ICIP 21

Reinforced Curriculum Learning

FOR AUTONOMOUS URBAN DRIVING IN CARLA

Luca Anzalone (UniBo), Dr. **Silvio Barra** (UniNa), Prof. **Michele Nappi** (UniSa)

Outline

Autonomous Driving

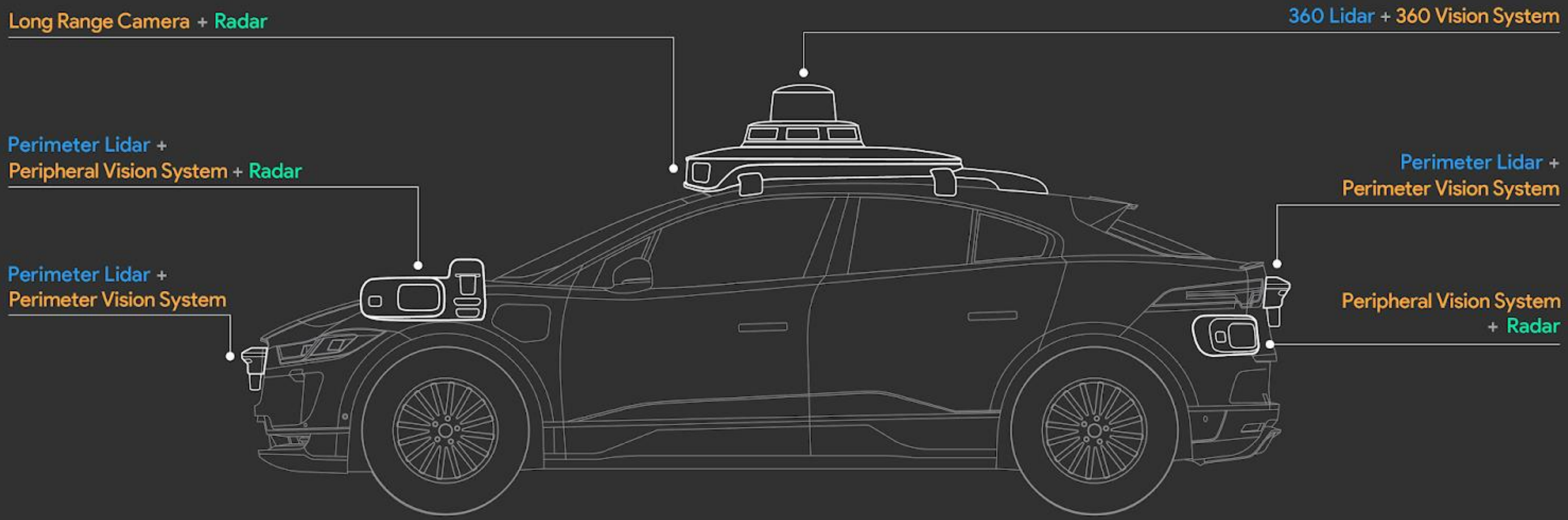
Reinforcement Learning

CARLA

Proposed Approach

Autonomous Driving

Autonomous Vehicles: **Sensors**



credit: Waymo

SAE AUTOMATION LEVELS

LEVEL 0

There are no autonomous features.

LEVEL 1

These cars can handle one task at a time, like automatic braking.

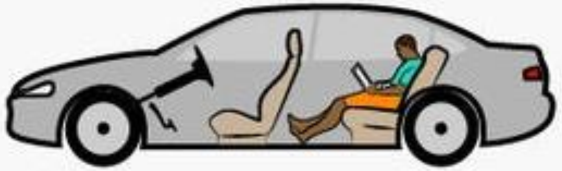
LEVEL 2

These cars would have at least two automated functions.

LEVEL 3

These cars handle "dynamic driving tasks" but might still need intervention.

