Vector Quantization in the Brain: Grid-like Codes in World Models

Xiangyuan Peng*, Xingsi Dong*, Si Wu

Peking University

Wechat Public Account

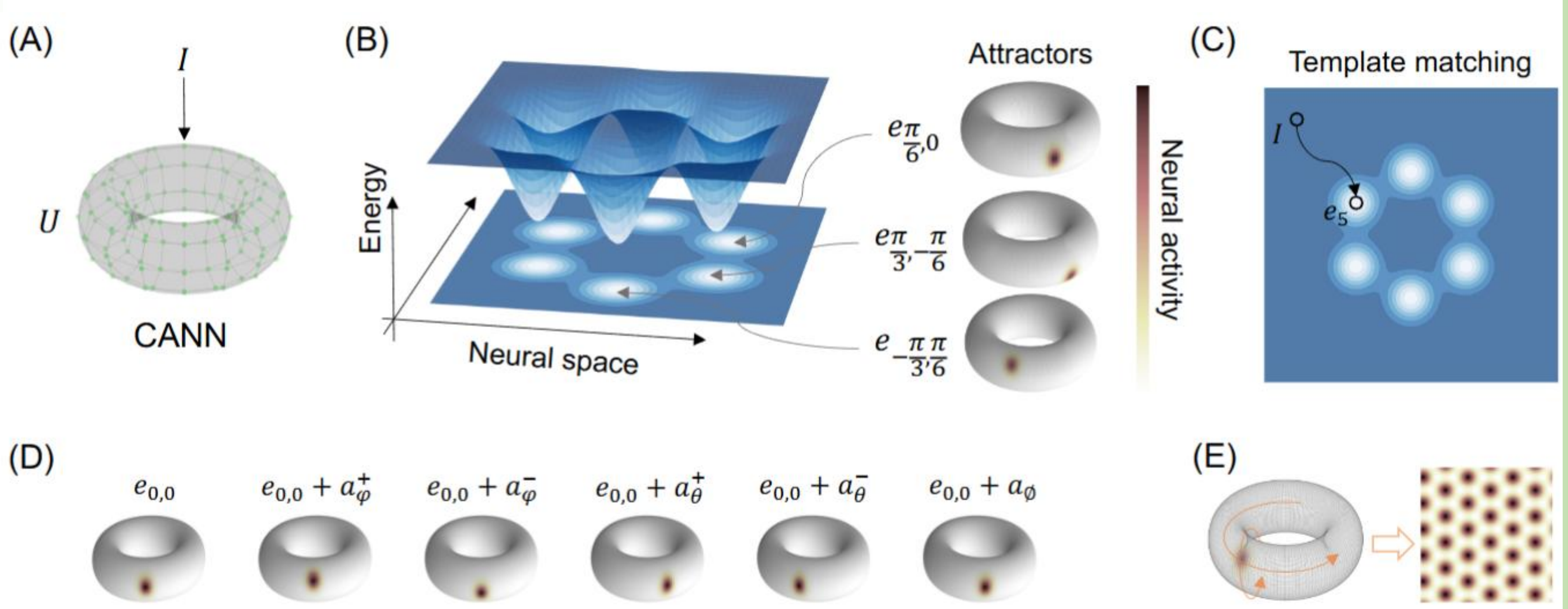
Email: 2301111588@pku.edu.cn

dxs19980605@pku.edu.cn

Introduction

- We propose **Grid-like Code Quantization** (GCQ), a brain-inspired method for compressing observation-action sequences into discrete representations using grid-like patterns in **attractor** dynamics.
- GCQ performs **spatiotemporal compression** through an action-conditioned codebook, where codewords are derived from continuous attractor neural networks and dynamically selected based on actions.
- GCQ jointly compresses space and time, serving as **a unified world model**, differs from the conventional two-stage architecture.
- GCQ yields a cognitive map, which supports long-horizon prediction, goal-directed planning, and the derivation of an inverse model.
- Insights for neuroscience: the brain may simultaneously achieve compression and semantic organization of sensory information.

