

ScenaGen

A CP Model for Grounding Qualitative Driving Scenarios

Nassim Belmecheri Arnaud Gotlieb Nadjib Lazaar Helge Spieker
CP 2026

simula

université
PARIS-SACLAY

 Microsoft

1. Scenario-generation hierarchy and problem setting
2. QXG scenario representation
3. Grounding QXGs with ScenaGen
4. Experimental protocol and results
5. Usage demonstration and conclusion

Part 1

Scenario-generation hierarchy and problem setting

1. **Scenario-generation hierarchy and problem setting**
2. QXG scenario representation
3. Grounding QXGs with SCENAGEN
4. Experimental protocol and results
5. Usage demonstration and conclusion

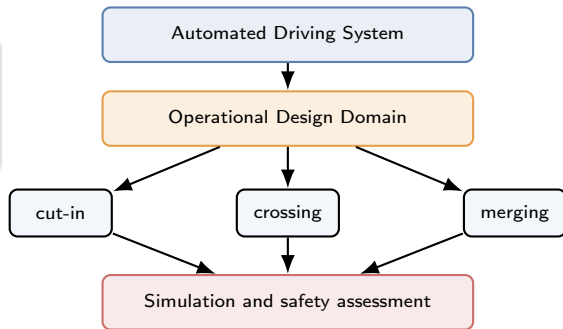
Automated driving: why scenario-based validation matters

Automated Driving Systems (ADS)

ADS combine perception, prediction, planning, and control to operate vehicles under a specified Operational Design Domain.

- ▶ Validation cannot rely only on public-road driving: rare critical interactions are difficult to observe and reproduce.
- ▶ Scenario-based testing gives repeatable and controllable evaluation conditions.
- ▶ The central requirement is to generate many meaningful scenarios with precise executable coordinates.

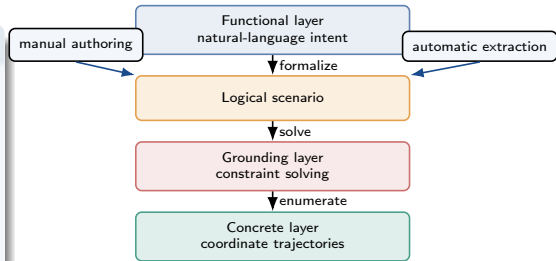
References: Ulbrich et al. 2015; Menzel et al. 2018; PEGASUS; ASAM OpenSCENARIO; ISO 34502.



Scenario-generation hierarchy: from intent to executable tests

Scenario abstraction levels

- ▶ **Functional scenario:** informal traffic intent, e.g., “a car cuts in close to ego”.
- ▶ **Logical scenario:** symbolic structure, parameter ranges, relative positions, distances, and motion categories.
- ▶ **Concrete scenario:** fixed coordinates, object sizes, and trajectories ready for simulation.



Grounding objective

The goal is to transform a logical scenario description into one or more concrete scenario instantiations that can be simulated.

The logical scenario layer can be represented as a QXG

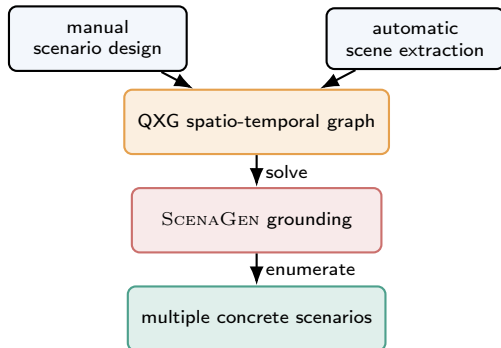
Qualitative Explainable Graph

A QXG is a spatio-temporal qualitative graph for the logical scenario layer.

- ▶ **Nodes:** detected or user-defined traffic objects.
- ▶ **Edges:** qualitative relations between object pairs.
- ▶ **Time:** each edge stores a relation sequence across frames.

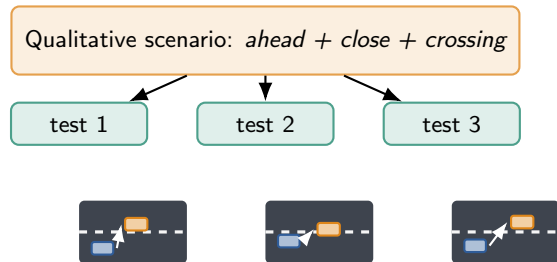
Two ways to obtain a QXG

It can be manually specified by a scenario designer, or automatically constructed from detected objects, tracks, and bounding boxes.



