ChatGPT

Predictive Coding with Cyclic Attractors: Integrating Ring and Toroidal Dynamics

Conceptual Background: Predictive Coding and Attractor Networks

Predictive Coding Frameworks – Friston's Approach: Predictive coding (especially in Friston's formulation) casts the brain as a hierarchical generative model that continuously predicts sensory inputs [1]. Higher-level neural states encode hypotheses about causes of sensations, sending predictions downward; lower-level areas compare these predictions to incoming signals and emit *prediction errors* when mismatches occur. Neural dynamics then adjust beliefs to minimize these errors, a process formalized under the Free Energy Principle as *model inversion* or Bayesian inference [1]. In essence, perception and action result from minimizing prediction error by aligning internal predictions with the world, or actively changing the world to meet predictions (active inference). This framework requires explicit coding of error signals and iterative feedback loops between levels of a neural hierarchy.

Attractor Dynamics and Continuous Variables: Attractor networks are recurrent neural circuits that settle into stable patterns of activity, which serve as *memory states* or *representations* of information. A **continuous attractor** maintains a *continuum* of stable states, typically forming a "bump" of neuronal activity that can drift along a low-dimensional manifold without dissipating [2]. Common manifolds include a **ring** (a one-dimensional loop) or a **torus** (a two-dimensional surface with wraparound) for representing periodic or multi-dimensional variables [2]. In the brain, many *continuous variables* – such as spatial orientation, head direction, or spatial location – are believed to be encoded by such attractor dynamics. For example, head-direction (HD) cells in rodents and insects exhibit activity consistent with a **ring attractor**: neurons are arranged such that only one "bump" of activity is active at any angle, and as the animal turns, this bump shifts around the ring, encoding the animal's heading [3] [4]. Similarly, models of grid cells in the entorhinal cortex often assume a 2D continuous attractor (sometimes implemented with toroidal wrap-around) to explain how grid-like firing patterns tile space [2]. In these networks, local recurrent excitation and global inhibition typically stabilize one coherent activity packet (for example, one location or direction) while allowing it to move smoothly with input changes [5].

Bridging Predictive Coding and Attractors: The convergence of these ideas is natural: an attractor network provides an internal *dynamics-based prediction* of a variable's state, and predictive coding supplies the error-driven adjustment mechanism. The free-energy perspective explicitly connects them: *attractor states can be seen as prior beliefs in a generative model*, whereas sensory inputs provide evidence to update those beliefs [6]. In such a view, the stable patterns (e.g. a bump's position on a ring) encode the brain's current prediction of a continuous latent cause (like "current head angle"), and any deviation from expected input produces an error signal. The system then evolves (via recurrent dynamics or synaptic plasticity) to reduce that error, which corresponds to aligning the attractor state with reality [6]. This fusion is essentially a form of **predictive processing on a dynamical manifold** – the brain leverages the intrinsic stability of attractors to carry its predictions, and uses prediction errors to nudge the attractor when surprises occur.

Models Integrating Predictive Coding with Ring/Torus Attractors

Biologically Inspired Models: In neural circuits, ring attractors have been identified or postulated in several "cognitive map" systems, often with an implicit predictive role. A prominent example is the head-direction system: neurons in the rodent dorsal thalamus and entorhinal cortex maintain a persistent representation of heading via a ring attractor network (local excitatory connections between cells with similar preferred direction, and broad inhibitory feedback) [5] [4]. This network integrates angular velocity input to update the heading – effectively performing a **path integration** prediction of orientation. The *generative model* here assumes the heading changes only by continuous rotation; if the animal's actual sensed orientation deviates (e.g. a sudden visual landmark indicating a different direction), that discrepancy serves as a prediction error to correct the internal representation. Models have demonstrated how adding a *calibration mechanism* can align the internal HD bump with external cues: for instance, a landmark or visual cue can inject activity at the "true" angle, creating an attraction basin that pulls the bump to the correct location [7]. This aligns the prediction with reality, analogous to error correction. Recent work by Secer *et al.* (2025) showed that **explicit error-coding neurons** can take this a step further: by encoding the difference between the attractor's estimated state and the true state, these error neurons can drive synaptic plasticity that *recalibrates the attractor's integration gain* [8] [9]. In other words, beyond simply resetting the current state, the network learns to predict better next time. Without an explicit error code, a ring attractor will correct its representation when a cue appears but continue to accumulate drift in the interim; with error feedback, it can adjust its own dynamics to slow down future drift [10] [11]. This is a clear example of integrating predictive coding principles (explicit error signals and learning to minimize long-term surprise) into a continuous attractor circuit.

