Robust Autonomy Emerges from Self-Play

Presented by: Jesse Silverberg for IFT 6757

2025-11-26

Driving is just one big RL problem

Reward Hypothesis:

"That all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)."

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t r_t \right]$$

Policy:

$$\pi(a \mid s)$$

Value function:

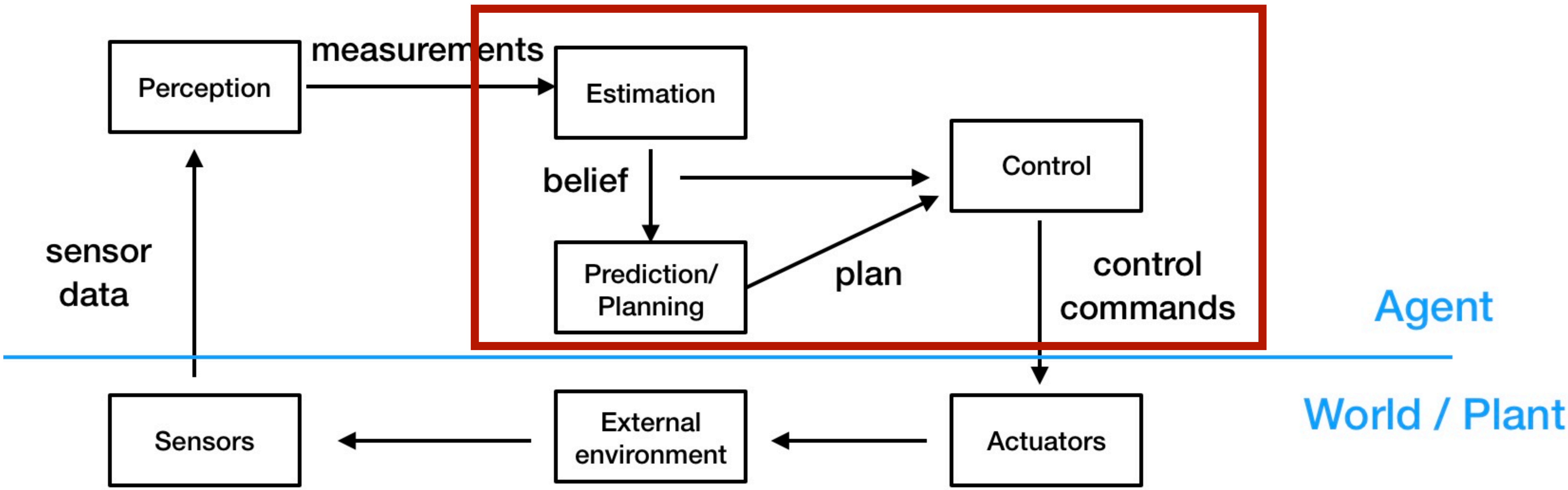
$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{i=t}^{T} \gamma^{i-t} r_i \mid s_t = s \right] = \mathbb{E}_\pi \left[r_t + \gamma V^\pi(s_{t+1}) \right]$$

Task

Reward Function

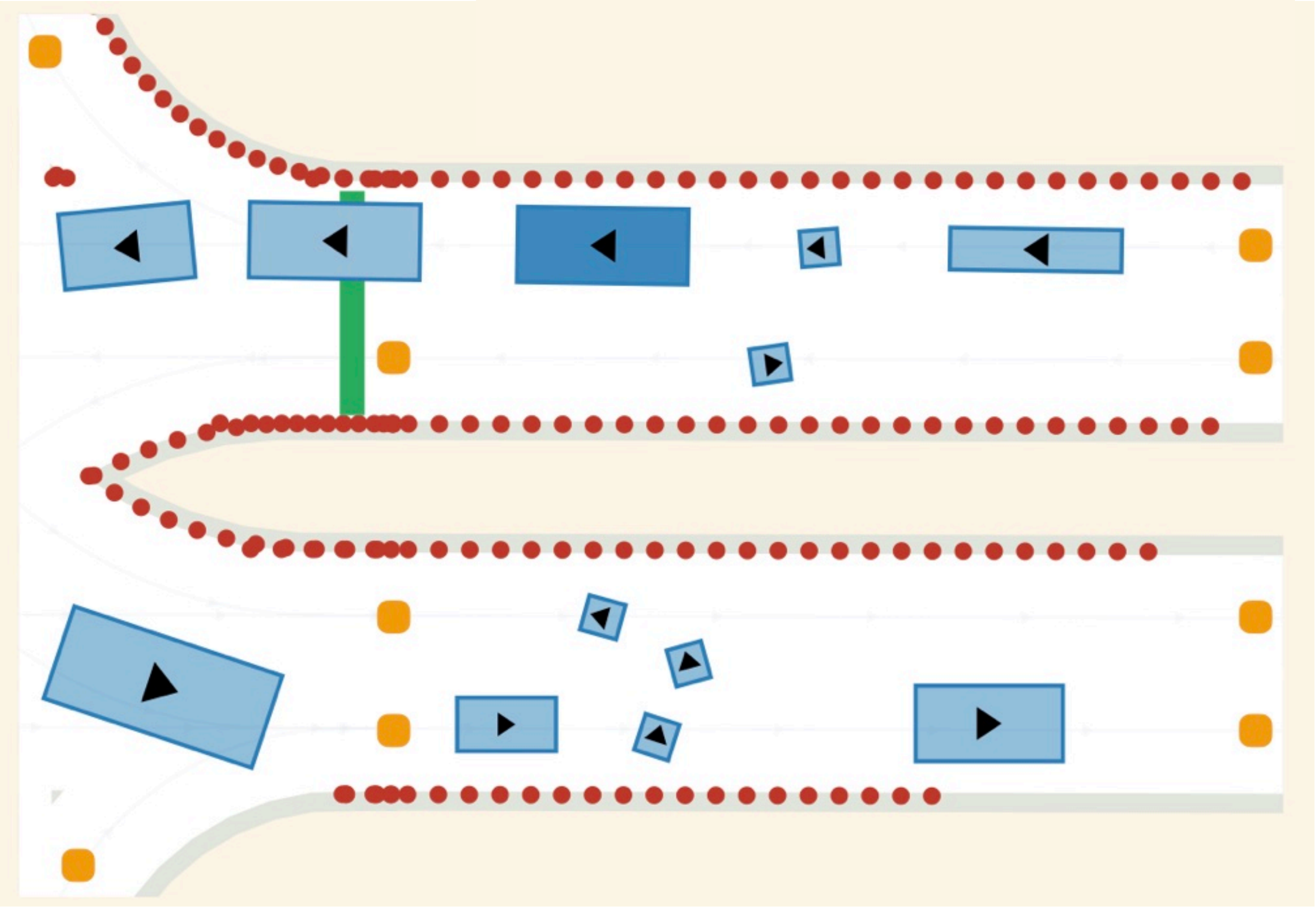
Term	Approximate Magnitude
Reward within goal radius	1
Forward velocity incentive	0.0025
Timestep penalty	-0.000025
Driving in reverse	-0.001
Misaligned lane direction/centering	-0.001
High acceleration or jerk	-0.1
Running red lights	-0.5
Out-of-road penalty	-1.5
Collision penalty, scaled by velocity	-5

Problem Setting



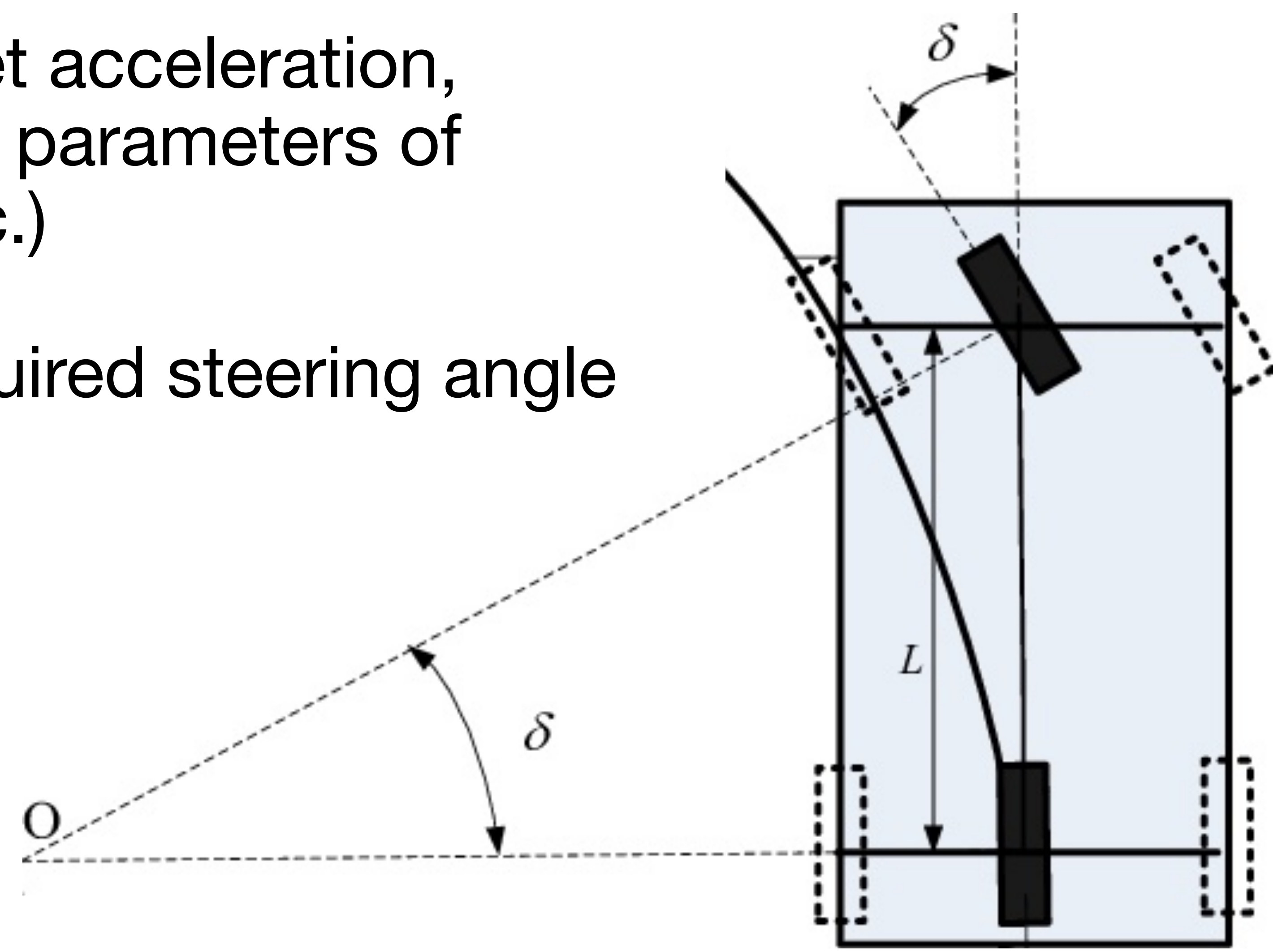
Observations

- Lane points, stop lines, traffic lights
- Ego observation: lane offset, lane bearing, road curvature, velocity, speed limit, steering angle, acceleration
- Other vehicles: sizes, locations, orientations, and velocities
- Obstacles (immobile vehicles)



Bicycle Dynamics Model

- Lateral jerk: $\{-4,0,4\}\ \mathrm{m/s^3}$
- Longitudinal jerk: $\{-15,-4,0,4\}\ \mathrm{m/s^3}$
- Numerically integrate to get acceleration, velocity (based on intrinsic parameters of throttle/steer response, etc.)
- Can then calculate the required steering angle



[Li et. al., 2013]

Is driving just one big RL problem?

