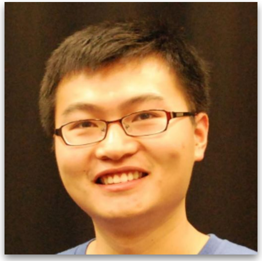
Autonomous Driving: From Basics to Behavior Challenges

Aleksandr Petiushko (https://petiushko.info/)
Nuro, ML Research
7/13/2023

HSE MCV Industrial Webinars

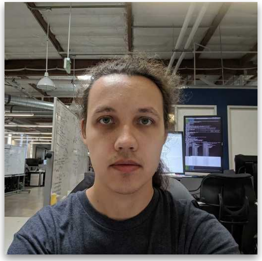
The team



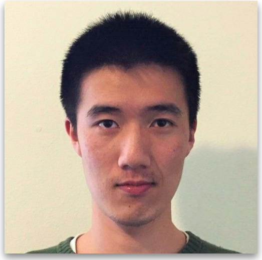
Wei
Liu

Jonathan
Booher

Ashwin
Balakrishna



Vladislav
Isenbaev



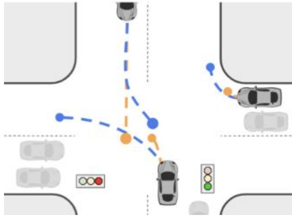
Zhenli
Zhang

Content

Introduction

01

Alex's Intro



- **Motto**: *Standing on the shoulders of giants*
- **Approach**: to combine Academia and Industry Research
  - Academia: Ph.D., lecturer on theory of ML/DL
  - Industry: TLM, ML Research (Behavior)

LOMONOSOV MOSCOW STATE UNIVERSITY

PhD THESIS

MIPT

LOMONOSOV MOSCOW STATE UNIVERSITY

SOFIA UNIVERSITY

LSI

HUAWEI

AIRI

nuro

time

AD and SDV

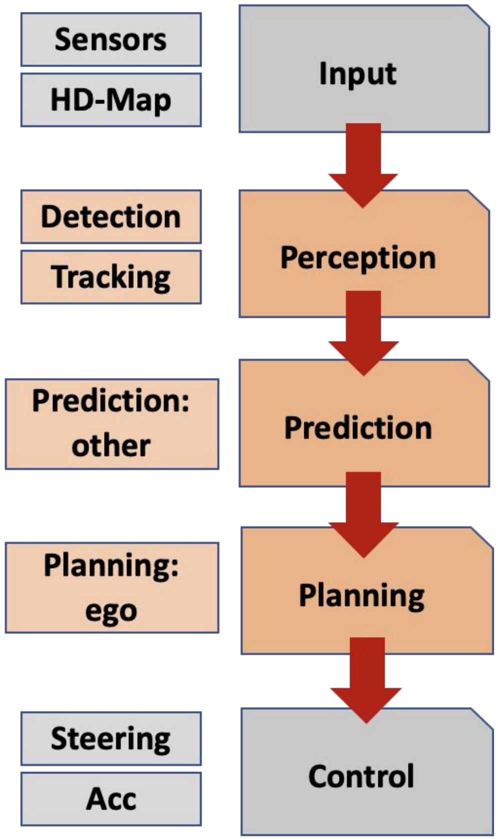
- **AD** = Autonomous Driving: the *task*
- **SDV** = Self-Driving Vehicle: the *car*
- *AD* is one of the most complex and difficult tasks, both theoretically and practically

Image source

Safety of SDV and other agents on the road is crucial

AD: ML Stack of Technologies

- The main **software** parts are the so-called **P**^{3}:
  - Perception, Prediction and Planning
- **Hardware** parts:
  - Input: Sensors
  - Output: Control (steering, acceleration)
- High-Definition Map as the helper
  - **HD-Map** contains info about the road



SDV: Sensors

- Various **sensors** are used:
  - LIDAR
  - Radar
  - Ultra Sound
  - Cameras (*x N*)

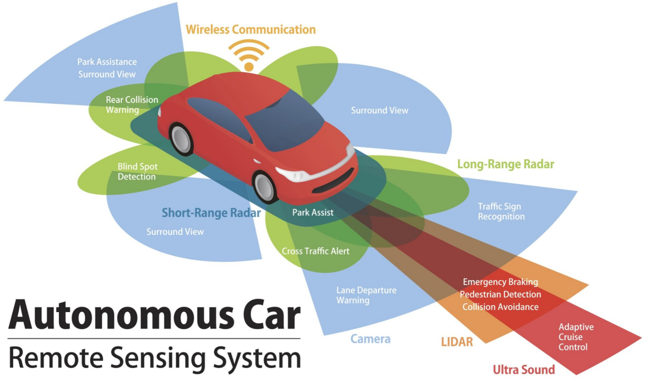
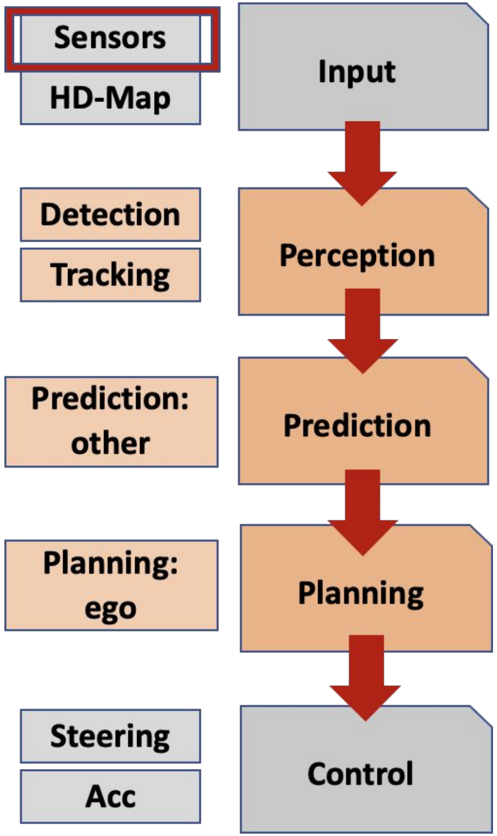


Image [source]

- **Problems**:
  - Expensive
  - Hard to synchronize



AD: HD-Map

- Helpful for prediction and planning
  - Contains information about a **road**:
    - Lanes, crosswalks, traffic lights, etc.
- **Problems**:
  - Every company has its own format
  - Significant overhead

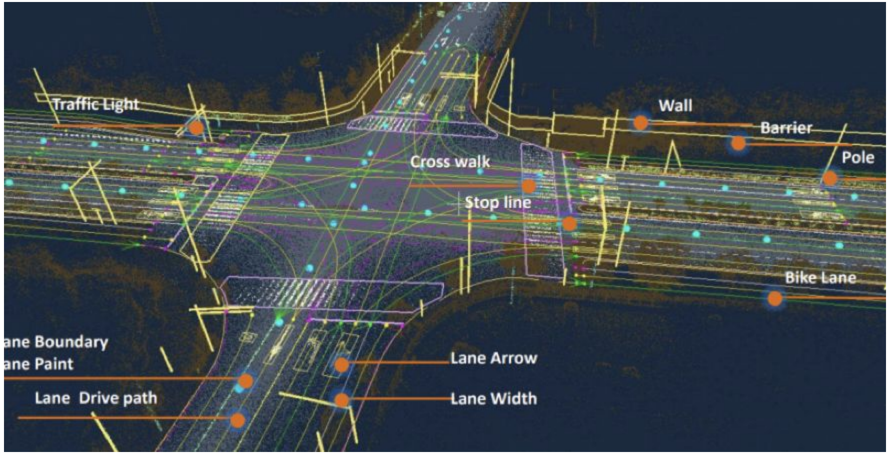
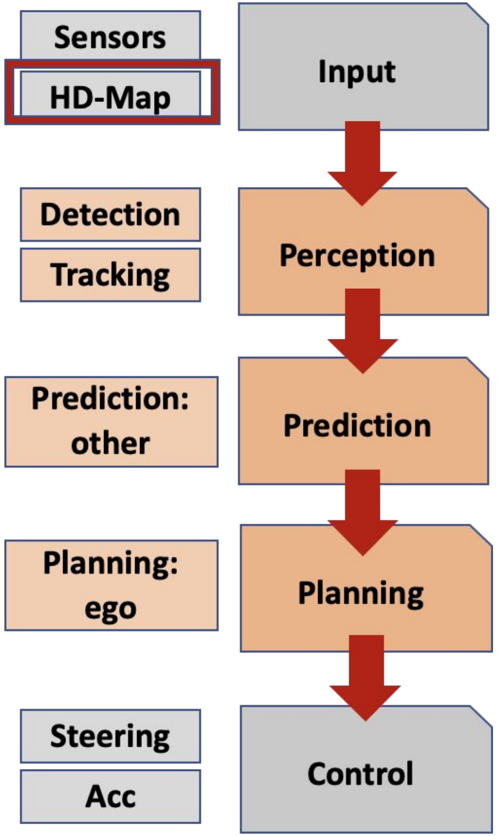


Image [source]



AD: Detection

- The *first* step of the Perception part:
  - **Detection** (segmentation, depth-estimation, etc.) of the objects around
- **Problems**:
  - Long tail (small and unusual objects) and anomalies

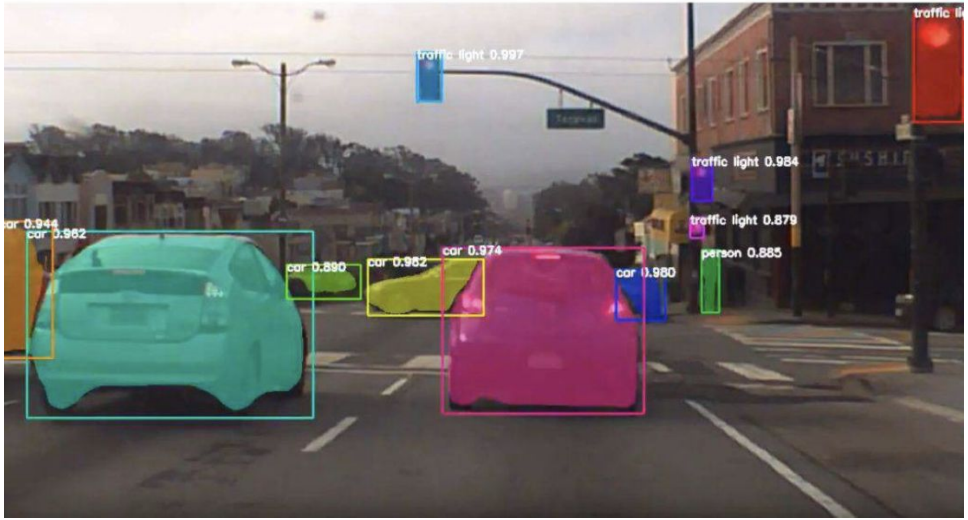
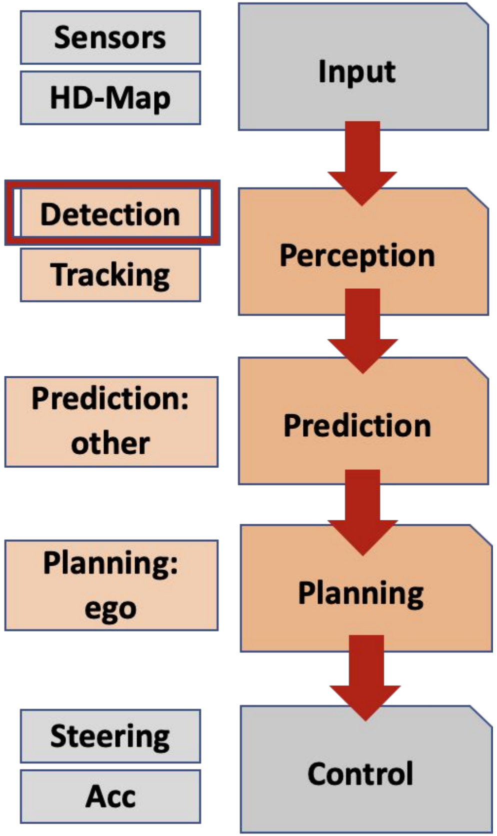


Image source



AD: Tracking

- The *second* step of the Perception part:
  - **Tracking** of the detected objects and estimation of their coordinates for the Prediction part
- **Problems**:
  - Track association of flickering objects