Another biological case is the grid cell system in the medial entorhinal cortex. Grid cells form a periodic spatial code, and continuous attractor models (often imagined as a toroidal sheet of neurons) can generate grid firing by maintaining multiple bumps or a moving bump in 2D [2]. **Predictive coding** ideas have been applied here too: recent studies suggest grid cells exhibit **predictive shifts** – e.g. firing fields that advance slightly in the direction of travel, effectively encoding a predicted future position [12] [13]. A 2024 model by Tang *et al.* demonstrated that a neural network trained with predictive coding rules (local learning to minimize prediction error) can *self-organize grid-cell representations* [13] [14]. In their framework, a temporal predictive coding network (tPCN) learned to integrate velocity inputs and develop hexagonal grid firing in its latent units, just by trying to predict the agent's location – no hand-crafted attractor dynamics were imposed [14]. Intriguingly, this suggests that **predictive coding itself can give rise to attractor-like representations** (grid patterns) as an emergent strategy for encoding a cyclically continuous variable (space) in a self-consistent way. It extends predictive processing into the domain of hippocampal spatial maps, hinting that the brain's internal model for navigation may naturally form toroidal attractors through error-driven learning [14] [15].

Computational Implementations: On the artificial intelligence and robotics front, researchers have begun implementing these ideas in embodied agents and simulators. One example is the work by Knowles *et al.* (2023), who built a biomimetic navigation system combining **spiking neural network (SNN) ring attractors** with a predictive coding-based memory module [16] [17]. In their model, three coupled ring attractors (representing different directional axes of movement in 2D) integrate the agent's self-motion (idiothetic) cues to keep an internal estimate of its position [17]. Simultaneously, a deep **Predictive Coding Network (PCN)** processes visual and tactile sensory streams to learn a multisensory representation of places the agent has experienced [18] [19]. During navigation, this PCN can recall expected sensory signatures of familiar locations; when the agent revisits a known location (even without landmarks), the recalled sensory pattern is compared against the agent's current internal position estimate. Any discrepancy is used as a *corrective input* to the ring attractors [20] – effectively serving as a prediction error to realign the path-integrated position with the known map. This system demonstrates loop closure (recognizing a previously visited spot and adjusting the internal map) in a way analogous to predictive processing: the generative attractor (path integrator) is corrected by a high-level prediction from memory. Notably, the authors showed that weighting these corrections by the system's *confidence* (i.e. how reliable the cue is, akin to precision in predictive coding) improved performance [21]. The SNN attractor+PCN hybrid was able to navigate 2D environments, and the authors suggest such bio-inspired designs could benefit mobile robotics in conditions where GPS or external sensors are unreliable [22].

