01 Vehicle Rescheduling Problem

The vehicle rescheduling problem (VRSP) arises when a previously assigned trip is disrupted. A traffic accident, a medical emergency, or a breakdown of a vehicle are examples of possible disruptions that demand the rescheduling of vehicle trips.

FLATLAND

Amos Dinh, Matthias Fast,
Henrik Rathai, Jannik Völker

02 Previous Challenges

2019 Challenge

CHF 7’500.- for first prize
CHF 5’000.- for second prize
CHF 2’500.- for third prize

NEURAL INFORMATION PROCESSING SYSTEMS

AMLD

icaps

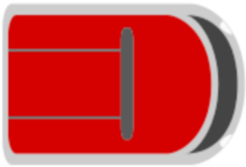
FLATLAND

Multi-Agent Reinforcement Learning on Trains | AMLD 2021 Competition NeurIPS 2020 Competition

By SNCF SNCF & SBB & DB Deutsche Bahn

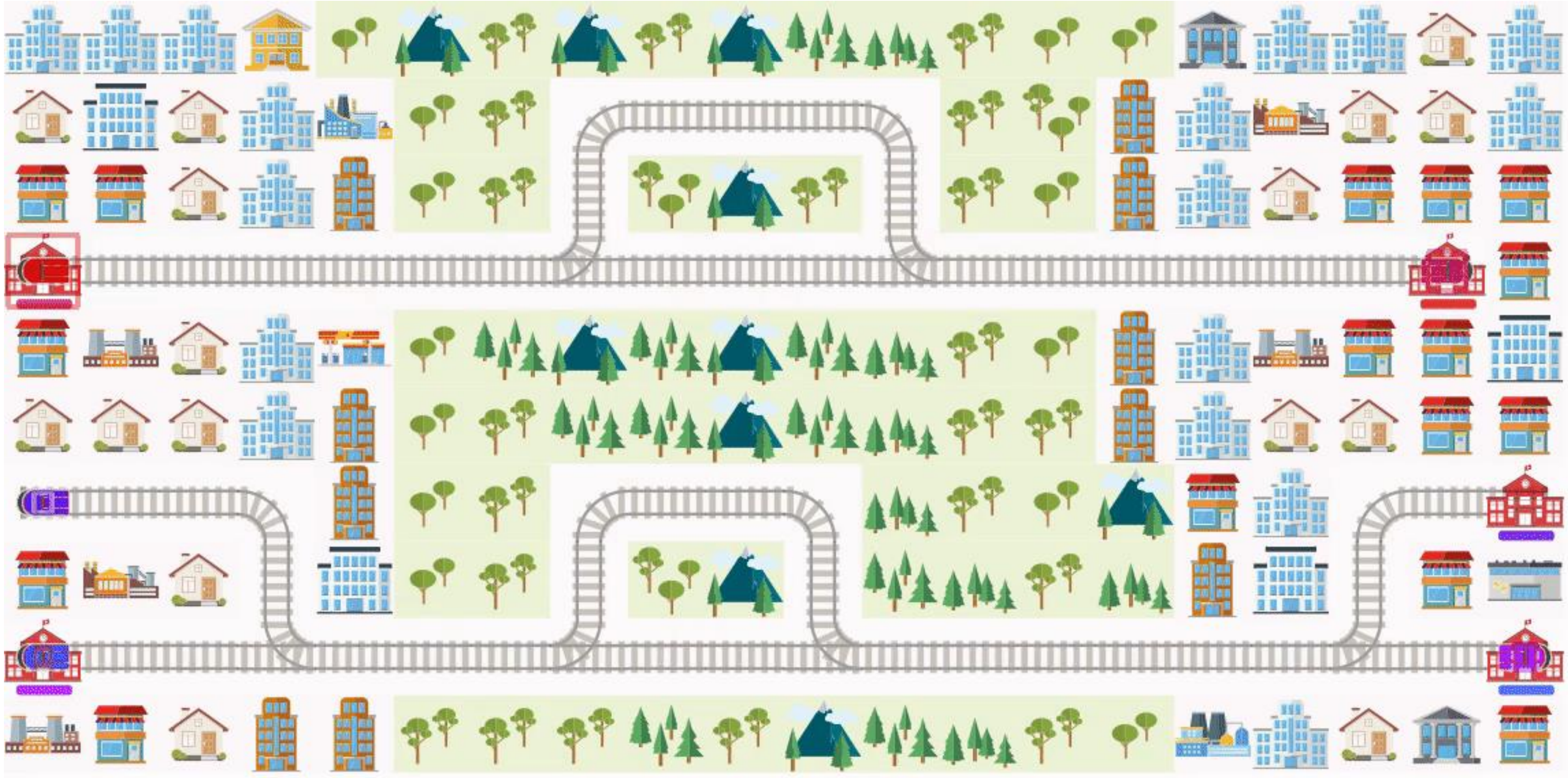
02 Environment

Agent

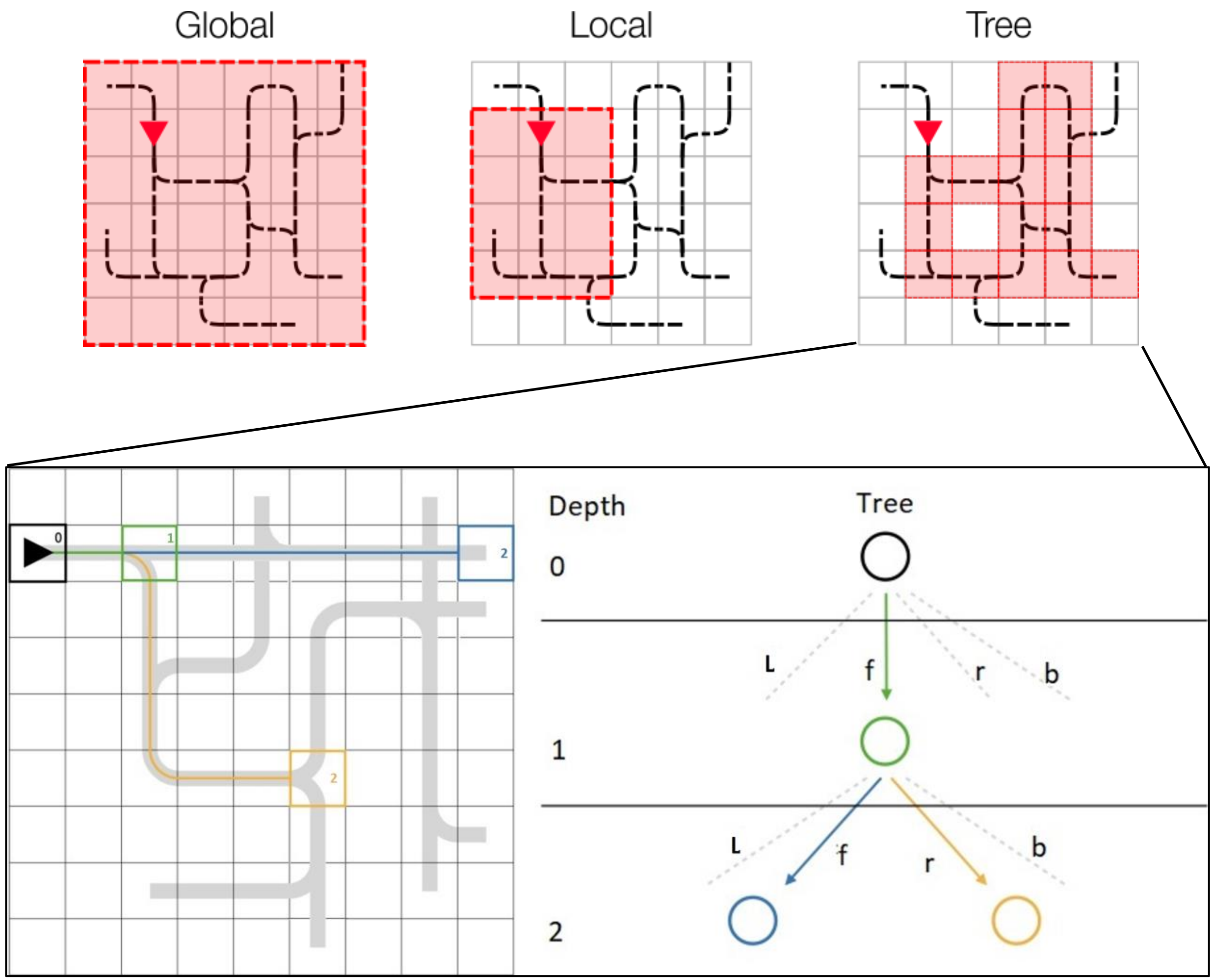


Target

02 Environment



02 Environment – Observations



02 Evaluation Parameter & Rewards

Env	# agents	W	H	# cities	Malfunction rate
0	7	30	30	2	0 %
1	10	30	30	2	1 %
2	20	30	30	3	0,5 %
3	50	30	35	3	0,5 %
speed = 1; max_rails_between_cities = 2; max_rails_in_city = 2; n_envs_run = 10					