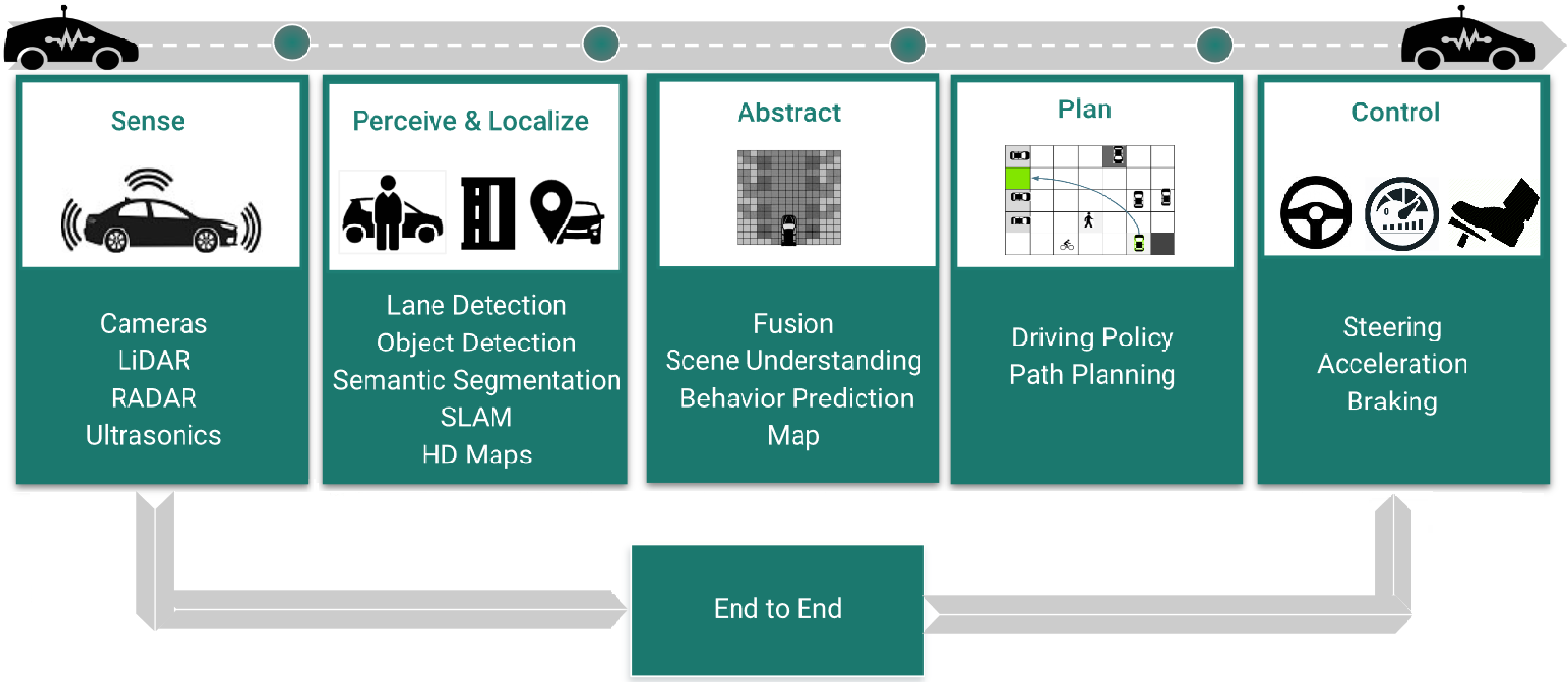
LEVEL 4

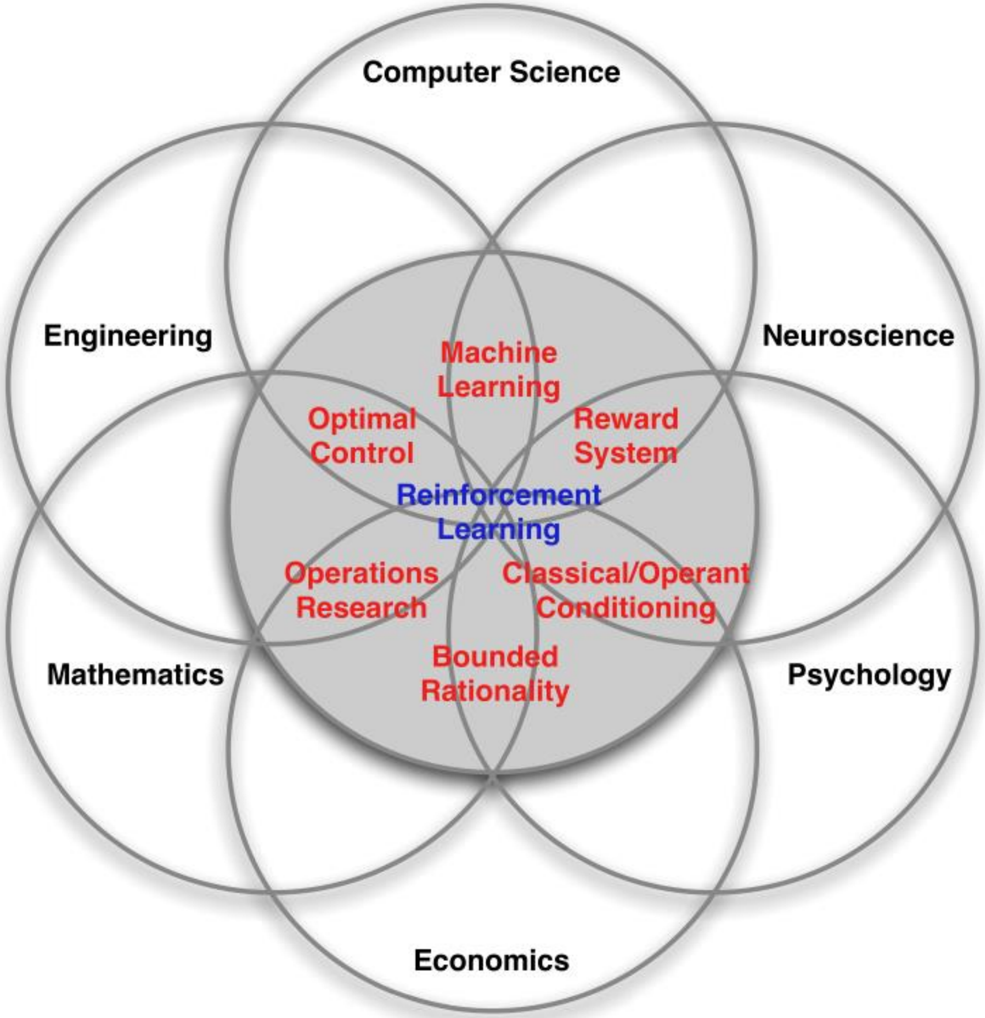


These cars are officially driverless in certain environments.

LEVEL 5

These cars can operate entirely on their own without any driver presence.

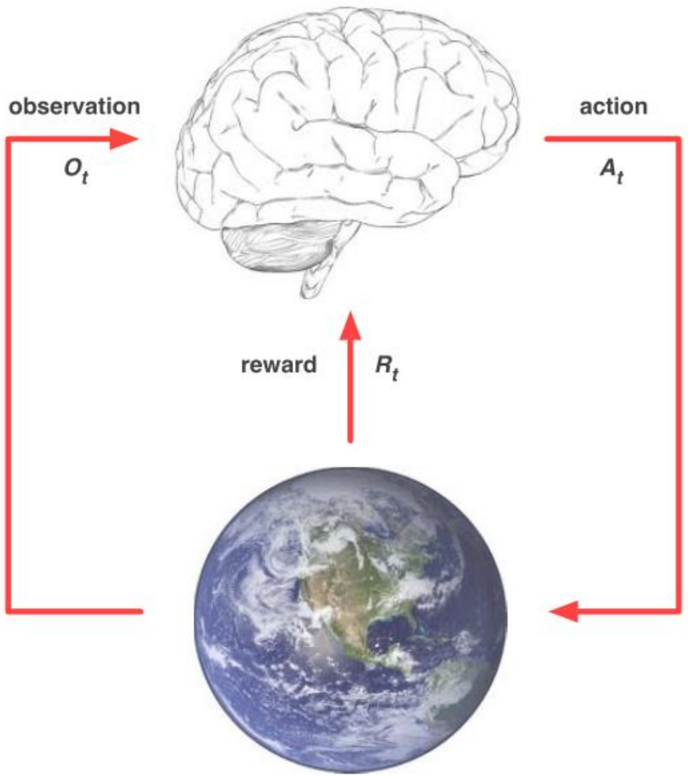

Sense
Cameras
LiDAR
RADAR
Ultrasonics
Perceive & Localize
Lane Detection
Object Detection
Semantic Segmentation
SLAM
HD Maps
Abstract
Fusion
Scene Understanding
Behavior Prediction
Map
Plan
Driving Policy
Path Planning
Control
Steering
Acceleration
Braking
End to End



Reinforcement Learning

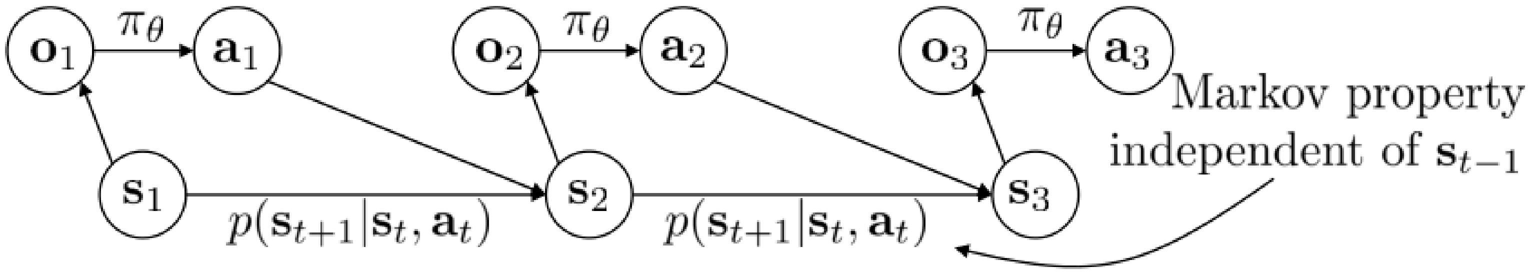
Credits: D. Silver & S. Levine

Markov Decision Processes (**MDPs**)



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

- Environment = MDP
- MDP $M = \langle S, A, P, r, \gamma \rangle$:
  - S: state space
  - A: action space
  - $P(s' \mid s, a)$
  - $r: S \times A \to \mathrm{R}$
  - $\gamma \in (0,1]$: discount factor
- Transition (s, a, s', r)
- Trajectory $\tau = \{(s_t, a_t, s_{t+1}, r_t)\}_t$



Performance Objective

Difficult to maximize directly!

