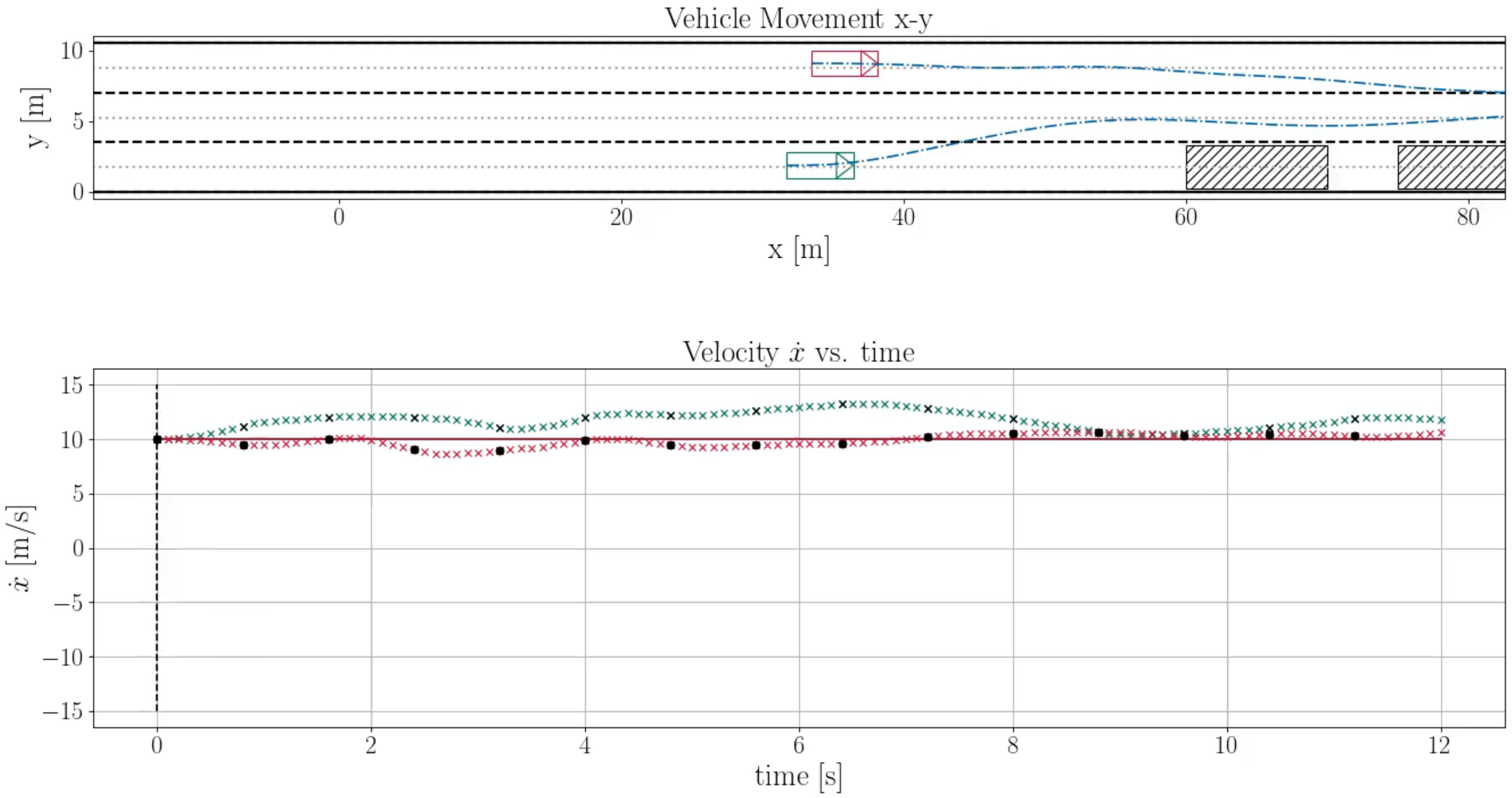
Combining Reinforcement Learning and Search for Cooperative Trajectory Planning

Timo Klein

ANGEWANDTE TECHNISCH-KOGNITIVE SYSTEME
Institut für angewandte Informatik und formale Beschreibungsverfahren, Forschungszentrum Informatik

Cooperative Trajectory Planning



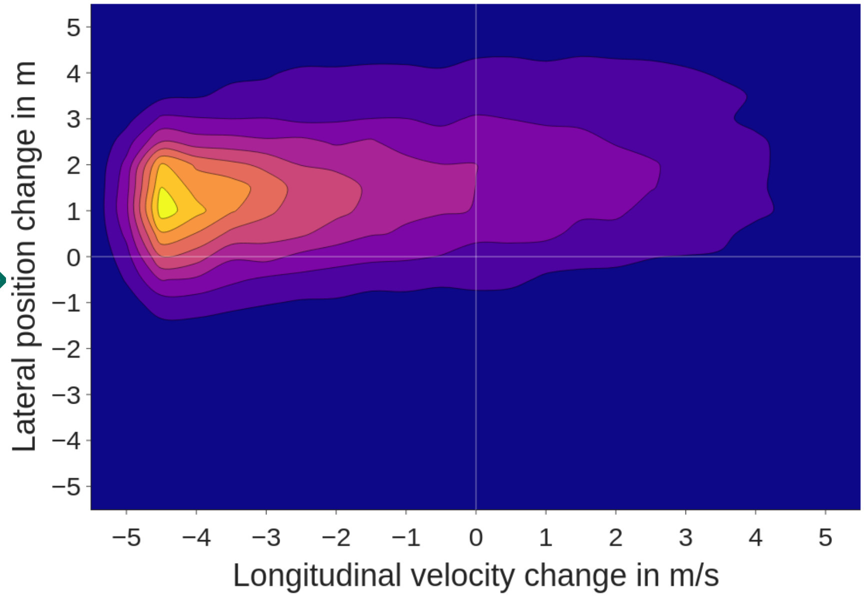
Sources: Kurzer et al. (2018)

Research Question

- Uniform sampling in 2D continuous action space is inefficient
- **Goal**: Increasing sample efficiency through focused sampling
- **Method**: Integrate learned knowledge into the search