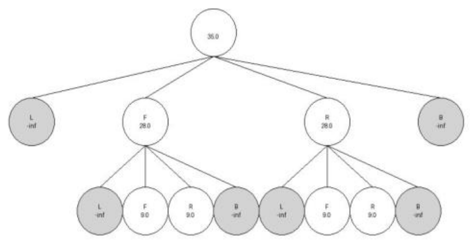
$$\text{NormalizedReward} = \frac{\text{cumulativeReward}}{\text{self.env.maxEpisodeSteps} * \text{self.env.getNumAgents()}}$$

03 Ziel und Vorgehen

- Lernen eines Reinforcement Learning Agents der die Züge zu ihrem Ziel bringt
- Der Agent muss lernen, Deadlocks der Züge zu vermeiden:
  - Frontal Kollisionen
  - Defekte
- Wir benutzen einen einzelnen Agenten, der per Zeitschritt nacheinander für jeden Zug die nächste Aktion auswählt.
- Multi-Agenten System oder einzelner Agent, der mehrere Entscheidungen per Zeitschritt trifft
- Anwenden von Temporal Difference Learning Methoden
- Modellieren von Q(S, A) als Neurales Netz

03 Lernen des Agenten: Modellierungsdimensionen

- **Aktionen**: {L, F, R, Stop, Nothing} (oder: Wahl zwischen N-kürzesten Wegen...)
- **Observation**: Baumbasiert per Zug, Tiefe N (oder: Graphbasiert, Gridbasiert)
    - Dist_to_target, other_train_passes_at_step_t, malfunction



- **other_train_passes_at_step_t**: Predictor ist naiver Shortest Path
- **TD Methoden**: (n-step, expected) SARSA, (double, dueling) Q-Learning
- **Reward**: Summe der Rewards der Agenten per Zeitschritt
    - Negativer Reward: for not moving at all, per time step
    - Positiver Reward: Erreichen des Ziels

03 Lernen des Agenten: Modellierungsdimensionen

- Training:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right]$$ [1]

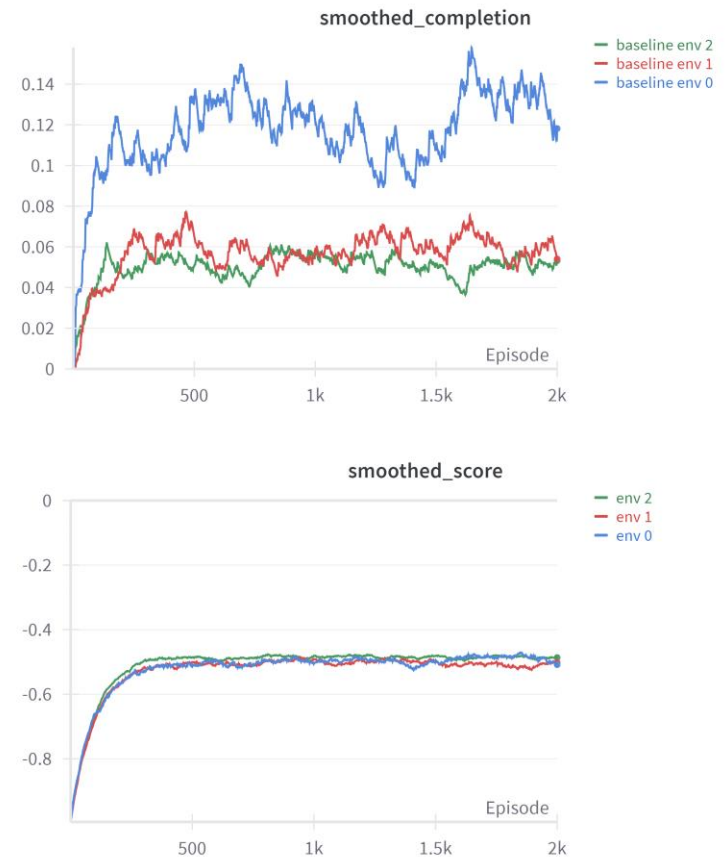
$$L = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right)^2\right]$$

  - Replay Buffer
  - Epsilon-greedy
  - Training zufällig gesampelten Mini-Batches
  - 2000 Episoden