- What exactly is the environment?
  - *What controls the other vehicles?*
- Other vehicles need to behave (and react!) realistically
- If we train a driving policy end-to-end, can we control or tweak how it drives?
- **Good driving is subjective!**
- Can we interpret the policy?

Self-Play: a Recipe for Success

[DeepMind, 2016]

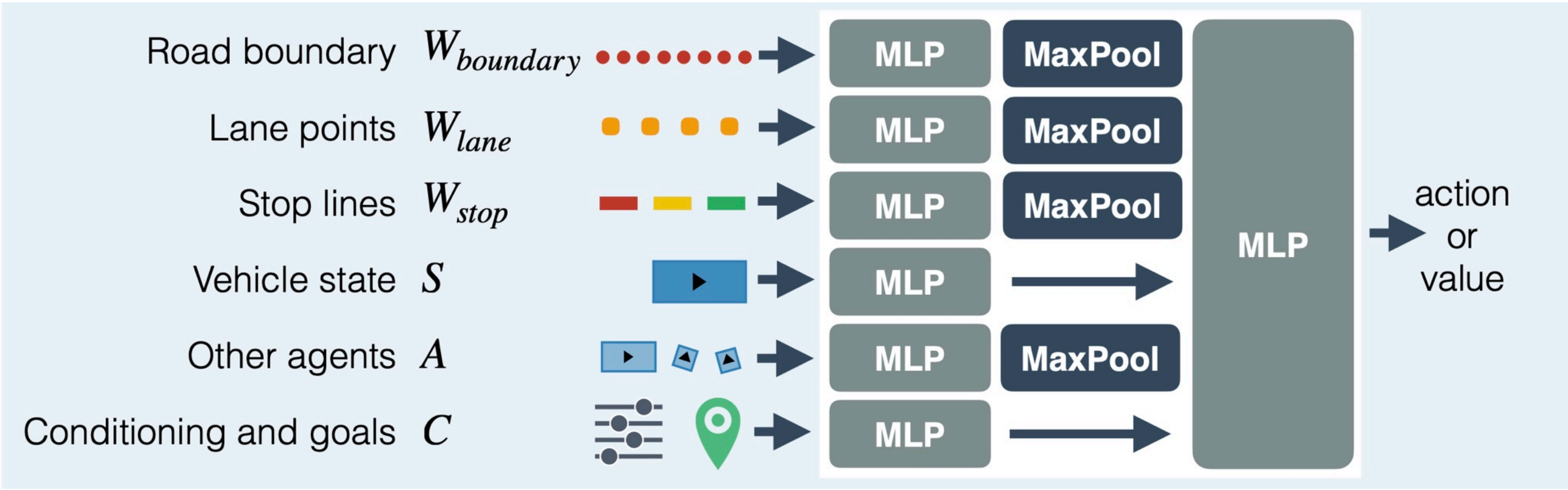
[OpenAI, 2019]

Single-Agent RL

- Multi-agent RL is hard…
- Assuming society will not do a cold-turkey switch to AVs, then any approaches cannot assume coordination
- Communication might lead to “hacking”-style behaviors
- **Humans can do it!**

Architecture

- 1024 x 1024 x 1024 main MLP for actor and critic (3M each)
- Permutation-invariant encoders
- Many lane and boundary features; dropout 50% and 40%, resp.



Implementation Details

Proximal Policy Optimization

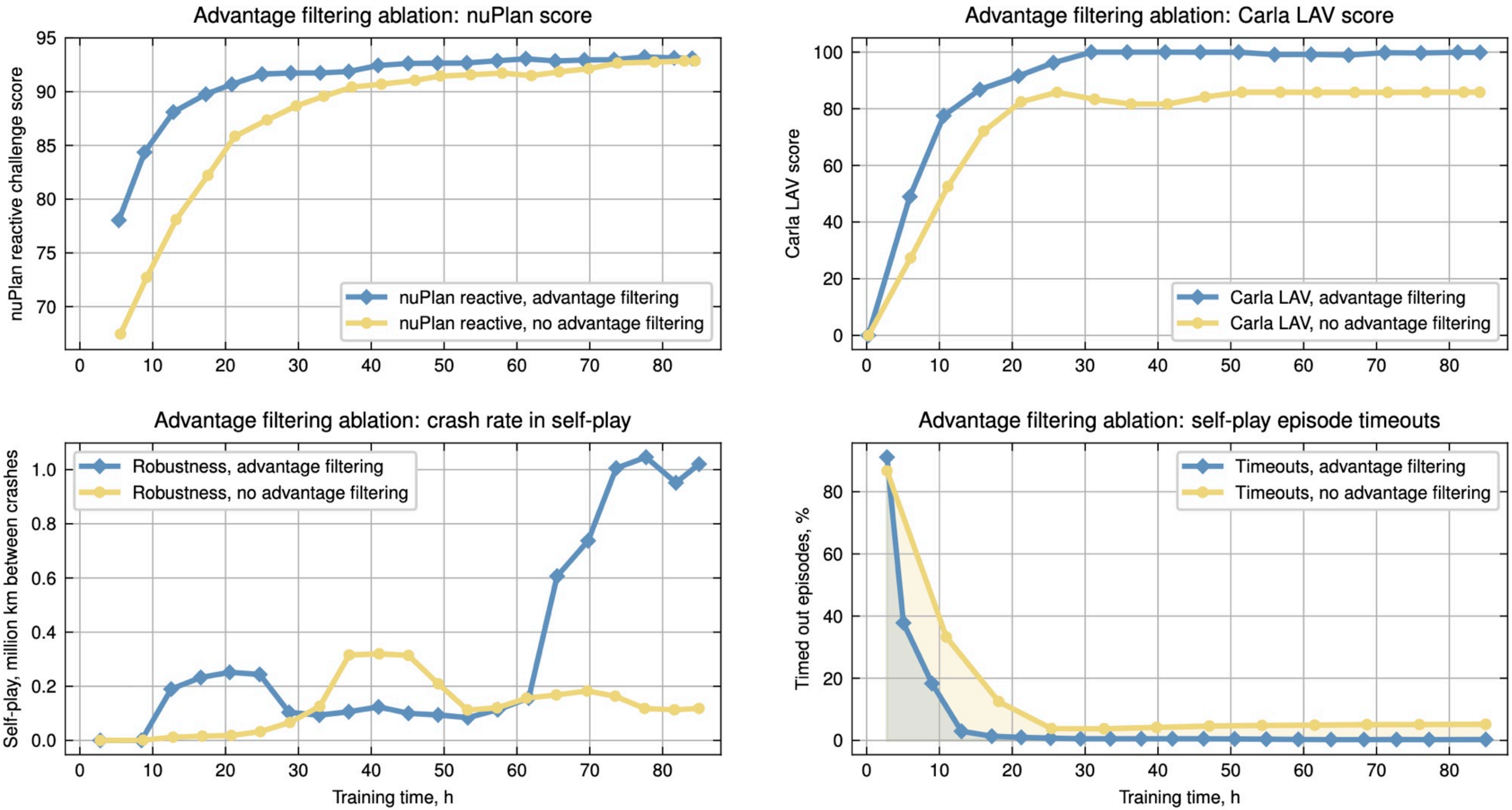
$$
\begin{aligned}
A_t^{(1)} &= r_t + \gamma V(s_{t+1}) && - V(s_t) \\
A_t^{(2)} &= r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) && - V(s_t) \\
A_t^{(3)} &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) && - V(s_t) \\
&\vdots
\end{aligned}
$$

$$A_t^{\mathrm{GAE}} = (1 - \lambda)\left(A_t^{(1)} + \lambda A_t^{(2)} + \lambda^2 A_t^{(3)} + \cdots\right)$$

$$J(\pi) = \mathbb{E}_\pi \left[\min \left\{ \frac{\pi(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} A^{\mathrm{GAE}}(s, a), \left[\frac{\pi(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} \right]_{1-\epsilon}^{1+\epsilon} A^{\mathrm{GAE}}(s, a) \right\} \right]$$

Advantage Filtering

- In a fast simulator, data is very cheap; is it all useful?
- Each epoch of data collection is ~90M samples
- **Filter out** any data with $|A| < 0.01|A_{\max}|$
- In practice, this **removes 80-90%!**
- Result: a reasonable per-device batch size of 32,000 transitions



Advantage filtering ablation: nuPlan score
nuPlan reactive challenge score
95
90
85
80
75
70
0
10
20
30
40
50
60
70
80
nuPlan reactive, advantage filtering
nuPlan reactive, no advantage filtering
Advantage filtering ablation: Carla LAV score
Carla LAV score
100
80
60
40
20
0
Carla LAV, advantage filtering
Carla LAV, no advantage filtering
Advantage filtering ablation: crash rate in self-play
Self-play, million km between crashes
1.0
0.8
0.6
0.4
0.2
0.0
Robustness, advantage filtering
Robustness, no advantage filtering
Training time, h
Advantage filtering ablation: self-play episode timeouts
Timed out episodes, %
80
60
40
20
0
Timeouts, advantage filtering
Timeouts, no advantage filtering
Training time, h

Designing for Scaling

- On-policy RL is very sample-inefficient
- Modern accelerators are good at batched computations
- Authors wrote a bespoke parallel simulator that runs on GPU:
  - 4.4 billion transitions = 7.2 M km = 42 years per hour per node
  - **360,000x real time**
  - < $5 per 1M km
  - 10 days, 8xA100: 1T transitions = 1.6B km = **9500 years**