Another well-known implementation is **RatSLAM**, a robotic SLAM system inspired by rodent hippocampus. RatSLAM uses a continuous attractor neural network to maintain an estimate of the robot's pose (position and heading) and fuses odometry with visual scene recognition to correct drift. Essentially, the robot continuously "predicts" its position by integrating wheel movements (like an attractor doing path integration), and when the camera recognizes a familiar location, it signals a loop closure – an error between predicted and actual location – to update the map. The attractor network in RatSLAM is a dynamic **pose cell network** arranged in a loop (for heading) and grid (for position) that can represent all possible poses with recurrent activity [23]. When a loop closure occurs, the network activity is shifted or re-centered to the correct pose, and over time the system learns a consistent spatial map. This bears strong resemblance to predictive coding: the internal state provides a running prediction of pose, and sensory recognition events provide prediction errors to adjust that state. In fact, the RatSLAM README explicitly notes it "uses a competitive attractor network to fuse visual and odometry information" and can map environments by *closing loops to correct odometry error* [23]. Such frameworks have been successfully deployed on robots (e.g. the *iRat* wheeled robot) navigating real-world environments, underscoring the practical value of cyclic attractor models with error correction for robust spatial awareness.

Topology and Prediction Error Propagation (Ring vs. Linear vs. Toroidal)

Manifold Topology Matters: The choice of topology for an attractor (e.g. ring vs. line vs. torus) crucially affects how the network handles predictions and errors. A continuous attractor network can be thought of as living on a specific manifold – for a ring attractor, this manifold is circular (0° and 360° are the same point), whereas a line attractor has hard boundaries (ends that do not meet). Likewise, a 2D plane attractor can be "open" (flat with edges) or "toroidal" (opposite edges wrap around). These differences alter how a prediction error manifests and is corrected:

- **Ring vs. Linear Boundary:** In a ring attractor encoding a cyclic variable (like direction or orientation), the representation is *periodic*. This means the network naturally treats, say, 359° and 0° as neighboring states. Prediction errors for a true change that crosses the boundary are small and local – the activity bump can smoothly wrap around the circle. In a linear representation of the same variable (e.g. 0–360° without wrap-around), a shift from 359° to 0° would appear as a huge discontinuity, potentially causing a large error or even failure to represent the change. The ring topology thus avoids artificial edge effects: it **minimizes error at boundaries** because there are none – the space is continuous [2]. Empirically, models of HD cells needed a ring attractor to properly handle the circular nature of angle; using a line of neurons would introduce instability at the ends or require ad hoc connection tricks [5] [4]. In terms of error propagation, a ring attractor localizes any drift errors (e.g. slight misalignment of the bump) – the error is the angular difference, which remains small as long as drift is small. A linear manifold could accumulate offset and hit a boundary, causing a sudden jump (a large prediction error when the bump "falls off" the edge).

- **Toroidal (2D) vs. Planar Attractors:** For 2D continuous variables (like location in an environment, which has two degrees of freedom), many models use toroidal attractors – effectively connecting the edges of a 2D neural sheet – as a simplifying assumption [2]. The torus allows uniform treatment of drift in any direction; there's no edge where the bump would get stuck or dissipate. If the environment is truly unbounded, a toroidal code implies periodic repetition (as seen in grid cell firing patterns which repeat every certain distance). This periodicity can lead to *aliased predictions* if not managed (since the model might predict a location "wrapping around" the torus that corresponds to a different place in reality). However, the brain mitigates this by using multiple scales of grids or context (e.g. place cells) to disambiguate positions. From an error propagation standpoint, a toroidal attractor makes the internal **error dynamics isotropic** – a small error in position is the same no matter where it occurs, because the network looks the same everywhere (no borders). In contrast, an open planar attractor has edges where the bump might behave differently (e.g. require strong rebound forces or else it could fall off the network). With edges, a prediction error pushing the bump outward might dissipate or saturate at the boundary, leading to **systematic errors** near extremes. Thus, a toroidal topology contributes to more uniform and constrained error correction across the represented space.

- **Multiple Attractor Loops:** Sometimes, combining attractors (product spaces) yields a higher-dimensional torus. For example, the **"three-ring" attractor model** for grid cells links three ring attractors at 60° offsets to effectively span 2D movement [24]. The result is akin to a torus comprising the phase of each ring. Topologically, this ensures that as the animal moves in 2D, the combined state of the three rings (which each on their own would repeat periodically) produces a unique grid cell pattern for each location within a certain area before tiling. The important point is that *the topology of the internal model should match the topology of the latent variable* as closely as possible. If the world variable is cyclic (like directions, or time-of-day, etc.), a cyclic generative model prevents spurious errors. If the variable is bounded or linear, using a cyclic model might introduce false wrap-around predictions (which the brain might avoid by segregating contexts or using separate attractor modules).

