# Efficient Reinforcement Learning for Autonomous Driving with Parameterized Skills and Priors

na podstawie algorytmu ASAP-RL

Anna Ostrowska, Igor Rudolf, Norbert Frydrysiak

# ZACZNIJMY OD INTUICJI

Nie wszystko człowiek jest w stanie ogarnąć i zauważać

# WIELE REGULACJI, PRZYPADKÓW


# Czy “klasyczny” dobrze nam znany reinforcement learning rozwiązuje problem?

FAIL TO LEARN
REASONABLE PERFORMENCE

LARGE AMOUNT OF DATA

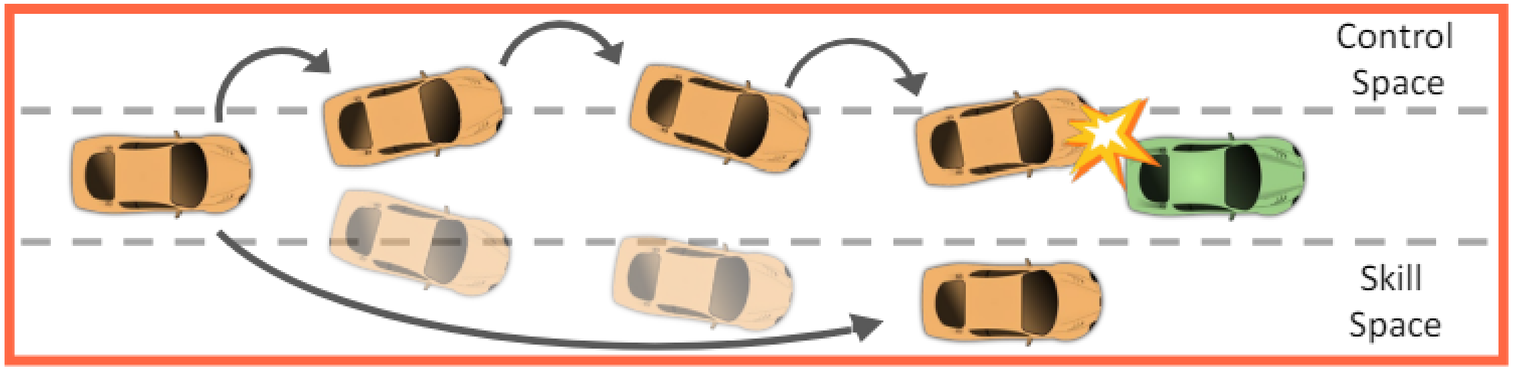
# Przystępujemy do technikaliów

NA CZYM OPIERA SIĘ OBECNIE WIELE AGENTÓW?

HASŁO TO “CONTROL SPACE”-DEFINIUJEMY PRZESTRZEŃ AKCJI

# JAKI W TYM ROZWIĄZANIU LEŻY PROBLEM?

“(...) SEQUENCE OF SINGLE-STEP CONTROL SIGNAL REARLY ACHIEVES TYPICAL DRIVING MANEUVERS ”

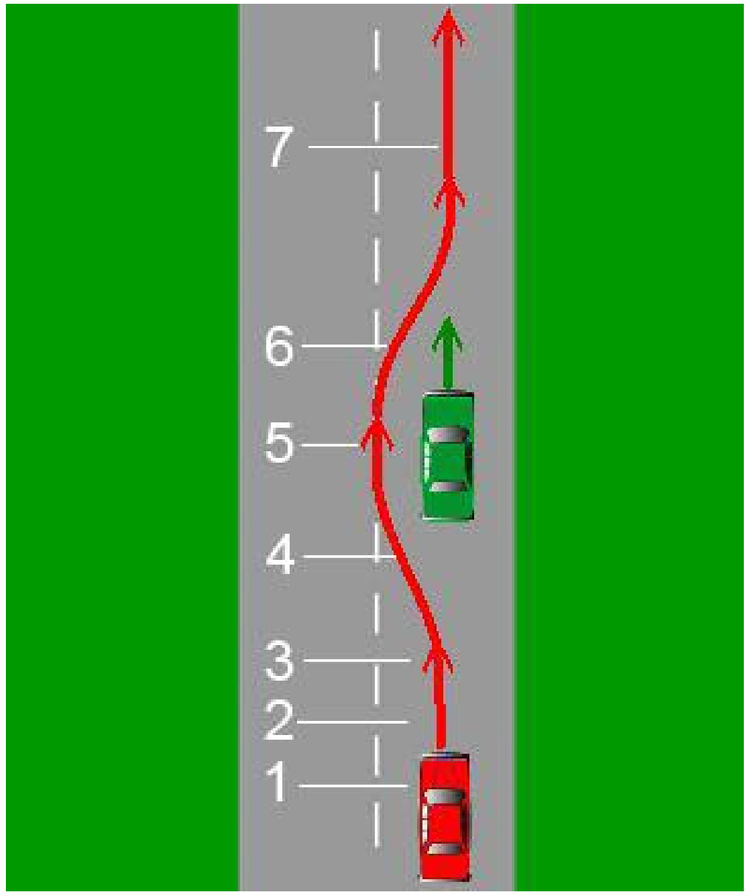

Control
Space
Skill
Space

# LUDZIE NIE UCZĄ SIĘ JAK MASZYNY

MOTION SKILLS-KONKRETNE “ZESTAWY” CZYNNOŚCI
JAKIE MOŻEMY PODJĄĆ NP: WYPRZEDZANIE

SKILL SPACE-PRZESTRZEŃ UMIEJĘTNOŚCI, ABSTRAKCYJNA
KONCEPCJA ROZWAŻAJĄCA
WSZYSTKIE MOTION SKILLS

# Z CZEGO SIĘ TAK NAPRAWDĘ SKŁADA MOTION SKILL?




1. IDEA: U NAS WYPRZEDZANIE

2.LOW-LEVEL CONTROL COMMANDS

# SPOSOBY IMPLEMENTACJI MOTION SKILLS

1.MANUALNIE(ZBYT SKOMPLIKOWANE)

