# ODD-based Query-time Scenario Mutation Framework for Autonomous Driving Scenario databases

Yun Tang[1*], Dhanush Raj[1], Xingyu Zhao, Xizhe Zhang, Antonio A. Bruto da Costa, Siddartha Khastgir, Paul Jennings

***Abstract*— Large-scale scenario databases have been set up containing hundreds of thousands of traffic scenarios for the verification and validation (V&V) of autonomous vehicles (AV). Scenarios in the database are often labelled with semantic Operational Design Domain (ODD) tags (e.g., *WeatherRainy*, *RoadTypeHighway* and *ActorTypeTruck*) to be queried via exact tag matching. Such a scenario database design has two major limitations, i.e. combinatorial scenario generation inevitably leads to many redundant scenarios, and each ODD query matches only a small number of scenarios in the database (0.2% in our case study), rendering most of the database wealth wasted. We propose a novel scenario database design and an ODD-based query-time scenario mutation framework to address the limitations. Our case study results show that the proposed framework has the potential to fully utilize all the database scenarios at query time while eliminating scenario redundancy in the database (in our case study, given the same ODD query, the number of final matched scenarios increased by 36 times, diversity increased by 99 times, and scenario database utilization rate increased from 0.2% to 36%).**

## I. INTRODUCTION

**Background** The verification and validation (V&V) approach of autonomous vehicles has shifted from distance-based [1]–[3] to scenario-based in academia and industry. Many scenario-generation methods have been proposed toward different objectives, such as diversity coverage [4]–[7], criticality [8]–[10] and risk estimation [11]–[13]. To enable efficient organization, exchange and reuse of the generated scenarios, scenario databases [14], such as Safety Pool$^{TM}$ [15] and KITTI [16], have been established and published. Fig. 1 illustrates the existing scenario database design and the scenario query mechanism. In the scenario generation phase, combinatorial sampling is often adopted to maximize the diversity coverage of the scenario parameters (e.g., shape, direction and colour in Fig. 1). Such scenario generation methods inevitably result in many similar scenarios in the database and do not scale as the number of scenario parameters increases. In the scenario query phase, given the system-under-test's Operational Design Domain (an ODD example is shown in Fig. 3), users query the scenario database with a set of ODD tags (e.g., *ColourOrange* & *ShapeT* & *DirectionDownward*) and the database returns scenarios that perfectly match the ODD query tags. Such

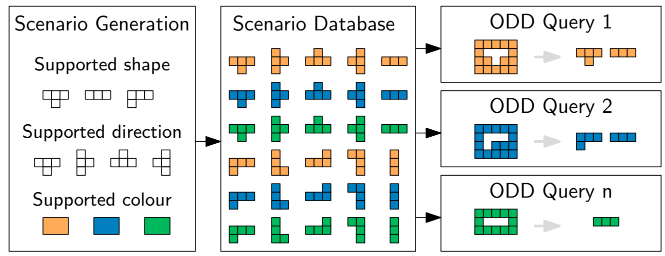



Fig. 1: Illustration of the existing scenario database design and query mechanism with unnecessary redundancy and low query-time scenario utilization rate (e.g., $2/30 < 7\%$)

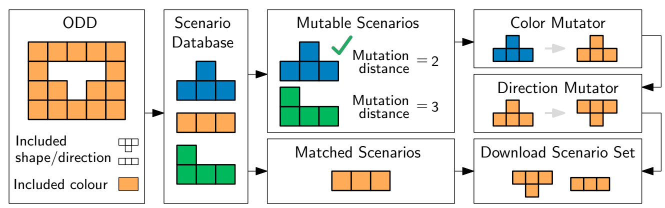



Fig. 2: Illustration of our scenario database design and mutation framework where unmatched scenarios are ranked, mutated and selected to match the ODD query input. There are no redundant scenarios in the database and query time scenario utilization rate = $2/3 \approx 67\%$.

an exact tag-matching mechanism will only match a small portion of the scenarios from the entire database. From the database user's point of view, the abandoned majority of the scenarios in the database during the query may still be very "close" to the system's ODD, containing highly relevant and diverse scenario dynamics useful for testing purposes. Thus, increasing the scenario utilization ratio at query time is highly beneficial.

**Our Contributions** To address the aforementioned limitations, we propose a novel scenario database design combining generic default types and mutation tags to minimize the scenario redundancy and a scenario mutation framework to maximize the database utilization rate at query time (illustrated in Fig. 2). Specifically, in the scenario generation phase, scenarios are assigned only generic attribute values (e.g., actor type being *Vehicle* instead of *Sedan*, *Truck*, or *Wheelchair*) and are labelled with additional mutation tags (e.g., *ActorMutableTruck*, *WeatherMutableRainy*) indicating valid semantic mutations; in the scenario query phase, we rank, mutate and select those highly relevant, diverse but unmatched scenarios according to the ODD query input such that the wealth of the scenario database can potentially be fully exploited.

1. Yun Tang and Dhanush Raj contributed equally.
All authors are with Warwick Manufacturing Group, University of Warwick, Coventry, United Kingdom. Email: {dhanush.raj, yun.tang, xingyu.zhao, jason.zhang, antonio.bruto-da-costa, s.khastgir.1, paul.jennings}@warwick.ac.uk
* Corresponding author.

