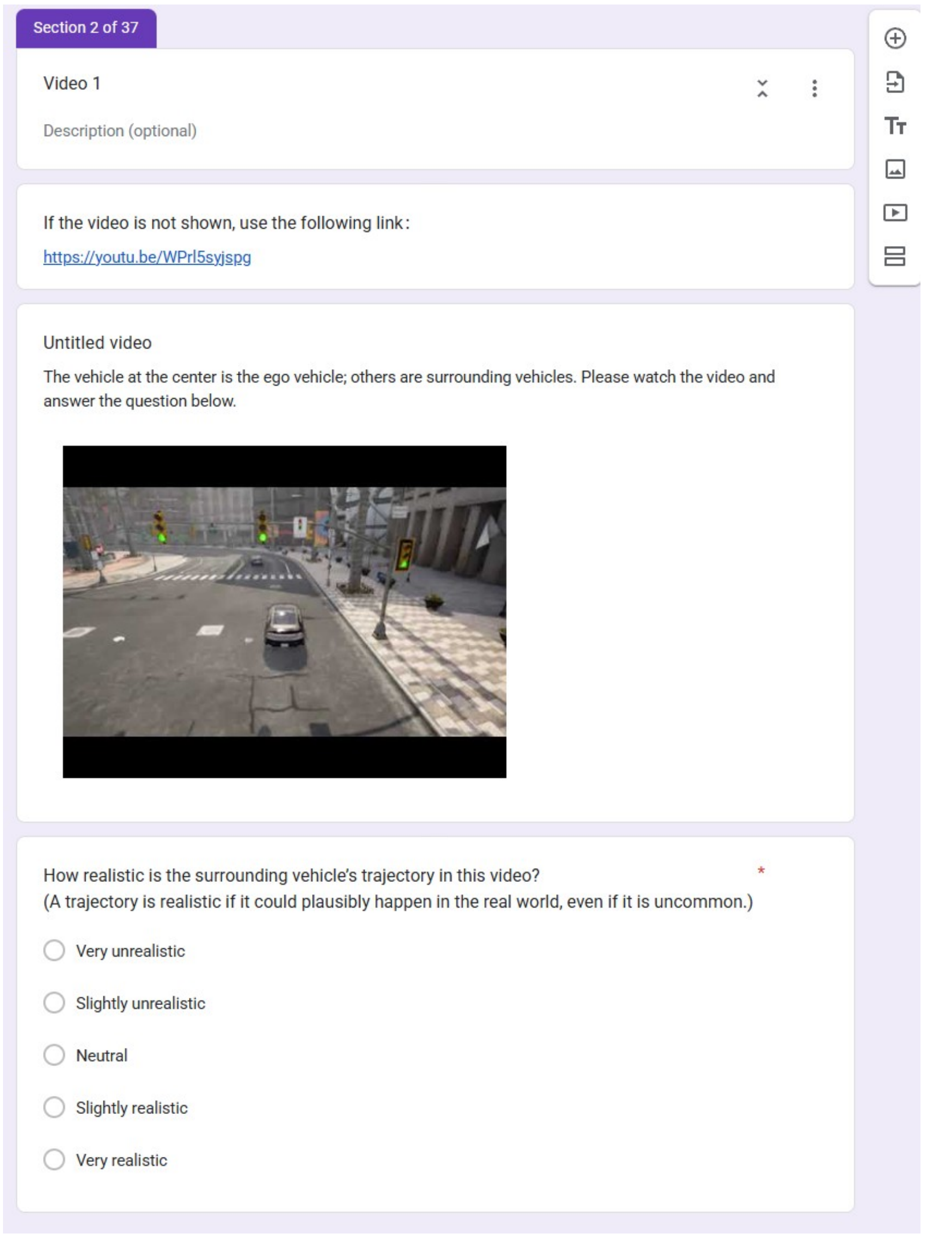
SANDWICH Supplementary MATERIAL

1 Neural Network Setting

Table 1: Neural Network Setting

Component	**Parameter**	**Value**
MARL training	Neural network layers	2
	Neurons per layer	128
	Learning rate	0.001
	Initial noise	0.75
	Noise decay	0.999995
	Minimum noise	0.01
	Terminate reward (**I**)	10

2 User Study Interface



Section 2 of 37

Video 1

Description (optional)

If the video is not shown, use the following link:
https://youtu.be/WPrl5syjspg

Untitled video

The vehicle at the center is the ego vehicle; others are surrounding vehicles. Please watch the video and answer the question below.

