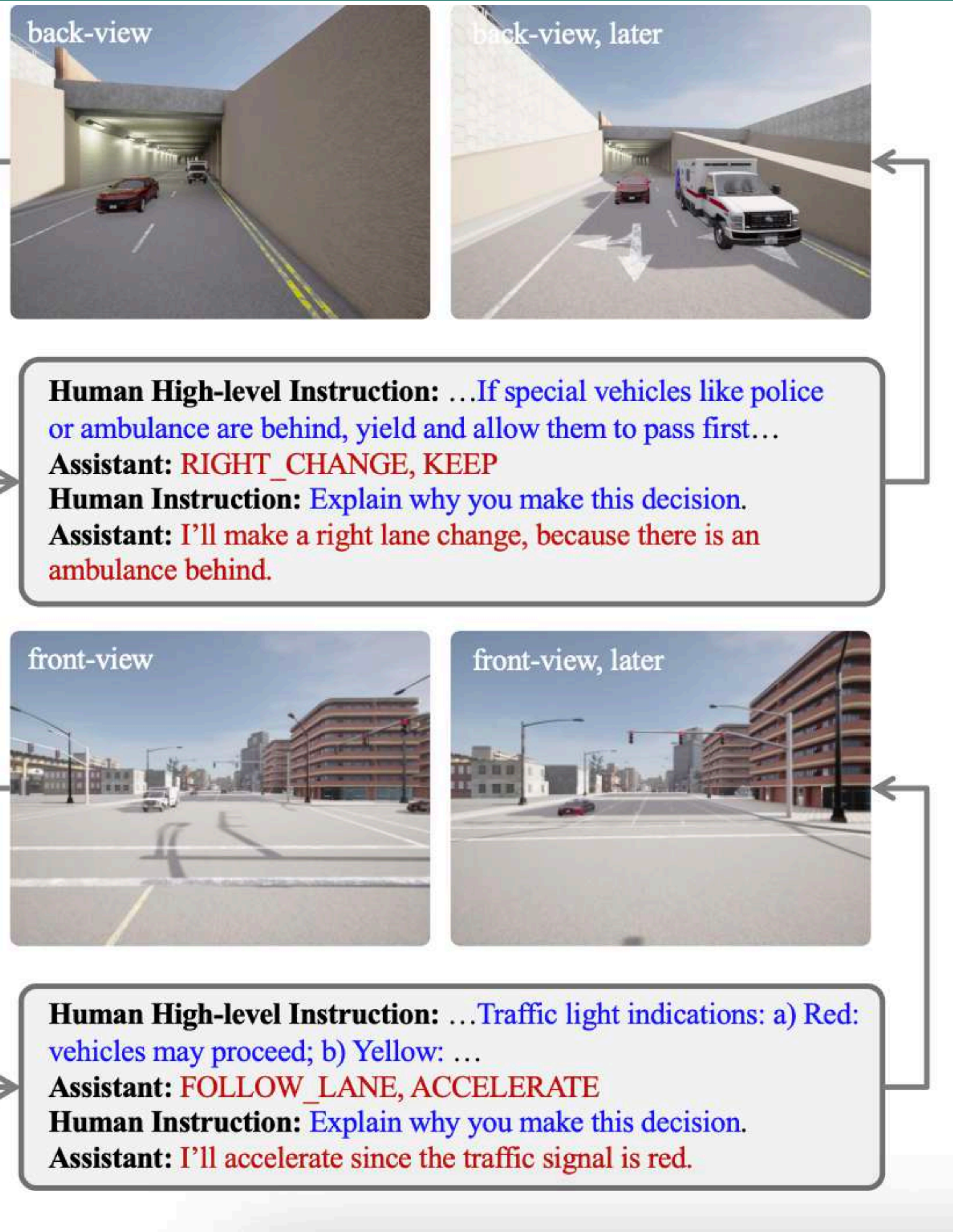
EMBODIED INTELLIGENCE FOR AUTONOMOUS SYSTEMS ON THE HORIZON

ETA: **E**fficiency through **T**hinking **A**head

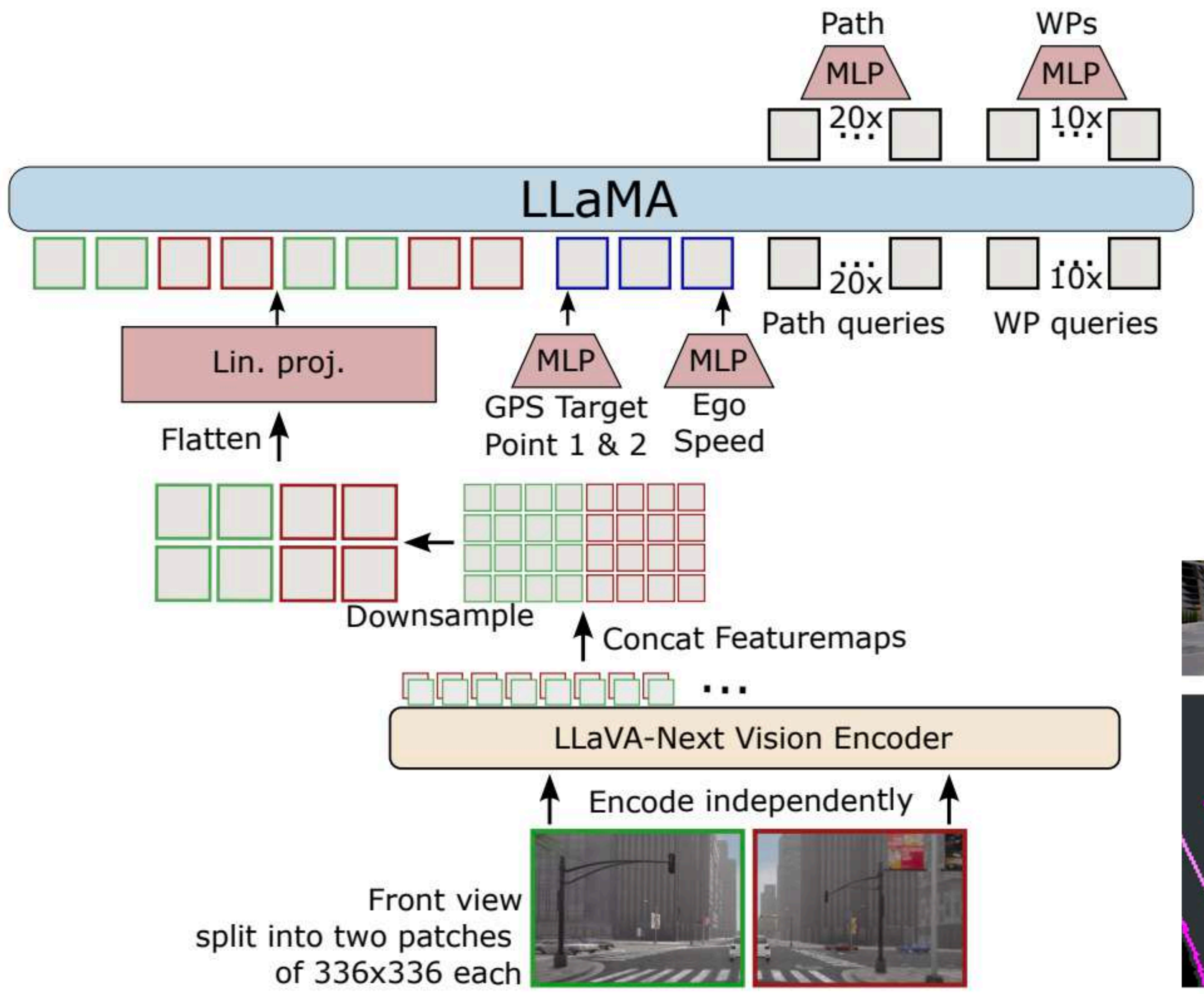
A Dual Approach to Self-Driving with Large Models

with Shadi Hamdan, Chonghao Sima, Zetong Yang, and Hongyang Li

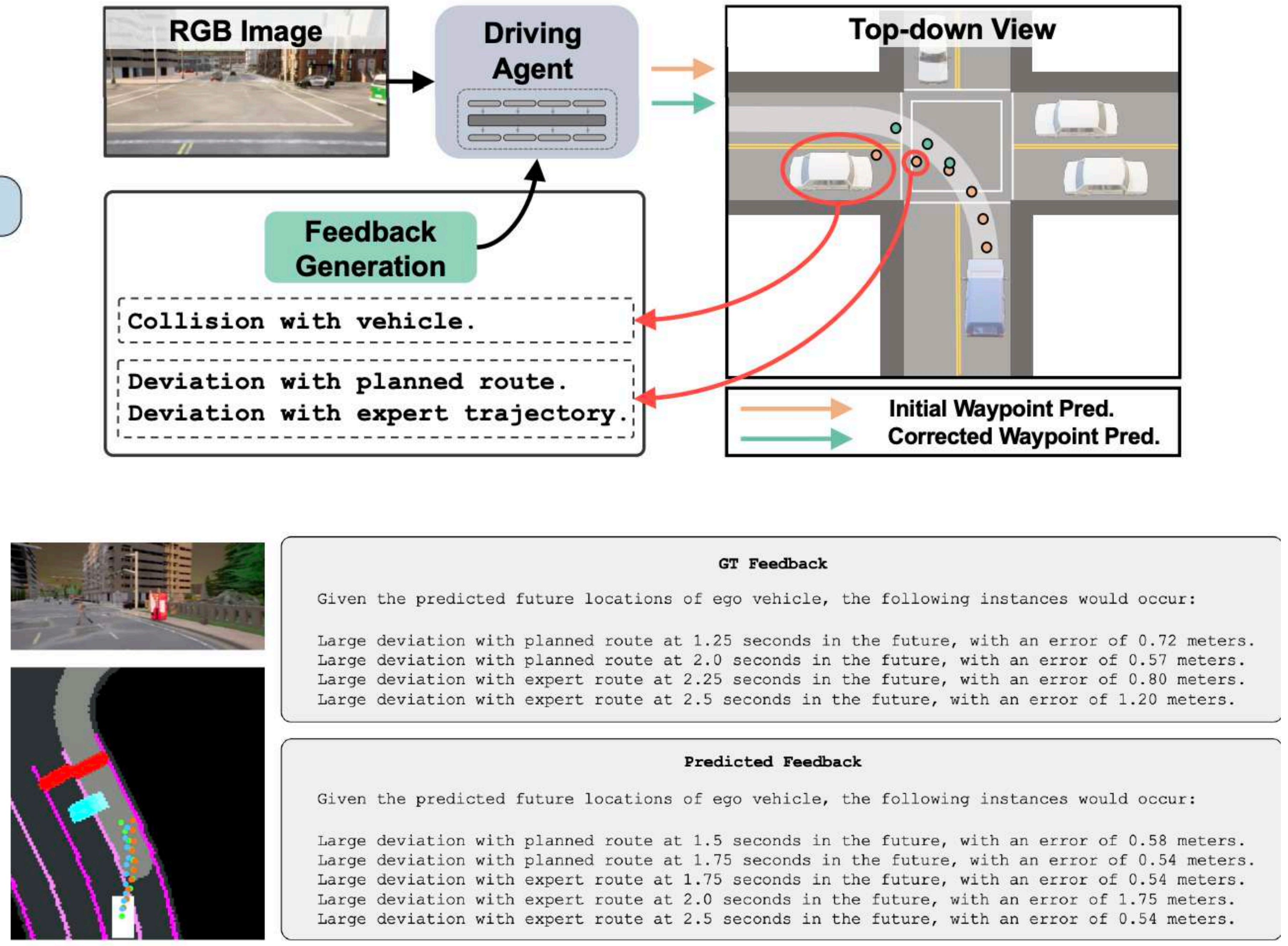
Large models are everywhere!



DriveMLM; 2023

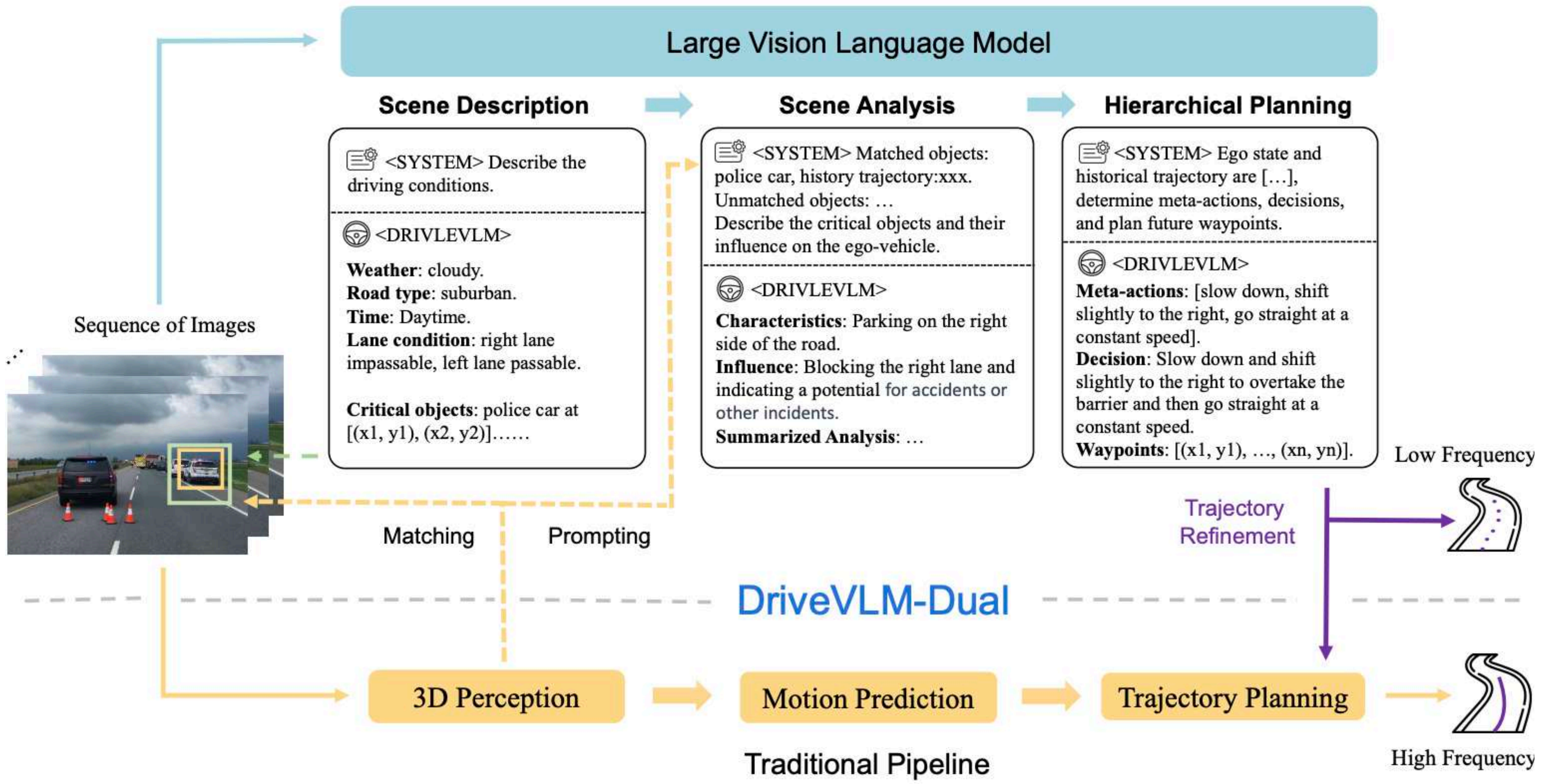


LLM4AD/CarLLaVA; 2024

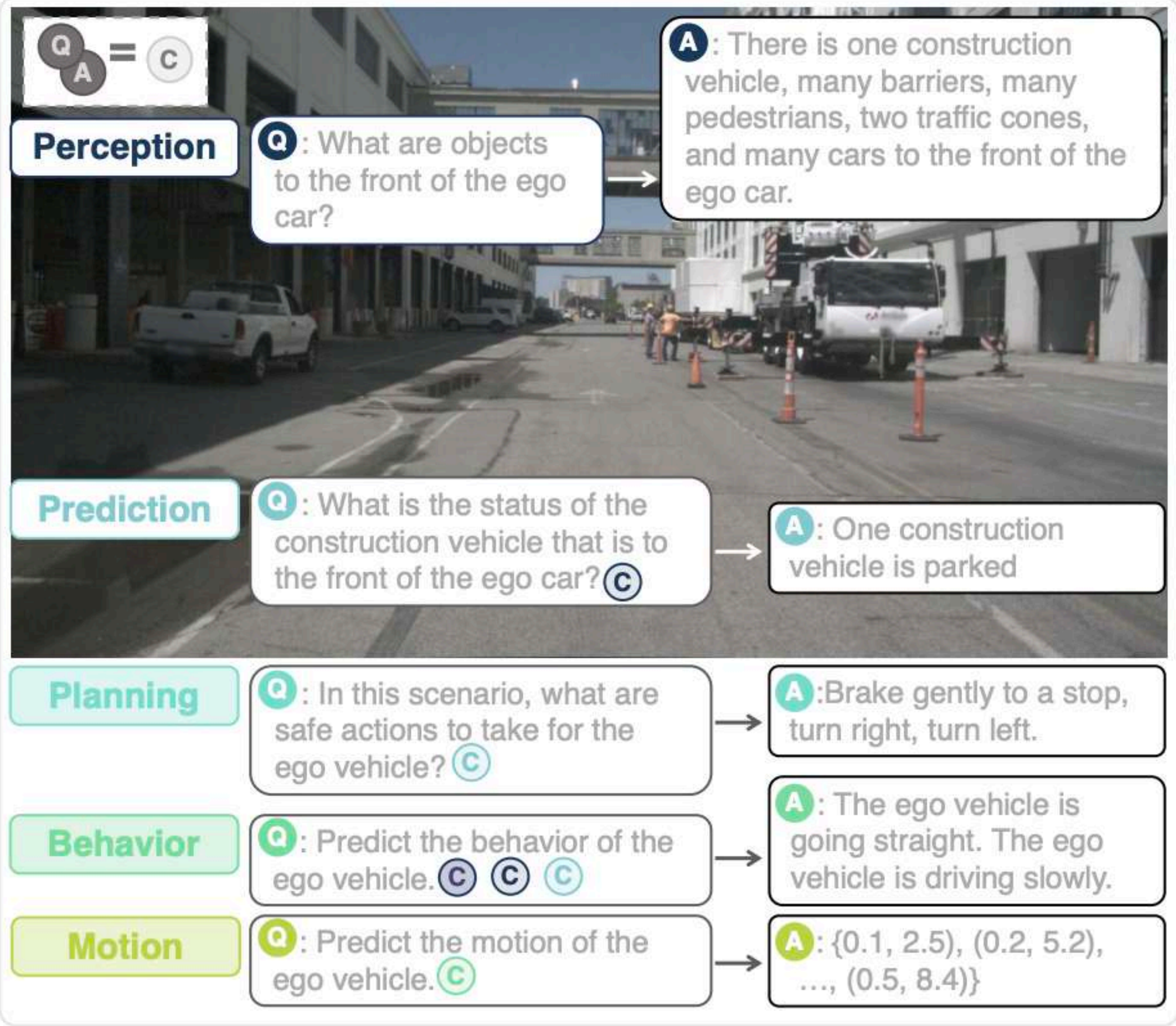


FeD; 2024

..but they are too slow.



DriveVLM; 2024

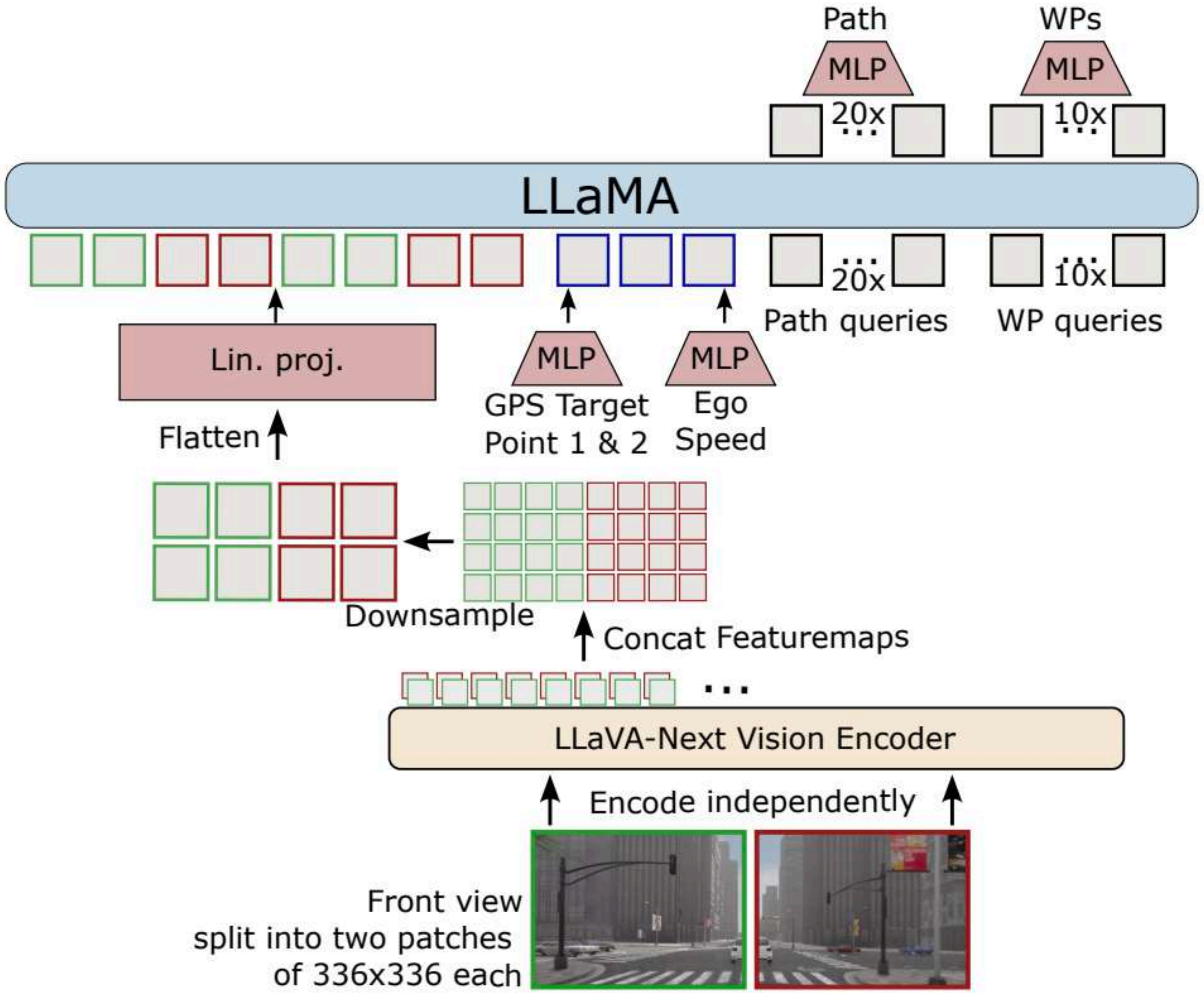


DriveLM; 2024

How to use large models efficiently for self-driving?