Visual input

Sample space



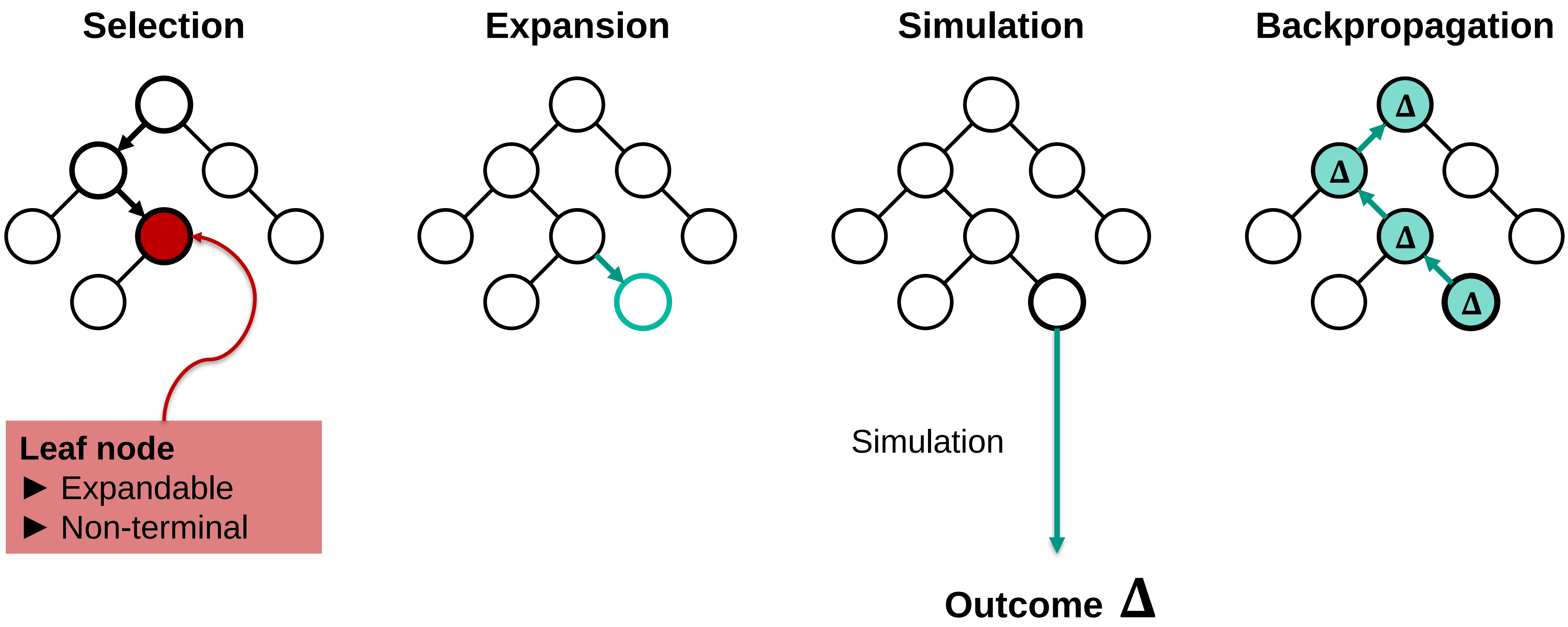
Reinforcement Learning and Search

Karlsruher Institut für Technologie

FZI

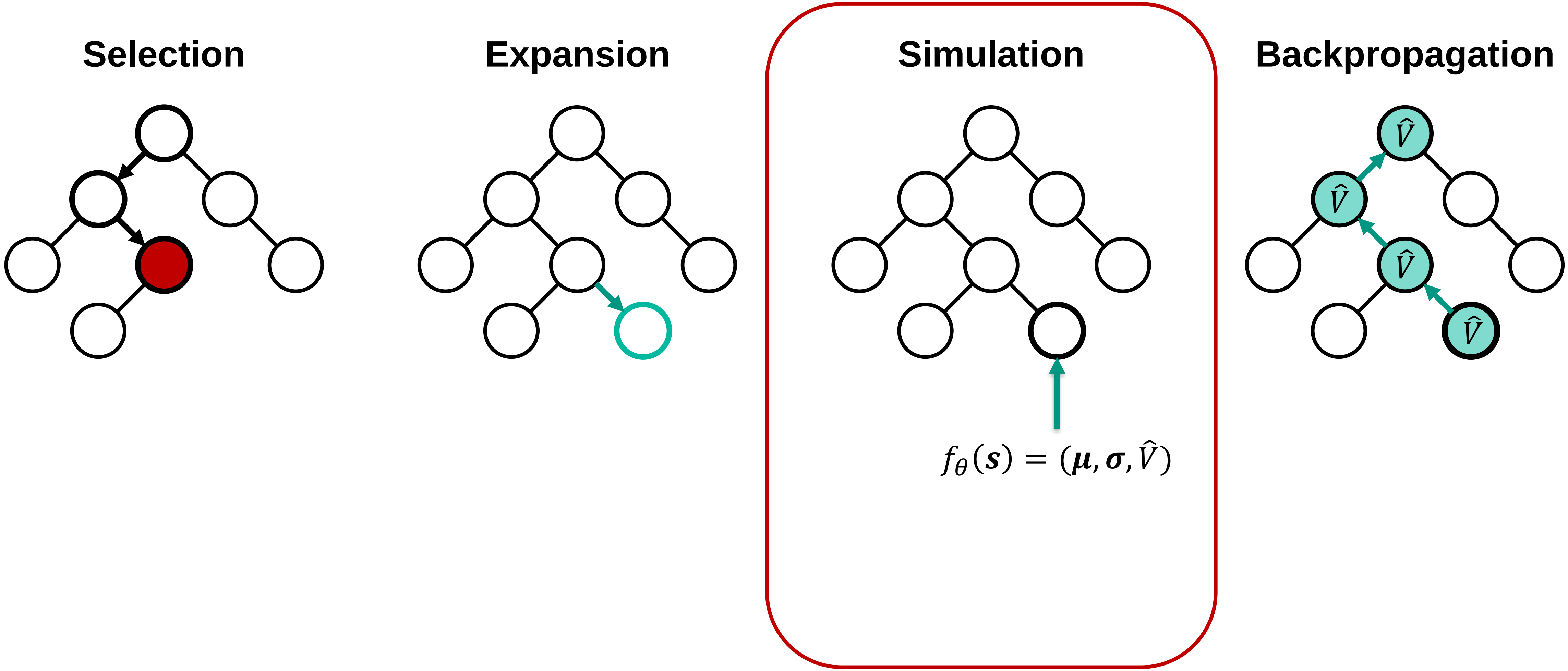
		Action space	
		Discrete	Continuous
Number of agents	Single agent	▪ *AlphaGo* (Silver et al.) ▪ *AlphaGo Zero* (Silver et al.) ▪ *AlphaZero* (Silver et al.) ▪ *MuZero* (Schrittwieser et al.) ▪ *SAVE* (Hamrick et al.) ▪ *Tactical Decision* (Hoel et al.)	▪ *A0C* (Moerland et al.) ▪ *Continuous MuZero* (Yang et al.) ▪ *Sampled MuZero* (Hubert et al.)
	Multi agent	▪ *Multiplayer AlphaZero* (Petosa et al.)	▪ ***This work***

Monte Carlo Tree Search (MCTS)



Sources: Browne et al. (2012)

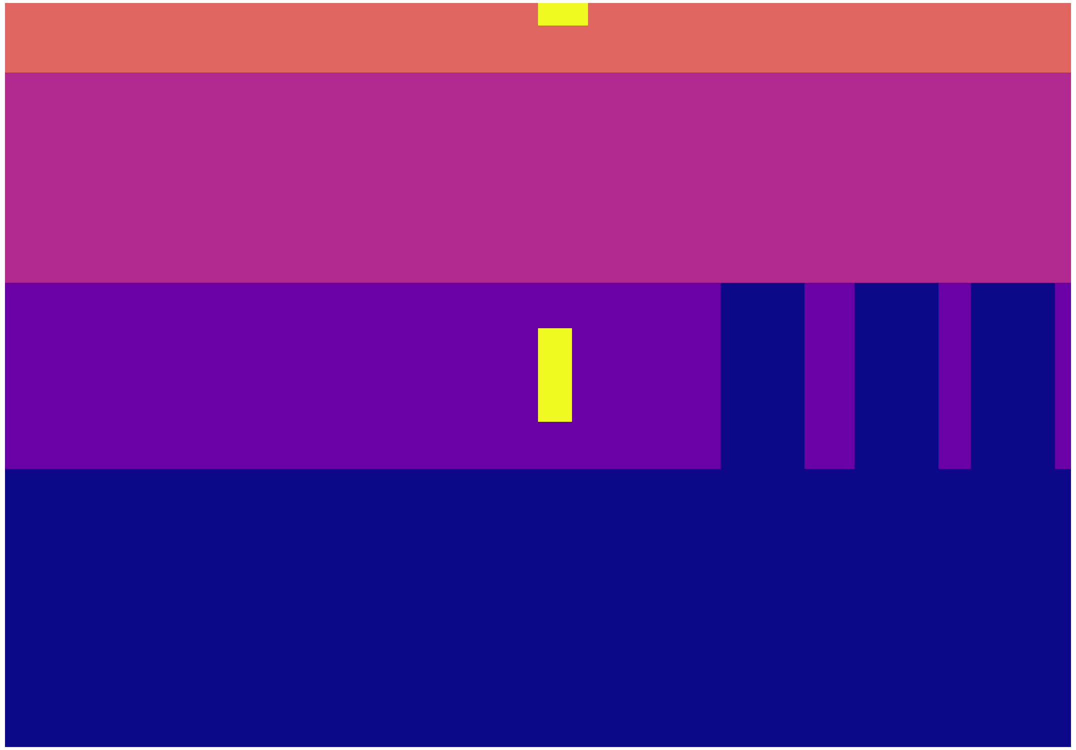
Concept: Guided MCTS



Sources: Moerland et al. (2018), Silver et al. (2017), Silver et al. (2017)

