



# A Unified Storage Format of Traffic Data: **Atomic Files in LibCity**

Jingyuan Wang

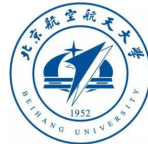
Beihang University, Beijing, China



- In order to uniformly represent different types of traffic data, LibCity defines five kinds of atomic files, that is, the five smallest information units in traffic data.

File Name	Information	Meaning
<b>xxx.geo</b>	Geographical entity information	Describe the attribute information of three types of entities of point, line, and area in geographic space, such as POI, road segment, area, etc.
<b>xxx.usr</b>	User entity information	Describe the attributes of people involved in transportation, such as age, gender, etc.
<b>xxx.rel</b>	Entity relational information	Describe the relationship between entities, such as the adjacency relationship between road sections.
<b>xxx.dyna</b>	Traffic state information	Describe the state of the traffic system on each entity, such as the speed of each intersection, etc.
<b>xxx.ext</b>	Additional auxiliary information	Describe information that helps traffic forecasts, such as weather, temperature, etc.
<b>config.json</b>	Configuration information	Used to supplement the description of the above table information.

# Atomic Files



**.geo**



Point

LineString

Polygon

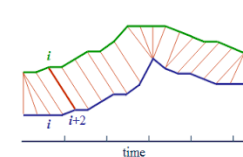
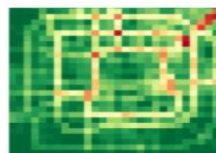
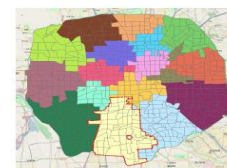
< geo\_id, type, coordinates, properties >

**.usr**



< user\_id, properties >

**.rel**



< rel\_id, type, origin\_id, destination\_id, properties >

**.dyna**

Group traffic dynamics

Flow/Speed/

Demand...

Individual traffic dynamics

Individual travel trajectory

Individual OD peering...

Relationship dynamics

Changes in connection

Changes in OD flow

**.ext**

Describe information that helps traffic forecasts, such as weather, temperature, etc.

# Atomic Files



- For different traffic prediction tasks, different atomic files may be used, and a dataset may not contain all six kinds of atomic files.
- The format of .geo, .usr, .rel, .dyna, and .ext is similar to the csv file, which consists of multiple columns of data.

geo_id	type	coordinates
773869	Point	[-118.32,34.15]
767541	Point	[-118.24,34.12]
...	...	...
769373	Point	[-118.32,34.10]

METR-LA.geo

rel_id	type	origin_id	destination_id	cost
0	geo	716328	716328	0.0
1	geo	716328	716331	4123.8
...	...	...	...	...
11752	geo	774207	774207	0.0

METR-LA.rel

dyna_id	type	time	entity_id	traffic_speed
0	state	2012-03-01T00:00:00Z	773869	64.375
1	state	2012-03-01T00:05:00Z	773869	62.667
...	...	...	...	...
7094303	state	2012-06-27T23:55:00Z	769373	61.778

METR-LA.dyna

## Geo Table: Geographical entity information

### **geo\_id, type, coordinates, properties (multiple columns).**

- **geo\_id:** The primary key uniquely determines a geo entity. (E.g. Number of sensors, latitude and longitude points, road sections, areas, etc.)
- **type:** The type of geo. Range in [Point, LineString, Polygon]. These three values are consistent with the points, lines and planes in Geojson.
- **coordinates:** Array or nested array composed of float type. Describe the location information of the geo entity, using the coordinates format of Geojson.
- **properties:** Describe the attribute information of the geo entity. If there are multiple attributes, you can use different column names to define multiple columns of data, such as POI\_name, POI\_type.