**Taxonomy**: ISO/DIS 34503 Road Vehicles: Test scenarios for automated driving systems: Taxonomy for operational design domain
**Base state**: Permissive
**Extension**: None
**#Composition statements**

| | |
|---|---|
| Included | drivable area type is [motorway] |
| Included | number of lanes is [2,–] |
| Included | lane dimension is [3.7,–] |
| Included | direction of travel is [right hand travel] |
| Included | drivable area surface type is [asphalt, concrete] |
| Excluded | weather is [snowfall] |
| Included | intersections are [y junctions] |
| Excluded | roundabouts are [all] |
| Excluded | Actor types are [Animals, VRUs, Non–motor vehicles] |
| c1 Conditional | horizontal plane is [curved roads] |
| ... | |

**#Conditional statements**

| | |
|---|---|
| c1 Excluded | radius of curved road is [0, 500 m] |

Fig. 3: ODD specification example for motorway-only ADS.

To summarize, our contributions are the following:

- We propose a novel scenario database design combining generic attribute values and mutation tags to eliminate the scenario database redundancy.
- We propose a novel query-time scenario mutation framework, including heuristic scenario ranking and selection criteria to maximize scenario diversity while maintaining reasonable relevancy during mutation.
- We implement the proposed mutation framework and report our case study with Safety Pool$^{TM}$ scenario database updated with our design. Results show that the mutation framework fulfils its designed objectives.

## II. METHODOLOGY

### A. Scenario Database Design

Fig. 4 presents a diagram of the proposed scenario database design and the query-time scenario mutation framework. The entire scenario engineering workflow can be described as follows:

**Generation** The test specification (e.g., system-under-test description, test objective and performance evaluation criteria) is first defined. Then, scenario-generation methods can be applied to generate functional and logical [17] scenarios, commonly toward higher diversity coverage. Generic parameter values are used to minimize database redundancy. For example, the generic vehicle type *Vehicle* is considered to cover all the detailed vehicle types, e.g., *Van*, *Truck*, *Sedan*.

**Storage** The generated scenarios are labelled with applicable semantic tags (e.g., Fig. 5c) when saved to the database. The ODD tags describe nominal semantic features, such as *DriveOnRight* (scenery-related), *WalkToward* (dynamic-related) and *SunInFront* (environment-related), while mutation tags denote mutation potentials, such as *SubjectMutableTruck* which means the subject (or ego) vehicle type can be mutated to *Truck* in the scenario. Each ODD attribute, e.g., *Weather*, can have one or more ODD tags, e.g., *WeatherRainy*, *WeatherSnowy* and *WeatherWindy*, and each ODD tag has its corresponding mutation tag, e.g., *WeatherRainy* and *WeatherMutableRainy* where they are mutually exclusive.

**Query** Scenarios are queried by matching ODD and mutation tags. For example, the ODD query in Fig. 3 matches not only all the right-hand-drive highway scenarios but also the left-hand-drive ones as they are, by default, labelled with the "DirectionMutableRight" tag. Eventually, all the matched scenarios and mutable scenarios are returned for user selection. The terms "unmatched scenarios" and "mutable scenarios" are used interchangeably.

**Mutation** Theoretically, any scenario can be mutated to match any given ODD query. However, different mutable scenarios require different mutation steps and types, i.e., some may require only one mutation, e.g., traffic direction mirroring, while others may require multiple mutations in the scenery, dynamics and environment elements. Intuitively, the "further" (less relevant) a scenario is from the ODD, the more mutation steps it usually demands to match the ODD. Thus, we propose a novel pre-mutation distance criterion based on which the mutation candidates are ranked. Meanwhile, to eliminate redundancy after mutation, we propose a heuristic post-mutation diversity criterion to select only those different from the download scenarios set. The download set thus contains the initially matched scenarios and mutated scenarios added after each selection step.

**Execution** The downloaded scenario can then be used for critical concrete scenarios sampling for system evaluation.

The scenario generation, storage, query and execution methods have been discussed extensively in the literature and are out of the scope. The following section focuses on the proposed query-time scenario mutation framework.

### B. ODD-based Scenario Mutation Framework

Given the ODD query input (e.g., Fig. 3) and a set of unmatched scenarios, the tasks of the scenario mutation engine are to 1) rank unmatched scenarios according to their pre-mutation distance and short-list the relevant mutation candidates; 2) mutate the selected candidates such that they match the ODD specification, and 3) iteratively select diverse mutated scenarios and add to the download set.

