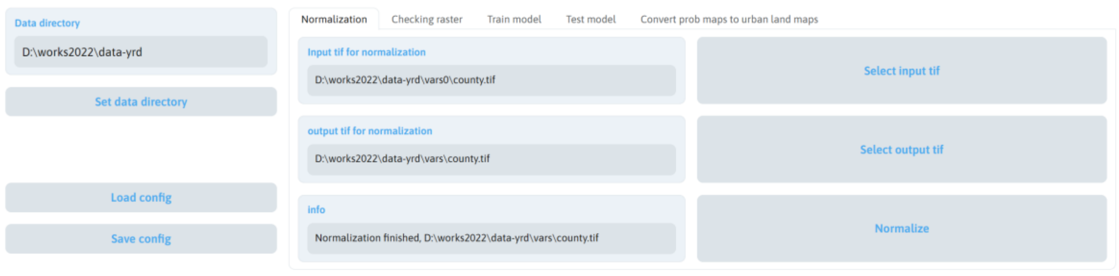


Figure 5-1 Normalization interface

5.3.2 Case

The spatial variable rasters prepared in Chapter 4 are county.tif, town.tif, slope.tif. They are normalized separately and saved under the vars path in the data folder.



Data directory: D:\works2022\data-yrd

Input tif for normalization: D:\works2022\data-yrd\vars0\county.tif

output tif for normalization: D:\works2022\data-yrd\vars\county.tif

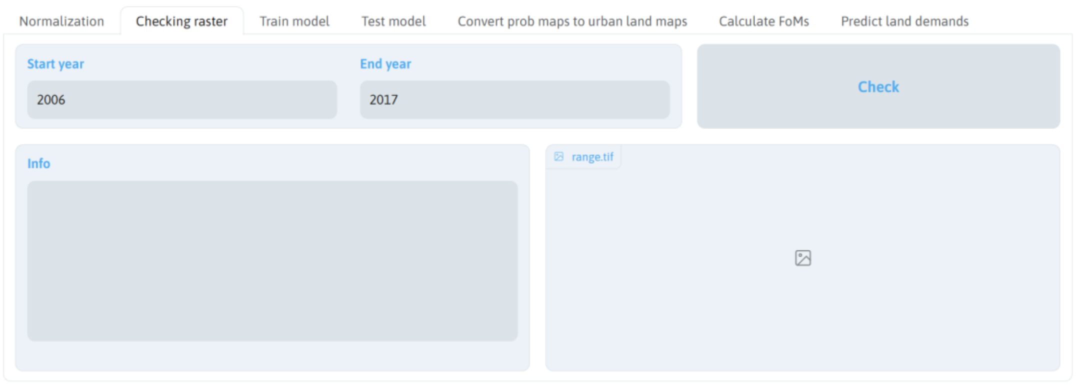
info: Normalization finished, D:\works2022\data-yrd\vars\county.tif

Figure 5-2 Normalization case

5.4 Checking raster availability

5.4.1 Note

UrbanM2M requires high spatial consistency of raster data, if inputting non-compliant rasters may get unexpected wrong results. Therefore, it is strongly recommended to use the [Checking rasters] module to check whether the raster data under $data_dir meets the requirements before training or testing the model. This module will read $data_dir and check whether urban land rasters between [Start year] and [End year] and spatial variable rasters meet the requirements. After checking (without error), a binary image of the study area is displayed in the image box of range.tif.



Start year: 2006

End year: 2017

Figure 5-3 Check Raster Availability module interface

The module checks for the following:

（1） Does the range raster (range.tif) exist, is there at least one spatial variable raster in the vars folder, and if the years folder contains all land use rasters between the [Start year] and [End year].