Problem 1: Input Representation

Visual map



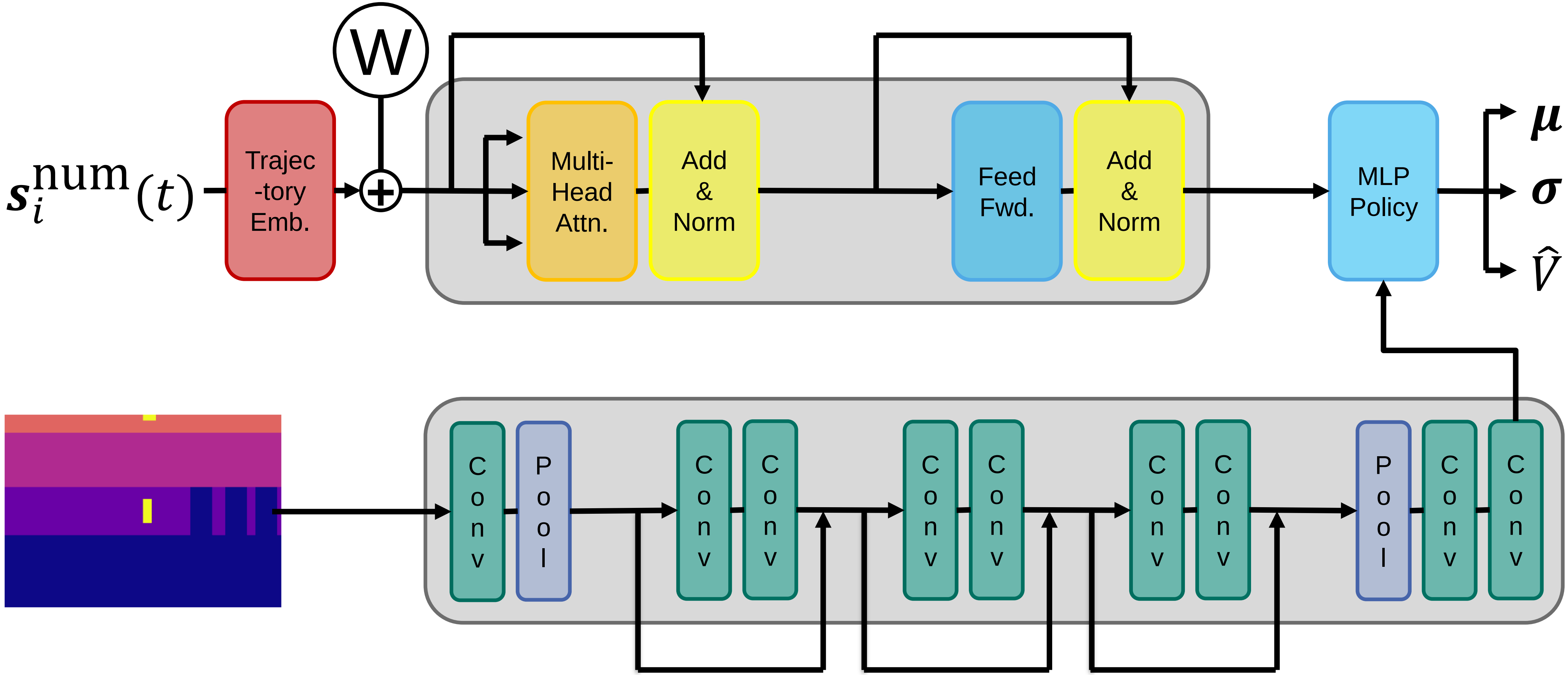
Agent trajectories

$$\boldsymbol{s}_i^{\text{num}}(t) = \begin{pmatrix} \boldsymbol{n}_0(t) \\ \boxed{\boldsymbol{n}_1(t)} \\ \vdots \\ \boldsymbol{n}_i(t) \\ \vdots \\ \boldsymbol{n}_\Upsilon(t) \end{pmatrix}$$

Trajectory of agent *1* relative to the ego agent *i*

Sources: Kurzer et al. (2020), Bacchiani et al. (2019)

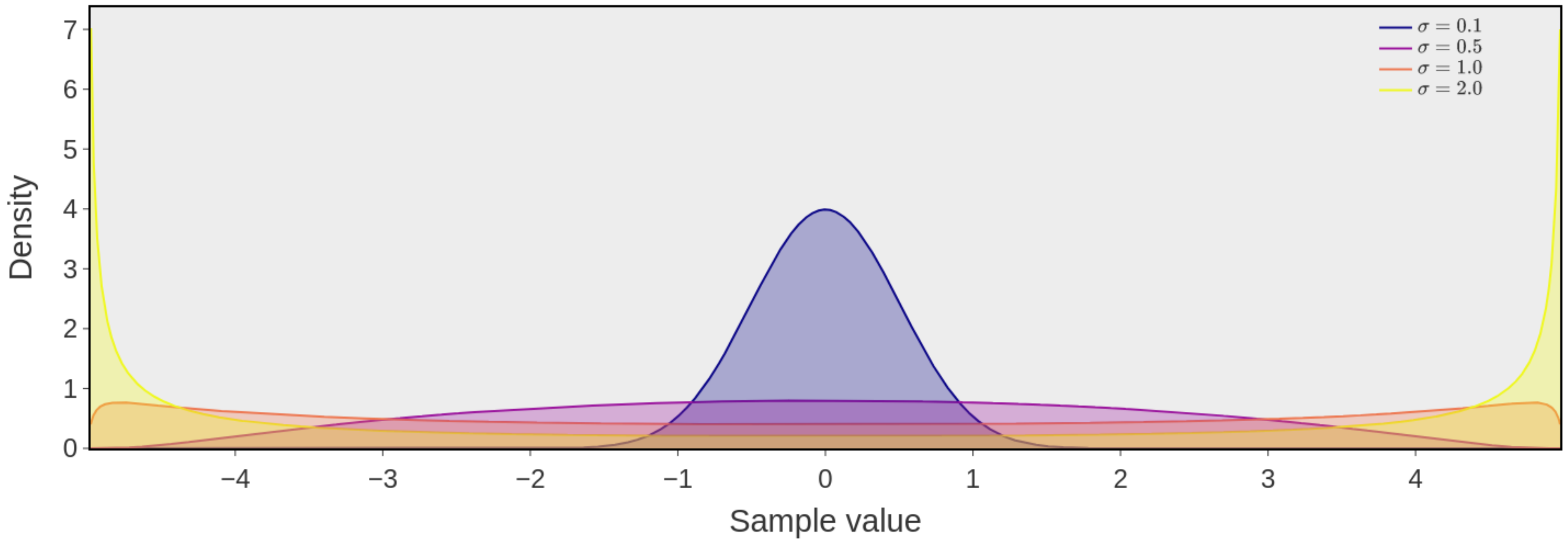
Problem 2: Flexible Number of Agents



Sources: Kurzer et al. (2020), Wang et al. (2020), Bacchiani et al. (2019), Vaswani et al. (2017)

Problem 3: Constraints of Vehicle

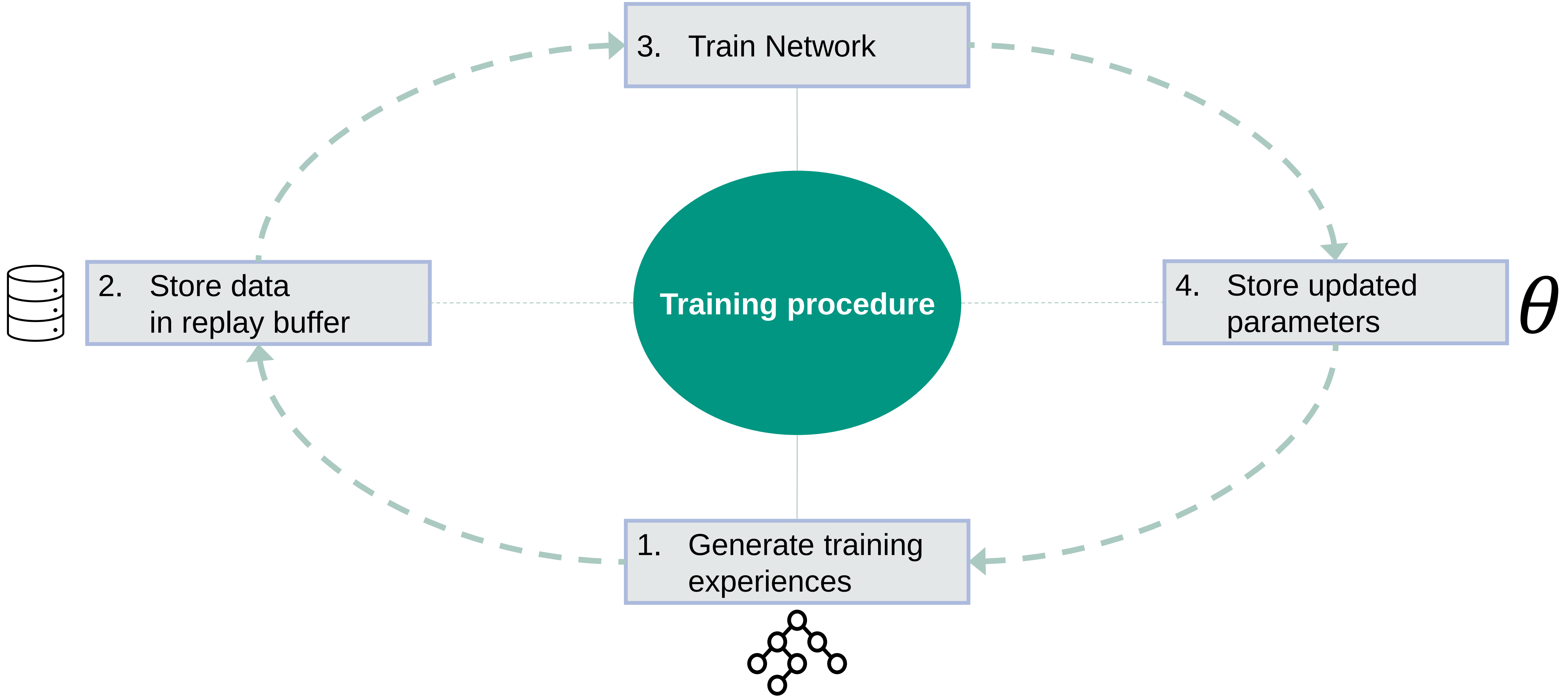
- The support of a normal distribution is unbounded
- Sampled actions $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ can theoretically take any value
- Transformed distribution $\boldsymbol{a} = c \cdot \tanh(\boldsymbol{u})$ enforces action bounds