Assume we have a complete set of ODD attributes $\mathbb{A}$, a complete set of ODD tags $\mathbb{T}$, a function that returns an ODD tag's corresponding ODD attribute $attribute : \mathbb{T} \to \mathbb{A}$, the ODD query tag set $\mathbb{T}_{ODD} \subseteq \mathbb{T}$, the corresponding ODD query attribute set $\mathbb{A}_{ODD} \subseteq \mathbb{A} = \cup_{T \in \mathbb{T}_{ODD}} \{attribute(T)\}$, the download scenario set initialized by matched scenarios $\mathbb{S}_D$, the mutable scenario set initialized by unmatched scenarios $\mathbb{S}_M$, the set of mutated scenarios $\mathbb{S}'_M$, a function that takes a scenario $s$ as input and outputs the ODD tags $tags : \mathbb{S} \to \mathbb{T}$ where $s \mapsto \{T_s | T_s \in \mathbb{T}\}$, a function that mutates a scenario to match the ODD $mutate : \mathbb{S}_M \to \mathbb{S}'_M$ where $s \mapsto s'$ and $tags(s') \subseteq \mathbb{T}_{ODD}$, then the query-time scenario mutation process can be described as follows.

**Pre-mutation Scenario Ranking** We propose a mutation candidate ranking metric $distance : \mathbb{S}_M \to \{0, 1, ..., |\mathbb{A}_{ODD}|\}$ for each mutable scenario $s_m \in \mathbb{S}_M$ which equals to the number of unmatched ODD attributes, i.e.,

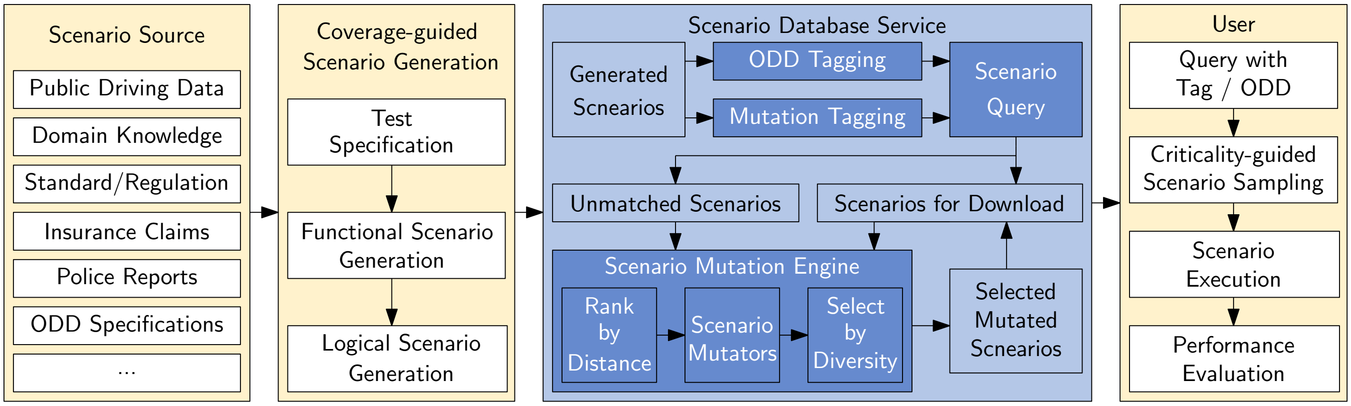



Fig. 4: Scenario database overview including generation, storage, query and mutation (Contributions are highlighted in blue).

$$distance(s_m) = |\{attribute(T) \in \mathbb{A}_{ODD} | T \in tags(s_m) \setminus \mathbb{T}_{ODD}\}| \tag{1}$$

Note that the distance is calculated assuming the ODD attributes are independent of each other, and its value remains constant if the ODD query is fixed.

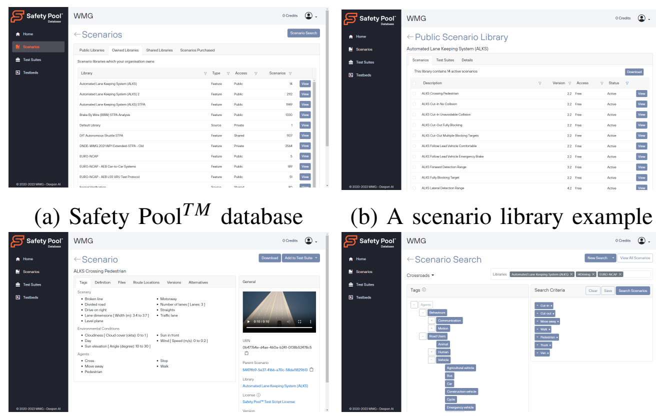
**ODD Attribute-based Mutators** Each ODD attribute has a corresponding mutator performing two main functions: 1) to analyse and label the scenario with applicable mutation tag, e.g., "WeatherMutableRainy", and 2) to perform the actual mutation such that the mutated scenario matches the "included" ODD attribute value. The top-ranked (e.g., mutation distance $\leq$ a threshold) is short-listed to pass through a series of mutators for each unmatched ODD attribute. For some attributes, mutation can incur too much deviation from the original scenario context, and those scenarios will be marked unsuitable for mutation. Note that being unsuitable for mutation is different from being immutable (without mutation tags), where the former still allows mutation while the latter renders the mutated scenario invalid (e.g., Fig. 9c).

