**Lifelong Learning and Memory Consolidation for TORI: An Interdisciplinary Roadmap**

**Introduction**

TORI’s vision of a **lifelong-learning cognitive engine** requires a memory architecture that integrates insights from physics, mathematics, and biology. Unlike stateless AI models, TORI (via the ALAN 2.x engine) maintains a persistent **Large Concept Network (LCN)** with dynamic “ψ-phase” states – oscillatory activations that capture concept relationships over time. Achieving true lifelong learning means new knowledge must be continually integrated (like the hippocampus rapidly storing episodes) while preserving global stability (like the neocortex gradually consolidating schemas). This report surveys key techniques at the intersection of energy-based physics models, advanced mathematical frameworks, and neurobiological memory mechanisms, and outlines how to fuse them into TORI’s memory system. We emphasize implementation-ready approaches (≈70%) alongside frontier theory (≈30%), focusing on:

* **Energy-Based Consolidation:** Using energy-based models (EBMs), simulated annealing, or free-energy minimization to drive memory replay and concept selection.
* **Spectral/Koopman Continual Learning:** Applying Koopman operator theory and sparse optimization to capture evolving dynamics (the “ψ-phase” spectral modes) in a stable, continual learner.
* **Category-Theoretic Knowledge Integration:** Structuring the LCN as category-typed graphs (with typed morphisms) to enable robust symbolic–subsymbolic integration and memory pruning.
* **Biologically Inspired Mechanisms:** Incorporating hippocampus→neocortex style consolidation, synaptic pruning of unused connections, and replay during “offline” periods (analogous to sleep) for system-level memory health.

We then provide: (1) a comparative matrix evaluating candidate techniques on **stability, memory retention, ψ-phase compatibility, and engineering effort**; (2) designs for a **Reference Implementation Kit** (including an energy-based replay consolidator, a Koopman-based continual learner with sparsity, and a category-typed concept merge utility); (3) a UML-style integration map situating these components in ALAN’s memory/planning loop; and (4) a pilot benchmark suite with datasets and KPIs to guide development. The goal is to ground TORI’s roadmap in solid scientific principles so that it “grows with you, for the rest of your life” – a true lifelong learning partner.

**Interdisciplinary Foundations of Memory Consolidation**

**Physics-Inspired Energy-Based Memory Models**

**Energy-Based Models (EBMs)** provide a principled way to encode memories as **attractor states** in an energy landscape. Memories correspond to low-energy basins, and recall/replay can be performed by descending energy (e.g. via gradient descent or simulated annealing) to settle into a stored state. This paradigm is analogous to a Hopfield network’s associative memory, where each memory is a stable pattern (local minimum of an energy function)[en.wikipedia.org](https://en.wikipedia.org/wiki/Hopfield_network#:~:text=A%20Hopfield%20network%20,addressable%20memory). Modern approaches have significantly extended classical Hopfield networks: for instance, *continuous Hopfield networks* with high capacity have been shown to connect to attention mechanisms in transformers (treating attention updates as Hopfield retrieval)[ml-jku.github.io](https://ml-jku.github.io/hopfield-layers/#:~:text=Hopfield%20Networks%20is%20All%20You,In%20classical%20Hopfield). Recent research by Alonso & Krichmar (2024) introduced a **Sparse Quantized Hopfield Network (SQHN)** that learns online via local, Hebbian-like updates, yielding **state-of-the-art associative memory performance in continual learning settings** with non-i.i.d. streaming data[nature.com](https://www.nature.com/articles/s41467-024-46976-4#:~:text=approach%20to%20this%20problem%20has,on%20an%20episodic%20memory%20task). The SQHN’s discrete graphical model and online MAP learning rule allow it to outperform deep networks on noisy sequential memory tasks, highlighting the promise of energy-based associative memory for lifelong learning[nature.com](https://www.nature.com/articles/s41467-024-46976-4#:~:text=approach%20to%20this%20problem%20has,on%20an%20episodic%20memory%20task).