Sampling is involved:
- High-variance,
- Unstable.

$\rho(s_1)$ = initial state distribution.

Trajectory $\tau = (s_1, a_1, s_2, a_2, s_3, \dots, a_{T-1}, s_T)$

$$\theta^\star = \arg\max_\theta \mathbb{E}_{(s_t,a_t)\sim p_\theta}\left[\underbrace{\sum_{t=1}^{T}\gamma^{t-1} r(s_t, a_t)}_{\text{return } R_t}\right]$$

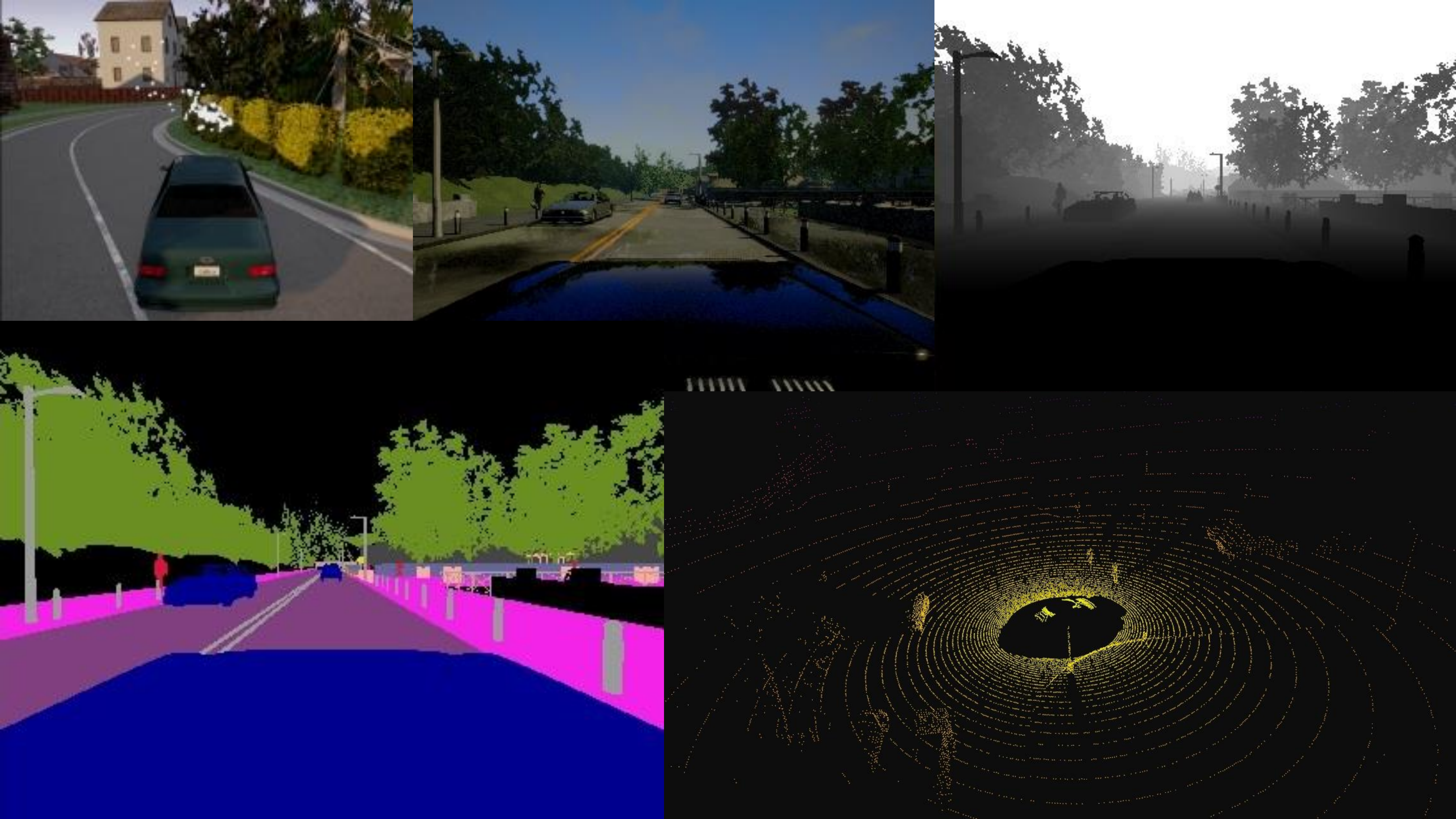
$$p_\theta(\tau) = \rho(s_1)\prod_{t=1}^{T}\mathcal{P}(s_{t+1} \mid s_t, a_t)\ \pi_\theta(a_t \mid s_t),$$

$$J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\gamma^{t-1} r\left(s_t^{(i)}, a_t^{(i)}\right)$$

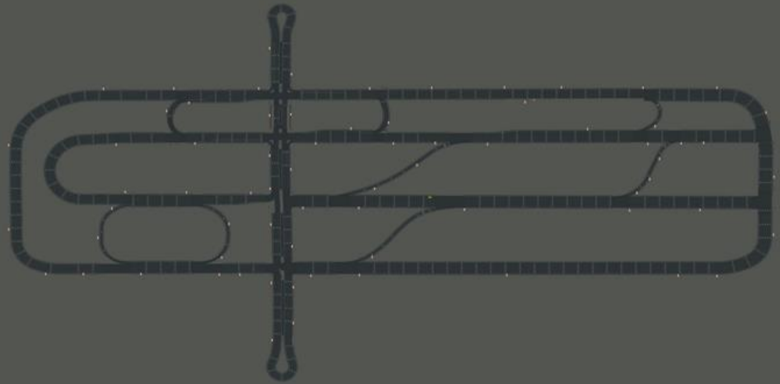
$$\theta^\star = \arg\max_\theta J(\theta)$$