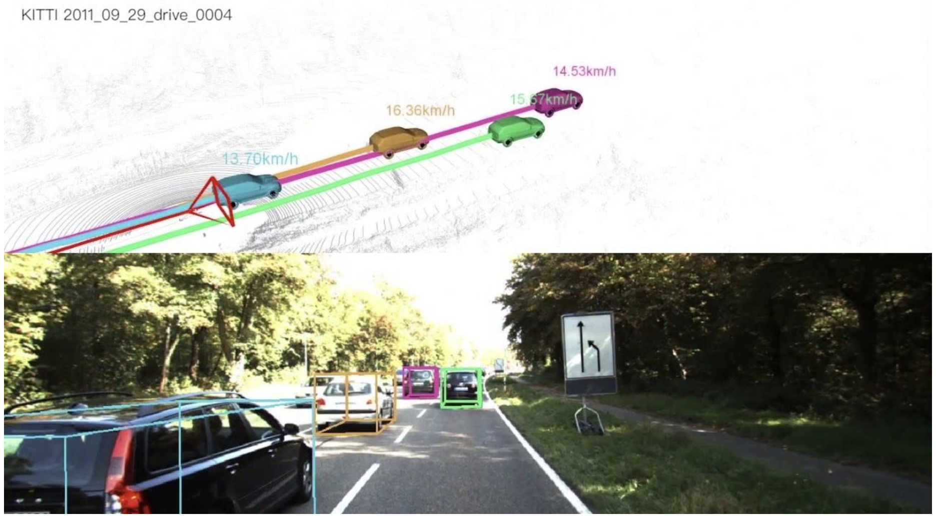
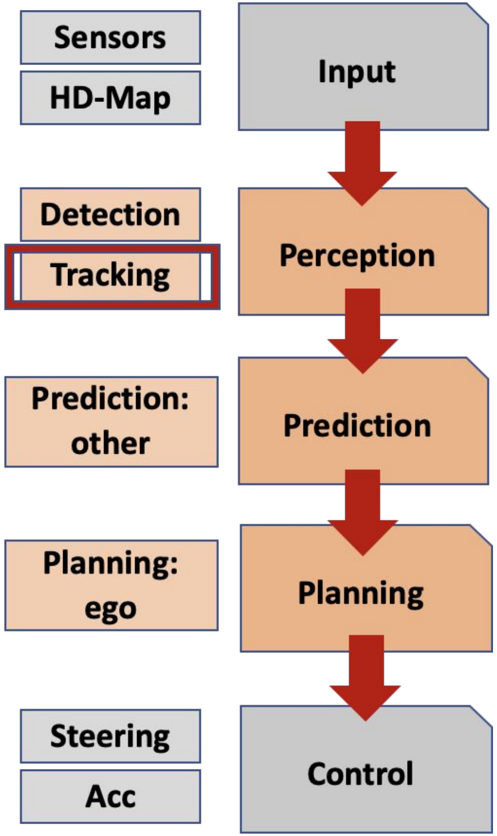


Image [source]



AD: Prediction

- Future trajectories **prediction** of all surrounding objects based on the *tracking history* and *HD-Map*
  - Usually, 1-10 second
- **Problems**:
  - Multi-modality for recall

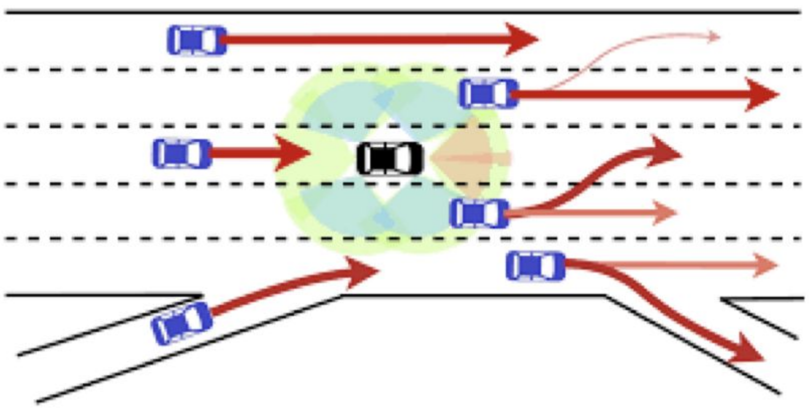
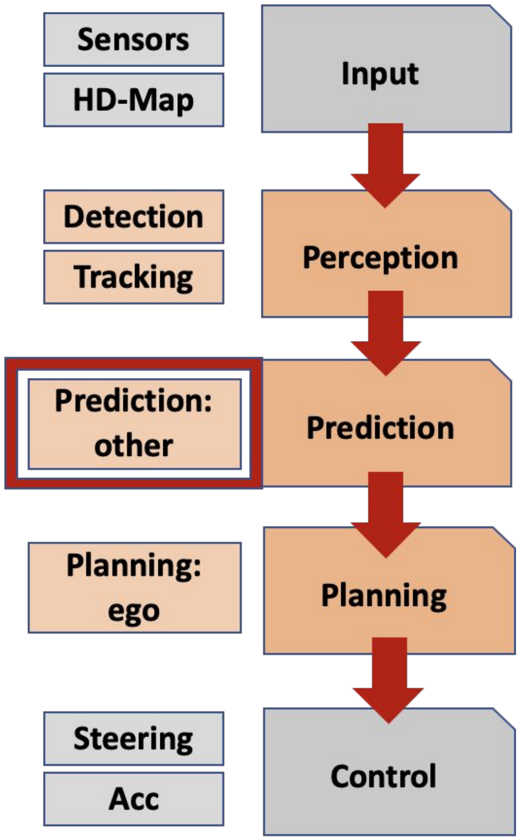


Image source



AD: Planning

- **Planning** of SDV future actions based on the *predictions* and *HD-Map*
- **Problems**:
  - Consistent joint prediction and planning

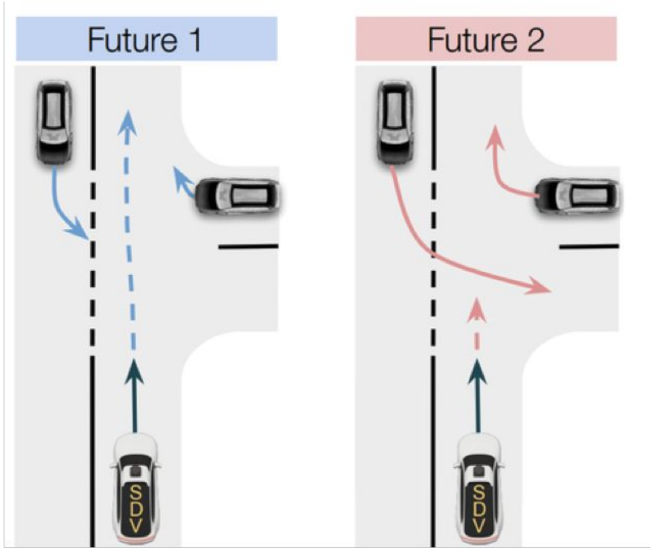
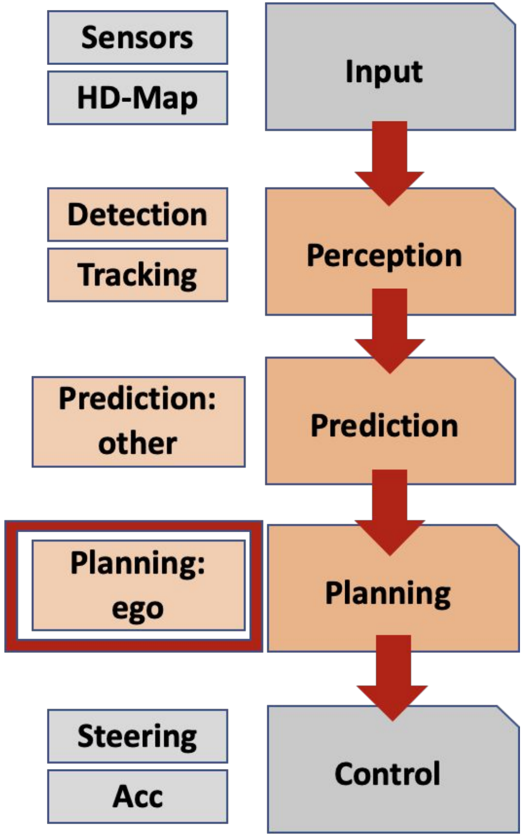


Image [source]()



SDV: Control

- Realization and **control** of SDV actions based on *motion plan*
  - Steering control, acceleration control, etc.

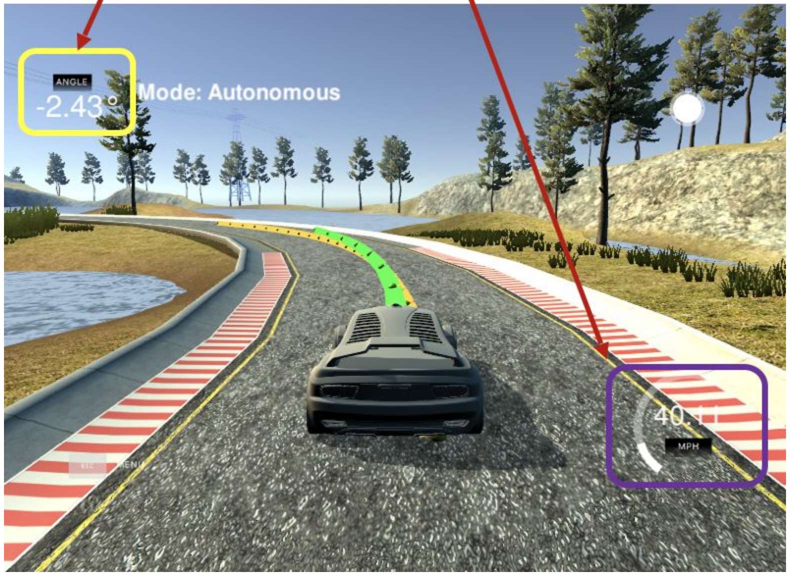
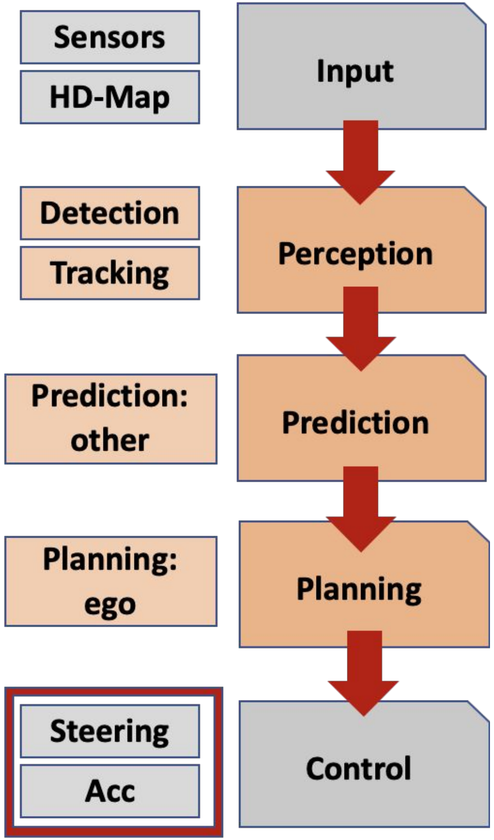


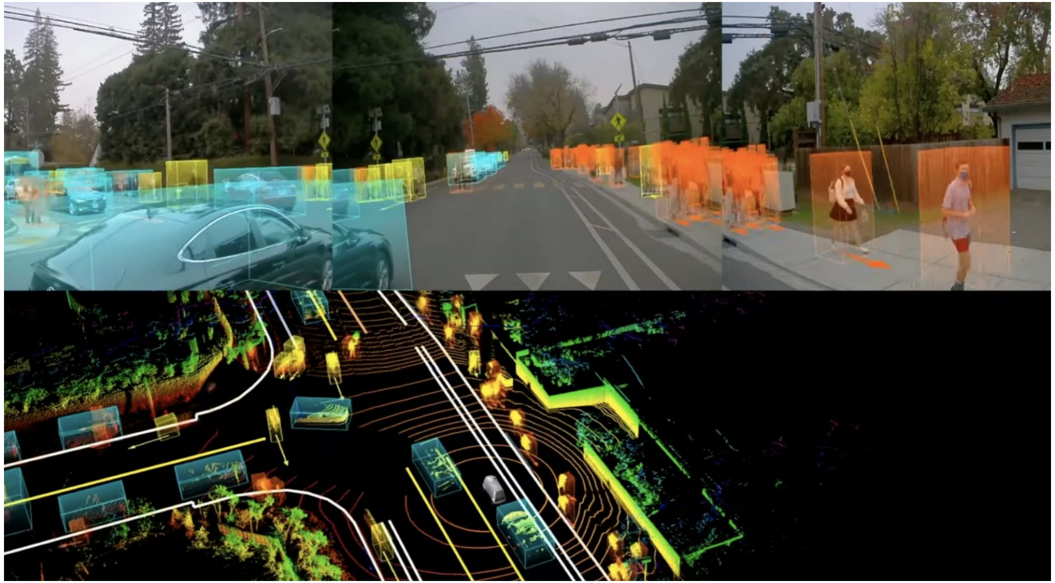
Image source

- **Problems**:
  - Dynamic and kinematic limitations

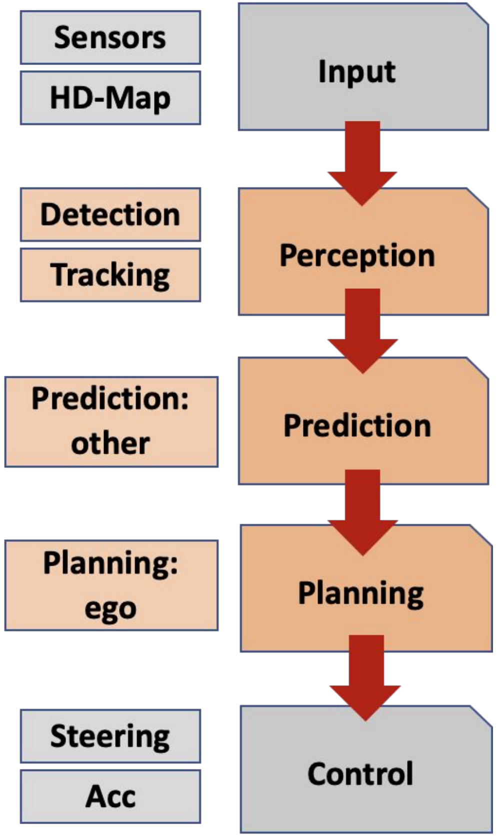


Autonomy Stack at Nuro

- An **Overview** of the **Nuro Autonomy Stack** — Albert Meixner, Nuro's Head of Software



Video source



Motivation

01

Content

Problem 1: Road Agents

Trustworthy predictions for use in both **Prediction** and **Simulation**

Problem 2: AV Motion

Flexible and safe **selection** process allowing **ego** proposals of **any source**

Our Approach

Better **training**, **evaluation** and **reasoning** leading to **safer driving**!

