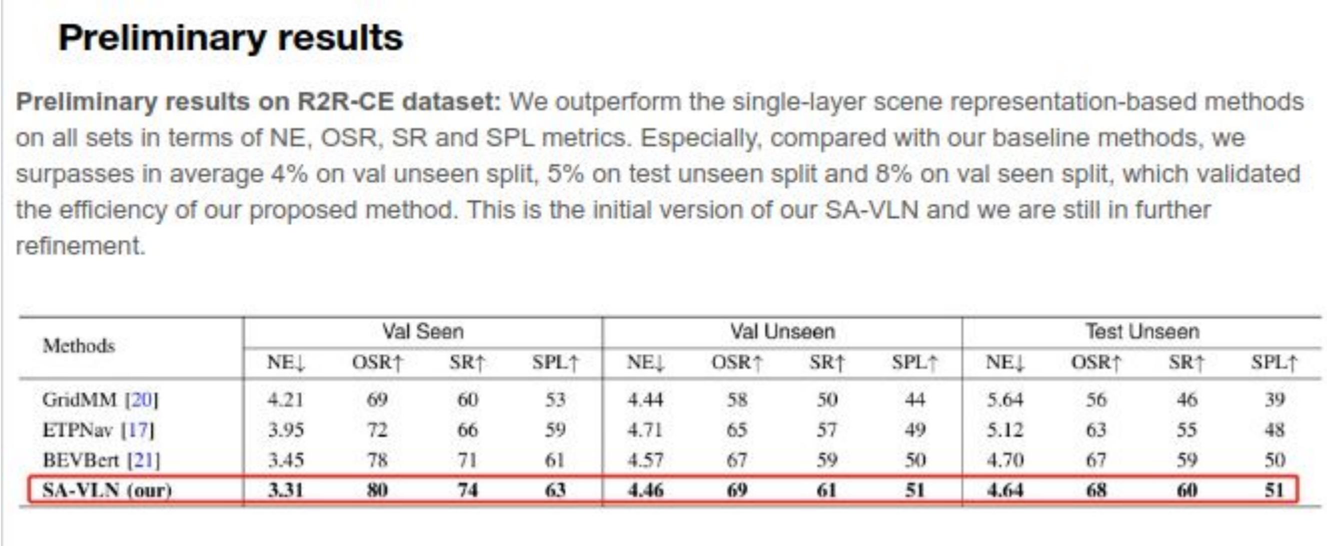
Improvement of intersection coordination

Final presentation
Team 2

Review: Spatial-Aware Vision-Language Navigation of Mobile Agents

- Vision-language-navigation
- Viewpoint-robust feature extraction



Preliminary results

Preliminary results on R2R-CE dataset: We outperform the single-layer scene representation-based methods on all sets in terms of NE, OSR, SR and SPL metrics. Especially, compared with our baseline methods, we surpasses in average 4% on val unseen split, 5% on test unseen split and 8% on val seen split, which validated the efficiency of our proposed method. This is the initial version of our SA-VLN and we are still in further refinement.

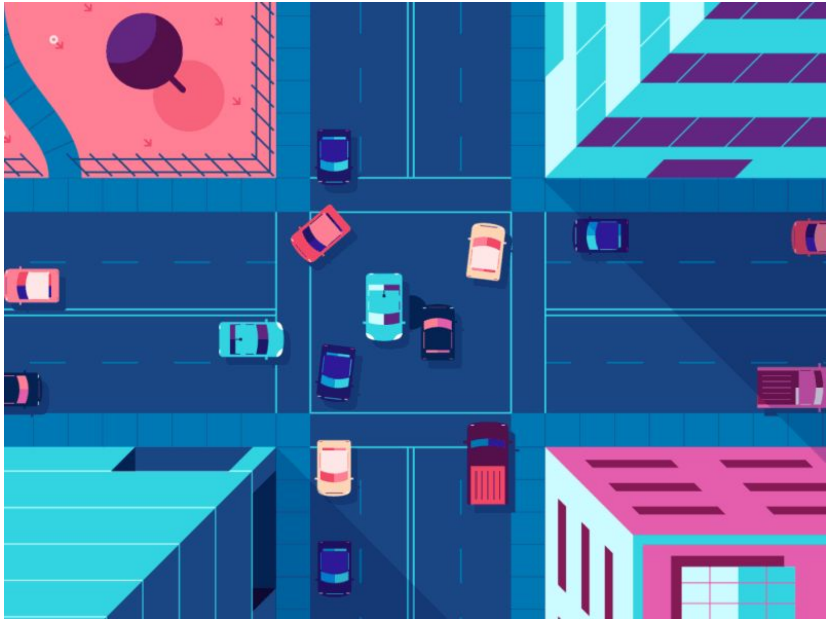
Methods	Val Seen				Val Unseen				Test Unseen			
	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑
GridMM [20]	4.21	69	60	53	4.44	58	50	44	5.64	56	46	39
ETPNav [17]	3.95	72	66	59	4.71	65	57	49	5.12	63	55	48
BEVBert [21]	3.45	78	71	61	4.57	67	59	50	4.70	67	59	50
SA-VLN (our)	**3.31**	**80**	**74**	**63**	**4.46**	**69**	**61**	**51**	**4.64**	**68**	**60**	**51**