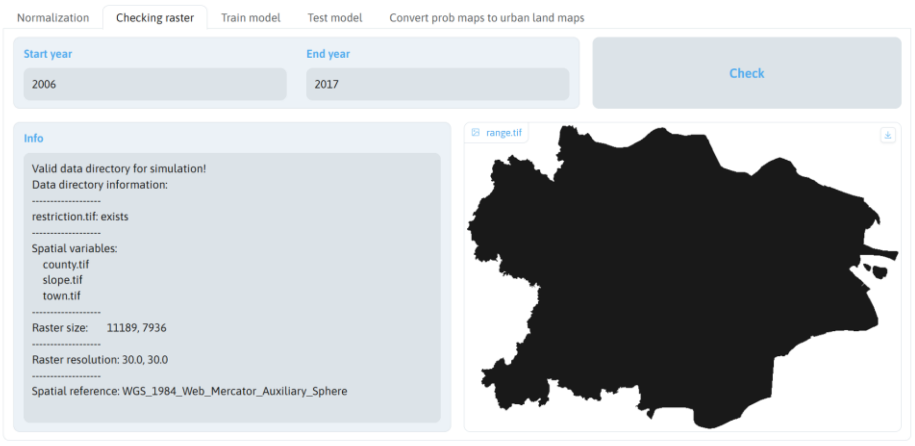
（2） Whether the range raster (range.tif), the constraint raster (constraint.tif, if exists), the spatial variable raster in the vars folder, and the land raster in the years folder have the exact same spatial reference and number of raster rows and columns.

（3） Whether the effective range of the range raster and the spatially variable raster is larger than the urban land raster. (It must be larger to ensure that edge tiles can be trained and predicted without error).

（4） Whether the spatial variable raster is normalized; whether the site raster consists of 0,1 values.

5.4.2 Case

Set the start year to 2000 and the end year to 2017 and run. After checking that there are no errors, it is as follows:



Start year: 2006

End year: 2017

Info

```
Valid data directory for simulation!
Data directory information:
-------------------
restriction.tif: exists
-------------------
Spatial variables:
   county.tif
   slope.tif
   town.tif
-------------------
Raster size:     11189, 7936
-------------------
Raster resolution: 30.0, 30.0
-------------------
Spatial reference: WGS_1984_Web_Mercator_Auxiliary_Sphere
```

Figure 5-4 Checking raster availability case

5.5 Model training (optional)

5.5.1 Introduction

The ConvLSTM model is trained for predicting urban land probability. The inputs to the model are: spatial variables + time-series urban land raster tiles; and the outputs are: time-series probability maps for future years.

5.5.2 Training data setup

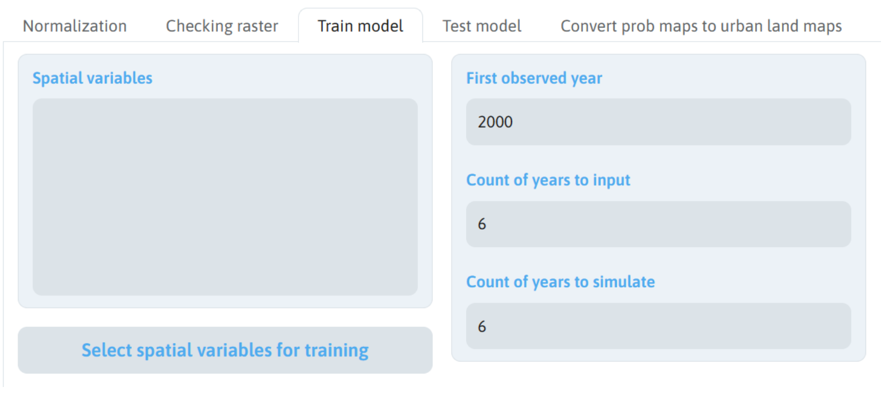
Determine the data used to train the model, including four parameters: training start year, training input length, training output length, and spatial variables.

Spatial variables: the spatial variables used for training the model. Please directly select the spatial variables in $data_dir/vars that need to be used as input.

First observed year: the first year in the training data. For example, if you want to train the model using 2000-2005 as input and 2006-2011 as output, the training start year is 2000.

Count of years to input: the number of years to input for training. For example, if you want to train the model using 2000-2005 as input and 2006-2011 as output, the training input length is 6.