Better Agents Prediction/Simulation

Problem

Usage of **IL-based Prediction model** for other **agents** can lead to unreasonable proposals due to distribution shift

Historical Approach

Use **only heuristic**-based agents

Better Solution

Target prediction model to **better distribution coverage/recall** (not only precision) with some ways to use it for getting good minADE

Long-Horizon Planning by Selection

Problem

Decisions have long-term, **delayed consequences**

Historical Approach

Use long-term **predictions** to approximate long-horizon planning

Better Solution

Learn a model that takes into account an **expectation over all futures**. Selection to narrow down the search space

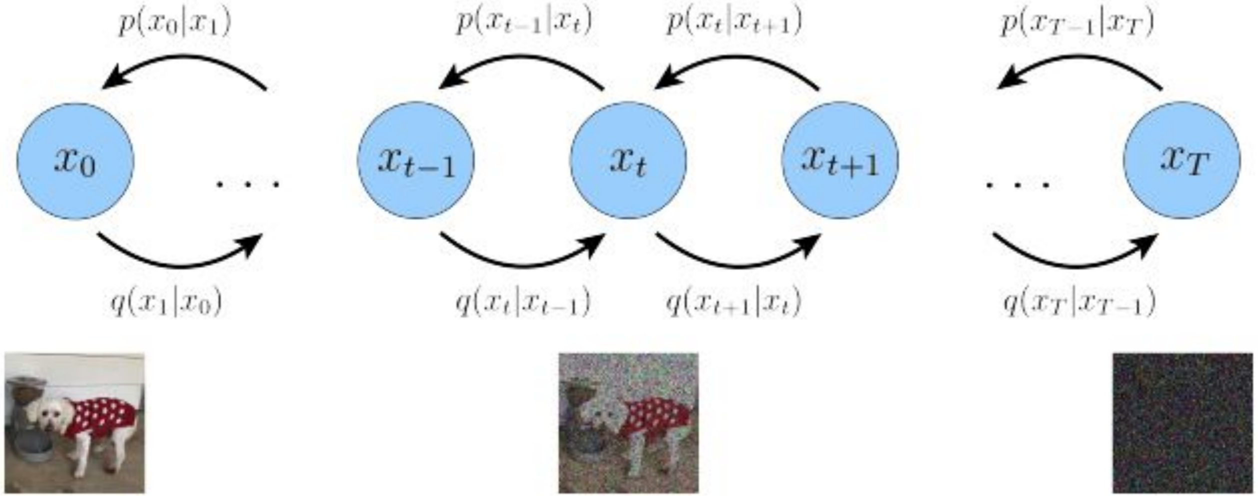
Diffusion models: Background and Trajectory Generation

02

What is a Diffusion Model?

Diffusion Model: it is a **generative** model (markovian hierarchical variational autoencoder)

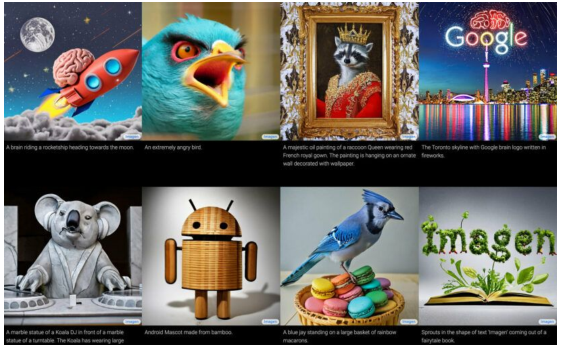
- Adding step by step some portion of noise as a **diffusion** *analogy*
- **Forward** diffusion process: adding noise by $q(x_t|x_{t-1})$. Also known as *encoding*
- **Reverse** diffusion process: de-noising by $p(x_{t-1}|x_t)$. Also known as *decoding*



x_0: input signal / signal to restore
x_T: noise
q: known
p: learnable

Image credit: https://arxiv.org/pdf/2208.11970.pdf

Success of Diffusion Models



https://imagen.research.google/

https://openai.com/dall-e-2

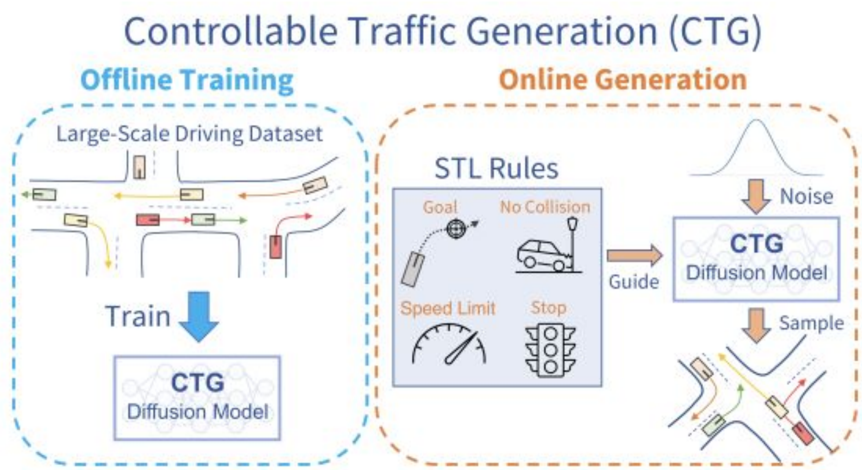
https://www.midjourney.com/

Text2Image (along with audio, video) generation: Done! (sorry, GANs 😢)

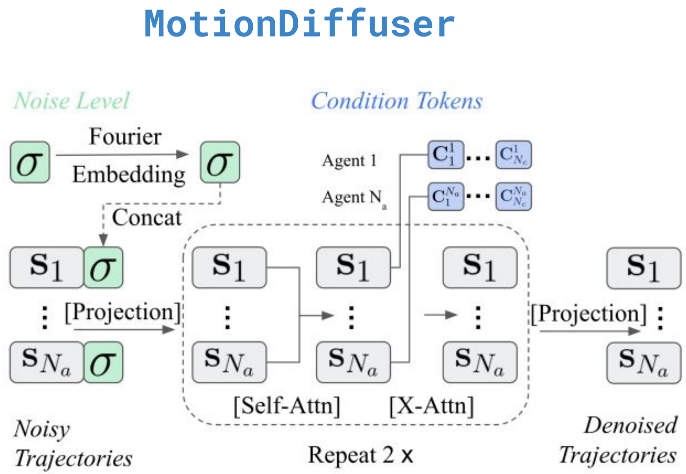
But what about **other** tasks?

Diffusion Models for Autonomous Driving

But what about **other** tasks?



Traffic simulation



Joint Prediction

We are combining both functionalities: **prediction** and **simulation**

Zhong, Ziyuan, et al. "[Guided Conditional Diffusion for Controllable Traffic Simulation]", 2022
Jiang, Chiyu, et al. "[MotionDiffuser: Controllable Multi-Agent Motion Prediction using Diffusion]", 2023

DTG: Main Goals

01

Development of Trajectory Generation module capable of a good distribution coverage

DTG = Diffusion-based Trajectory Generator

02

Improvement of closed-loop simulations

DTG: Main Goals

01

Development of Trajectory Generation module (decoder) capable of a good distribution coverage

Is theoretically ensured by using *Variational Diffusion Model* (VDM) by explicit ELBO (~NLL) optimization

DTG: Main Goals

Will provide more useful signal for RL-based trainings

02

Improvement of closed-loop simulations

DTG: Features

01

Learn diverse behaviors with distribution that matches real-world driver behaviors

02

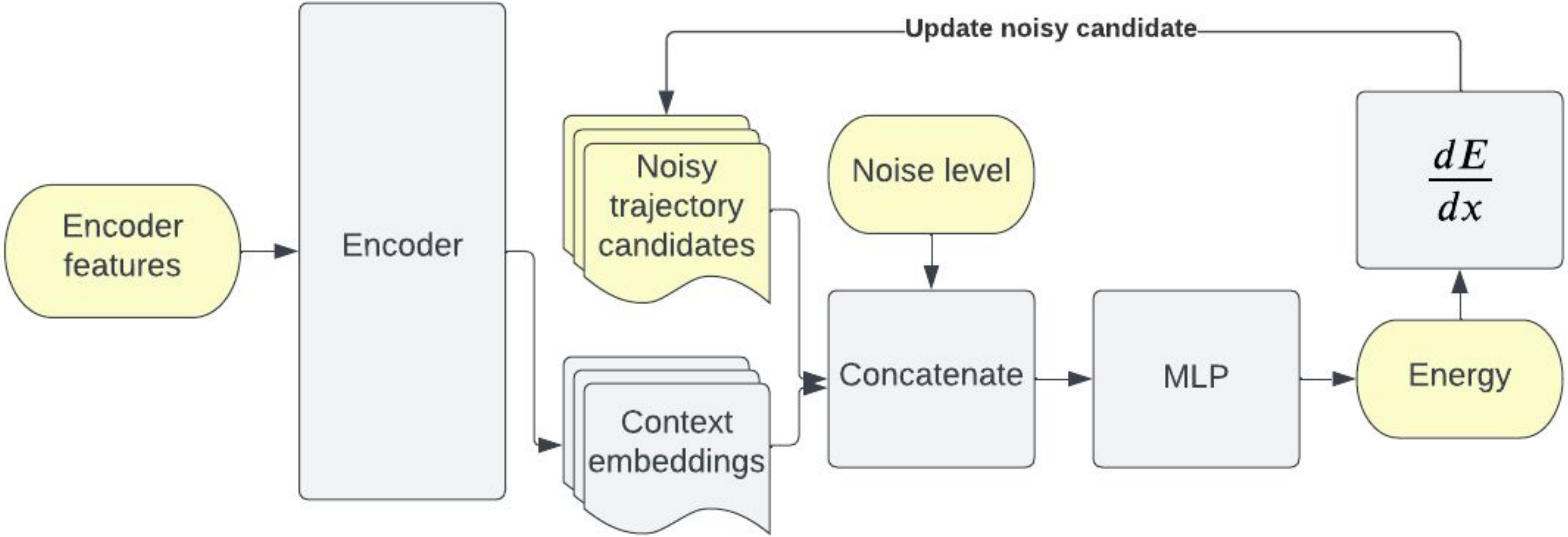
Provide good NLL, minADE, and other Prediction-aware metrics