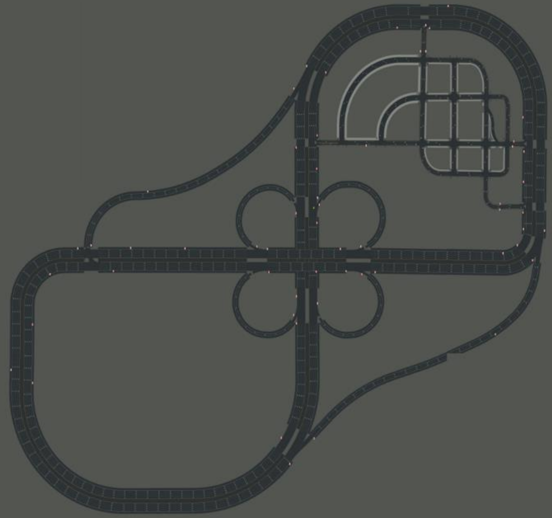
CARLA

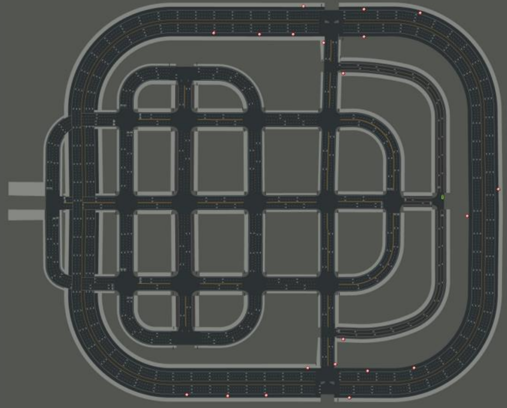
AUTONOMOUS DRIVING SIMULATOR

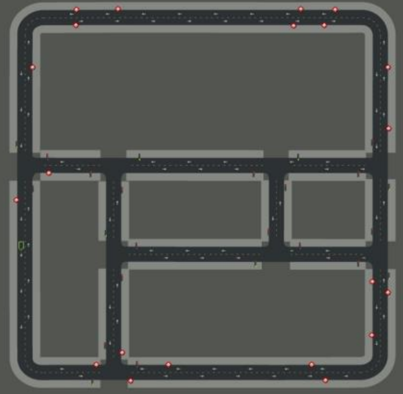


Town 3









Reinforced Curriculum Learning

Bengio et al. Curriculum Learning (2009)

Reinforced Curriculum Learning

Combine **Deep Reinforcement Learning** with **Curriculum Learning**!

Curriculum learning guides agent training.

Per stage: 500 episodes, 512 timesteps ⇒ 5x1.28M timesteps.

Training is divided into **five stages** of increasing difficulty:

- Same map (**town03**) and *same* vehicle.
- Start from few fixed *spawn locations.*
- *No*, *regular*, *dense* traffic: up to **100 vehicles** and **200 pedestrians**.
- Change **weather conditions**: day/sunset/night, clear/cloudy/rain.
- Data augmentation.

Reinforcement learning further improves policy at each stage.

Reward Function

$$r_t = \begin{cases} -c_p & \text{if collision,} \\ s_{\text{limit}} - v_{\text{speed}} & \text{if } v_{\text{speed}} > s_{\text{limit}}, \\ \frac{v_{\text{speed}} \cdot v_{\text{sim}}}{(d_w/2)^2} & \text{otherwise} \end{cases}$$

c_p: collision penalty,
d_w: distance to next waypoint w,
v_{sim}: cosine similarity with vehicle heading and w.

s_{limit}: current speed limit,
v_{speed}: actual vehicle speed.

Observation Space

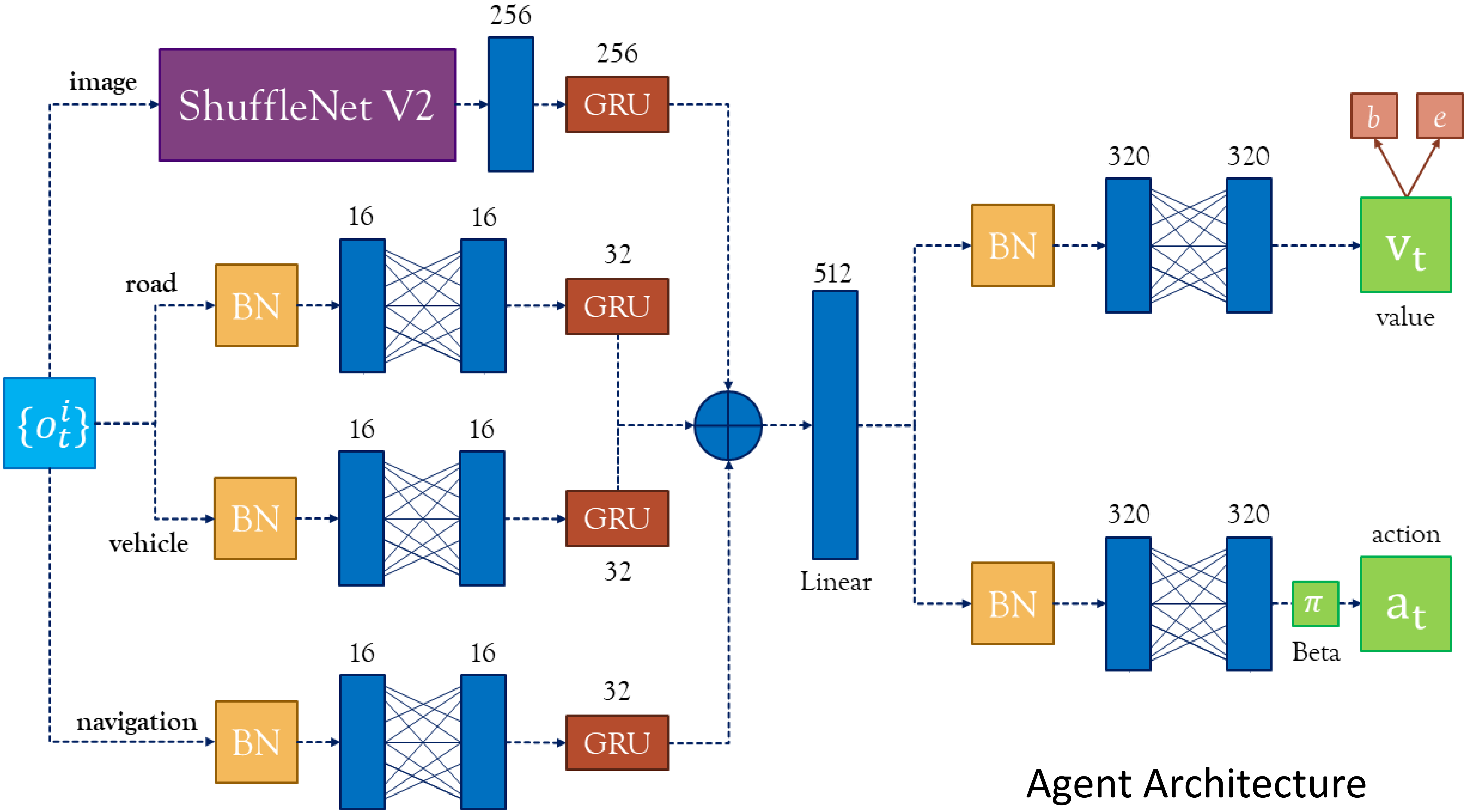
Stack of *four tensors* in time: a tensor is stacked after **discarding three frames**.