Error Propagation and Correction: The manifold shape also influences how errors are corrected. In a ring, if the bump is misaligned by a small angle, the error is represented by a small shift – the network can correct by gently moving the bump through local interactions (each neuron "dragging" its neighbors). In a line attractor, a misalignment near the edge might not have enough neighbor neurons to shift into, requiring either a different mechanism or incurring residual error. Similarly, consider a **prediction loop (limit cycle)** – essentially an attractor that cycles through a sequence (e.g. an oscillatory prediction of a periodic event). If that loop is represented on a ring (one full cycle around the ring = one period of the sequence), then a timing error (sequence lag or advance) will appear as the bump being slightly ahead or behind on the ring. Because the loop is closed, a small timing error remains local on the ring and could be corrected by slight phase adjustments. If the sequence were represented linearly (say along a track that resets at the end), a mistiming would, at reset, look like a big jump (end vs. beginning of track), causing a larger error signal at the reset point. Indeed, models of sequential attractors (e.g. *heteroclinic cycles* or ring oscillators) highlight that continuous cyclic representation can gracefully handle temporal prediction – *deviations are error signals that smoothly modulate the cycle's phase*, rather than derailing the sequence [25].

Overall, using the appropriate topology makes predictive processing *more error-tolerant*. It prevents errors from accumulating at representational boundaries and ensures that correction forces (whether via internal dynamics or external cues) can smoothly act *along the manifold* to realign predictions with observations.

Conceptual Implications for Perception, Memory, and Action

Predictive coding with cyclic attractors carries rich implications for various cognitive domains:

- **Perception (Spatial and Sensory Cycles):** The brain often deals with inherently cyclic sensory dimensions – orientation of a visual edge is cyclic (180° wraps to 0°), color hues can be thought of cyclic, and even periodic stimuli (like rhythms) have phase cycles. A ring-attractor-based predictive model could explain how the brain *perceives continuity* in these dimensions. For example, neurons in primary visual cortex are tuned to edge orientation and arranged roughly in a loop (orientation columns), so that 179° and 1° activate neighboring neurons. If the visual scene rotates slowly, the cortical activity bump shifts predictively around this loop. A sudden change (e.g. a jitter in orientation) triggers a localized prediction error, which might be perceived as a salient feature. This could underlie phenomena like the brain's ability to track smooth motion and detect deviations. In a more cognitive sense, **circular inference** might occur in hierarchical perception: e.g. in understanding biological motion or speech prosody, the brain may employ oscillatory priors (like theta or alpha rhythms) as attractors, such that expected patterns repeat cyclically and unexpected breaks (off-beat events) stand out as errors. Notably, head direction perception is clearly circular – we *sense* direction as a seamless 360° space because the underlying neural code is ring-like. Disruption of that attractor (e.g. by caloric vestibular stimulation or disorienting VR cues) yields large prediction errors (dizziness, disorientation) until the network re-stabilizes.