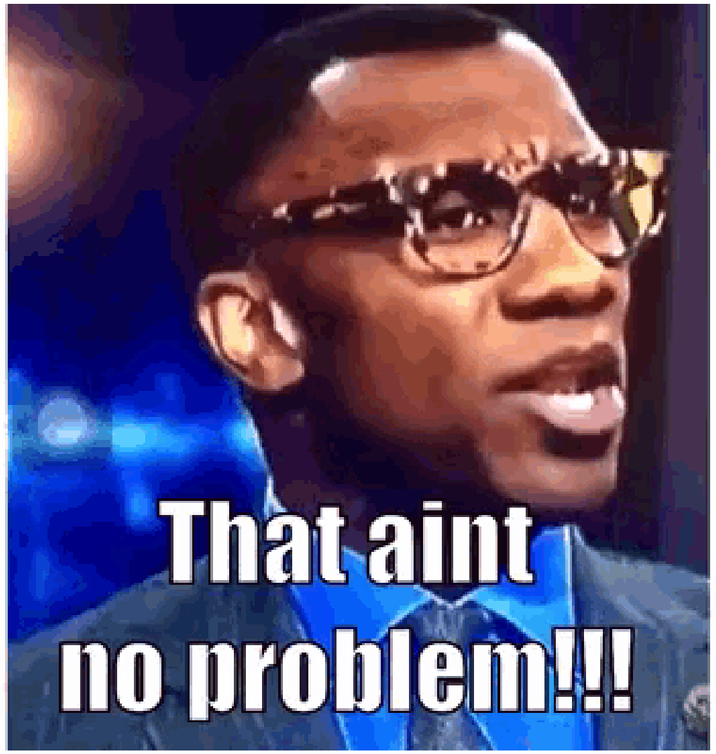
2.EXTRACTING MOTION SKILLS FROM OFFLINE MOTION DATASETS

CLUSTERING

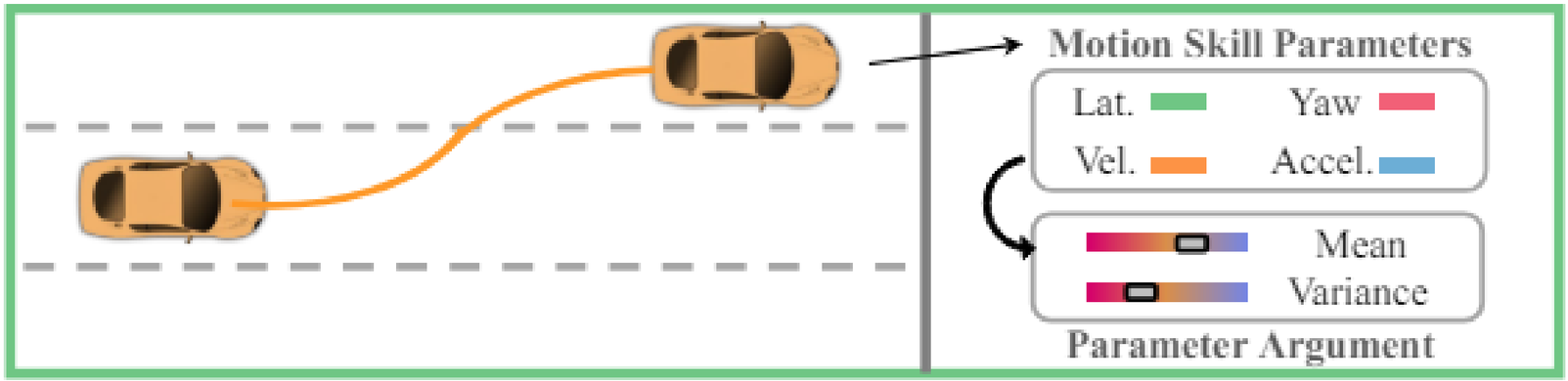
SEGMENTING MOTION TRAJECTORIES

# Jaki jest problem tego rozwiązania?

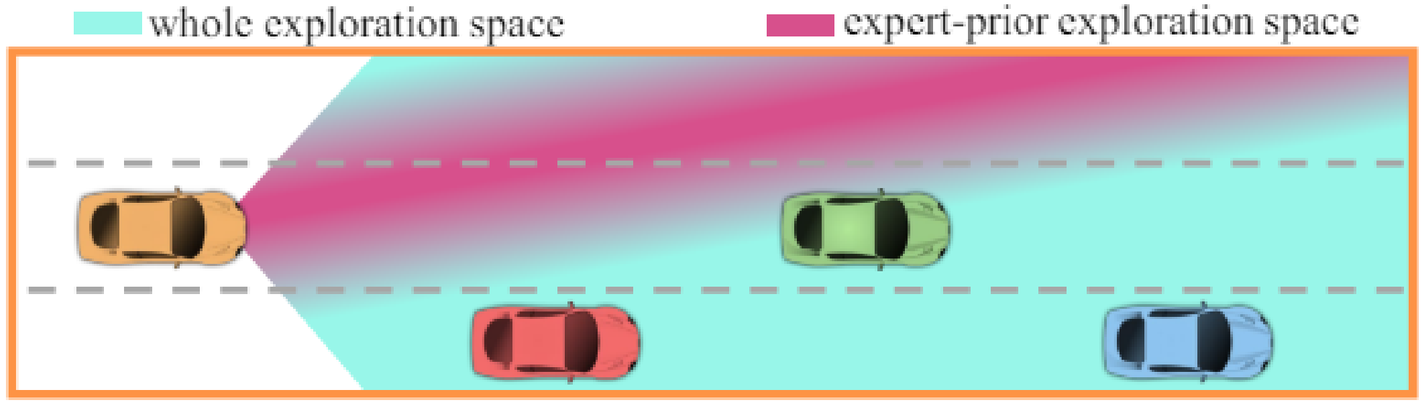
# BRAK RÓŻNORODNOŚCI DANYCH ORAZ SŁABY ROZKŁAD




VECHICLE MOTION VIEW
TO EXPLOIT MOTION SKILLS


Motion Skill Parameters
Lat.
Yaw
Vel.
Accel.
Mean
Variance
Parameter Argument

**Czy istnieje jakiś inny sposób poza wprowadzeniem “Motion Skills”, które by poprawiły uczenie się AV?**


whole exploration space
expert-prior exploration space

GŁÓWNY ZAMYSŁ:

USTALIĆ EKSPERCKĄ POLITYKĘ, KTÓRA BĘDZIE NAM MÓWILA CZY AGENT POSTĘPUJE DOBRZE CZY TEŻ ŹLE

**Problem jest taki, że czasem znamy tylko “control information”, nie znamy nagród oraz motion skills**

SQP(SEQUENTIAL QUADRATIC PROGRAMMING)- JEST TO ODWRÓCONA OPTYMALIZACJA, KTÓRA NA PODSTAWIE DANYCH EKSPERTA OKREŚLA MOTION SKILLS

POZWALA TO NA:

CONTROL SPACE ⟶ SKILL SPACE

In summary, the contributions of the proposed ASAP-RL (RL with p**A**rameterized **S**kills **a**nd **P**riors) are threefold:

- Propose an RL method to learn over the parameter of motion skills for more informative exploration and improved reward signaling. Such skills are defined in the ego vehicle motion view, which are diverse and thus generalizable to different complex driving tasks.
- Propose an inverse skill parameter recovery method to convert expert demonstration from control space to skill space, and a simple but effective double initialization method to better leverage expert prior without issues of performance drop or suppression. Thus we can take advantage of both skills and priors simultaneously.
- Validate our method for autonomous driving tasks in three challenging dense-traffic scenarios and demonstrate our method outperforms previous methods that consider skills and priors differently.

# EXPERT PRIORS USAGE

1.WARM UP PRETRAINING

2.TRAINING AN EXPERT POLICY AND AND GUIDE RL PROCESS

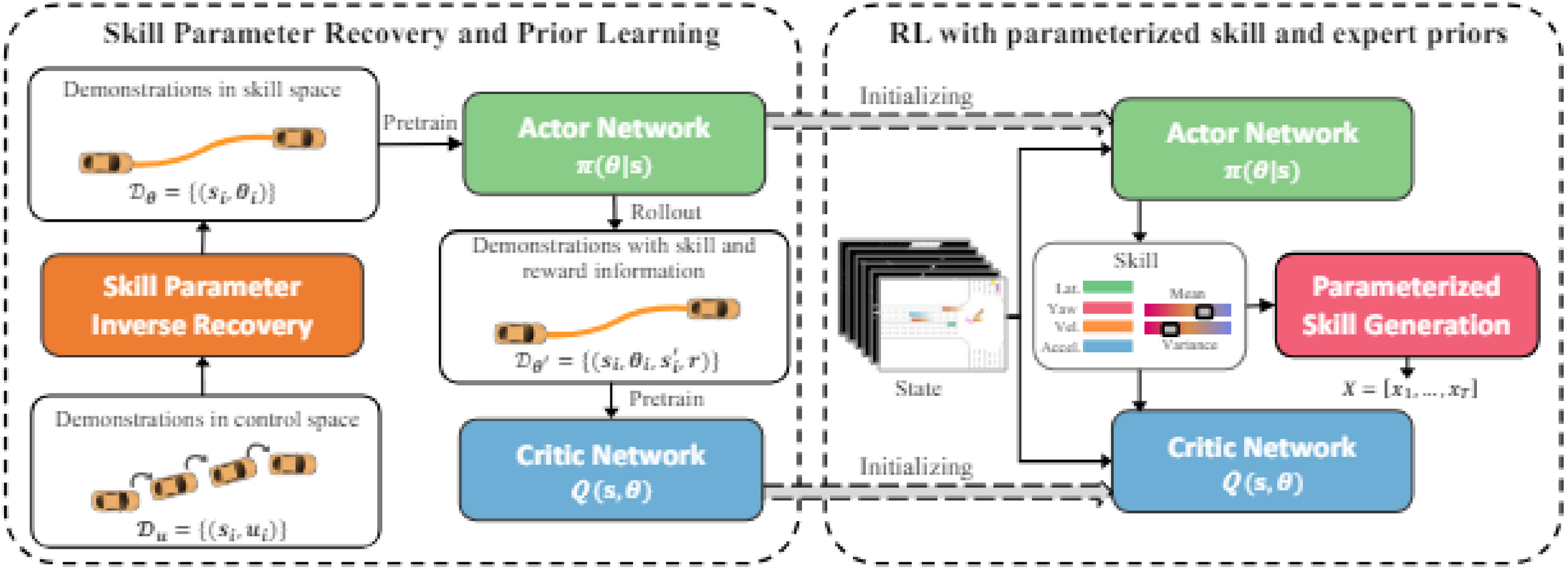
3.EXPERT DATA BUFFER MIXED WITH INTERACTIONS