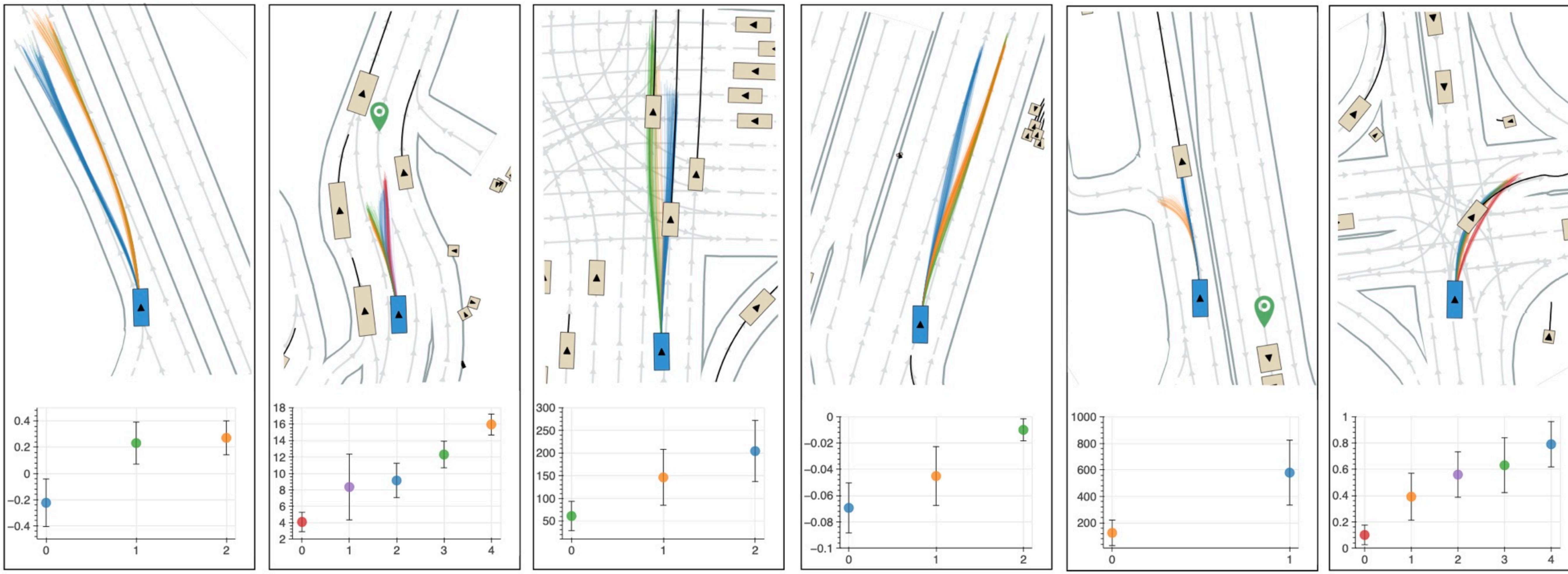
Increasing Diversity

Designing for Simulation

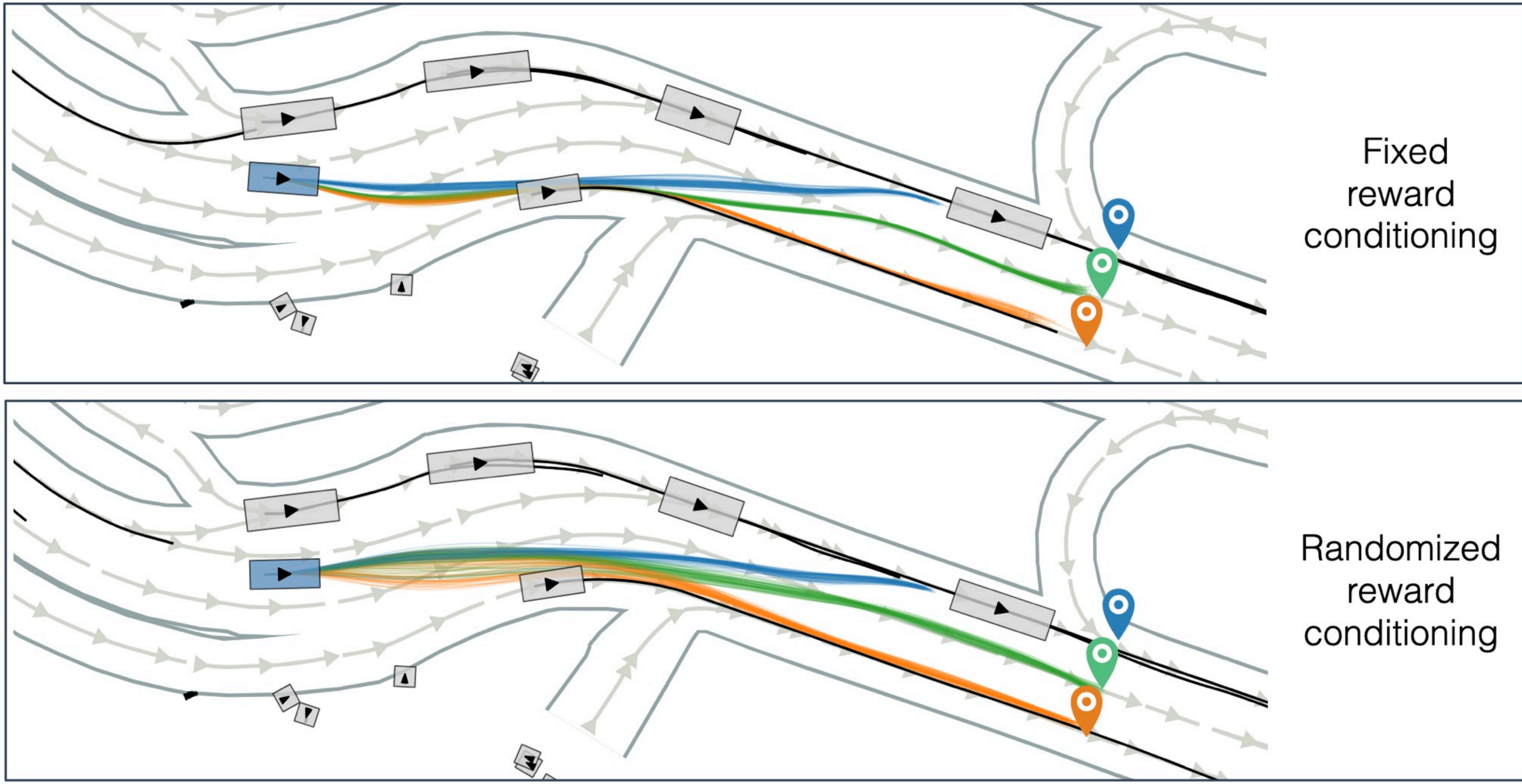
- We can cheaply generate data; how to solve the long tail issue?
- Random goals encourage good coverage, what else?
- How to elicit **meaningful** varied behavior, not just random noise?

Reward Conditioning

- **Key idea:** weight the reward function terms differently for each agent, and **condition the policy** on the weights
- These weights are sampled from some distribution at initialization
- At test-time, condition on the median of the range
- **Efficient** for inference and training
- Easy to produce **combinatorially many variants!**