**Energy-based continual learning:** Instead of using explicit memory buffers or regularization tricks, one can change the learning objective itself to reduce forgetting. Li *et al.* (2022) propose reframing classification as training an unnormalized energy model, using a **wake-sleep** paradigm for updates[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf#:~:text=learning%20performance%20in%20classification%20problems,new%20input%2C%20creating%20updates%20that)[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf#:~:text=lead%20to%20catastrophic%20forgetting,gain%20based%20on%20the%20class). In their approach, learning a new class involves *decreasing the energy* of correct outputs and *increasing the energy* of selected incorrect outputs, rather than adjusting a normalized softmax across all classes (which would inadvertently raise energies of all old classes and cause forgetting)[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf" \l ":~:text=truth%20label%20is%20decreased%20while,the%20model%20to%20select%20and" \t "_blank). This selective update is akin to focusing on *what portion of the energy landscape to reshape*, thereby preserving previous memory basins. Singh *et al.* (2024) extend this with a **biologically inspired wake-sleep EBM training regime**: a short “wake” phase where the model learns the new input (minimizing free energy for ground-truth labels), followed by a longer “sleep” phase that **contrastively minimizes the free energy of the whole system** (globally adjusting energies, which consolidates knowledge)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=remains%20a%20challenge,phase%2C%20where%20the%20free%20energy)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=by%20long%20sleep%20phases,Our%20empirical%20evaluation). This two-phase update mimics how **sleep consolidates memories in humans** – active learning while awake, then global re-balancing during rest[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=remains%20a%20challenge,model%20towards%20the%20correct%20solutions). Notably, EBMs inherently offer **longer retention of prior knowledge** compared to standard neural nets, and the wake-sleep training further improves stability and performance[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=learning,the%20short%20wake%20phase%2C%20the)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=of%20the%20whole%20system%20is,Furthermore%2C%20our%20proposed). Such energy-based consolidation mechanisms could be directly leveraged in TORI: e.g., an *Energy-Based Replay Consolidator* process (detailed later) can simulate a “sleep” phase for the LCN, periodically performing global free-energy optimization to rehearse and solidify important memories while marginalizing or forgetting inconsistent ones. Simulated annealing techniques may aid this process by finding deep minima (stable states) in the concept network’s energy landscape for robust long-term memory storage.

**Free-energy principle and predictive coding:** Another physics-inspired insight comes from Karl Friston’s **Free Energy Principle**, which posits that brains minimize a variational free-energy bound on surprise. In TORI’s context, one could treat memory consolidation as a free-energy minimization problem: the system generates an internal model (conceptual schema) that predicts incoming information, and consolidation adjusts the model to reduce prediction errors (free energy). This aligns with **predictive coding** frameworks where memory recall and imagining future scenarios are generative processes that refine the internal model[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=Episodic%20memories%20are%20,based%20distortions%20including%20boundary)[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=Hippocampal%20%E2%80%98replay%E2%80%99%20of%20patterns%20of,abstract%20representation%2C%20a%20process%20sometimes). By integrating a free-energy optimizer, TORI could autonomously strengthen memories that improve prediction (useful, frequent patterns) and prune those that do not, achieving a self-organizing balance between plasticity and stability.

**Mathematical Frameworks for Continual Learning and Knowledge Organization**

**Koopman Operators and Spectral Memory** – TORI’s use of spectral methods (the “ψ-phase” eigenfunctionsfile-7msf4dlgfjylvyotqb8xbg) naturally aligns with **Koopman operator theory**. The Koopman operator provides a linear, global view of nonlinear dynamics by operating on a high-dimensional space of observable functions. For a cognitive system, one can imagine each concept’s state evolving in time; the Koopman eigenfunctions (ψ-modes) capture recurring **dynamical patterns of thought** or concept activation. Integrating Koopman learning allows the system to detect and preserve these patterns as it learns continually. *Koopman Learning with Episodic Memory* (Redman *et al.*, 2025) is a recent advance that explicitly addresses continual learning: it **stores spectral components (eigenfunctions, modes) from past temporal windows** and reuses them when similar dynamics reoccur[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=becomes%20clear%20when%20considering%20applying,implicit%20forgetting%20of%20the%20past)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=Inspired%20by%20the%20integration%20of,with%20episodic%20memory%20we%20develop). In other words, the system remembers dynamics from previous “episodes” and recalls them instead of relearning from scratch. A specific implementation, **Extended Dynamic Mode Decomposition (EDMD) with memory**, showed significant improvements in predicting non-stationary time-series by **not repeatedly making the same mistakes** – the episodic memory of past spectral patterns provides global context to new observations[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=approximations%20of%20the%20Koopman%20operator%2C,with%20memory%2C%20sees%20large)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=the%20general%20framework%20we%20introduce,to%20better%20applications%20of%20Koopman). For TORI, a **Koopman Continual Learner** could maintain a sparse basis of ψ-eigenfunctions that represent stable concept oscillation modes. As new concepts and interactions arise, it would update this basis *incrementally*, perhaps adding new eigenfunctions for novel patterns while using **sparse optimization to prune** redundant or low-utility modes. Sparse spectral learning (related to techniques like SINDy or L1-regularized DMD) would enforce an economy of memory: only a compact set of significant oscillatory patterns are retained, aligning with the need to prevent unbounded growth of memory. This component directly supports TORI’s ψ-phase representation – since ψ functions *are* Koopman eigenfunctions, any technique that learns and updates these eigenfunctions is inherently ψ-phase compatible.