Take CARLA Leaderboard-v2 winner:

LLM4AD/ CarLLaVA

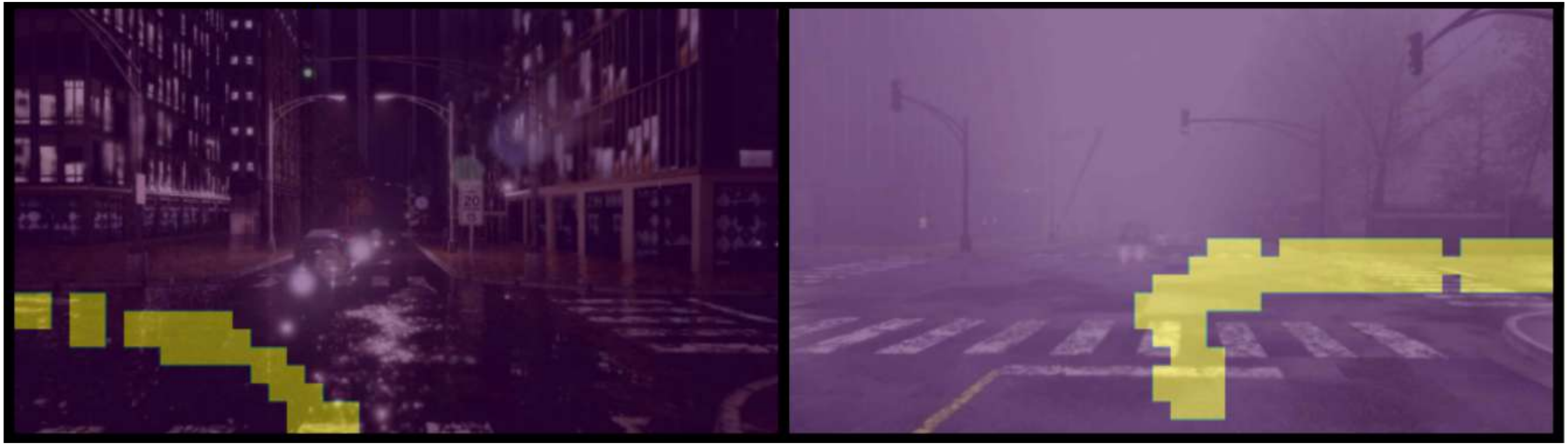


Action Mask

Better aligning actions with observations

■ Path

● Waypoints

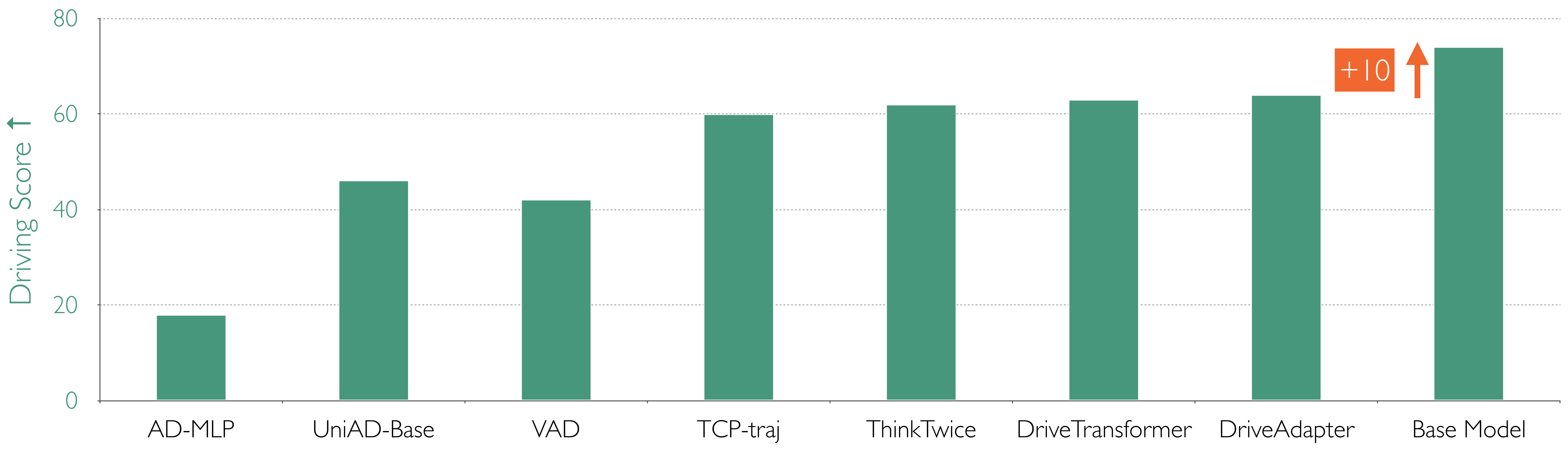


Action Mask

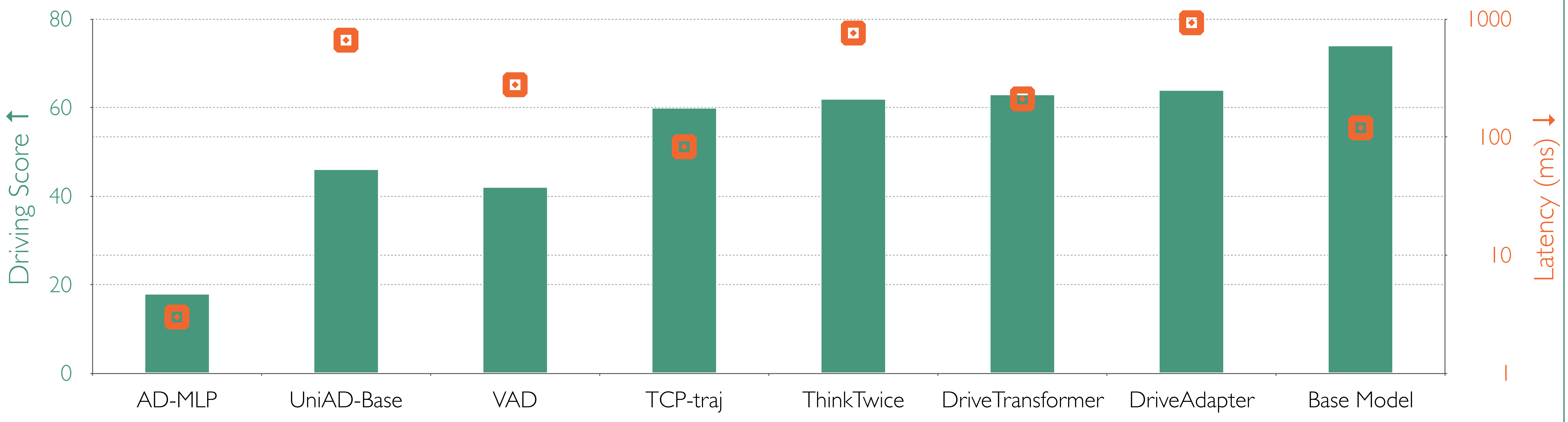
Better aligning actions with observations

Patches with path/waypoint

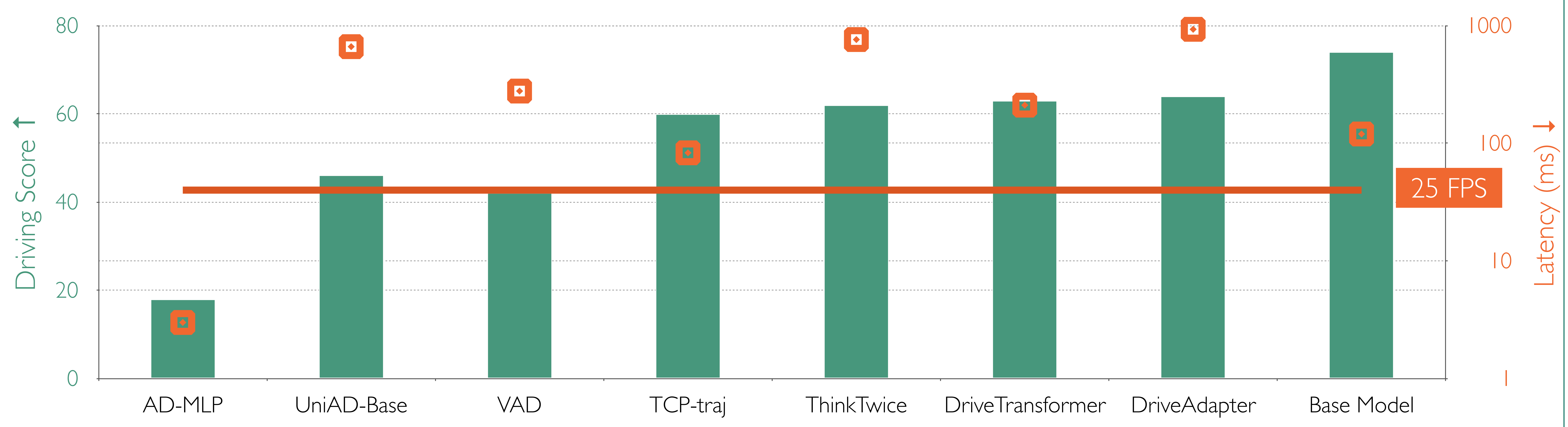
Results on Bench2Drive



Results on Bench2Drive

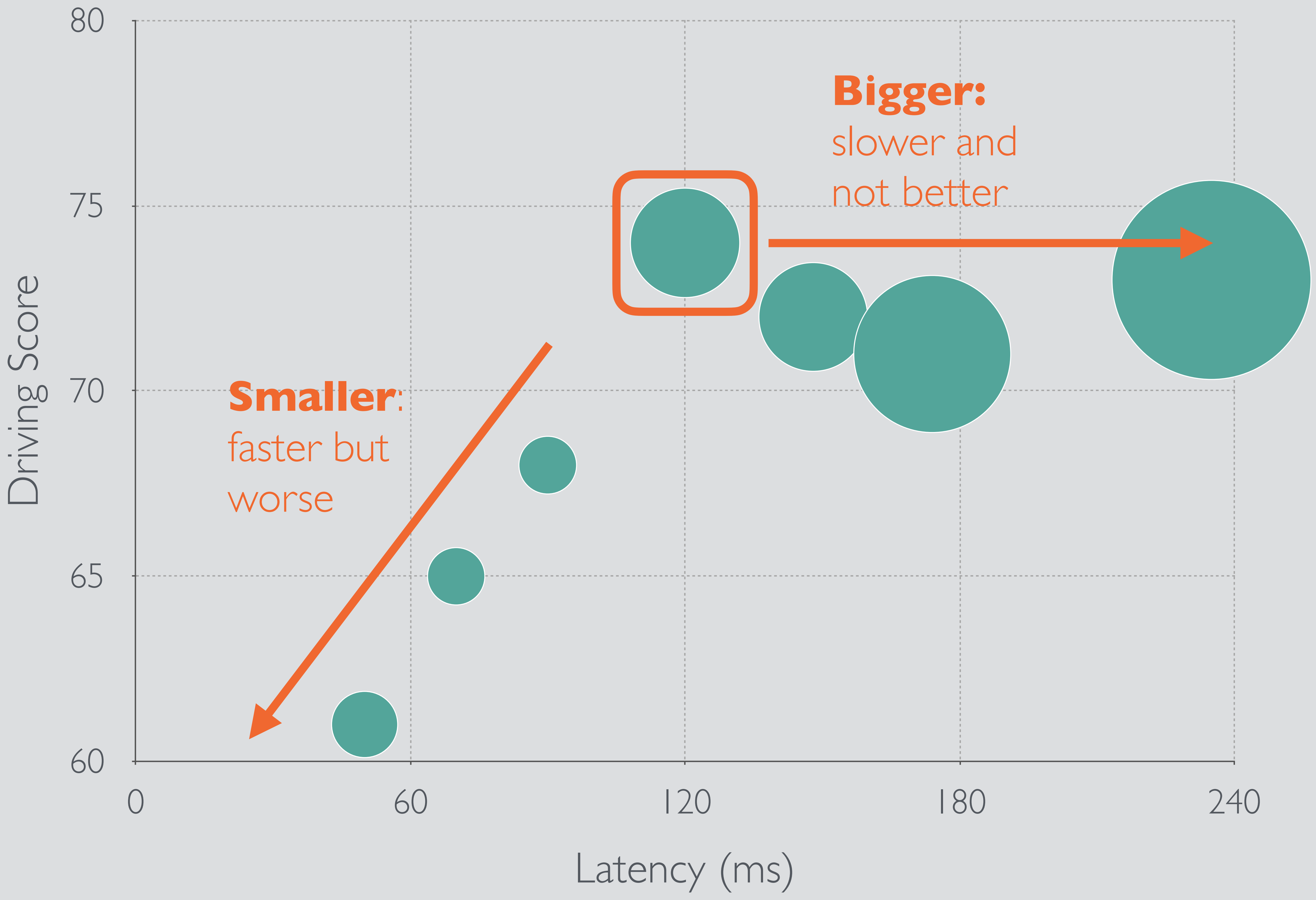


Results on Bench2Drive



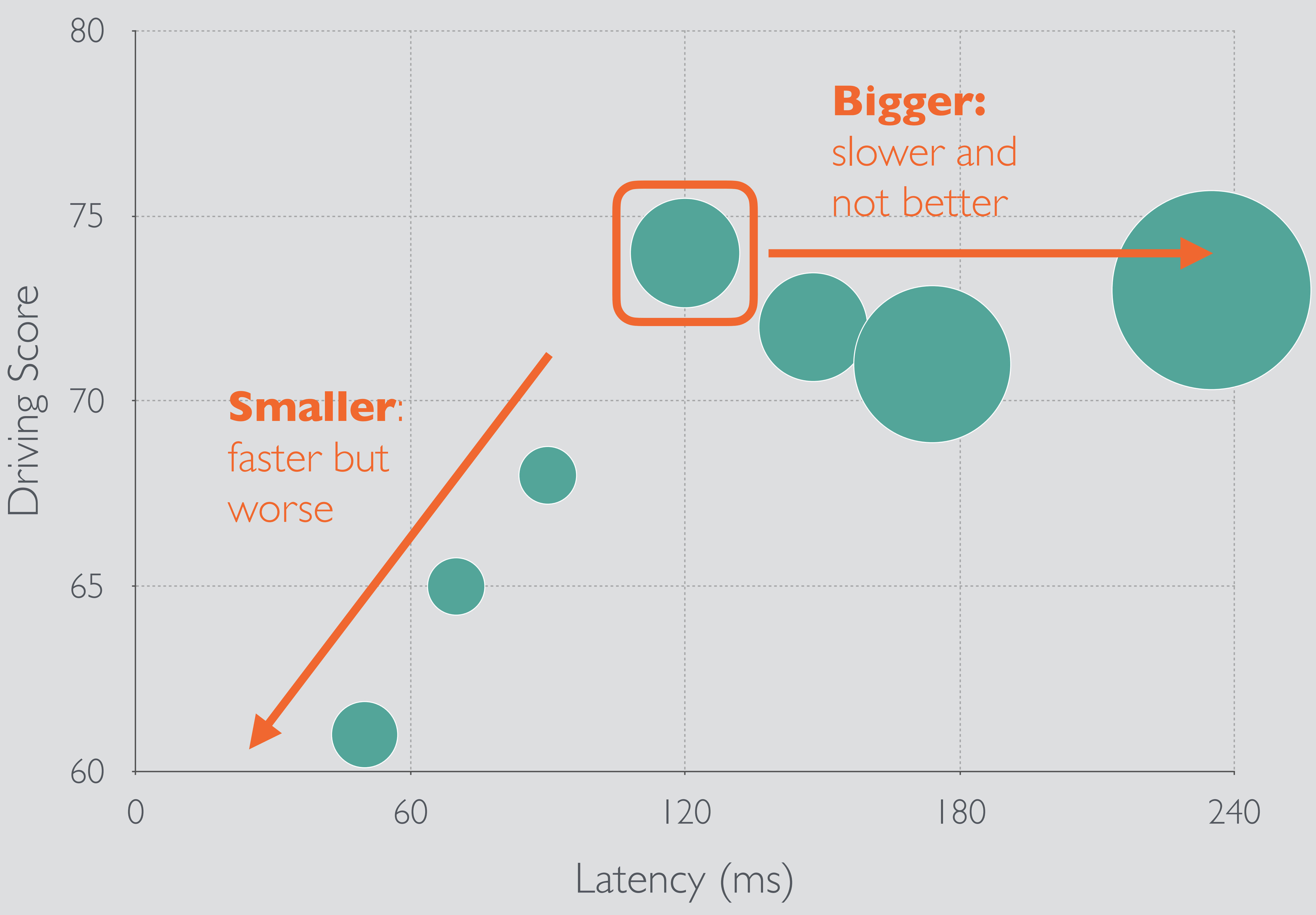
What about different large models?

- **Smaller**: faster but worse performance
- **Bigger**: slower and not better

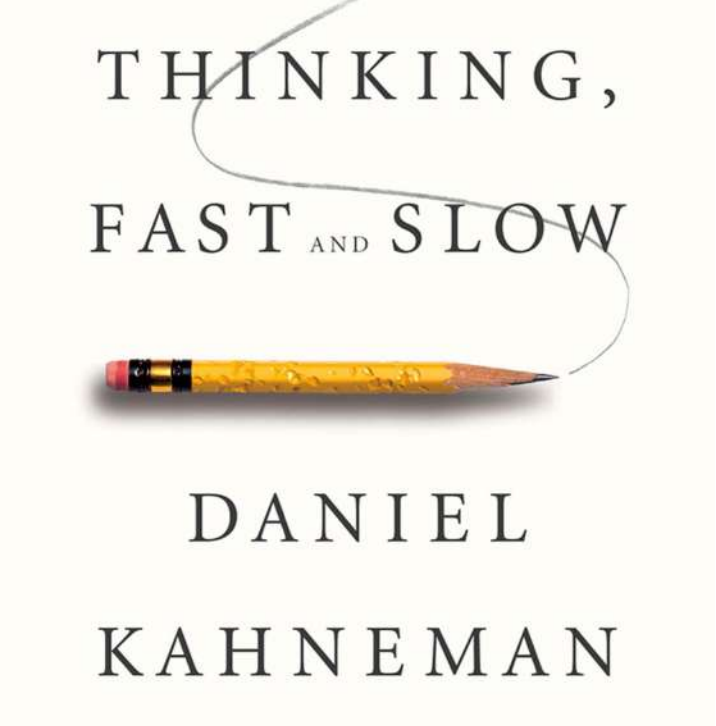


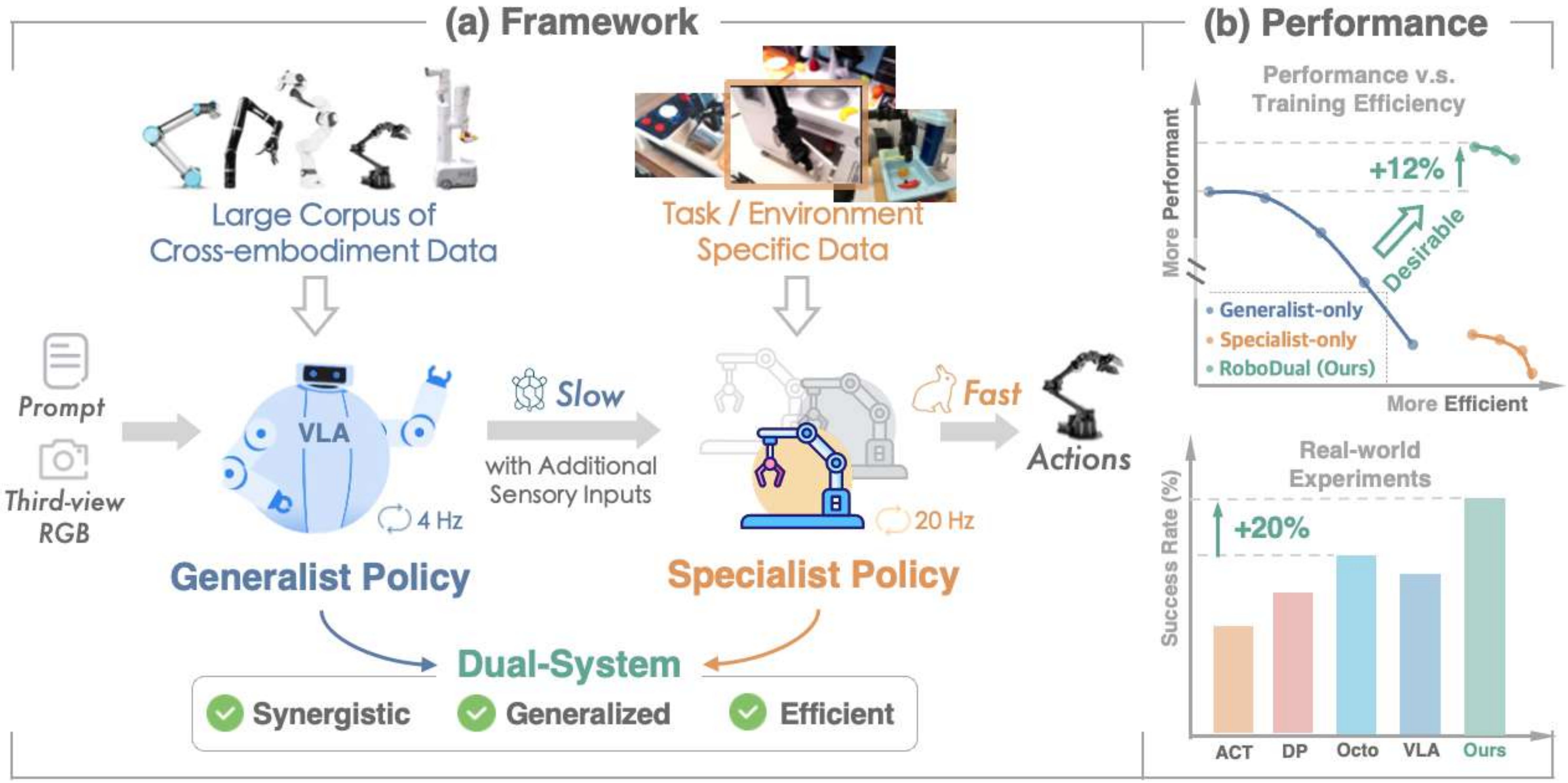
What about different large models?

- **Smaller**: faster but worse performance
- **Bigger**: slower and not better
  - Underfitting?
  - Upperbounded by expert?