## Geo Table: Geographical entity information

**geo\_id, type, coordinates, properties (multiple columns).**

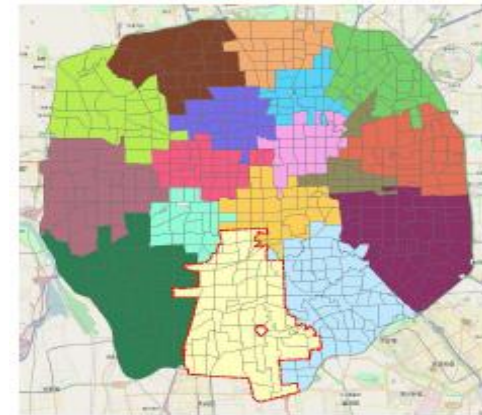
- Note: Geojson's coordinates format: (Longitude first, latitude second)
  - Point: [102.0,0.5]
  - LineString: [ [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0] ]
  - Polygon: [ [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ] ]



Point



LineString



Polygon

## Geo Table: Geographical entity information

**geo\_id, type, coordinates, properties (multiple columns).**

geo_id	type	coordinates
773869	Point	[-118.31828, 34.15497]
767541	Point	[-118.23799, 34.11620]
767542	Point	[-118.23818, 34.11640]
717447	Point	[-118.26772, 34.07248]
717446	Point	[-118.26572, 34.07142]

METR\_LA.geo

geo_id	type	coordinates	venue_category_id	venue_category_name
0	Point	[-74.003,40.733]	4bf58dd8d48988d1e7931735	Music Venue
1	Point	[-73.975,40.758]	4bf58dd8d48988d176941735	Gym / Fitness Center
2	Point	[-74.003,40.652]	4bf58dd8d48988d1e4931735	Bowling Alley
3	Point	[-73.980,40.726]	4bf58dd8d48988d118941735	Bar
4	Point	[-73.967,40.756]	4bf58dd8d48988d11d941735	Bar

Foursqaure.geo

- Note: Geojson's coordinates format: **(Longitude first, latitude second)**
  - Point: [102.0, 0.5]
  - LineString: [ [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0] ]
  - Polygon: [ [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ] ]

## Usr Table: User entity information

**usr\_id, properties (multiple columns).**

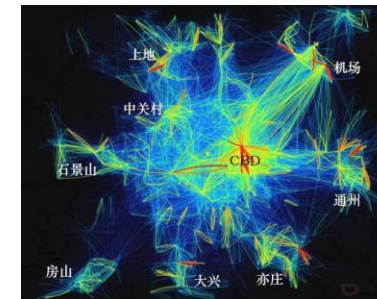
- **usr\_id**: The primary key uniquely determines a user entity.
- **properties**: Describe the attribute information of the usr entity. If there are multiple attributes, different column names can be used to define multiple columns of data, such as gender, birth\_date.

usr_id
0
1
2
3
4

Foursqaure usr



User portrait



Travel preferences



## Rel Table: Entity relational information

**rel\_id, type, origin\_id, destination\_id, properties (multiple columns).**

- **rel\_id**: The primary key uniquely determines the relationship between entities.
- **type**: The type of rel. Range in [usr, geo], which indicates whether the relationship is based on geo or usr.
- **origin\_id**: The ID of the origin of the relationship, which is either in the Geo table or in the Usr table.
- **destination\_id**: The ID of the destination of the relationship, which is one of the Geo table or the Usr table.
- **properties**: Describe the attribute information of the relationship. If there are multiple attributes, different column names can be used to define multiple columns of data.

## Rel Table: Entity relational information

**rel\_id, type, origin\_id, destination\_id, properties (multiple columns).**

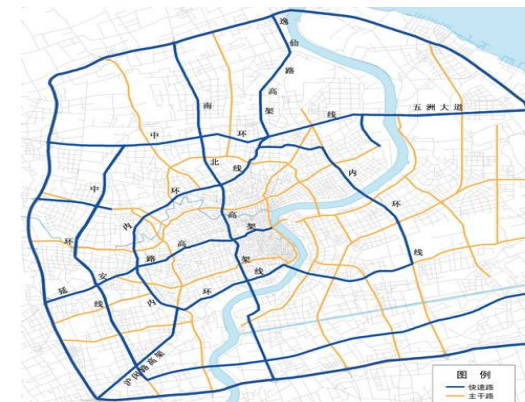
rel_id	type	origin_id	destination_id	cost	geo_id	type	coordinates
0	geo	716328	716328	0	773869	Point	[-118.31828, 34.15497]
1	geo	716328	716331	4123.8	767541	Point	[-118.23799, 34.11620]
2	geo	716328	716337	5179.6	767542	Point	[-118.23818, 34.11640]
3	geo	716328	716339	7245.5	717447	Point	[-118.26772, 34.07248]
4	geo	716328	716939	4785.1	717446	Point	[-118.26572, 34.07142]