03

Lead to stable, consistent and realistic simulation

VDM for Trajectory Generation

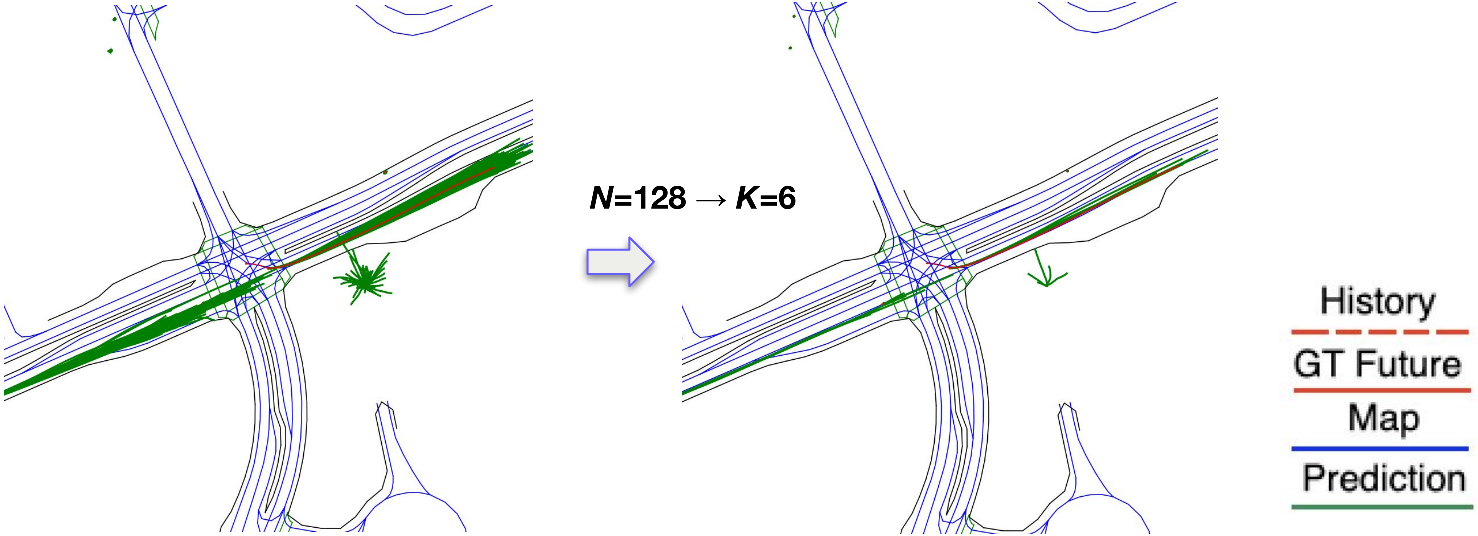
DTG: Current Architecture



Note: we can use **different encoders** (lstm-based, transformer-based)

DTG: Ensuring good minADE

- Vanilla **VDM** models the distribution of trajectories, the sampled *N* trajectories not necessarily have 1 close to GT
- We can mitigate it through clustering for getting a good minADE
  - And even probability as a size of cluster!

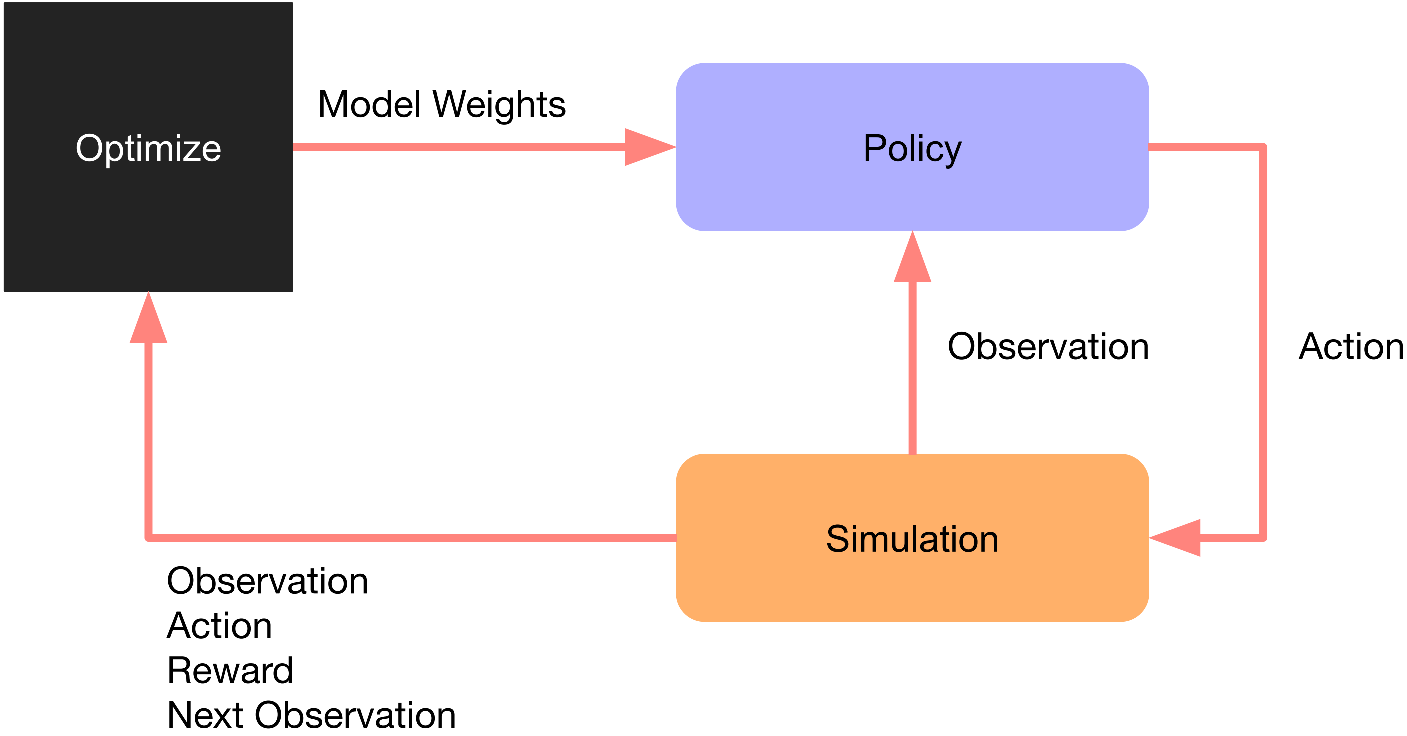


Kingma, Diederik, et al. "Variational diffusion models", 2021

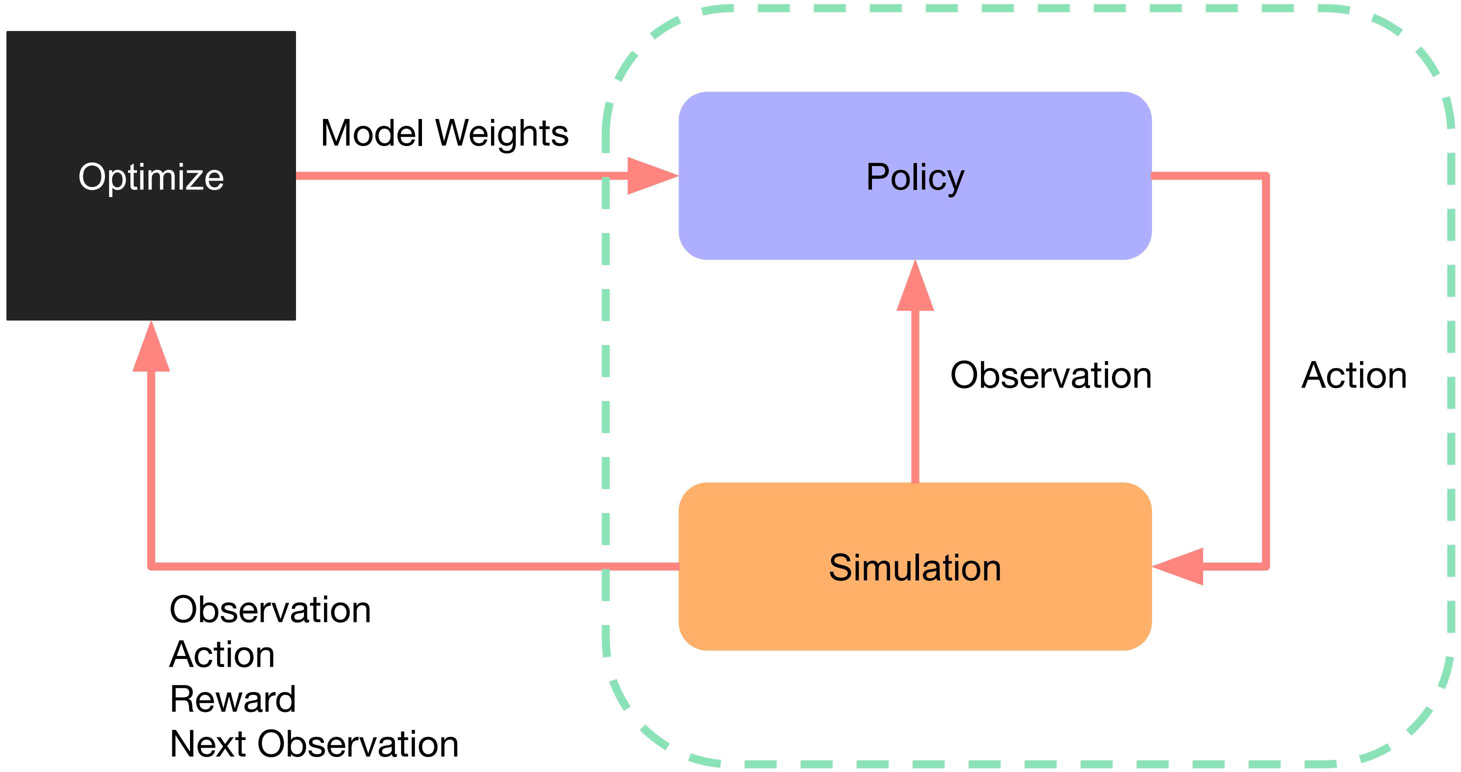
RL: Background and Motion Selection

03

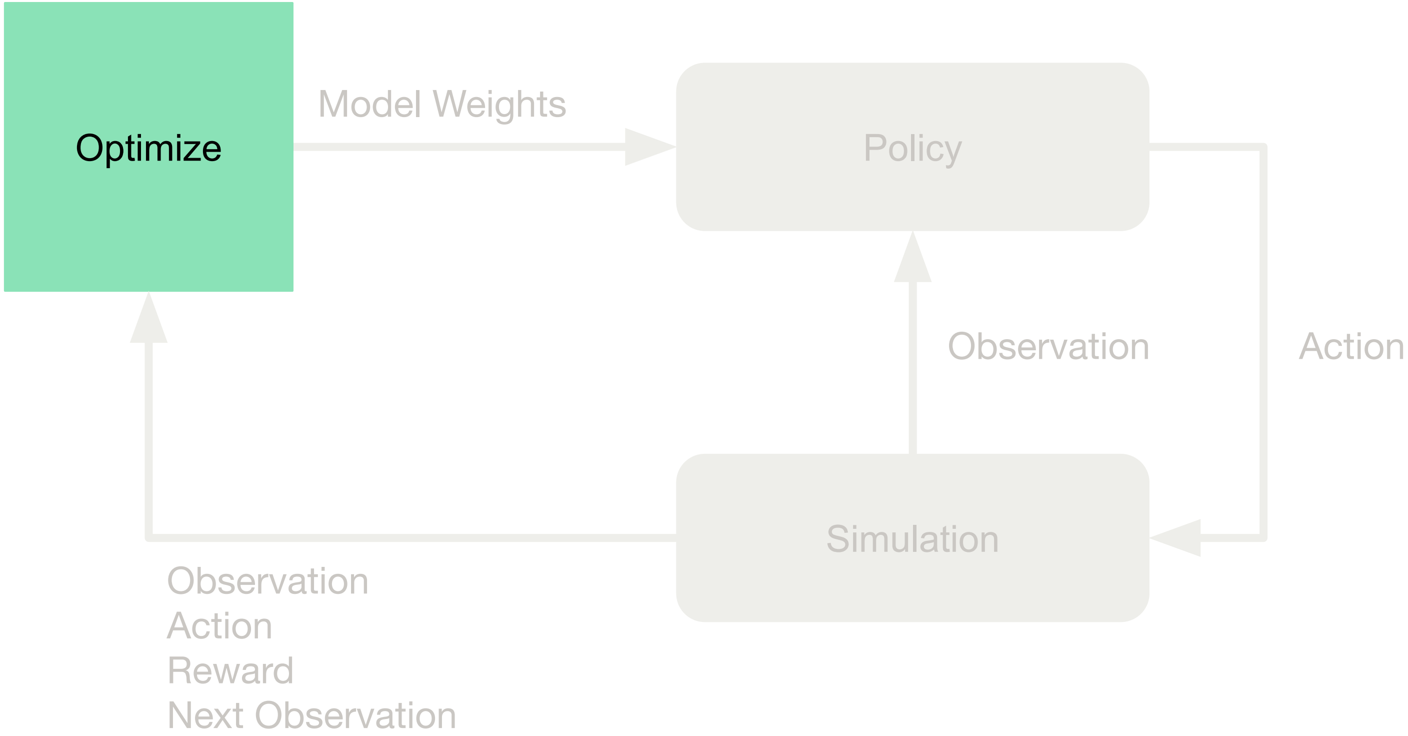
What is Deep RL?