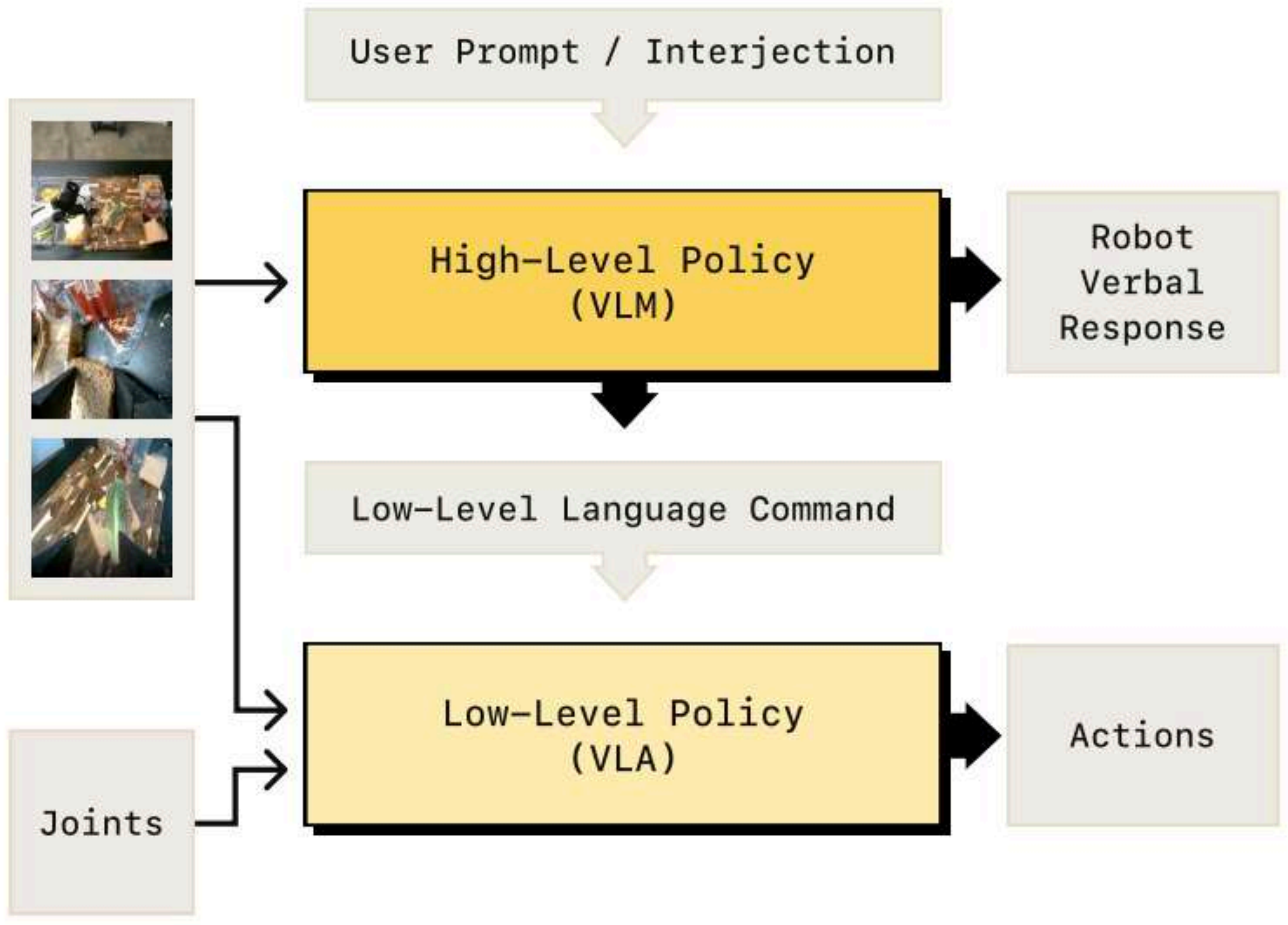


Dual approach in embodied systems



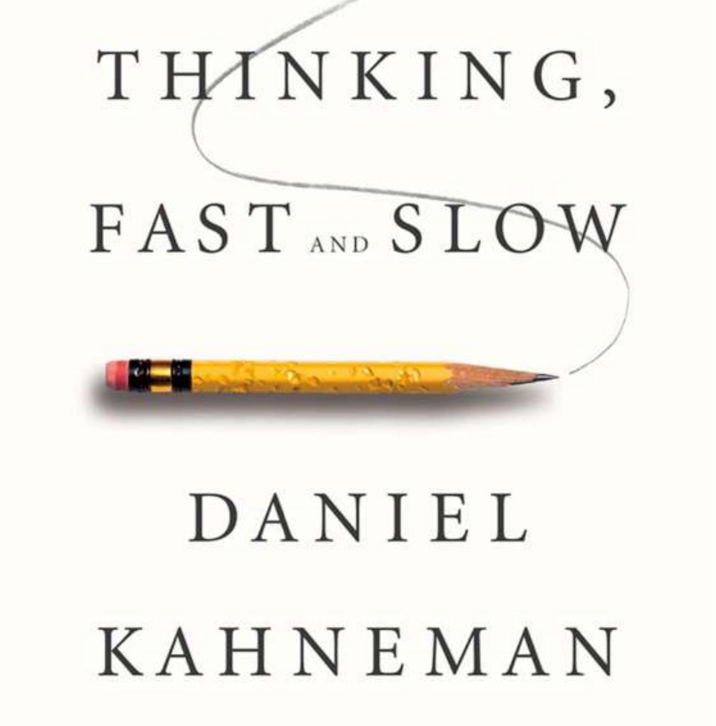


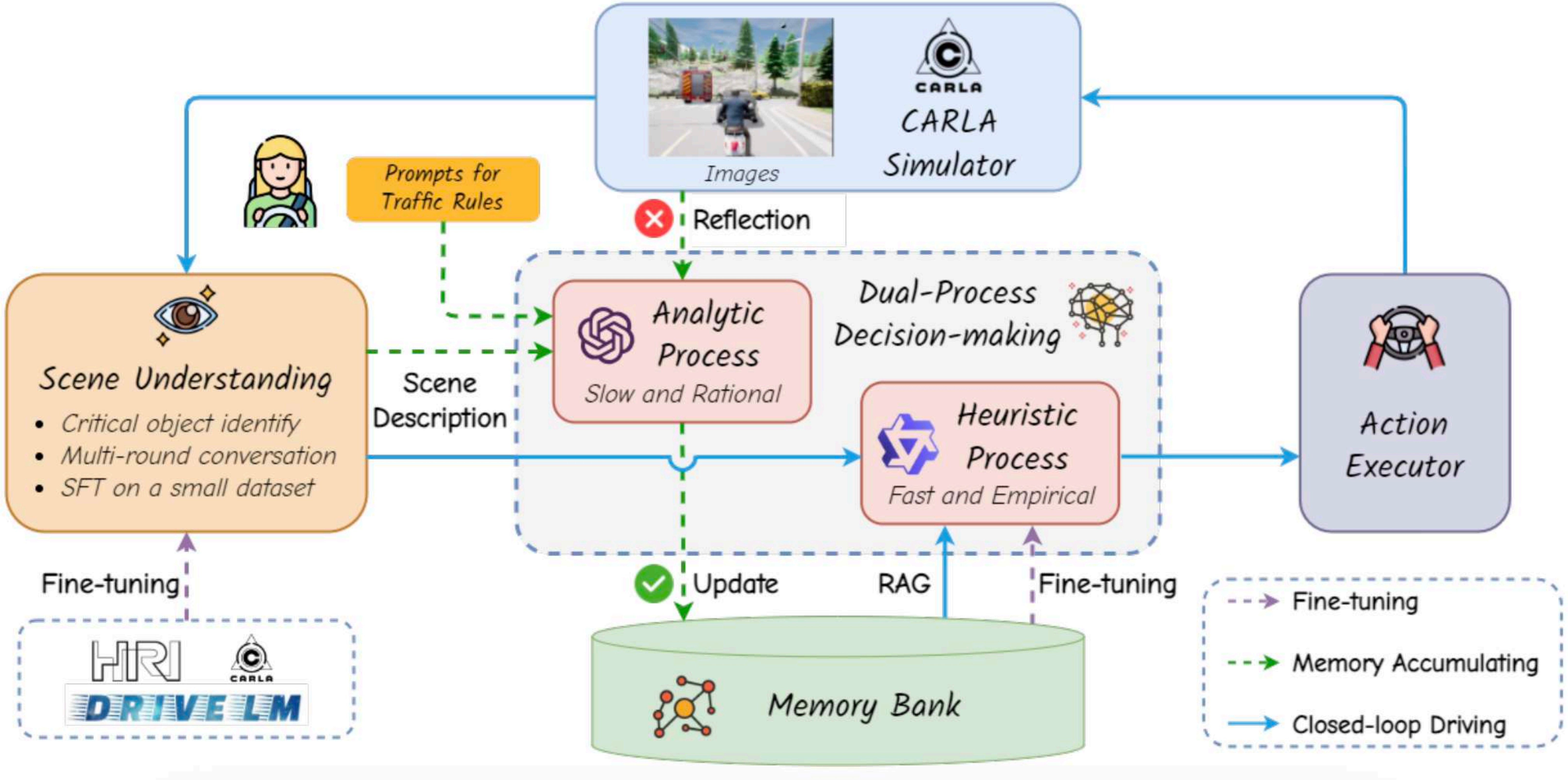
RoboDual; 2024



Hi Robot; 2025

Dual approach in driving

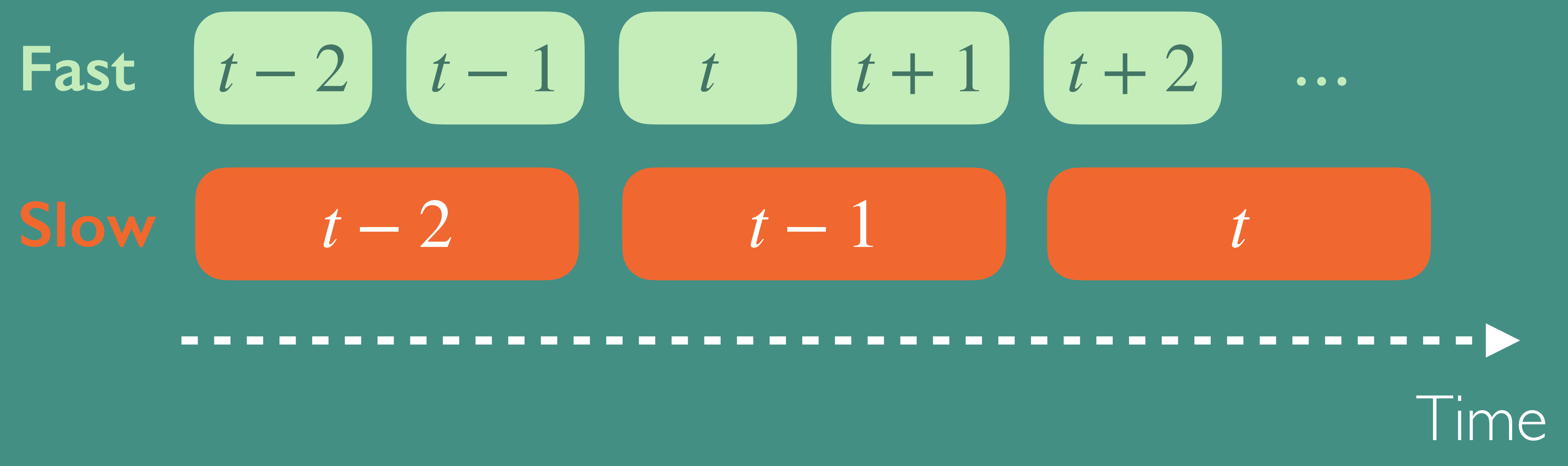




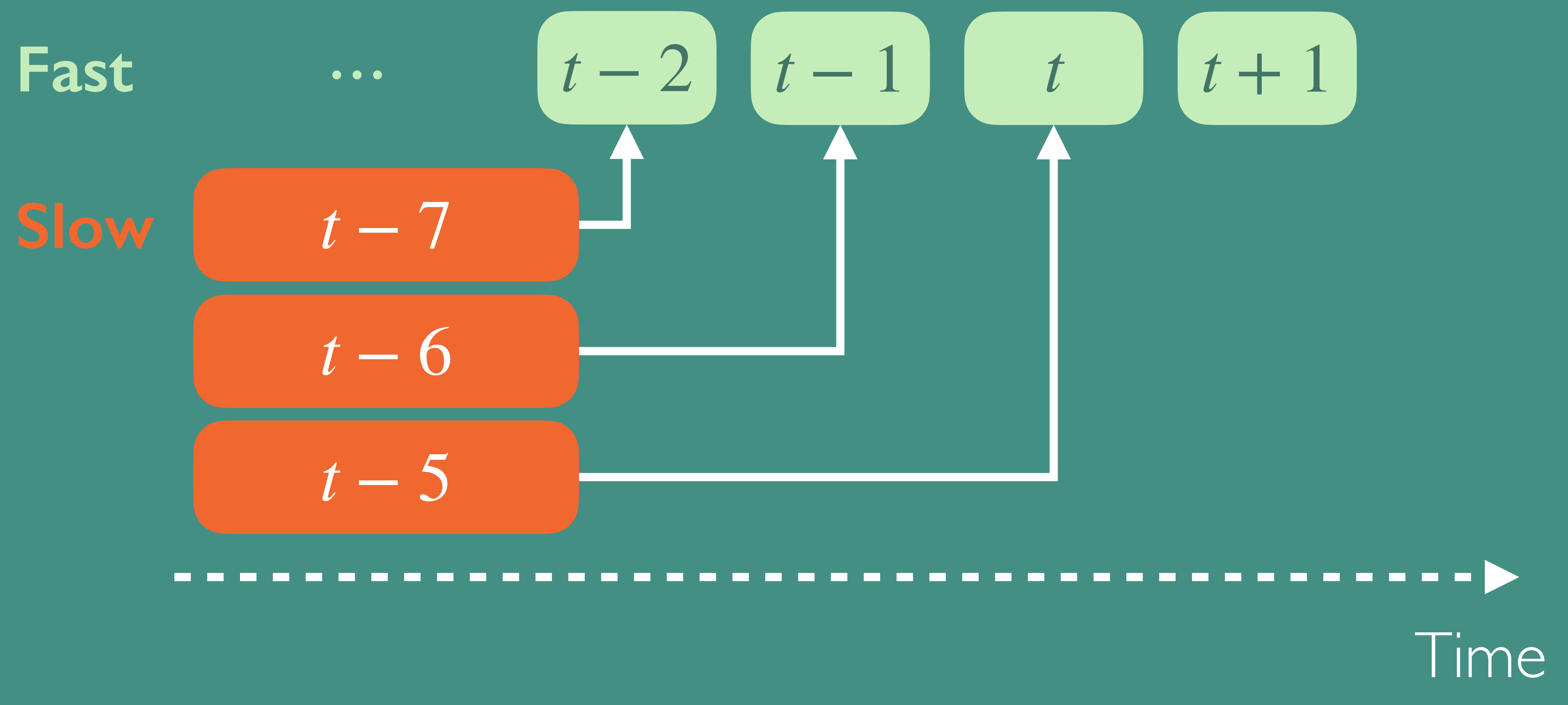
LeapAD; 2024

Dual approach

Typical Dual Paradigm

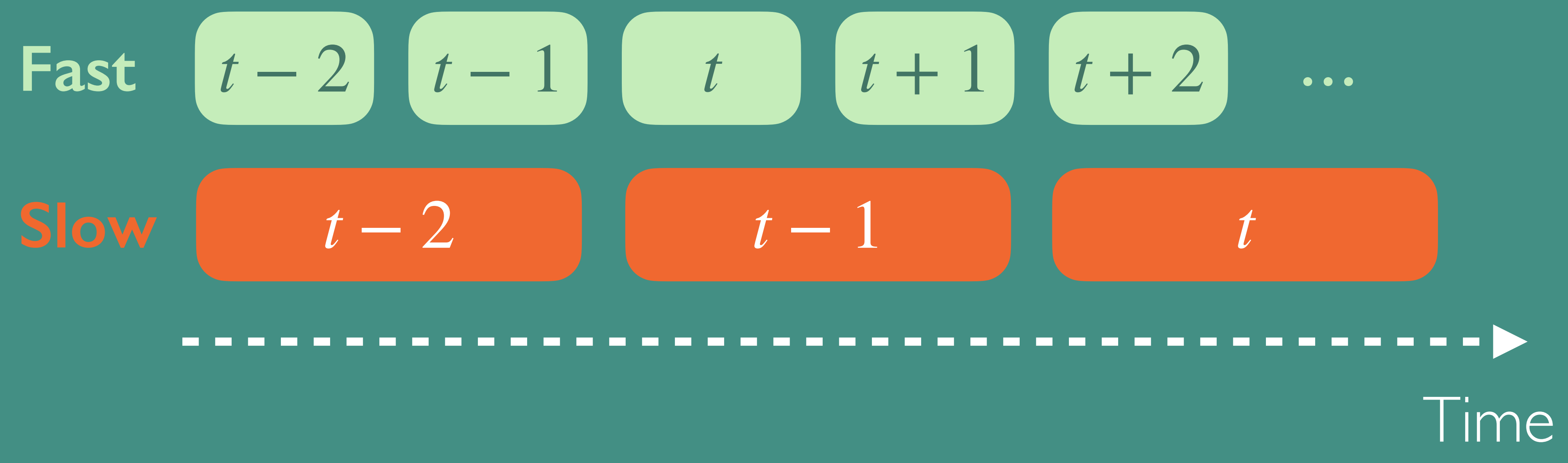


Efficiency through **T**hinking **A**head

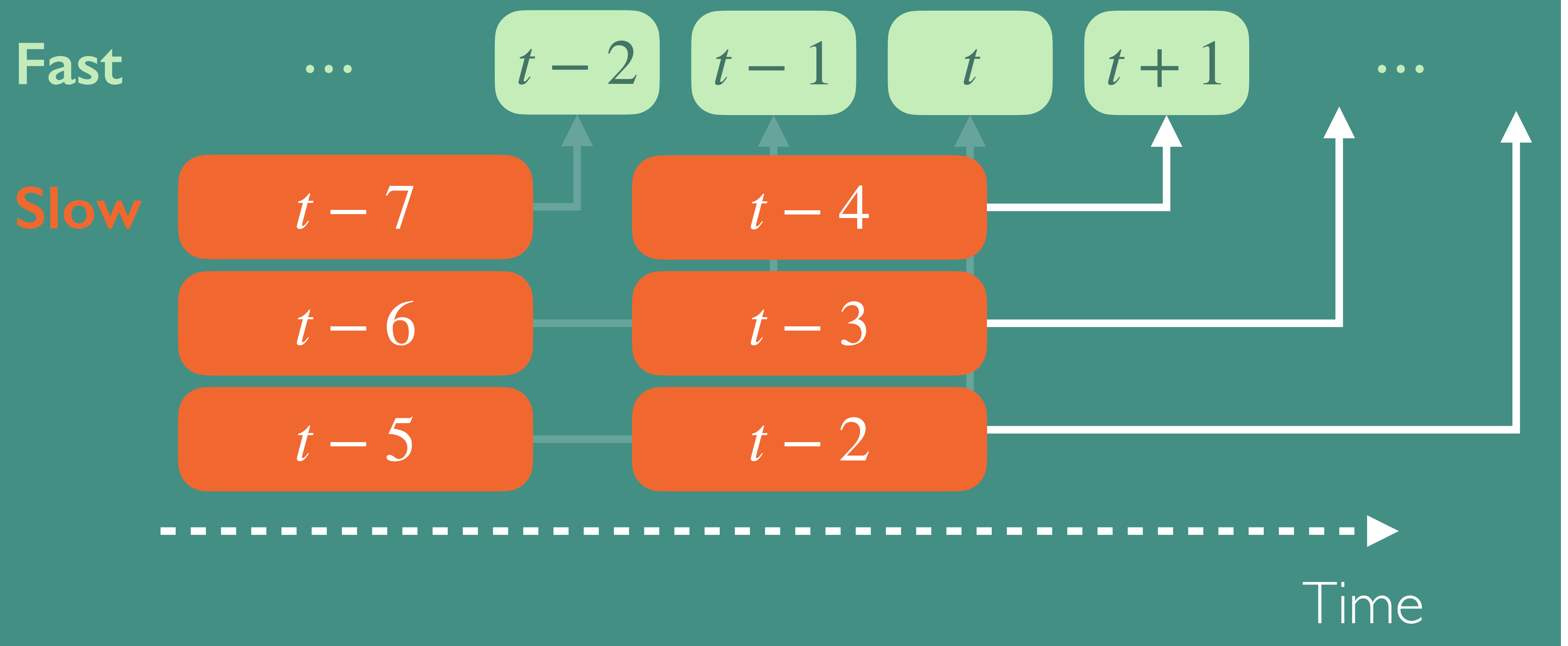


Dual approach

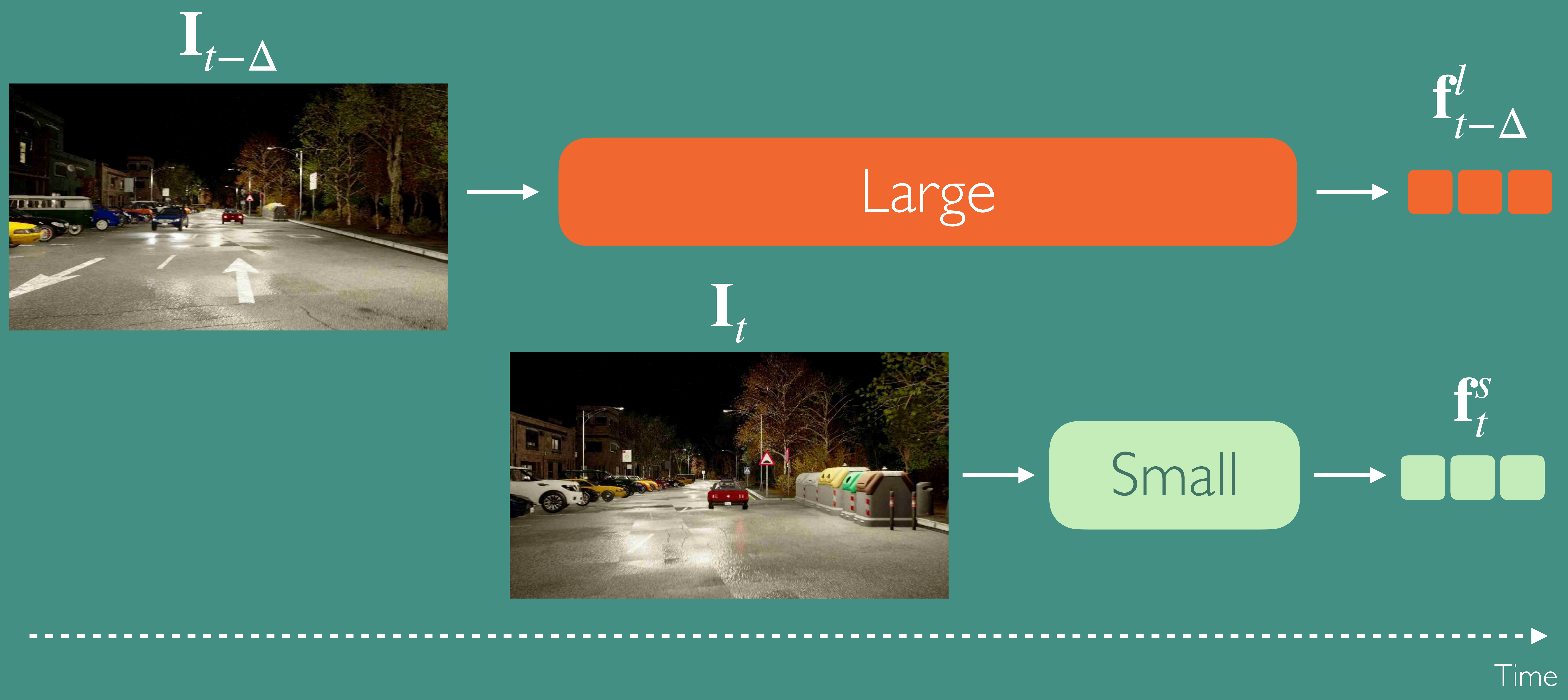
Typical Dual Paradigm



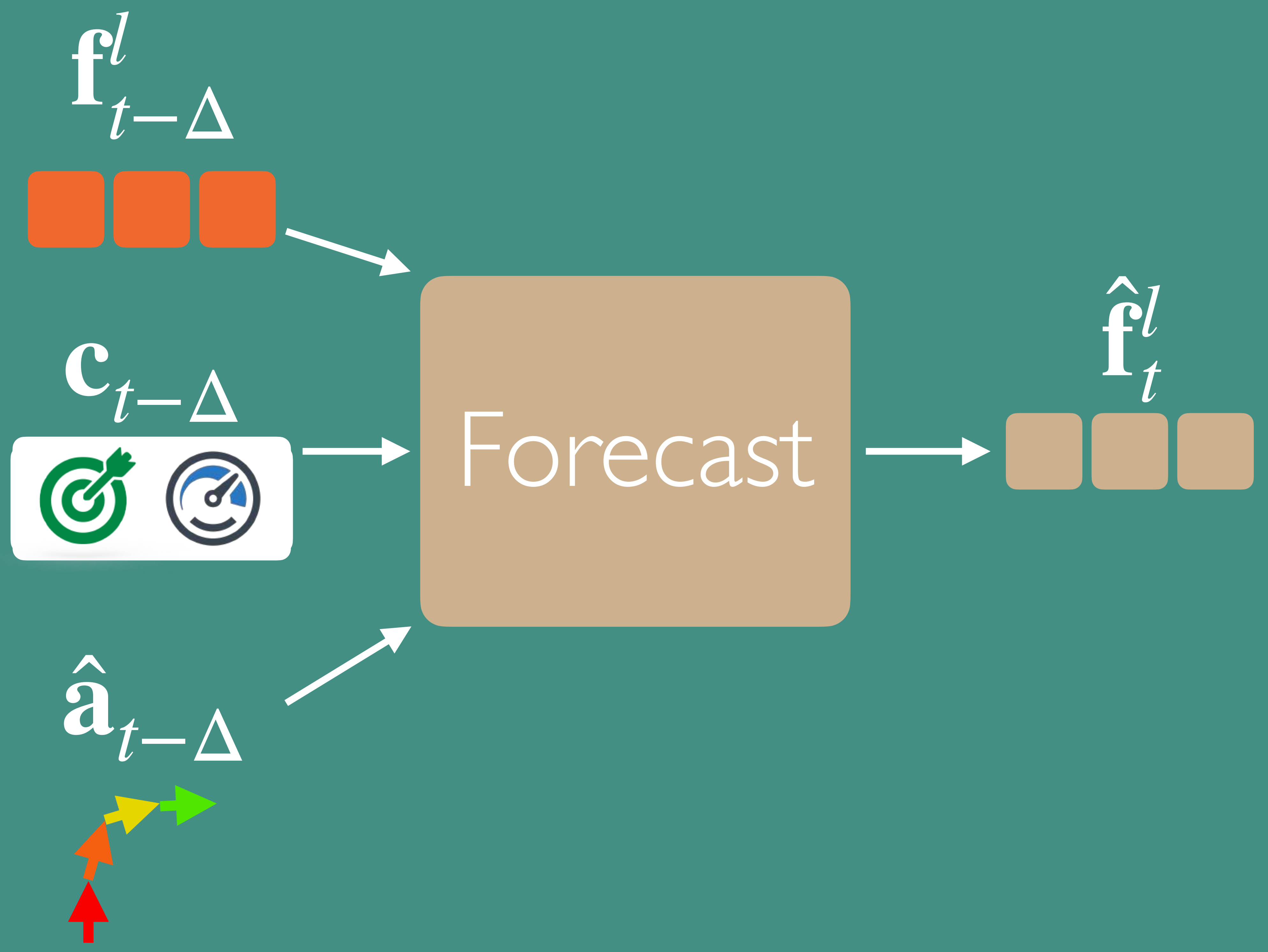
Efficiency through **T**hinking **A**head