The observation $\boldsymbol{o}_t = \{[\boldsymbol{I}, \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{N}]_k\}_{k=1}^{4}$:

- **Image** I: shaped $90 \times 360 \times 3$; is the concatenation of three $90 \times 120 \times 3$ RGB images from left, middle and right camera sensors.
- **Road features** G (9-dimensional): *is intersection*, *is junction*, *is at traffic light*, *speed limit*, and *traffic light state* (5-dimensional one-hot encoded).
- **Vehicle features** V (4-dimensional): *similarity* (i.e., cosine similarity between heading direction and next route waypoint), *speed*, *throttle*, and *brake* values.
- **Navigational features** N (5-dimensioal): vector of *five distances* from current vehicle location to next five waypoints.

Image Augmentations

Color distortion, **Gaussian blur**, **Gaussian noise**, **salt-and-pepper noise**, **cutout**, and **coarse dropout**.



Agent Architecture

Base-Exponent **Value Decomposition**

$$\boxed{\mathcal{L}_v(\phi) = \sum_{t=0}^{T-1} \frac{(b_{v_t} - b_{R_t})^2}{\alpha} + \frac{(e_{v_t} - e_{R_t})^2}{\beta}}$$

$$b \in [-1, 1]$$

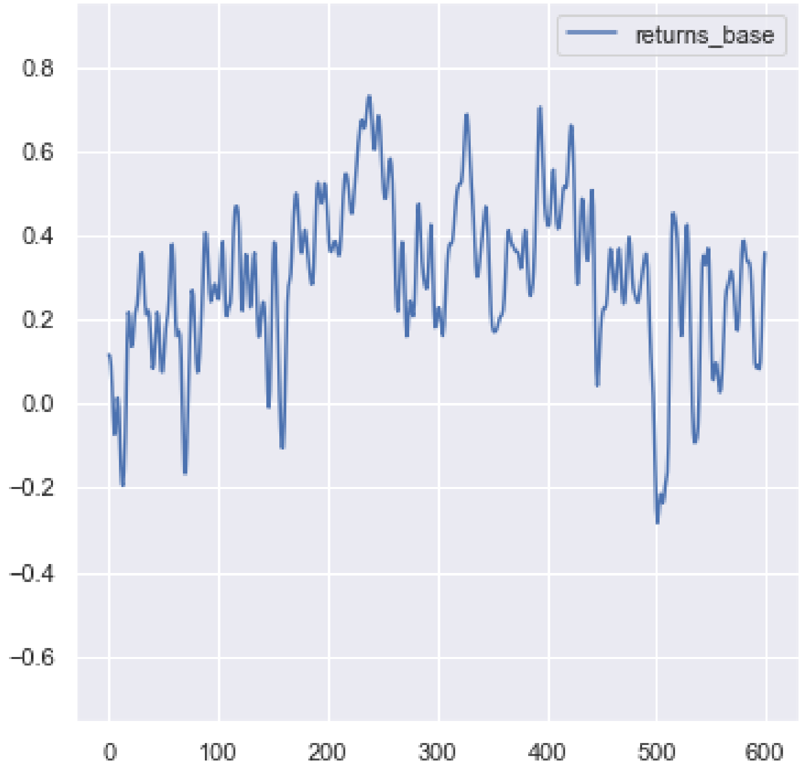
$$e \in [0, k]$$

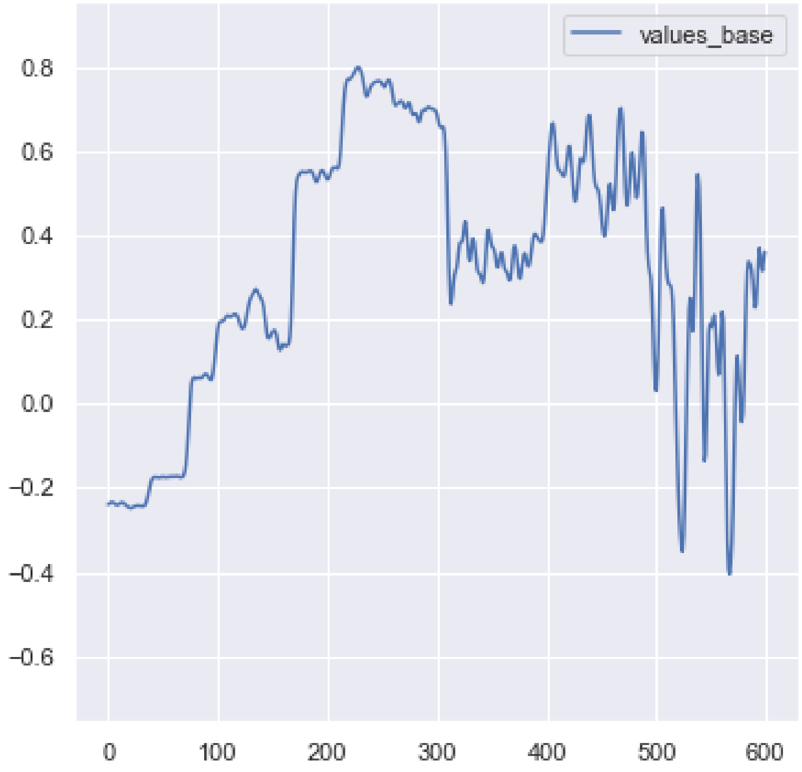
$$R_t = \sum_{i=t}^{T-1} \gamma^i r_i$$

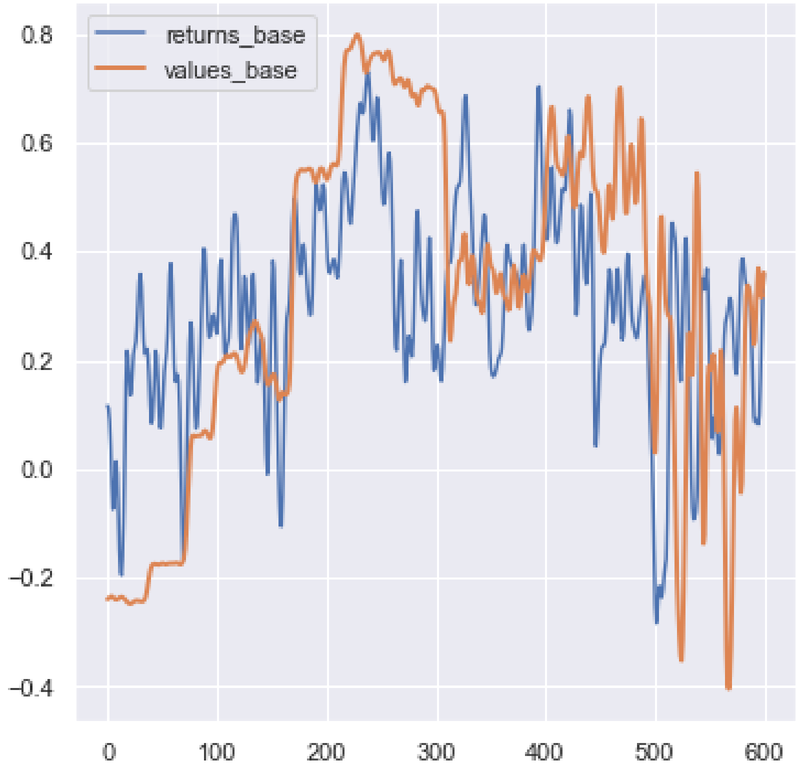
$$\alpha = 4, \beta = k^2$$

$$\boxed{v = b \cdot 10^e}$$

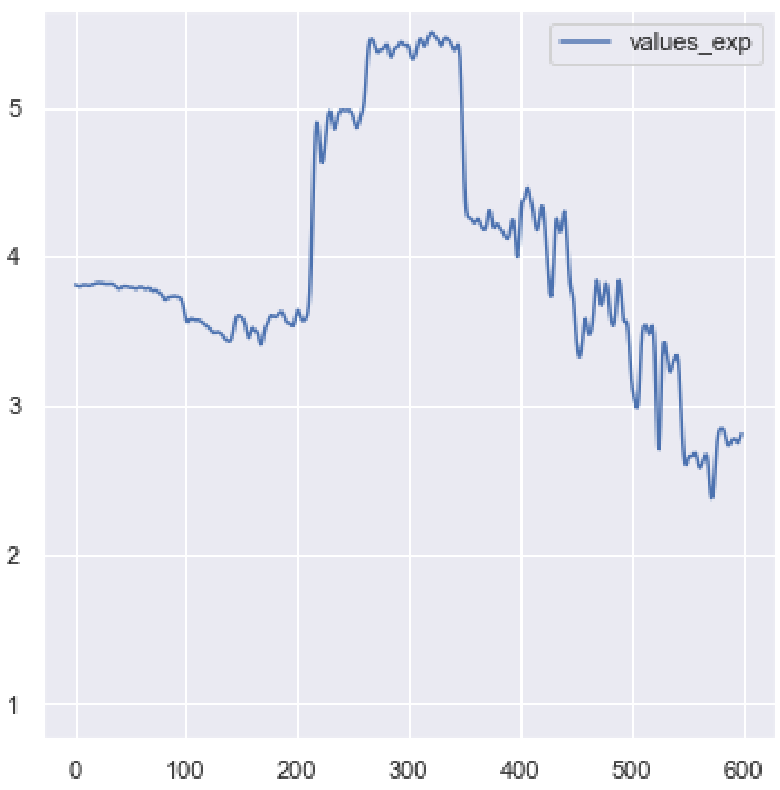
Regress **Base** b

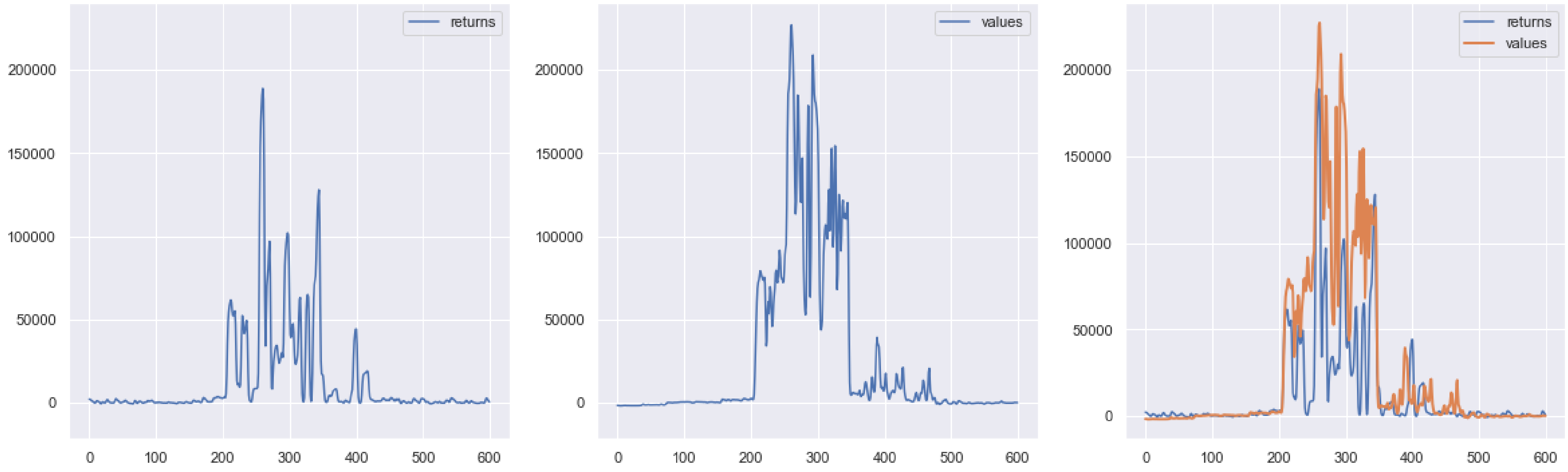






Regress **Exponent** *e*





returns
200000
150000
100000
50000
0
0
100
200
300
400
500
600
values
200000
150000
100000
50000
0
0
100
200
300
400
500
600
returns
values
200000
150000
100000
50000
0
0
100
200
300
400
500
600