**Category Theory for Knowledge Graphs** – Category theory provides a unifying mathematical language for structure, making it well-suited to **symbolic knowledge graphs with typed relations**. In TORI’s LCN, concepts have types and relationships (e.g., “function A calls function B” is a relation of a certain type in the code ontology). By viewing the LCN as a **category** (or as a functor into **C-Set** – the category of sets, which is a common formalism for labeled graphs), we gain powerful tools for integration and transformation. For example, Aguinaldo *et al.* (2023) propose a representation of world states using category-theoretic constructs like **C-sets and double-pushout (DPO) rewriting**, which **formalize an ontology’s semantics and preserve those semantics across state updates**[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=concise%20action%20models,This). In their framework, knowledge graphs and databases are given a formal categorical semantics, so that when the world state is updated (via a pushout operation), the **typed relationships and constraints remain consistent**[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=during%20planning,states%20and%20up%02dates%20in%20planning). Applying this to TORI, we could ensure that when new knowledge is merged into the LCN or two concepts are unified, the operation is *structure-preserving*: a **Category-Typed Concept Merge** utility (described later) would use pushout colimits or other categorical merge operations to **safely integrate or compress concept graphs without violating type rules**. Category theory also supports **morphisms** that map one knowledge schema to another – useful for connecting TORI’s internal ontology with external knowledge bases or user ontologies, all while maintaining consistent type mappings. In summary, category-theoretic tools can provide *mathematically guaranteed coherence* when evolving TORI’s symbolic memory structures, complementing the subsymbolic spectral techniques.

**Sparse Optimization and Memory Pruning** – In both neural networks and symbolic graphs, **sparsity** is key to efficient lifelong learning. Biological brains famously **prune synapses**: during development and sleep, weaker or unused connections are eliminated, which refines and streamlines neural circuits. We can mimic this through sparse optimization techniques. In neural continual learning, methods like **Elastic Weight Consolidation (EWC)** and **Synaptic Intelligence (SI)** compute the importance of weights for past tasks and add penalties or damping on important weights[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf#:~:text=usage%20of%20external%20memory%20%28Lopez,ap%02proaches%20tackle%20these%20tasks%20by), effectively allowing insignificant weights to drift (or even be pruned) while “locking in” the important connections. This is analogous to synaptic consolidation in neuroscience, where important synapses are stabilized by molecular mechanisms. Another angle is direct weight pruning or regularization: applying L1 regularization or periodic pruning to encourage the network to drop redundant connections. Recent theoretical work (Shaham *et al.*, 2022) combines Hebbian learning with **synaptic decay** (a form of uniform weight weakening) plus a novel **stochastic consolidation via replay**, resulting in a network that retains memories far longer than would be expected by decay alone[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=maintaining%20old%20memories%20for%20a,The%20number%20of%20retrievable)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits). In their model, memories undergo **stochastic rehearsals (replays) with a frequency proportional to the memory’s basin stability**, which leads to **self-reinforcing consolidation** of strong memories[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits). This yields a gracefully decaying retention curve (older memories fade slowly instead of catastrophic drop) and memory lifetimes much extended relative to raw decay constants[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=decay%20and%20a%20novel%20memory,memory%20impairments%20following%20neurological%20trauma). These insights suggest TORI should incorporate both **use-driven reinforcement** (replaying important memories more often) and **automatic sparsification** (weakening or removing little-used links). Engineering-wise, this could mean a background process that monitors concept node/edge “activation frequency” (or phase coherence) and prunes those that fall below a threshold, similar to how the brain might eliminate rarely fired synapses. Sparse coding techniques (e.g., maintaining sparse activations in the concept network) could also improve ψ-phase stability by reducing overlapping interference between concepts – each concept or memory should ideally have a distinct, sparse signature in the network.