Overview

- Recap
- Limitations
- Proposal
- Conclusion

Recap

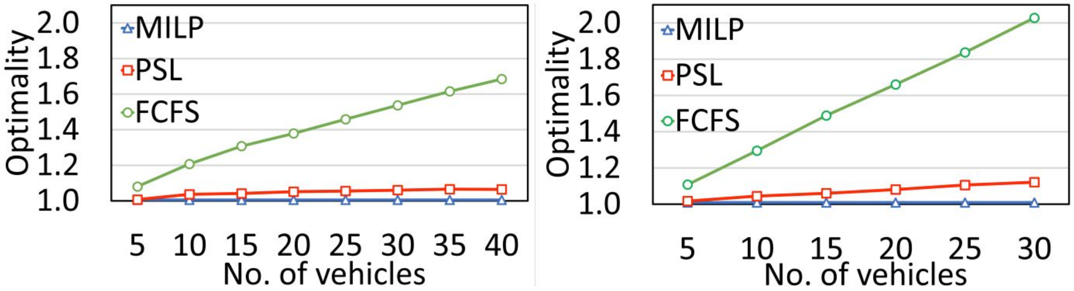
- Improving intersection coordination for fully autonomous vehicles
- Base paper: Intersection Coordination with Priority-Based Search for Autonomous Vehicles
  - MAPF inspired algorithm called PSL



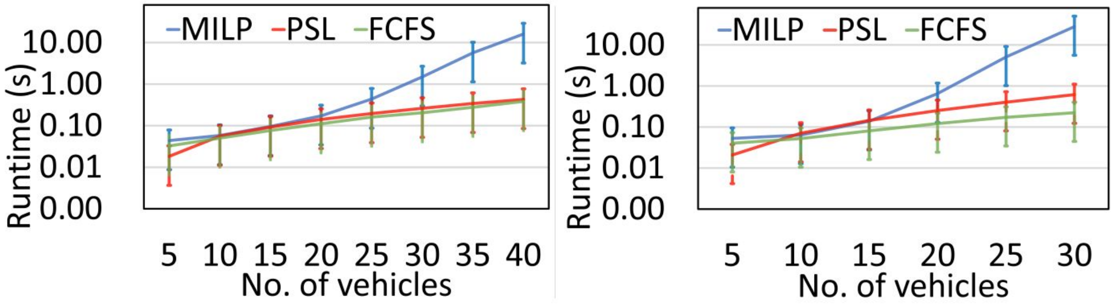
PSL

- PBS: find priority ordering of agents
- SIPP: find safe trajectory and time intervals
- LP: find optimal entry time and speed

Result of PSL



(a) 500vphpl (b) 800vphpl



(a) 500vphpl (b) 800vphpl

Limitations

- PSL is exponential in number of conflict points
    - Will be expensive to compute large junctions or time-windows