How realistic is the surrounding vehicle's trajectory in this video? *
(A trajectory is realistic if it could plausibly happen in the real world, even if it is uncommon.)

- ◯ Very unrealistic
- ◯ Slightly unrealistic
- ◯ Neutral
- ◯ Slightly realistic
- ◯ Very realistic

Figure 1: User Study Interface

3 Weighted Fleiss' Kappa Calculation

In our rating scheme, we acknowledge that larger rating discrepancies should correspond to greater disagreement. To account for this, we introduce the Weighted Fleiss' Kappa calculation. We first transfer our Likert scale rating category from text value to numerical value:

- *"Very unrealistic"* is assigned the value 1.
- *"Slightly unrealistic"* is assigned the value 2.
- *"Neutral"* is assigned the value 3.
- *"Slightly realistic"* is assigned the value 4.
- *"Very realistic"* is assigned the value 5.

Definition of the Weight Matrix

Since ratings are ordinal, we apply a quadratic weight function to penalize large discrepancies more than small ones. The weight matrix w_{ij} is defined as:

$$w_{ij} = 1 - \frac{(i-j)^2}{(k-1)^2} \tag{1}$$

where:

- i, j are rating categories $(1, 2, \ldots, k)$.
- k is the total number of rating categories.
- The further apart the categories, the smaller the weight.

In our case, the $k = 5$, an example of the weight matrix is:

$$W = \begin{bmatrix} 1.00 & 0.75 & 0.44 & 0.19 & 0.00 \\ 0.75 & 1.00 & 0.75 & 0.44 & 0.19 \\ 0.44 & 0.75 & 1.00 & 0.75 & 0.44 \\ 0.19 & 0.44 & 0.75 & 1.00 & 0.75 \\ 0.00 & 0.19 & 0.44 & 0.75 & 1.00 \end{bmatrix} \tag{2}$$

Observed Agreement

For each subject (e.g., a video being rated), let f_{ij} be the number of raters assigning a rating of j. The total number of ratings for a given subject i is:

$$p_i = \sum_{j=1}^{k} f_{ij} \tag{3}$$

The observed agreement for subject i is calculated as:

$$P_i = \frac{\sum\limits_{j=1}^{k}\sum\limits_{l=1}^{k} w_{jl} \cdot f_{ij} \cdot f_{il}}{p_i(p_i - 1)} \tag{4}$$

where:

- f_{ij} is the count of raters giving category j to subject i.
- f_{il} represents the number of raters who assigned rating l to subject i.
- w_{jl} is the weight between rating category j and l.
- $p_i(p_i - 1)$ is the total number of rating pairs.
- If $p_i = 1$, then P_i is set to 0 (as no agreement is possible).

The overall observed agreement across all subjects is:

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{5}$$

where N is the total number of subjects.

Expected Agreement

The proportion of ratings assigned to each category across all subjects is:

$$p_j = \frac{\sum_{i=1}^{N} f_{ij}}{\sum_{i=1}^{N} p_i} \tag{6}$$

The expected agreement, assuming ratings were randomly assigned, is:

$$P_e = \sum_{j=1}^{k}\sum_{l=1}^{k} w_{jl} \cdot p_j \cdot p_l \tag{7}$$

Computation of Weighted Fleiss' Kappa

The final formula for the weighted Fleiss' Kappa is:

$$\kappa_w = \frac{\bar{P} - P_e}{1 - P_e} \tag{8}$$

where:

- $\bar{P}$ is the observed agreement.
- P_e is the expected agreement.
- If $\bar{P} = P_e$, then $\kappa_w = 0$, indicating no agreement beyond chance.

Interpretation of Weighted Fleiss' Kappa

The interpretation of κ_w is as follows:

- $0.00 - 0.20$: Slight agreement
- $0.21 - 0.40$: Fair agreement
- $0.41 - 0.60$: Moderate agreement
- $0.61 - 0.80$: Substantial agreement
- $0.81 - 1.00$: Almost perfect agreement

For example, if we compute:

$$\kappa_w \approx 0.714 \tag{9}$$

this suggests **substantial agreement** among raters.