**Biological Memory Mechanisms and Cognitive Architectures**

**Complementary Memory Systems (Hippocampus ↔ Neocortex)** – Neuroscience has long distinguished between fast, temporary memory storage in the **hippocampus** and slow, long-term storage in the **neocortex**. New episodic memories are quickly encoded in hippocampal circuits (within *one shot* or a few exposures), but over time (particularly during sleep) these memories are **replayed and gradually taught to the neocortex**, which integrates them into semantic schemas[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=model%20in%20which%20hippocampal%20replay,the%20use%20of%20limited%20hippocampal)[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=Hippocampal%20%E2%80%98replay%E2%80%99%20of%20patterns%20of,abstract%20representation%2C%20a%20process%20sometimes). This complementary learning system (CLS) avoids catastrophic interference by separating the concerns: the hippocampus can rapidly learn specifics without overwriting the cortex’s structured knowledge, and the cortex slowly generalizes those specifics into stable abstractions[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=Episodic%20memory%20concerns%20autobiographical%20experiences,re%29construction%20of). TORI’s architecture can mirror this. We envision a **dual-memory module**: a *fast “working” memory* (ψₕ, analogous to hippocampus) where new concepts or contexts are first encoded, and a *slow “long-term” memory* (ψₙ, analogous to neocortex) in the LCN where knowledge is accumulated. When TORI encounters a novel problem or user query, it would create rapidly formed structures in the fast memory (e.g., a transient subgraph of concepts with high plasticity). Then, during consolidation cycles (perhaps during idle times or low-load periods), TORI replays those recent memory patterns – e.g. reactivating the concept oscillator patterns representing the new knowledge – and gradually **integrates them into the main LCN** by adjusting connections or creating new generalized nodes. This process aligns with models like Spens & Burgess (2024), who implemented a hippocampal-neocortical system using an **autoassociative memory (Hopfield-like) in the hippocampus and generative models (VAEs) in the cortex**[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=model%20in%20which%20hippocampal%20replay,the%20use%20of%20limited%20hippocampal). In their model, **hippocampal replay trains the cortical generative model**, enabling the reconstruction of full experiences from partial cues and explaining phenomena like semantic abstraction and imagination[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=model%20in%20which%20hippocampal%20replay,the%20use%20of%20limited%20hippocampal)[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=during%20hippocampal%20replay%20,research%20into%20the%20relationship%20between). In TORI, the **ψ-phase oscillator network** could serve as the autoassociative hippocampal memory – storing new concept constellations as phase-locked patterns – and a generative model over concept embeddings could play the role of cortical schema learning, gradually absorbing the replayed patterns. The end effect is that **specific episodic traces become generalized conceptual knowledge**, supporting both retention and inference (much as replay-induced schema extraction in brains leads to general knowledge with some loss of detail[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=during%20hippocampal%20replay%20,research%20into%20the%20relationship%20between)). Recent computational neuroscience work also supports having **dual forms of memory to prevent forgetting**: Shi *et al.* (2025) demonstrate a *cortico-hippocampal hybrid network* that uses a standard ANN (for generalized memory) plus a spiking neural network (for specific memory) – this **dual system (CH-HNN)** dramatically mitigates catastrophic forgetting in both task and class incremental learning by leveraging prior knowledge when learning new concepts, akin to how the brain differentiates specific vs generalized memory traces[nature.com](https://www.nature.com/articles/s41467-025-56405-9#:~:text=Current%20artificial%20systems%20suffer%20from,neural%20functions%20of%20both%20feedforward). TORI could incorporate a similar hybrid, perhaps using a spiking or pattern-based module for short-term concept traces and a deep net or structured graph for long-term memory.

**Synaptic Pruning and Regularization** – The brain’s developmental pruning and nightly renormalization (the “synaptic homeostasis” hypothesis of sleep) suggest that **periodic cleanup is essential for lifelong stability**. Without pruning, a learning system will saturate or destabilize (just as a child with epilepsy may lack proper synaptic pruning). TORI should include a **memory pruning daemon** that runs during consolidation phases: it could remove low-weight edges in the concept graph, merge duplicate or highly similar concepts (to avoid proliferation of redundant nodes), and decay connection strengths slightly to allow new information to redistribute importance (analogous to “synaptic downscaling” during sleep). The **ψ-SyncMonitor** in TORI (a hypothetical monitoring rule base that watches oscillator synchrony) could flag candidates for pruning or consolidation: e.g., if a concept node has not phase-aligned with any others in a long time (indicating it’s isolated or irrelevant), the system might review and possibly archive or remove it. Conversely, if two concept nodes are almost always in sync, ψ-SyncMonitor might suggest **merging them into one concept** (they might represent the same underlying idea discovered via different paths). In this way, TORI’s use of **phase synchrony as a signal for coherence**file-7msf4dlgfjylvyotqb8xbg directly informs a biological-like pruning/merging strategy: synchronous = integrate (strengthen or fuse), asynchronous = segregate (weaken or eliminate inconsistent linkages). This is in line with Hebb’s rule (“cells that fire together, wire together”): coherence in ψ-phase would lead to wiring together in the LCN, whereas dissonance leads to unwiring.