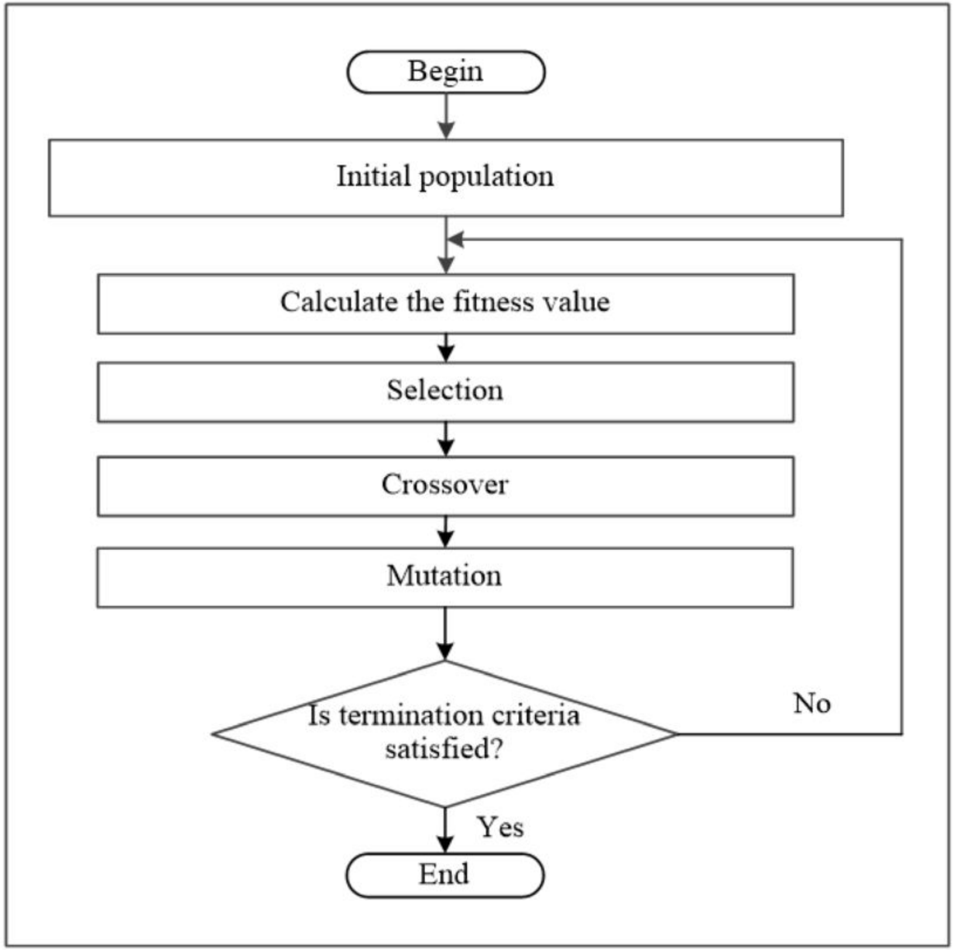
- No time guarantee
    - In dynamic situation where we need quick decisions

- Hard to extend to multi-objective context
    - Balancing maximum delay for one car and total delay for example

Genetic algorithm

- Can terminate at any point and return best solution found so far

- Simple MOO extension



Begin
Initial population
Calculate the fitness value
Selection
Crossover
Mutation
Is termination criteria satisfied?
No
Yes
End

Challenges with GA

- In GA every individual needs to have fitness evaluated
    - Running SIPP + LP on every agent