4 Statistical Analysis

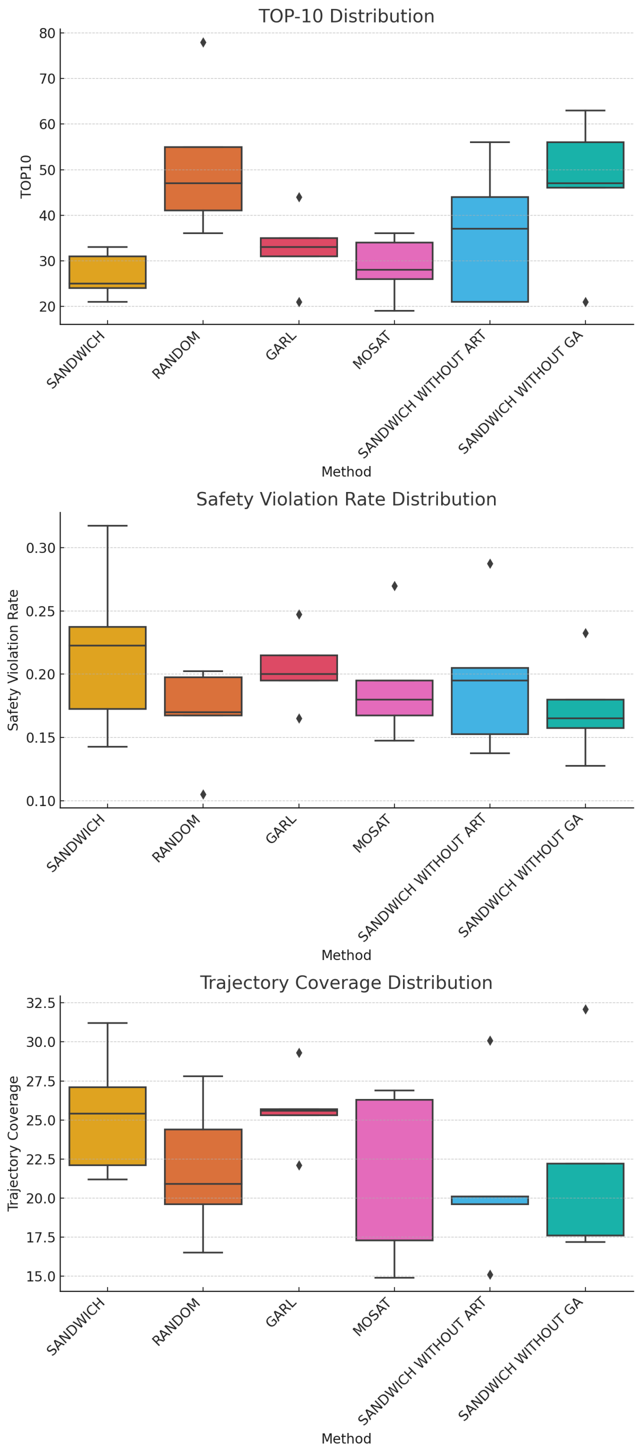


Figure 2: Statistical analysis across all baselines in Town01

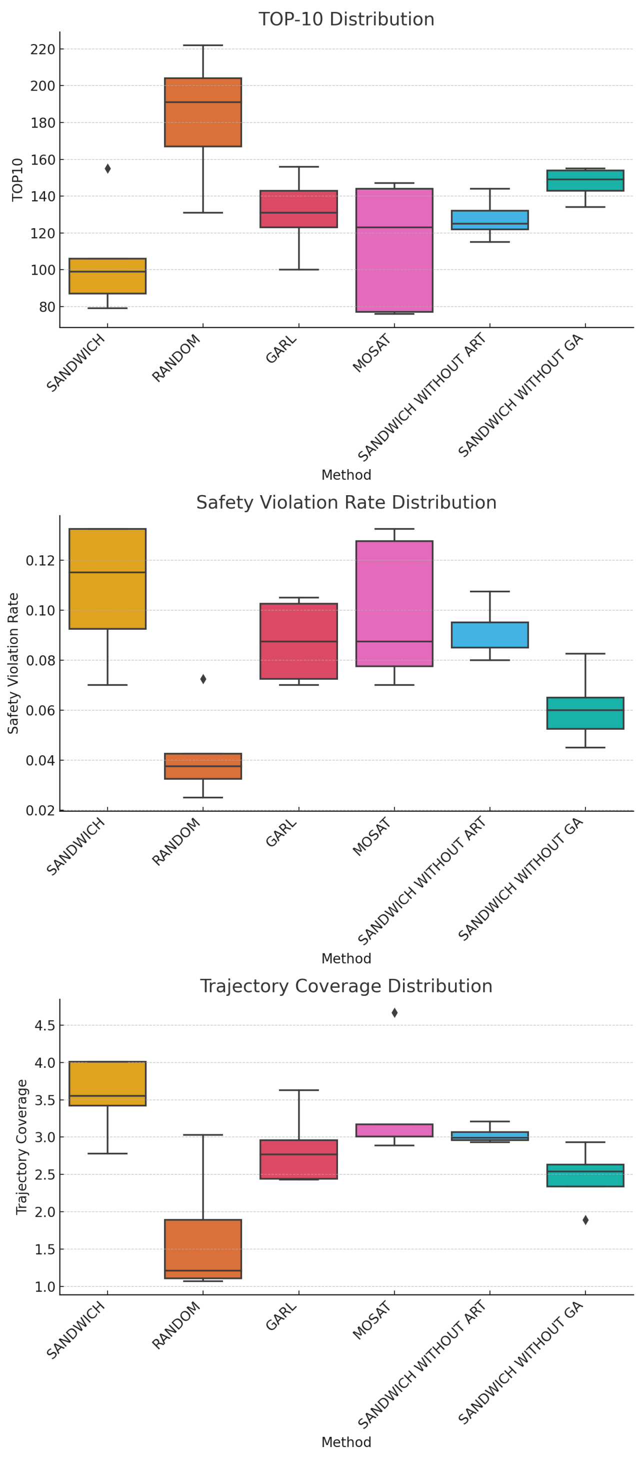


Figure 3: Statistical analysis across all baselines in Town04

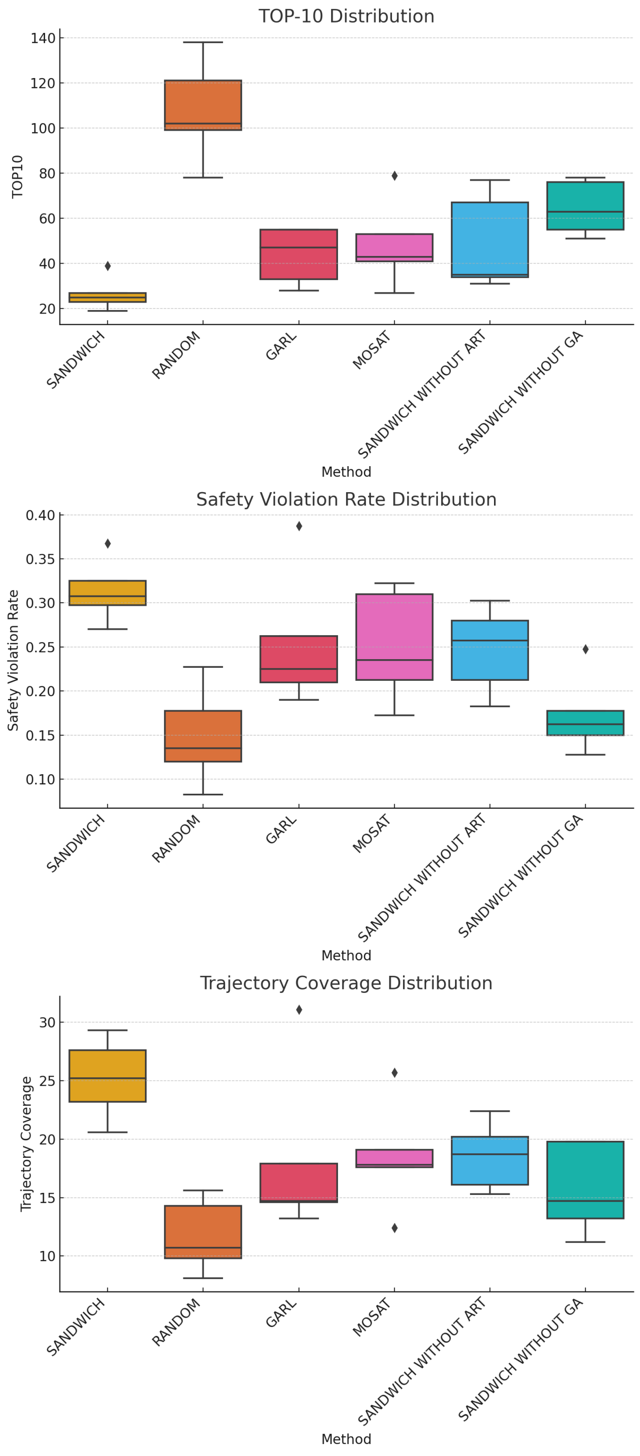


Figure 4: Statistical analysis across all baselines in Town10

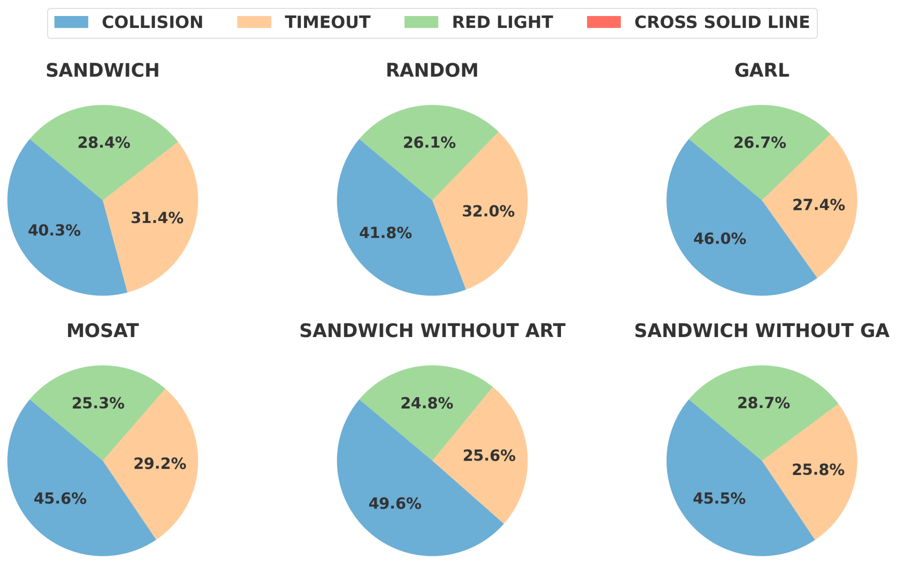


Figure 5: Safety violation type distribution across all baselines in Town01

5 Safety Violation Type Distribution

6 Baseline Implementation

6.1 GARL

For GARL, we follow a structure similar to GA, comprising both offline and online phases. In the offline GA phase, GARL uses the same approach as ours, except it excludes the ART-related objective in the fitness function.

In the online phase, GARL employs single-agent reinforcement learning. The state for each agent is defined as the relative position between a surrounding vehicle and the ego vehicle:

$$S_i = (P_{sur_i,x} - P_{ego,x},\ P_{sur_i,y} - P_{ego,y}) \tag{10}$$

Here, S_i represents the state of the i-th surrounding vehicle. The action space in GARL is identical to ours.

For the reward function, we adopt the same Stage 1: Tracking Reward (r_{track}) as in our method. This reward encourages the single agent to stay close to the ego vehicle. A large reward of 100 is given if a surrounding vehicle is within one lane width of the ego vehicle. Conversely, a large penalty of -100 is applied if the surrounding vehicle collides with the ego vehicle.