From one logical scenario to many concrete tests

- ▶ A qualitative statement does not describe a single coordinate assignment.
- ▶ It defines a **set of admissible configurations**.
- ▶ Enumerating solutions gives scenario diversity without manually designing every test.



Testing benefit

Whether the QXG is manually specified or extracted automatically, SCENAGEN solves the same grounding model and can enumerate multiple quantitative variations of the same qualitative scenario.

Part 2

QXG scenario representation

1. Scenario-generation hierarchy and problem setting

2. QXG scenario representation

3. Grounding QXGs with SCENAGEN

4. Experimental protocol and results

5. Usage demonstration and conclusion

QXG construction: from detections to a qualitative graph

1. Scene frame



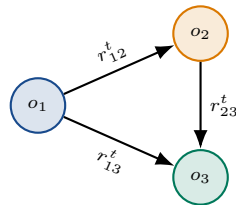
2. Detected objects

o_1 : ego
box $(x_1^{min}, x_1^{max}, y_1^{min}, y_1^{max})$

o_2 : car
box $(x_2^{min}, x_2^{max}, y_2^{min}, y_2^{max})$

o_3 : VRU
box $(x_3^{min}, x_3^{max}, y_3^{min}, y_3^{max})$

3. Pairwise graph



QXG in one sentence

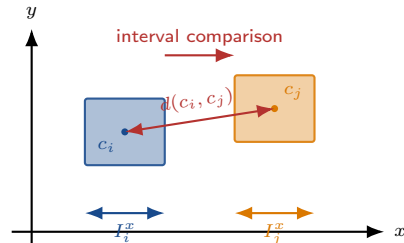
A QXG is built by turning object detections and tracks into a graph whose edges describe relative position, center-to-center distance, and motion through time.

Reference: QXG construction and qualitative scene representation - Belmecheri, Gotlieb, Lazaar, and Spieker, 2023/2024.

How edge relations are computed from bounding boxes

For each ordered pair (o_i, o_j) at frame t

- 1 **RA**: compare bbox intervals on x using Allen relations.
- 2 **RA**: compare bbox intervals on y using Allen relations.
- 3 **QDC**: compute the center-to-center distance bucket.
- 4 **QV**: compute the center displacement bucket through time.



Example edge:

RA: before on x , overlaps on y ;
QDC: close center distance; QV: slow motion.

Edge label

$$r_{ij}^t = ((r_x, r_y)_{RA}, q_{QDC}, v_{QV})$$

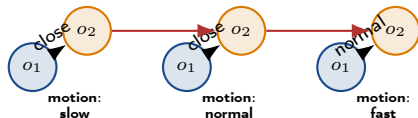
Example: $(([b, o]), c, sl)$.

Through time: a QXG stores relation sequences

- ▶ A driving scenario is a sequence of frames.
- ▶ Tracking preserves object identities through time.
- ▶ For each ordered pair (o_i, o_j) , the graph stores:

$$R_{ij} = \langle r_{ij}^1, r_{ij}^2, \dots, r_{ij}^T \rangle.$$

- ▶ Each r_{ij}^t contains RA relative position, QDC center-distance bucket, and QV motion bucket.



Part 3

Grounding QXGs with ScenaGen

1. Scenario-generation hierarchy and problem setting
2. QXG scenario representation
- 3. Grounding QXGs with ScenaGen**
4. Experimental protocol and results
5. Usage demonstration and conclusion

Formal grounding problem

Input

A qualitative scenario over T frames and n traffic entities:

$$\mathcal{R} = \langle \mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T \rangle, \quad \mathcal{O} = \{o_1, \dots, o_n\}.$$

Output

A concrete scenario assigning integer coordinates to every object at every frame:

$$(c_{i,t}^x, c_{i,t}^y) \in \mathbb{Z}^2 \quad \text{for all } i \in [1, n], \quad t \in [1, T].$$

Correctness condition

For every frame and every ordered pair of objects, the computed quantitative boxes must reproduce the qualitative relation stored in the QXG.

Decision variables

For object o_i at frame t :

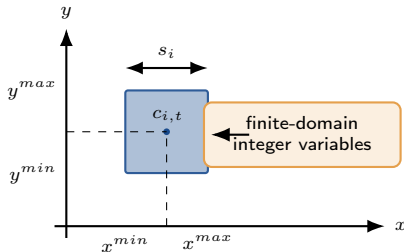
$$(c_{i,t}^x, c_{i,t}^y), \quad x_{i,t}^{min}, x_{i,t}^{max}, y_{i,t}^{min}, y_{i,t}^{max}$$

plus object size s_i .

$$\#vars = 6nT + n$$

Constraint families

The model combines local geometric definitions, semantic size ordering, pairwise spatial relations, center-distance categories, temporal motion bounds, and an ego-centered reference frame.