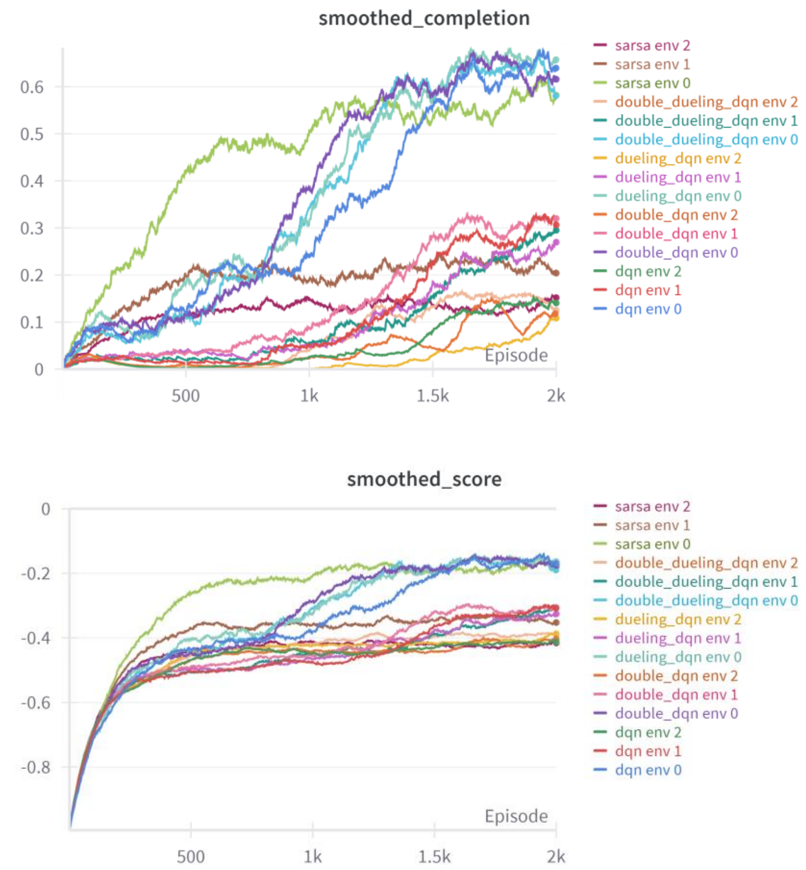
[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *nature* 518.7540 (2015): 529-533.