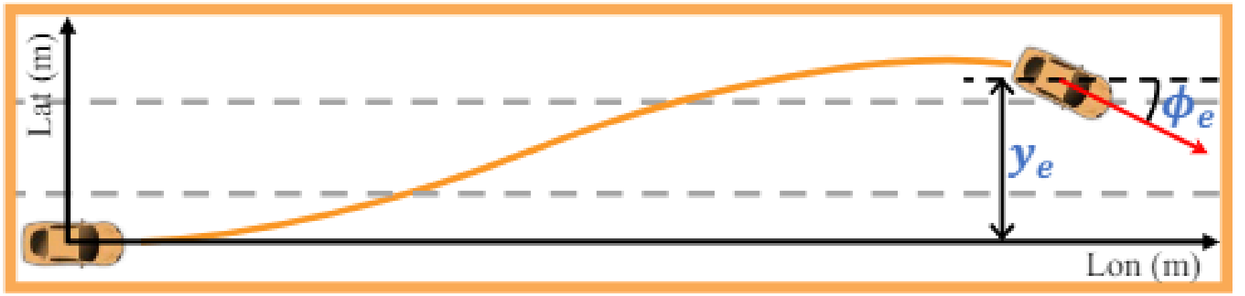
# MOTION SKILLS GENERATION

1.PATH GENERATION

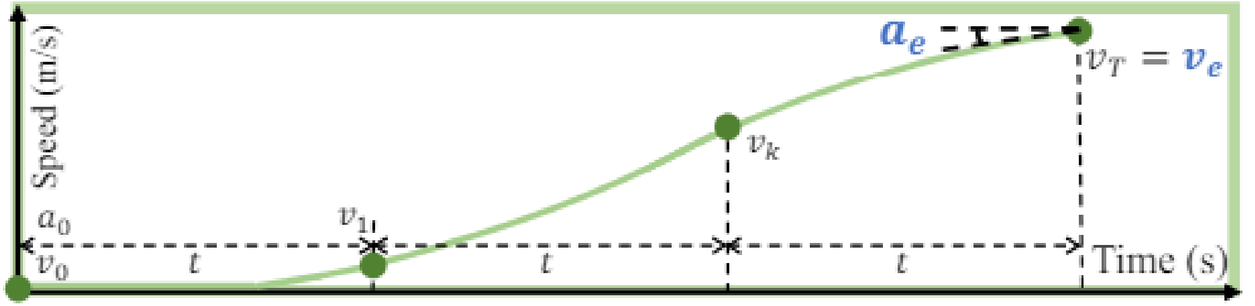
2.SPEED PROFILE GENERATION

3.PARAMETRIZED MOTION SKILL GENERATION

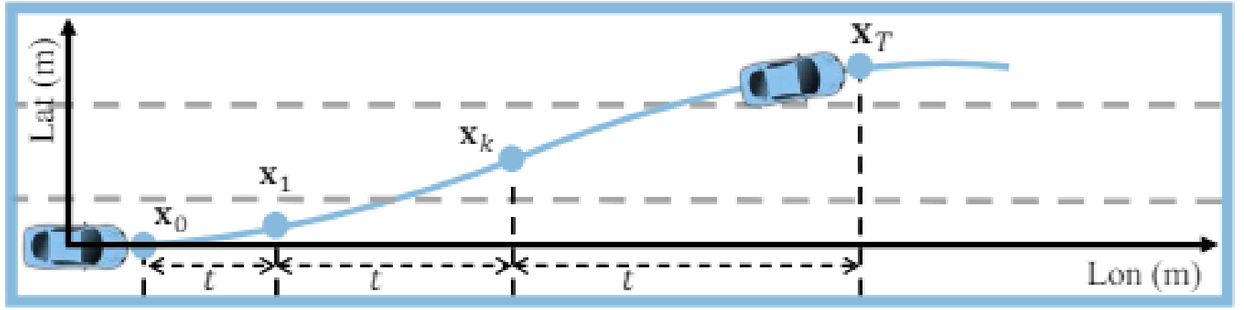

Skill Parameter Recovery and Prior Learning
RL with parameterized skill and expert priors
Demonstrations in skill space
$\mathcal{D}_\theta = \{(s_i, \theta_i)\}$
Pretrain
Actor Network
$\pi(\theta|s)$
Rollout
Demonstrations with skill and reward information
$\mathcal{D}_{\theta'} = \{(s_i, \theta_i, s_i', r)\}$
Pretrain
Critic Network
$Q(s, \theta)$
Skill Parameter Inverse Recovery
Demonstrations in control space
$\mathcal{D}_u = \{(s_i, u_i)\}$
Initializing
Actor Network
$\pi(\theta|s)$
State
Skill
Lat.
Yaw
Vel.
Accel.
Mean
Variance
Parameterized Skill Generation
$X = [x_1, \ldots, x_T]$
Initializing
Critic Network
$Q(s, \theta)$




(a) Path Generation




(b) Speed Profile Generation




(c) Trajectory (Motion Skill) Generation

# Wykorzystanie wiedzy a priori z działań eksperta

Expert

$$\mathcal{D}_u = \{(s_i, u_i)\}$$

Control Space

no skill and reward information

inverse parameter recovery

$$\mathcal{D}_{\boldsymbol{\theta}} = \{(s_i, \boldsymbol{\theta}_i)\}.$$

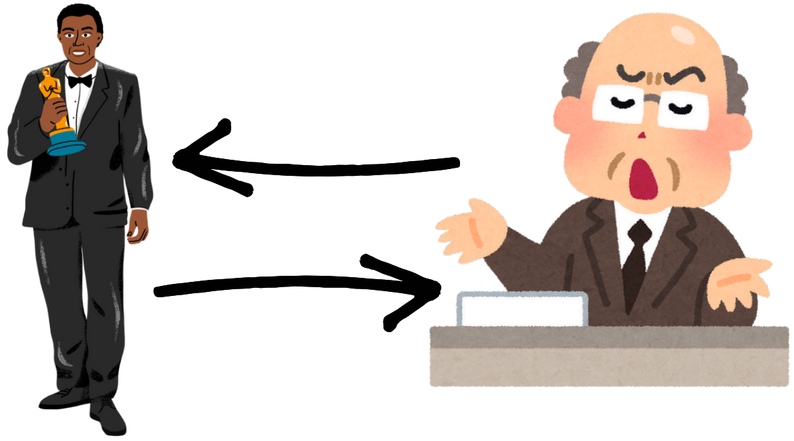
skill space

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} ||\mathbf{X}_d - \mathbf{X}||_2$$

$$\text{s.t.} \quad \mathbf{X} = f_s(\boldsymbol{\theta})$$

# Expert Prior Learning - Actor and Critic Pretraining

$$\mathcal{D}_{\boldsymbol{\theta}} = \{(s_i, \boldsymbol{\theta}_i)\}.$$



$$\mathbb{E}_{(s,\boldsymbol{\theta})\sim\mathcal{D}_{\boldsymbol{\theta}}}\Big[\log\pi(\boldsymbol{\theta}|s) + \beta\mathcal{H}(\boldsymbol{\theta})\Big]$$

wstępnie nauczony agent zbiera skill and reward information

$$\mathcal{D}_{\boldsymbol{\theta}}^{'} = \{(s_i, \boldsymbol{\theta}_i, s_i', r)\}$$