**Post-mutation Scenario Selection** To avoid adding duplicated scenarios to the download set, the mutated scenarios are filtered by a heuristic criterion, $diversity: \mathbb{S}'_M \rightarrow \{0, 1, ..., |\mathbb{T}|\}$, representing the marginal diversity gain if the mutated scenario were to be added to the download set, which equals to the minimum difference regarding ODD tags between the mutated scenario $s'_m = mutate(s_m)$ and the entire download set, i.e.,

$$diversity(s'_m) = \min_{s_d \in \mathbb{S}_D} |(\mathbb{T}_{s'_m} \cup \mathbb{T}_{s_d})) \setminus (\mathbb{T}_{s'_m} \cap \mathbb{T}_{s_d})| \tag{2}$$

where $\mathbb{T}_{s'_m} = tags(s'_m)$ and $\mathbb{T}_{s_d} = tags(s_d)$. Only scenarios of diversity $\geq$ a threshold are selected. The default threshold is set to 1 to maximise the database utilisation rate.

Fig. 2 illustrates a flowchart for the scenario mutation framework. The blue T-tetromino ranks higher than the green L-tetromino and is selected for colour and direction mutation to match the ODD. Note that the orange straight-tetromino is also considered a match. For example, if the ODD "includes" "wet" in the induced road surface conditions, then scenarios whose road surfaces are dry will also match the ODD by default, although "dry" is not one of the options for the induced road surface condition ODD attribute.



(a) Safety Pool$^{TM}$ database (b) A scenario library example

(c) Semantic tags of scenarios (d) Scenario query using tags

Fig. 5: Safety Pool$^{TM}$ scenario database [15]

## III. SAFETY POOL$^{TM}$ CASE STUDY

To demonstrate the effectiveness of the scenario mutation framework, we conduct a case study on the Safety Pool$^{TM}$.

### A. Introduction to Safety Pool

Unlike other scenario databases (e.g., KITTI [16]) containing segmented scenarios and raw sensor data recorded on public roads, Safety Pool$^{TM}$ contains nearly a quarter million scenarios formatted in the scenario description language SDL [18] (An example of a jaywalker scenario in SDL is presented in Fig. 6). In Safety Pool$^{TM}$, scenarios are organized using libraries (Fig. 5a) for different testing purposes (Fig. 5b). Safety Pool$^{TM}$ assigns semantic tags (Fig. 5c) following the ASAM OpenLABEL [19] schema and allows users to query scenarios via semantic tag matching (Fig. 5d). Note that the limitations discussed in Section I apply to any scenario database implementing equivalent tag-matching mechanisms and are not limited to Safety Pool$^{TM}$.

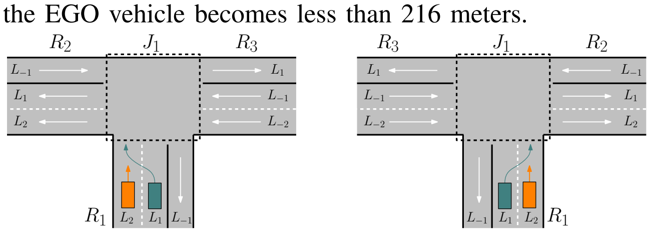
There are in total 46948 scenarios from public libraries as private scenarios are only accessible by owners.

### B. Mutator Implementation

The implementation complexity for mutators varies with attributes. For some, e.g., *time of day* and *road surface condition*, the mutator implementation is straightforward,

```
VERSION: 8.1 EXTENSION: None
AUTHOR: 'WMG, Intelligent Vehicles - V&V Team'
SCENERY ELEMENTS:
DO:      Map -   highway network [Highway1] as: Junctions: None
Roads:   R1:     START Road type [Motorway] as [R1] ...
                 Number of lanes [3] as [R1.L1, R1.L2, R1.L3] ...
                 Length [9000 to 11000] AND Lane width [3.4 to 3.7] ...
DYNAMIC ELEMENTS:
INITIAL: Vehicle [Ego] in [R1.L2] AND Pedestrian [P1] in [R1.L3]
         with a [Longitudinal] offset of [490 to 510]
AND      at relative position [FSR] with relative heading angle [85 to 95] to [Ego]
AND      Global timer [T1] = [0]
WHEN:    [Ego] is [Going_Ahead] DO:
[P1]     PHASE 1: [Stop][-, 0 to 0, 0 to 0] [Ego: -17 to -16, FSR]
         WHILE:   [P1] [Longitudinal] offset to [Ego]>[216]
         PHASE 2: [Walk_Cross] [-, 1.3 to 1.4, 0 to 1] [Ego: -16 to -15, F]
         PHASE 3: [Walk_Away][-, 1.3 to 1.4, -0.5 to 0.5] [Ego: -16 to -15, FSL]
END:     [T1] == [40]
ENVIRONMENT ELEMENTS: DO: [Env1] Wind [0 to 0.2] Cloudiness [0 to 1] ...
```

Fig. 6: A simplified jaywalking scenario formatted in SDL, where a pedestrian ahead starts to jaywalk across EGO's path (road $R1$ lane $L2$) from the right lane (road $R1$ lane $L3$) when the longitudinal distance between the pedestrian and the EGO vehicle becomes less than 216 meters.