Sources: Haarnoja et al. (2018)

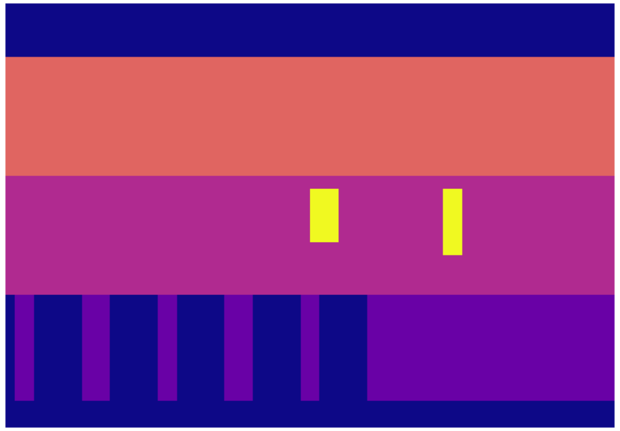
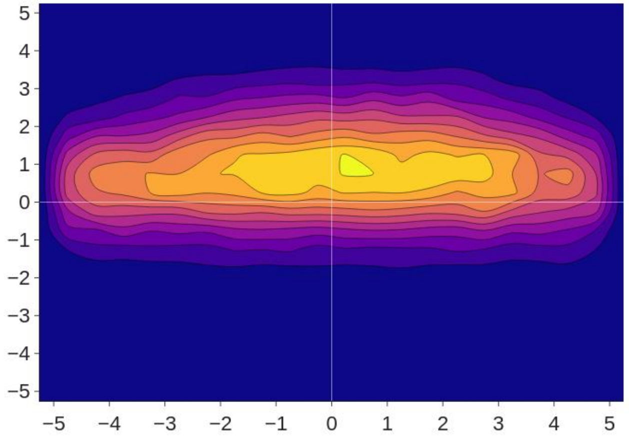
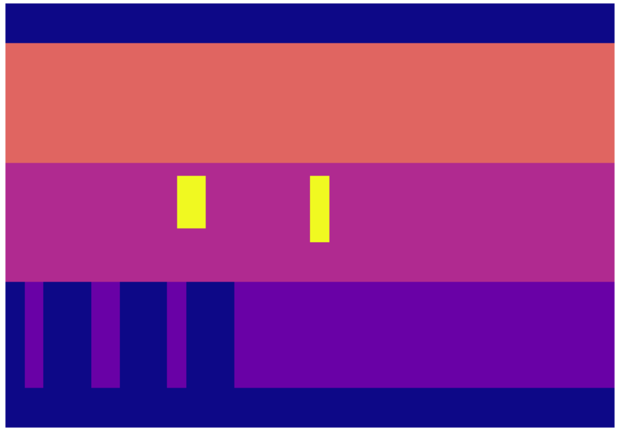
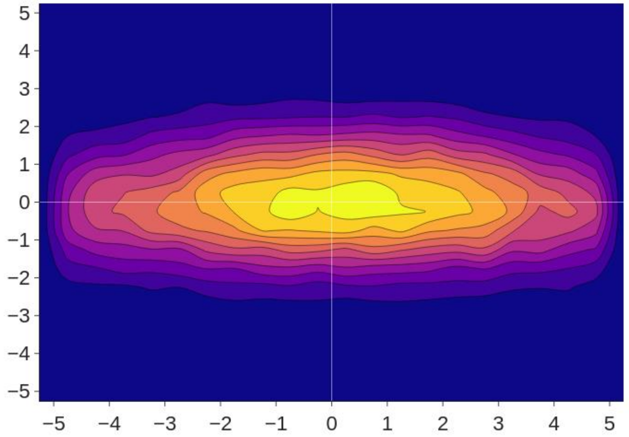
Training

$$\mathcal{L} = \mathcal{L}^{\pi} - \alpha\,\mathcal{L}^{H} + \mathcal{L}^{V}$$

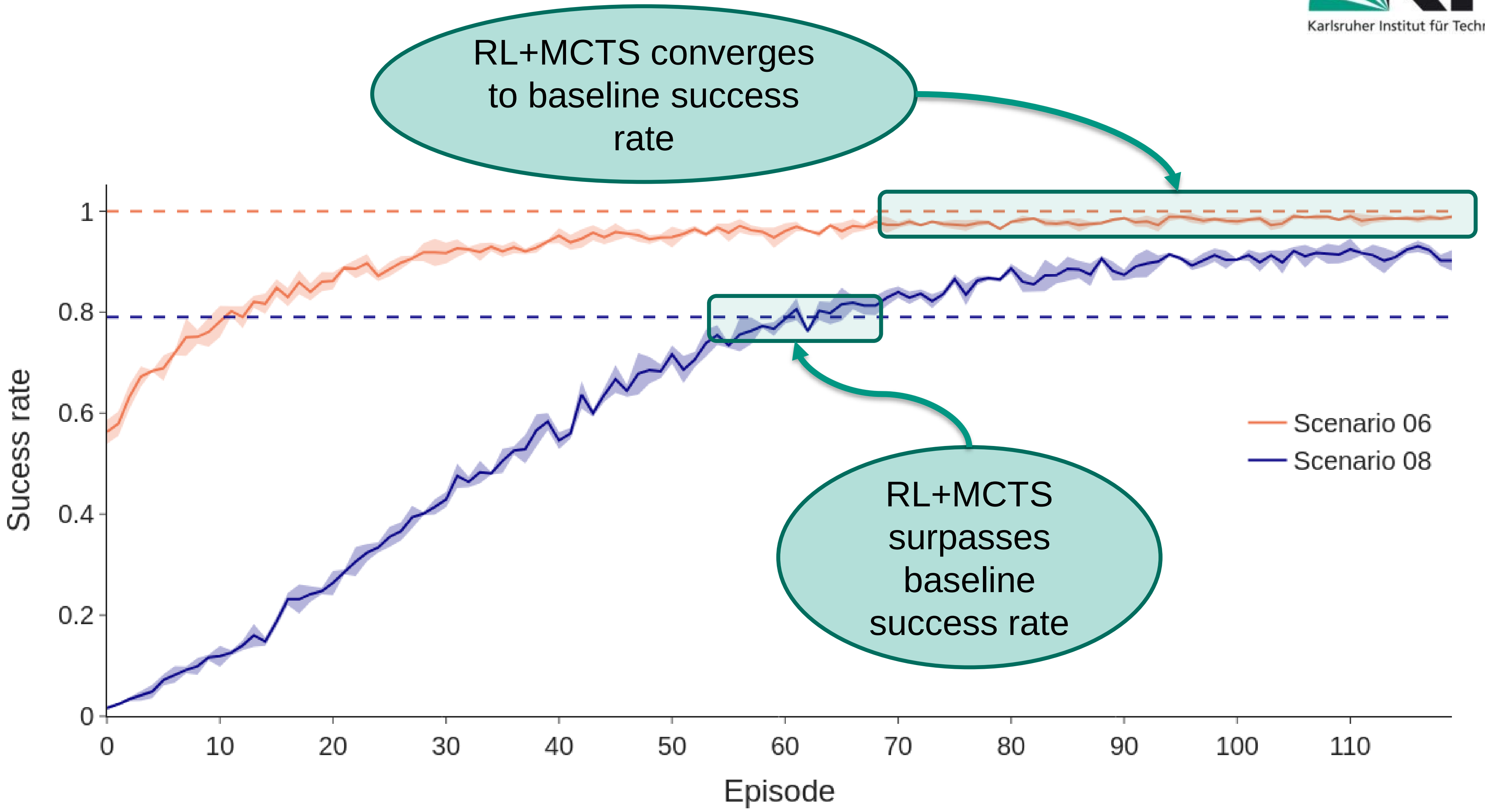


Sources: Moerland et al. (2018), Haarnoja et al. (2018)

Qualitative Evaluation (Scenario 06)

