

Output_SUMO

1. 基于车辆的信息和分类

1-1. 车辆位置转储--RawDump

- [RawDump - SUMO Documentation](#)
- 输出:网络转储是一个 xml 文件, 其中包含每个时间步长的网络的每个边缘以及该边缘的每条车道以及该车道上的所有车辆。对于每辆车, 它的名称、速度和在各自车道上的位置都会被写下来。

```
1 <netstate>
2   <timestep time="<TIME_STEP>">
3     <edge id="<EDGE_ID>">
4       <lane id="<LANE_ID>">
5         <vehicle id="<VEHICLE_ID>" pos="<VEH_POSITION>" speed="<VEH_SPEED>"/>
6
7         ... more vehicles if any on this lane ...
8
9       </lane>
10
11     ... more lanes if the edge possesses more ...
12
13     ... optional persons and containers if currently active on that edge
14     <person id="<ID>" pos="<OFFSET_FROM_EDGE_BEGIN>" speed="<SPEED>"/>
15     ...
16     <container id="<ID>" pos="<OFFSET_FROM_EDGE_BEGIN>" speed="<SPEED>"/>
17     ...
18
19   </edge>
20
21   ... more edges ....
22
23 </timestep>
24
25 ... the next timestep ...
26
27 </netstate>
```

1-2. 完整输出--FullOutput

- [FullOutput - SUMO Documentation](#)

- 作用: 完整转储是一个 xml 文件, 其中包含每个时间步的每条边、车道、车辆和交通灯的信息。完整的转储文件如下所示:

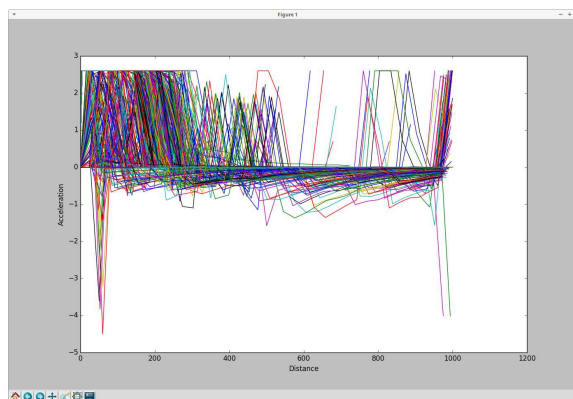
```
1 <full-export>
2   <data timestep="<TIME_STEP>">
3
4   <vehicles>
5     <vehicle id="<VEHICLE_ID>" eclass="<VEHICLE_ECLASS>" co2="<VEHICLE_CO2>"
6     nox="<VEHICLE_NOX>" pmx="<VEHICLE_PMX>" fuel="<VEHICLE_FUEL>" electricit
7     waiting="<VEHICLE_WAITING>" lane="<VEHICLE_LANE>" pos_lane="<VEHICLE_POS
8     angle="<VEHICLE_ANGLE>" x="<VEHICLE_POS_X>" y="<VEHICLE_POS_Y>"/>
9
10    ... more vehicles ...
11
12  </vehicles>
13
14  <edges>
15
16    <edge id="<EDGE_ID>" traveltime="<EDGE_TRAVELTIME>">
17
18    <lane id="<LANE_ID>" co="<LANE_CO>" co2="<LANE_CO2>" nox="<LANE_NOX>" pm
19    hc="<LANE_HC>" noise="<LANE_NOISE>" fuel="<LANE_FUEL>" electricity="<LAN
20    occupancy="<LANE_OCCUPANCY>" vehicle_count="<LANE_VEHICLES_COUNT>"/>
21
22    ... more lanes of the edge if exists
23
24  </edge>
25
26    ... more edges of the network
27
28  </edges>
29
30  <tls>
31    <trafficlight id="0/0" state="GgGr"/>
32    ... more traffic lights
33
34  </tls>
35
36 </data>
37
38 ... the next timestep ...
39
40 </full-export>
```