METR\_LA.rel

METR\_LA.geo



Social Network



Road Network

## Dyna Table: Traffic state information

**dyna\_id, type, time, entity\_id(multiple columns), properties(multiple columns).**

- **dyna\_id:** The primary key uniquely determines a record in the Dyna table.
- **type:** The type of dyna. There are two values: trajectory (for trajectory based task) and state (for traffic state prediction task).
- **time:** Time information, using the date and time combination notation in ISO-8601 standard, such as: 2020-12-07T02:59:46Z.
- **entity\_id:** Describe which entity the record is based on, which is the ID of geo or usr.
- **properties:** Describe the attribute information of the record. If there are multiple attributes, different column names can be used to define multiple columns of data, such as both speed data and flow data.

## Dyna Table: Traffic state information

- type: state
- Point-based / Road-based / Region-based / Grid-based / Od-based / Grid-Od-based
- The format is: **dyna\_id, state, time, entity\_id, properties.**
  - The entity\_id column varies with the changes of different data structures for ease of use.
- The rows in the table should be aggregated according to <entity\_id>, rows with the same <entity\_id> are sorted by <time>.

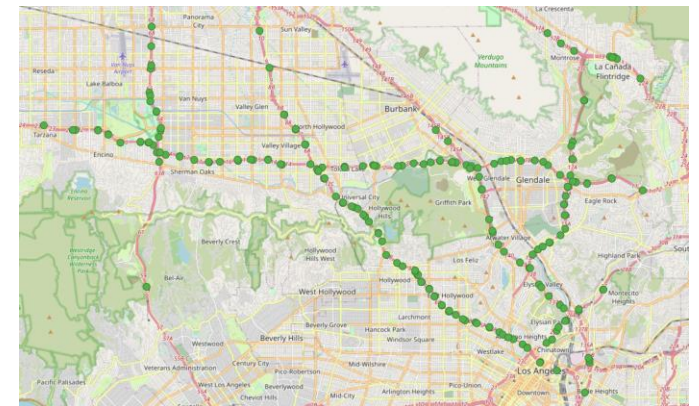
## Dyna Table: Traffic state information

- type: state— —Point-based / Road-based / Region-based
- The format is: **dyna\_id, state, time, entity\_id, properties.**
- For entities that can use one-dimensional numbering for sensors, road sections, areas, etc., **entity\_id** is the corresponding ID, the column name is [**entity\_id**], and the file suffix name is **.dyna**.

dyna_id	type	time	entity_id	traffic_speed	geo_id	type	coordinates
0	state	2012-03-01T00:00:00Z	773869	64.375	773869	Point	[-118.31828, 34.15497]
1	state	2012-03-01T00:05:00Z	773869	62.66667	767541	Point	[-118.23799, 34.11620]
2	state	2012-03-01T00:10:00Z	773869	64	767542	Point	[-118.23818, 34.11640]
3	state	2012-03-01T00:15:00Z	773869	0	717447	Point	[-118.26772, 34.07248]
4	state	2012-03-01T00:20:00Z	773869	0	717446	Point	[-118.26572, 34.07142]

METR\_LA.dyna

METR\_LA.geo



Point-based

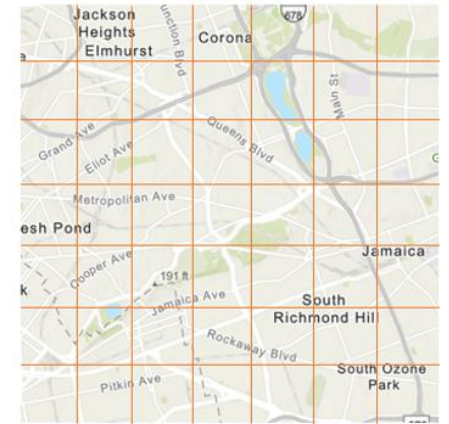
## Dyna Table: Traffic state information