04 Baseline & Algorithms

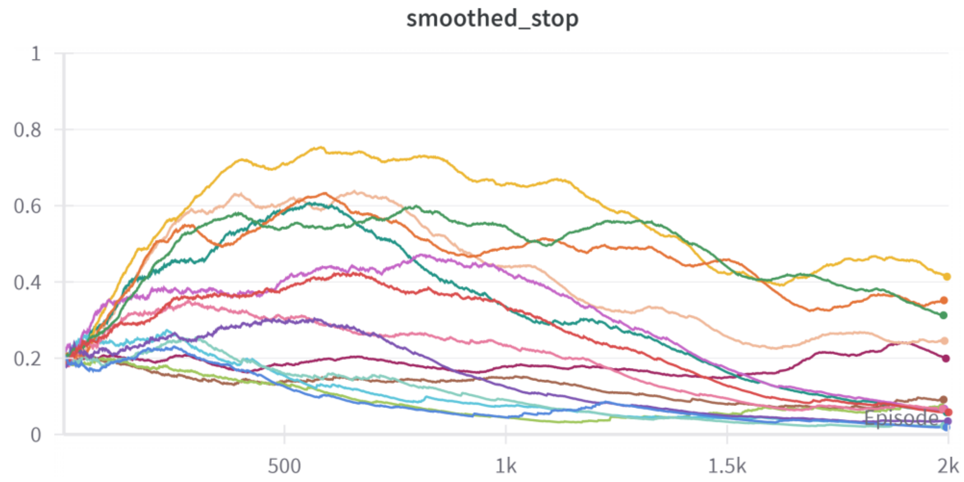
Baseline

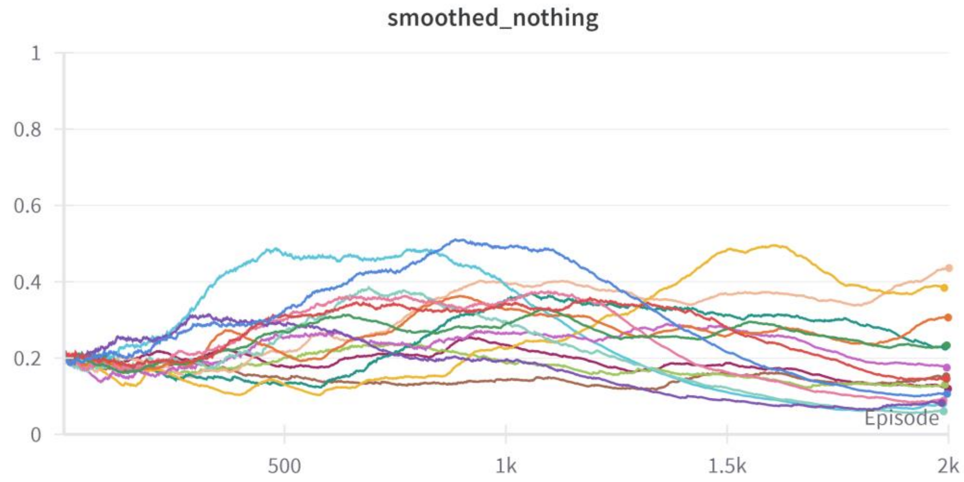


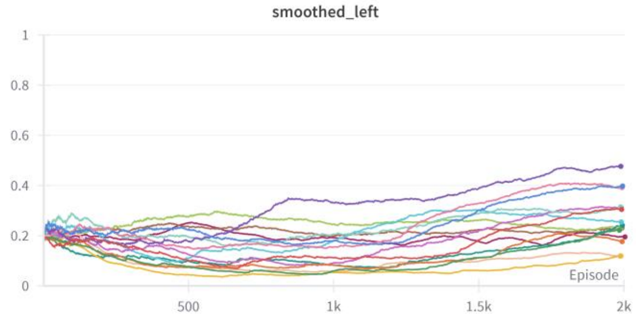
Comparison of RL Algorithms