wstępne uczenie krytyka

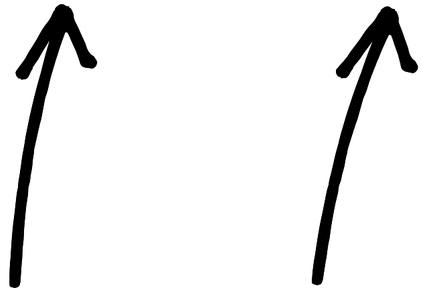
**Algorithm 1** ASAP-RL

**Input:** Raw demonstrations $\mathcal{D}_u$, discount $\gamma$, target entropy $\delta$, learning rates $\lambda_\pi, \lambda_Q, \lambda_\alpha$, target update rate $m$, temperature hyperparameter $\alpha$, motion skill model $f_s$
**Require:** actor $\pi_\varphi(\boldsymbol{\theta}_t|s_t)$, critic $Q_\phi(s_t, \boldsymbol{\theta}_t)$, target network $Q_{\bar{\phi}}(s_t, \boldsymbol{\theta}_t)$, replay buffer $\mathcal{D}$, demonstration in skill space $\mathcal{D}_{\boldsymbol{\theta}}$, demonstration with skill and reward information $\mathcal{D}'_{\boldsymbol{\theta}}$.
Skill Parameter Recovery
**for** each trajectory in $\mathcal{D}_u$ **do**
  **for** every $\mathbf{X}_d$ split from the trajectory **do**
    $\boldsymbol{\theta} = \arg\min ||\mathbf{X}_d - f_s(\boldsymbol{\theta})||_p$ (Eq 1)
    $\mathcal{D}_{\boldsymbol{\theta}} \leftarrow \mathcal{D}_{\boldsymbol{\theta}} \cup \{(s, \boldsymbol{\theta})\}$
  **end for**
**end for**
Prior Learning
**for** each iteration **do** % Actor Pretraining
  Sample $(s, \boldsymbol{\theta})$ from $\mathcal{D}_{\boldsymbol{\theta}}$
  Update $\pi_\varphi$ according to Eq 2
**end for**
**for** each iteration **do** % Roll-out $\pi_\varphi$ to Collect $\mathcal{D}'_{\boldsymbol{\theta}}$
  **for** every $T$ environment step **do**
    rollout pretrained actor to collect $\{s_t, \boldsymbol{\theta}_t, \tilde{r}, s_{t'}\}$
    $\mathcal{D}'_{\boldsymbol{\theta}} \leftarrow \mathcal{D}'_{\boldsymbol{\theta}} \cup \{s_t, \boldsymbol{\theta}_t, r, s_{t'}\}$
  **end for**
**end for**
**for** each iteration **do** % Critic Pretraining
  Sample $(s, \boldsymbol{\theta}, r, s')$ from $\mathcal{D}'_{\boldsymbol{\theta}}$
  Update $\pi_\varphi$ according to Line 38
**end for**

RL with Parameterized Skills and Priors
Initialize actor and critic with the pretrained weight
**for** each iteration **do**
  **for** every $T$ environment step **do**
    $\boldsymbol{\theta}_t \sim \pi_\varphi(\boldsymbol{\theta}_t|s_t)$ % sample skill parameter
    $\mathbf{X}_t = \{\mathbf{x_i}\}_{i=1}^{T} \sim f_s(\mathbf{X}_t|\boldsymbol{\theta}_t)$ % generate skill
    $s_{t'} \sim p(s_{t+T}, r_{t:t+T}|s_t, \mathbf{X}_t)$ % execute skill
    $\tilde{r}_t(s_t, \boldsymbol{\theta}_t) = \sum_{i=0}^{T-1} r_{t+i}$ % reward calculation
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, \boldsymbol{\theta}_t, \tilde{r}, s_{t'}\}$ % replay buffer
  **end for**
  **for** every gradient step **do** % typical SAC training
    $\bar{Q} = \tilde{r}(s_t, \boldsymbol{\theta}_t) + \gamma\big[Q_{\bar{\phi}}(s_{t'}, \pi_\varphi(\boldsymbol{\theta}_{t'}|s_{t'})) - \alpha\mathcal{H}\big(\pi_\varphi(\boldsymbol{\theta}_{t'}|s_{t'})\big)\big]$
    $\varphi \leftarrow \varphi + \lambda_\pi \nabla_\varphi\big[Q_\varphi(s_t, \pi_\varphi(\boldsymbol{\theta}_t|s_t)) + \alpha\mathcal{H}\big(\pi_\varphi(\boldsymbol{\theta}_t|s_t)\big)\big]$
    $\phi \leftarrow \phi - \lambda_Q \nabla_\phi\big[\frac{1}{2}\big(Q_\phi(s_t, \boldsymbol{\theta}_t) - \bar{Q}\big)^2\big]$
    $\alpha \leftarrow \alpha - \lambda_\alpha \nabla_\alpha\big[\alpha \cdot ((\mathcal{H}\big(p_{\boldsymbol{\theta}}(\boldsymbol{\theta}_t|s_t) - \delta\big)\big]$
    $\bar{\phi} \leftarrow m\phi + (1-m)\bar{\phi}$
  **end for**
**end for**

# Modified maximum-entropy RL

$$J = \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^t r_t + \alpha \mathcal{H}\big(\pi(\boldsymbol{\theta}|\boldsymbol{s})\big)\right]$$