Constraint family I: geometry and object dimensions

Purpose

Define a finite-domain geometric object that the solver can reason about.

Constraints

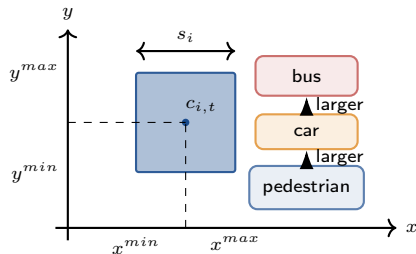
For each object o_i and frame t :

$$x_{i,t}^{max} - x_{i,t}^{min} = s_i, \quad y_{i,t}^{max} - y_{i,t}^{min} = s_i$$

$$2c_{i,t}^x = x_{i,t}^{min} + x_{i,t}^{max}, \quad 2c_{i,t}^y = y_{i,t}^{min} + y_{i,t}^{max}.$$

Example

If $s_i = 4$ and $c_{i,t}^x = 10$, then the box can be encoded by $x^{min} = 8$ and $x^{max} = 12$.



Constraint family II: relative position between objects

Purpose

Translate qualitative statements such as “left of”, “overlapping”, or “before” into linear interval constraints.

Interval comparison

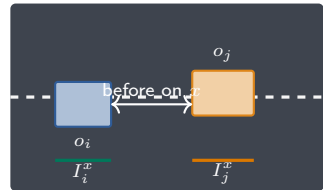
Each box induces one interval on each axis:

$$I_{i,t}^x = [x_{i,t}^{min}, x_{i,t}^{max}], \quad I_{i,t}^y = [y_{i,t}^{min}, y_{i,t}^{max}].$$

A qualitative position is encoded by constraints over these intervals.

Example

“ o_i is before o_j on the x -axis” becomes $x_{i,t}^{max} < x_{j,t}^{min}$.
Overlap on the y -axis is encoded by the corresponding interval-overlap inequalities.



Qualitative relation $\rightarrow$ finite-domain constraints

Constraint family III: center-distance categories

Purpose

Encode proximity requirements independently of exact metric coordinates.

Center-distance constraint

For each pair (o_i, o_j) :

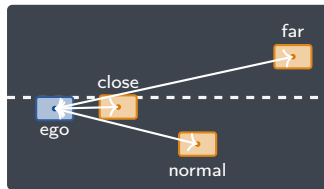
$$d_{ij,t}^2 = (c_{i,t}^x - c_{j,t}^x)^2 + (c_{i,t}^y - c_{j,t}^y)^2.$$

The distance is then assigned to one category:

very close, close, normal, far, very far.

Example

With squared thresholds $4 < 16 < 64 < 144$, $d_{ij,t}^2 = 10$ means the objects are *close*.



Distance is computed between bounding-box centers

Constraint family IV: temporal motion and reference frame

Purpose

Ensure that consecutive frames form a plausible trajectory rather than independent snapshots.

Motion bucket constraint

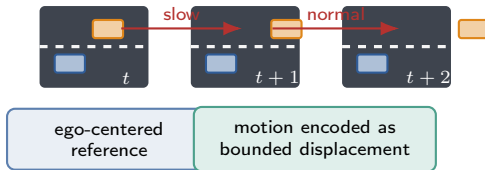
For object o_i between frames t and $t + 1$:

$$|c_{i,t+1}^x - c_{i,t}^x| + |c_{i,t+1}^y - c_{i,t}^y| \leq \sigma(v_{i,t}).$$

The qualitative bucket $v_{i,t}$ can be: not moving, slow, normal, or fast.

Example

If $v_{i,t}$ is *slow* and $\sigma(\text{slow}) = 3$, then the Manhattan displacement between the two centers cannot exceed 3.



Part 4

Experimental protocol and results

1. Scenario-generation hierarchy and problem setting
2. QXG scenario representation
3. Grounding QXGs with SCENAGEN
- 4. Experimental protocol and results**
5. Usage demonstration and conclusion

Experimental setup: implementation and instance generation

Implementation and solver

- ▶ SCENAGEN implemented in **MiniZinc 2.8**.
- ▶ Preliminary backend comparison: Gecode 6.3, Chuffed 0.13, and **CP-SAT** from OR-Tools 9.8.
- ▶ Reported experiments use **CP-SAT**, 8 parallel workers, and free search.
- ▶ Machine: Intel Core i9-13900K CPU, 64GB RAM.
- ▶ Timeout: **3600s** per instance.