**Replay and “Sleep” Mechanisms** – Empirical studies of sleep show that the hippocampus “replays” daily experiences (in compressed form) during deep sleep (sharp-wave ripples), believed to drive consolidation. TORI can implement an **offline replay mechanism** we term the *Energy-Based Replay Consolidator*. During idle times, TORI can sample recent high-importance memories (or even stochastic samples proportional to each memory’s current stability[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits)) and replay them through the reasoning core. Because TORI’s core is oscillator-based, replay could mean re-imposing the phase pattern corresponding to a past episode and letting the system “reason” on it briefly again. This would not produce external output, but internally it allows the concept network to re-experience and thus reinforce that memory trace. We can draw inspiration from **generative replay** in machine learning, where instead of storing actual data, the model re-generates samples from old tasks to intermix with new training data (Shin et al. 2017’s DGR). In TORI’s case, the generative aspect could be the imagination of a prior context via its concept graph. By **integrating replay with an energy minimization objective** (as in wake-sleep EBMs or the stochastic rehearsal model[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits)), TORI ensures that each replay moves the network toward a *stable basin* that represents a consolidated memory. The “sleep” cycle might involve many such replays (diverse or sampled based on uncertainty) and global adjustments to synaptic weights or concept link strengths to settle the network into an equilibrium of knowledge.

In summary, these interdisciplinary foundations guide an architecture where: **(a)** Memories are stored and recalled as **energy minima / phase-locked patterns** (physics EBMs, Hopfield attractors); **(b)** The system continually **identifies spectral patterns** and retains a sparse basis of them (Koopman eigenfunctions for dynamics); **(c)** Knowledge is structured in a **typed graph** that can grow and merge without chaos (category-theoretic consistency); and **(d)** The overall system emulates the brain’s separation of fast and slow learning, using **replay, stability metrics, and pruning** to achieve both adaptability and long-term retention.

**Comparative Matrix of Techniques**

To evaluate various systems and techniques for TORI’s lifelong learning needs, we compare a selection of approaches across key criteria: **Stability** (resistance to forgetting or chaotic drift), **Memory Retention** (ability to preserve knowledge over time/tasks), **ψ-Phase Compatibility** (how naturally it integrates with TORI’s phase-based reasoning and concept eigenfunctions), and **Engineering Lift** (practical implementation complexity and computational cost). **Table 1** provides a summary (scoring each aspect qualitatively as High ✓✓, Medium ✓, or Low – for brevity):