Count of years to output: the number of years predicted during training. For example, if you want to train the model using 2000-2005 as input and 2006-2011 as output, the training output length is 6.


Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Spatial variables
Select spatial variables for training

First observed year: 2000

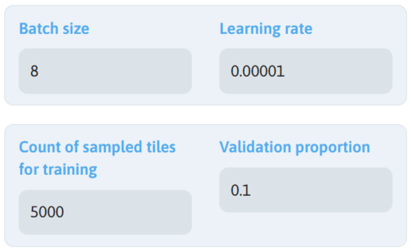
Count of years to input: 6

Count of years to simulate: 6

Figure 5-5 Model training data setup interface

5.5.3 Model parameter setting

The model parameter settings are described below:



Batch size: 8

Learning rate: 0.00001

Count of sampled tiles for training: 5000

Validation proportion: 0.1

Figure 5-6 Model training parameter setting interface

Batch size: affects the training effect of the model, the **batch size should be an integer**, the default and recommended size is 8. The batch size should be an integer, the default and recommended size is 8. The larger the batch size is, the more graphics memory will be consumed during training, about 600MB per 1 batch size. Every 1 batch size occupies about 600MB of video memory, please adjust the appropriate size according to your GPU video memory.

Learning rate (Learning rate): affects the speed of model training. The larger the learning rate, the faster the training. However, too large a learning rate may cause the model to fail to converge. The default is 0.00001, it is not recommended to adjust the learning rate to a number greater than 0.0001.

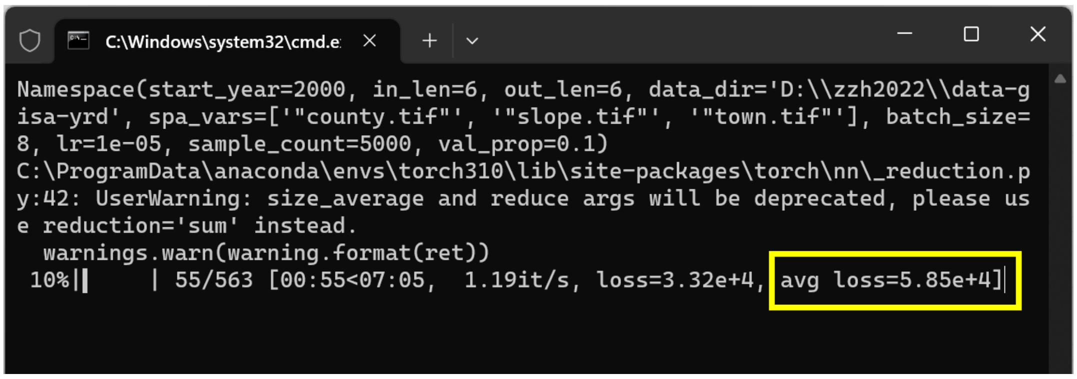
Count of sampled tiles for training: the number of samples used for training. **It is recommended to set it as a number equals or greater than 2000**. Using too few training samples may not result in a good model.

Validation proportion (Validation proportion): the proportion of the training sample used for validation. Default is 0.1.

5.5.4 Training process

After clicking on [Train], a CMD window will appear, in which information about the training process will be displayed.

During the training process, please observe the "avg_loss" (i.e. the average loss function) displayed at the end of the progress bar after each epoch of training. When the value no longer decreases or even increases slightly after several rounds of training, you can press ctrl+z to terminate the training. (It is recommended to train at least 20 epochs)



```
Namespace(start_year=2000, in_len=6, out_len=6, data_dir='D:\\zzh2022\\data-g
isa-yrd', spa_vars=['"county.tif"', '"slope.tif"', '"town.tif"'], batch_size=
8, lr=1e-05, sample_count=5000, val_prop=0.1)
C:\ProgramData\anaconda\envs\torch310\lib\site-packages\torch\nn\_reduction.p
y:42: UserWarning: size_average and reduce args will be deprecated, please us
e reduction='sum' instead.
  warnings.warn(warning.format(ret))
 10%|▌     | 55/563 [00:55<07:05,  1.19it/s, loss=3.32e+4, avg loss=5.85e+4]
```