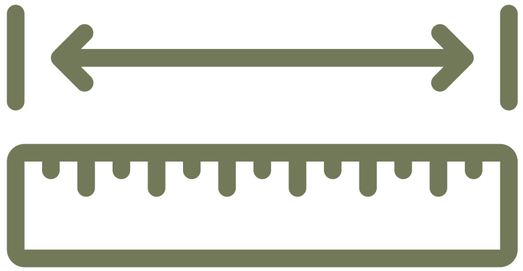


To chcemy maksymalizować

# EXPERIMENTS

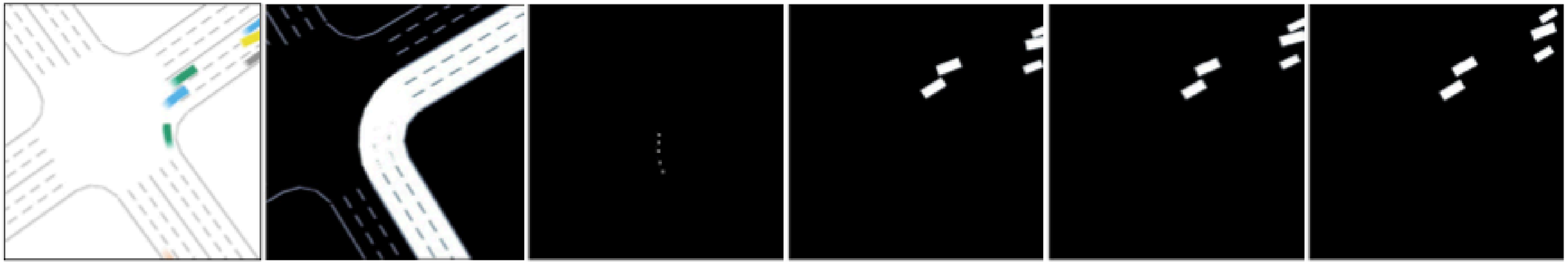
PERFORMANCE

INFLUENCE OF THE LENGTH OF SKILL



INFLUENCE OF EXPERT PRIOR

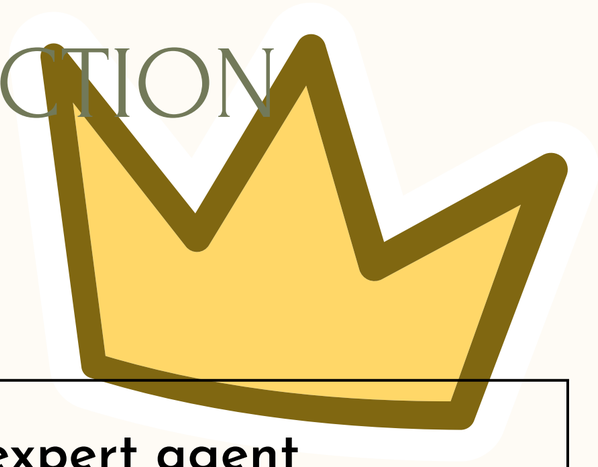
# Environment



$$r_t = R_{progress} + R_{destination} + R_{crash} + R_{overtaking}. \quad (4)$$

- $R_{progress}$: The agent gets a sparse reward of 1 for every 10 m distance completed.
- $R_{destination}$: The agent gets a reward of 1 if it reaches the destination.
- $R_{crash}$: If the agent collides with other vehicles or the road curbs, it gets a negative reward of -5.
- $R_{overtaking}$: If the agent passes one vehicle, it gets a reward of 0.1.

# EXPERT DEMONSTRATION COLLECTION



| hand-designed rule-based expert planner | trained RL expert agent |
| --- | --- |
| • higher performance than the RL expert agent after time-consuming manual tuning<br>• require more designs in data collection to collect more diverse and react-to-danger actions | • can collect high-quality and more diverse demonstrations, and agents trained from such demonstrations achieved stronger performance |

# BASELINES

| | |
|---|---|
| **PPO** (Proximal Policy Optimizatiom) | • jednoetapowy algorytm on-policy |
| **SAC** (Soft Actor-Critic) | • jednoetapowy algorytm off-policy, zachowuje pewien poziom losowości w działaniach agenta, co pozwala na eksploracje |
| **Constant SAC** | • modyfikacja algorytmu SAC, w której akcja agenta jest powtarzana wielokrotnie |
| **SPiRL** (Skill-Prior Reinforcement Learning) | • algorytm uczenia ze wzmocnieniem, który wykorzystuje umiejętności (skills) i wiedzę wstępną eksperta do poprawy wydajności i stabilności uczenia się agenta |
| **TaEcRL** (Task-agnostic and Egocentric Reinforcement Learning) | • koncentruje się na nabywaniu uniwersalnych i egocentrycznych umiejętności ruchowych przez agenta, nie korzysta z wiedzy eksperta |

# EVALUATION METRIC

## Episode Reward

- the sum of all the rewards in an episode

## Success Rate

- the percentage of episodes where the agent reaches the destination on time without collisions

## Road Completion Ratio

- the ratio of road length completed by the agent to the total road length per episode

## Collision Rate

- the percentage of episodes in which a collision occurs

## Passed Cars Per Episode

- the number of cars overtaken by the agent in each episode

# A THREE-STAGE STRATEGY FOR PERFORMANCE EVALUATION

## Episode Reward

- the sum of all the rewards in an episode

1

## Success Rate

- the percentage of episodes where the agent reaches the destination on time without collisions

## Road Completion Ratio

- the ratio of road length completed by the agent to the total road length per episode