- type: state — — Grid-based
- The format is: **dyna\_id, state, time, entity\_id, properties.**
- For grid-based traffic data, the **entity\_id** is **[row\_id, column\_id]**, and the file extension is **.grid**.

dyna_id	type	time	row_id	column_id	risk	geo_id	type	coordinates	row_id	column_id
0	state	2013-01-01T00:00:00Z	0	0	0	0	Polygon	[]	0	7
1	state	2013-01-01T01:00:00Z	0	0	0	1	Polygon	[]	0	8
2	state	2013-01-01T02:00:00Z	0	0	0	2	Polygon	[]	0	10
3	state	2013-01-01T03:00:00Z	0	0	0	3	Polygon	[]	0	11
4	state	2013-01-01T04:00:00Z	0	0	0	4	Polygon	[]	0	12

NYC\_RISK.grid

NYC\_RISK.geo



Grid-based

## Dyna Table: Traffic state information

- type: state— —Od-based
- The format is: **dyna\_id, state, time, entity\_id, properties.**
- For od-based traffic data, the **entity\_id** is **[origin\_id, destination\_id]**, and the file suffix name is **.od**.

dyna_id	type	time	origin_id	destination_id	flow	geo_id	type	coordinates
0	state	2012-03-01T00:00:00Z	0	1	345	0	Point	[-118.31828, 34.15497]
1	state	2012-03-01T00:05:00Z	0	2	277	1	Point	[-118.23799, 34.11620]
2	state	2012-03-01T00:10:00Z	0	3	64	2	Point	[-118.23818, 34.11640]
3	state	2012-03-01T00:15:00Z	0	4	0	3	Point	[-118.26772, 34.07248]
4	state	2012-03-01T00:20:00Z	1	2	0	4	Point	[-118.26572, 34.07142]

Data.od

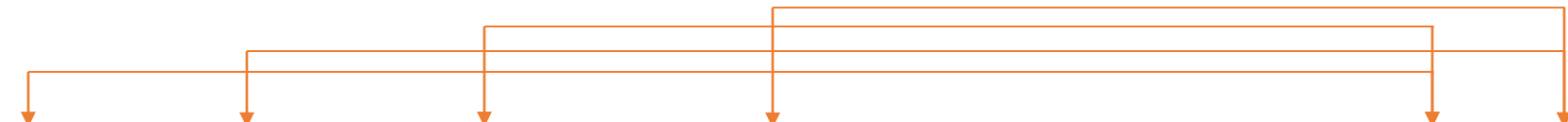
Data.geo



OD-based

## Dyna Table: Traffic state information

- type: state — — Grid-Od-based
- The format is: **dyna\_id, state, time, entity\_id, properties.**
- For grid-od-based traffic data, the **entity\_id** is [**origin\_row\_id, origin\_column\_id, destination\_row\_id, destination\_column\_id**], and the file extension is **.gridod**.



dyna_id	type	time	origin_row_id	origin_column_id	destination_row_id	destination_column_id	geo_id	type	coordinates	row_id	column_id
0	state	2013-01-01T00:00:00Z	0	1	2	1	0	Polygon	[]	0	7
1	state	2013-01-01T01:00:00Z	0	1	2	1	1	Polygon	[]	0	8
2	state	2013-01-01T02:00:00Z	0	1	2	1	2	Polygon	[]	0	10
3	state	2013-01-01T03:00:00Z	0	1	2	1	3	Polygon	[]	0	11
4	state	2013-01-01T04:00:00Z	0	1	2	1	4	Polygon	[]	0	12

Data.gridod Data.geo



## Dyna Table: Traffic state information

- type: trajectory
- GPS point trajectory / Road segment-based trajectory / Check-in trajectory
- The format is: **dyna\_id, type, time, entity\_id, (traj\_id), properties.**
  - The entity\_id column should be **usr\_id**.
  - The traj\_id column represents the number of multiple trajectories of the same user (starting from 0) and if the user has only one trajectory this column can be empty.
- The rows in the table should be aggregated according to <entity\_id>, rows with the same <entity\_id> are sorted by <traj\_id>, and rows with the same <traj\_id> are sorted by <time>.