	Visual input	Distribution
Agent 0		
Agent 1		

RL+MCTS vs Baseline



Wall Clock Time

Karlsruher Institut für Technologie

FZI

Scenario 06

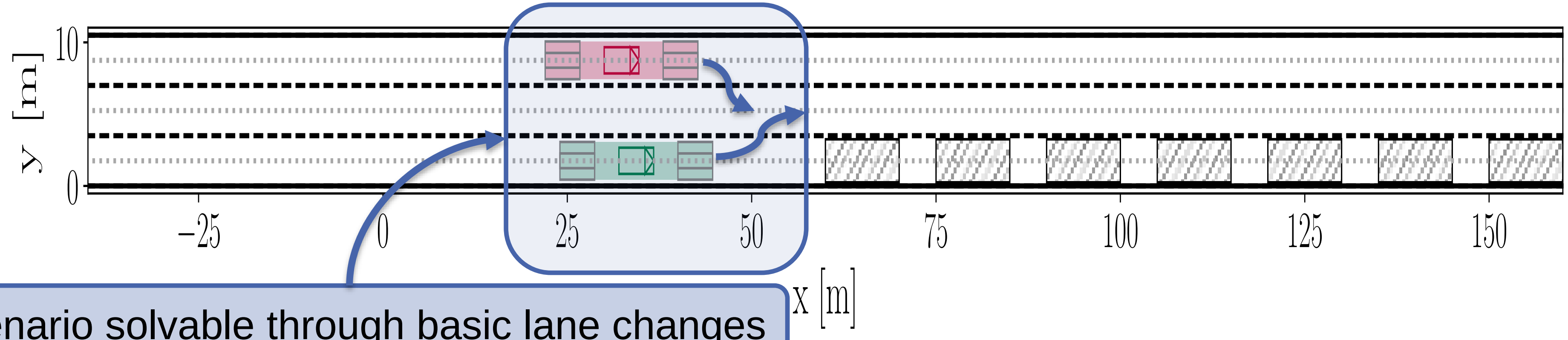
Model	Iterations	Success	Wall clock
RL+MCTS	100	0.99	2m43s
Baseline	100	0.99	4s

Scenario 08

Model	Iterations	Success	Wall clock
RL+MCTS	50	0.84	49s
Baseline	400	0.87	57s

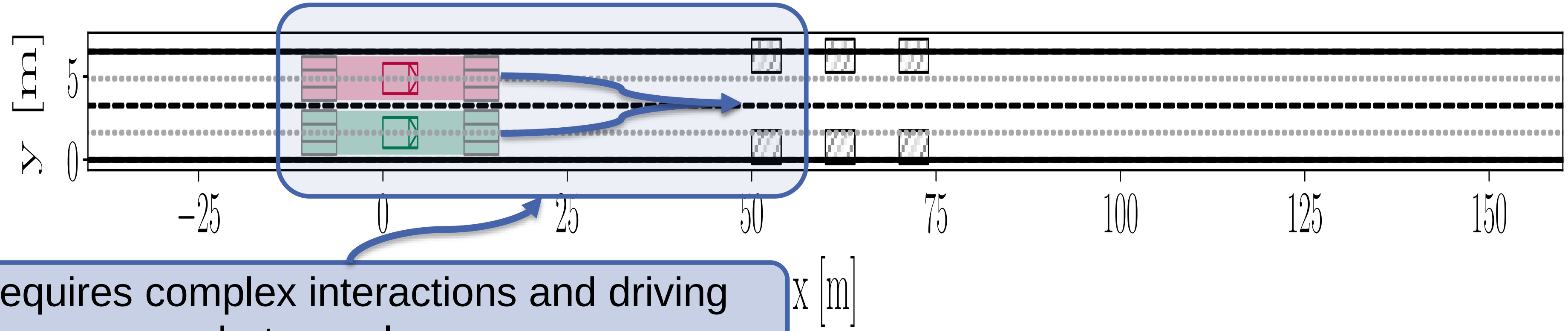
Scenario 08 vs Scenario 06

Scenario 06



Scenario solvable through basic lane changes

Scenario 08

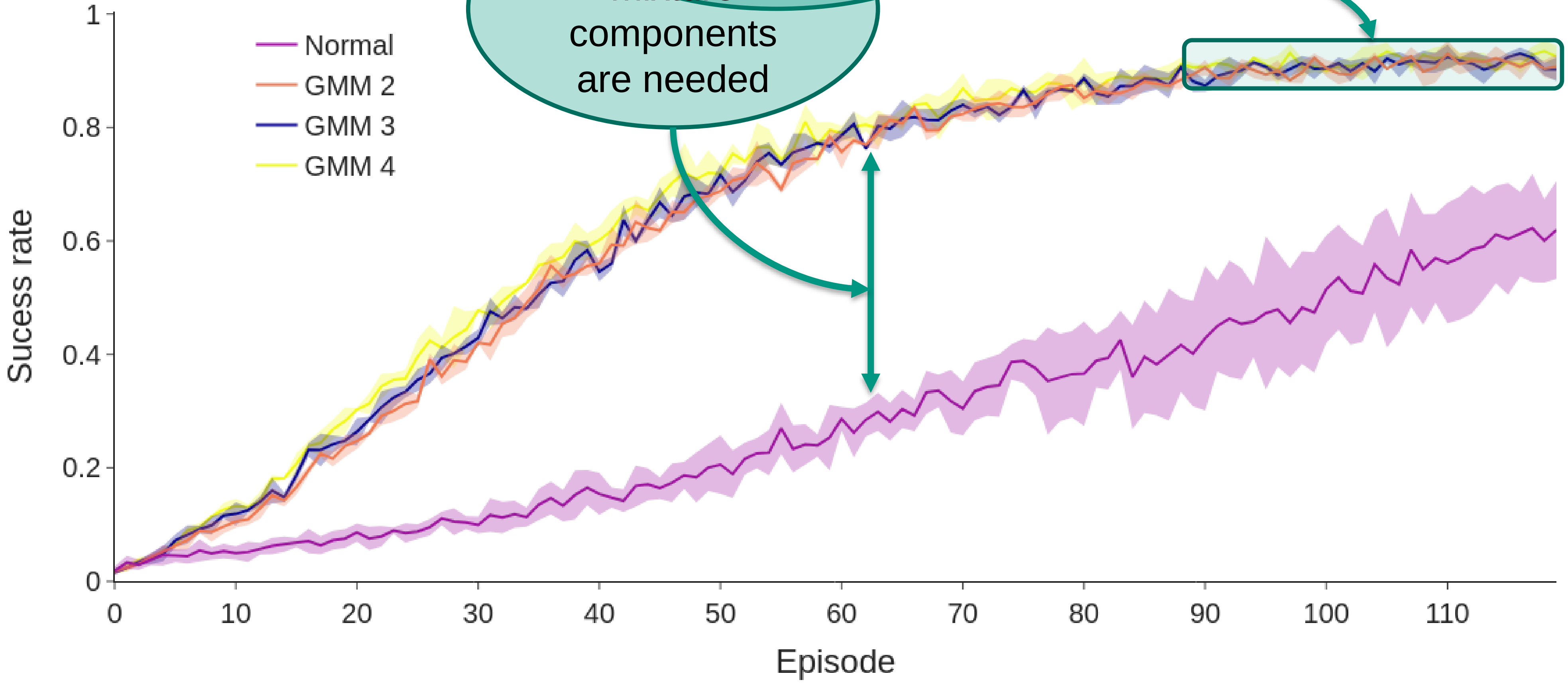


Requires complex interactions and driving between lanes

Mixture Components (Scenario 08)