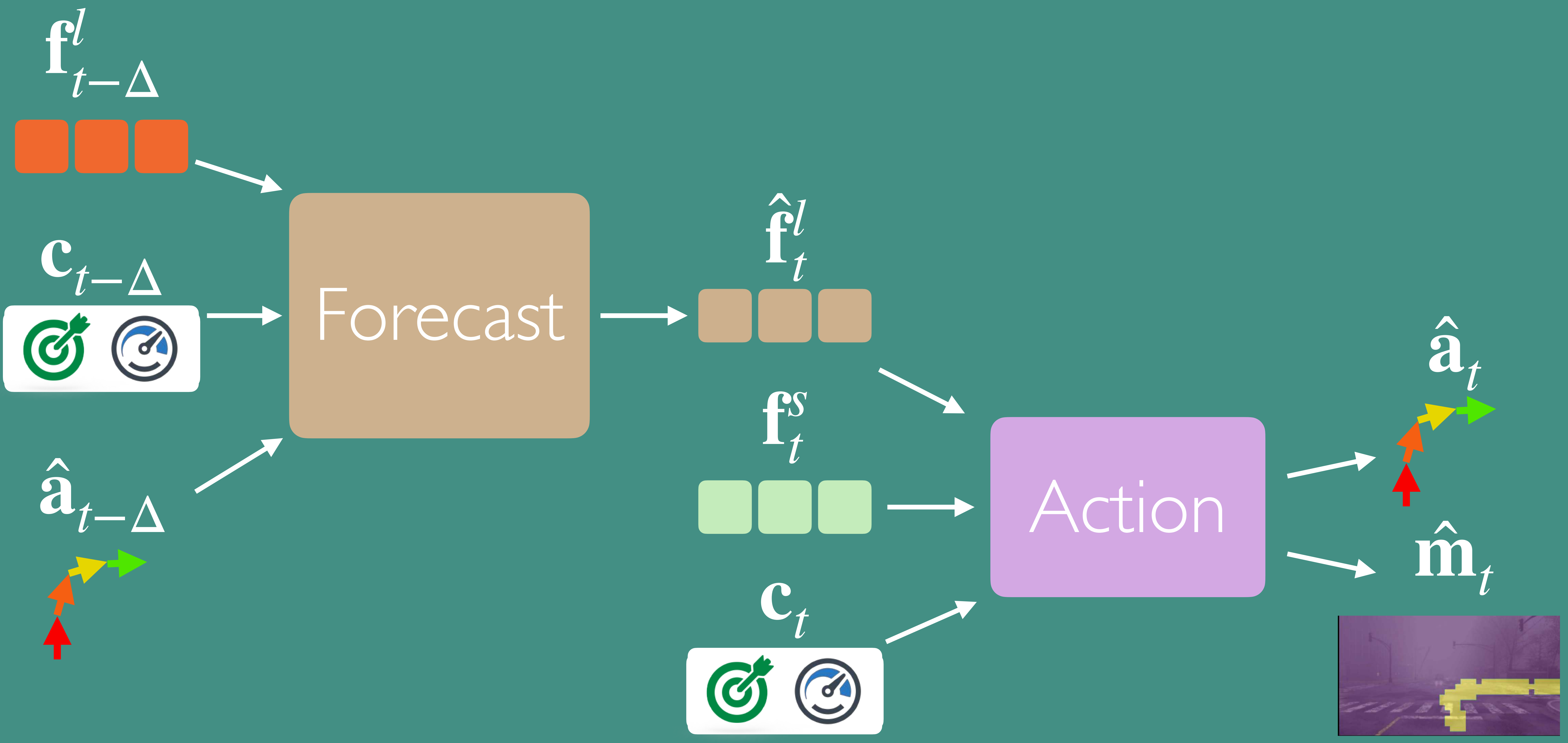
ETA: Async model



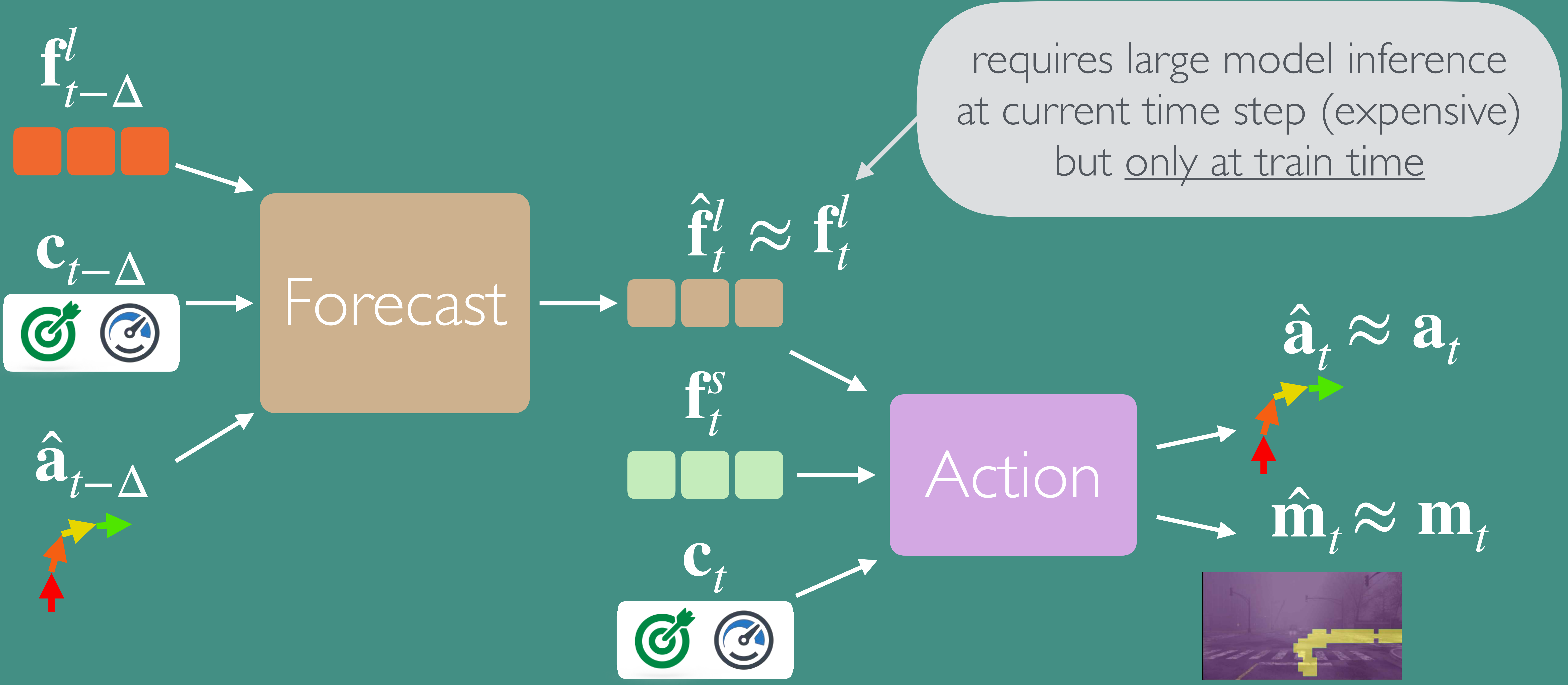
ETA: Async model



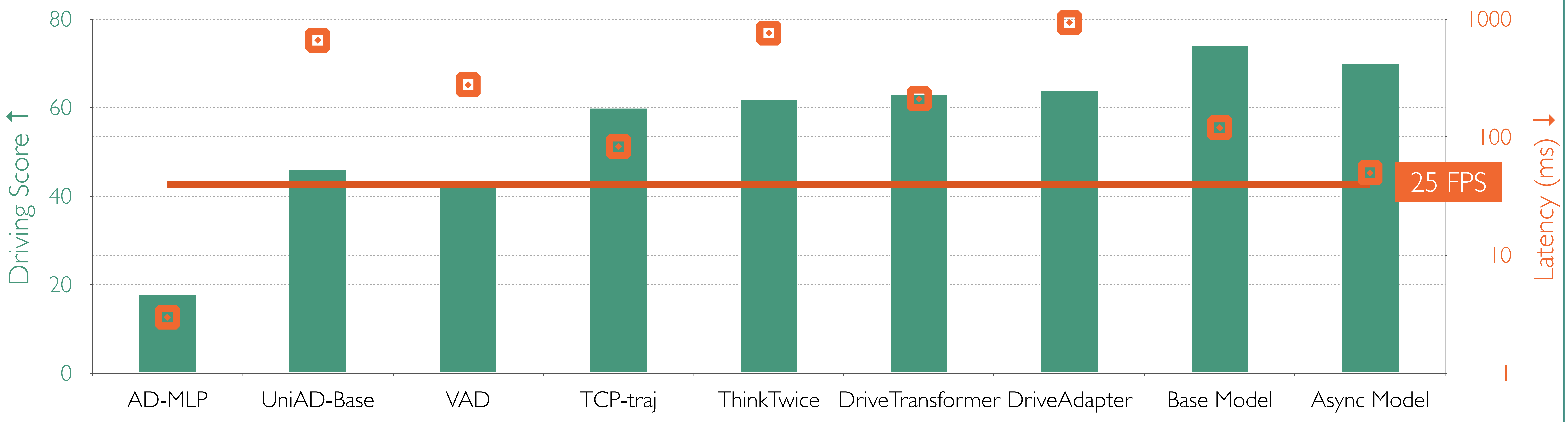
ETA: Async model



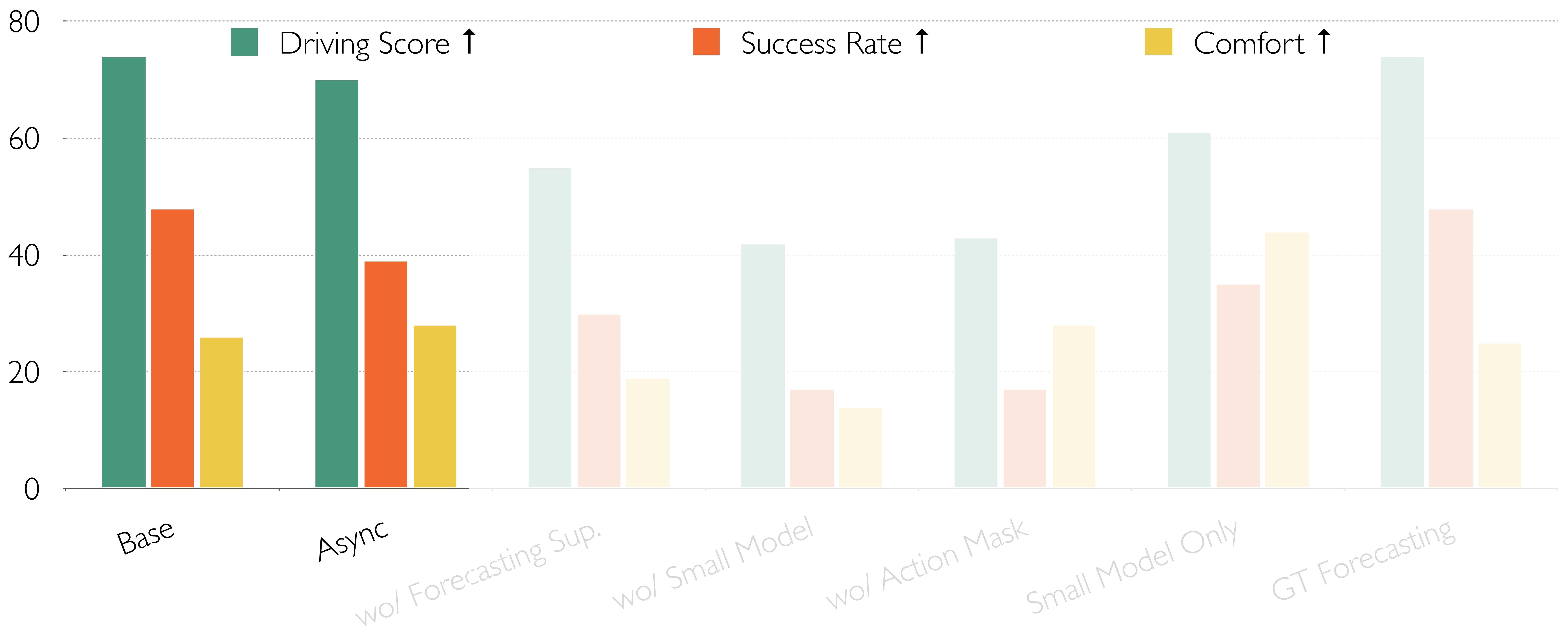
ETA: Async model



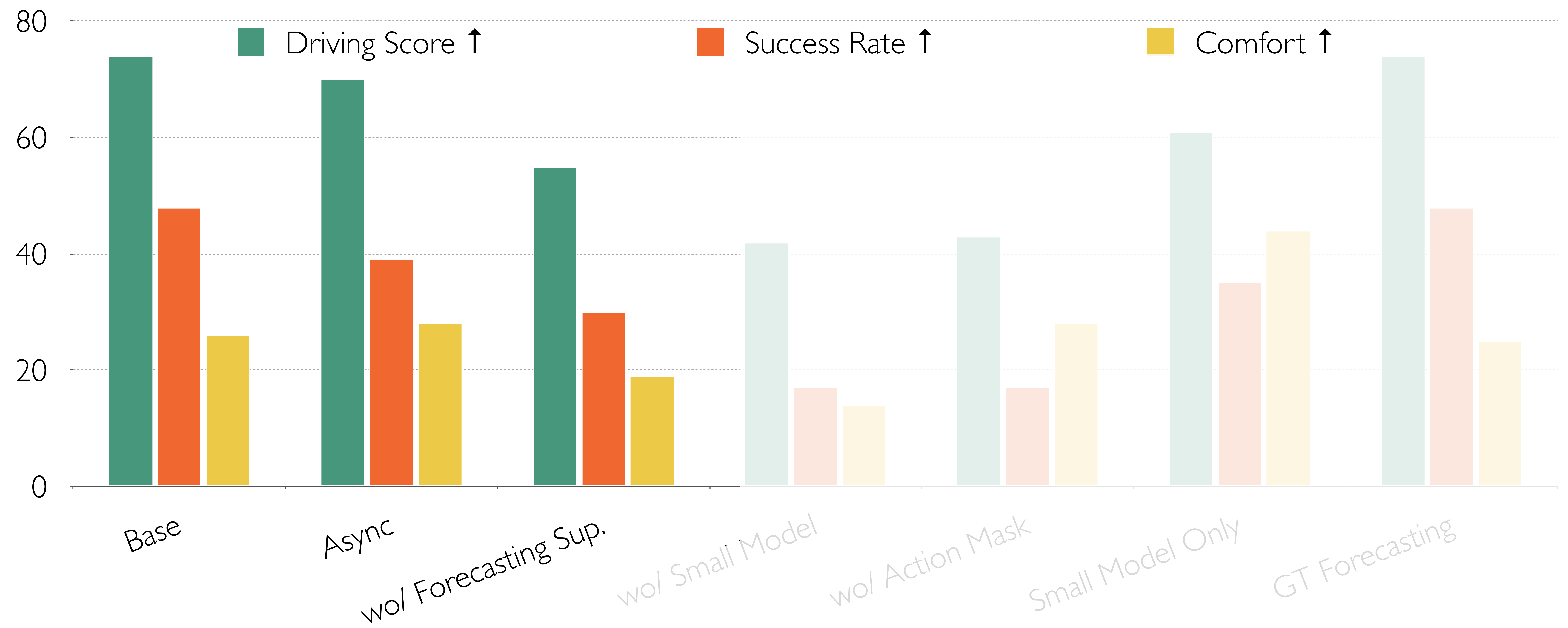
Results on Bench2Drive



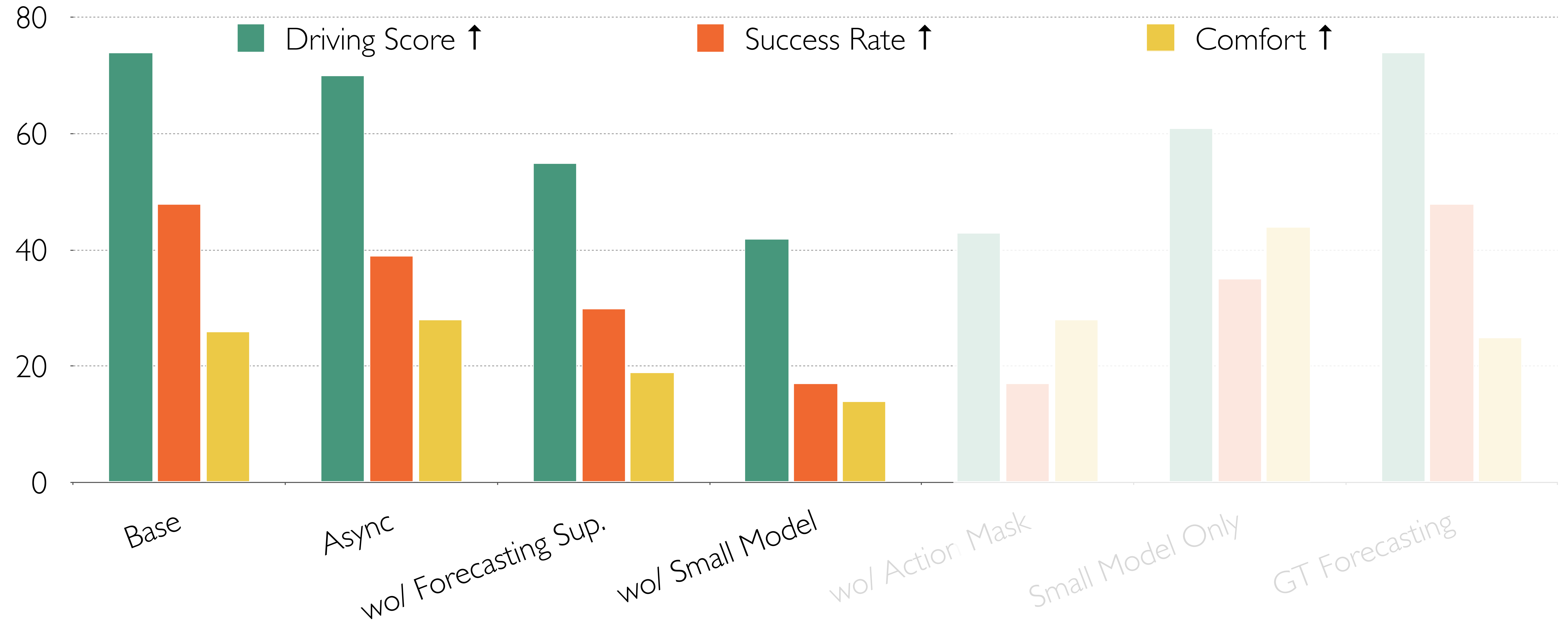
How does it work?



How does it work?

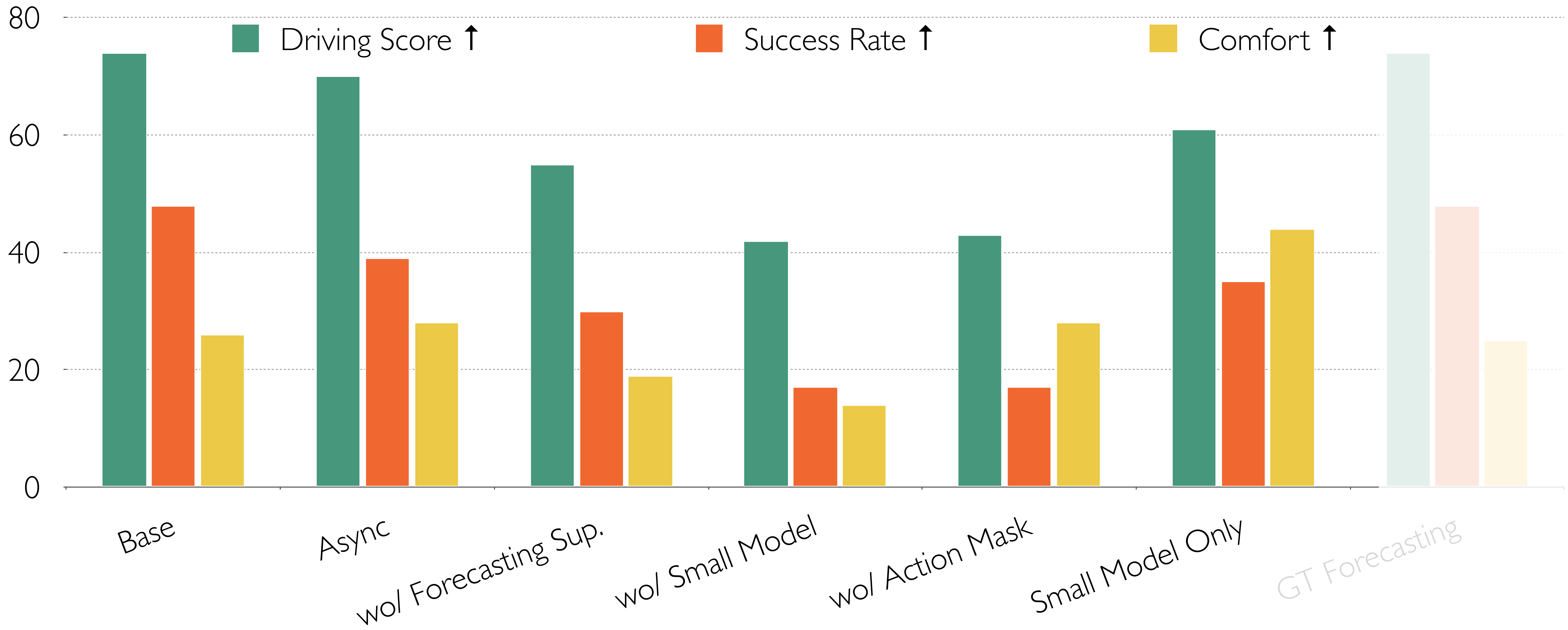


How does it work?

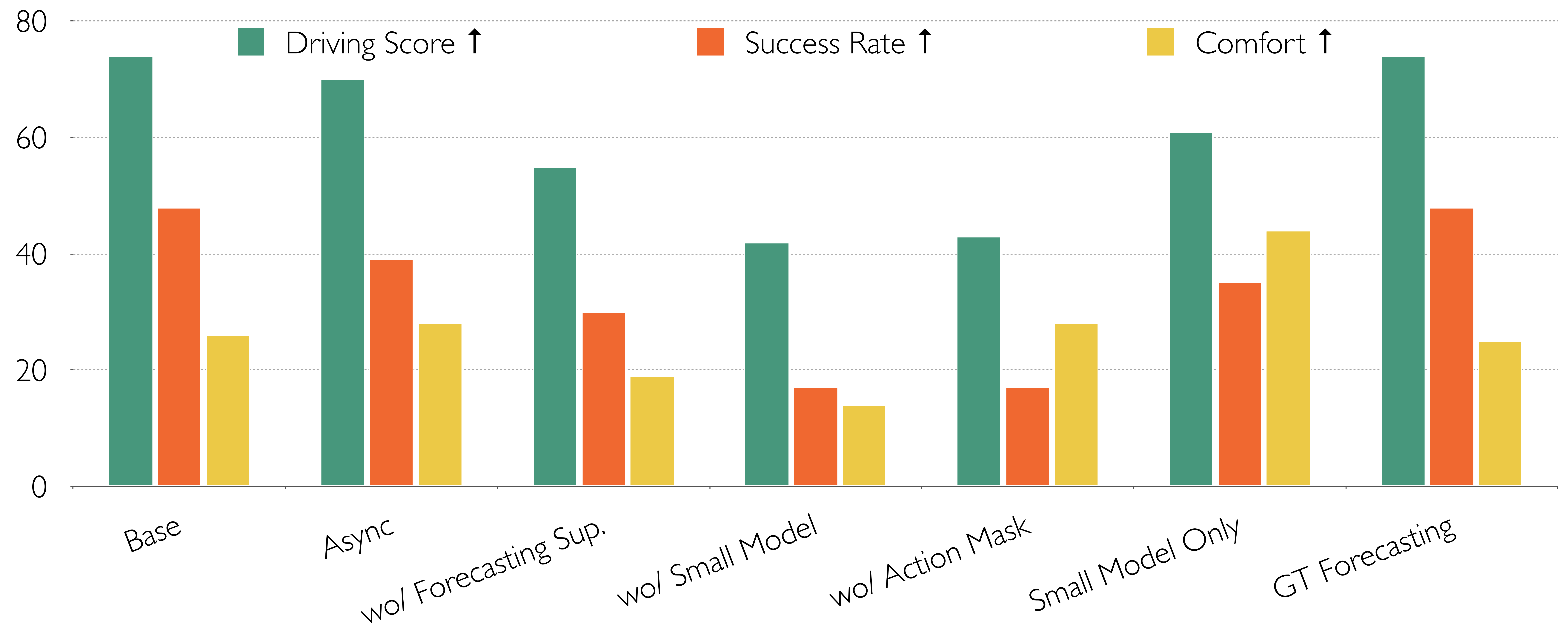


How does it work?