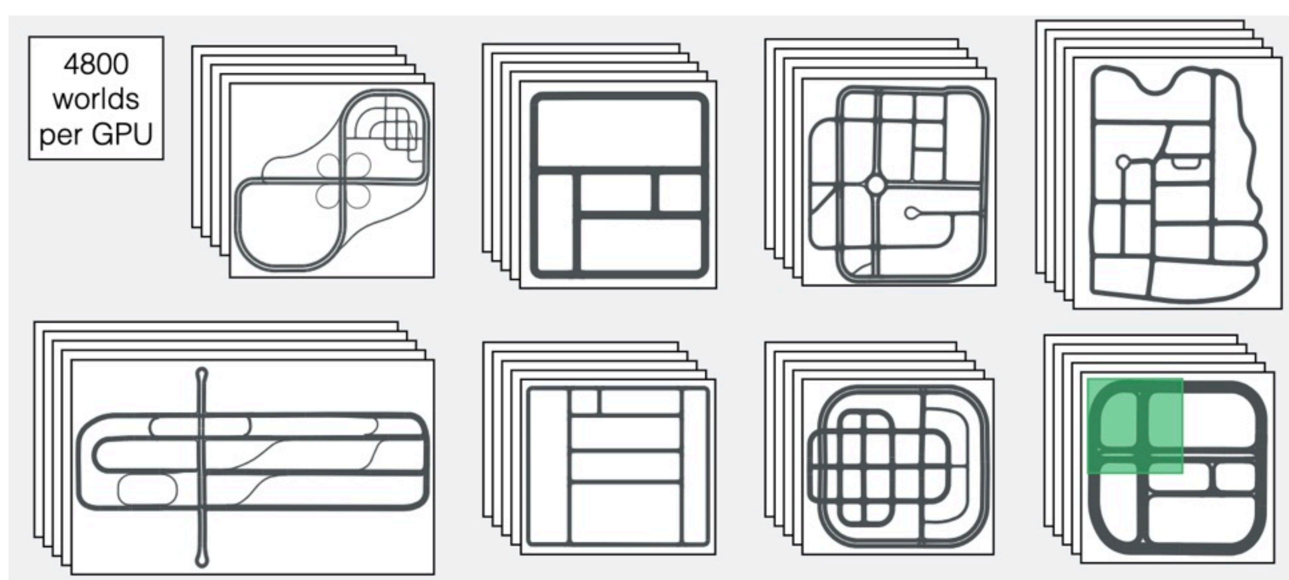


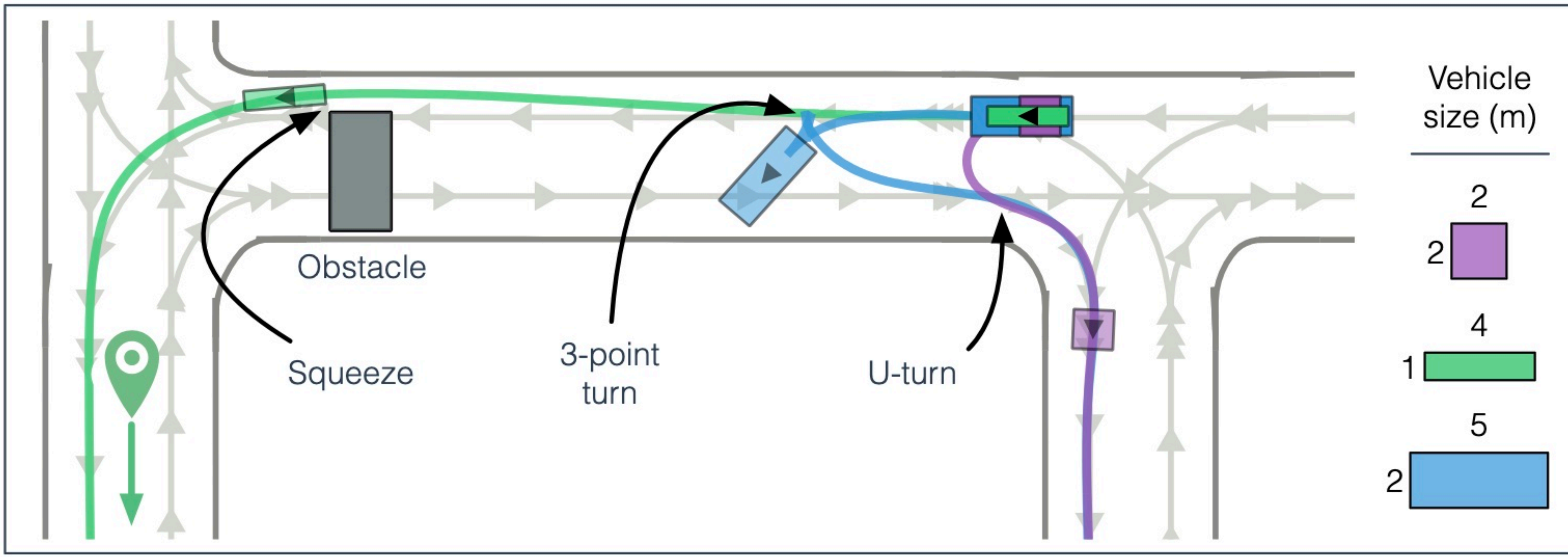
a $\alpha_{\text{center-bias}}$ **b** α_{goal} **c** $\alpha_{\text{l-center}}$ **d** α_{comfort} **e** $\alpha_{\text{l-align}}$ **f** $\alpha_{\text{vel-align}}$


Fixed reward conditioning
Randomized reward conditioning

Data Augmentation

- Driving on left vs. right side of road (2x maps)
- Random number of agents, altering traffic density
- Noisy state observations and transitions
- Sample random vehicle parameters (throttle, steering sensitivity)
- Random goal radii, traffic light duration, traffic light dropout (20%)

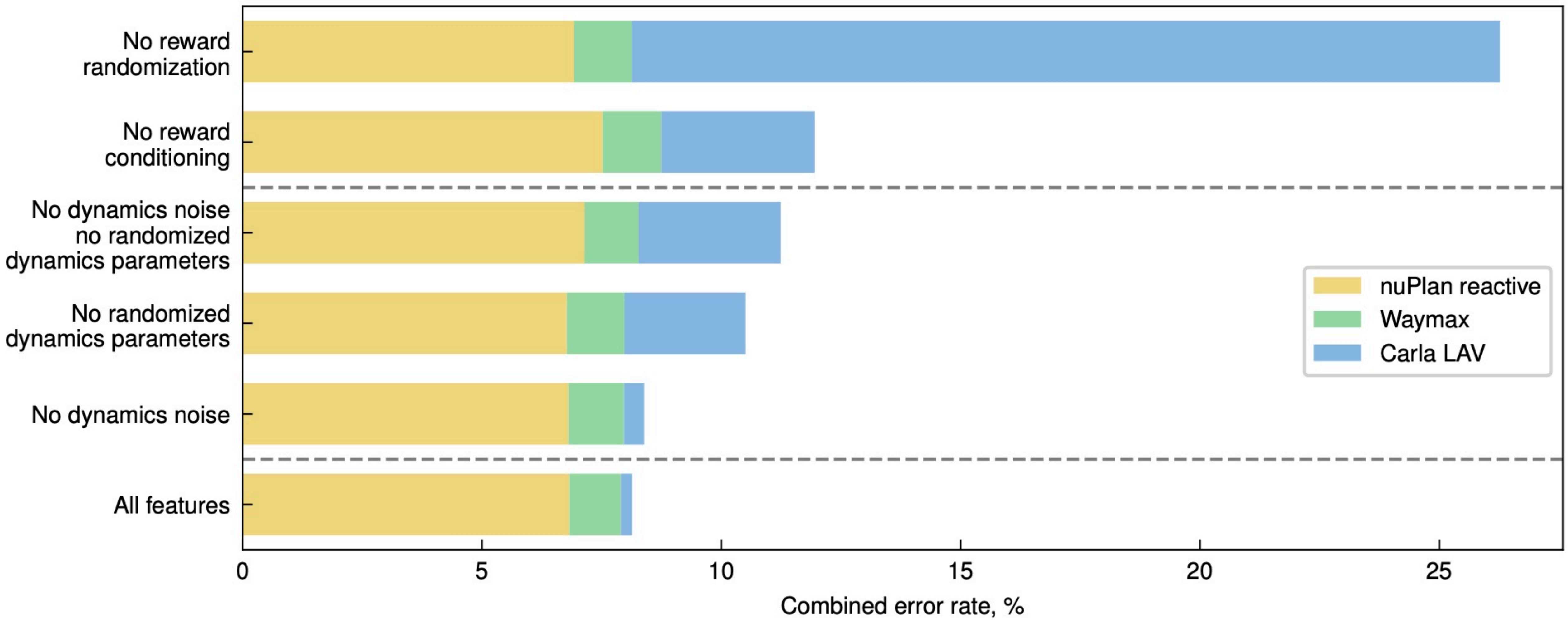



Vehicle size (m)
2
2
4
1
5
2
Obstacle
Squeeze
3-point turn
U-turn

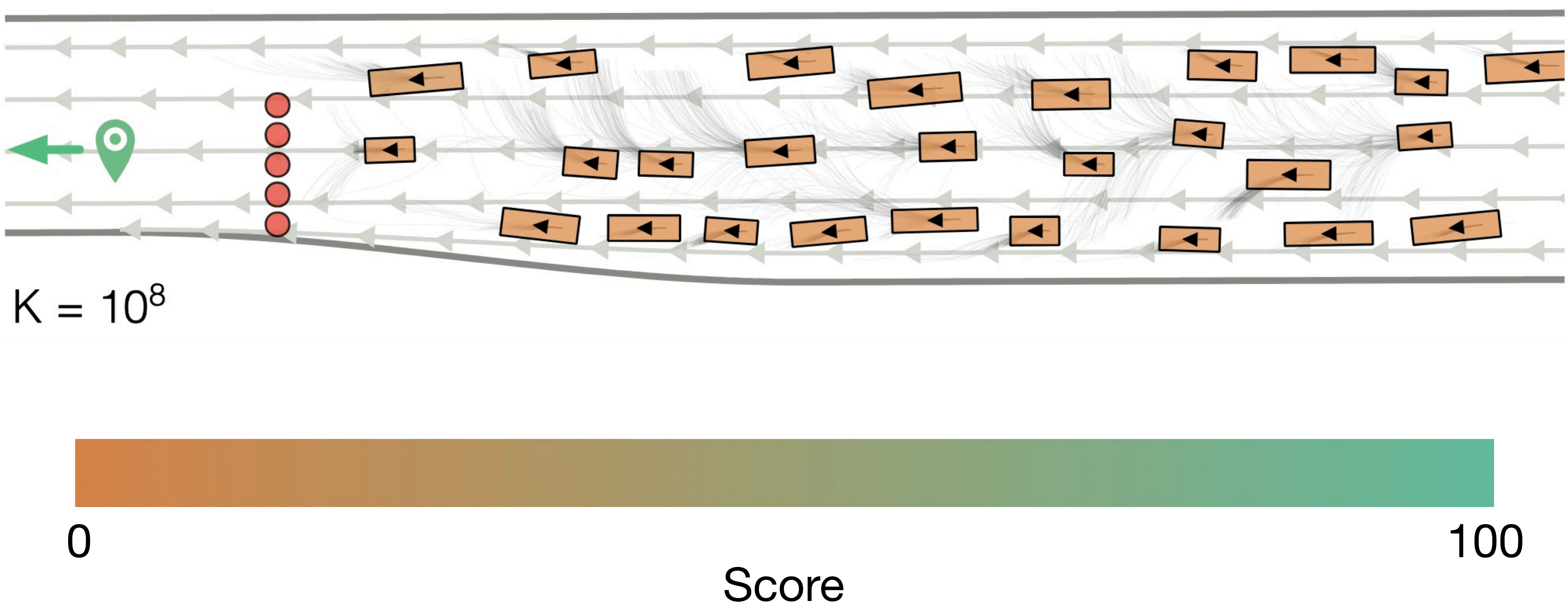
More Uncertainty

- The goals of other agents are unknown
- If a vehicle crashes, it becomes immobile, altering the topology
- Add intentionally bad drivers exhibiting:
  - “Blind spots” (from random observation masking)
  - Sudden stops (from random action masking)