(a) (b)

Fig. 7: (a) Original left-hand scenario. (b) Mutated right-hand scenario. Note the change of relative manoeuvre from "cut-in from right side" to "cut-in from left side" after mutation.

affecting only a few scenario elements. For others, the mutation could affect the entire scenario depending on the implementation design. In the following, we discuss in detail our implementations of three most-demanded mutators by Safety Pool$^{TM}$ users, i.e., *Traffic Direction Mutator*, *Actor Type Mutator* and *Background Traffic Mutator*. In SDL, road users are referred to as actors, which are used interchangeably in the sequel. Due to limited space, only the mutation function is presented, and the tagging function is omitted.

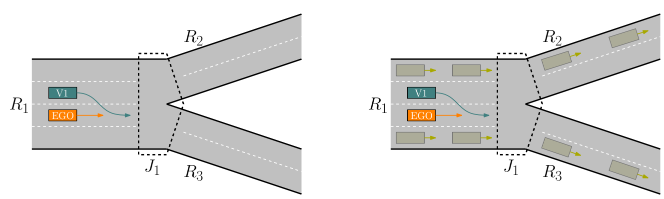
**Traffic Direction Mutator** Most countries/regions adopt the right-hand traffic system while less than 30% of the countries/regions drive on the left side [20]. In Safety Pool$^{TM}$, most scenarios are left-handed, thus not directly applicable to most countries/regions. To best preserve the original scenario's nature, we design the traffic direction mutator (Algorithm 1) that mirrors the entire scenario (Fig. 7) instead of simply toggling the road elements' travel directions.

**Background Traffic Mutator** In the Safety Pool™ database, most scenarios generated from standards/regulations do not have background traffic and only contain one or two non-ego actors. Introducing background traffic increases both scenario fidelity and complexity. We implement the background traffic mutator (Algorithm 2) to introduce background a traffic element on each suitable route passing through junctions as shown in Fig. 8.

**Actor Type Mutator** In most of the Safety Pool™ scenarios, the vehicle actors are of the generic type *Vehicle*, which is by default translated into "Car" in the OpenSCENARIO format. To support fine-grained types of vehicles (e.g., Van, Truck, Trailer) and humans (e.g., Pedestrian, Cyclist), we implement the actor type mutator (Algorithm 3) such that it explores all the valid actor type combinations. Fig. 9 illustrates the valid and invalid actor-type mutations.

**Algorithm 1:** function mutation_traffic_direction()

```
Data: SDL scenario input
Result: mutated SDL with opposite traffic direction
/* mutate scenery component */
for each road do
    mirror road direction, curvature and bank angle;
end
for each junction do
    mirror the angles between connected road pairs;
end
/* mutate dynamic component */
mirror the relative location information for each actor
  in SDL's initialisation phase;
for each manoeuvre sequence do
    for each when condition do
        mirror the EGO's manoeuvre;
    end
    for each serial manoeuvre sequence do
        for each manoeuvre phase do
            mirror the phase type description, relative
              agent parameters and when conditions;
        end
    end
end
/* mutate environment component */
mirror the light position;
```




(a) (b)

Fig. 8: (a) Original scenario without traffic. (b) Mutated scenario with two background traffic elements by the background traffic mutator.

## C. Scenario Query and Mutation

We take the ODD query specified in Fig. 3 for example. Based on the query results (Table II), it can be seen that Safety Pool$^{TM}$ indeed has more left-hand travel than right-hand travel scenarios. Both ODDs only match a small fraction (original ODD: $111/46948 \approx 0.2\%$, slightly modified ODD: $6456/46948 \approx 13.8\%$) of the entire dataset. The number of scenarios against the mutation distance given an ODD adheres to the Central Limit Theorem (i.e., approximates a normal distribution). This can guide the configuration of the

**Algorithm 2:** function mutation_background_traffic()

```
Data: SDL scenario input, background traffic density
Result: mutated SDL with background traffic of
        specified density
for each actor do
    gather the road and lane ID from the actor's
      initialisation information;
end
for each junction do
    collect all the incoming-outgoing lane pairs
      connected to this junction and select those
      empty lane pairs without actors;
end
for each empty lane pair do
    instantiate a background traffic element of the
      given traffic density atop the lane pair where the
      incoming lane is the source and the outgoing
      lane is the sink;
end
```

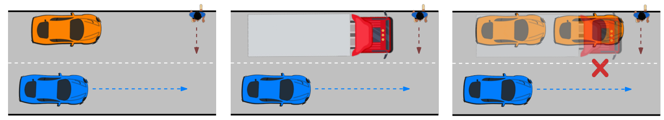
**Algorithm 3:** function mutation_actor_type()

```
Data: SDL scenario input, (optional) target actor
      type combination
Result: mutated SDL scenarios of given/all actor
        type combination(s)
recursively resolve all the actors' positions and map
  their positions onto road s-l axes;
if target actor type combination is given then
    if the mutation is valid without any two actors
      overlapping each other, using their positions
      and dimensions mapped on the shared axes then
        mutate and save the scenario;
        return;
    end
end
for each possible actor type combination do
    mutate and save the scenario if the combination
      is valid;
end
```

mutation distance threshold in case we prefer scenarios that are not "too far" from the ODD, e.g., we can mutate scenarios of mutation distance $\leq 2$ only. Table I details the unmatched ODD attributes for the ODD query input.

