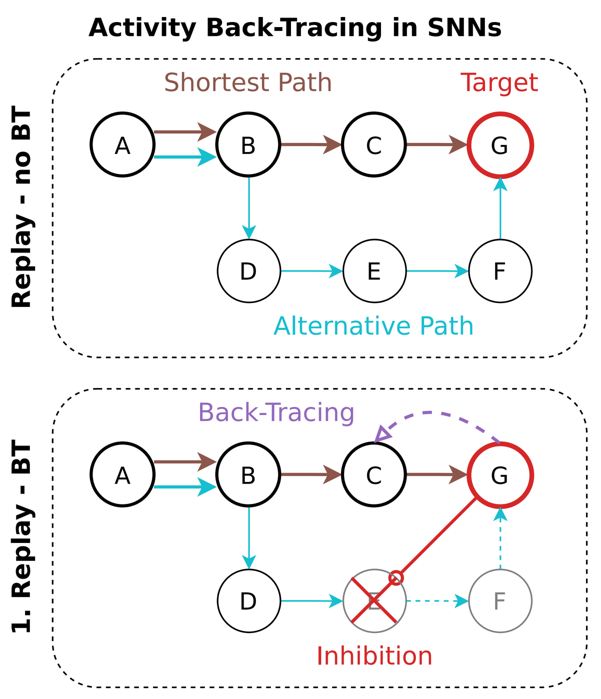
Threshold Adaptation in Spiking Networks Enables Shortest Path Finding and Place Disambiguation

Robin Dietrich[1,2], Tobias Fischer[2], Nicolai Waniek[3], Nico Reeb[1,2], Michael Milford[2], Alois Knoll[1], Adam Hines[2]

[1]School of Computation, Information and Technology, Technical University of Munich, Munich, Germany.
[2]QUT Centre for Robotics, School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, Australia.
[3]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.

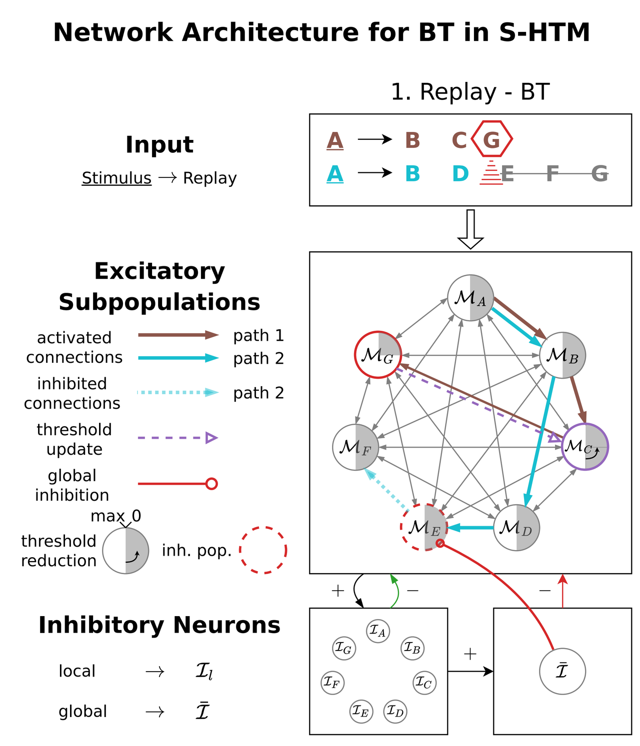
Introduction



Motivation

- Training of Spiking Networks challenging
  → no backwards information, not differentiable
  → backpropagation only using approximations
- Classic graph algorithms use back-tracing of information for, e.g., path planning
- SNNs for graph computations
  → require additional learning rules [1] or backwards connections [2]
  → rely on single neuron representation for location [1, 2]

⇒ Back-tracing in SNNs without connection modifications not yet studied



Background: S-HTM

- Spiking hierarchical temporal memory (S-HTM) [3]
  → Sequence prediction, mismatch det., and replay
  → Sequences: excitatory con., e.g., $\mathcal{M}_A \rightarrow \mathcal{M}_B$