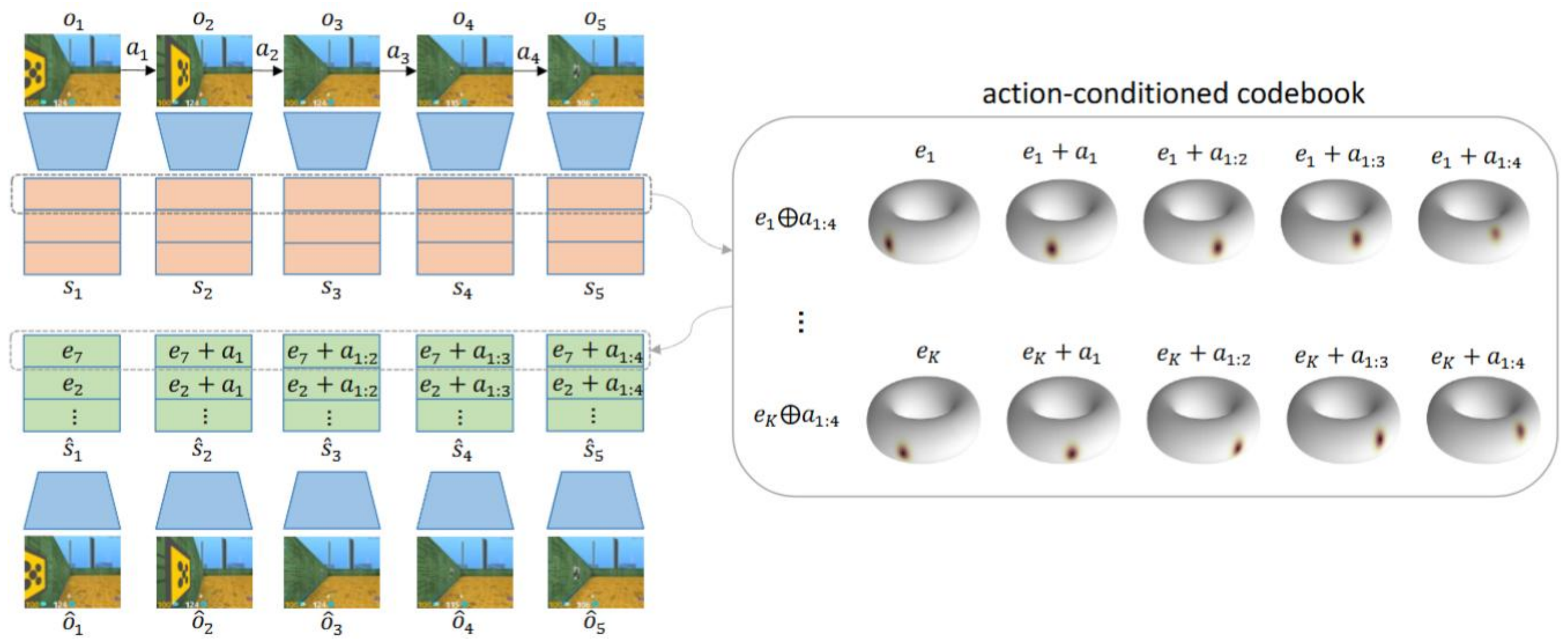
Attractors and template matching



- (A) Schematic of a CANN: Each green dot represents a neuron uniformly distributed on a torus. The neurons receive external input I.
- (B) Energy landscape of CANN dynamics: Each local minimum in the energy landscape corresponds to an attractor state, which manifests as a 2D Gaussian bump on the torus.
- (C) Template matching via CANN dynamics: The CANN inherently performs template matching between the external input I and its attractor states. The input I is matched to the attractor that maximizes their inner product.
- (D) Attractor transition: Under four distinct actions, the attractor initially at position (0,0) stabilizes to four new attractor states.
- (E) Due to the periodic boundary conditions of the CANN, bump movements along the two axes naturally form grid-like patterns.

Grid-like Code Quantization

Action-conditioned codebook and sequence matching



- The action-observation sequence is encoded by the encoder into a latent state composed of m codes. Through sequence template matching with the action-conditioned codebook, the decoder reconstructs the predicted observations.

- **Codeword candidate:** $e_i \oplus a^j_{1:n-1} = \{e_i, e_i + a^j_1, \dots, e_i + a^j_{1:n-1}\}$
- **Best-matching codeword:** $k_j = \arg\min_{i\in\{1,\dots,K\}} ||s^j_{1:n} - (e_i \oplus a^j_{1:n-1})||$

$$\hat{s}^j_{1:n} = e_{k_j} \oplus a^j_{1:n-1}, \quad \text{and} \quad \hat{s}_{1:n} = \{\hat{s}^j_{1:n}\}^m_{j=1}$$

Operations on cognitive map

- GCQ constructs a cognitive map defined by bump dynamics.
- By establishing a mapping between observations and this map, actions in the real space can be projected onto the map to determine position changes, and conversely, movements within the map can be mapped back to real-space actions.
- Greedy step: $s_i \ominus s_j = \arg\min_{a\in\mathcal{A}} |s_j + a - s_i|$

References

[1] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. Advances inneural information processing systems, 30, 2017.