- **Memory and Sequence Learning:** Memory for sequential patterns can be facilitated by loop attractors. Consider working memory for a sequence of items (or a repeated task routine). One way the brain might sustain a sequence is by using a **ring attractor as a looped timeline**: as time progresses, the network's state moves around a ring, encoding the position within the sequence. By the time it returns to the start, one cycle has passed. This is essentially a **neural clock**. If an expected sequence repeats (like days of the week, or the beat in a musical meter), a toroidal or ring representation naturally captures the repeating structure. Deviations – e.g. an extra beat, or a skipped step in a routine – would register as the activity bump being out-of-place on the loop, prompting an error signal. There is also theoretical work suggesting that *heteroclinic cycles* (trajectories through a series of unstable states) or *sequential attractors* in recurrent networks can generate flexible sequences [25]. Predictive coding can interact with this by *coupling sequential attractors across hierarchical levels*: a fast loop (coding immediate transitions) can be guided by a slower loop (coding higher-order context), each predicting the next part of the sequence. For episodic memory, which is often considered a linear timeline, one could imagine fragmenting an experience into cycles (e.g. daily cycles, or narrative arcs) – a toroidal attractor might then link spatial context and temporal context in a cognitive map. In any case, the presence of cyclic dynamics means the system expects a *loop structure* in the experience; if reality diverges (e.g. a story doesn't follow the usual plot loop, or one's daily routine is interrupted), prediction errors ensue, potentially flagging a novel or salient event to remember.

- **Action and Motor Control:** The planning and execution of actions, especially rhythmic or cyclical movements, may leverage ring attractor principles. **Central Pattern Generators (CPGs)** in the spinal cord are circuits that produce rhythmic outputs (for locomotion, chewing, etc.) and can be seen as limit-cycle attractors. Integrating these with predictive coding means treating the CPG's cycle as the generative model for expected proprioceptive and sensory feedback during movement. For instance, when walking, the brain has an intrinsic expectation of the alternating gait cycle; a ring attractor in a motor layer could represent the phase of this gait. As you walk, the phase bump advances around the ring, predicting the timing of footfalls. If you stumble (the footfall timing deviates), a prediction error arises (phase doesn't match), which can quickly correct the cycle (perhaps via a reflex or an active inference adjustment to speed/stride to get back in phase). This explains how we maintain a steady rhythm yet adapt to perturbations. Even for non-rhythmic actions, there are often cyclic components – think of reaching to a target: the hand trajectory might be monitored by an attractor representing the *expected path* (which could be a closed loop in a phase-space of joint angles). Any divergence from the trajectory is an error that the controller tries to nullify (as in smooth pursuit eye movements which have an internal oscillator locking onto moving targets). Moreover, in skill learning, repetitive practice could engrain a movement cycle as an attractor, allowing the motor system to "auto-pilot" the sequence while higher levels predictively modulate it (e.g. playing a musical piece involves internal timing loops; if a beat is missed, error signals cue the player to jump to the correct position in the measure). Thus, cyclic generative models make actions *robust and predictive*: they keep the system on a stable limit cycle and gracefully handle small perturbations by absorbing them (small errors corrected within one cycle) rather than letting them accumulate.

In sum, the marriage of cyclic attractors with predictive coding provides a powerful explanatory lens: perceptual stability in circular feature spaces, reliable memory for repetitive sequences, and smooth execution of rhythmic actions can all be seen as outcomes of the brain encoding *loops within loops*, with deviations quickly flagged by error units. It emphasizes the brain's ability to treat time and space not only linearly but also cyclically when the task demands, ensuring both anticipation and adaptability.

Practical Applications: Robotics, User Interfaces, and Cognitive Mapping

Beyond theoretical neuroscience, these concepts are inspiring various applications in technology and AI:

- **Robotics and Navigation:** As mentioned, robotics has directly borrowed the idea of ring and toroidal attractors with predictive correction for navigation. The biomimetic SNN ring attractor system by Knowles *et al.* is aimed at mobile robots operating with limited sensors [22]. Such a robot can internally integrate its movement (just like an animal performing path integration) using an attractor network to maintain a belief of its position and orientation. When an occasional reliable sensory cue (camera image or touch sensor reading) comes in, it's fed into a predictive coding module that compares it with expected sensory input for the robot's current estimated location. A difference indicates the robot might be off-track, prompting a correction to its internal map [20]. This approach is valuable in autonomous drones, planetary rovers, or underwater vehicles where continuous external reference (GPS, magnetic compass) might be unavailable or sporadic. Indeed, the system showed resilience to missing or even extraneous sensory inputs by relying on multi-sensory predictive calibration [26]. Another practical implementation, RatSLAM, was used in robots to perform **SLAM (Simultaneous Localization and Mapping)** in real-time. RatSLAM's internal pose attractor (a toroidal network of "pose cells") fuses wheel odometry (predicting pose change) with visual place recognition events (detecting loop closures) [23]. This enables the robot to map large areas with very low drift, demonstrating that bio-inspired **error-correcting continuous attractors** can outperform naive dead-reckoning. The concept is also being extended to **drone navigation**, where 3D orientation must be tracked (which can be represented as attractors on a sphere or torus) in concert with position maps – here too, actively balancing internal gyro integration with occasional absolute fixes (like recognizing a known landmark) follows the predictive coding philosophy.

- **User Interfaces and VR/AR:** In human-computer interaction, predictive models with cyclic attractors can enhance responsiveness and stability. One key example is in **virtual reality headsets** or **augmented reality**: tracking the user's head orientation and position is critical. VR systems often internally integrate inertial sensor data at high rates to predict head movement, then correct with slower camera-based tracking – conceptually very similar to a head-direction ring attractor with visual error correction. Using an attractor-like integrator ensures the orientation estimate is smooth and has continuity (no jumps when you spin 360°), while periodic re-calibration prevents drift. Predictive coding principles suggest weighting the correction by certainty – e.g. if the visual tracking confidence is low (dim light), rely more on the internal gyro (prior); if it's high, rapidly correct the internal model. This can reduce latency and jitter in interfaces, providing a more stable user experience. More generally, any interface involving cyclic control or feedback could benefit. Consider a **circular menu or dial** that a user controls (say with a rotary encoder): a software agent could use a ring attractor to model the dial's state and predict the user's turning behavior (many users have rhythmic patterns when scrolling or adjusting knobs). The UI could then pre-fetch or pre-scroll content in anticipation, only adjusting if the user deviates – effectively a form of UI predictive processing. In brain-computer interfaces (BCIs), if one is decoding a rhythmic motor intention (e.g. imagined tapping or cycling through options by attention), incorporating a cyclic generative model could make the decoding more robust. The BCI would expect the signal to follow an attractor trajectory and flag any deviations as significant (improving signal-to-noise for detecting intentional changes). While these specific applications are nascent, the principle is clear: **predict and assist** user actions that have repetitive structure.

- **Cognitive Mapping and AI Representations:** Cognitive maps aren't only spatial – they can be abstract (maps of concepts, task states, etc.). In AI, representing states of the world in a continuous attractor can allow an agent to *plan and predict* outcomes efficiently. For example, an AI assistant managing a user's schedule might have an internal ring attractor for the 24-hour day cycle, learning the user's daily routine. It can then predict when the user usually performs certain tasks and proactively assist (with reminders or automation), but if one day the user's behavior deviates (prediction error: e.g. skipping the usual morning coffee), the system can flag this and adjust its suggestions (maybe it infers the user is out of coffee and suggests ordering more). In autonomous driving, one could imagine an attractor in the system representing the car's expected trajectory around a curve – if the car starts to skid (deviate from the curve's attractor), the system immediately detects the error and can intervene faster than a linear model that doesn't "expect" the cyclic steering pattern. Additionally, continuous attractors with toroidal topology are useful in **modular robotics or animation**, where multiple joint angles must be coordinated cyclically (e.g. bipedal walking robots). Researchers use *Central Pattern Generator networks* (often rings of neurons) to generate gaits; integrating those with predictive coding means the robot can adjust its gait on the fly if it senses a slip or a load change – effectively fusing the feedforward rhythm with feedback corrections. The attractor ensures the robot's movements remain rhythmic and energy-efficient, while predictive error correction ensures adaptability to terrain.