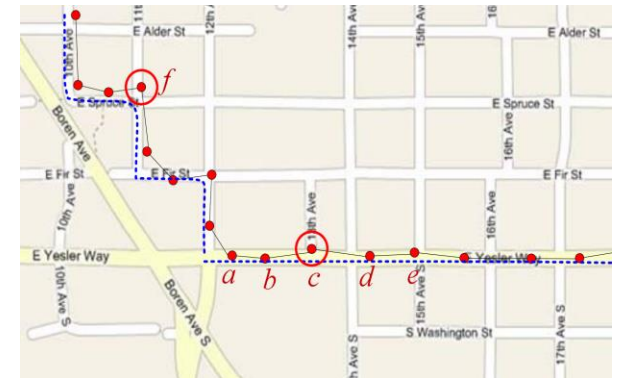
## Dyna Table: Traffic state information

- type: trajectory — — GPS point trajectory
- The format is: **dyna\_id, type, time, entity\_id, (traj\_id), coordinates, properties.**
- The coordinates column is the latitude and longitude of the GPS point.

dyna_id	type	time	entity_id	traj_id	coordinates	current_state	usr_id
0	trajectory	2014-08-03T18:29:00Z	810	0	"[104.115353,30.64392]"	1	810
1	trajectory	2014-08-03T18:29:40Z	810	0	"[104.113091,30.642129]"	1	811
...	...	...	...	...	...	...	812
21	trajectory	2014-08-03T18:53:23Z	810	0	"[104.076552,30.626844]"	0	813
22	trajectory	2014-08-03T18:13:00Z	810	1	"[104.106701,30.6916]"	1	814
...	...	...	...	...	...	...	...
241	trajectory	2014-08-03T18:16:00Z	11919	7	"[104.100816,30.706191]"	1	...

chengdu.dyna

chengdu.usr



GPS Point Trajectory



## Dyna Table: Traffic state information

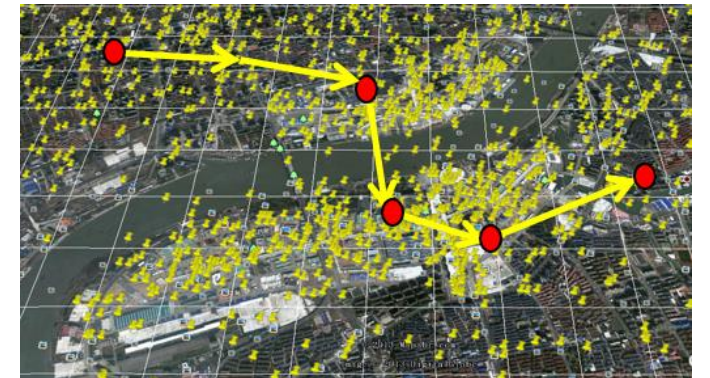
- type: trajectory — — Check-in trajectory
- The format is: **dyna\_id, type, time, entity\_id, (traj\_id), location, properties.**
- The content of the location column is geo\_id, which refs to the geo table and represents a POI.

usr_id	dyna_id	type	time	entity_id	location	geo_id	type	coordinates
0	0	trajectory	2009-01-17T20:27:37Z	0	0	0	Point	[-122.7323,47.8899]
1	1	trajectory	2009-01-17T20:27:38Z	0	1	1	Point	[-122.7321,47.8903]
2	2	trajectory	2009-01-17T20:27:39Z	0	2	2	Point	[-122.7318,47.8910]
3	3	trajectory	2009-01-17T20:27:40Z	0	2	3	Point	[-122.7313,47.8921]
4	4	trajectory	2009-01-17T20:27:41Z	0	3	4	Point	[-122.7307,47.8933]