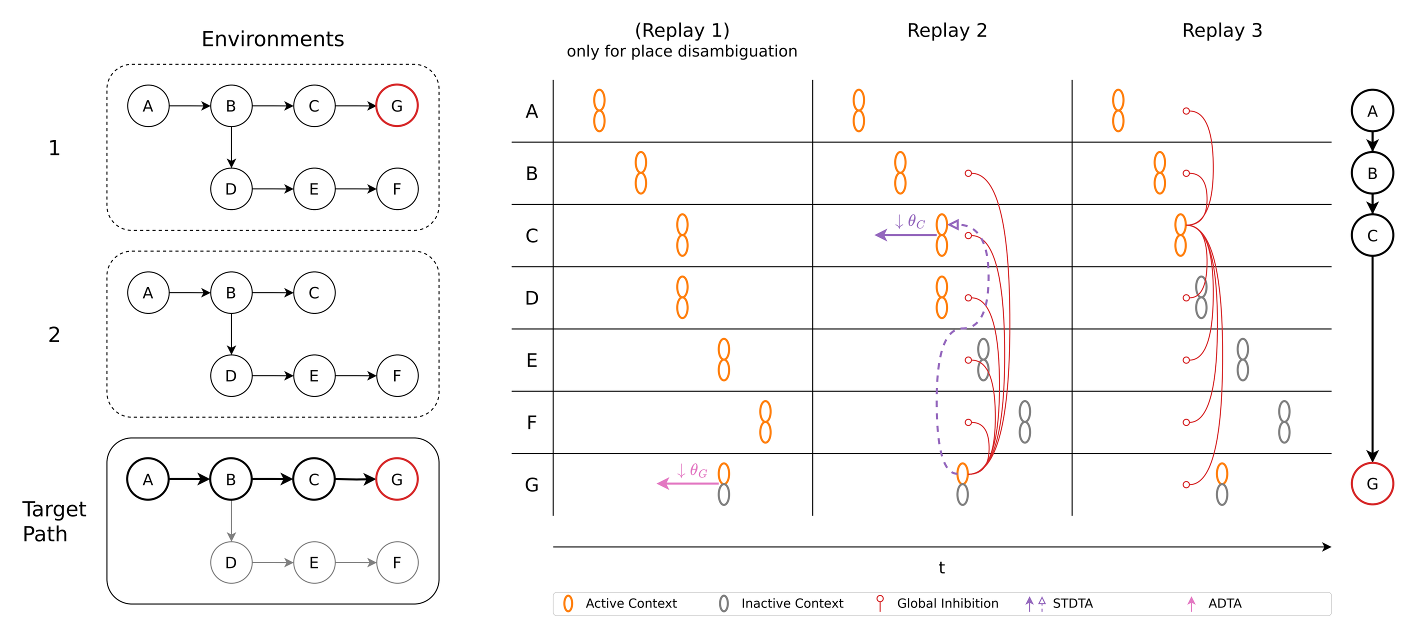
Approach: Network for Back-Tracing with Replay

- Population encoding for locations ($\mathcal{M}_l$)
- Parallel replay of all sequences from start location
- Path/Place selection through global inhibition, triggered by threshold adaptation during replay

Method: Back-Tracing for Path Planning & Place Disambiguation

Shortest Path Finding

1. **Manual target selection** (prior Replay 2)
   → Reduce threshold $\theta_{\mathcal{M}_\Phi}$ of target pop. ($\mathcal{M}_\Phi$):
   $$\theta_{\mathcal{M}_\Phi}(r_0) = \theta_{\mathcal{M}_\Phi}(r_0) \cdot \lambda_\Phi, \quad \text{target rate } \lambda_\Phi$$
2. **Back-tracing rule** (Replay 2 → 3)
   → Spike timing-dependent threshold adaptation (STDTA) for each $l \in \{A, B, C, \ldots\}$:
   $$\theta_{\mathcal{M}_l}(r) = \theta_{\mathcal{M}_l}(r-1) \cdot \lambda_b, \quad \text{back-tracing rate } \lambda_b$$