2

## Collision Rate

- the percentage of episodes in which a collision occurs

## Passed Cars Per Episode

- the number of cars overtaken by the agent in each episode

3

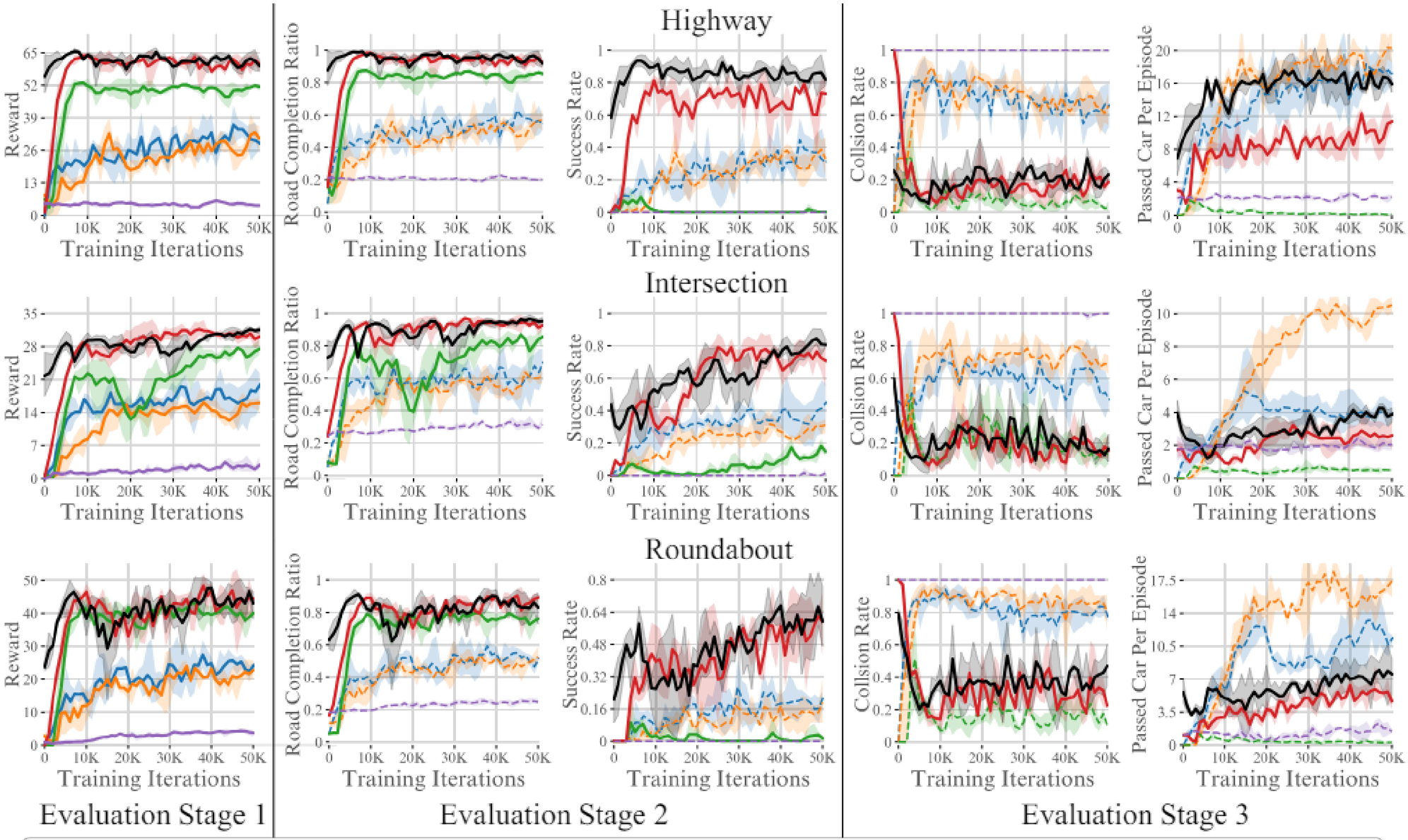
# SPOSTRZEŻENIA

- SPiRL okazał się skuteczny w zadaniach manipulacyjnych, ale słabo radzi sobie w naszym środowisku związanym z jazdą

- SPiRL uczy się na podstawie ograniczonych demonstracji eksperta - agenci szkoleni za pomocą SPiRL głównie poruszają się prosto i rzadko wykonują manewry, co prowadzi do wysokiej liczby kolizji

- Constant-SAC: agent jeździ bardzo spokojnie i powoli, aby unikać kolizji: niskie Collision Rate i przyzwoite Road Completion Ratio, agent zwykle nie jest w stanie dotrzeć do celu w określonym czasie

# EXPERIMENTS



# ASAP-RL vs TaEcRL

- ASAP-RL achieves better performance in the first 10k iterations in terms of all metrics due to the usage of the expert prior
- ASAP-RL has more passed cars per episode than TaEcRL by a large margin, with better or similar success rates, road completion ratio, and collision rates
- TaEcRL is limited by its own algorithm design and cannot utilize expert data

# ABLATION STUDY

- Influence of the length of skill (T=10)
- Influence of expert prior

- No Prior
- Behavior Cloning (BC)
- Initiate Actor
- KL + Initiate Actor

# ABLATION STUDY

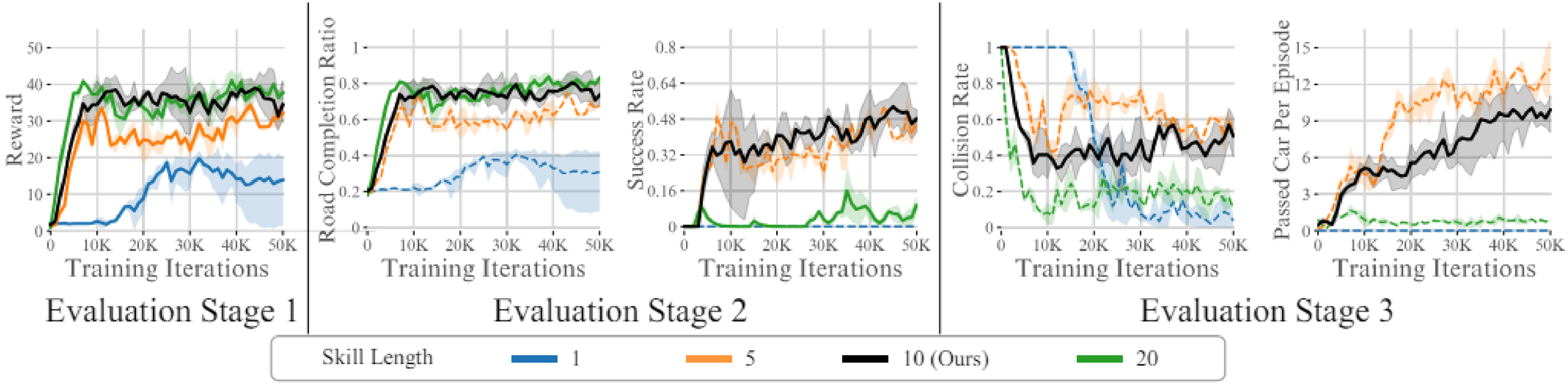
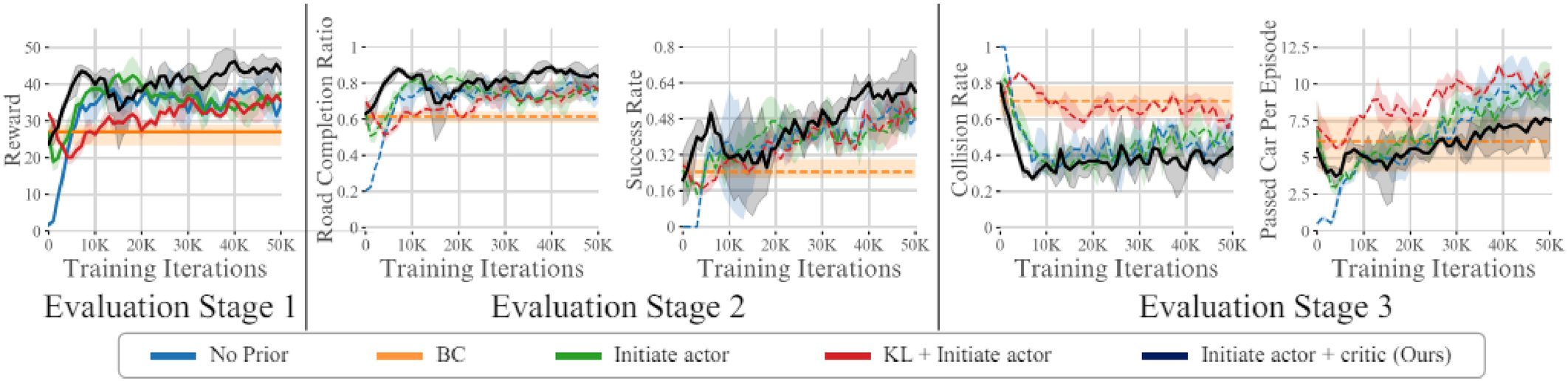