How does it work?



How does it work?



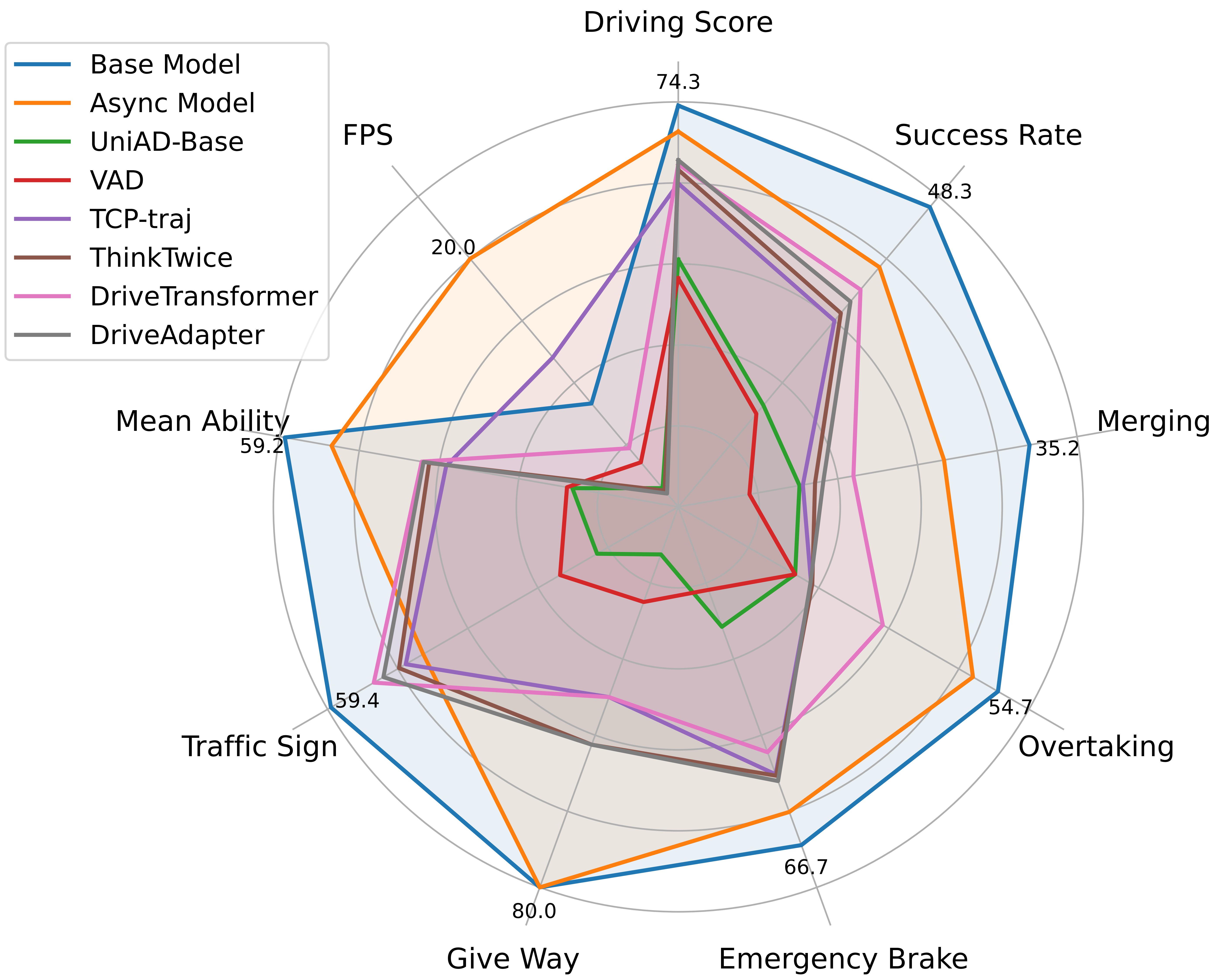
Specific skills

Async ≈ Base in:

- Overtaking
- Emergency braking
- Give way

Large gaps in:

- Traffic sign handling
- Merging



What is missing?

- Improving forecasting
- Figuring out how to use larger vision encoders
- Trying different experts, i.e. PDMLite
- Improving run-time further

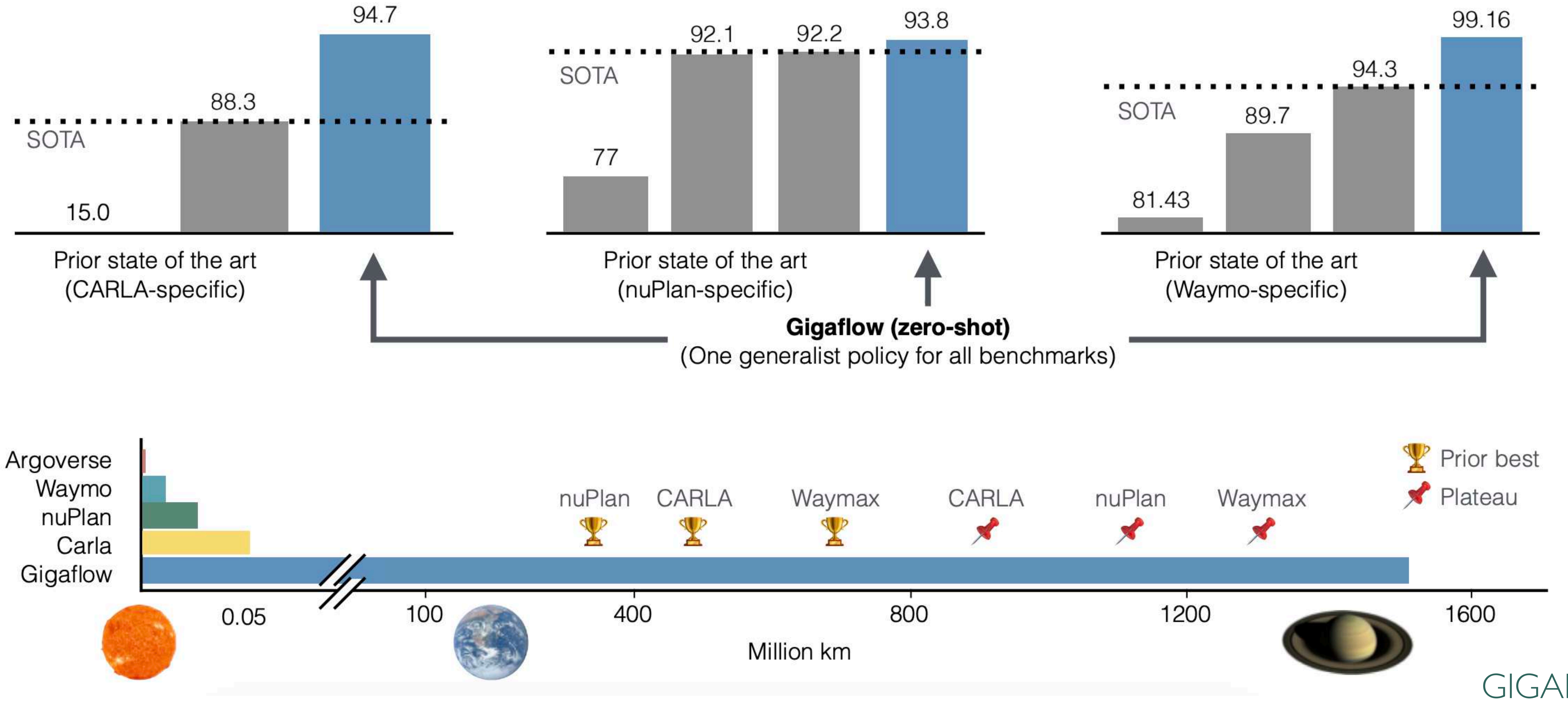
EMBODIED INTELLIGENCE FOR AUTONOMOUS SYSTEMS ON THE HORIZON

RELACS:

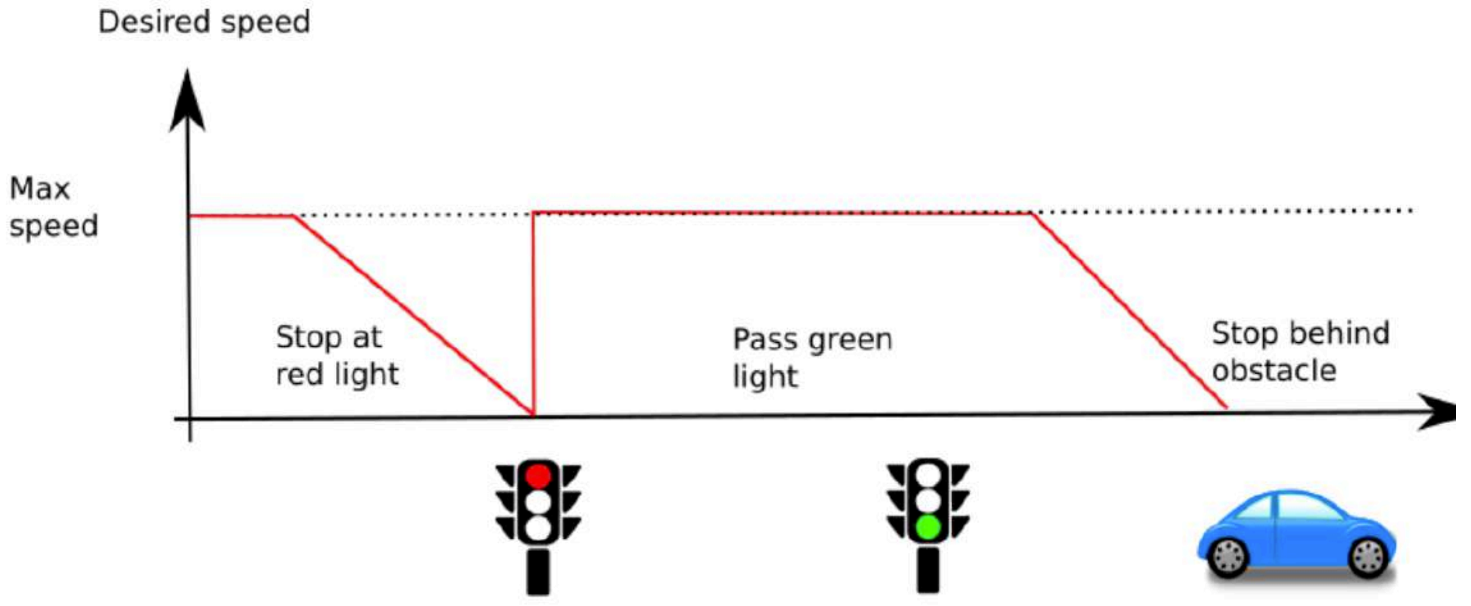
Reward Learning for Autonomous Driving using Counterfactuals

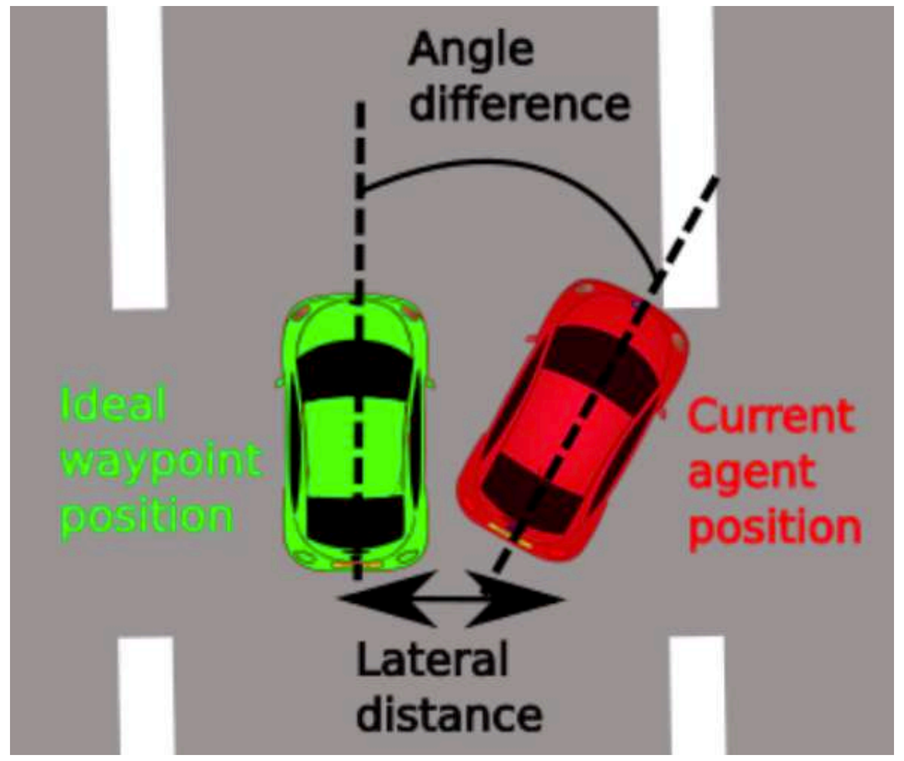
with Eray Çakar, Shadi Hamdan, Jiazhi Yang, Tianyu Li, Caojun Wang, and Hongyang Li

Increasing importance of planning



GIGAFLOW; 2025



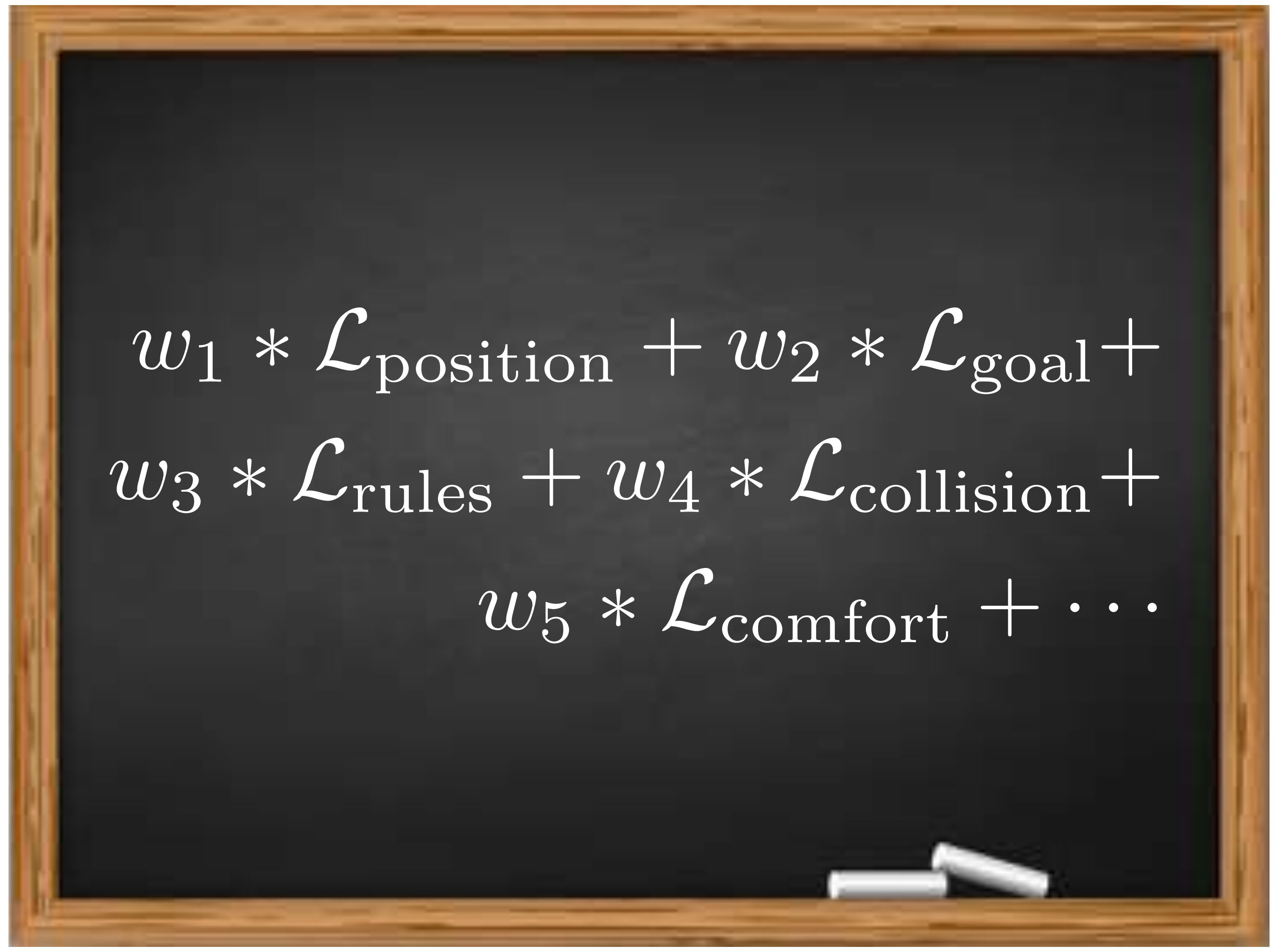


Hand-designed rewards

Implicit Affordances, ROACH

Ground truth measurements

...



Hand-designed rewards

Implicit Affordances, ROACH

Ground truth measurements

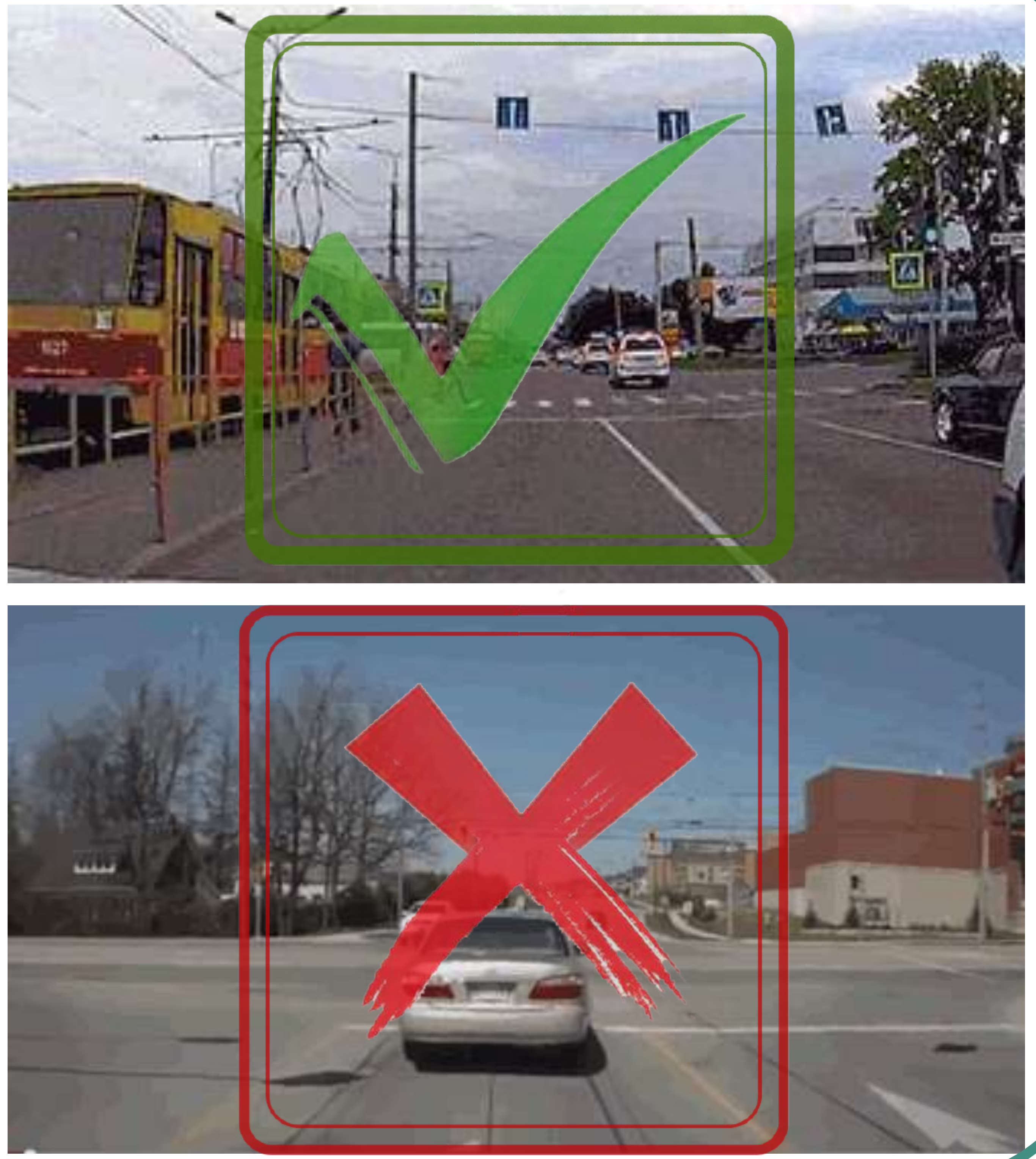
…

Reward engineering

What to consider?

How to weight?

…



Hand-designed rewards

Implicit Affordances, ROACH

Ground truth measurements

…

Reward engineering

What to consider?

How to weight?