1-3. 车辆GPS位置输出--FCDOOutput

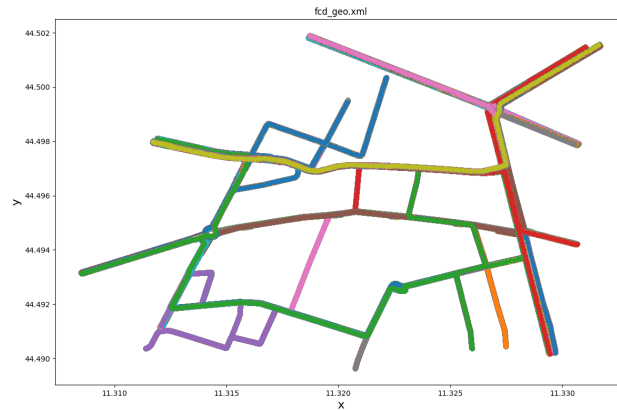
- [FCDOOutput - SUMO Documentation](#)
- 作用: FCD（浮动汽车数据）导出包含位置和速度以及网络中每辆车在每个时间步长的其他信息。输出的行为有点像每辆车的超精确高频 GPS 设备。可以使用[TraceExporter 工具](#)进一步处理输出，以调整频率、设备速率、准确性和数据格式。

```
1 <fcd-export>
2
3   <timestep time="<TIME_STEP>">
4       <vehicle id="<VEHICLE_ID>" x="<VEHICLE_POS_X>" y="<VEHICLE_POS_Y>" angle="
5           speed="<VEHICLE_SPEED>"/>
6
7       ... more vehicles ...
8
9   </timestep>
10
11   ... next timestep ...
12
13 </fcd-export>
```

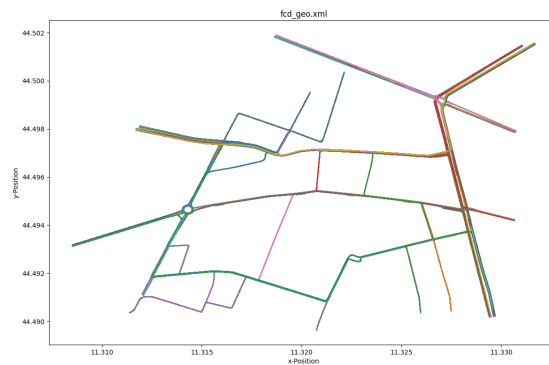
- 可视化举例
 - 加速度与距离



- 随时间变化的轨迹



- 基于 FCD 的速度随时间变化



2. 基于车辆的信息

2-1. 行程信息

- [TripInfo - SUMO Documentation](#)--TripInfo
- 作用: 此输出包含有关每辆车的出发时间、车辆想要出发的时间（可能低于实际出发时间）和车辆到达时间的信息。一旦车辆到达目的地并从网络中删除，就会为每辆车生成信息。

```

1 <tripinfos>
2   <tripinfo id="<VEHICLE_ID>" \
3     depart="<DEPARTURE_TIME>" departLane="<DEPARTURE_LANE_ID>" \
4     departPos="<DEPARTURE_POSITION>" departSpeed="<DEPARTURE_SPEED>" \
5     departDelay="<DEPARTURE_DELAY>" \
6     arrival="<ARRIVAL_TIME>" arrivalLane="<DEPARTURE_LANE_ID>" \
7     arrivalPos="<ARRIVAL_POSITION>" arrivalSpeed="<ARRIVAL_SPEED>" \
8     duration="<TRAVEL_TIME>" routeLength="<ROUTE_LENGTH>" \
9     waitingTime="<SECONDS_WAITING_FOR_TRAFFIC>"
10    waitingCount="<NUMBER_OF_WAITING_EPISODES>" \
11    rerouteNo="<REROUTE_NUMBER>" \
12    devices="<DEVICE_LIST>" vtype="<VEHICLE_TYPE_ID>"/>
13

```

```

14     ... information about further vehicles ...
15
16 </tripinfos>

```

2-2. 车辆路线--VehRoutes

- [VehRoutes - SUMO Documentation](#)

- 车辆路线输出包含有关车辆所走路线的信息，如果他的路线在任何时候被新路线替换，则报告之前的每条路线以及更换发生时的边。此外，车辆进入和离开网络的时间存储在此处。在正常情况下，当所有车辆都使用预定义路线时，输出不包含任何无法从路线和 tripinfo 输出中检索到的信息。但是，一旦您在模拟中重新安排车辆路线（如果使用重新安排路线），它将包含新信息。

```

1 <routes>
2   <vehicle id="<VEHICLE_ID>" [type="<TYPE_ID>"] depart="<INSERTION_TIME>" arri
3     <routeDistribution>
4       <route replacedOnEdge="<EDGE_ID>" replacedAtTime="<TIME>" probabilit
5
6       ... further replaced routes ...
7
8       <route edges="<LAST_ROUTE>" [exitTimes="<EXIT_TIMES>"]/>
9     </routeDistribution>
10  </vehicle>
11
12  <person id="<PERSON_ID>" depart="<INSERTION_TIME>" arrival="<ARRIVAL_TIME>">
13    <ride from="..." to="..." lines="..." [started="<START_TIME>" ended="<EN
14    <walk edges="..." speed="..." [exitTimes="<EXIT_TIMES>" started="<START_
15  </person>
16
17  ... information about further vehicles and persons ...
18
19 </routes>