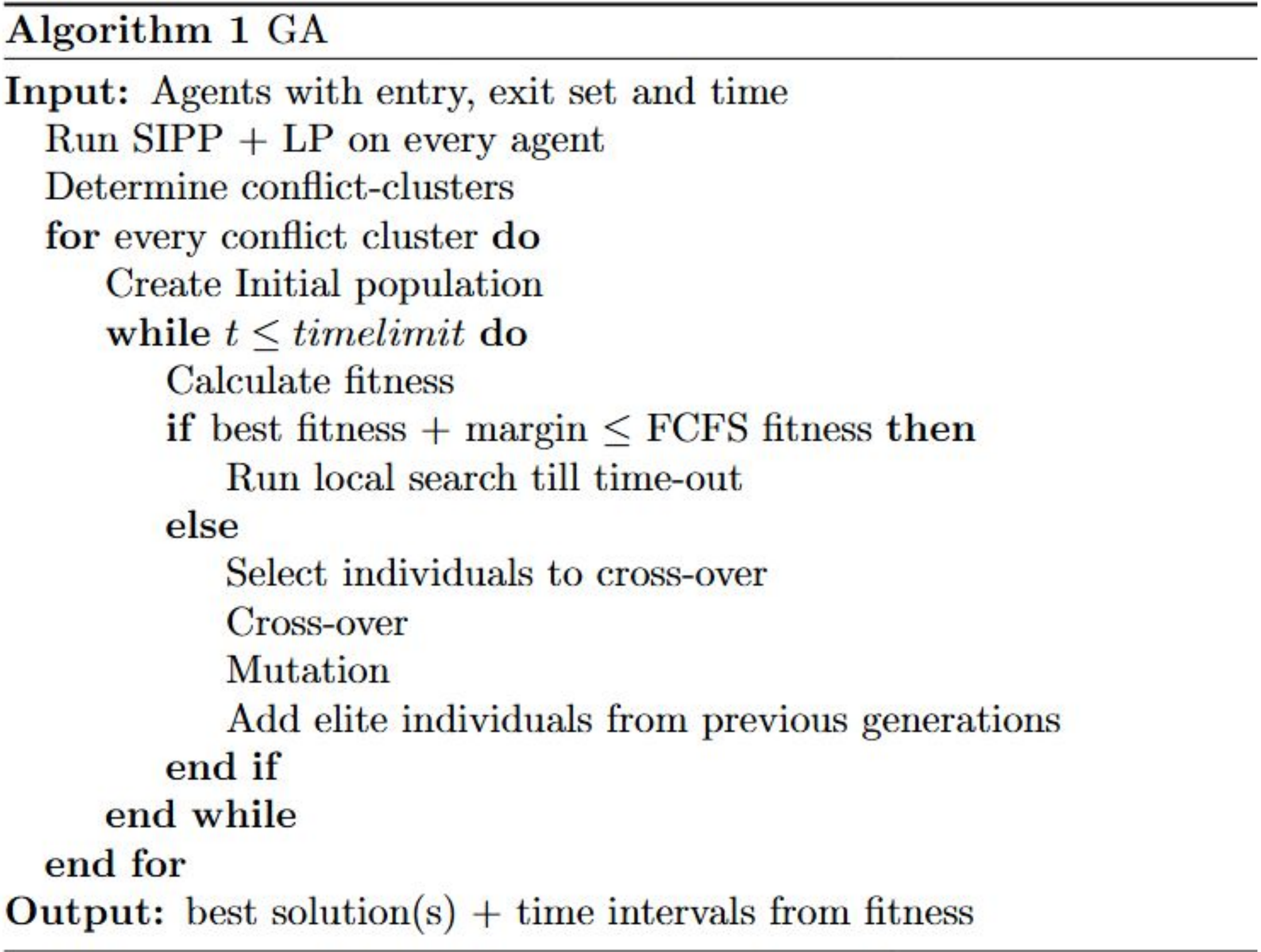
- How to design crossovers that improve solution with limited popsize and generations?

Proposal

- To reduce expensive calls keep population size small and start with heuristics
    - Initialize population using heuristic like FCFS or ATC

- Initially run FCFS ordering to detect non-conflicting paths and conflict clusters
    - Reduces amount of agents needed to evaluate fitness

- Use local search when ‘good’ solution is found
    - Local search finds local minima but doesn’t explore very different solutions



Algorithm 1 GA

```
Input: Agents with entry, exit set and time
  Run SIPP + LP on every agent
  Determine conflict-clusters
  for every conflict cluster do
      Create Initial population
      while $t \leq timelimit$ do
          Calculate fitness
          if best fitness + margin $\leq$ FCFS fitness then
              Run local search till time-out
          else
              Select individuals to cross-over
              Cross-over
              Mutation
              Add elite individuals from previous generations
          end if
      end while
  end for
Output: best solution(s) + time intervals from fitness
```

Limitations

- In smaller junctions can result in significantly worse performance
    - Hypothesis is that it will outperform (in speed) on larger junctions

- Dependent on tuning and optimization of GA
    - Especially since population size will be small

Other Ideas to Explore

- Use heuristic evaluation function
    - More suboptimal but allows for larger population size and more generations

- Instead of GA find better search strategy for PBS than depth-first

- Learn combined heuristic rule using decision tree

Thanks for listening