…

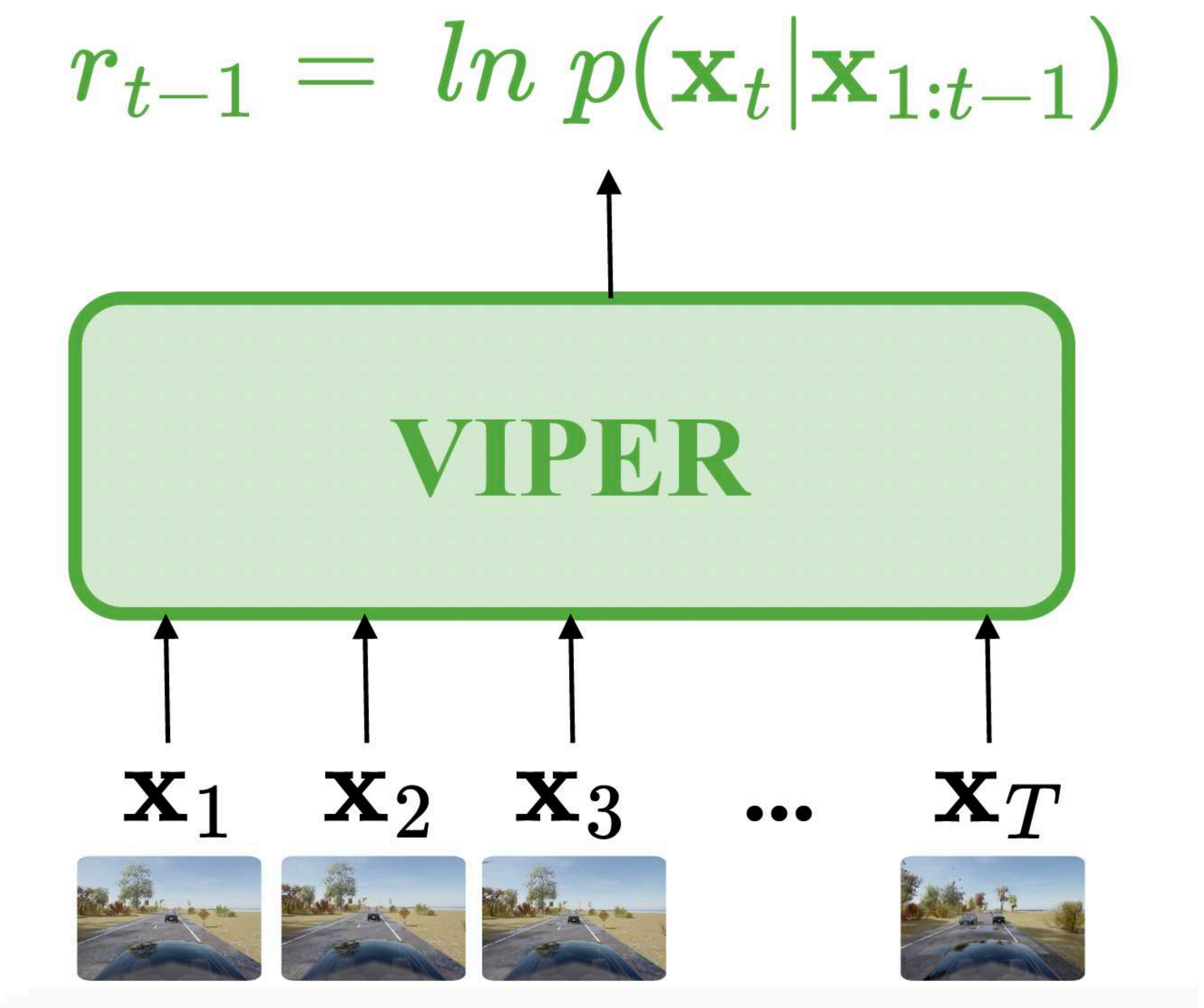
Reward learning

Potential to scale to real-world

Without reward engineering

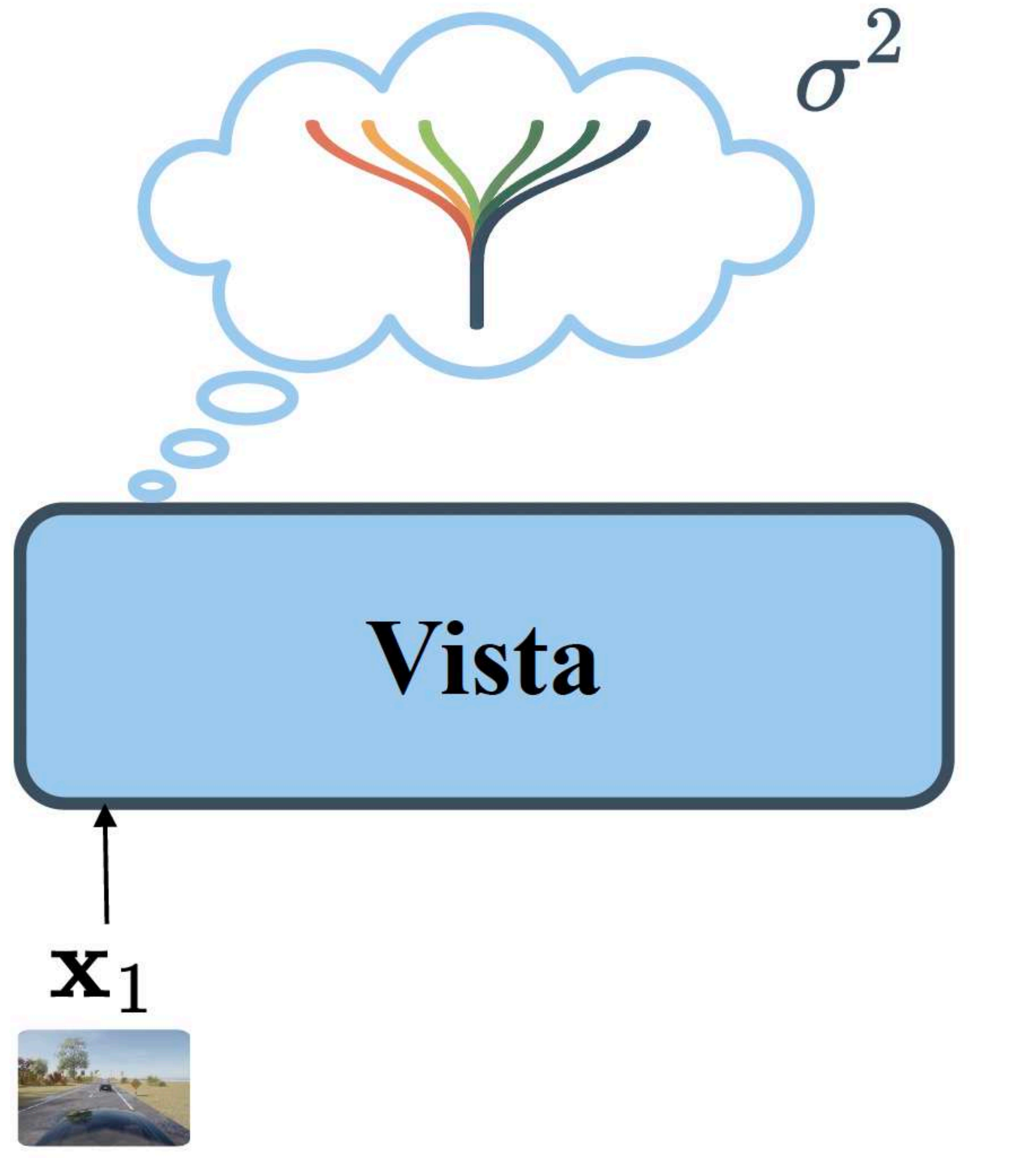
Likelihood-based rewards

VIPER; 2023



Likelihood-based rewards

Vista; 2024



Expert

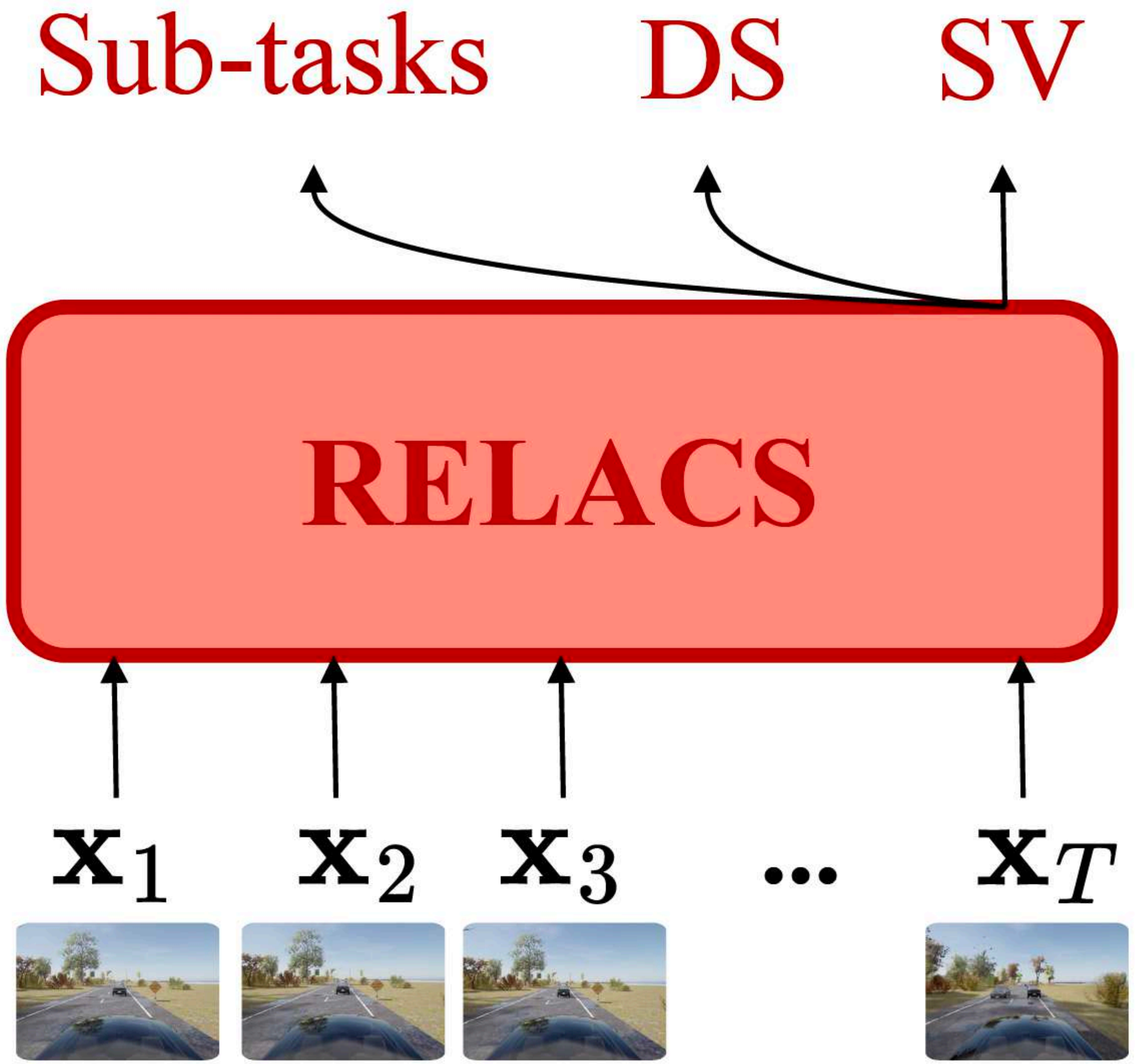
Crash

Likelihood-based: ✓

Likelihood-based: ✓

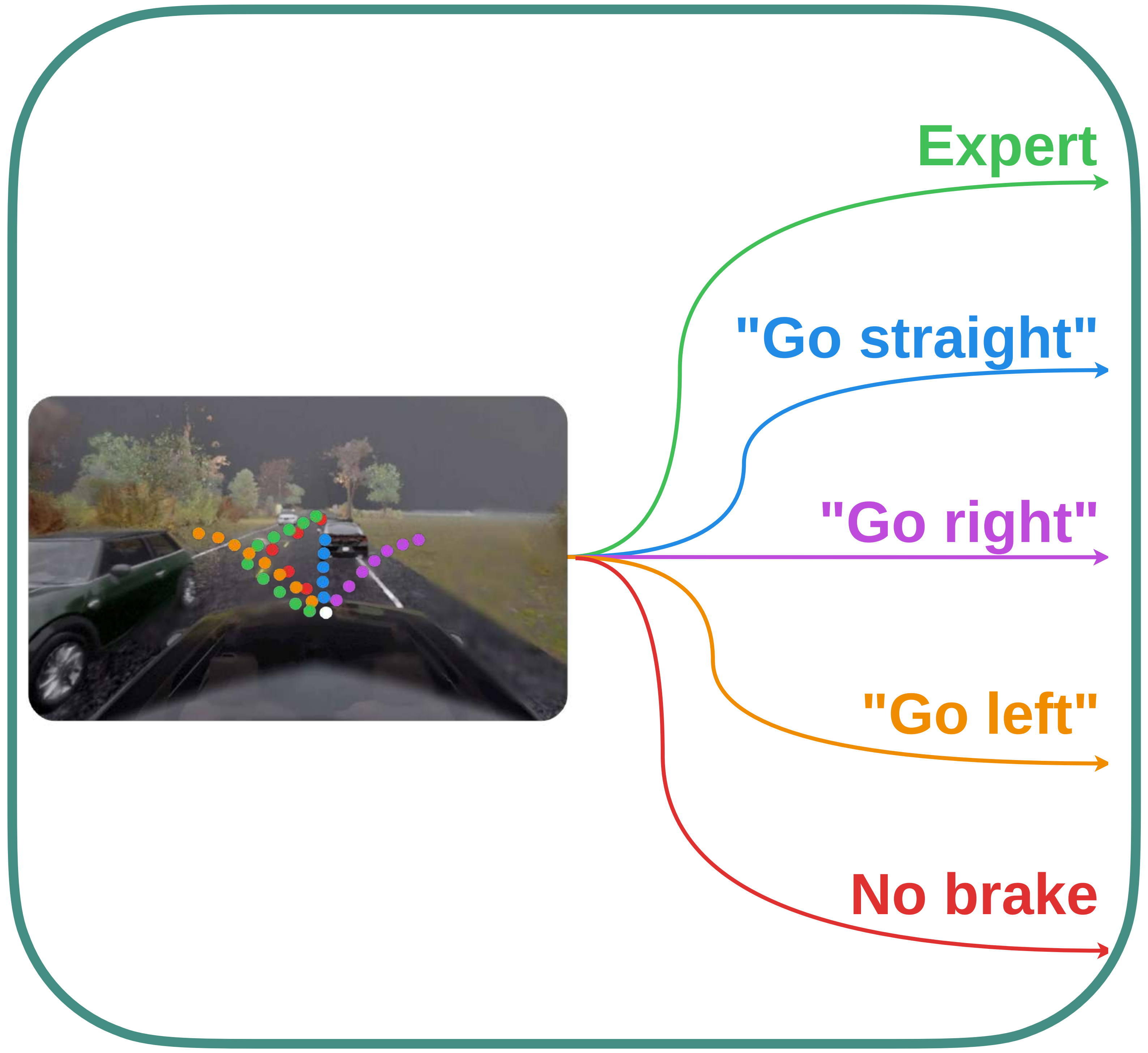

Expert
Vista: 0.87
VIPER: 0.47
Likelihood-based:
Crash
Vista: 0.84
VIPER: 0.36
Likelihood-based:
High Uncertainty
Vista: 0.77
VIPER: 0.22
Likelihood-based:

A dedicated reward model, independent of future prediction.

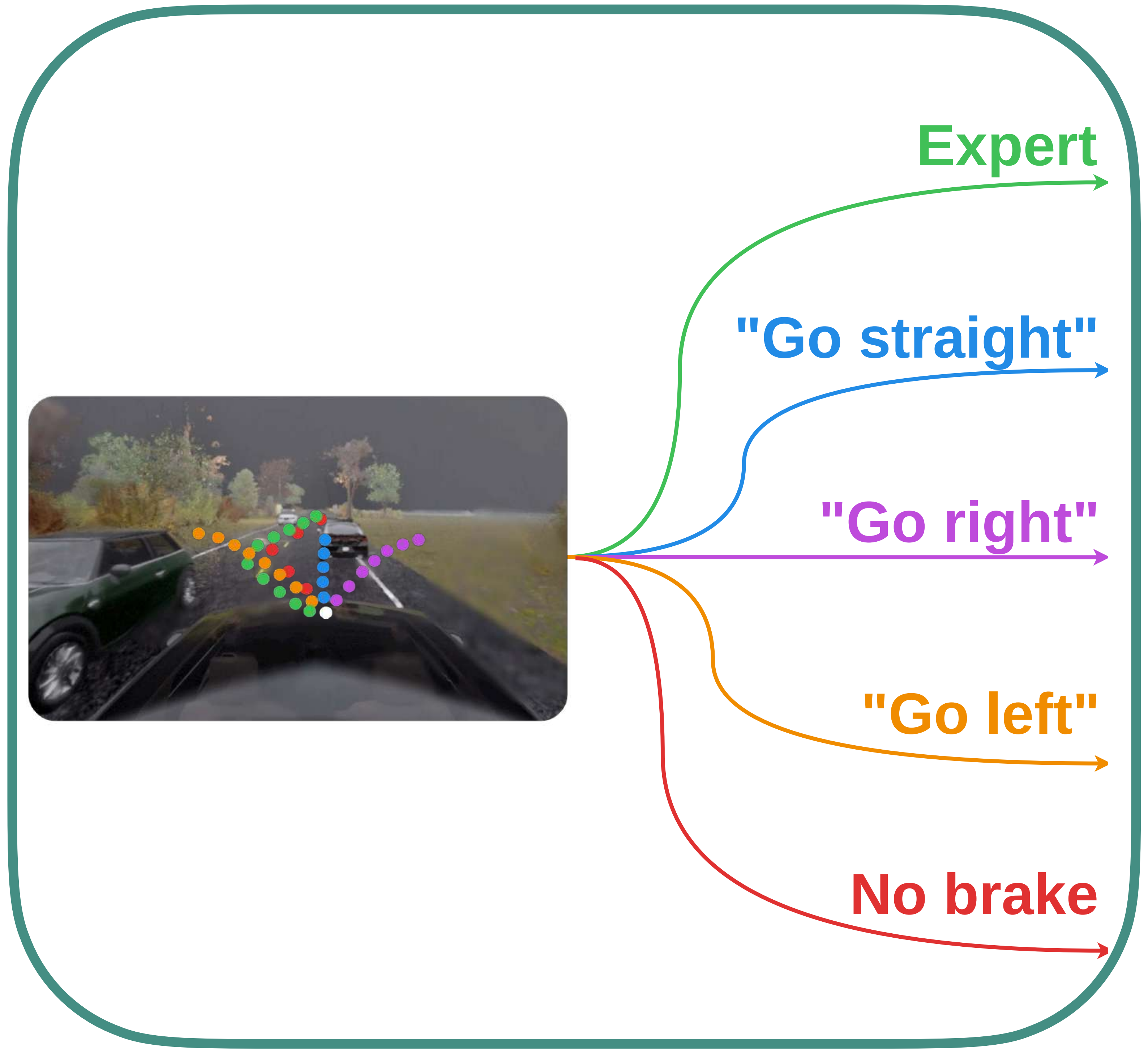



Expert
Vista: 0.87
VIPER: 0.47
RELACS: 1.00
Likelihood-based: ✓
Our method: ✓
Crash
Vista: 0.84
VIPER: 0.36
RELACS: 0.46
Likelihood-based: ✓
Our method: ✓
High Uncertainty
Vista: 0.77
VIPER: 0.22
RELACS: 1.00
Likelihood-based: ✗
Our method: ✓

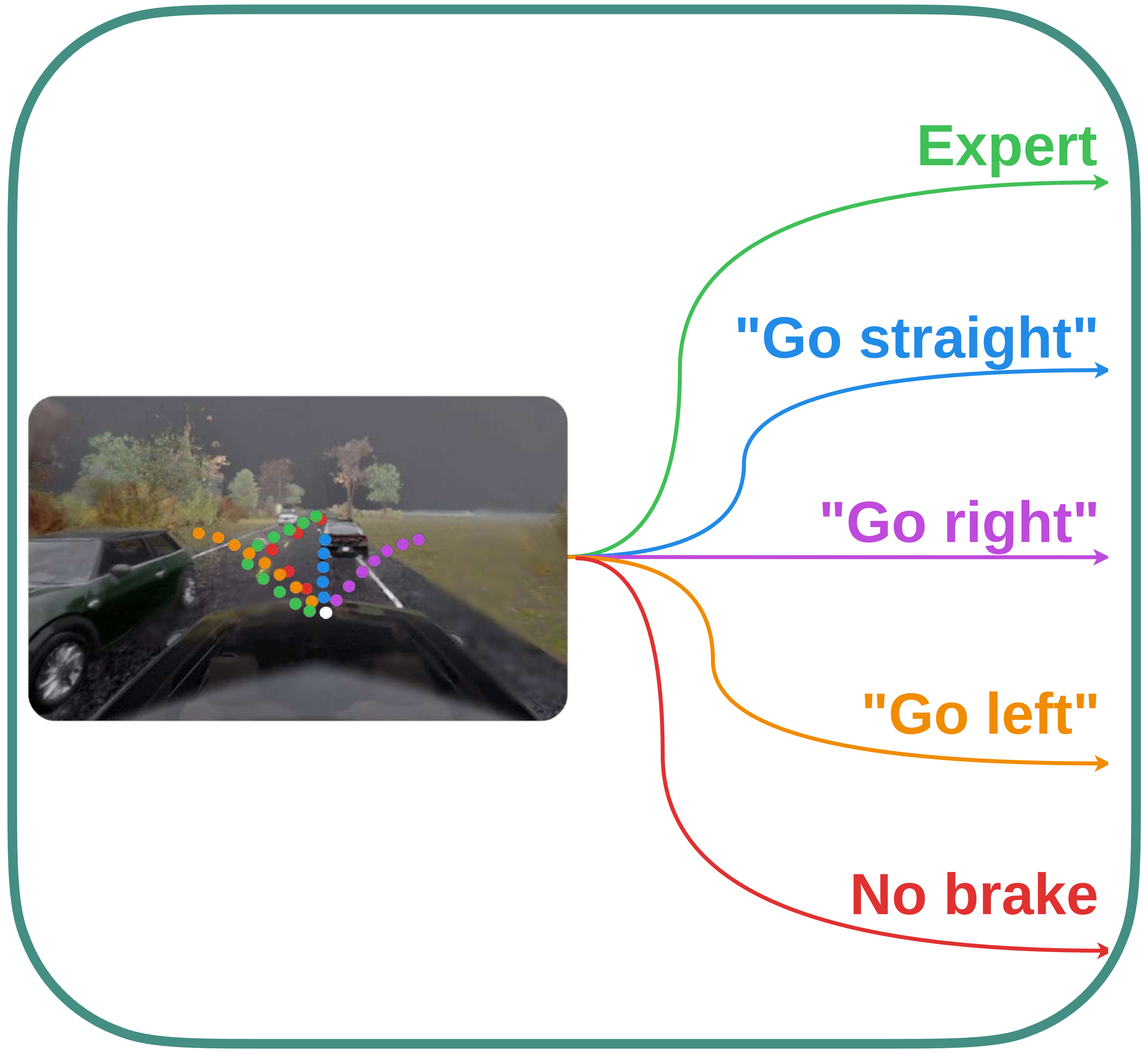
How to learn what should be low-reward?

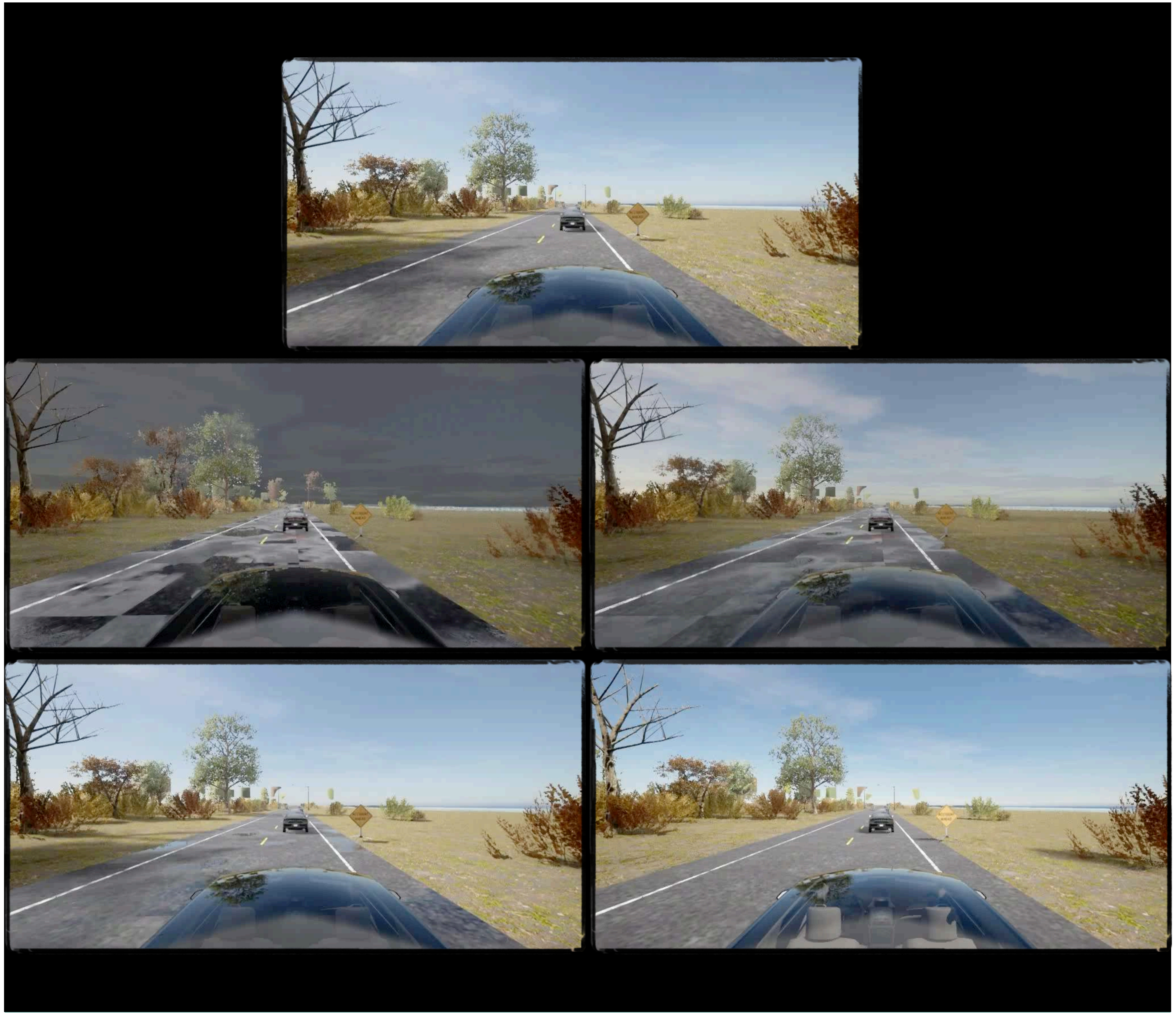


How to learn what should be low-reward?