Synthetic benchmark grid

- ▶ Objects: $n \in \{10, 20, 30, 40, 60, 80, 100\}$.
- ▶ Frames: $T \in \{10, 20, 30, 40, 60, 80, 100\}$.
- ▶ Relations sampled uniformly over object pairs and frames.
- ▶ Balanced RA, QDC, and QV relation types.
- ▶ Each configuration averaged over **10 seeds**.

Validity criterion

A grounding is valid iff all qualitative constraints from the input QXG are satisfied by the generated coordinates.

Experimental setup: parameters and real-world datasets

Model parameters

- ▶ Integer map half-size: $M = 500$.
- ▶ Grid: 1000×1000 integer coordinate space.
- ▶ Velocity thresholds:
 $\sigma(\text{slow}) = 5, \quad \sigma(\text{normal}) = 15, \quad \sigma(\text{fast}) = 30$.
- ▶ QDC thresholds calibrated to the grid scale.

GA baseline note

A genetic algorithm baseline was explored, but feasible solutions were found only for the smallest setting ($n = 10, T = 10$) within the one-hour timeout.

Real-world reconstruction

- ▶ **nuScenes**: 850 scenes, average 19 objects, average 39 frames.
- ▶ **PandaSet**: 76 scenes, average 50 objects, average 273 frames.
- ▶ Object trajectories are converted to QXGs by the builder.
- ▶ QXGs are given to SCENAGEN without manual simplification.

Measured quantity

The reported metric is mean solving time in seconds; real-world experiments also report successful qualitative reconstruction.

Experimental density: what does d measure?

Constraint density in the QXG

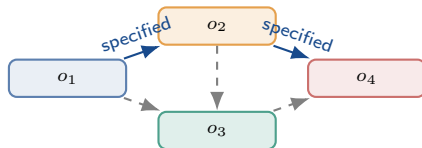
For n objects and T frames, the maximum number of ordered pair/frame relation tuples is:

$$N_{\max} = T n(n - 1).$$

The experimental density is:

$$d = \frac{N_{\text{specified}}}{T n(n - 1)}.$$

Here, $N_{\text{specified}}$ is the number of object-pair/frame relation tuples explicitly present in the QXG.



one low-density frame

Experimental regimes

- ▶ $d = 100\%$: all pair/frame relations are included.
- ▶ $d \approx 3\text{--}5\%$: only salient relations are specified.
- ▶ Refinement increases d by adding relations.

Clarification

Density means **how much of the QXG is constrained**; it is not traffic density or vehicles per meter.

Research questions from the paper

RQ1 — Scalability

How does solving time scale with the number of objects n and the temporal horizon T ?

- ▶ Object count controls the number of pairwise interactions.
- ▶ The temporal horizon adds repeated frame-level constraints.

RQ2 — Constraint density

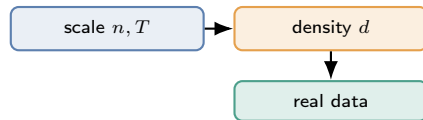
How does the fraction of specified object-pair/frame relation tuples affect solver performance?

- ▶ Density controls how complete the QXG relation set is.
- ▶ Higher density gives a more constrained grounding problem.

RQ3 — Real-world reconstruction

Can SCENAGEN reconstruct realistic traffic scenes from large-scale autonomous-driving datasets within tractable computation times?

- ▶ QXGs are automatically constructed from recorded detections and tracks.
- ▶ The generated scenario is checked under qualitative equivalence.

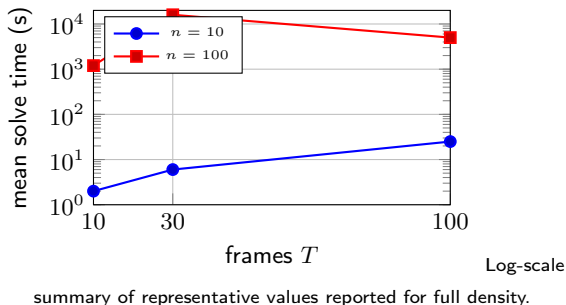


RQ1: scalability under full constraint density

Protocol

Full density: every pairwise qualitative relation is specified for every frame ($d = 100\%$).