Figure 5-7 The CMD that appears during model training

The trained models will be stored in the trained_models folder. The naming convention for the models is as follows

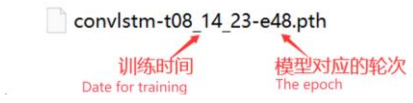
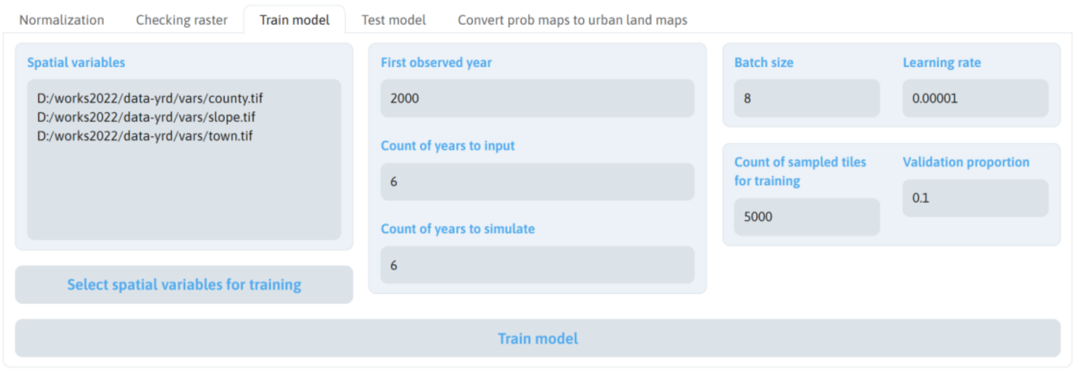


Figure 5-8 Explanation of model names

5.5.5 Case

Select the three prepared spatial variables, set the training start year to 2000, input 6 years, and output 6 years. Perform the training. Stop training after 40 rounds.


Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Select spatial variables for training
Train model

Spatial variables
D:/works2022/data-yrd/vars/county.tif
D:/works2022/data-yrd/vars/slope.tif
D:/works2022/data-yrd/vars/town.tif

First observed year: 2000
Count of years to input: 6
Count of years to simulate: 6
Batch size: 8
Learning rate: 0.00001
Count of sampled tiles for training: 5000
Validation proportion: 0.1

Figure 5-9 Example of a training model

5.6 Model testing

5.6.1 Introduction

The trained model is used to predict the probability maps. The model predicts the probability maps one slice at a time, and after completing the prediction of all the tiles, the program automatically merges the predicted probability tiles into a complete probability map for the study area.

5.6.2 Test data setting

There are five parameters that need to be set for the step.

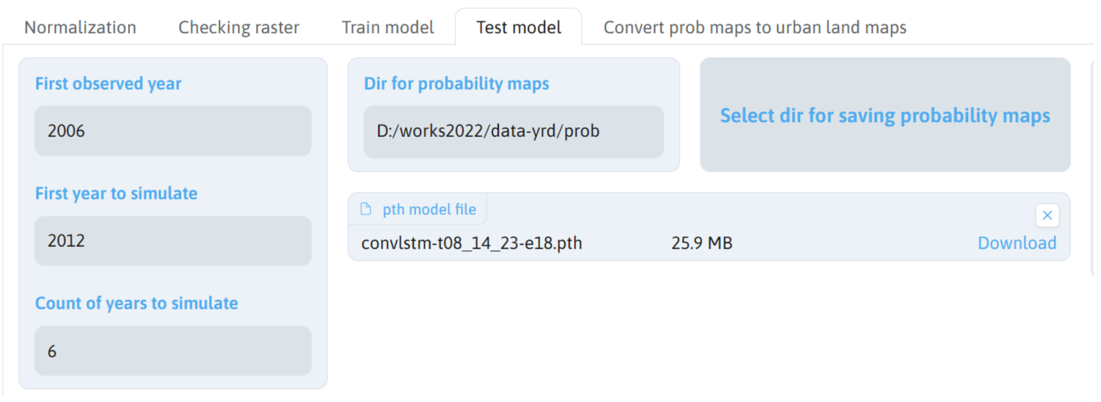
First observed year: the first year in the test data. For instance, if you want to use 2006-2011 as input to predict the probability maps of 2012-2017, the starting year of the test is 2006.