[2] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin ADunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in gridcells. Nature, 602(7895):123–128, 2022.

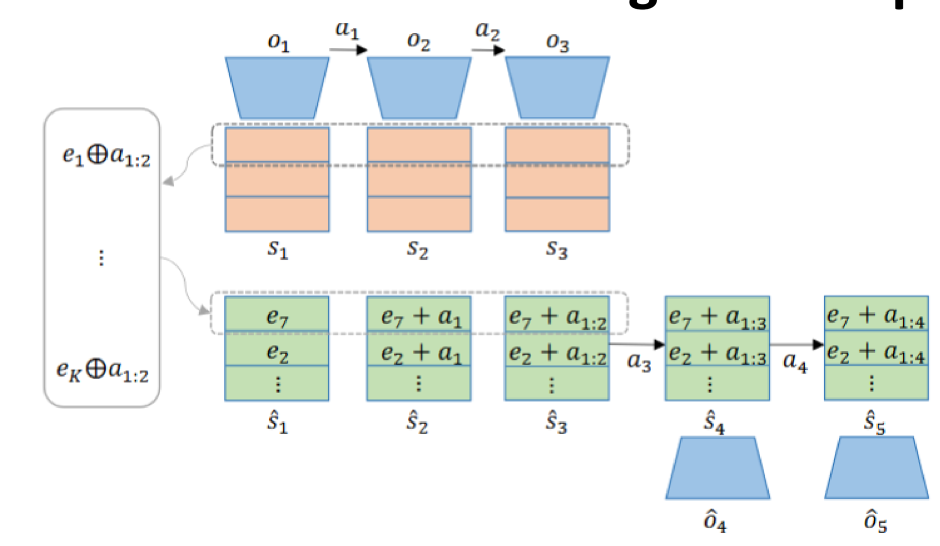
[3] Si Wu, Kosuke Hamaguchi, and Shun-ichi Amari. Dynamics and computation of continuousattractors. Neural Computation, 20(4):994–1025, 2008.

[4] Xingsi Dong, Xiangyuan Peng, and Si Wu. Predictive learning in energy-based models withattractor structures. arXiv preprint arXiv:2501.13997, 2025.

Acknowledgment

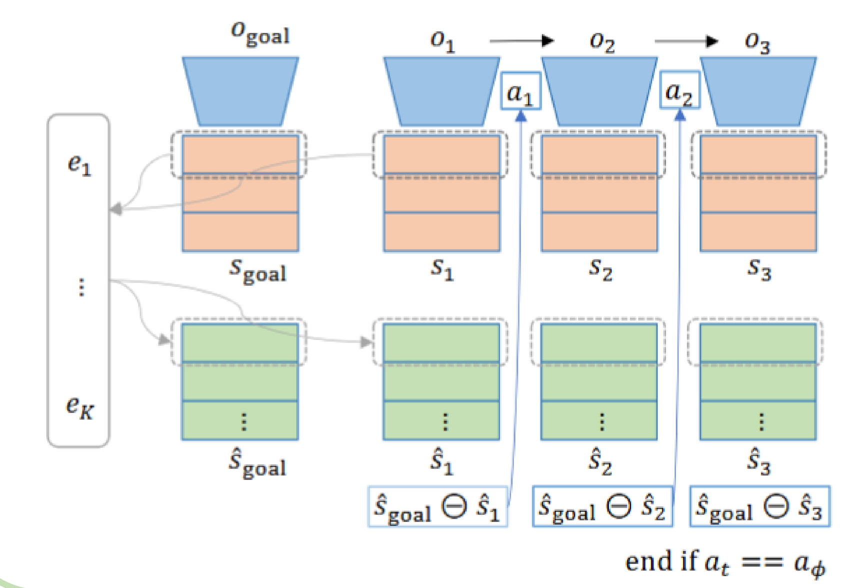
This work was supported by the National Natural Science Foundation of China (no. T2421004 to S.W.), the Science and Technology Innovation 2030-Brain Science and Brain-inspired Intelligence Project (no. 2021ZD0200204, S.W.)..

Long-horizon prediction



- Schematic of long-horizon prediction. The figure illustrates the process of initializing with a sequence of length 3 and predicting two future observations

Goal-directed planning



- Schematic of goal-directed planning. Through iterative action generation, environment interaction, and observation, the agent continues until it outputs a no-op action, indicating that the goal has been reached.

Results