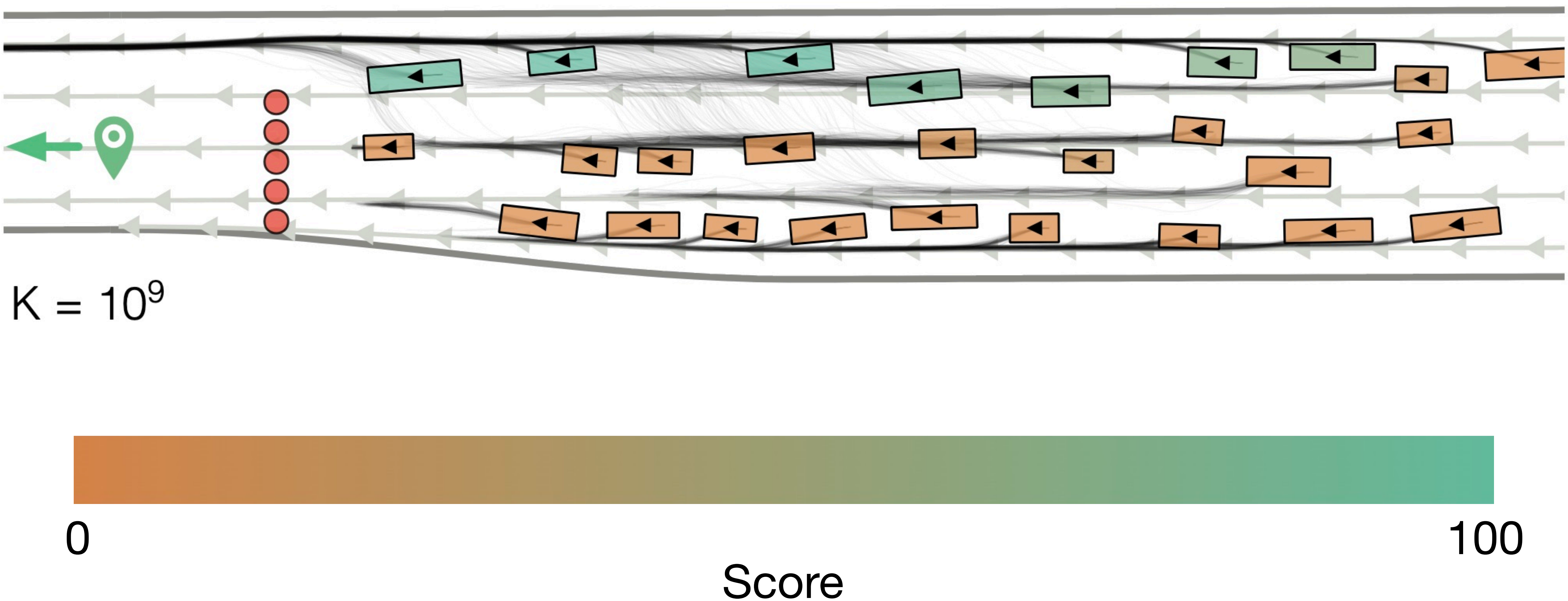
Ablation

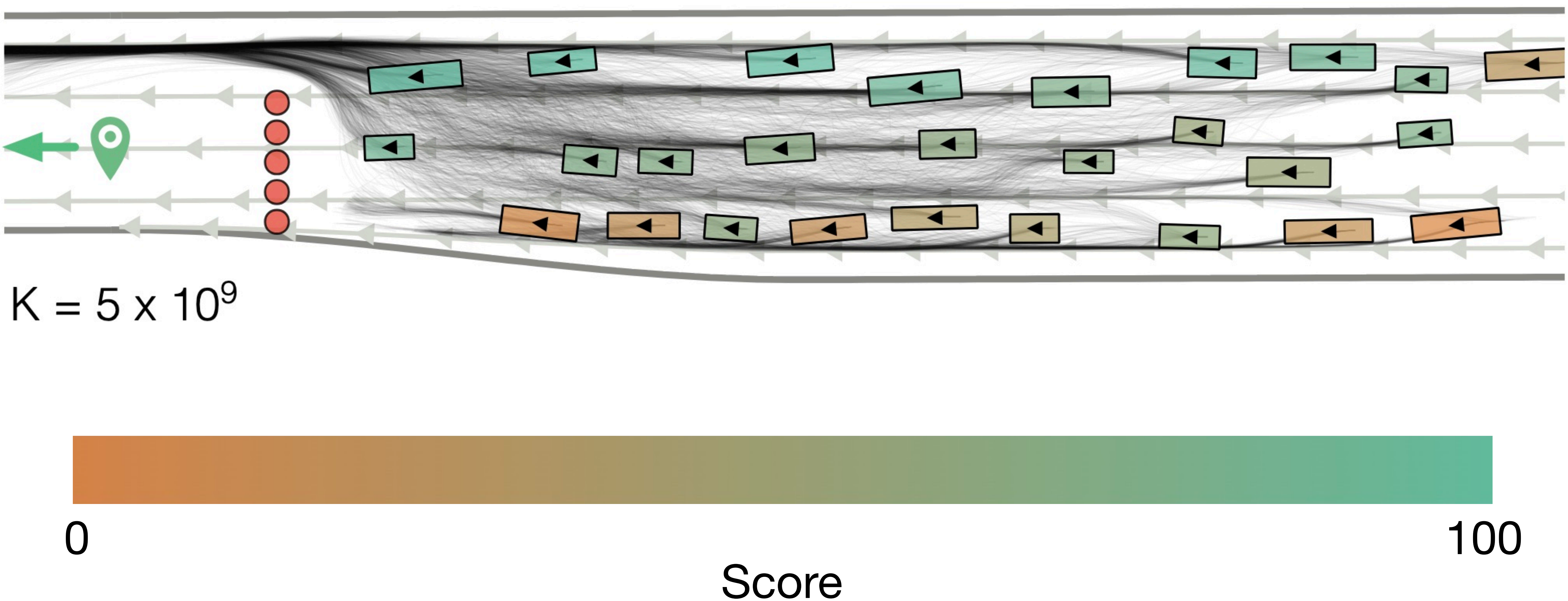

No reward randomization
No reward conditioning
No dynamics noise no randomized dynamics parameters
No randomized dynamics parameters
No dynamics noise
All features
0
5
10
15
20
25
Combined error rate, %
nuPlan reactive
Waymax
Carla LAV