Sign-preserving **Advantage Normalization**

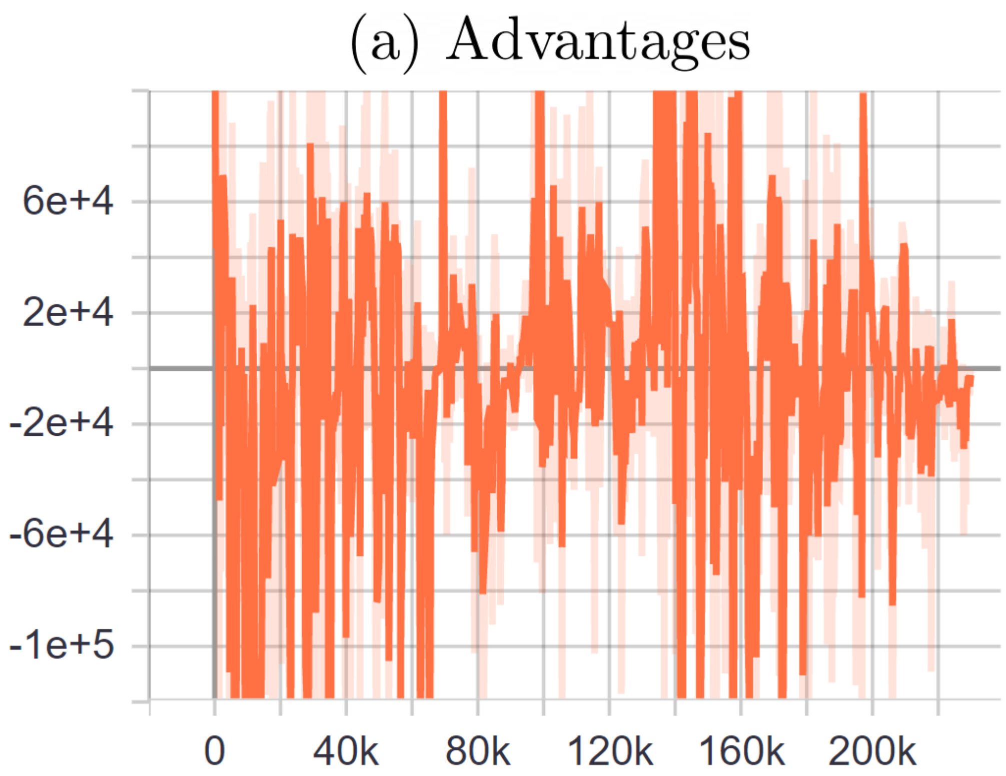
$$A_t \approx R_t - \hat{V}_t$$

If values $\hat{v}_t$ are **not accurate**, advantages A_t will be **large**!

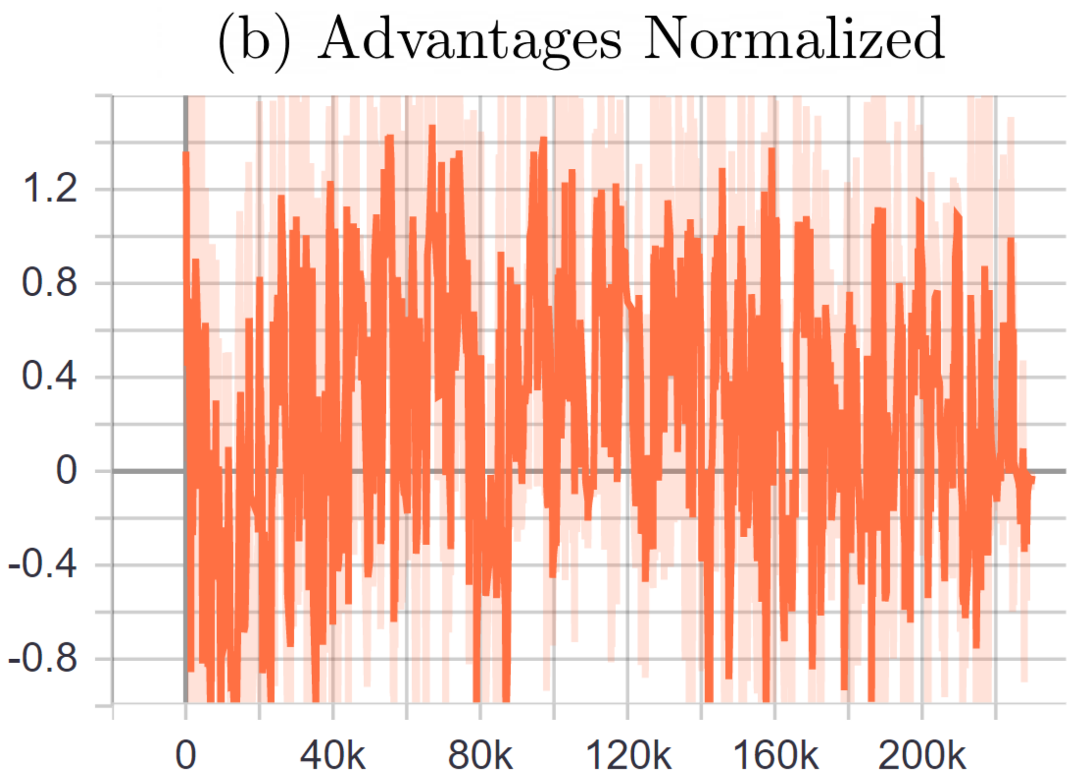
Remember: $\nabla_\theta \mathcal{L} = \nabla_\theta \log \pi_\theta(a \mid s) A(s, a)$

Normalizes **negative** and **positive** advantages **separately** ⇒ sign is preserved.

```
def sign_preserving_normalization(adv):
  pos = adv * float(adv > 0.0)   # mask
  neg = adv * float(adv < 0.0)
  return (pos /  tf.reduce_max(adv)) +
         (neg / -tf.reduce_min(adv))
```