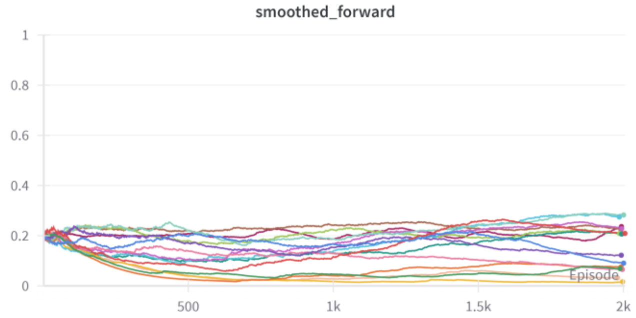


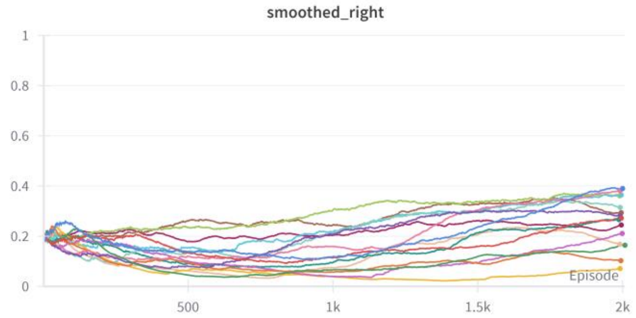
04 Actions


smoothed_stop
1
0.8
0.6
0.4
0.2
0
500
1k
1.5k
2k
Episode


smoothed_nothing
1
0.8
0.6
0.4
0.2
0
500
1k
1.5k
2k
Episode