```

3. 基于模拟网络的信息

3-1. 模拟状态汇总统计--车辆计数等

- [Summary - SUMO Documentation](#)

- 作用: 此输出包含模拟范围内已装载、插入、运行、等待插入、已到达目的地的车辆数量以及完成路线所需的时间。最后一个值对目前已到达目的地的所有车辆进行归一化。

```

1 <summary>

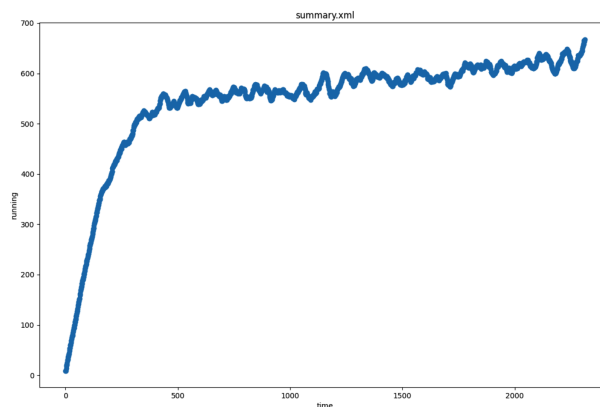
```

```

2      <step time="<SIMULATION_TIME>" \
3          loaded="<LOADED_VEHICLE_NUMBER>" \
4          inserted="<INSERTED_VEHICLE_NUMBER>" \
5          running="<RUNNING_VEHICLE_NUMBER>" \
6          waiting="<NUMBER_OF_VEHICLES_WAITING_FOR_INSERTION>" \
7          ended="<ENDED_VEHICLE_NUMBER>" \
8          meanWaitingTime="<MEAN_WAITING_TIME>" \
9          meanTravelTime="<MEAN_TRAVEL_TIME>"/>
10
11      ... further time steps ...
12
13 </summary>

```

可视化示例
随着时间的推移运行和停止车辆的数量



3-2. 统计输出--模拟的总体统计数据

- [Statistic Output - SUMO Documentation](#)
- 作用: 模拟的总体统计数据（车辆、传送、安全、人员、vehicleTripStatistics、rideStatistics 等）

```

1 <statistics>
2   <vehicles 'ATTRIBUTES'.../>
3   <teleports 'ATTRIBUTES'.../>
4   <safety 'ATTRIBUTES'.../>
5   <persons 'ATTRIBUTES'.../>
6   <vehicleTripStatistics 'ATTRIBUTES'.../>
7   <pedestrianStatistics 'ATTRIBUTES'.../>
8   <rideStatistics 'ATTRIBUTES'.../>
9   <transportStatistics 'ATTRIBUTES'.../>
10 </statistics>

```

附录: SUMO所有输出总结

🏠 / 模拟 / 输出

输出

介绍

sumo允许生成大量不同的措施。所有人都按照写入文件的通用规则将收集到的值写入文件或套接字连接中。默认情况下，所有这些都被禁用，并且必须单独触发。一些可用的输出（原始车辆位置转储、行程信息、车辆路线信息和模拟状态统计）是使用命令行选项触发的，其他输出必须在附加文件中定义。

转换输出

SUMO 编写的所有输出文件默认都是 XML 格式。但是，使用 python 工具xml2csv.py，您可以将它们中的任何一个转换为可以用大多数电子表格软件打开的平面文件 (CSV) 格式。如果您需要更压缩但仍然“标准化”的二进制版本，您可以使用 xml2protobuf.py。此外，所有文件都可以以压缩形式 (gzip) 写入和读取，这是由文件扩展名 .gz 触发的。

分离重复运行的输出

为了使多个模拟运行的输出分开，选项--output-prefix <STRING> 可用于为所有输出文件名添加前缀。设置--output-prefix TIME时，所有输出都将使用模拟开始的时间作为前缀，从而使它们自动分开。