| **Technique/System** | **Stability** | **Memory Retention** | **ψ-Phase Compatibility** | **Engineering Effort** |
| --- | --- | --- | --- | --- |
| **Hopfield Network (Associative Memory)** | ✓✓ (attractors provide robust recall)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=decay%20and%20a%20novel%20memory,memory%20impairments%20following%20neurological%20trauma) | ✓✓ (memorizes patterns indefinitely if capacity not exceeded) | ✓ (phase patterns can represent attractors; possible to map phases to energy minima) | ✓ (classic algorithms exist; modern continuous Hopfield requires tuning) |
| **Boltzmann Machine / Energy-Based Model** | ✓✓ (global energy minima = stable states)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=by%20long%20sleep%20phases,Our%20empirical%20evaluation) | ✓ (can retain multiple modes; some risk of mode collapse if not managed) | ✓ (free-energy optimization aligns with phase alignment goals) | – (training Boltzmann is high cost; EBMs require careful sampling) |
| **Wake-Sleep EBM Training** | ✓✓ (separates fast vs slow updates)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=remains%20a%20challenge,phase%2C%20where%20the%20free%20energy) | ✓✓ (demonstrated superior retention on class-IL)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=of%20the%20whole%20system%20is,Furthermore%2C%20our%20proposed) | ✓ (uses free-energy, can be applied to phase updates) | ✓ (added complexity in scheduling wake/sleep cycles, but biologically grounded) |
| **Koopman EDMD w/ Episodic Memory** | ✓✓ (the linearity of Koopman yields predictable updates)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=Inspired%20by%20the%20integration%20of,with%20episodic%20memory%20we%20develop) | ✓✓ (remembers past spectral patterns, avoiding relearning)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=the%20general%20framework%20we%20introduce,to%20better%20applications%20of%20Koopman) | ✓✓ (TORI already uses Koopman eigenfunctions as ψ)file-7msf4dlgfjylvyotqb8xbg | ✓ (requires spectral decomposition; moderate effort with existing libraries) |
| **Sparse Koopman (Sparse DMD/SINDy)** | ✓✓ (prunes unstable modes, focusing on invariants) | ✓ (fewer modes = less overfitting, but risk losing minor patterns) | ✓✓ (eigenfunctions directly inform concept phase coupling) | ✓ (convex optimization for sparsity; manageable) |
| **Category-Theoretic Graph Merging** | ✓✓ (guarantees consistency via pushouts)[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=concise%20action%20models,This) | ✓ (prevents knowledge base corruption; retention depends on manual merge policy) | ✓ (operates on structure, orthogonal to phase dynamics) | ✓ (leverages existing category libraries, but needs custom ontology design) |
| **Double-Pushout Graph Rewriting** | ✓ (formal approach ensures valid updates)[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=during%20planning,states%20and%20up%02dates%20in%20planning) | ✓ (can maintain history of changes for rollback) | ✓ (not inherently phase-based, but can trigger phase re-sync after structural changes) | – (category tools still niche; higher learning curve) |
| **Elastic Weight Consolidation (EWC)** | ✓ (mathematically bounds drift on important weights) | ✓ (mitigates forgetting for moderate task sequences) | – (works at weight level, not phase; could indirectly stabilize phases by preserving critical connections) | ✓✓ (simple to add to training loops; well-tested) |
| **Synaptic Intelligence (online reg.)** | ✓ (online tracking of weight importance) | ✓ (similar benefits to EWC in preventing forget) | – (weight-level method, phase compatibility indirect) | ✓ (lightweight regularization) |
| **Generative Replay (DGR, etc.)** | ✓ (prevents forgetting by re-introducing old samples) | ✓✓ (in ideal case, old performance fully preserved) | ✓ (if replayed content is integrated as concept phase patterns) | – (training generative model is heavy; storage or compute cost for replay) |
| **Experience Replay (buffer)** | ✓ (good short-term retention; less effective long-term without limits) | ✓ (retains raw past data; memory retention high until buffer full) | – (raw data not in concept form; conversion to phase patterns needed) | ✓ (straightforward to implement buffer; but scalability issues) |
| **Complementary Learning (Hpc–Cx)** | ✓✓ (the dual-system avoids interference)[nature.com](https://www.nature.com/articles/s41467-025-56405-9#:~:text=Current%20artificial%20systems%20suffer%20from,neural%20functions%20of%20both%20feedforward) | ✓✓ (fast uptake + slow consolidation = both plastic and stable) | ✓✓ (maps naturally: fast oscillatory store + stable LCN) | ✓ (architectural complexity in building two subsystems, but conceptually clear) |
| **Oscillatory Synchrony Detection** | ✓ (flags instabilities via phase conflicts)file-7msf4dlgfjylvyotqb8xbg | ✓ (can trigger consolidation when incoherence detected) | ✓✓ (core to TORI’s reasoning paradigm) | ✓ (already part of ALAN’s Banksy core; just extend rules) |
| **Synaptic Pruning (network sparsification)** | ✓ (pruning improves stability if done carefully) | ✓ (removes spurious memories, though over-pruning can harm retention) | ✓ (pruning weak phase links likely increases overall phase coherence by removing noise) | ✓ (many techniques available; need to tune frequency/threshold) |
| **Stochastic Replay Consolidation** | ✓✓ (self-amplifies stable memories via randomness)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits) | ✓✓ (achieves very long lifetimes of memories)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=decay%20and%20a%20novel%20memory,memory%20impairments%20following%20neurological%20trauma) | ✓ (concept of basin stability could translate to phase stability metrics) | ✓ (requires ability to simulate random reactivations; not too complex) |

**Table 1:** Comparative evaluation of candidate systems/techniques for memory consolidation in TORI. **Stability** – resistance to catastrophic forgetting or instability; **Memory Retention** – capacity to retain learned knowledge over sequences; **ψ-Phase Compatibility** – alignment with TORI’s phase-based concept model; **Engineering Effort** – difficulty of implementation (✓✓ easiest, – most difficult). (✓✓ = High/Very Good, ✓ = Medium/Good, – = Low/Poor)