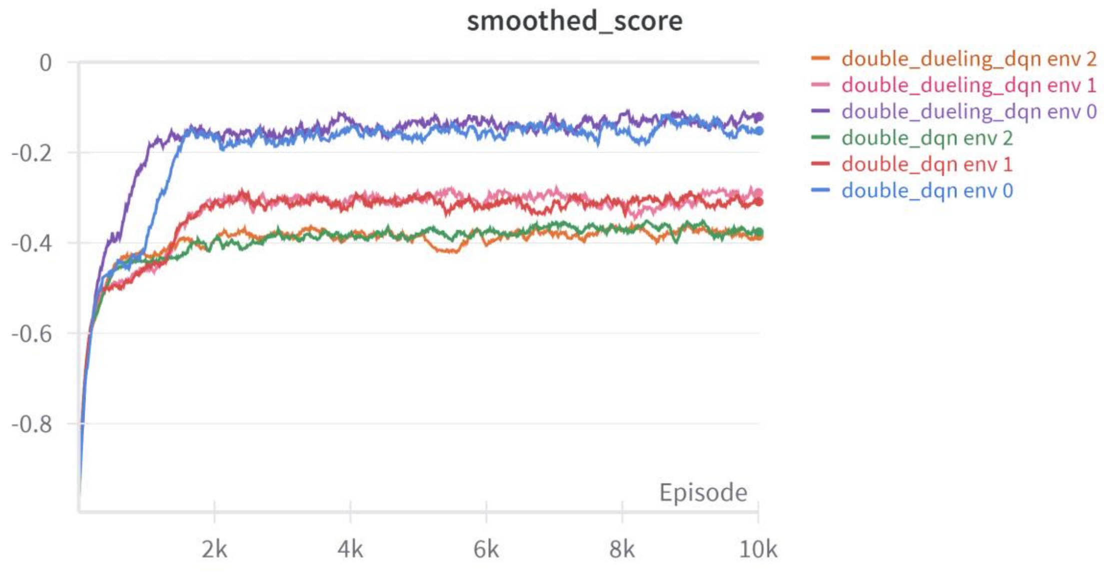

smoothed_left
Episode


smoothed_forward
Episode

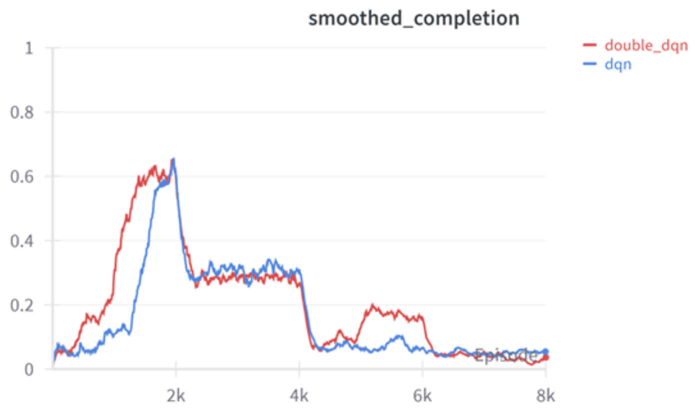

smoothed_right
Episode

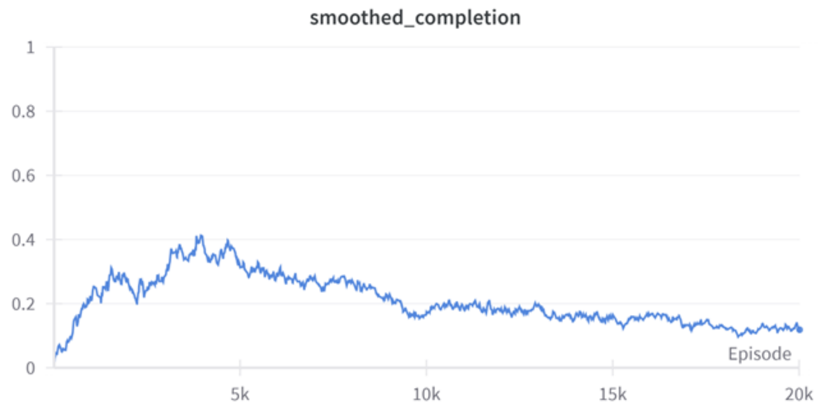