Data.usr

Data.dyna

Data.geo



Check-in Trajectory

## Ext Table: Additional auxiliary information

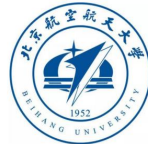
### **ext\_id, time, properties (multiple columns).**

- **ext\_id**: The primary key uniquely determines a record in the external data table.
- **time**: Time information, using the date and time combination notation in ISO-8601 standard, such as: 2020-12-07T02:59:46Z.
- **properties**: Describe the attribute information of the record.

ext_id	time	temperature
0	2012-03-01T00:00:00Z	272.03
1	2012-03-01T00:05:00Z	271.46
2	2012-03-01T00:10:00Z	271.19
3	2012-03-01T00:15:00Z	271.07
4	2012-03-01T00:20:00Z	270.83

Data.ext

# Data Type Definition

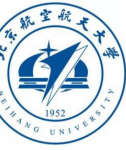


The data type definition of each column in the dataset needs to be given in the config file, which is helpful for subsequent data processing.

Type	Description
geo_id	Discrete limited IDs that exist in the Geo table.
usr_id	Discrete limited IDs that exist in the Usr table.
rel_id	Discrete limited IDs that exist in the Rel table.
time	Time string conforming to ISO-8601 standard.
coordinate	String conforming to the coordinate representation in Geojson format.
num	Real number.
enum	Enum string.
other	The rest are stored in string type.

- **The config file is used to supplement the information describing the above five tables. It is stored in json format and consists of six keys: geo, usr, rel, dyna, ext, and info.**
  - **For geo, rel, dyna:**
    - Contains a key of `including_types`, and uses an array to describe the type values in the table. After that, each type is used as a key, describing which keys are contained in the properties table and their data types under the type.
  - **For usr, ext:**
    - Contains a `properties` key, describing which keys are contained in the properties table and their data types.
  - **For info:**
    - Contains other necessary statistical information of the dataset, for different traffic prediction tasks, contains different contents.

# Config File



```
"geo":{
  "including_types":[
    "Point"
  ],
  "Point":{
    "poi_name":"other",
  }
}

"usr":{
  "properties":{
    "user_type":"enum",
    "birth_year":"time",
    "gender":"enum"
  }
},

"rel":{
  "including_types":[
    "geo"
  ],
  "geo":{
    "link_weight":"num"
  }
},

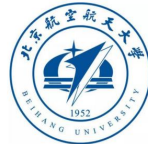
"dyna":{
  "including_types":[
    "state"
  ],
  "state":{
    "entity_id":"geo_id",
    "traffic_speed":"num"
  }
},

"ext":{
  "properties":{
    "temperature":"num"
  }
},

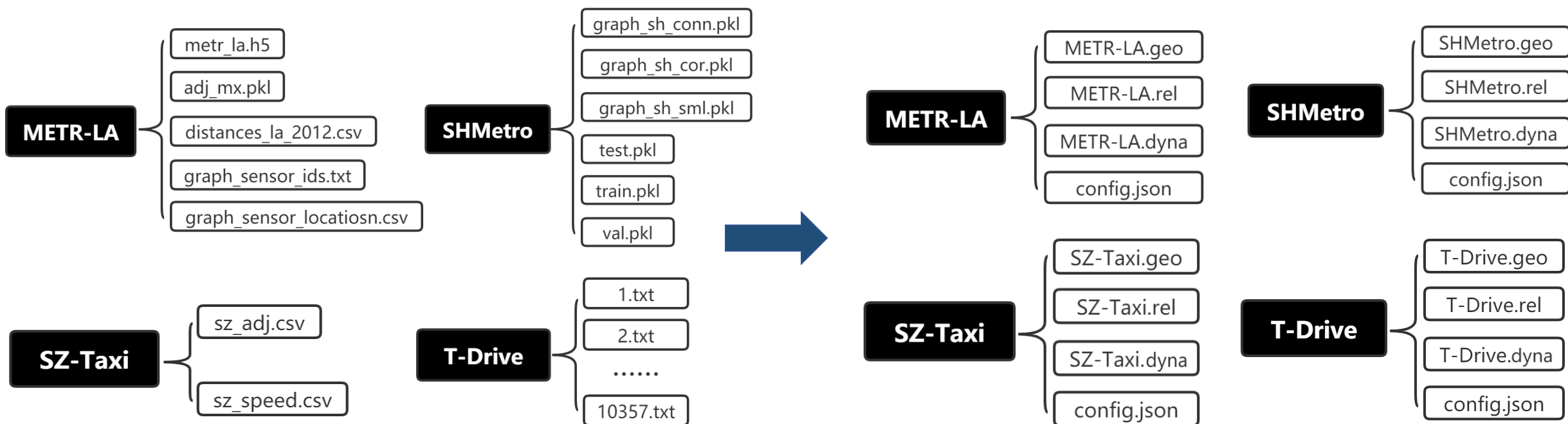
"info": {
  "time_interval": 300,
}
```