No significant difference when $K > 1$

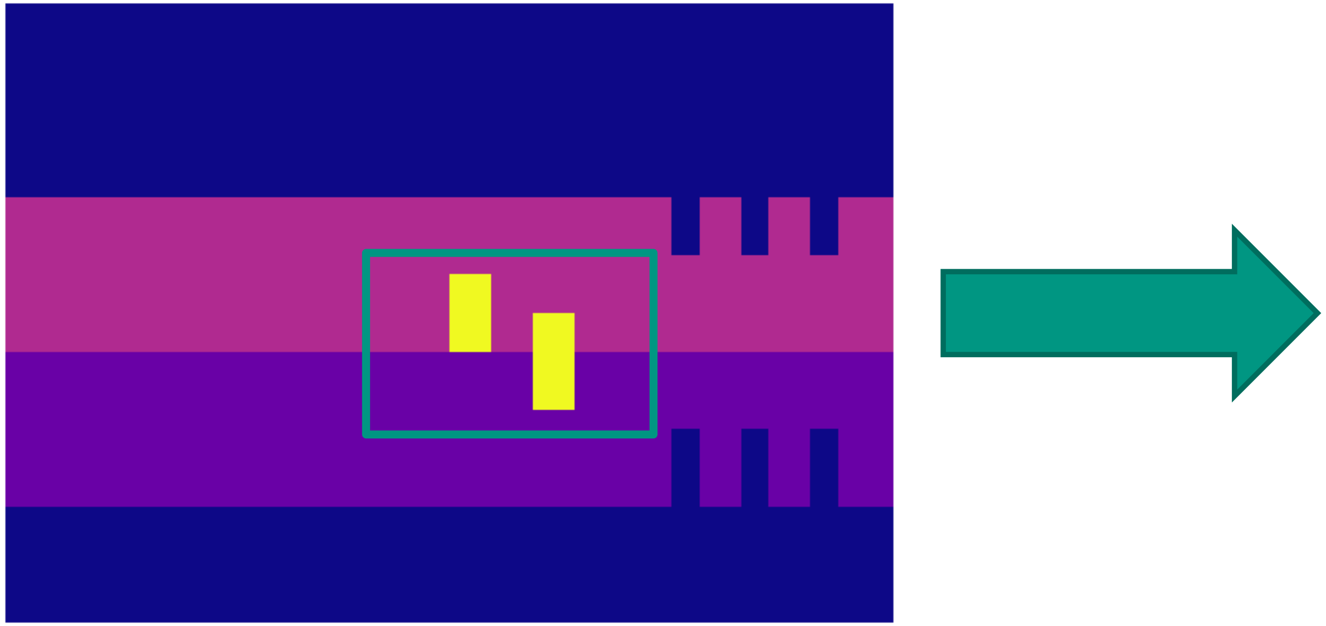
Multiple mixture components are needed



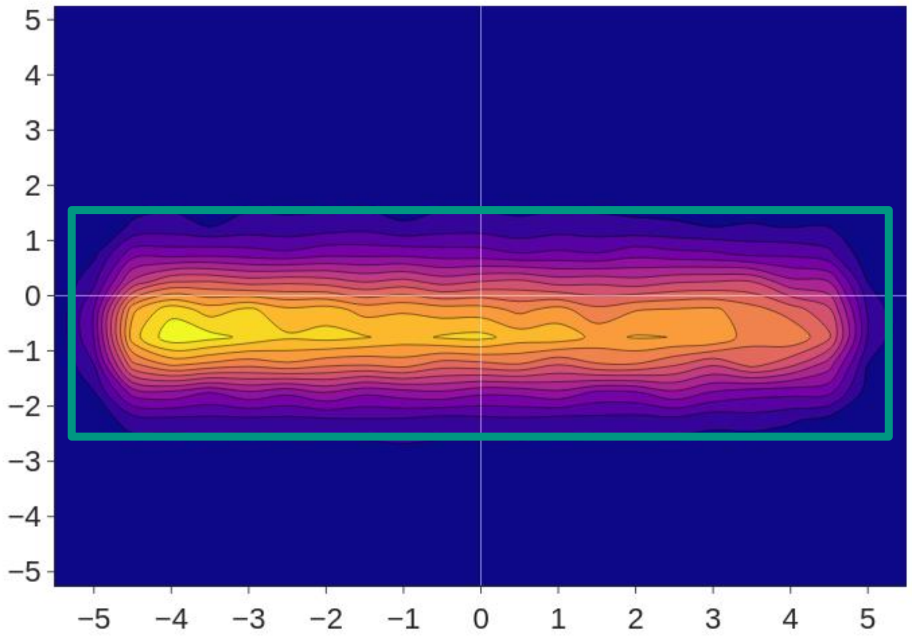
Mixture Components

- Where do multiple components help?
- Sampling over whole longitudinal range helps avoid collisions!

Visual input



Sample space



Conclusion & Outlook

Conclusion
- Efficacy shown
- Moderate generalization
- Scenario dependent

Limitations
- Not efficient
- Hard to scale & tune
- No model learning

Future work
- Scale-up
- Variance scaling
- Integrating heuristics

Reference list

- Electrek.co (2021). Watch Tesla Full Self-Driving Beta navigate through the eye of the system in impressive video. https://electrek.co/2021/03/29/tesla-full-self-driving-beta-navigate-video/. Accessed 2021-08-17.
- Kurzer, K., Engelhorn, F., & Zöllner, J. M. (2018, November). Decentralized cooperative planning for automated vehicles with continuous monte carlo tree search. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 452-459). IEEE.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Lillicrap, T. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. nature, 550(7676), 354-359.
- Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., ... & Silver, D. (2020). Mastering atari, go, chess and shogi by planning with a learned model. Nature, 588(7839), 604-609.
- Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Pfaff, T., Weber, T., Buesing, L., & Battaglia, P. W. (2019). Combining q-learning and search with amortized value estimates. arXiv preprint arXiv:1912.02807.
- Moerland, T. M., Broekens, J., Plaat, A., & Jonker, C. M. (2018). A0C: Alpha zero in continuous action space. arXiv preprint arXiv:1805.09613.
- Yang, X., Duvaud, W., & Wei, P. (2020). Continuous Control for Searching and Planning with a Learned Model. arXiv preprint arXiv:2006.07430.
- Hubert, T., Schrittwieser, J., Antonoglou, I., Barekatain, M., Schmitt, S., & Silver, D. (2021). Learning and Planning in Complex Action Spaces. arXiv preprint arXiv:2104.06303.
- Petosa, N., & Balch, T. (2019). Multiplayer AlphaZero. arXiv preprint arXiv:1910.13012.

Reference list

- Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., ... & Colton, S. (2012). A survey of monte carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games, 4(1), 1-43.
- Kurzer, K., Fechner, M., & Zöllner, J. M. (2020). Accelerating Cooperative Planning for Automated Vehicles with Learned Heuristics and Monte Carlo Tree Search. arXiv preprint arXiv:2002.00497.
- Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020, June). ECA-Net: efficient channel attention for deep convolutional neural networks, 2020 IEEE. In CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Bacchiani, G., Molinari, D., & Patander, M. (2019). Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning. arXiv preprint arXiv:1903.01365.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.
- Wang, H., Emmerich, M., Preuss, M., & Plaat, A. (2019, December). Alternative loss functions in alphazero-like self-play. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 155-162). IEEE.

Generalization Performance

Model Scenario	SC06	SC08	Reg. SC08	SC06 & SC08	SC06 & SC08 + Exp.	Random	Random + Exp.	Baseline
SC01	0.44	0.98	0.93	0.98	**1.00**	0.01	0.99	**1.00**
SC02	0.27	0.72	0.83	0.78	0.95	0.01	0.90	**0.97**
SC03	0.48	0.99	0.98	0.96	**1.00**	0.00	**1.00**	**1.00**
SC04	0.10	0.82	0.87	0.60	**1.00**	0.00	0.99	0.99
SC05	0.70	0.21	0.17	0.61	0.96	0.34	0.95	**1.00**
SC06	0.99	0.39	0.32	0.90	0.98	0.31	0.98	**0.99**
SC07	0.00	0.00	0.02	0.05	0.01	0.00	0.00	**0.07**
SC08	0.39	**0.84**	0.81	**0.84**	0.54	0.00	0.04	0.20
Mean 01-05, 07	0.3317	0.6200	0.6333	0.6630	0.8200	0.0600	0.8050	**0.8383**
Mean unseen	0.3400	0.5871	0.5886	0.6630	**0.8200**	0.0838	0.7313	0.7775
Mean	0.4212	0.6187	0.6162	0.7150	**0.8050**	0.0838	0.7313	0.7775

Trained model with heuristics is overall the strongest

Success Rate

$$I_{success} = \max(1 - I_{collision} - I_{invalid} - I_{unableContinue}, 0)$$

$$P_{success} = \frac{1}{N}\sum_{n=1}^{N} I_{success}^{n}$$

- $I_{collision}$: Indicator for occurred collision
- $I_{invalid}$: Indicator for invalid state, e.g. driving off road
- $I_{unableContinue}$: Situation where the algorithm was unable to find actions for an agent

Numerical State in Detail

$$\mathbf{n}_i^{\text{dynamic}}(t) = \boxed{\left(x_i(t),\ y_i(t),\ \dot{x}_i(t),\ \dot{y}_i(t),\ \ddot{x}_i(t),\ \ddot{y}_i(t),\ \phi_i(t)\right)}$$

$$\mathbf{n}_i^{\text{static}} = \boxed{\left(\dot{x}_i^{\text{desire}},\ l_i^{\text{desire}},\ v_i^{\text{width}},\ v_i^{\text{length}}\right)}$$

$$\mathbf{n}_i(t) = \boxed{\mathbf{n}_i^{\text{dynamic}}(t-7) \oplus \mathbf{n}_i^{\text{dynamic}}(t-6) \oplus \ldots \oplus \mathbf{n}_i^{\text{dynamic}}(t) \oplus \mathbf{n}_i^{\text{static}}}$$