Fig. 6: Ablation analysis of skill length in the roundabout scenario. When $T$ increases from 1 to 10, performance improvement is observed, benefiting from temporal abstraction. But when $T$ reaches 20, it is too long for the agent to react to accidents during the skill execution due to delayed replanning. We observe that a skill length of 10 reached a good trade-off.



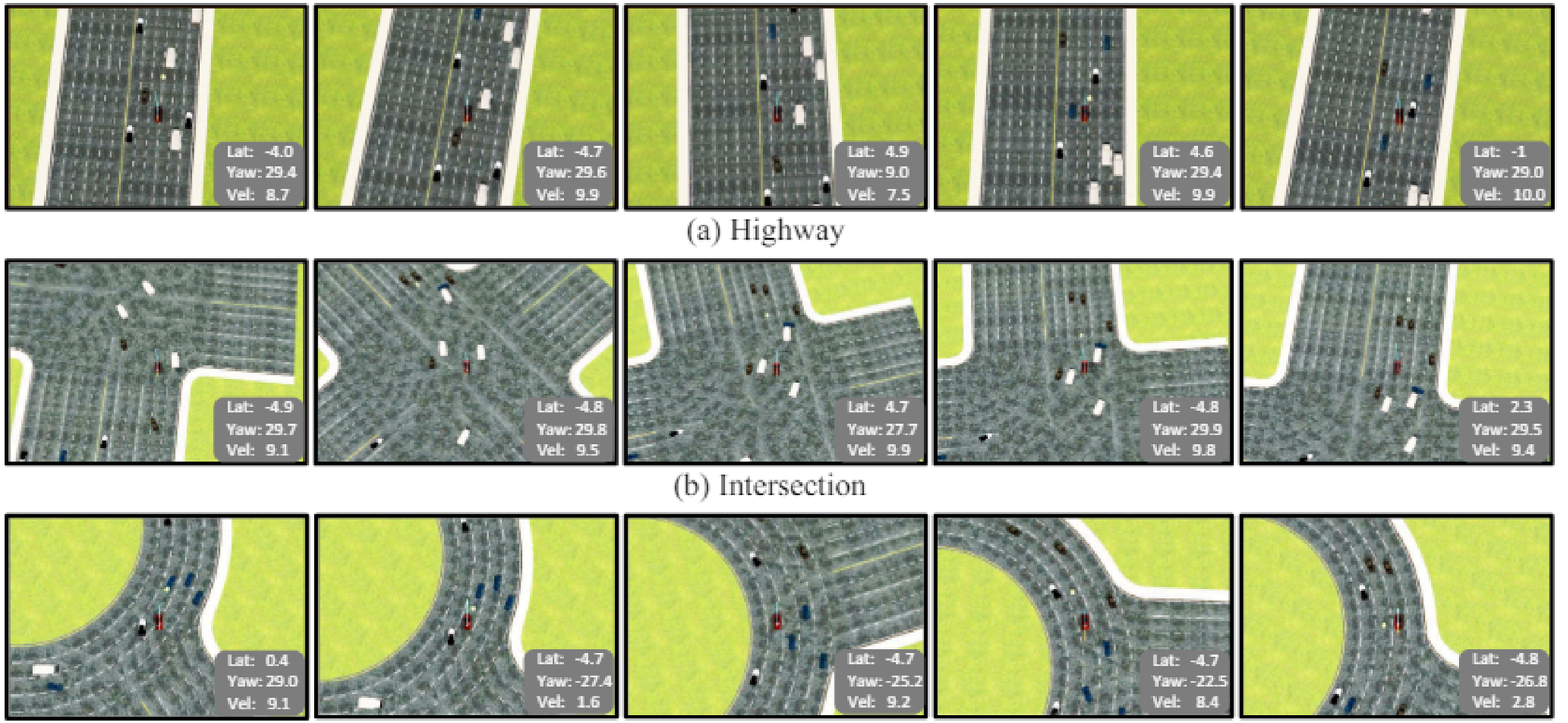
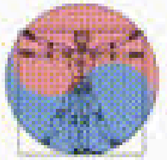



Fig. 8: Visualizations of the trained agents in three dense-traffic scenarios. The dark red vehicle denotes the ego agent and the blue line denotes the motion skill trajectory output by the agent. Generated parameters corresponding to the skill are shown in the bottom right. (a) In the highway scenario, the agent performs consecutive lane changes to overtake vehicles ahead. (b) In the intersection scenario, the agent turns left and overtakes vehicles ahead with a high velocity. (c) In the roundabout scenario, the agent slows down to cautiously overtake the vehicle on the left (the second image), and drives slowly when vehicles ahead block the way (the last image).



ROBOTICS
SCIENCE AND SYSTEMS

# Efficient Reinforcement Learning for Autonomous Driving with Parameterized Skills and Priors

Letian Wang, Jie Liu, Hao Shao, Wenshuo Wang, Ruobing Chen, Yu Liu, Steven L. Waslander

UNIVERSITY OF TORONTO



McGill UNIVERSITY

# Źródła

- https://arxiv.org/pdf/2305.04412.pdf
- https://github.com/Letian-Wang/asaprl