*Key observations:* Techniques like **Koopman spectral memory** and **complementary dual-memory systems** score very high on both stability and ψ-phase alignment, making them prime candidates for TORI’s core design. Energy-based methods (Hopfield, EBMs) also rank strongly, though engineering complexity can be non-trivial (especially for Boltzmann machines or generative replay). Purely neural regularization methods (EWC, SI) are easy to implement but operate at a lower level of abstraction and might need adaptation to work with TORI’s concept graph (they don’t natively address phase dynamics or symbolic structure). Category-theoretic merging ensures structural coherence of the knowledge graph, an often overlooked aspect (no matter how good the learning algorithm, if the knowledge base becomes inconsistent or cluttered, performance will suffer). Therefore, a combination is likely needed: spectral and energy-based learners for **continual adaptation**, plus symbolic graph management for **organizational memory hygiene**.

**Reference Implementation Kit Components**

Based on the above analysis, we propose three prototype components to jump-start TORI’s lifelong learning capabilities. Each is grounded in both research and practical considerations:

**(a) Energy-Based Replay Consolidator**

This module acts as TORI’s “sleep-driven brain cleaner.” It leverages an energy-based model to consolidate and prune memory via replay. **Key design elements:**

* **Memory Energy Model:** We instantiate a simplified **energy-based memory** for concept patterns – for example, a modern Hopfield Network or a constrained Boltzmann machine over the concept activations. The energy function $E(x)$ is defined such that low energy corresponds to desirable, coherent concept configurations (e.g. consistent knowledge states), and high energy to incoherent ones. One could define $E$ using the adjacency matrix of the concept graph (like a Hopfield energy $E = -\frac{1}{2} x^T W x$ where $W$ reflects concept-concept affinities) or via a learned EBM that outputs lower energy for well-rehearsed memories.
* **Replay Batch Sampling:** During consolidation (which might occur at a scheduled time or when the system is idle), the module selects a batch of memories to replay. This selection can be **stochastic proportional to memory strength** (as per the stochastic consolidation model[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits)) – e.g., each concept or memory trace has a probability of being rehearsed based on how wide/deep its basin of attraction is (a measure of stability). Alternatively, we can prioritize recently added or slightly unstable memories (to integrate them) alongside a few older ones (to ensure long-term retention). In practice, TORI can maintain a **Memory Replay Queue** that collects new or high-error cases.
* **Simulated Annealing & Free Energy Minimization:** For each sampled memory (which could be represented as a partial pattern of active concepts or an initial oscillator phase configuration), the consolidator performs a **settling process**. Using *simulated annealing*, we start with the memory’s initial activation pattern and gradually reduce “temperature,” letting the system’s energy dynamics drive it to a nearby energy minimum. Essentially, we are asking: “if this memory were re-encoded now, where would it settle in our concept network?” This process may refine the memory (e.g., dropping spurious details, aligning it with current knowledge). If the memory settles into a *different* attractor than originally, that indicates the memory has been altered by new learning – TORI can decide to either preserve the new form (if it’s more coherent) or, if the drift is undesirable, adjust weights to pull it back. The **wake-sleep EBM paradigm** can be applied here: the memory recall is like the “wake” (we ensure the memory pattern has low energy), and then a global “sleep” update adjusts weights to increase energy of competing patterns and decrease energy of consistent ones[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=consolidation%20and%20propose%20a%20new,is%20minimized%20contrastively%2C%20which%20allows)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=by%20long%20sleep%20phases,We%20provide%20a%20theoretical%20analysis). Concretely, we can perform gradient updates: for each replayed memory, slightly **lower the energy** for that pattern and **raise energy** for confounding patterns (e.g., randomly generated negatives or slightly corrupted versions), akin to contrastive divergence.
* **Synaptic Scaling & Pruning:** After replay/consolidation cycles, the module applies a light **synaptic scaling** – e.g., uniformly decrease all concept link weights by a small factor (simulating global downscaling in sleep) – which prevents runaway weight growth and provides room for new info. During this, any connection that falls below a threshold is pruned (set to zero and removed from the graph). This realizes a form of **Hebbian pruning**: only connections consistently reinforced via replay remain strong, others fade away. Such pruning should be conservative initially (to avoid losing info) but can increase over time to keep the knowledge base size manageable.
* **Outcome:** The energy-based consolidator outputs an updated concept network with reweighted edges and possibly merged nodes. Importantly, it also provides **feedback to the phase oscillator core**: since it identifies stable attractors, it can adjust concept oscillator natural frequencies or coupling to reflect these stable states (ensuring that next time the system is “awake” reasoning, those memories manifest as strong synchronous patterns). By consolidating in this manner, TORI’s knowledge base becomes more **robust (low-energy states are deepened, increasing recall success**[**nature.com**](https://www.nature.com/articles/s41598-022-16407-9#:~:text=decay%20and%20a%20novel%20memory,memory%20impairments%20following%20neurological%20trauma)**)** and more **efficient (spurious or high-energy configurations are eliminated)**.