可用的输出文件

下面列出了可用的输出，并加入到主题/聚合类型的组中。有关每个输出的更多信息，请访问其链接。

基于车辆的信息，分类

- 原始车辆位置转储：所有车辆随时间的位置 包含：所有车辆在所有模拟时间步长中的位置和速度， 用于：获取节点的移动（V2V，对于 ns-2）
- 排放输出：每个模拟步骤中所有车辆的排放值
- 完整输出：所有边缘、车道和车辆的各种信息（有利于可视化目的）
- vtk 输出：以众所周知的VTK [🔗](#)（可视化工具包）格式生成文件，以显示每辆车的位置和速度值
- fcd 输出：浮动汽车数据包括每辆车的名称、位置、角度和类型
- 轨迹输出：轨迹数据如下，包括符合 Amitran 标准的每辆车的名称、位置、速度和加速度
- lanechange output：车道变更事件，每辆车都有相关的变更动机
- 替代安全措施（SSM）：安全相关措施、车头时距、刹车率等的输出
- vehicle type probe：特定车辆类型随时间推移的车辆位置（已弃用，请改用带有 FCD 输出的 vType 过滤器）

模拟探测器

- 感应回路探测器（E1）：模拟感应回路
- 即时感应回路：模拟未聚合感应回路
- 车道区域探测器（E2）：捕获车道段的探测器（即模拟车辆跟踪摄像机）
- 多进出探测器 (E3)：通过检测定义位置的进出事件来跟踪区域交通的模拟器
- Route Detectors：采样路径分布探测器

边缘或车道的值

- edgelane 流量：基于边缘/车道的网络性能测量
- 聚合的 Amitran 度量：遵循 Amitran 标准的基于边缘/车道的网络性能度量
- edgelane emissions：基于边缘/车道的车辆污染物排放
- edgelane noise：基于边缘/车道的车辆噪声发射；基于Harmonoise
- 队列输出：基于车道的路口前实际尾部计算

连接点的值

交叉路口的交通没有专门的输出格式。相反，可以通过在交叉路口放置测量交通量的探测器来测量与交叉路口相关的交通量。

- Tools/Output#generateTLSE1Detectors.py 脚本，用于在所有 TLS 控制的交叉路口周围生成感应环探测器（在单个车道上基于点检测）
- Tools/Output#generateTLSE2Detectors.py 脚本，用于在所有 TLS 控制的交叉路口周围生成车道区域探测器（对单个车道进行基于区域的检测）
- Tools/Output#generateTLSE3Detectors.py 脚本，用于在所有 TLS 控制的交叉路口或任意交叉路口列表周围生成多入口-出口探测器。探测器可以配置为聚合或分离接近的边缘，并包括或排除路口内部。（基于区域的边缘检测）

或者，可以手动聚合边缘或车道的值以获得交叉点处的流量。



基于车辆的信息

- 行程信息：关于每辆车行程的汇总信息（可选排放数据）
- 车辆路线信息：每辆车模拟运行的路线信息
- stop output：车辆停靠 和人员、集装箱装卸信息
- 电池使用情况：有关电动汽车电池状态的信息
- collision output：车辆之间和车辆与行人之间的碰撞信息

基于模拟（网络）的信息

- **模拟状态汇总统计**：关于模拟当前状态的信息（车辆计数等）
- **模拟状态人员汇总统计**：有关模拟人员当前状态的信息（人数等）
- **统计输出**：模拟的总体统计数据（车辆、传送、安全、人员、vehicleTripStatistics、rideStatistics 等）

基于交通信号灯的信息

- **交通灯状态**：有关交通灯状态（灯）的信息
- **stream-based traffic light switches**：有关负责特定链接的交通灯信号开关的信息
- **traffic light states, by switch**：有关交通灯信号状态（灯）的信息，仅在更改时写入
- 与 **tls 耦合的区域检测器**：由 tls 触发的模拟车辆跟踪摄像机

额外的调试输出

- 选项 **--link-output <FILE>** 保存交叉模型的调试数据。该数据揭示了每辆车打算占用即将到来的十字路口的时间。
- 选项 **--movereminder-output <FILE>** 为车辆设备、车道和输出设施之间的交互保存调试数据。它仅在使用调试标志编译 **sumo** 时可用。
- 选项 **--railsignal-block-output <FILE>** 保存有关铁路信号块的信息。对于每个受控的 railSignal 链接，都会生成以下信息：
 - **forwardBlock**：从信号化链路向前到达下一个铁路信号的所有车道
 - **bidiBlock**：构成反向轨道的所有车道在前向块内遇到并跟随到下一个 railSignal 超出允许通过的铁路道岔
 - **backwardBlock**：所有从外部进入 forwardBlock 或 bidiBlock 的车道都跟随上游到入口铁路信号
 - **conflictLinks**：所有从外部进入冲突区域（forwardBlock、bidiBlock、backwardBlock）的受控链接，编码为 **<SIGNALID>_<LINKINDEX>**