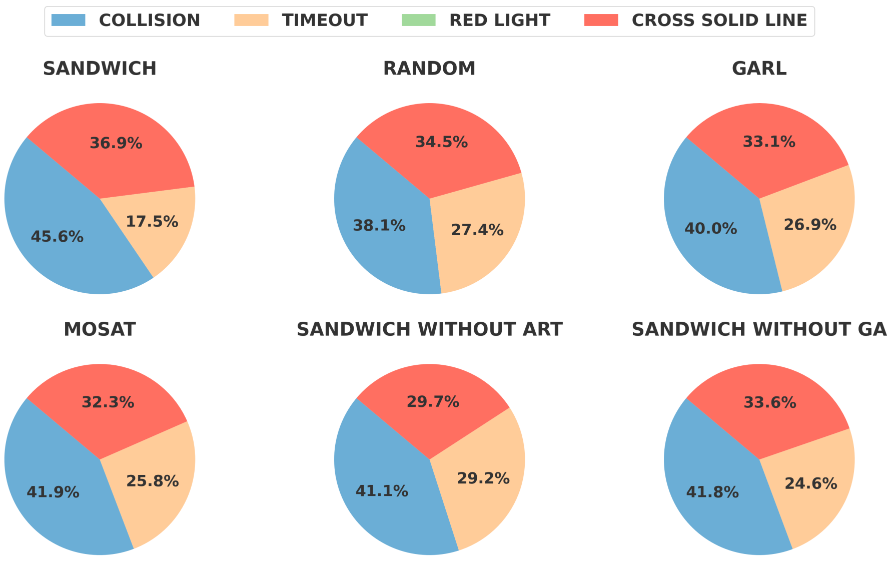


Figure 6: Safety violation type distribution across all baselines in Town04

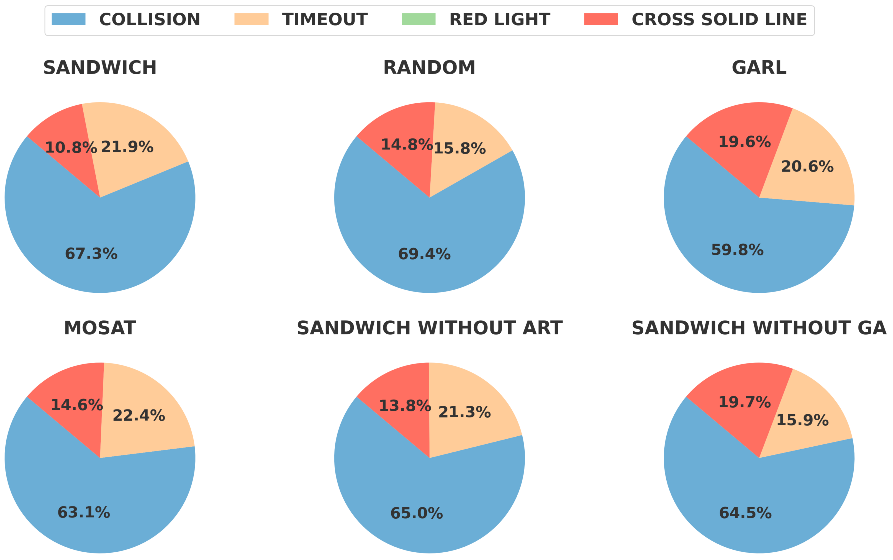


Figure 7: Safety violation type distribution across all baselines in Town10

6.2 MOSAT

For MOSAT, the GA used for initial configuration is exactly the same as ours, exclude the ART-related objective in the fitness function. In addition, MOSAT includes another GA specifically for action generation, referred to as the *action GA*. Since our test cases typically last between 30 seconds and 1 minute, the chromosome representation (Figure 8) for the action sequence consists of 60 actions, which is sufficient to cover the duration of the test case.

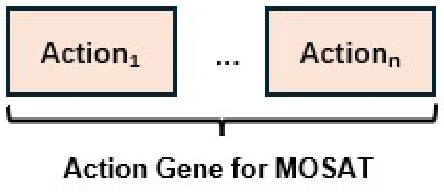


Figure 8: Chromosome representation of the action sequence in the Action GA.

The mutation and crossover settings for the action GA are identical to those used in the initial configuration GA. We also implement all action patterns and their corresponding trigger conditions, as these are central to the design of MOSAT.