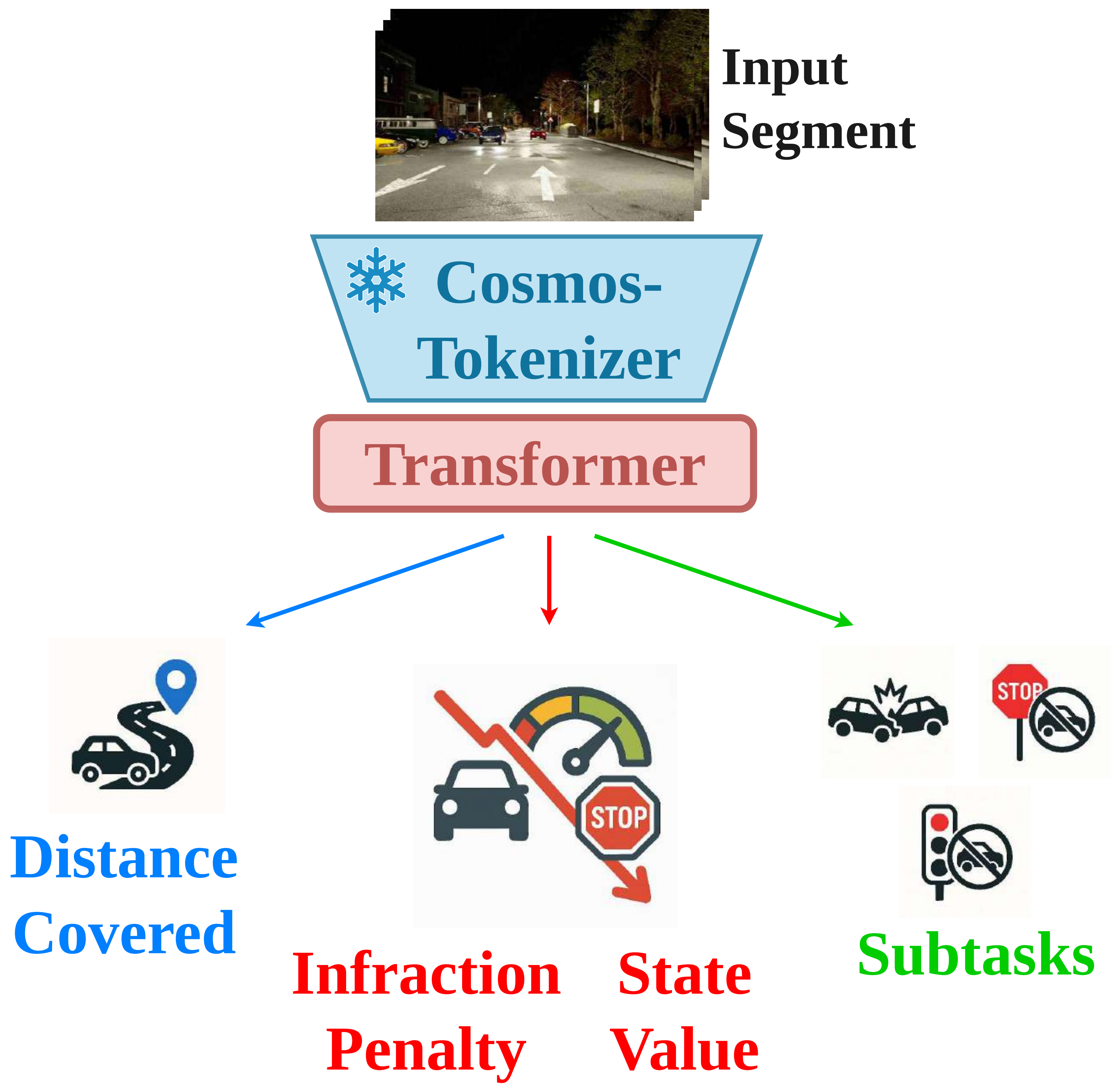


How to learn what should be low-reward?

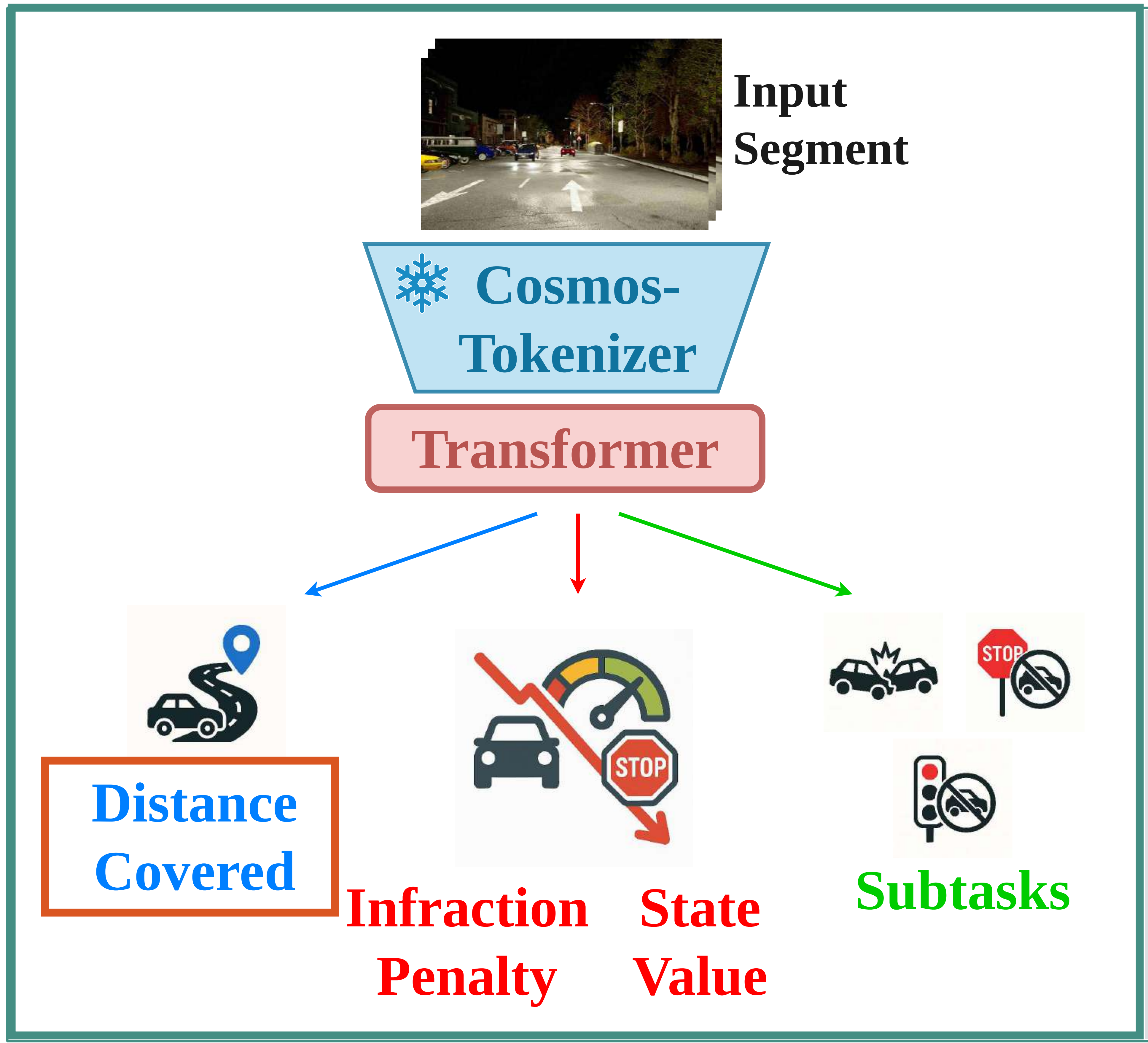




Reward learning from counterfactuals



Reward specification



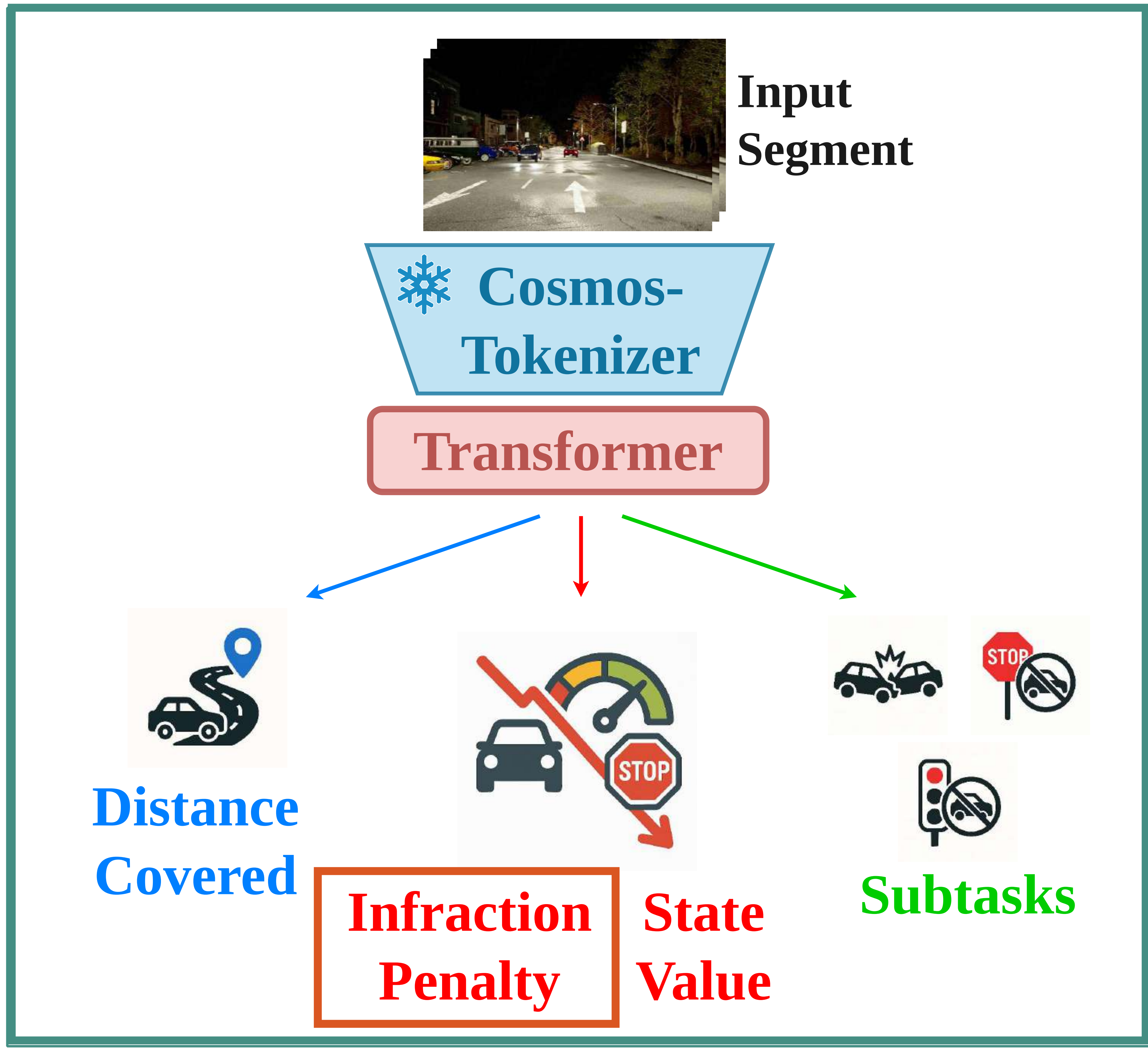
$$\text{DrivingScore} = \boxed{\text{RouteCompletion}} \times \text{InfractionPenalty}$$

RC : % of the route completed

map or goal dependency

$$\text{DistanceCovered} = \text{RC} \times \text{RouteLen}$$

Reward specification

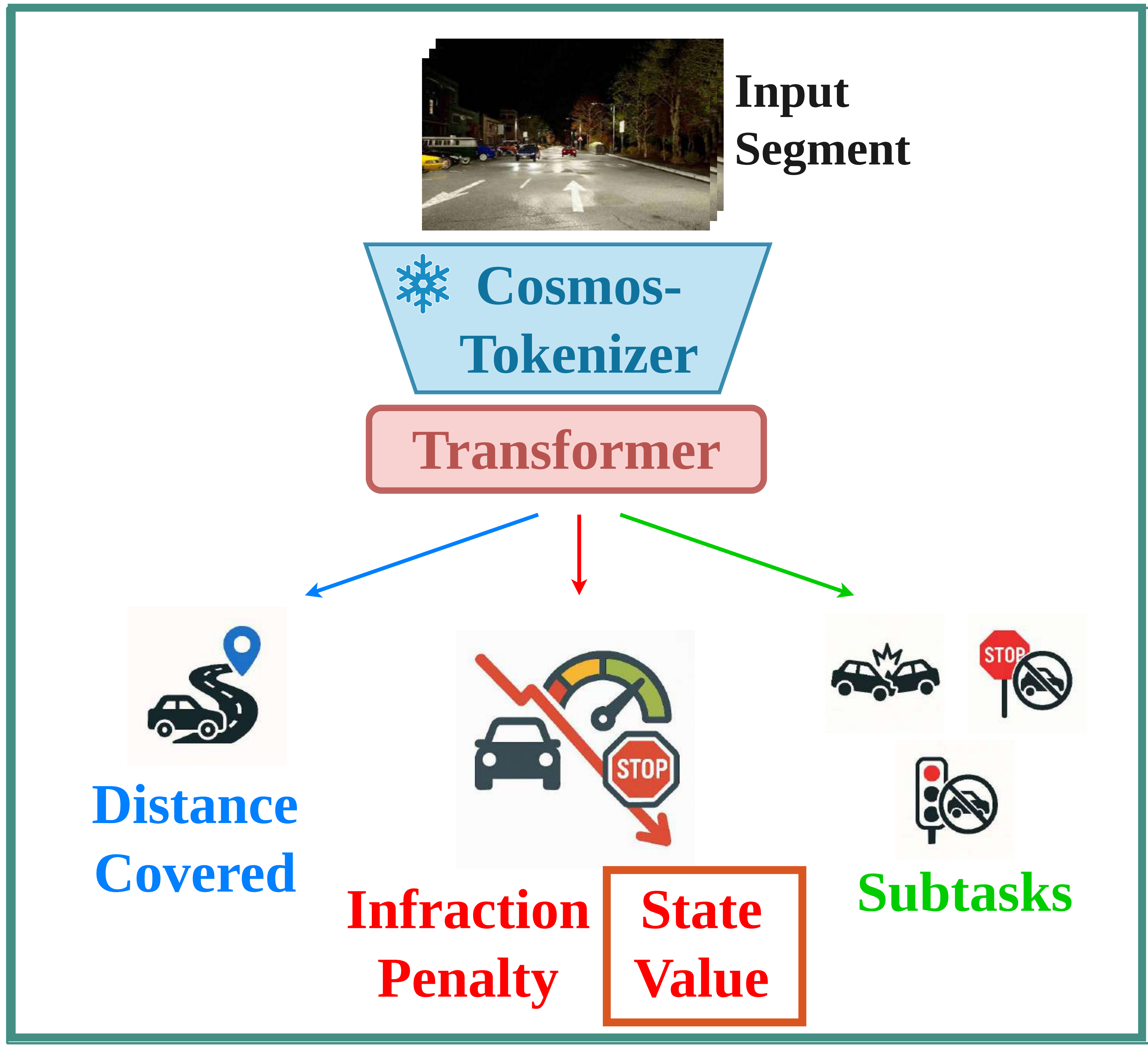


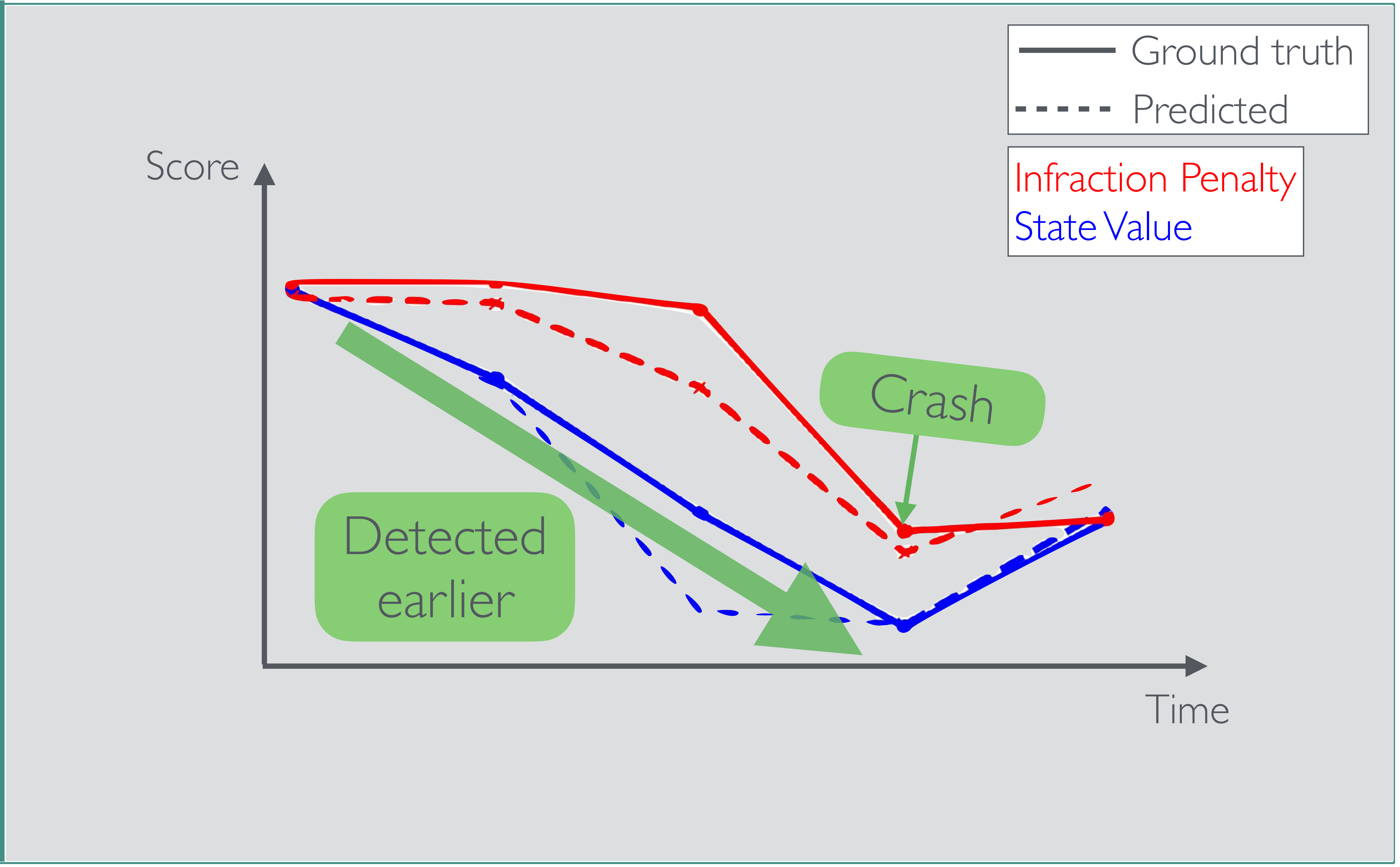
$$\text{DrivingScore} = \text{RouteCompletion} \times \boxed{\text{InfractionPenalty}}$$

$$\text{IP} = \prod_j c_j^{\#I_j}$$

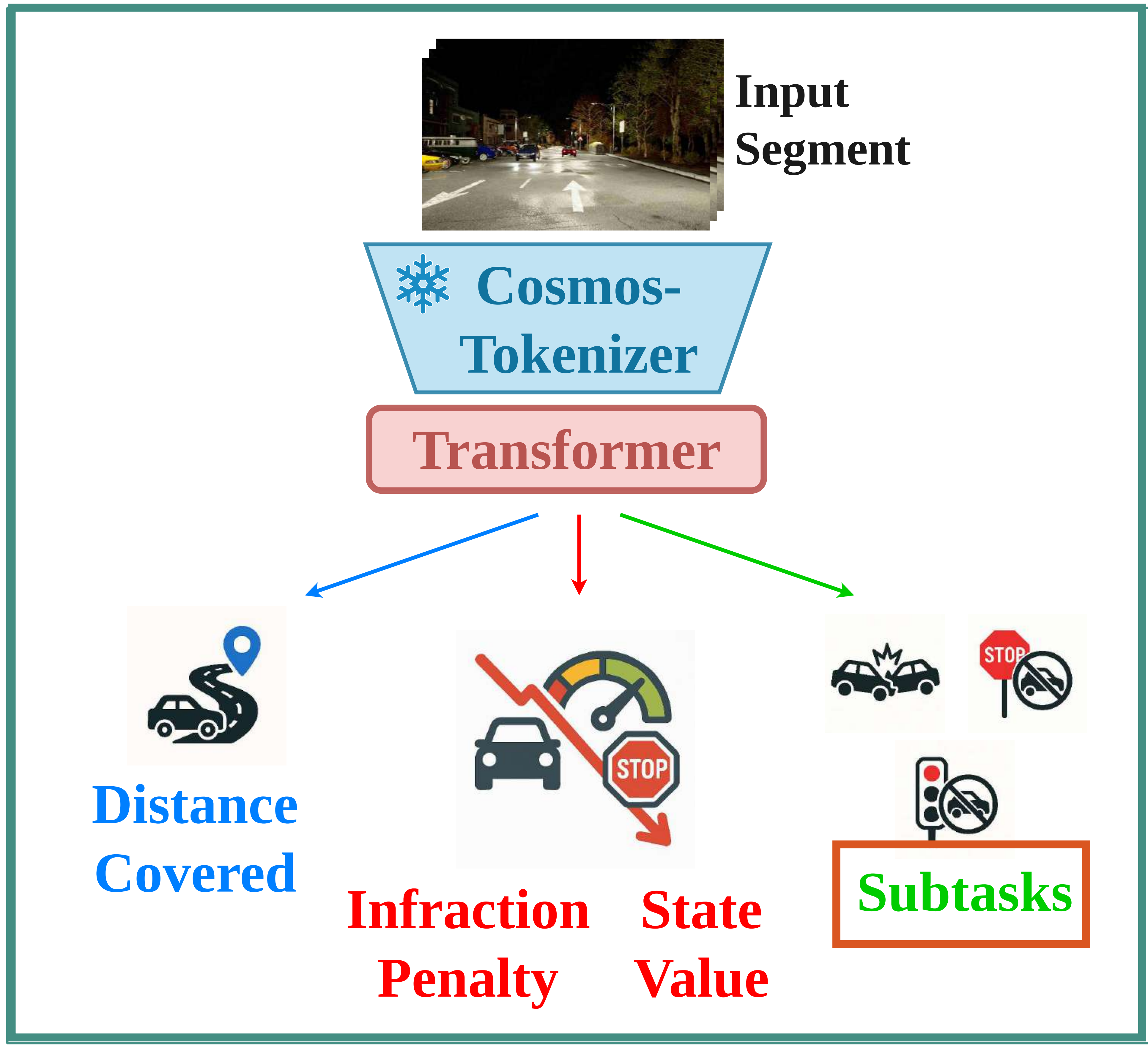
Infraction (I)	Coeff. (c)
Collision w/ pedestrians	0.50
Collision w/ vehicles	0.60
Collision w/ static	0.65
Running a red light	0.70
Running a stop sign	0.80