To quantify the diversity gain of the download set, we cluster the scenarios by comparing their ODD tags. Scenarios with the same set of tags are grouped into one cluster.

Table III lists the final query-time mutation results. It can be seen that: **1)** the final download set contains many more scenarios (in our case $4040/111 \approx 36$ times) compared to the matched scenario set, i.e., 111 scenarios; **2)** each mutated scenario that's selected and added to the download set increases the diversity (by adding one unique tag cluster) and the diversity (if quantified by the number of unique clusters) has increased $3969/40 \approx 99$ times; **3)** we have in



(a) (b) (c)

Fig. 9: (a) Original scenario with a parked car and jaywalker. (b) The mutated scenario with a parked truck and the same jaywalker by actor-type mutator. (c) The scenario with two yellow cars parked closely is invalid for truck-type mutation as overlap will occur.

total eliminated 13005 duplicate scenarios and marked 29903 scenarios unsuitable for mutation given the example ODD, which ensures the uniqueness and validity of the download set; and 4) the scenario utilisation rate increases from 0.2% ($\approx 111/46948$) to 36% ($\approx (46948-29903)/46948$).

Table IV lists the experiment results for modified ODD with left-hand travel. Although the modified ODD has matched many more scenarios initially compared to the original ODD, we can still see an improved download set where the number of scenarios increased by $(8786-6456)/6456 \times 100\% \approx 36\%$, the diversity increased by $(3965-1635)/1635 \times 100\% \approx 143\%$ and the scenario database utilisation rate increased from 13.8% ($\approx 6456/46948$) to 36% ($\approx (46948-29903)/46948$).

## IV. DISCUSSION

**Mutator Generalizability** Although the mutators are implemented and validated on our SDL language, the algorithms of each ODD attribute mutator apply to different scenario description languages, e.g., scenic 2.0 [21] and OpenSCENARIO [22]. This is because the scenario description languages share many design philosophies (for on-road scenarios only, as SDL has yet to support off-road scenarios), including but not limited to 1) the traffic networks are described using basic elements such as lane, road and intersections, 2) actors of the scenarios can specify absolute and relative positions, 3) environment attributes are common (e.g., rain, illumination, wind, snow and fog), and 4) dynamics of the scenario are driven by time or position-related events. Readers may notice that the algorithms are presented in natural language without SDL-specific symbols to reflect the generalizability.

**Mutation Complexity** Assume we have $N$ scenarios, $c$ ODD attributes, and the ODD matches zero scenarios, the rank phase complexity is $\mathcal{O}(cN)$, the mutation phase is also $\mathcal{O}(cN)$, and in the selection phase with all the $N$ scenarios added to the download set one by one, the complexity is $\mathcal{O}(1)+\mathcal{O}(2)+...+\mathcal{O}(N-1) = \mathcal{O}(N^2)$ due to the re-calculation of complexity value. In our experiment (ODD in Fig. 3 against 46948 public scenarios), the entire process took around 20 minutes ($\approx$ 39 scenarios per second). Although there is potential room for performance improvements, e.g., by exploring distributed mutation, the query time is not a critical issue as the ODDs are less prone to change, and the duration is trivial compared to the simulation time.

TABLE I: Unmatched ODD attributes for ODD in Fig.3.

| Mutation Distance | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Unmatched ODD attribute and no. scenarios | - | direction of travel: 6456<br>horizontal plane: 96<br>Actor types: 35<br>lane dimension: 1 | direction of travel: 17600<br>roundabouts: 6283<br>intersections: 2665<br>number of lanes: 2560<br>transverse plane: 1613<br>weather: 1605<br>drivable area type: 1536<br>induced surface conditions: 1416<br>special structures: 144<br>Actor types: 78<br>horizontal plane: 26<br>temporary road structures: 14 | direction of travel: 14995<br>roundabouts: 6630<br>weather: 5655<br>induced surface conditions: 4617<br>drivable area type: 4339<br>transverse plane: 3426<br>intersections: 2987<br>number of lanes: 2242<br>special structures: 364<br>Actor types: 360<br>horizontal plane: 184<br>lane dimension: 88<br>lane type: 82 | direction of travel: 5341<br>induced surface conditions: 3500<br>weather: 3123<br>drivable area type: 2746<br>roundabouts: 2678<br>transverse plane: 2096<br>intersections: 1285<br>number of lanes: 967<br>special structures: 414<br>horizontal plane: 366<br>lane dimension: 309<br>Actor types: 174<br>lane type: 25 | direction of travel: 1002<br>induced surface conditions: 975<br>weather: 914<br>transverse plane: 812<br>drivable area type: 765<br>roundabouts: 500<br>horizontal plane: 318<br>special structures: 283<br>Actor types: 275<br>intersections: 229<br>number of lanes: 203<br>lane dimension: 149 | transverse plane: 115<br>induced surface conditions: 115<br>weather: 115<br>direction of travel: 63<br>drivable area type: 63<br>Actor types: 52<br>horizontal plane: 52<br>special structures: 52<br>roundabouts: 41<br>number of lanes: 12<br>intersections: 10 |
| Total | direction of travel: 45457<br>number of lanes: 5984 | roundabouts: 16132<br>special structures: 1257 | weather: 11412<br>horizontal plane: 1042 | induced surface conditions: 10623<br>Actor types: 974 | drivable area type: 9449<br>lane dimension: 547 | transverse plane: 8062<br>lane type: 107 | intersections: 7176<br>temporary road structures: 14 |

TABLE II: ODD query result on the public scenarios.