04 Experiments

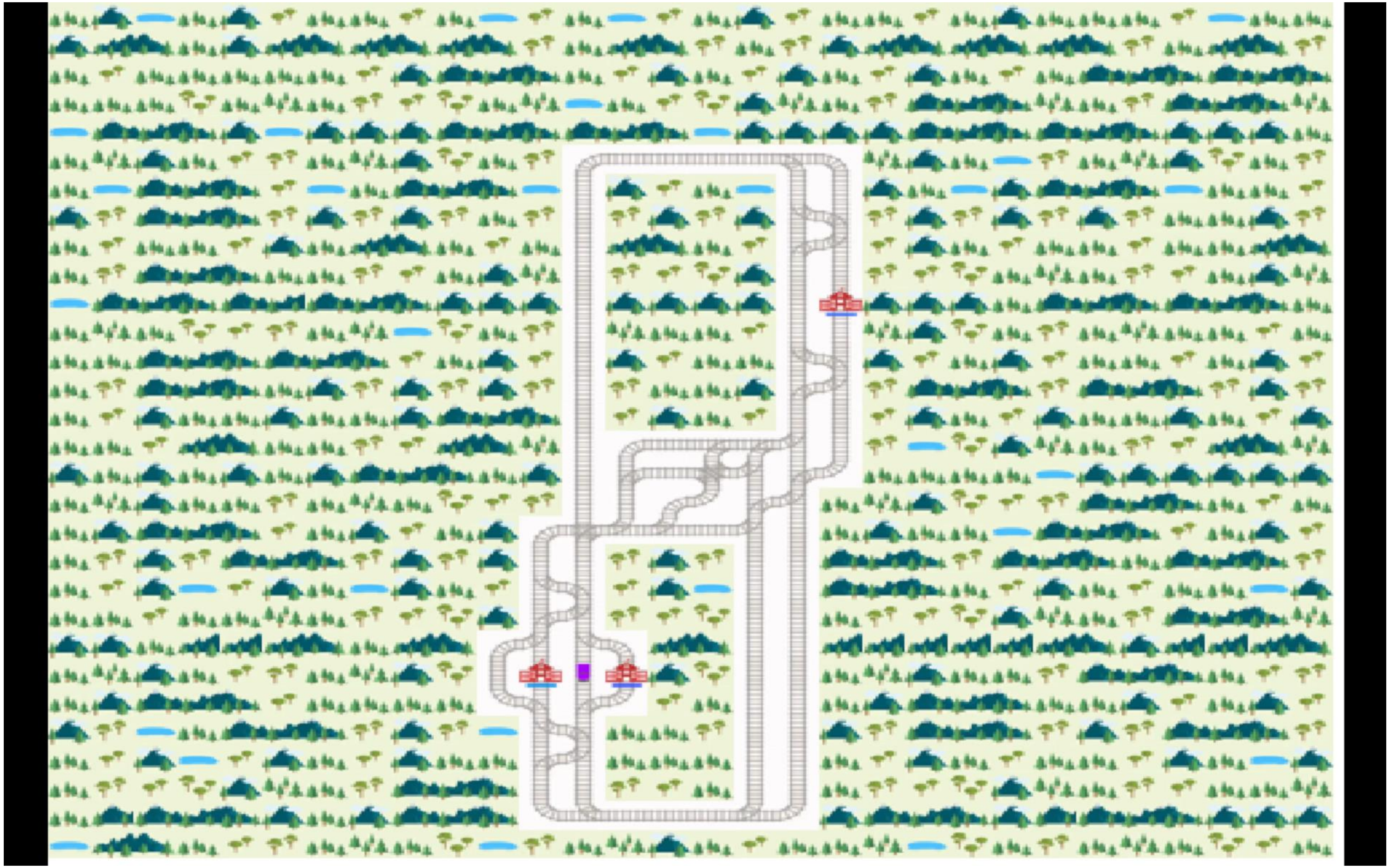
Training Length

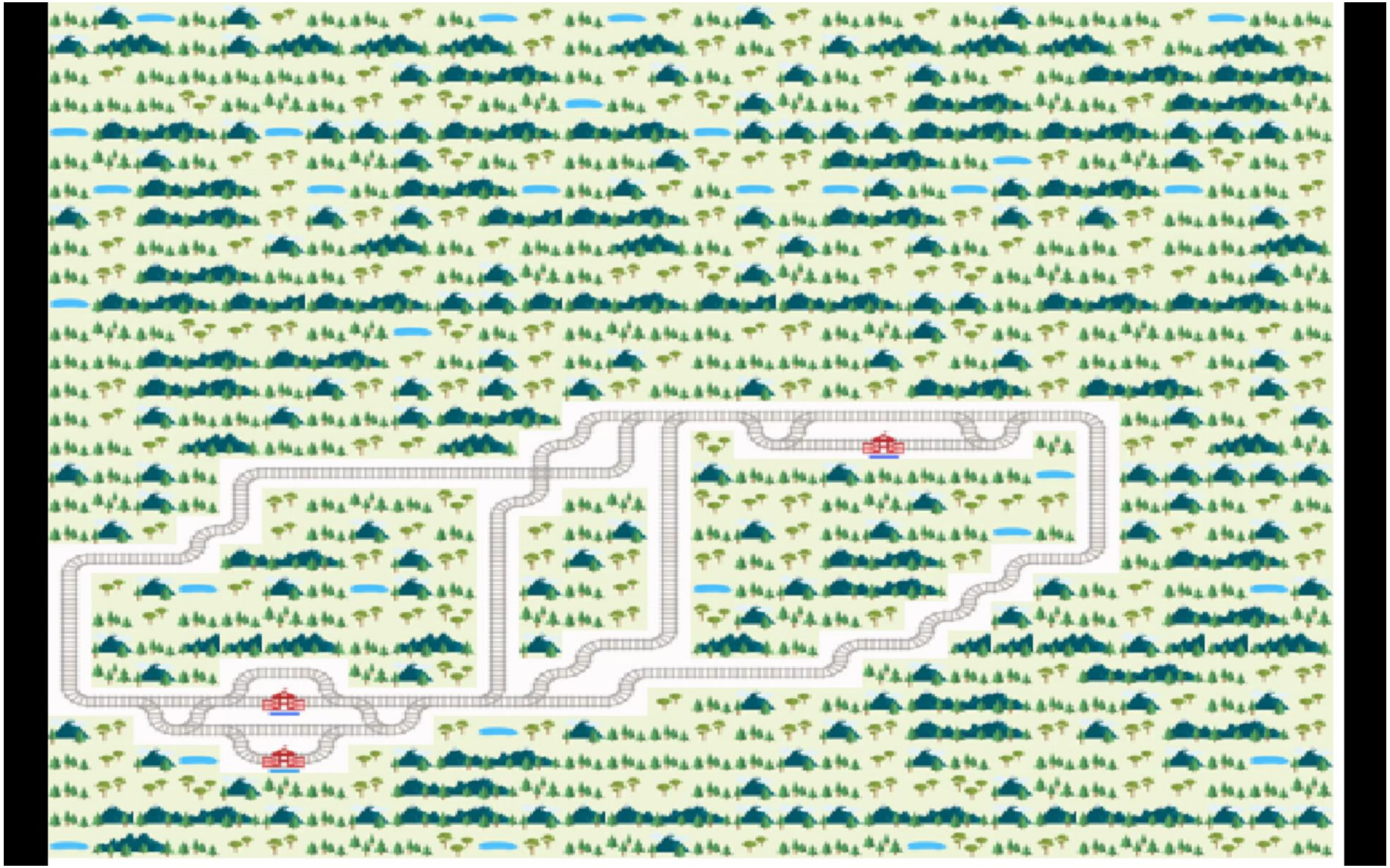


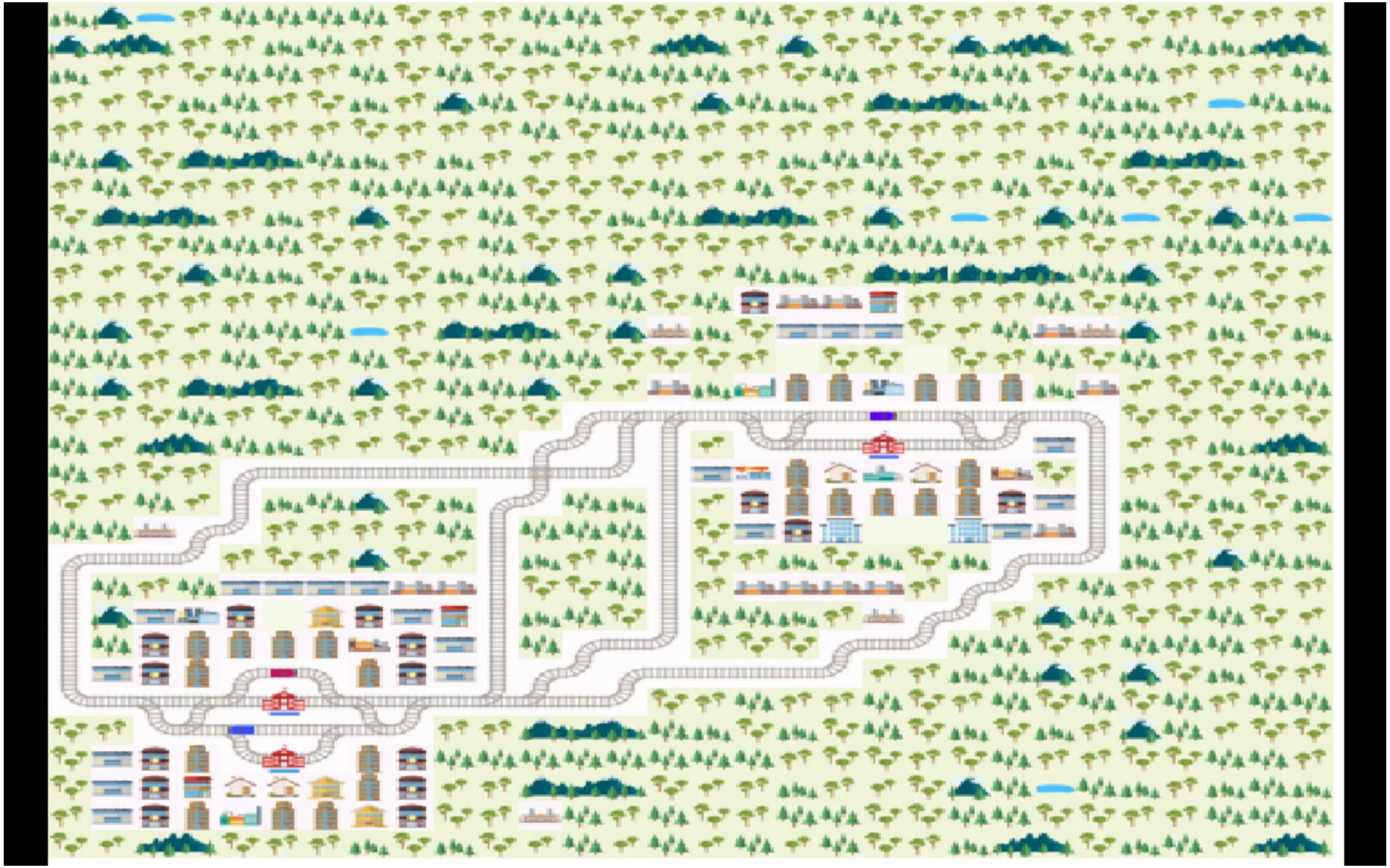
Progressive Training











FLATLAND

Questions?