What is Deep RL?



What is Deep RL?



How to Optimize?

Objective:
maximize reward under the policy while limiting probability of risky events

Learn:
state-action Q value function

Optimize:
iteratively improve Q for all s and a

What does this look like?

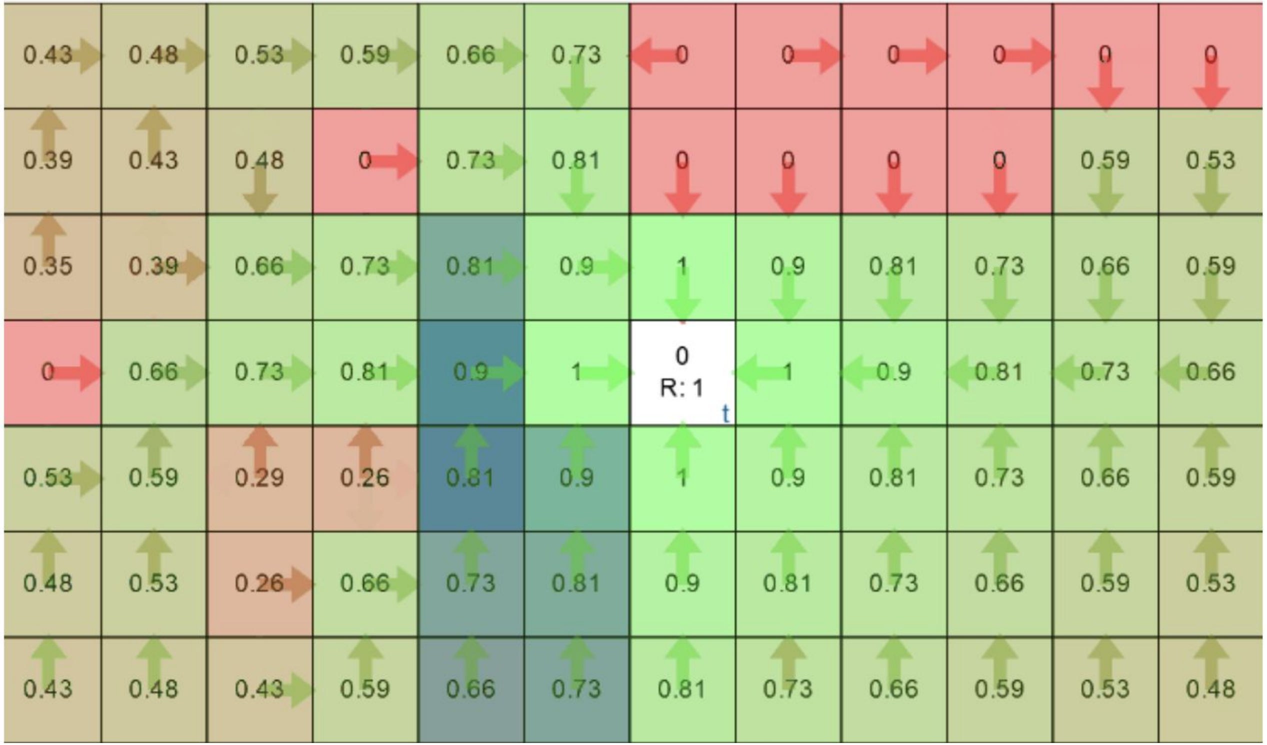


Image credit: https://towardsdatascience.com/interactive-q-learning-9d9203fdad70

RL for Selection

Why Motion Selection?

01

Discrete problem. Rank trajectories rather than produce them.

RLMS = RL for Motion Selection

02

Low-level decision making well handled by trajectory generation modules

Why Motion Selection?

01

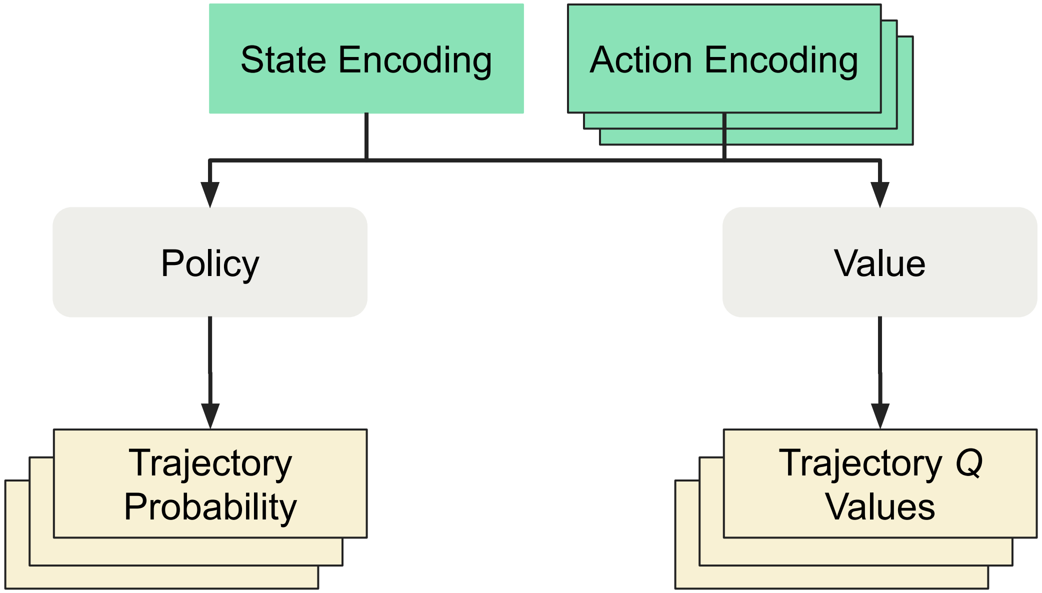
Discrete problem. Rank trajectories rather than produce them.

Allow heuristics and domain knowledge to filter the trajectory space for RL

02

Low-level decision making well handled by trajectory generation modules

Anatomy of the RLMS Model: Basic RL



Q values are **dense** rewards:

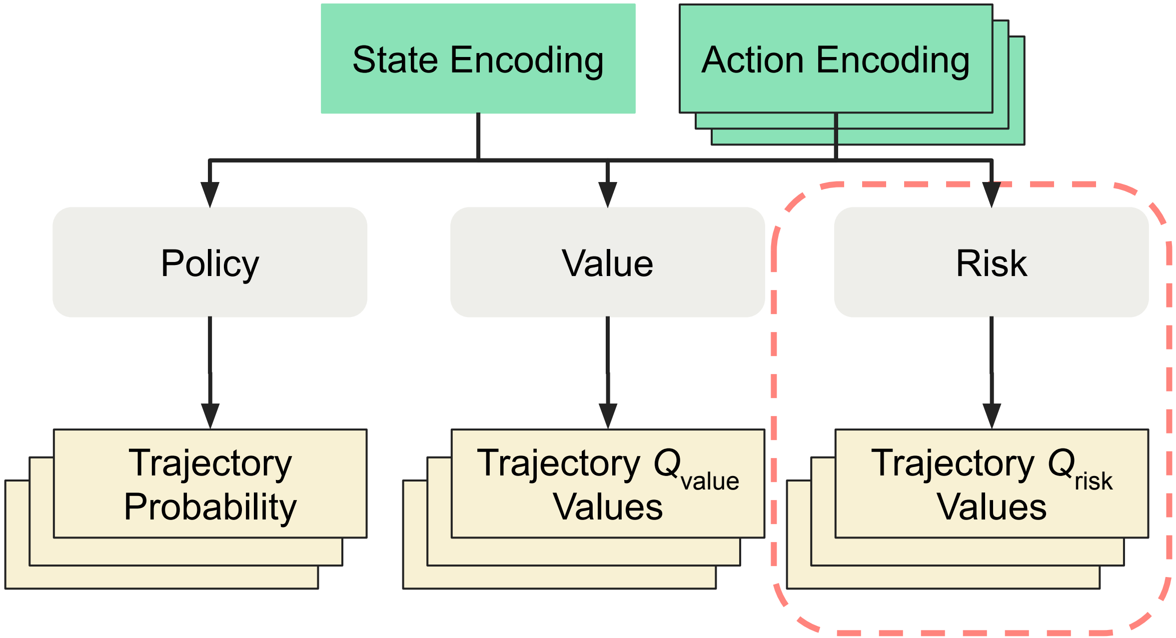
- Reward 1
- Reward 2
- Reward 3
- etc

Basic RL: Limitations

01

No concrete notion or **constraint** on safety

Anatomy of the RLMS Model: Risk Sensitive RL



Risk values are **sparse** rewards:

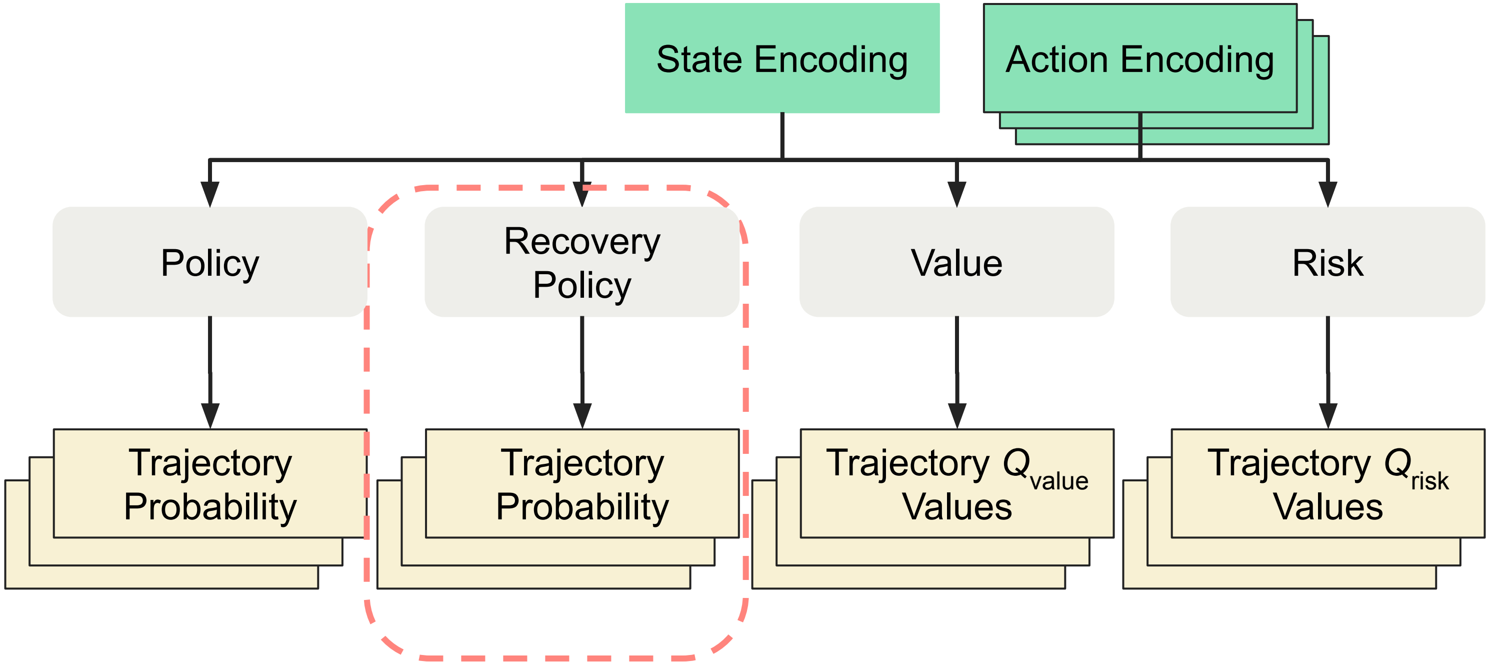
- Risk 1
- Risk 2
- Risk 3
- etc

Risk Sensitive RL: Limitations

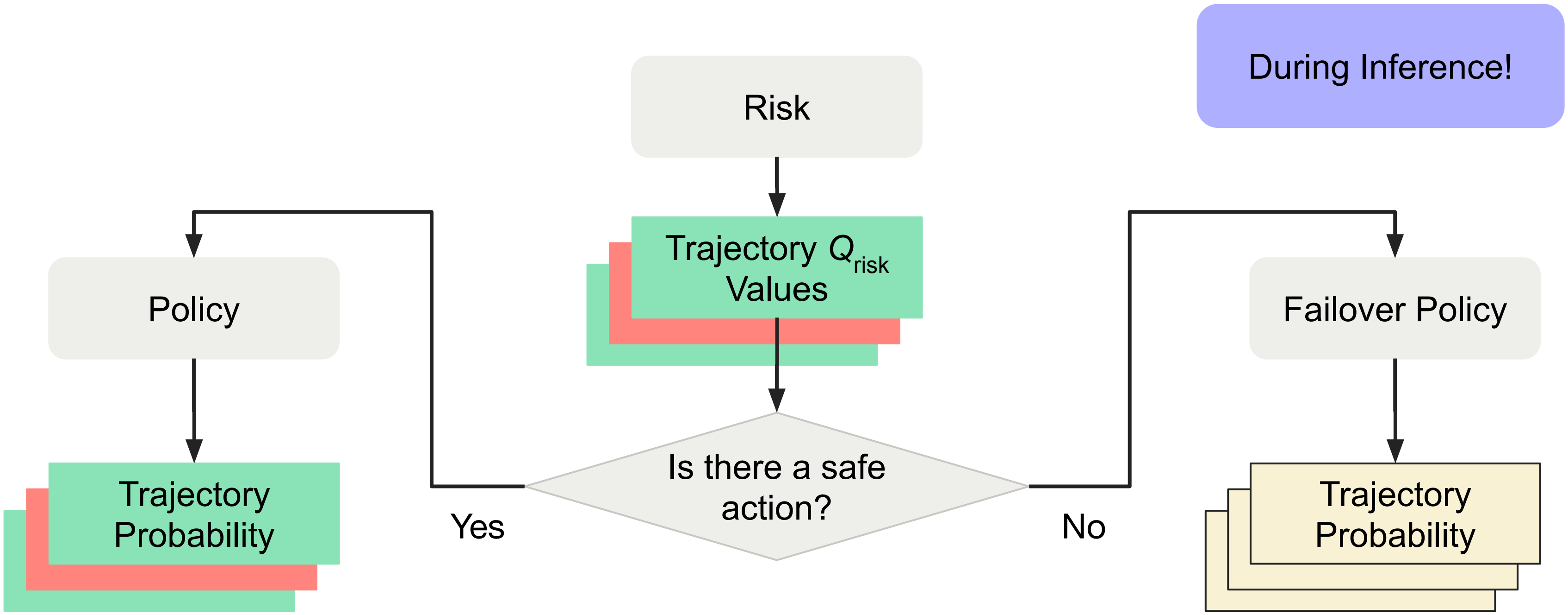
01

No **hard constraint** on safety

Anatomy of the RLMS Model: Constrained RL

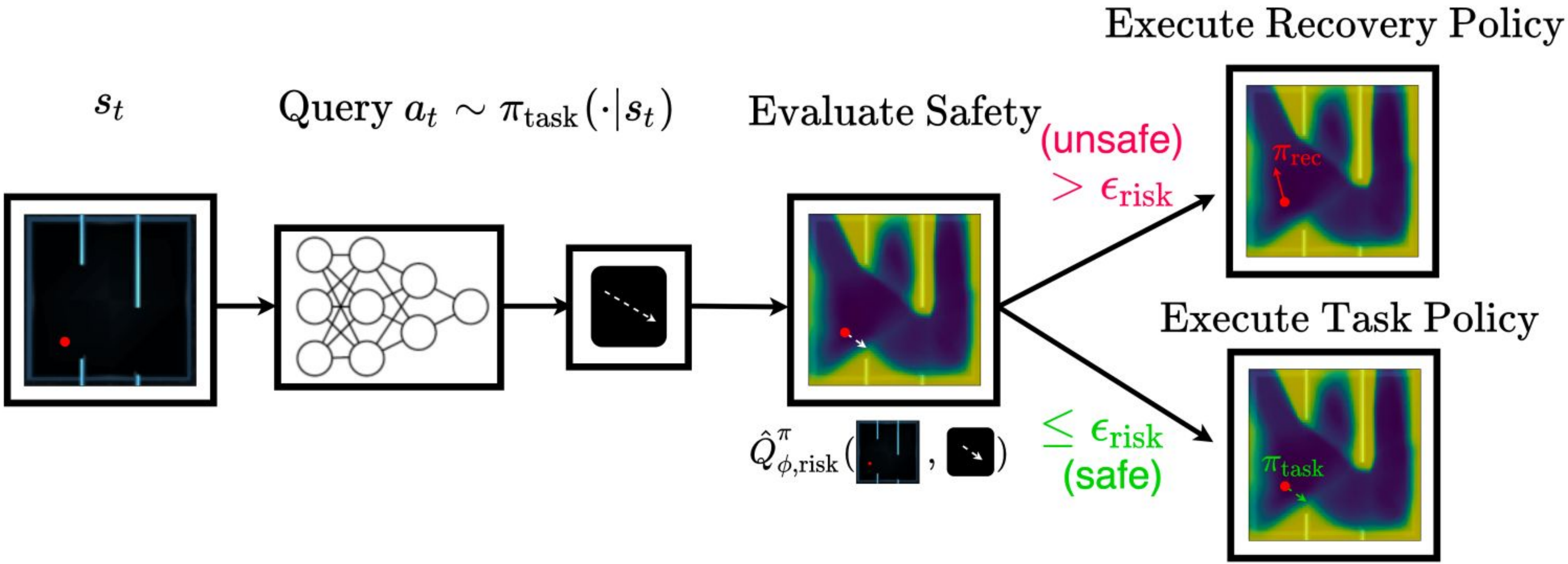


Anatomy of the RLMS Model: Constrained RL



Anatomy of the RLMS Model: Recovery RL

During Inference!



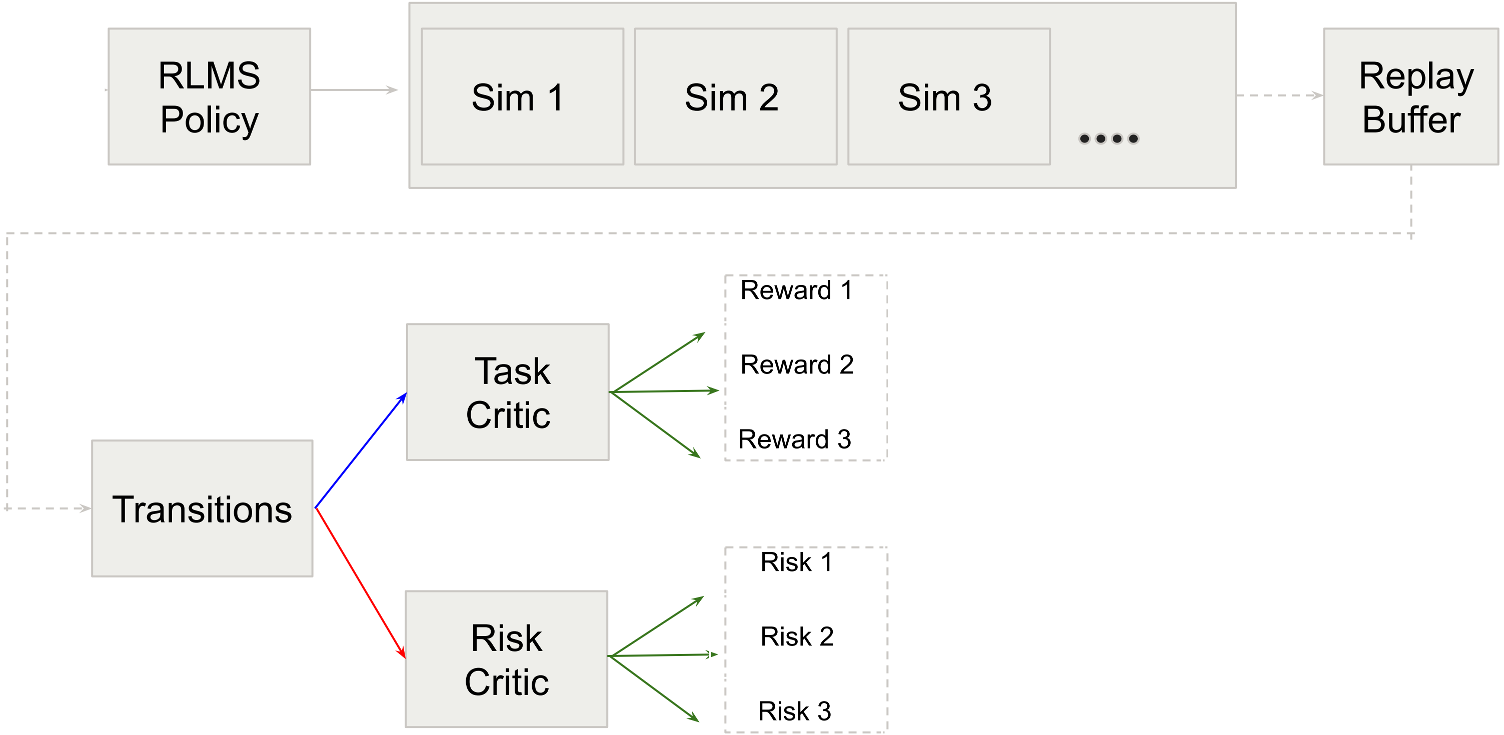
Thananjeyan, Brijen, et al. "Recovery rl: Safe reinforcement learning with learned recovery zones", 2021