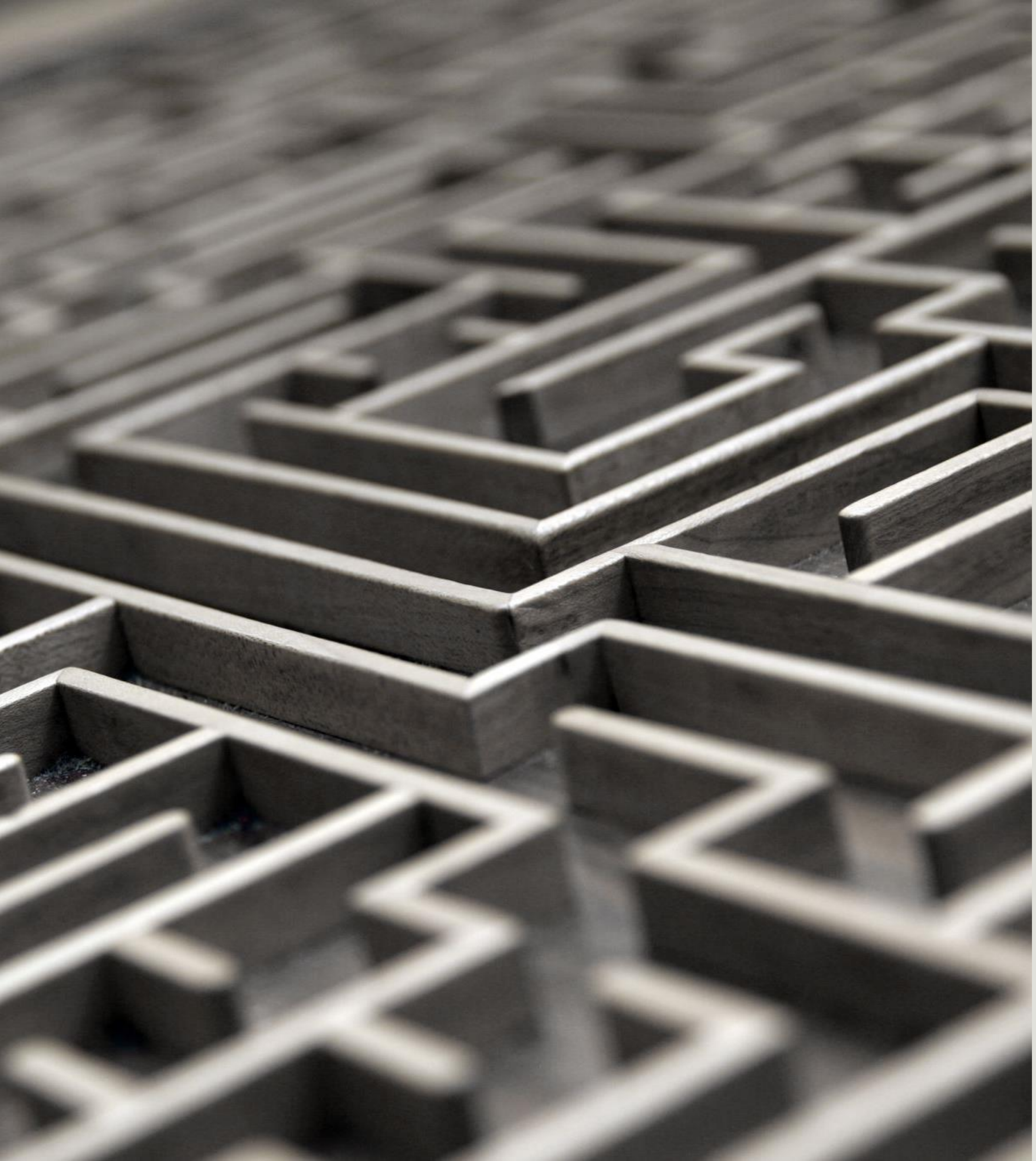


(a) Advantages
6e+4
2e+4
-2e+4
-6e+4
-1e+5
0
40k
80k
120k
160k
200k



(b) Advantages Normalized
1.2
0.8
0.4
0
-0.4
-0.8
0
40k
80k
120k
160k
200k

Small scale!



Results

PROPOSED DRIVING AGENT

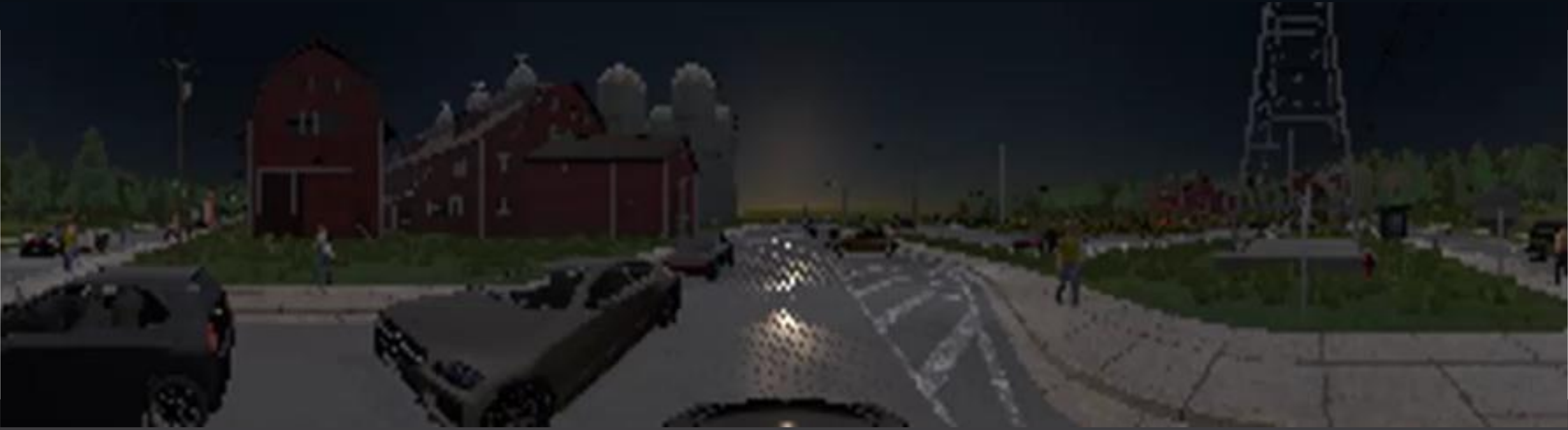
Metric/ Town		Town01	Town02	Town03	Town04	Town05	Town06	Town07	Town10	Total
Collision rate	C	0.86	**0.78**	0.88	**0.51**	0.49	0.33	**0.77**	**0.48**	64%
	S	**0.79**	0.84	**0.7**	0.63	**0.4**	**0.3**	0.78	0.57	**63%**
	U	0.99	0.99	0.98	0.99	0.98	0.92	0.88	0.89	95%
Similarity	C	**0.95**	**0.95**	**0.94**	**0.92**	**0.91**	**0.96**	**0.89**	0.85	**92%**
	S	0.94	0.93	0.9	**0.92**	0.9	**0.96**	0.86	**0.9**	91%
	U	0.84	0.8	0.8	0.82	0.72	0.7	0.76	0.72	77%
Speed	C	7.78	**8.46**	8.13	**9.05**	8.55	**9.63**	7.65	**8.76**	8.5 km/h
	S	**8.58**	8.22	**8.43**	**9.05**	**9.36**	9.33	**7.68**	8.57	**8.65 km/h**
	U	5.96	5.7	6.04	5.98	6.38	6.55	5.75	6.25	6.08 km/h
Timesteps	C	296	**335**	347	**413**	406	468	**323**	**400**	**374**
	S	**316**	331	**371**	375	**428**	**471**	282	373	368
	U	191	207	237	207	268	313	215	269	238
Total reward	C	1866	**2530**	**2157**	**2161**	1764	1951	**1813**	**1961**	**2025**
	S	**2135**	2036	1996	1780	**2190**	**2030**	1479	1906	1944
	U	503	496	589	542	484	688	357	524	523
Waypoint distance	C	**1.54**	**1.44**	**1.75**	3.75	**3.74**	5.16	**2.18**	4.3	**2.98 m**
	S	1.77	1.97	2.98	3.9	3.8	4.69	2.24	**3.3**	3.08 m
	U	2.4	2.77	3.24	**3.2**	4.66	**4.43**	3.34	4.05	3.51 m

Consistent results across towns!

Town02 daytime

Town07 evening

Town07 sunset



References

Code: https://www.github/carla-driving-rl-agent

PPO: Proximal policy optimization algorithms (J. Schulman et al. 2017)

GAE: High-dimensional continuous control using generalized advantage estimation (J. Schulman et al. 2016)

CARLA: "Carla: An open urban driving simulator" (A. Dosovitskiy et al. 2017, CoRL)

Curriculum Learning: Curriculum learning (Y. Bengio et al. 2009, ICML)

GRU: Learning phrase representations using RNN encoder–decoder for statistical machine translation (K. Cho et al. 2014, EMNLP)

ShuffleNetV2: "Shufflenet v2: Practical guidelines for efficient CNN architecture design" (N. Ma et al. 2018, ECCV)

End.

THANKS FOR THE ATTENTION