命令行输出（步骤日志）

默认情况下，sumo 会打印一些“心跳”信息以表明它仍在运行。每100个模拟步骤将打印以下信息：

- **步骤 #**：当前模拟时间
- **最后一步的持续时间（毫秒）**
- **实时因素（步长/持续时间）。（转发）**
- **每秒更新的车辆数量（UPS）**
- **TraCI**：当前步骤TraCI处理所花费的时间（包括外部脚本）
- **vehicles TOT**：到目前为止离开的车辆数量
- **ACT**：当前运行的车辆数量
- **BUF**：延迟插入的车辆数量

可以使用选项 **--no-step-log** 禁用此输出。它的周期可以用选项 **--step-log.period TIME** 配置。

命令行输出（详细）

当使用选项 **--verbose**（短 **-v**）运行模拟时，将打印以下数据（除非使用选项 **--duration-log false** 明确禁用）：

车辆计数

- **插入**：进入模拟网络的车辆数量
- **已加载**：从路线文件加载的车辆数量。这可能与发出的不同，原因有二：
 - 使用选项 **--scale** 运行，值小于 1.0
 - 拥有一个堵塞的网络，在模拟时间结束之前无法插入所有车辆
- **Running**：仿真结束时网络中当前活跃的车辆数量
- **等待**：由于堵塞而无法接入网络的车辆数量
- **传送**：由于以下任何原因传送车辆的次数（每当发出传送警告时都会给出这些原因）
 - **碰撞**：车辆违反了与其领导车辆相关的 minGap 要求
 - **超时**：车辆在 **--time-to-teleport** 秒内无法移动（默认 300）
 - 错误的车道：车辆无法移动，因为它无法在当前车道上继续行驶并且无法改变到正确的车道
 - **yield**：车辆无法通过没有优先权的十字路口
 - **jam**：车辆无法继续行驶，因为下一车道没有空间

如果模拟包含人员，将添加以下输出：

- **插入**：从路线文件中加载的人数
- **running**：模拟结束时网络活跃人数
- **Jammed**：一个人被卡住的次数

时序数据

- **持续时间**：计算模拟时经过的时间量（由挂在墙上的时钟测量）。
- **TraCI-Duration**：处理 TraCI 命令所花费的时间量（以挂在墙上的时钟衡量）。这包括在 traci 客户端脚本中花费的时间。
- **"实时因子"**：*模拟时间/计算时间* 的商。如果用 360 秒模拟一小时，则实时因子为 10。
- **UPS**：（每秒更新）。平均每秒计算时间执行的车辆更新次数。如果单个车辆更新平均需要 1 毫秒，则这将为 1000。

如果路由发生在仿真中，每个路由算法实例将报告

- **路由查询次数**
- **为找到最佳路线而检查的边数**
- **路由总花费时间和每次路由调用的平均时间**

聚合流量测量

当设置选项 **--duration-log.statistics** 时，（快捷方式 **-t**）自动启用详细输出（除非明确设置为 *false*）并且将打印所有车辆行程的以下平均值：

- **RouteLength**：平均路线长度

- 速度：平均行程速度
- 持续时间：平均行程持续时间
- WaitingTime：平均站立时间（不由自主地）
- TimeLoss：由于驾驶速度低于预期而损失的平均时间（包括 WaitingTime）。所需速度考虑了车辆的 [速度因子](#)。
- DepartDelay：由于道路空间不足导致车辆出发延迟的平均时间

笔记

默认情况下，这些统计数据中仅包含已到达的车辆。如果选项`--tripinfo-output.write-unfinished`被设置，正在运行的车辆也将包括在内。

- DepartDelayWaiting：模拟结束时由于道路空间不足而无法插入的车辆的平均等待时间。（仅在设置选项`--tripinfo-output.write-unfinished`时出现）

如果模拟包含行人（行走的人），将添加以下输出：

- `<walk> walks`：输入中不同元素的数量
- RouteLength：平均步行长度
- 持续时间：平均步行持续时间
- TimeLoss：由于以低于最大速度行走或停止而损失的平均时间

如果模拟包含乘客（乘坐车辆的人），将添加以下输出：

- `<ride>` 游乐设施：输入中不同元素的数量
- WaitingTime：等待搭车的平均时间
- RouteLength：平均骑行长度
- 时长：平均骑行时长
- 乘坐公共汽车：乘坐公共交通工具在道路上行驶的次数（公共交通通过设置 `line` 属性来标识）。
- 乘坐公共交通工具在轨道上行驶的次数
- *Bike rides*：车辆类自行车的骑行次数
- 中止的游乐设施：由于没有合适的车辆可用而无法完成的游乐设施

您还可以查看[统计输出](#)以获得整个模拟的更全面的统计数据，包括上面提到的那些以及与安全、乘坐和运输相关的其他数据。