RLMS: Block Diagram

First execute current RLMS policy in the simulator and store trajectories

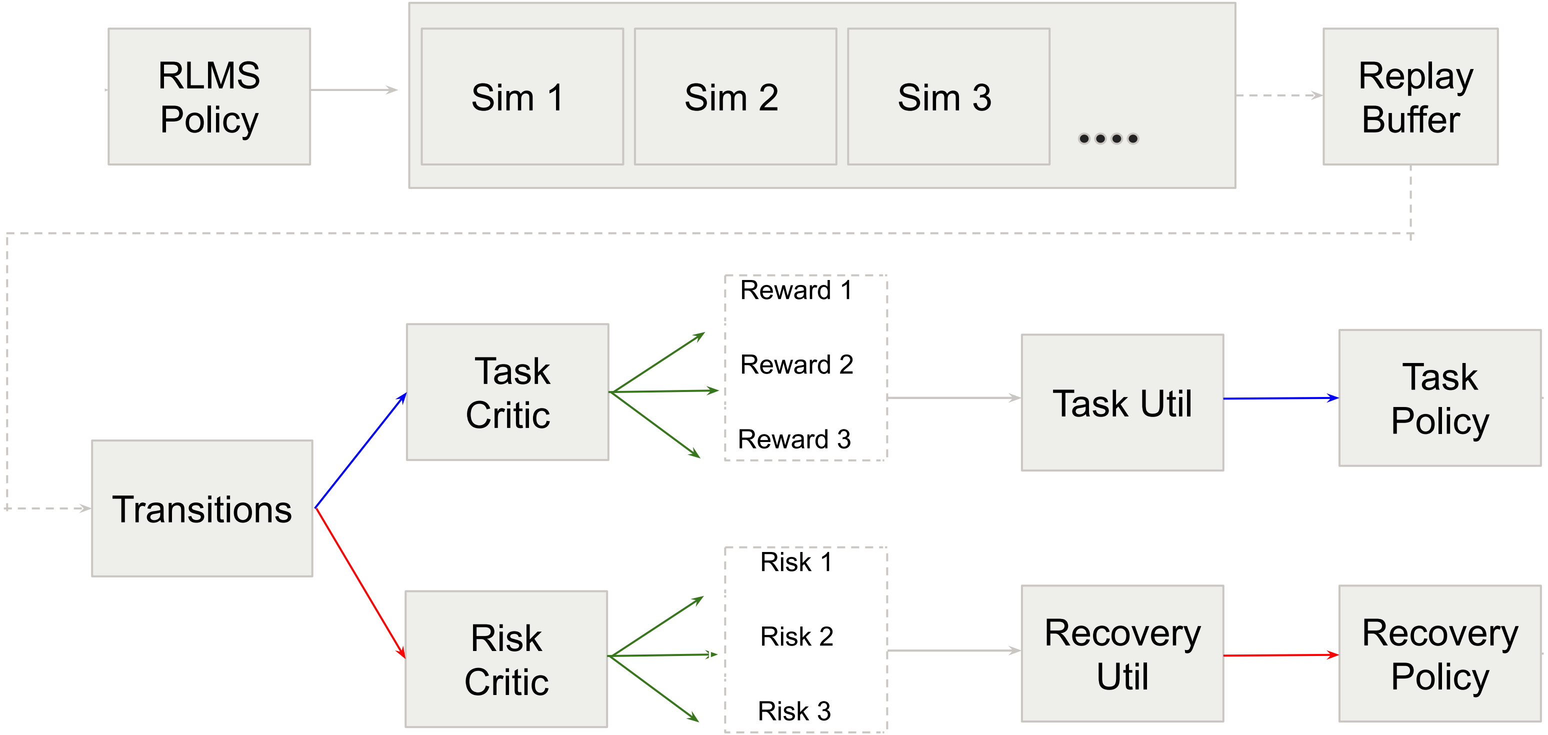
RLMS: Block Diagram

Use saved trajectories to train task critic and risk critic



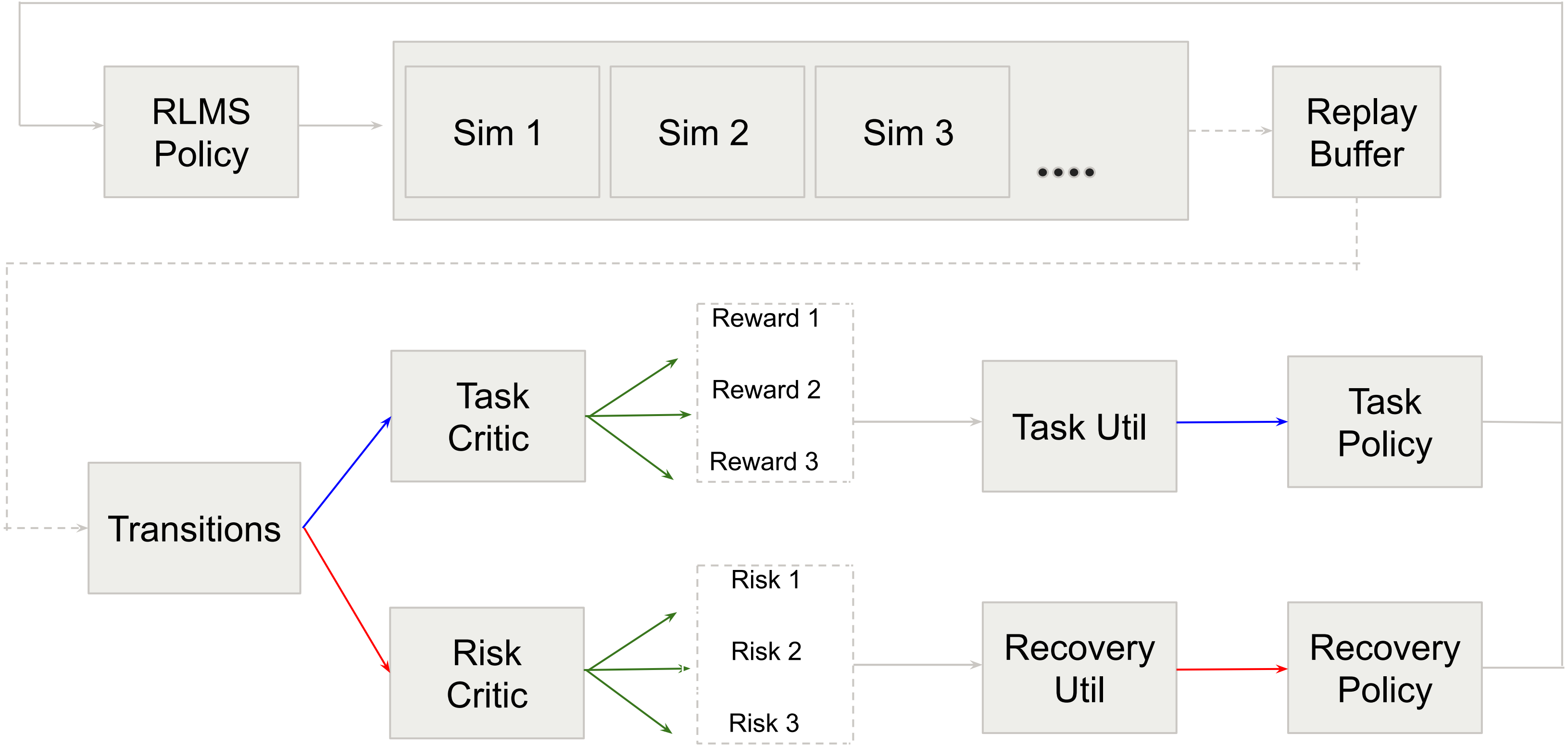
RLMS: Block Diagram

Combine task and risk critic values into utilities and train policies for each



RLMS: Block Diagram

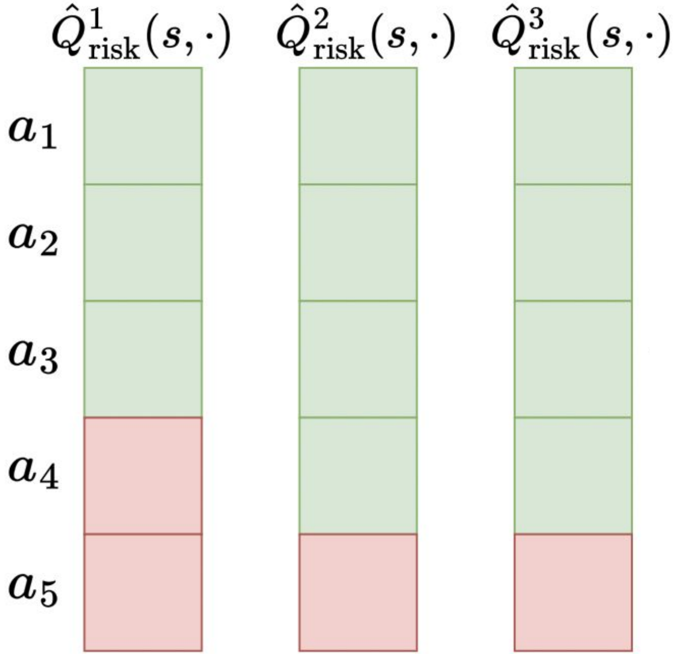
Combine task and recovery policy to get RLMS policy



Constructing RLMS Mixed Policy with Recovery RL

The final RLMS policy uses a combination of both the task and recovery policies to sample actions.

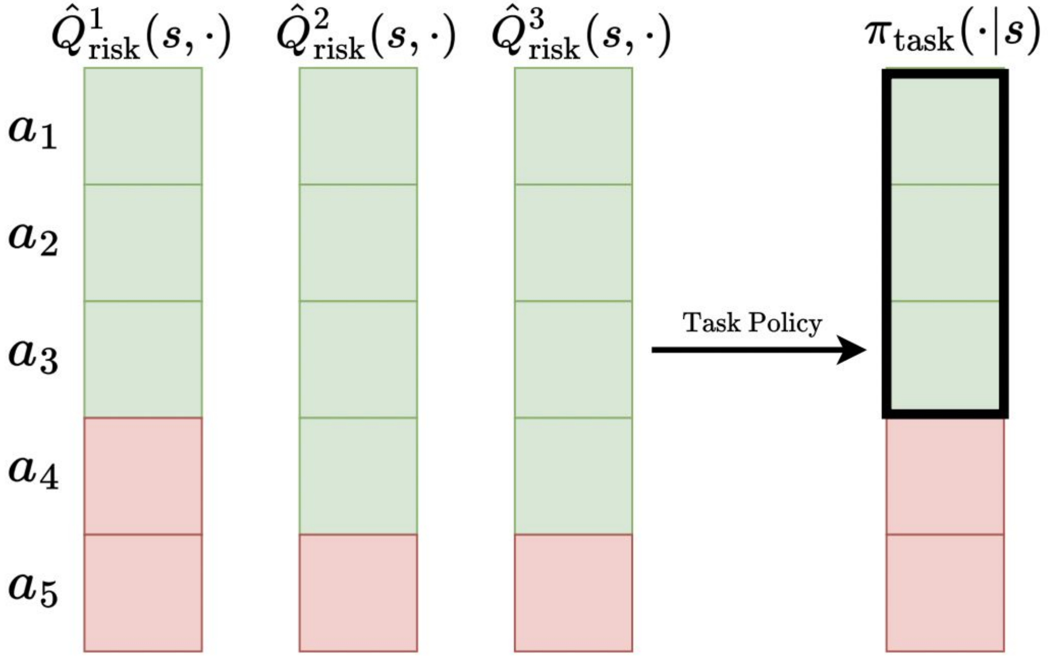
We first score all possible actions with each of our risk critics



Constructing RLMS Mixed Policy with Recovery RL

The final RLMS policy uses a combination of both the task and recovery policies to sample actions.

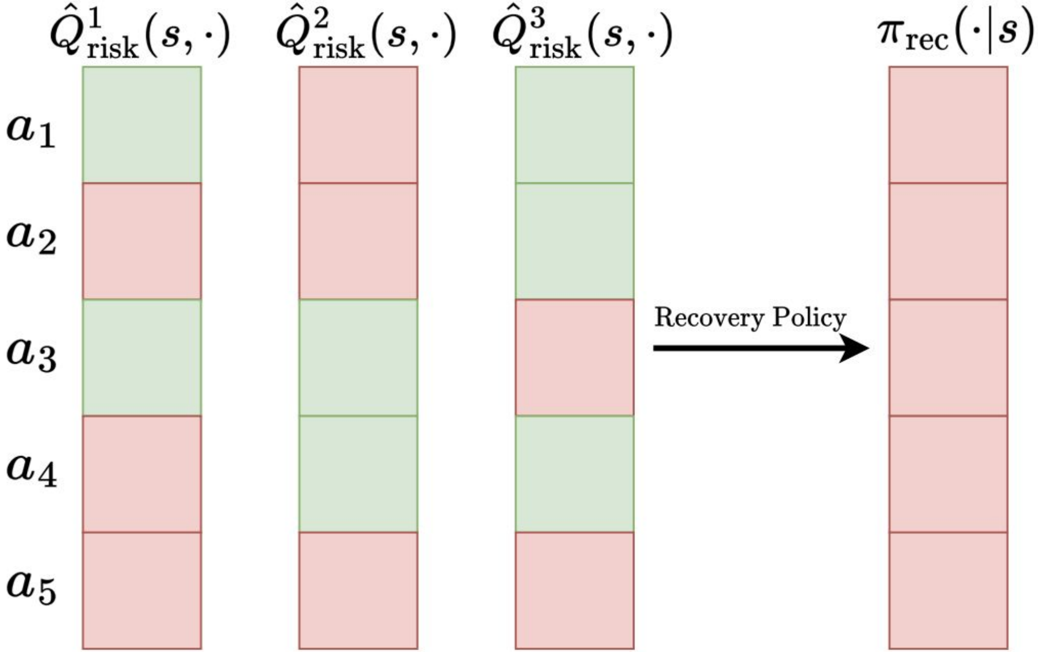
If there exist safe actions then sample from re-normalized task policy



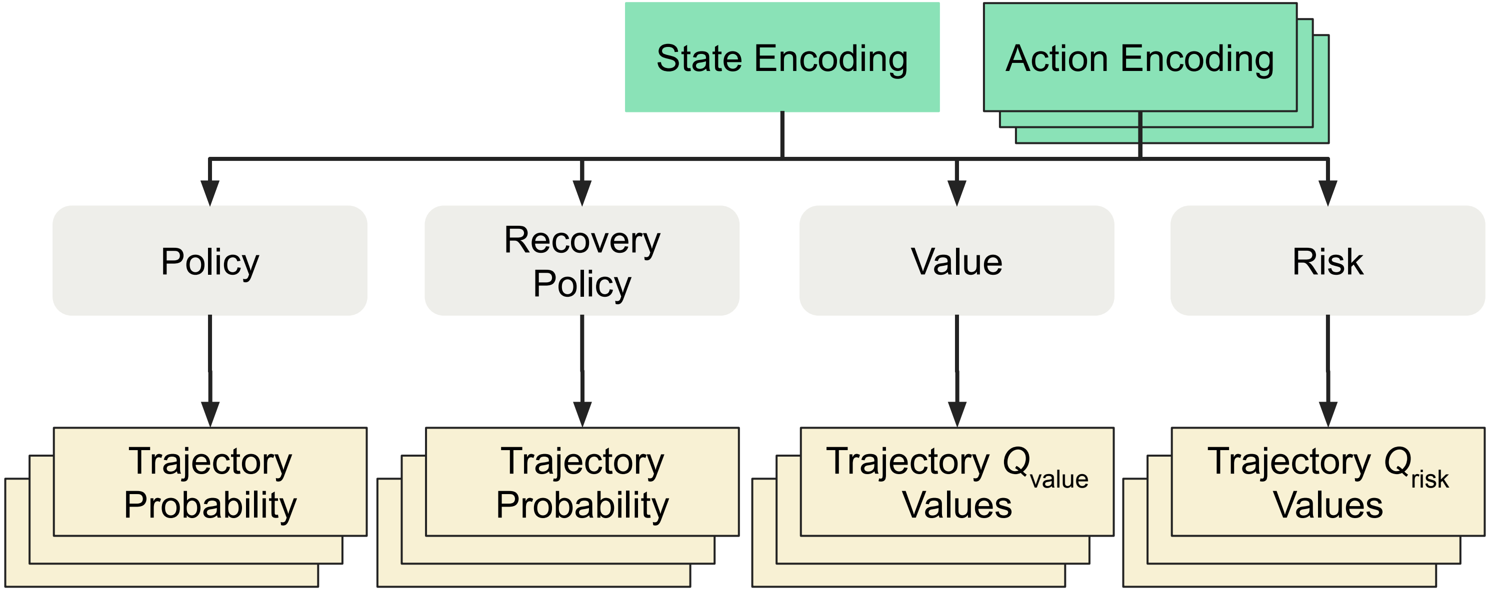
Constructing RLMS Mixed Policy with Recovery RL

The final RLMS policy uses a combination of both the task and recovery policies to sample actions.