Information at time step *t*

- $\left(x_i(t), y_i(t)\right)$: Position
- $\left(\dot{x}_i(t), \dot{y}_i(t)\right)$: Velocity
- $\left(\ddot{x}_i(t), \ddot{y}_i(t)\right)$: Acceleration
- $\phi_i(t)$: Steering angle

Static information

- $\left(\dot{x}_i^{\text{desire}}, l_i^{\text{desire}}\right)$: Target state
- $\left(v_i^{\text{width}}, v_i^{\text{length}}\right)$: Vehicle dimensions

Agent state

- History of past 8 dynamic states
- Scalar state

Sources: Moerland et al. (2018), Wang et al. (2019)

Training Wall Clock Time (Scenario 06)

- Results are averaged over three models

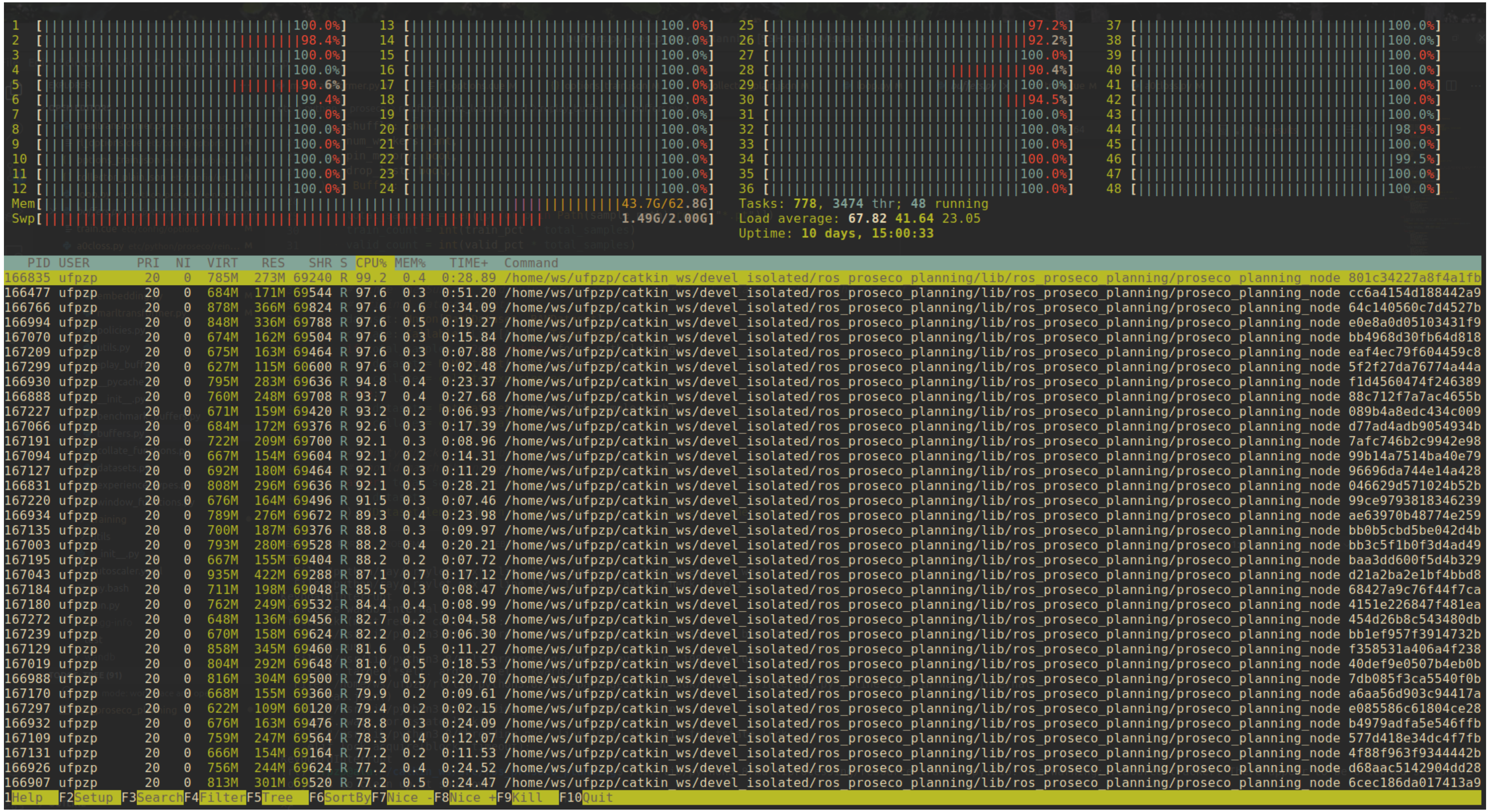
Iterations	50	100	200
Wall clock time	8h53m	14h28m	32h11m
Mean success rate	0.7686	0.7933	0.7867

- ▶ Fast training
- ▶ Decent success rate over tests with multiple iterations

- ▶ Trade-off between training time and success
- ▶ Best performing model

- ▶ 2.5x slower than 50 iterations
- ▶ Success rate moderately improved

Resource Consumption



Loss Function in Detail

$$\mathcal{L}(\phi) = \mathcal{L}_i^{\pi}(\phi) - \alpha_i \mathcal{L}_i^{H}(\phi) + \mathcal{L}_i^{V}(\phi)$$

- **Policy loss**

$$\mathcal{L}^{\pi}(\phi) = D_{KL}\left(\pi_\phi(\boldsymbol{a}|\boldsymbol{s}) \parallel \hat{\pi}(\boldsymbol{a}|\boldsymbol{s})\right) = \mathbb{E}_{\boldsymbol{a}\sim\pi_\phi(\boldsymbol{a}|\boldsymbol{s})}\left[\log \pi_\phi(\boldsymbol{a}|\boldsymbol{s}) - \log \hat{\pi}_\phi(\boldsymbol{a}|\boldsymbol{s})\right]$$

- **Entropy loss**

$$\mathcal{L}^{H}(\phi) = H\left(\pi_\phi(\boldsymbol{a}|\boldsymbol{s})\right) = \mathbb{E}_{\boldsymbol{a}\sim\pi_\phi(\boldsymbol{a}|\boldsymbol{s})}\left[-\log \pi_\phi(\boldsymbol{a}|\boldsymbol{s})\right]$$

- **Value loss**

$$\mathcal{L}^{V}(\phi) = \mathbb{E}_{\boldsymbol{s}\sim\mathcal{D}}\left[\left(V_\phi(\boldsymbol{s}) - \hat{V}(\boldsymbol{s})\right)^2\right]$$

Sources: Moerland et al. (2018), Wang et al. (2019)