In summary, applying predictive coding with ring/toroidal attractors gives engineered systems a taste of biological elegance: a strong internal model (with cyclic continuity) that **keeps track of state** even with sparse input, combined with an error-driven correction mechanism that **anchors the model to reality** when needed. This paradigm is proving valuable for robust navigation algorithms [23], smoother and smarter user experiences, and even new learning algorithms in AI that capture the periodicity or continuity in data. As research continues, we are likely to see more cross-pollination between neuroscience and technology in this arena – for instance, code repositories and simulators are emerging where one can experiment with **predictive coding networks that have attractor dynamics** (e.g. the interactive demos of self-organizing attractors under the Free Energy Principle [27] [28], or open-source PCN implementations for sequence learning [29] [30]). These tools will help design systems that *learn* the appropriate topology for the task (linear vs cyclic), possibly even reconfiguring their own attractor landscape as they adapt – a level of flexibility that would truly mirror the brain's architecture.

Sources:

- Friston, K. (2009). *Predictive coding under the free-energy principle.* Phil. Trans. R. Soc. B, **364**(1521), 1211–1221 [1].
- Secer, G., Knierim, J.J., & Cowan, N.J. (2025). *Explicit error coding can mediate gain recalibration in continuous bump attractor networks.* Nat. Commun. **16**, 9215 [8] [9].
- Knowles, T.C. *et al.* (2023). *Ring Attractors as the Basis of a Biomimetic Navigation System.* Biomimetics **8**(5):399 [16] [17].
- Tang, M., Barron, H., & Bogacz, R. (2024). *Learning grid cells by predictive coding.* arXiv:2410.01022 [13] [14].
- Frontiers (2025). *Population coding and self-organized ring attractors in RNNs for continuous variable integration.* (demonstrates self-organized ring attractor in a trained network) [31] [32].
- Noorman, M. *et al.* (2024). *Maintaining and updating accurate internal representations of continuous variables with a handful of neurons.* Nat. Neurosci. **27**, 2207–2217 [5] [4].
- Rao, R.P.N. & Ballard, D. (1999). *Predictive coding in the visual cortex:* Nat. Neurosci. **2**(1), 79–87. (classical PCN model for perception)
- Milford, M. & Wyeth, G. (2008). *Mapping a suburb with a single camera using a biologically inspired SLAM system.* IEEE Trans. Robotics **24**(5), 1038–1053. (RatSLAM; continuous attractor + loop closure) [23].

1 Predictive coding under the free-energy principle - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC2666703/

2 7 8 9 10 11 Explicit error coding can mediate gain recalibration in continuous bump attractor networks | Nature Communications
https://www.nature.com/articles/s41467-025-63817-0?error=cookies_not_supported&code=455d5fc3-6d63-497b-9842-110f80e4dc9d

3 6 25 27 28 Self-orthogonalizing attractor neural networks emerging from the free energy principle
https://arxiv.org/html/2505.22749v1

4 5 Maintaining and updating accurate internal representations of continuous variables with a handful of neurons | Nature Neuroscience
https://www.nature.com/articles/s41593-024-01766-5?error=cookies_not_supported&code=b1ffe25e-e2ba-420c-85ba-be0513943c64

12 13 14 15 Learning grid cells by predictive coding
https://arxiv.org/html/2410.01022v2

16 17 18 19 20 21 22 24 26 Ring Attractors as the Basis of a Biomimetic Navigation System | MDPI
https://www.mdpi.com/2313-7673/8/5/399

23 GitHub - davidmball/ratslam: Automatically exported from code.google.com/p/ratslam
https://github.com/davidmball/ratslam

29 30 Dynamic predictive coding: A model of hierarchical sequence learning and prediction in the neocortex | PLOS Computational Biology
https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1011801

31 32 Frontiers | Population coding and self-organized ring attractors in recurrent neural networks for continuous variable integration
https://www.frontiersin.org/journals/network-physiology/articles/10.3389/fnetp.2025.1693772/full