Otherwise sample from recovery policy



Anatomy of the RLMS Model

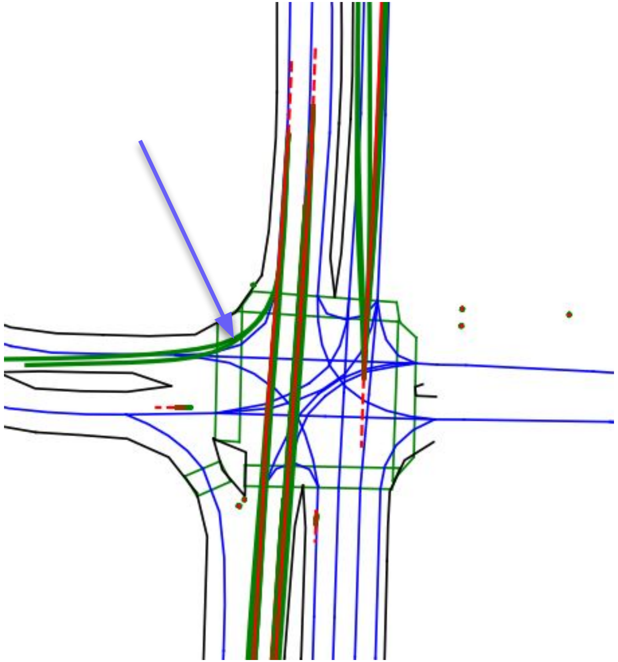


Examples

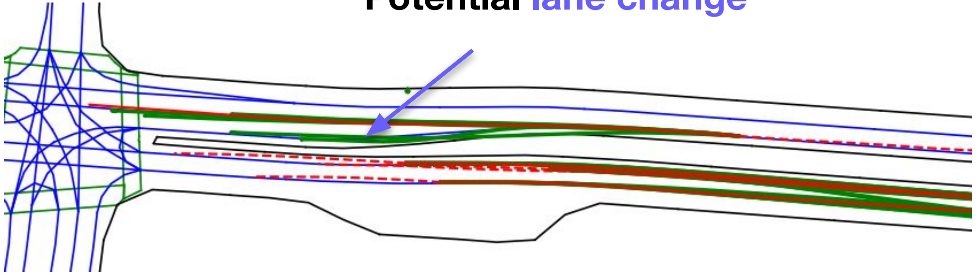
04

Experiments

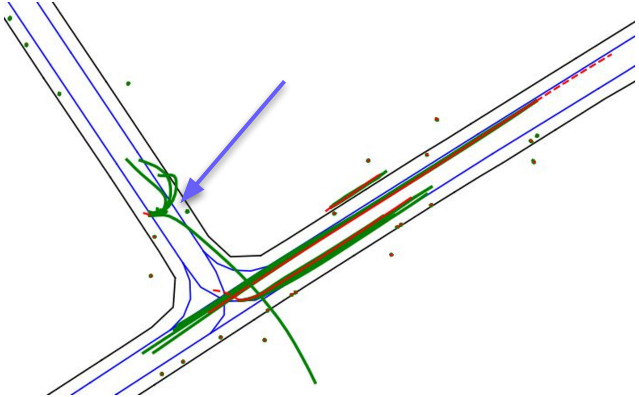
Potential right turn



Potential lane change

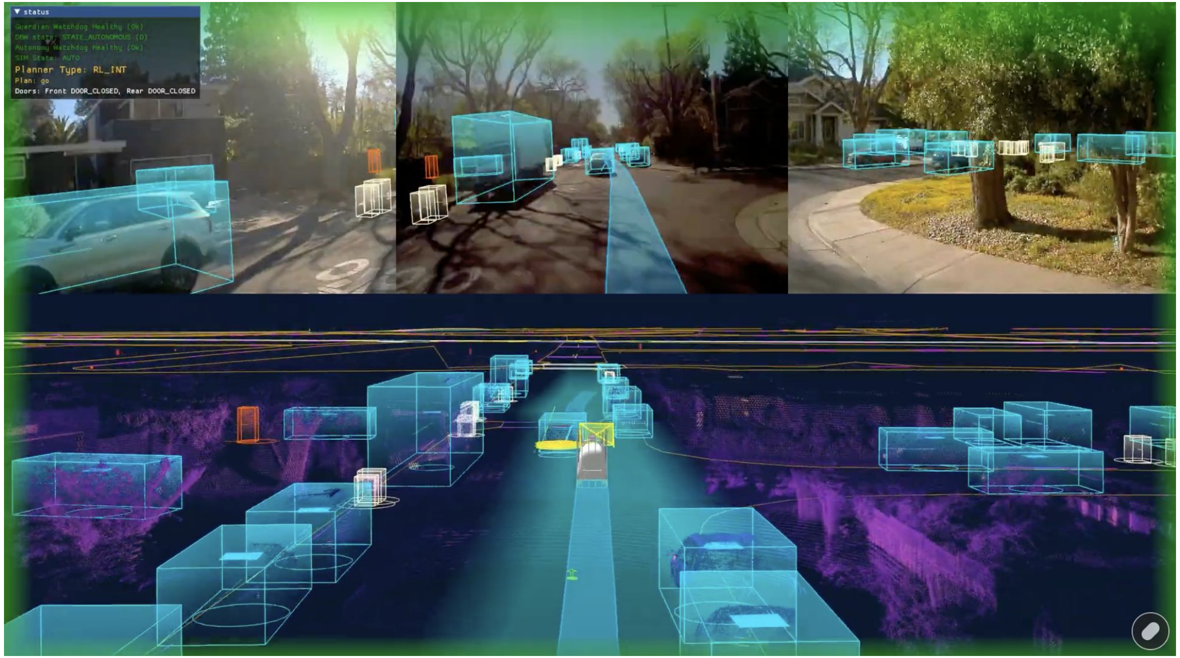


Uncertain Cyclist



Experiments

OK: Vehicle overtaking NuroBot on the left



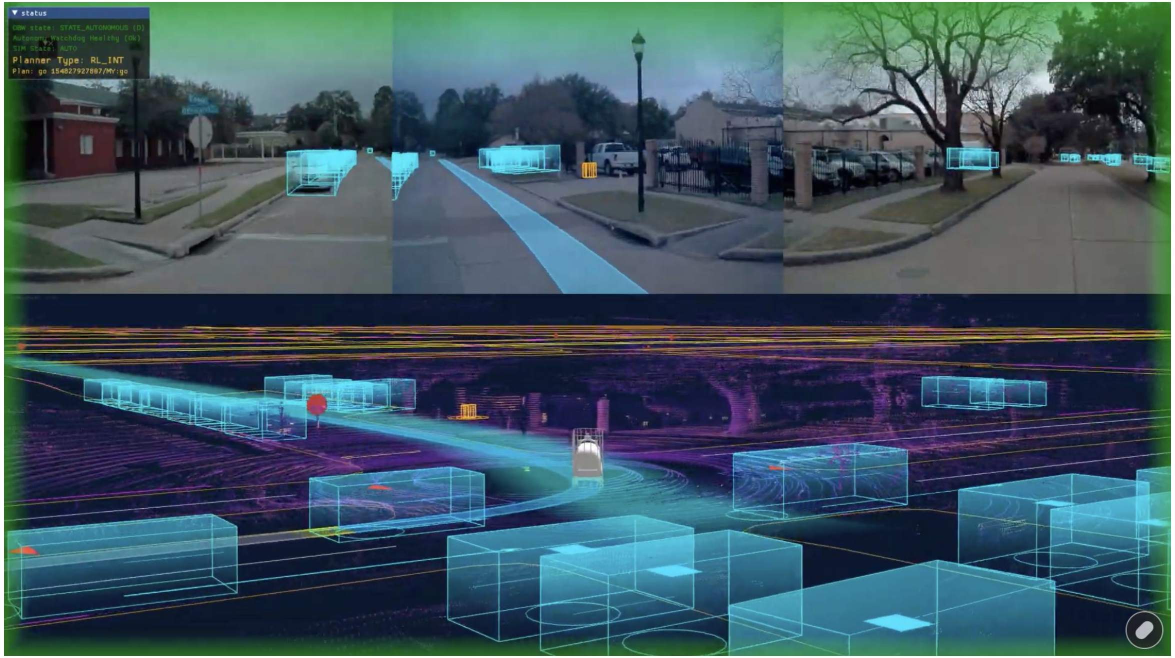
Top: **Onroad log**

Bottom: **Sim**

Video link: https://www.youtube.com/watch?v=FE7lR11uVB8

Experiments

OK: Occluded Unprotected Left



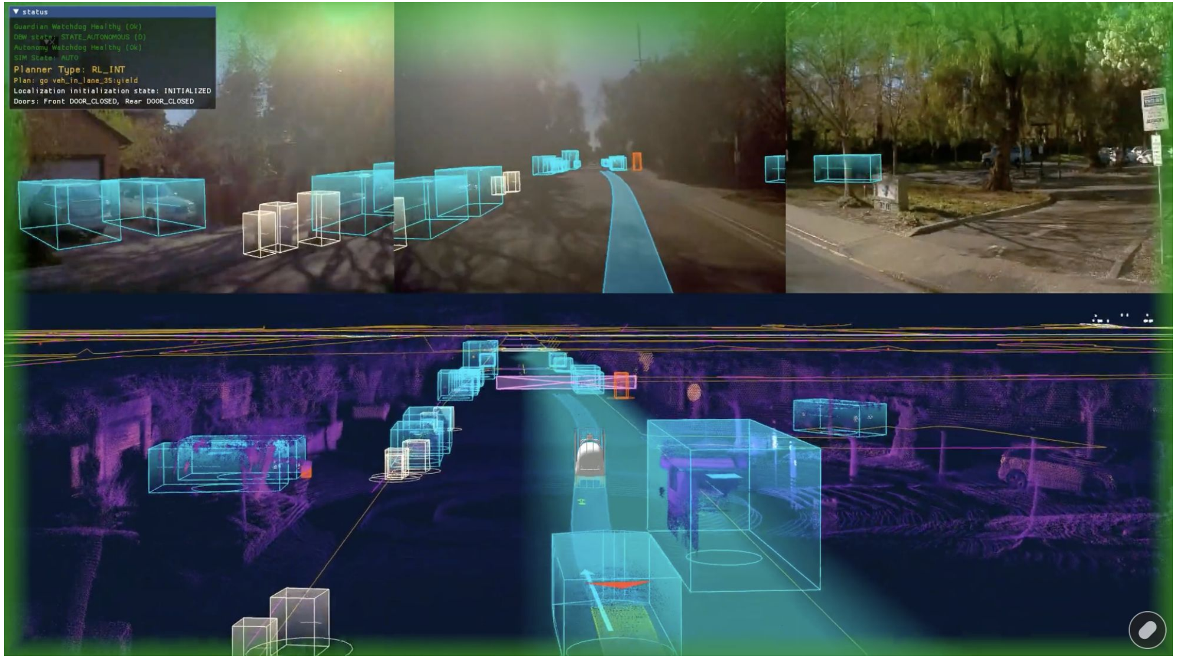
Top: **Onroad log**

Bottom: **Sim**

Video link: https://www.youtube.com/watch?v=scbIFi50oA8

Experiments

Not OK: Problems with stability - selection is a combination of plans because we don't have a single initial good source to choose from (making up its own plan via flicker yield)



Top: **Onroad log**

Bottom: **Sim**

Video link: https://www.youtube.com/watch?v=FE7lR11uVB8

Limitations and conclusions

05

DTG: Limitations

01

Sampling-only inference (hard to use in the production)

02

Latency-performance tradeoff

03

Non-deterministic simulation

RLMS: Limitations

01

Still no *hard* constraint on safety

02

Rare sparse events still challenging to learn (i.e. collisions)

03

Sample inefficient – takes many simulation steps to learn

Conclusions

01

Diffusion-based models help to match the distributions, not points

02

Learning selection provides long-horizon reasoning

03

Recent academic SotA can be used for practical tasks to add more safety!

Useful Links

- [MVHS]: Autonomy: Introduction of ML for High School (presentation)
- [BDD]: Autonomy Challenges (presentation, video)
- [BAIR]: Autonomy: Open Questions (presentation)
- [CVPR]: Behavior Modeling and Learned Motion Selection for Safe Driving (presentation)
- [YT]: Nuro Tech Talks (playlist)