Reward specification





Reward specification

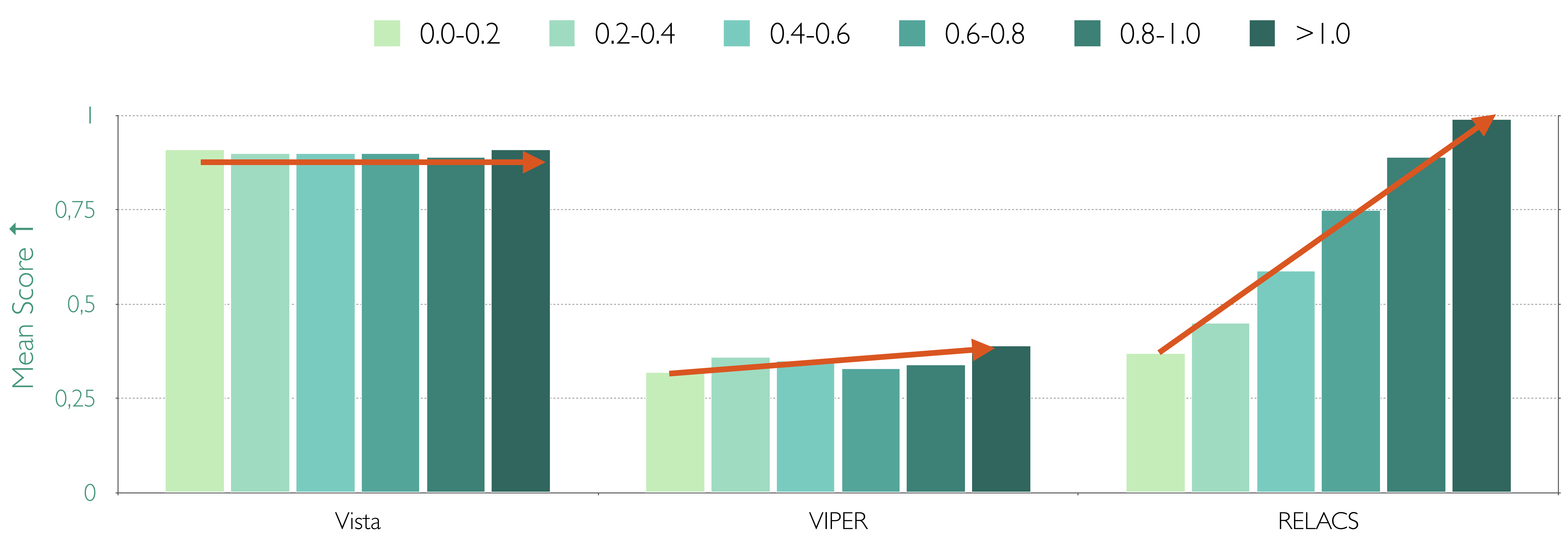


Infraction Type
Collision w/ pedestrians
Collision w/ vehicles
Collision w/ static
Running a red light
Running a stop sign

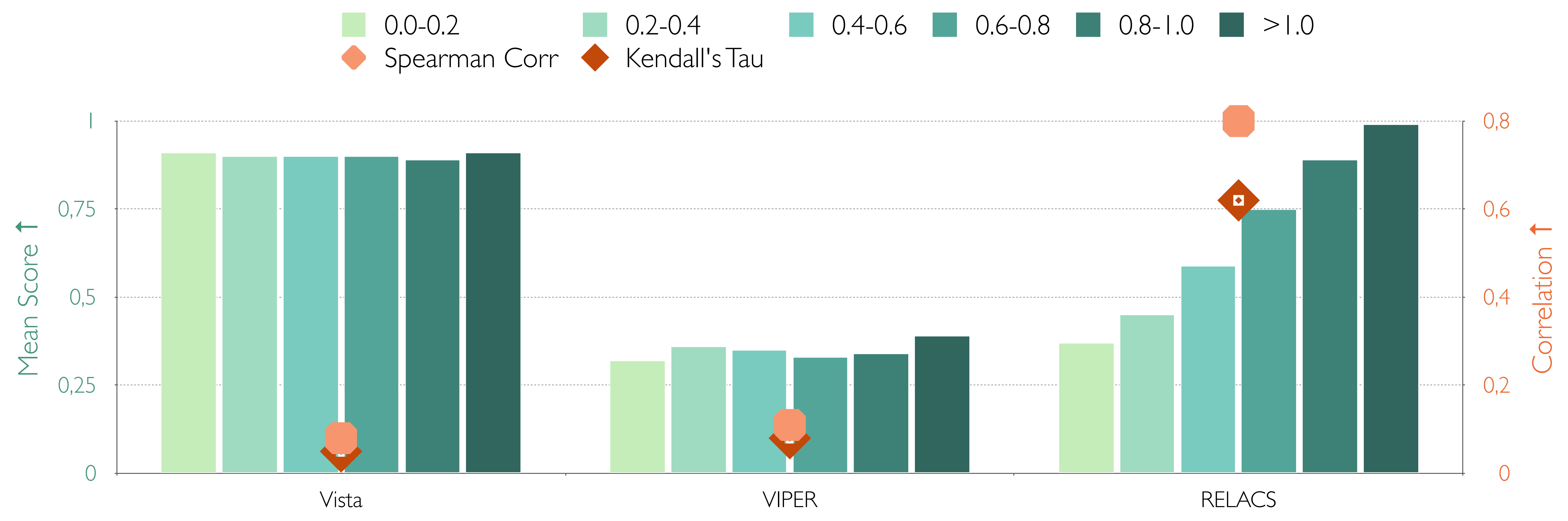
Validating on CARLA

0.0-0.2 | 0.2-0.4 | 0.4-0.6 | 0.6-0.8 | 0.8-1.0 | >1.0

Validating on CARLA



Validating on CARLA

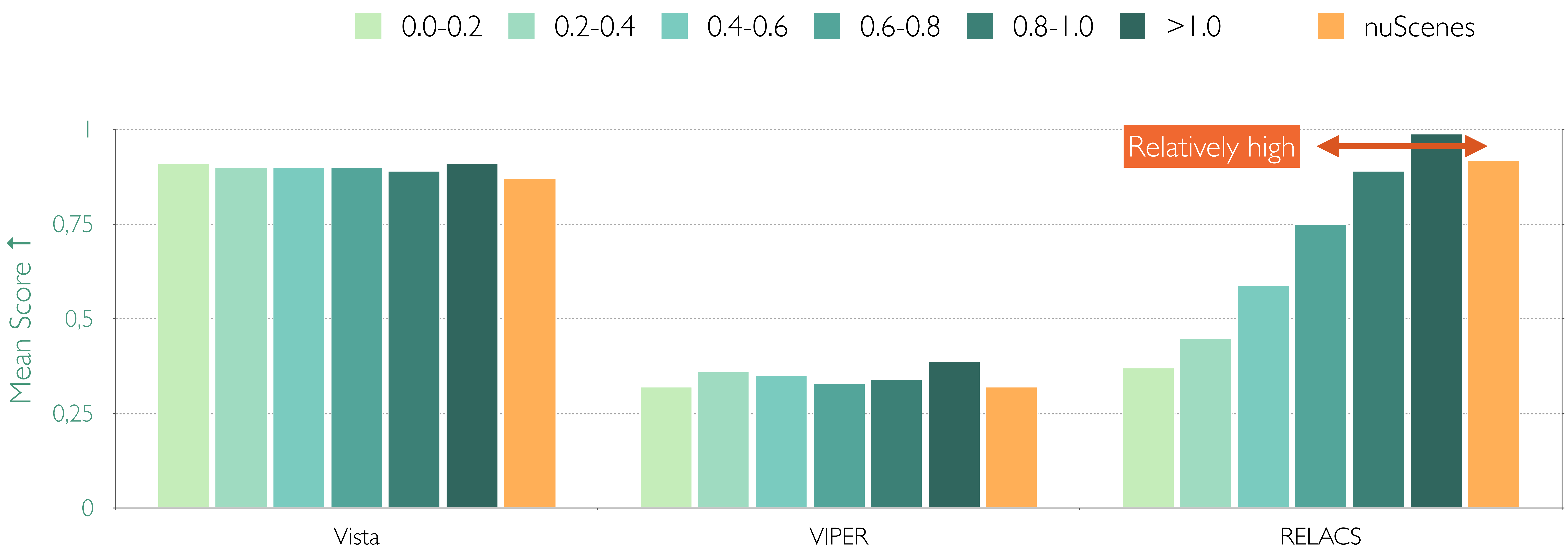


Validating on CARLA

0.0-0.2 | 0.2-0.4 | 0.4-0.6 | 0.6-0.8 | 0.8-1.0 | >1.0

Does it generalize to nuScenes?

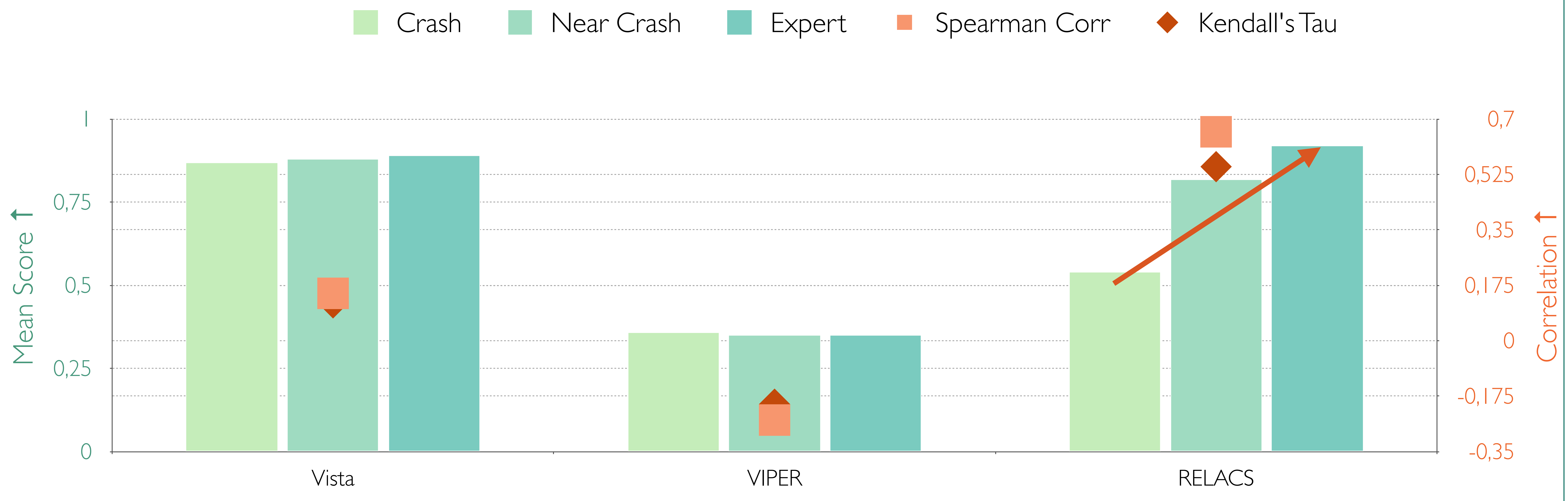
NUSCENES



YouTube CarCrash

Crash Near Crash Expert

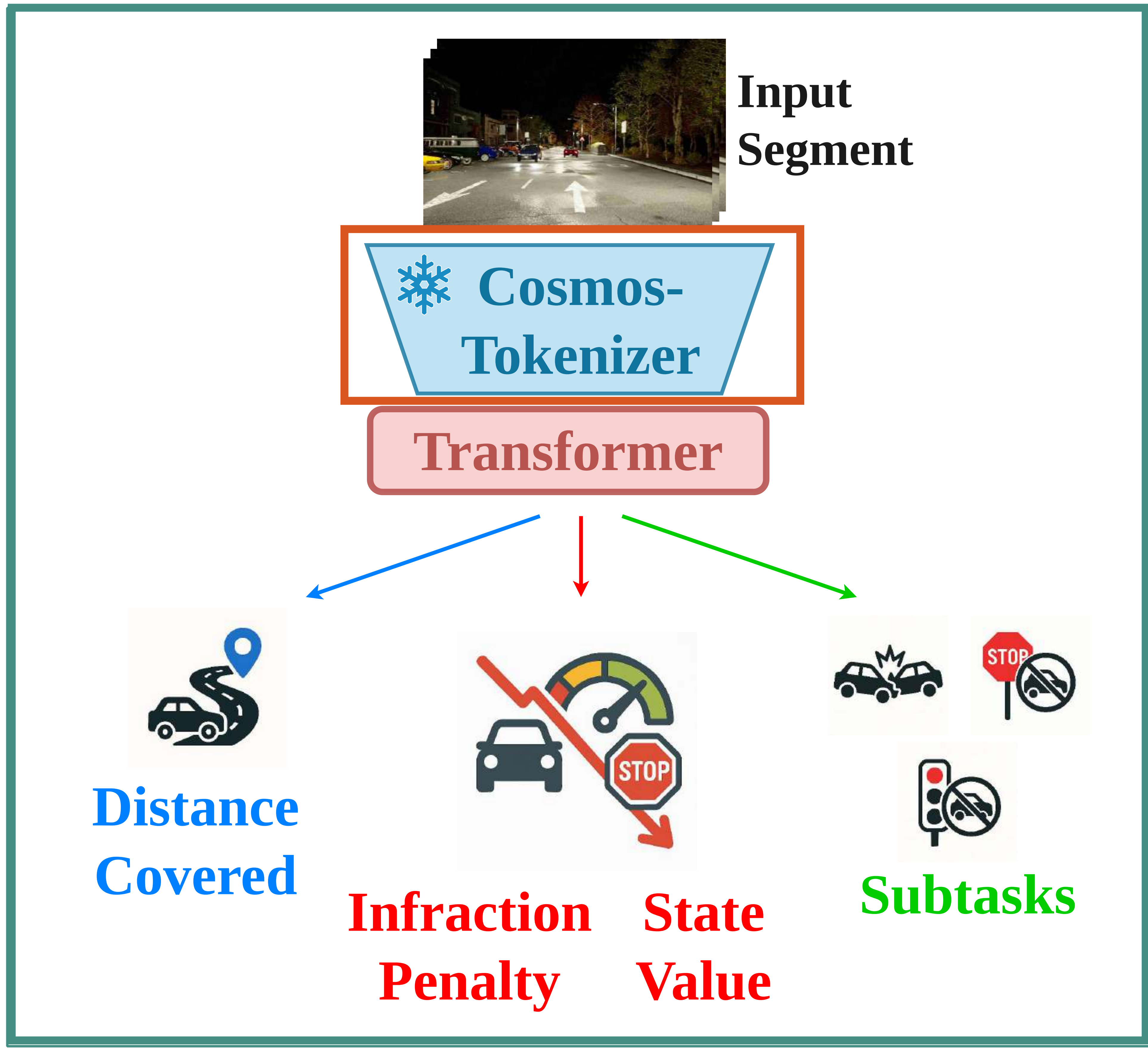
YouTube CarCrash

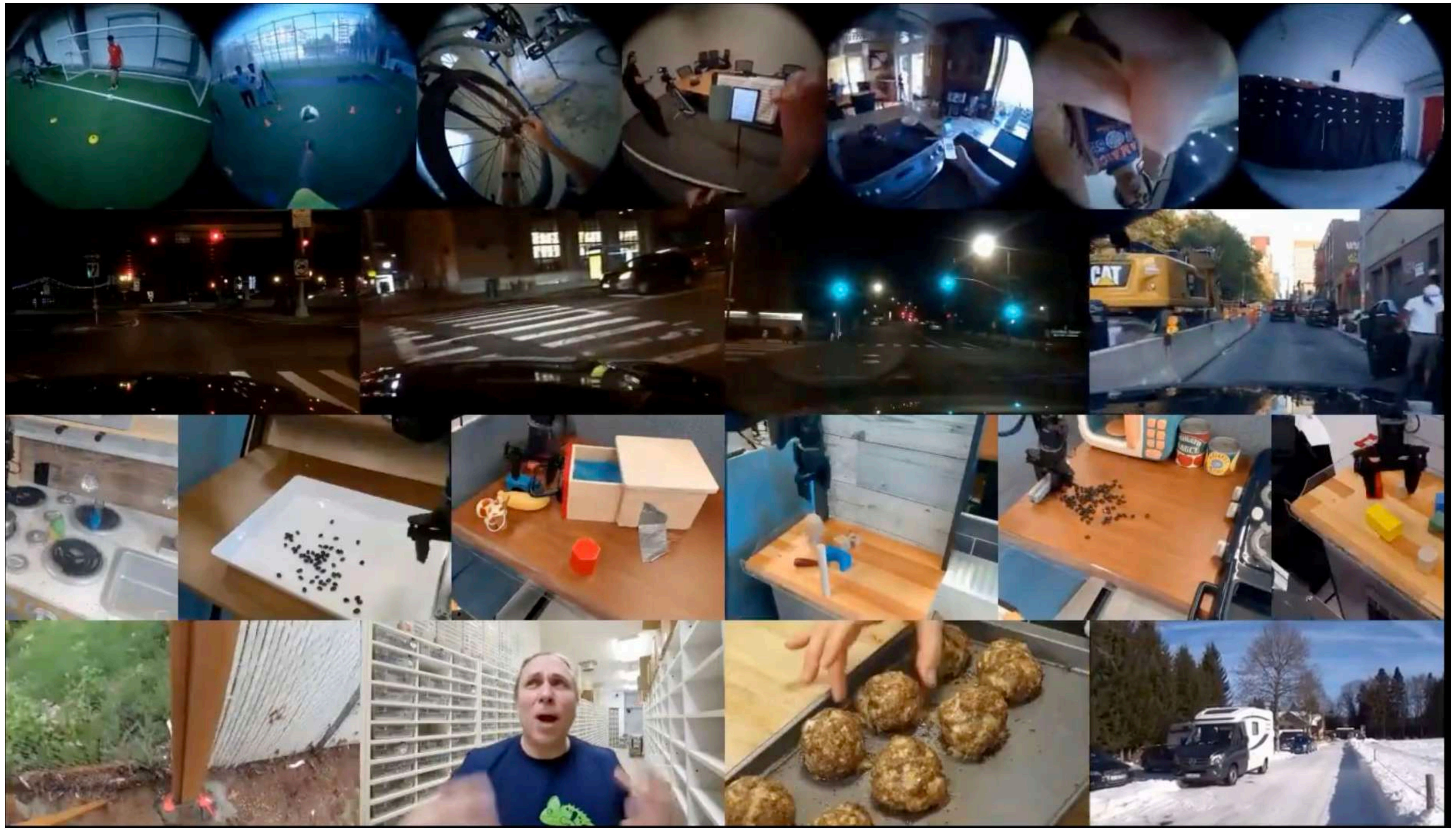


YouTube CarCrash

Crash Near Crash Expert

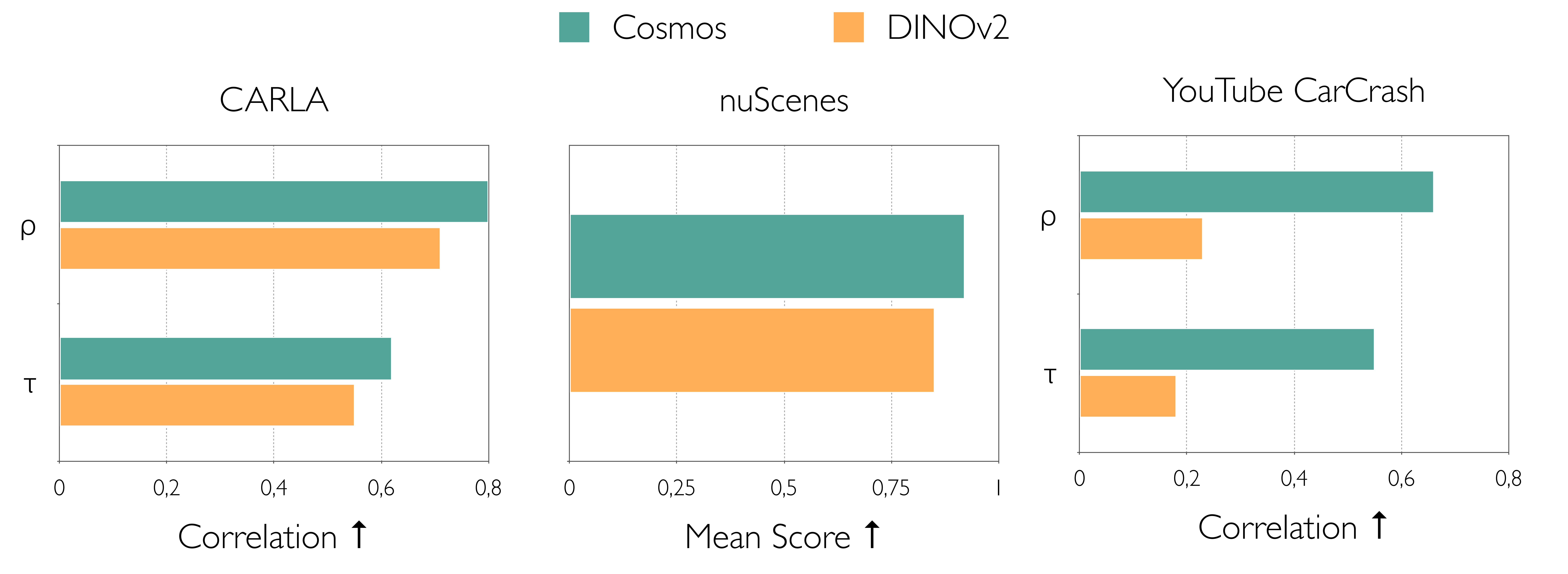
Why does it generalize?





NVIDIA Cosmos; 2025

Why does it generalize?



NAVSIM: Ego progress

Real

NAVSIM: Ego progress

Rendered slower

Real

Rendered faster

NAVSIM: Ego progress

Distance Covered

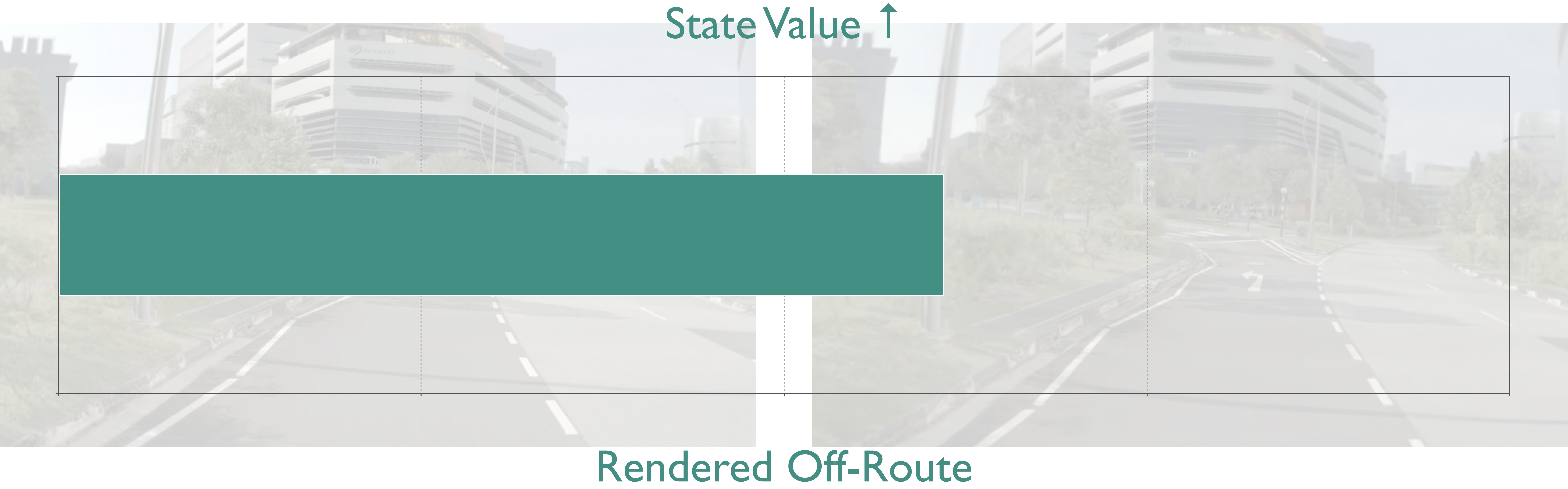
Faster

NAVSIM: Route deviation

Real

Rendered Off-Route

NAVSIM: Route deviation



Remaining problems and future work

- Imbalance in crash types
  - Data augmentation for synthetic infraction generation
  - Active data collection methods
- Infractions are temporally sparse.
  - Infraction localization
- Real-time closed-loop planners

Counterfactual

Thanks!