First year to simulate: the first year to **simulate**. For instance, if you want to use 2006-2011 as input and predict the probability maps of 2012-2017, the starting year of prediction is 2012.

Count of years to output: the number of years to test. For instance, if you wish to predict the probability maps for 2012-2017 using 2006-2011 as input, the number of years to test is 6.

Dir for probability maps: the folder where the conversion probability maps are stored. It is recommended to put it under $data_dir.

Model path (pth model file): the path of the trained model.


Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Select dir for saving probability maps
Download

First observed year: 2006

First year to simulate: 2012

Count of years to simulate: 6

Dir for probability maps: D:/works2022/data-yrd/prob

pth model file: convlstm-t08_14_23-e18.pth 25.9 MB

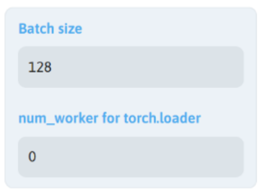
Figure 5-10 Model test data setup

5.6.3 Model parameters

In the testing phase, two parameters can be set that affect the speed of testing.

Batch size: The larger the batch size, the faster the prediction speed. However, larger batch size requires more GPU memory. Each 1 batch size occupies about 40Mb of video memory. Please adjust the appropriate size according to your GPU memory.

Number of threads (num_worker for torch.loader): if the user does not understand the mechanism of PyTorch's torch.loader, it is sufficient to keep the parameter to 0, otherwise it may lead to prediction failure. If you understand the mechanism of PyTorch's torch.loader, you can adjust it according to the demand.



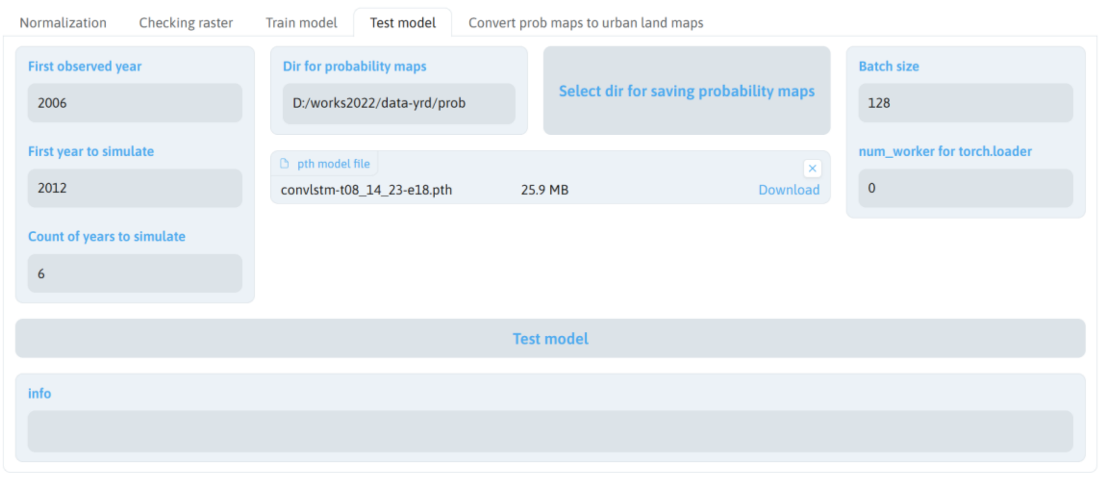
Batch size: 128

num_worker for torch.loader: 0

Figure 5-11 Model test parameter settings

5.6.4 Case

With 2006 as the first input year and 2012 as the starting year for the prediction, the prediction is made for six years, and the trained model is selected for testing, saving the time-series probability maps under $data_dir/prob. Click Test model to start the prediction. A progress bar will be displayed in the info text box.


Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Select dir for saving probability maps
Download
Test model

First observed year: 2006
First year to simulate: 2012
Count of years to simulate: 6
Dir for probability maps: D:/works2022/data-yrd/prob
pth model file: convlstm-t08_14_23-e18.pth 25.9 MB
Batch size: 128
num_worker for torch.loader: 0
info:

Figure 5-12 Test model case

5.7 Converting probability maps to simulated urban land maps

5.7.1 Introduction

This step changes a number of image elements in the conversion probability map with the highest conversion probability into urban land use based on the incremental land use demand.

5.7.2 Parameter setting

Four parameters need to be set.

Final observed landmap: The urban land map for the last input year. For instance, if 2006-2011 was used as input in the previous step [Test model], then $data_dir/year/land_2011.tif should be selected as input.

Dir for probability maps: the folder set in the previous step to save the conversion probability maps.

Dir for simulated maps: the folder where the prediction results are stored. It is recommended to put it under $data_dir.

Incremental land demand (Land demands): the incremental urban land demand (number of pixels) for each forecast year. Each year's demands are separated by commas. Please make sure that the number of land demands is equal to the number of years to predict. (Please note that the commas must be English commas.)

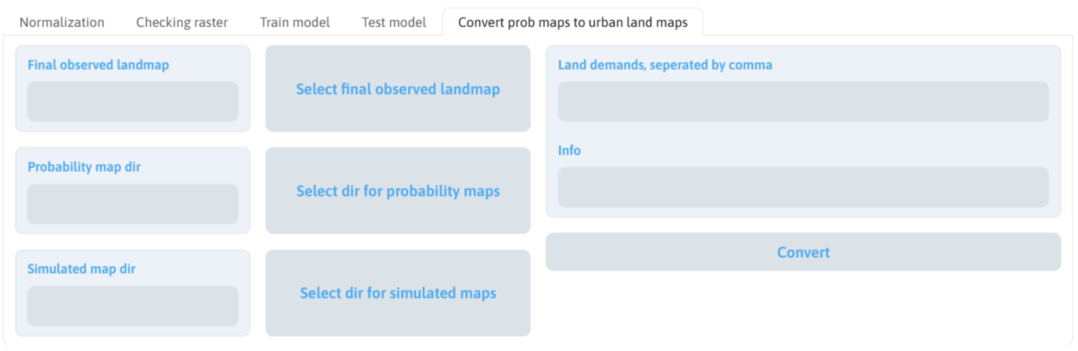
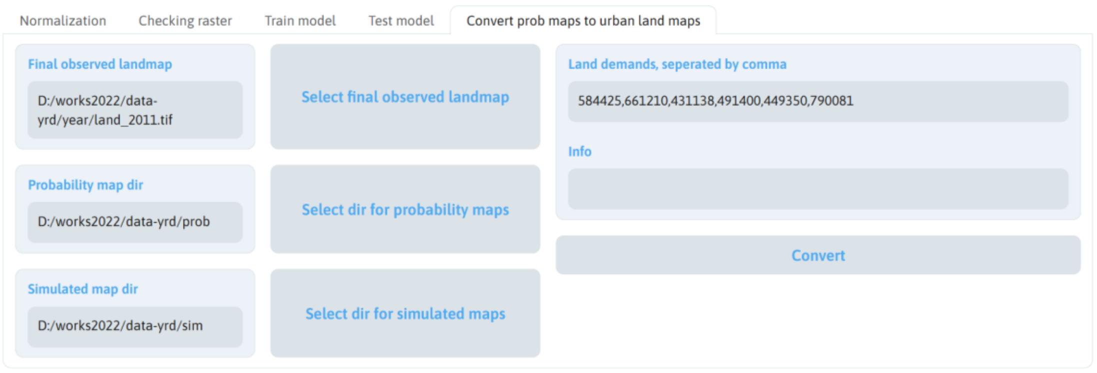

Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Final observed landmap
Select final observed landmap
Land demands, seperated by comma
Probability map dir
Select dir for probability maps
Info
Simulated map dir
Select dir for simulated maps
Convert

Figure 5-13 Results conversion interface

5.7.3 Case

Since the starting year of the prediction is 2012, select the last known map $data_dir/year/land_2011.tif, select the probability map folder as the setting in the previous step, and save the simulation results under $data_dir/sim.