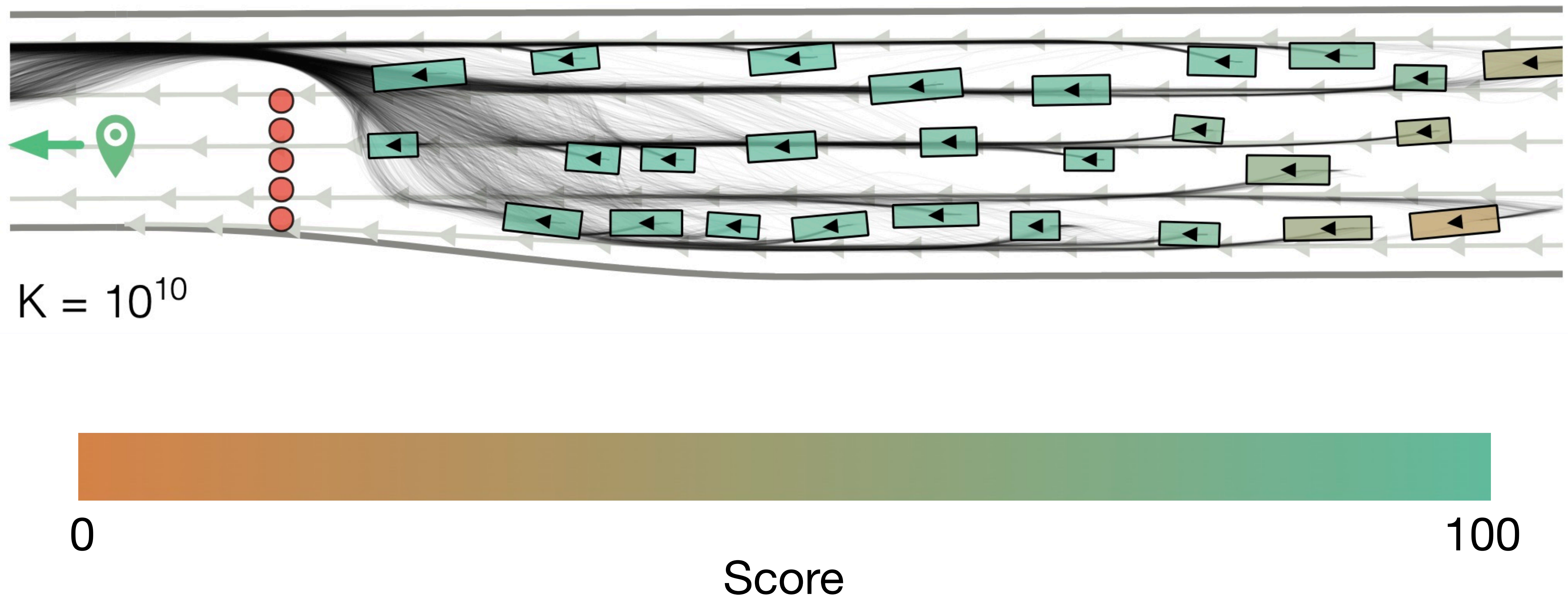
Results

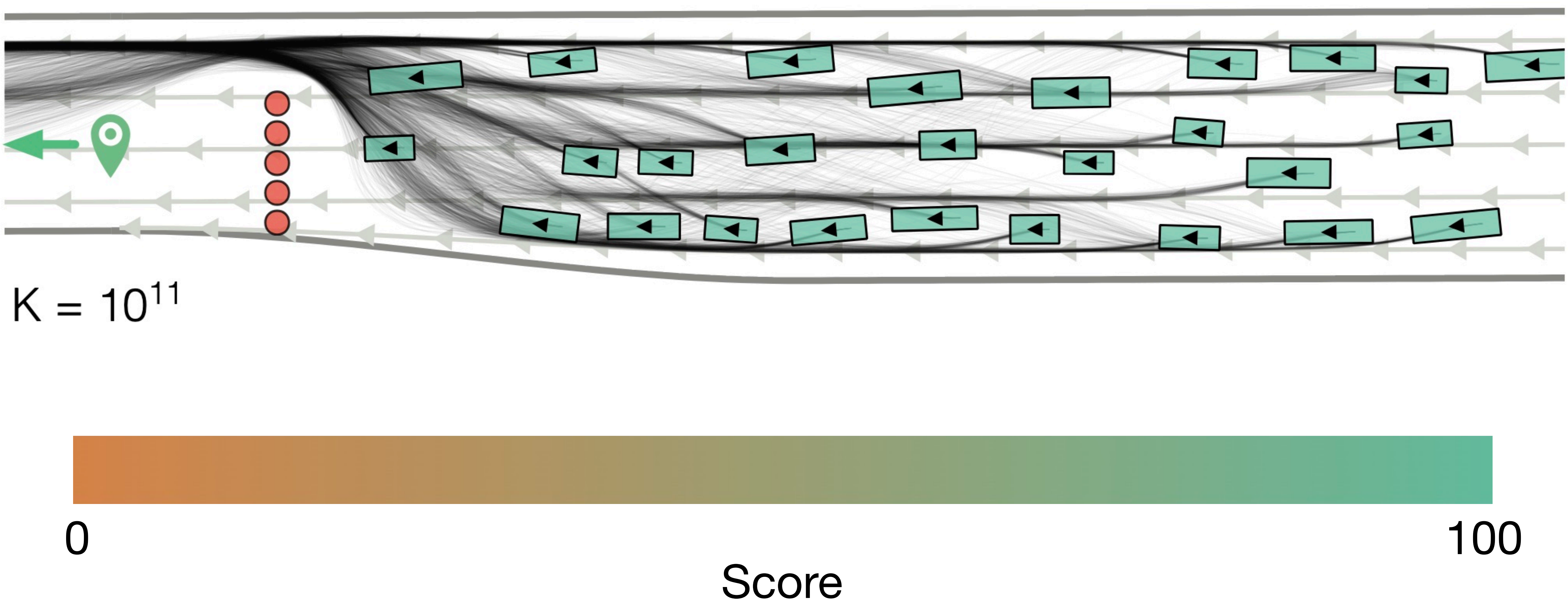


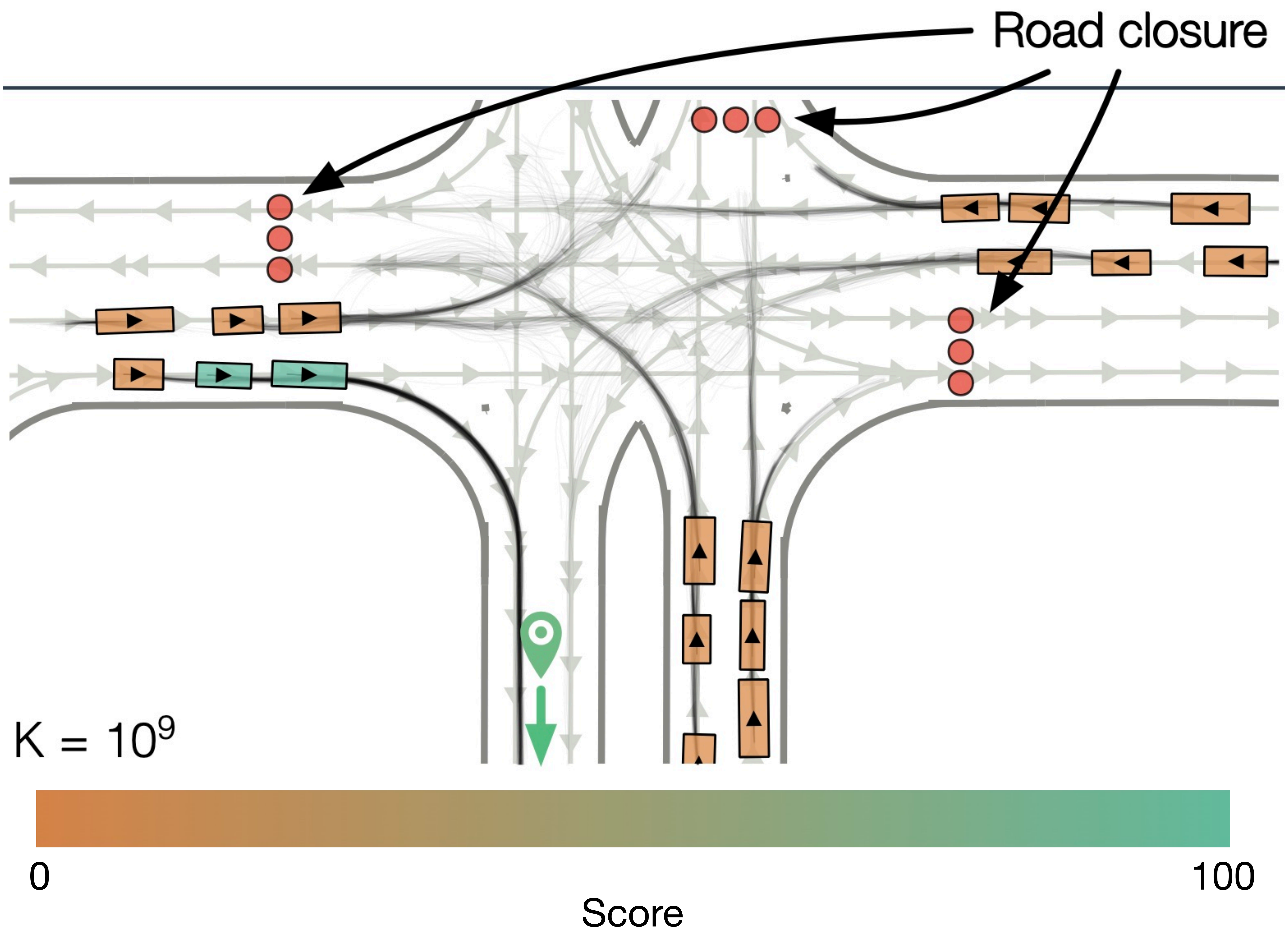
$K = 10^8$
0
100
Score

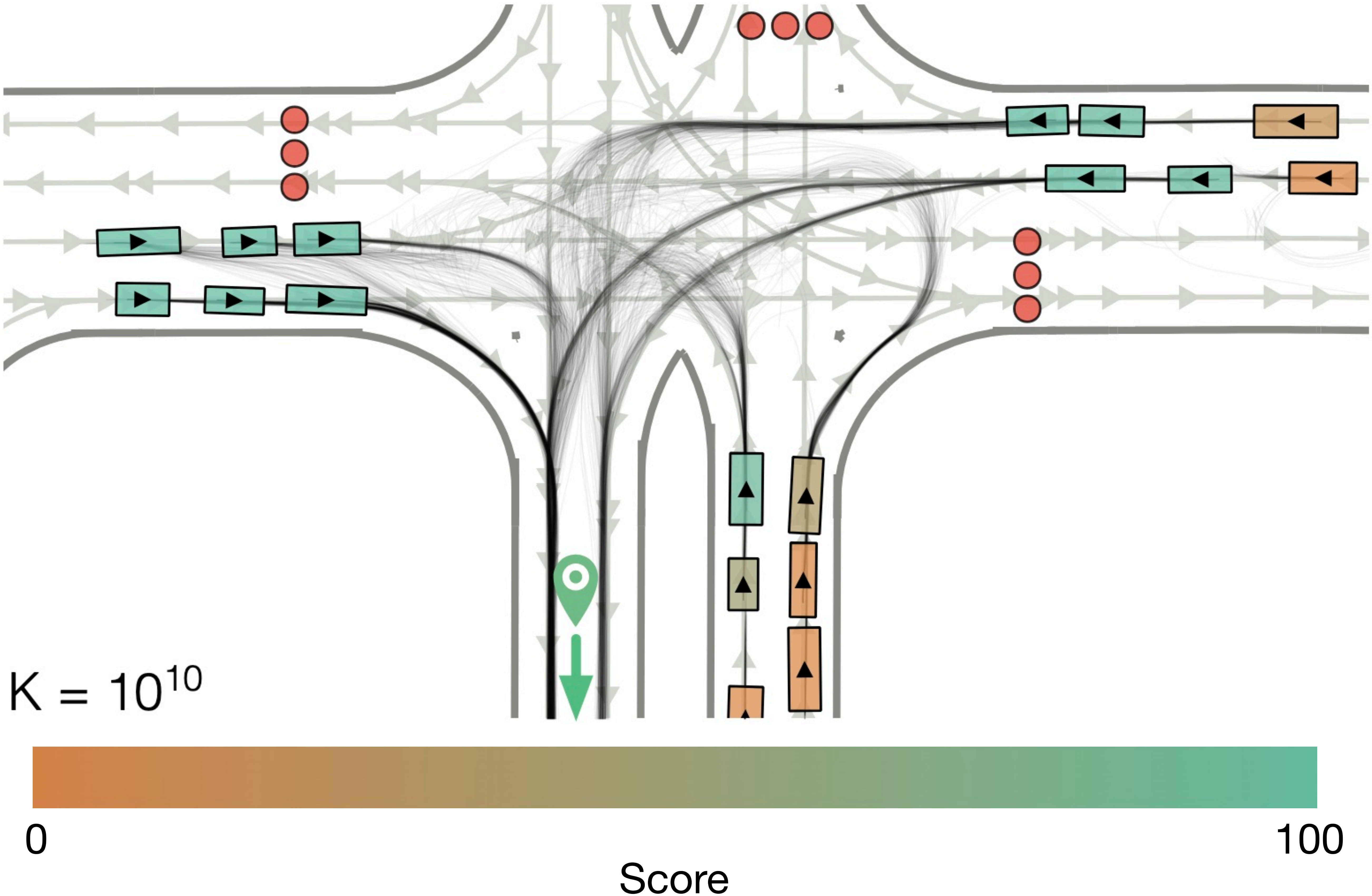

K = 10^9
0
100
Score

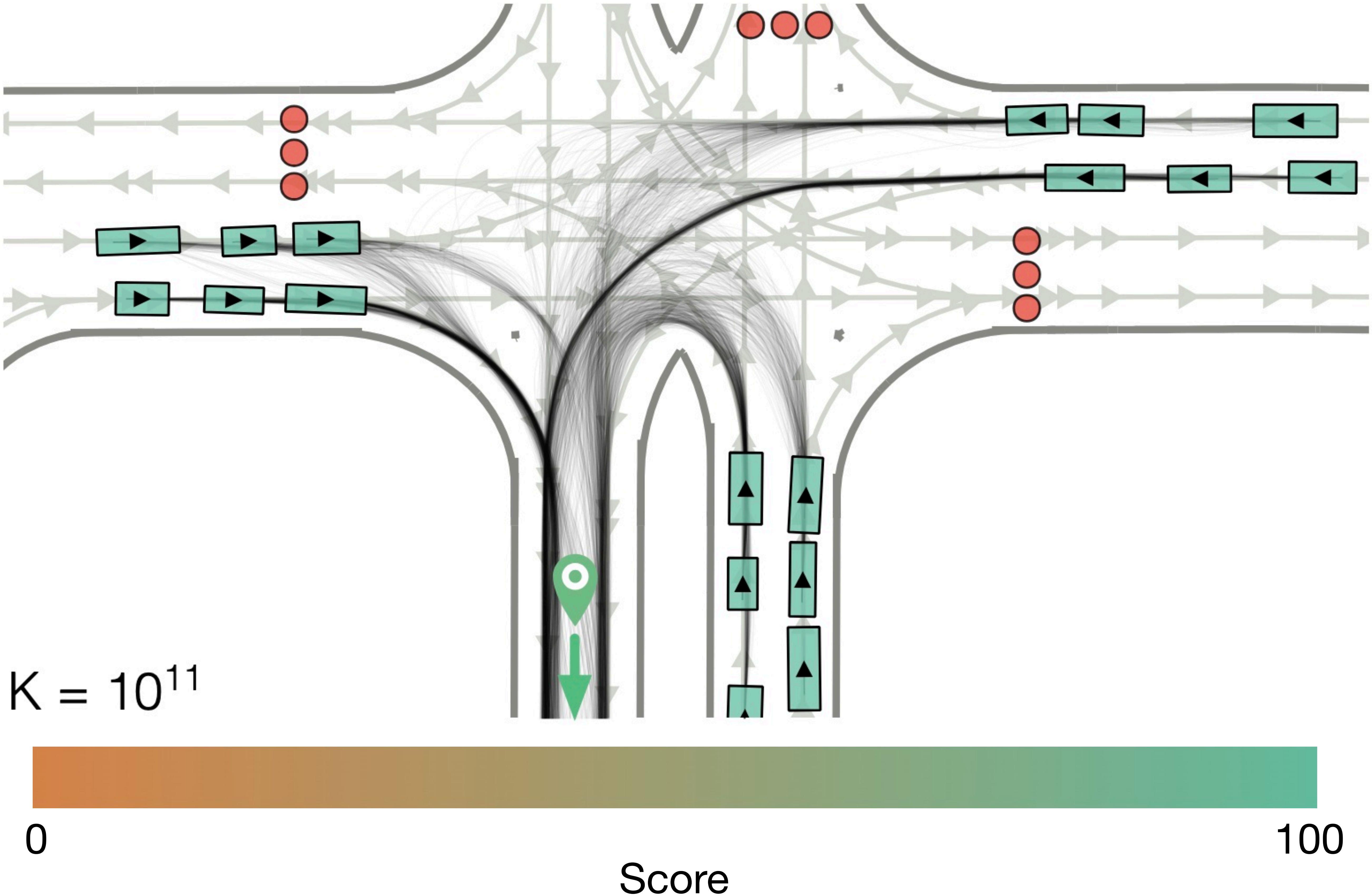

K = 5 x 10^9
0
100
Score

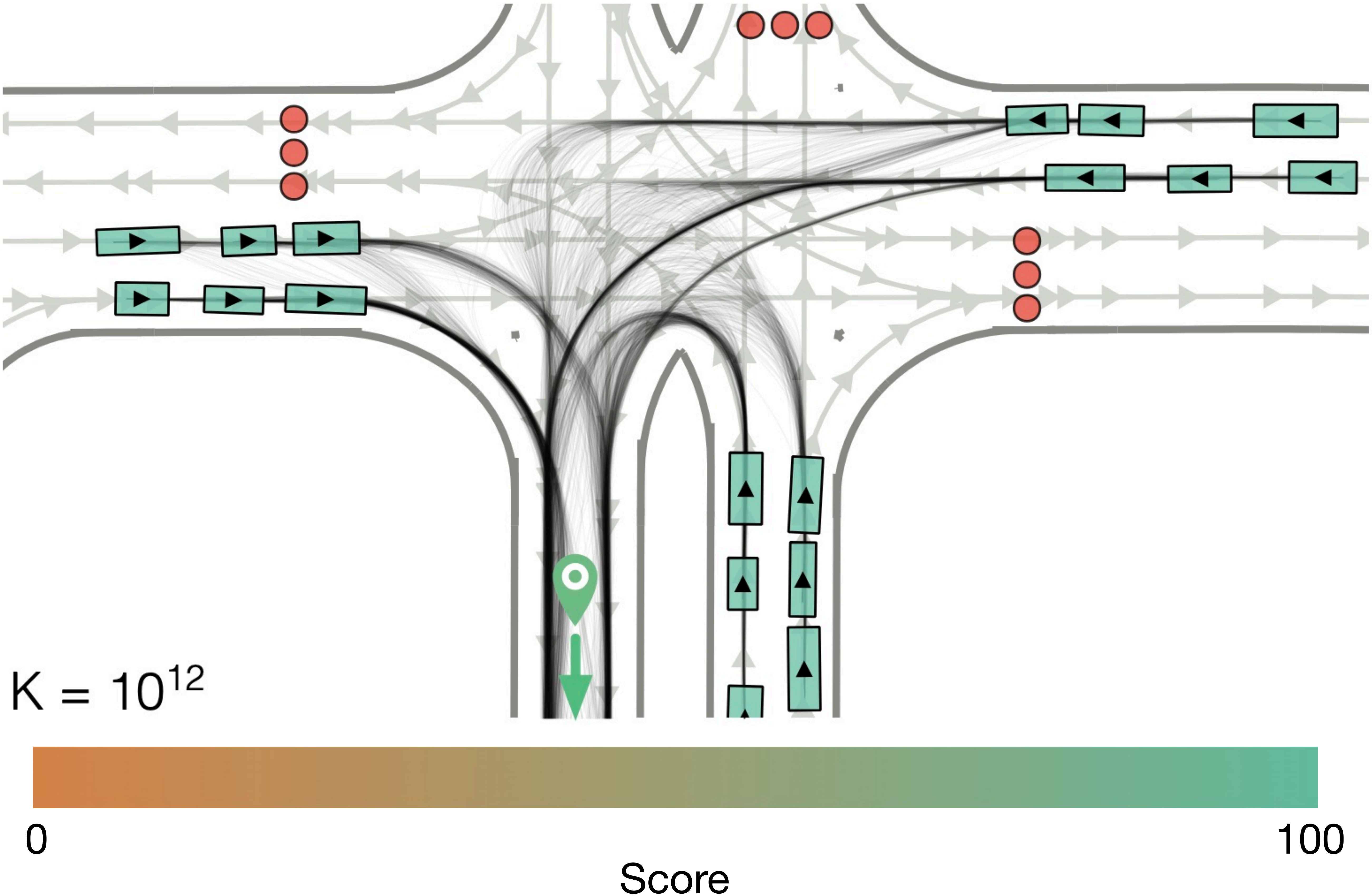

K = 10^{10}
0
100
Score

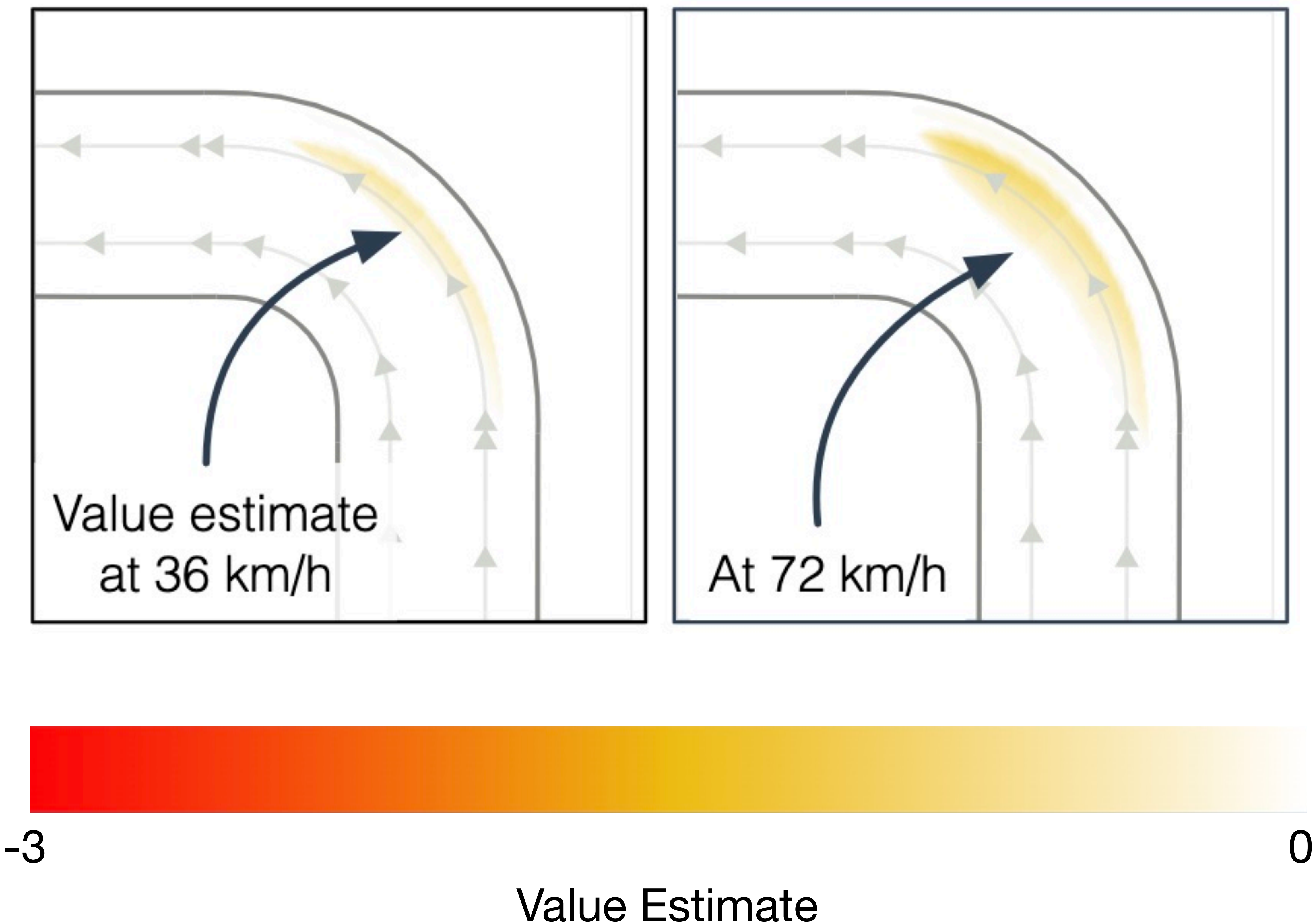

K = 10^{11}
0
100
Score

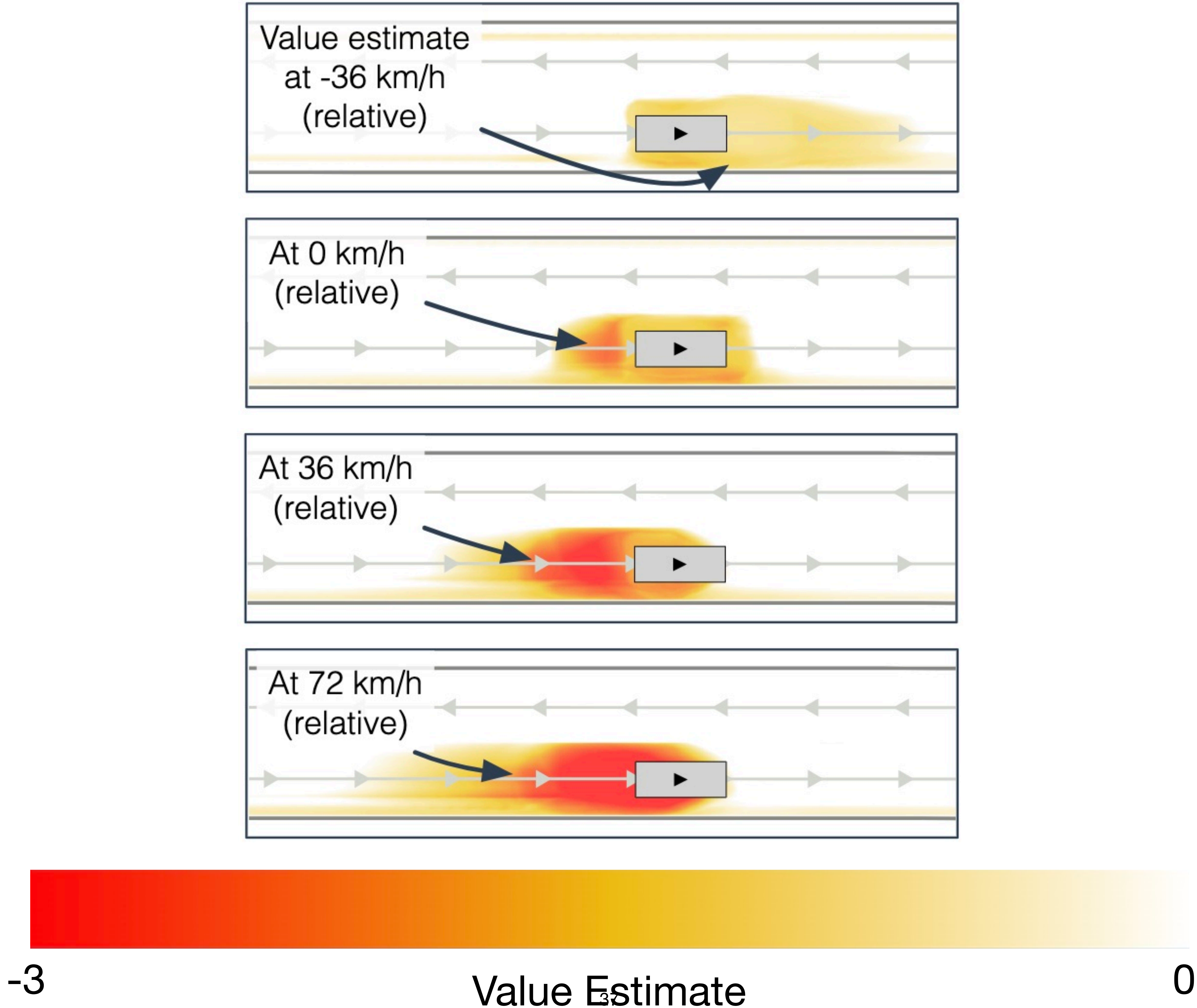

Road closure
K = 10^9
0
100
Score

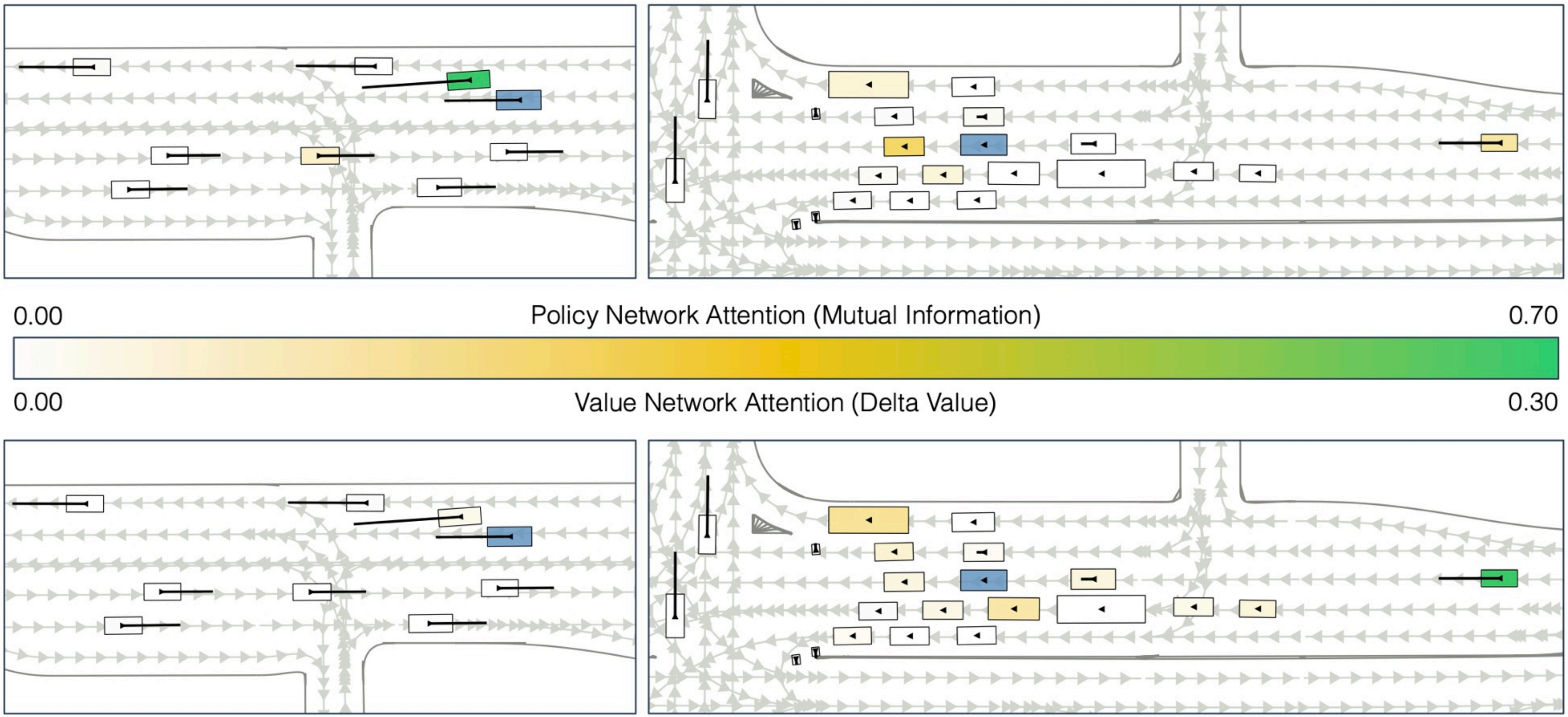

K = 10^10
0
100
Score

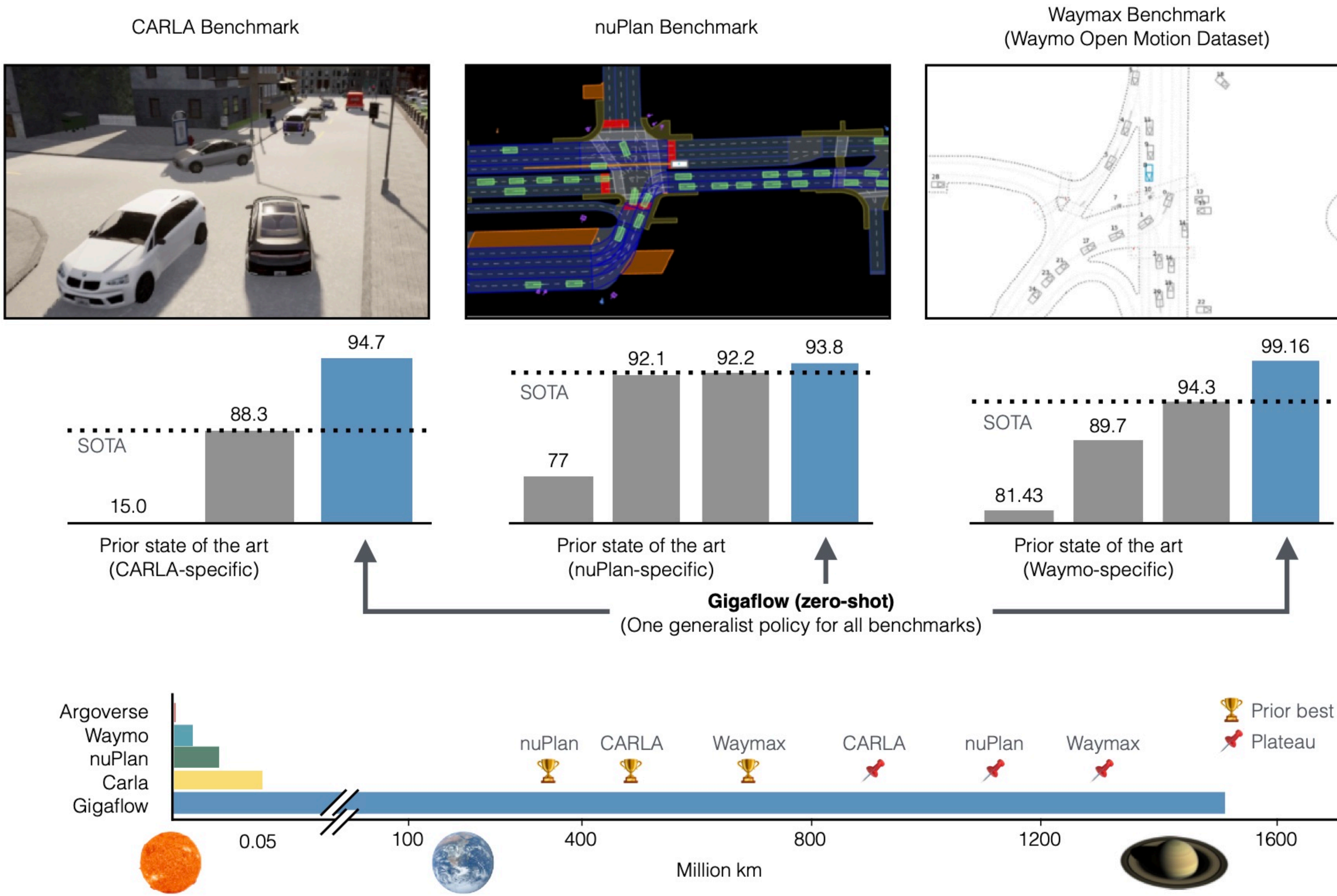

K = 10^11
0
100
Score


$K = 10^{12}$
0
100
Score


Value estimate
at 36 km/h
At 72 km/h
-3
0