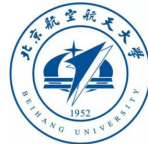
# Application



- We use the **atomic files** format to integrate 35 traffic datasets in **LibCity**, an open library for traffic prediction, which solves the problem of inconsistencies in the storage format of traffic datasets.



# Application



- LibCity has converted **35** open source datasets covering **22 cities in 11 countries** into standard atomic format datasets.
- LibCity also open sourced the **atomic file conversion scripts** for users to refer to when converting their own traffic datasets.

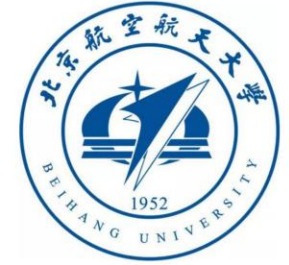


Datasets Spatial Distribution

DATASET	#GEO	#REL	#USR	#DYNA	PLACE	DURATION	INTERVAL
METR-LA[19]	207	11,753		7,094,304	Los Angeles, USA	Mar. 1, 2012 - Jun. 27, 2012	5min
Los-loop[43]	207	42,849		7,094,304	Los Angeles, USA	Mar. 1, 2012 - Jun. 27, 2012	5min
SZ-Taxi[43]	156	24,336		464,256	Shenzhen, China	Jan. 1, 2015 - Jan. 31, 2015	15min
Loop Seattle[8, 9]	323	104,329		33,953,760	Greater Seattle Area, USA	over the entirety of 2015	5min
Q-Traffic[21]	45,148	63,422		264,386,688	Beijing, China	Apr. 1, 2017 - May 31, 2017	15min
PeMSD3[31]	358	547		9,382,464	California, USA	Sept. 1, 2018 - Nov. 30, 2018	5min
PeMSD4[13]	307	340		5,216,544	San Francisco Bay Area, USA	Jan. 1, 2018 - Feb. 28, 2018	5min
PEMSD7[31]	883	866		24,921,792	California, USA	Jul. 1, 2016 - Aug. 31, 2016	5min
PEMSD8[13]	170	277		3,035,520	San Bernardino Area, USA	Jul. 1, 2016 - Aug. 31, 2016	5min
PEMSD7(M)[38]	228	51,984		2,889,216	California, USA	weekdays of May and June, 2012	5min
PEMS-BA[19]	325	8,358		16,937,700	San Francisco Bay Area, USA	Jan. 1, 2017 - Jun. 30, 2017	5min
Beijing subway[41]	276	76,176		248,400	Beijing, China	Feb. 29, 2016 - Apr. 3, 2016	30min
M_dense[11]	30			525,600	Madrid, Spain	Jan. 1, 2018 - Dec. 21, 2019	60min
Rotterdam[14]	208			4,813,536	Rotterdam, Holland	135 days of 2018	2min
SH-Taxi[32]	228	22,844		1,224,000	Shanghai, China	Jul. 1, 2016 - Sep. 30, 2016	15min

Dataset Statistics Table

LibCity Datasets: <https://github.com/LibCity/Bigcity-LibCity-Datasets>



# Thanks for listening.

Jingyuan Wang

Beihang University, Beijing, China

[jywang@buaa.edu.cn](mailto:jywang@buaa.edu.cn), <http://www.bigcity.ai>