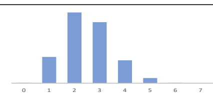
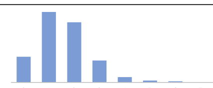
| ODD Query | Mutation Distance (No. unmatched ODD Attributes) | | | | Total | Bar Chart |
|---|---|---|---|---|---|---|
| Fig.1 | **0** | **1** | **2** | **3** | 46948 |  |
| | 111 | 6588 | 17770 | 15323 | | |
| | **4** | **5** | **6** | **7** | | |
| | 5756 | 1285 | 115 | 0 | | |
| Fig.1 + *left-hand travel* | **0** | **1** | **2** | **3** | 46948 |  |
| | 6456 | 17711 | 15127 | 5511 | | |
| | **4** | **5** | **6** | **7** | | |
| | 1330 | 478 | 283 | 52 | | |

TABLE III: ODD query-time mutation experiment results. Total means accumulated, and delta means per distance.

| **Mutation Distance** | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|
| Total Download | 111 | 1776 | 2874 | 3543 | 3972 | 4040 | 4040 |
| Total Mutation | 0 | 1665 | 2763 | 3432 | 3861 | 3929 | 3929 |
| Delta Mutation | 0 | 1665 | 1098 | 669 | 429 | 68 | 0 |
| Total Clusters | 40 | 1705 | 2803 | 3472 | 3901 | 3969 | 3969 |
| Delta Clusters | 40 | 1665 | 1098 | 669 | 429 | 68 | 0 |
| Total Duplicate | 0 | 4888 | 9974 | 12444 | 12874 | 13005 | 13005 |
| Total Unsuitable | 0 | 35 | 11621 | 23805 | 28702 | 29788 | 29903 |

## V. RELATED WORKS

*Scenario mutation* is not new for simulation-based V&V of autonomous vehicles. To effectively handle a large number of scenario parameters, researchers have widely adopted Genetic Algorithms [9], [23], where crossover and mutations of parameter values are performed to sample critical concrete scenarios. However, such parameter value mutations are different from our framework; primarily, (**objective**) ours aims to fully utilize the scenario database capacity without causing unnecessary scenario redundancy, while others aim to find critical parameter value combinations; (**runtime**) ours functions at query time, while others require prolonged simulations to guide the mutation direction.

TABLE IV: Modified ODD (left-hand travel) query-time mutation experiment results

| **Mutation Distance** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|---|
| Total Download | 6456 | 7550 | 8126 | 8479 | 8589 | 8722 | 8786 | 8786 |
| Total Mutation | 0 | 1094 | 1670 | 2023 | 2133 | 2266 | 2330 | 2330 |
| Delta Mutation | 0 | 1094 | 576 | 353 | 110 | 133 | 64 | 0 |
| Total Clusters | 1635 | 2729 | 3305 | 3658 | 3768 | 3901 | 3965 | 3965 |
| Delta Clusters | 1635 | 1094 | 576 | 353 | 110 | 133 | 64 | 0 |
| Total Duplicate | 0 | 5057 | 7429 | 7743 | 7928 | 8131 | 8259 | 8259 |
| Total Unsuitable | 0 | 11560 | 23739 | 28583 | 29618 | 29760 | 29851 | 29903 |

Despite the difference, however, the existing mutation methods and ours can work jointly where ours is applied first to initialize the logical scenario set at query time to maximize diversity and other mutation methods can then be applied to sample and mutate the concrete scenarios parameters at simulation time to maximize criticality as shown in Fig. 4.

Another related work, SceGene [24], performs scenario crossover and mutation offline, where it manipulates the motion trajectories of scenario actors to formulate diverse and complex traffic dynamics at the given road networks. Since the current ODD design [25], [26] mainly focuses on the road network and the environment conditions and has few attributes on the traffic dynamics (i.e., actor manoeuvres) besides the type of actors, our framework, as a result, has limited scenario dynamics mutators. Similarly, ours and SceGene augment each other. They can also work jointly where ours is applied first to query scenarios with maximized road network and environment diversity. Then, SceGene is applied later to search diverse traffic dynamics on our mutated road networks within the ODD definition.

## VI. CONCLUSION

In this work, we propose a novel scenario database design combining the generic logical scenario generation with mutation tags to eliminate the scenario database redundancy and an ODD-based scenario mutation framework consisting of novel ODD attribute-based mutators, scenario ranking and selection criteria, to maximize the scenario database utilization rate at query time. We implement the proposed mutation framework and conduct a case study on the scenario database Safety Pool$^{TM}$ with an example highway ODD query input, and results show that the mutation framework effectively fulfils its design objectives.