Value Estimate


Value estimate
at -36 km/h
(relative)
At 0 km/h
(relative)
At 36 km/h
(relative)
At 72 km/h
(relative)
-3
Value Estimate
0


0.00
Policy Network Attention (Mutual Information)
0.70
0.00
Value Network Attention (Delta Value)
0.30


CARLA Benchmark
nuPlan Benchmark
Waymax Benchmark
(Waymo Open Motion Dataset)
94.7
88.3
SOTA
15.0
Prior state of the art
(CARLA-specific)
92.1
92.2
93.8
SOTA
77
Prior state of the art
(nuPlan-specific)
99.16
94.3
SOTA
89.7
81.43
Prior state of the art
(Waymo-specific)
Gigaflow (zero-shot)
(One generalist policy for all benchmarks)
Argoverse
Waymo
nuPlan
Carla
Gigaflow
nuPlan
CARLA
Waymax
CARLA
nuPlan
Waymax
Prior best
Plateau
0.05
100
400
800
1200
1600
Million km

Safety Comparison

- From class 1:
  - Humans average 1 death / 100M mi
  - Waymo has had 0 deaths in 7 M, 3 minor injuries
- **Trained agent achieves 1 incident / 1.9M mi (= 17.5 years)**

Implications

- **This saturates current simulation benchmarks zero-shot!**
- The policy is quite small (3M params); could be used for planning or other inference-time strategies
- Proof of concept that such (relatively) **simple and cheap data** can get you a long way in terms of **long-tail performance**

Questions

- What is the role of simulators in autonomous driving?
- How can we evaluate the risk of a system before deploying?
- How can we best use successful RL techniques for self-driving?