The case is to test the effect of the model and use real land use data to verify the results, so the land use demand is set to be the actual urban land use demand during 2012-2017. The set land demands are 584425,661210,431138,491400,449350,790081.


Normalization
Checking raster
Train model
Test model
Convert prob maps to urban land maps
Select final observed landmap
Select dir for probability maps
Select dir for simulated maps
Convert

Final observed landmap
D:/works2022/data-yrd/year/land_2011.tif

Probability map dir
D:/works2022/data-yrd/prob

Simulated map dir
D:/works2022/data-yrd/sim

Land demands, seperated by comma
584425,661210,431138,491400,449350,790081

Info

Figure 5-14 Result conversion case

6 Validation of model effects (optional)

If you have observed data for the years to predict, you can use the Model Effectiveness Validation module to calculate the FoM (Figure of Merit) of the simulation results. The results are displayed in the [info] tab on the right side of the tab.

Parameters included:

Final observed landmap: The last known urban land map for the last year. If 2006-2011 was used as input in [Test model], then $data_dir/year/land_2011.tif should be selected as input.

Target observed map: The observed urban land map of the target year.

Target year results (simulated map): result for the target year.

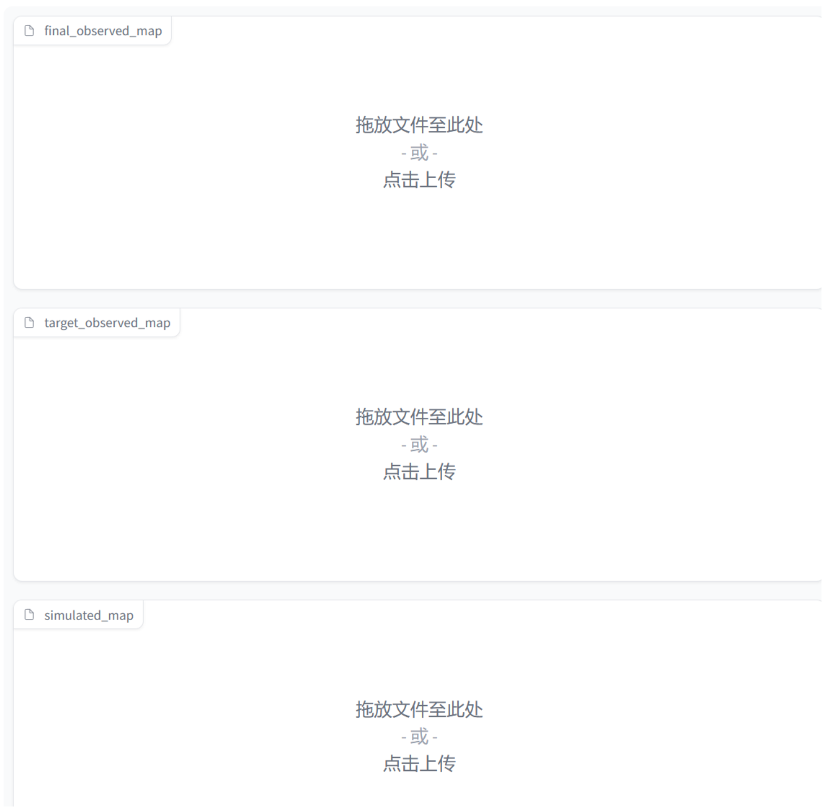


Figure 6-1 Model effect validation interface