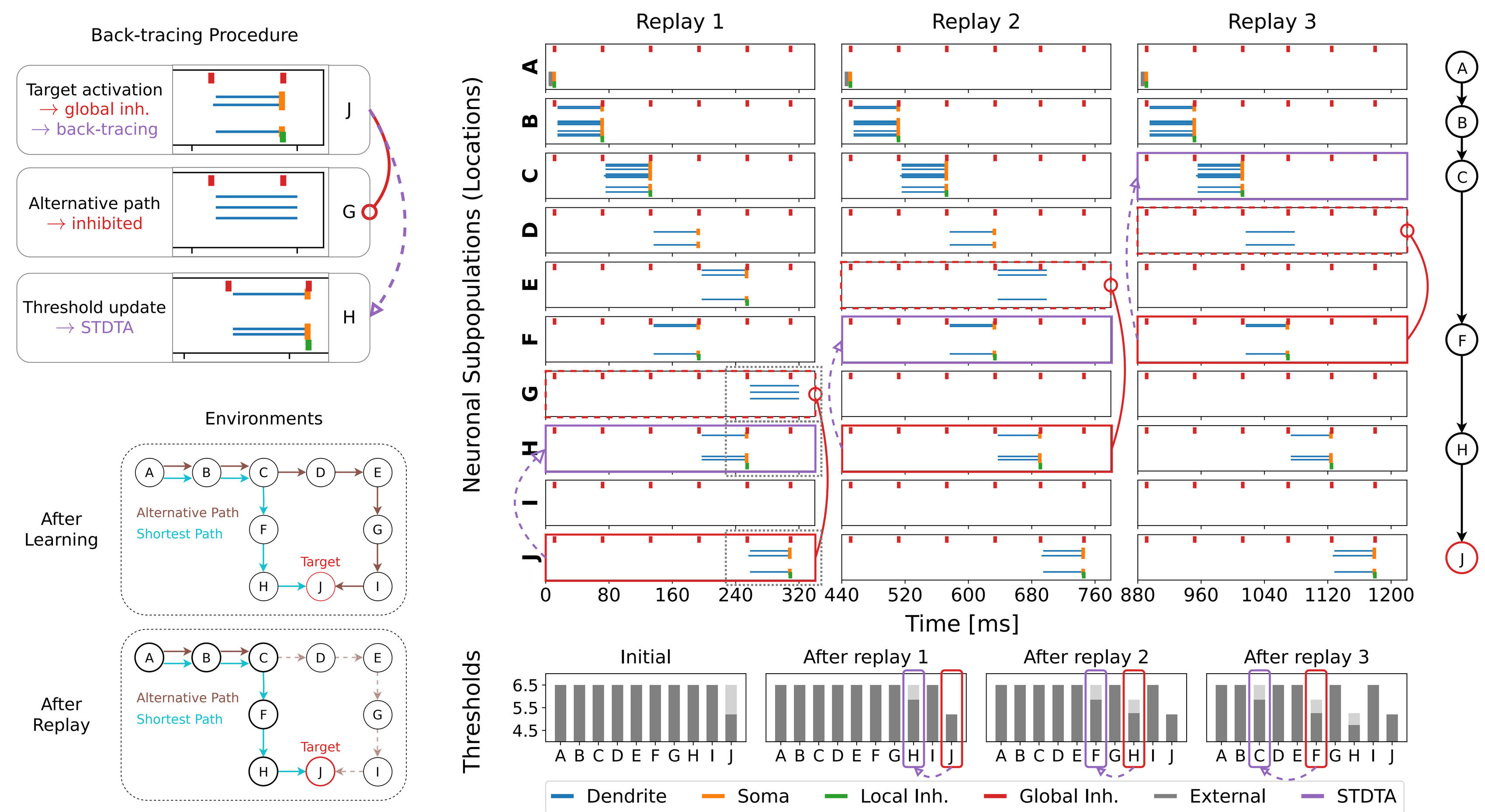


Place Disambiguation

1. **Neuronal populations encode ambiguity**
   → Population $\mathcal{M}_l$ represents multiple contexts
   → # active neurons (replay) ∝ ambiguity of place
2. **Target selection by ambiguity** (Replay 1)
   → Ambiguity dep. threshold adaptation (ADTA):
   $$\theta_{\mathcal{M}_l}(r) = \theta_{\mathcal{M}_l}(r) \cdot e^{\gamma(F_a - F_\rho)} \cdot \lambda_a, \text{ with}$$
   active (F_a) and targeted (F_ρ) # neurons per context, slope of in-/decrease (γ) and ambiguity rate (λ_a)