- ▶ Object count is the dominant factor.
- ▶ At fixed $T = 10$: $n = 10$ takes about **2s**, while $n = 100$ takes about **1200s**.
- ▶ This is a **600×** increase.
- ▶ Frames have a secondary, near-linear effect.
- ▶ Hardest full-density regime appears around $T \geq 30$ and $n \geq 80$.



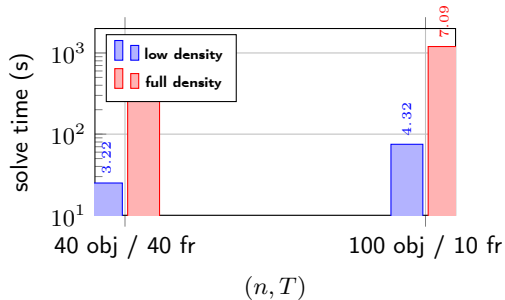
RQ2: effect of constraint density

Protocol

Compare dense and sparse QXGs on the same (n, T) grid and seeds:

$$d = 100\% \quad \text{vs.} \quad d \approx 3\text{--}5\%.$$

- ▶ Sparse QXGs contain only a selected subset of salient object-pair/frame relations.
- ▶ $(n = 40, T = 40)$: 300s $\rightarrow$ 25s.
- ▶ $(n = 100, T = 10)$: 1200s $\rightarrow$ 75s.



Answer to RQ2

Lower density gives **10–16 $\times$** speedups at matched (n, T) while preserving qualitative validity.

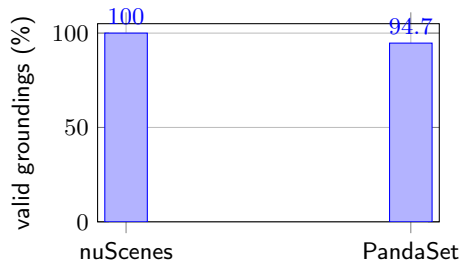
RQ3: reconstruction from real-world datasets

nuScenes

- ▶ 850 scenes.
- ▶ Average: 19 objects, 39 frames.
- ▶ Temporal horizon concentrated around 38–40 frames.
- ▶ Success: **850/850** valid groundings.

PandaSet

- ▶ 76 scenes.
- ▶ Average: 50 objects, 273 frames.
- ▶ Up to 210 objects and 275 frames.
- ▶ Success: **72/76** valid groundings.



Answer to RQ3

The CP model reconstructs real-world QXGs without manual simplification; PandaSet is harder due to longer horizons and denser object populations.

Part 5

Usage demonstration and conclusion

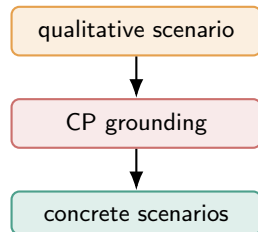
1. Scenario-generation hierarchy and problem setting
2. QXG scenario representation
3. Grounding QXGs with SCENAGEN
4. Experimental protocol and results
- 5. Usage demonstration and conclusion**

Contribution

SCENAGEN formulates qualitative-to-quantitative ADS scenario grounding as a CP problem.

- ▶ Interpretable qualitative graphs become concrete trajectories.
- ▶ Spatial, distance, size, and temporal constraints are handled jointly.
- ▶ One qualitative scenario can enumerate multiple concrete tests.
- ▶ Results show effective grounding on synthetic and real-world benchmarks.

Central conclusion: CP provides a deterministic and interpretable grounding engine for scenario-based ADS testing.



Thank you
Questions

Email: nbelmecheri@microsoft.com

Paper: doi.org/10.4230/LIPIcs.CP.2026.4

Code: github.com/NassimBelmecheri/ScenaGen

Scenario representation

- ▶ Ulbrich et al. 2015: scene, situation, and scenario terminology.
- ▶ Menzel et al. 2018: functional, logical, and concrete scenario layers.
- ▶ Belmecheri et al. 2023/2024: QXG for qualitative scene understanding.

Qualitative reasoning

- ▶ Allen 1983: interval relations.
- ▶ Renz and Nebel 2007; Dylla et al. 2017: qualitative spatial calculi.

Constraint programming and implementation

- ▶ Nethercote et al. 2007: MiniZinc modeling language.
- ▶ ScenaGen paper: CP 2026, LIPIcs Vol. 379, Article 4.

Benchmarks

- ▶ nuScenes: real-world multimodal driving dataset.
- ▶ PandaSet: real-world autonomous-driving dataset.