## ACKNOWLEDGEMENT

The work presented in this paper has been supported by UKRI Future Leaders Fellowship (Grant MR/S035176/1), Department of Transport, UK, and Transport Canada. The authors would like to thank the WMG center of HVM Catapult, and WMG, University of Warwick, UK for providing the necessary infrastructure for conducting this study. No new data were created in this study.

## References


[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182 – 193, 2016.

[2] X. Zhao, V. Robu, D. Flynn, K. Salako, and L. Strigini, "Assessing the safety and reliability of autonomous vehicles from road testing," in *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. Berlin, Germany: IEEE, 2019, pp. 13–23.

[3] X. Zhao, K. Salako, L. Strigini, V. Robu, and D. Flynn, "Assessing safety-critical systems from operational testing: A study on autonomous vehicles," *Information and Software Technology*, vol. 128, p. 106393, 2020.

[4] Y. Zhou, G. Lin, Y. Tang, K. Yang, W. Jing, P. Zhang, J. Chen, L. Gong, and Y. Liu, "Flyover: A model-driven method to generate diverse highway interchanges for autonomous vehicle testing," *arXiv preprint arXiv:2301.12738*, 2023.

[5] Y. Zhou, Y. Sun, Y. Tang, Y. Chen, J. Sun, C. M. Poskitt, Y. Liu, and Z. Yang, "Specification-based autonomous driving system testing," *IEEE Transactions on Software Engineering*, 2023.

[6] Y. Tang, Y. Zhou, F. Wu, Y. Liu, J. Sun, W. Huang, and G. Wang, "Route coverage testing for autonomous vehicles via map modeling," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 450–11 456.

[7] Y. Tang, "A map-based model-driven testing framework for automated driving systems," 2022.

[8] Y. Tang, Y. Zhou, Y. Liu, J. Sun, and G. Wang, "Collision avoidance testing for autonomous driving systems on complete maps," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 179–185.

[9] Y. Tang, Y. Zhou, T. Zhang, F. Wu, Y. Liu, and G. Wang, "Systematic testing of autonomous driving systems using map topology-based scenario classification," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 1342–1346.

[10] X. Zhang, Y. Kwan Mo, E. Chodowiec, Y. Tang, M. D. Higgins, S. Khastgir, P. Jennings *et al.*, "A novel scenario-based testing approach for cooperative-automated driving systems," *IEEE SMC (systems, man, and cybernetics) 2023*, 2023.

[11] Z. Huang, H. Lam, D. J. LeBlanc, and D. Zhao, "Accelerated evaluation of automated vehicles using piecewise mixture models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2845–2855, 2017.

[12] S. Zhang, H. Peng, D. Zhao, and H. E. Tseng, "Accelerated evaluation of autonomous vehicles in the lane change scenario based on subset simulation technique," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3935–3940.

[13] M. Arief, Z. Cen, Z. Liu, Z. Huang, B. Li, H. Lam, and D. Zhao, "Certifiable evaluation for autonomous vehicle perception systems using deep importance sampling (deep is)," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1736–1742.

[14] H. Ren, H. Gao, H. Chen, and G. Liu, "A survey of autonomous driving scenarios and scenario databases," in *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*. IEEE, 2022, pp. 754–762.

[15] W. U. of Warwick, "Safety pool - powered by deepen ai and wmg university of warwick," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.safetypool.ai/

[16] T. T. I. a. C. Karlsruhe Institute of Technology, "The kitti vision benchmark suite," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.cvlibs.net/datasets/kitti/

[17] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1821–1827.

[18] X. Zhang, S. Khastgir, and P. Jennings, "Scenario description language for automated driving systems: a two level abstraction approach," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 973–980.

[19] ASAM, "Asam openlabel," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.asam.net/standards/detail/openlabel/

[20] Wikipedia, "Left- and right-hand traffic - wikipedia," 2023, accessed on Sep 03, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Left-_and_right-hand_traffic

[21] D. J. Fremont, E. Kim, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A language for scenario specification and data generation," *Machine Learning*, pp. 1–45, 2022.

[22] ASAM, "Asam openscenario," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.asam.net/standards/detail/openscenario/

[23] Y. Huai, S. Almanee, Y. Chen, X. Wu, Q. A. Chen, and J. Garcia, "scenorita: Generating diverse, fully-mutable, test scenarios for autonomous vehicle planning," *IEEE Transactions on Software Engineering*, pp. 1–21, 2023.

[24] A. Li, S. Chen, L. Sun, N. Zheng, M. Tomizuka, and W. Zhan, "Scegene: Bio-inspired traffic scenario generation for autonomous driving testing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 859–14 874, 2022.

[25] ISO, "Iso 34503:2023 road vehicles — test scenarios for automated driving systems — specification for operational design domain," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.iso.org/standard/78952.html

[26] ASAM, "Asam openodd," 2023, accessed on Aug 27, 2023. [Online]. Available: https://www.asam.net/standards/detail/openodd/