Experimental Results

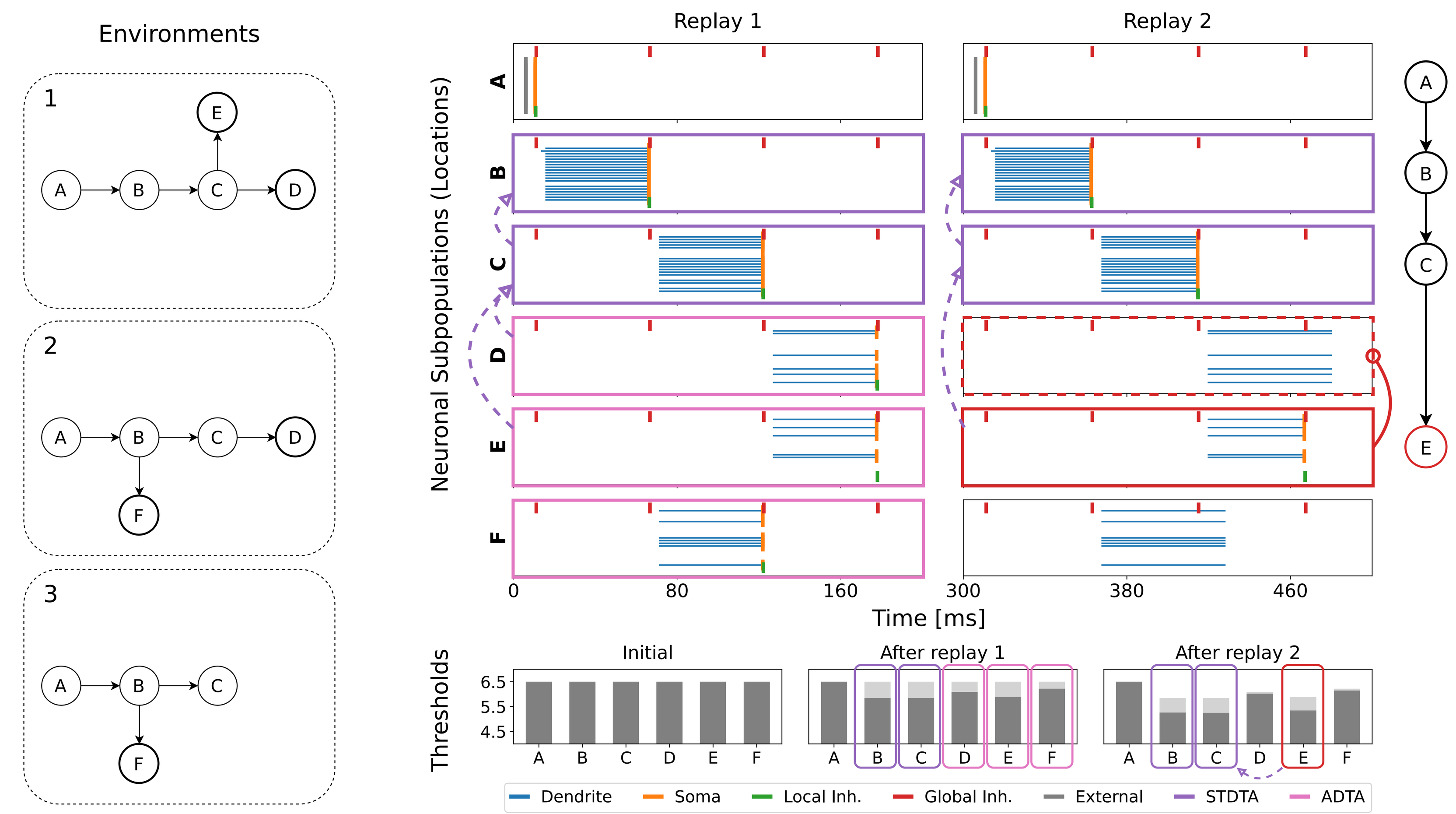
Shortest Path Finding



1. Reduced threshold of target population ($\mathcal{M}_J$)
   → Inhibition of simultaneously active populations ($\mathcal{M}_G$)
2. Back-tracing of information through backwards threshold update (STDTA)
   → Inhibition of alt. paths (C-D-E-G-I-J), selection of shortest path (C-F-H-J)

⇒ Shortest path calculation in 3 (max. L, shortest path length) replays

Place Disambiguation



1. Case: Binary disambiguation (Envs. 1, 2)
   → Populations with lower number of active neurons spike earlier (ADTA)
2. Case: Multiple ambiguities (Envs. 1, 2, 3)
   → Discrimination between ambiguities: adjust ambiguity rate λ_a

⇒ Place disambiguation between multiple places with varying ambiguity

Conclusion

1. **Shortest path finding in SNNs with back-tracing**
   → Threshold adaptation (STDTA) during replay enables information back-tracing with local learning rule
   → Shortest path found in $\leq L$ (path length) replays
2. **Place disambiguation for localization improvement**
   → Ambiguity representation with population code
   → Discrimination of places with varying ambiguity using ADTA
   → Path planning to selected place with low ambiguity

Limitations & Outlook

Current limitations

- Path length is limited by max. threshold adaptation
- Ambiguity discrimination requires parameter adjustment

Towards real-time neuro-robotic navigation

- Integration of real-world sensor data processing with multi-column structure for robotic application
- Neuromorphic implementation (Loihi 2) for real-time evaluation

Acknowledgements

This research was partially supported by funding from ARC DECRA Fellowship DE240100149 to TF and an ARC Laureate Fellowship FL210100156 to MM. The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. This work has been supported by a fellowship of the German Academic Exchange Service (DAAD) and by the FFG, Contract No. 881844: “Pro^{2}Future”.

References

[1] Filip J. Ponulak et al. “Rapid, Parallel Path Planning by Propagating Wavefronts of Spiking Neural Activity”. In: *Frontiers in Computational Neuroscience* 7 (July 2013).

[2] Catherine D. Schuman et al. “Shortest Path and Neighborhood Subgraph Extraction on a Spiking Memristive Neuromorphic Implementation”. In: *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*. NICE ’19. New York, NY, USA: Association for Computing Machinery, Mar. 2019, pp. 1–6.

[3] Younes Bouhadjar et al. “Sequence learning, prediction, and replay in networks of spiking neurons”. en. In: *PLOS Computational Biology* 18.6 (June 2022).