Entropy Correction

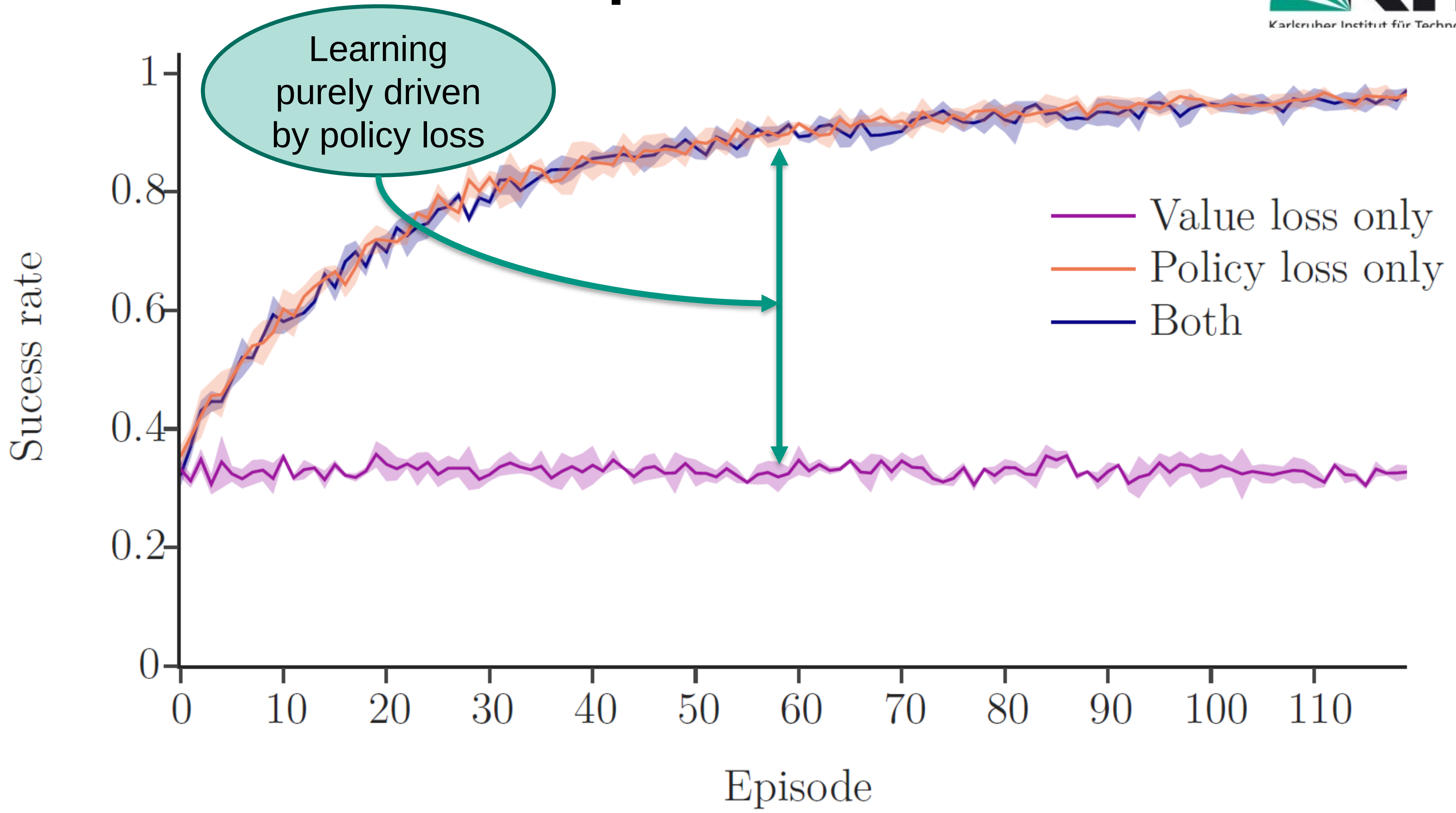
- **Goal**: Express density of transformed distribution $\pi(\boldsymbol{a}|\boldsymbol{s})$ in terms of untransformed distribution $\mu(\boldsymbol{u}|\boldsymbol{s})$
- From the change of variables formula and the multivariable inverse function theorem, we know that

$$\pi(\mathbf{a}|\mathbf{s}) = \mu(\mathbf{u}|\mathbf{s}) \left| \det \frac{d\mathbf{a}}{d\mathbf{u}} \right|^{-1}$$

$$= \log \mu(\mathbf{u}|\mathbf{s}) - \sum_{i=1}^{D} \log \left(1 - \tanh^2(u_i) \right)$$

1. Plugging the derivative into $\left|\det \frac{d\,\boldsymbol{a}}{d\,\boldsymbol{u}}\right|^{-1}$
2. Simplifying using the logarithm rules

Importance of Loss Components



Reward for Scenario 08

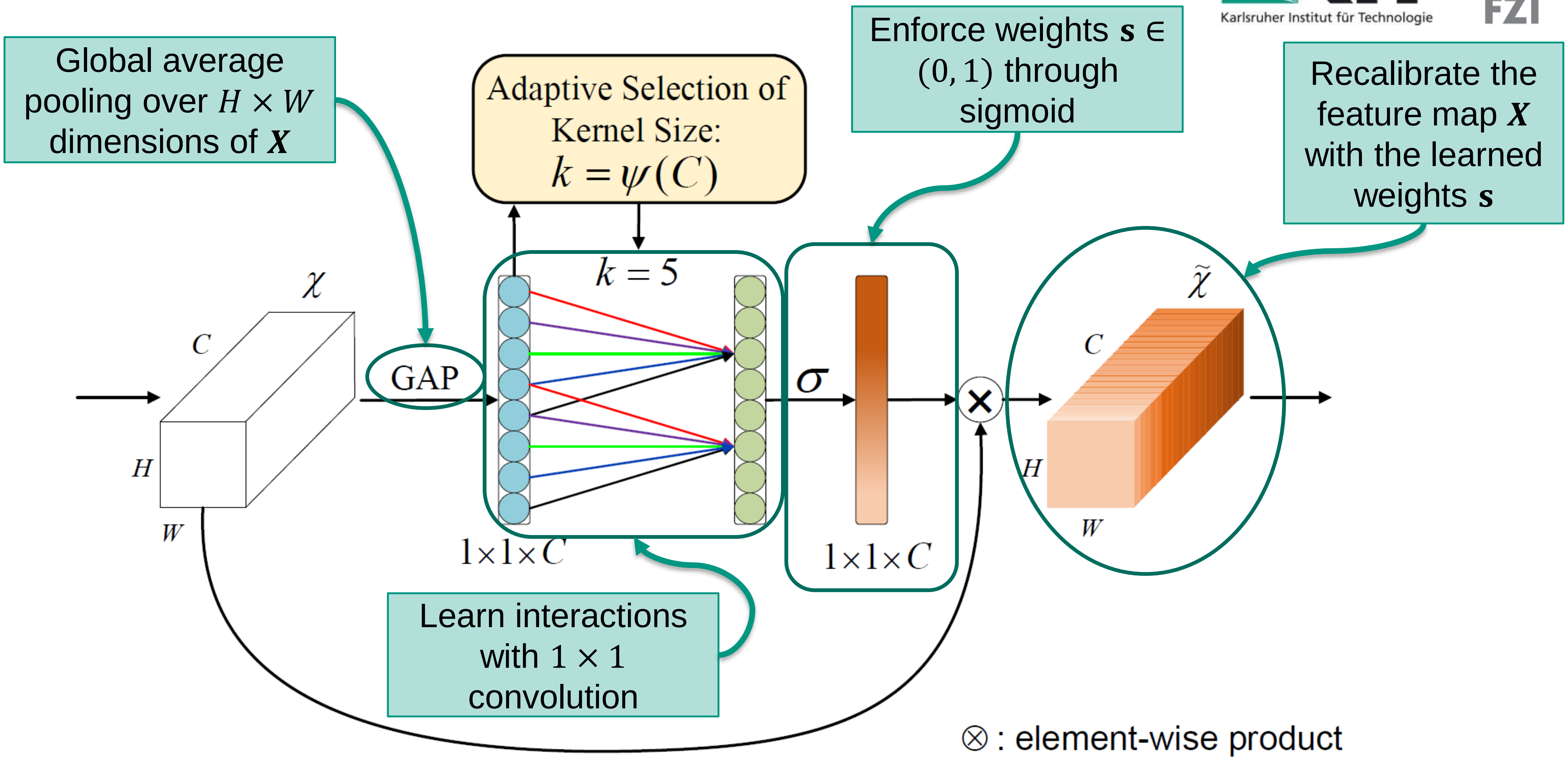
Model	Iterations / Metric	5	10	25	50	100	200	400	Mean
GMM 3	Success	0.3000	0.4967	0.6833	0.8267	0.8300	0.9433	0.9500	**0.7186**
	Reward	0.2779	0.2954	0.3119	0.3205	0.3276	0.3341	0.3367	**0.3149**
	Desire	0.0067	0.0200	0.0567	0.1767	0.1333	0.2900	0.3833	**0.1524**
Baseline	Success	0.0200	0.0100	0.0200	0.0900	0.2000	0.7900	0.8700	**0.2857**
	Reward	0.6738	0.6430	0.6404	0.6096	0.6145	0.6255	0.6238	**0.6330**
	Desire	0.0000	0.0100	0.0000	0.0000	0.0002	0.0002	0.1000	**0.0214**

- Baseline achieves highest reward for lowest success rate
- Reward is hand-crafted with possibly suboptimal parameters
- Other student worked on this via IRL

Full Results for Scenario 06

Model	Iterations / Metric	5	10	25	50	100	200	400	Mean
GMM 3	Success	0.4367	0.5733	0.7133	0.8533	0.9467	0.9900	0.9933	**0.7867**
	Reward	0.2168	0.2345	0.2564	0.2755	0.2901	0.2986	0.3045	**0.2681**
	Desire	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0000**
Baseline	Success	0.8700	0.9600	1.0000	0.9900	0.9900	1.0000	1.0000	**0.9729**
	Reward	0.5798	0.4976	0.4943	0.4633	0.4791	0.5126	0.5157	**0.5060**
	Desire	0.0000	0.0200	0.0100	0.0300	0.0000	0.0000	0.0000	**0.0086**

Efficient Channel Attention (ECA)



Lookout & Pitfalls

Instability of GMM policies in RL

Instability of Transformers in RL

Averaging over agents in loss

MCTS
+
RL

Generating enough training data

Inaccuary of entropy estimation

Discretization vs continuous control

Sources: Huber et al. (2008), Tian et al. (2019), Parisotto et al. (2019)