*Implementation notes:* This module could be prototyped using existing libraries for Hopfield networks or energy models. For instance, one could use a PyTorch implementation of a modern Hopfield layer to store concept vectors, and run a few iterations of its retrieval dynamics as the replay. Boltzmann machines or even simpler contrastive Hebbian learning rules could be tried. Ensuring scalability will be crucial – e.g., limit the size of replay batches per cycle, use partial network updates (so as not to recompute energy over the entire huge graph at once). The engineering lift is moderate: integrating this with TORI’s concept graph and oscillators will require careful interfacing (concept states ↔ EBM states), but each part (Hopfield retrieval, weight decay, etc.) is individually well-understood.

**(b) Koopman Continual Learner with Sparsity**

This component is TORI’s engine for **continual adaptation to concept dynamics**. It builds on the idea that TORI’s reasoning can be seen as a dynamical system – as ALAN processes inputs, concept node activations oscillate and interact. The **Koopman Continual Learner (KCL)** provides a linear but rich model of these dynamics that can be updated incrementally. Key aspects:

* **Streaming Koopman Model:** At its core, we maintain a **Koopman operator approximation** for the concept network’s state evolution. Define a state vector (could be phases or concept activation levels). We choose a set of observable functions (features) – for example, monomials of concept activations, or eigenfunction approximations (possibly using the known ψ from the Banksy core). Using an online Extended Dynamic Mode Decomposition (EDMD), the KCL computes the leading **eigenvalues and eigenfunctions** that capture how the state evolves. This yields modes representing cyclic patterns (oscillations), growth/decay patterns, etc. The *continual* aspect means we update this model with every new batch of experience: using methods like online DMD or sliding windows. However, rather than discarding the past wholesale, we **store spectral objects from each window** (following Redman et al.’s episodic memory approach)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=becomes%20clear%20when%20considering%20applying,implicit%20forgetting%20of%20the%20past)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=Inspired%20by%20the%20integration%20of,with%20episodic%20memory%20we%20develop). Each “episode” might correspond to a distinct context (e.g., a coding session, or a lengthy conversation on a topic), during which the system dynamics are relatively stationary. The spectral snapshot (eigenfunctions and eigenvalues) from that episode is saved in a memory library.
* **Episodic Recall in Spectral Space:** When a new context or task arrives, before analyzing it from scratch, the KCL can **search its library of past spectral patterns** to see if a similar dynamic appeared before. If yes, it can **initialize or bias** the model using those past eigenfunctions. For example, if TORI has seen a similar problem, the way concepts interacted then (perhaps a particular set of concept oscillators went into resonance) can be expected now as well – the KCL would retrieve that Koopman mode and incorporate it, rather than expecting the system to learn it anew. This implements a kind of **transfer learning via dynamics**. Redman *et al.* showed even a basic version of this significantly improved prediction accuracy and avoided relearning repeats[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=approximations%20of%20the%20Koopman%20operator%2C,with%20memory%2C%20sees%20large).
* **Sparse Mode Selection:** A crucial addition for sustainability is enforcing **sparsity**. As tasks accumulate, the library of all modes could grow large. But not all eigenfunctions are equally important. We apply a **sparse optimization** (such as L1-regularized regression or subset selection) to the mode set. This could be done per update: when adding a new spectral component, we solve a convex problem to represent it as a combination of a small number of existing components plus (optionally) a new basis vector. If it can be represented by existing ones (within some error tolerance), we don’t keep a duplicate. If it’s truly novel, we add it but perhaps consider “retiring” another that is now seldom used. Over time, this keeps the active basis limited to, say, the top N modes that explain most dynamics. This is analogous to how the brain might compress many experiences into a finite set of core patterns or schemas.
* **Integration with Phase Oscillators:** The KCL doesn’t operate in isolation – it informs the Banksy oscillator core. The identified eigenfunctions effectively are the ψ-phase patterns that resonantly synchronize groups of conceptsfile-7msf4dlgfjylvyotqb8xbgfile-7msf4dlgfjylvyotqb8xbg. Thus, KCL can update the coupling weights or natural frequencies of the oscillator network to reflect these modes. For instance, if a Koopman mode indicates concept A and B always oscillate in lockstep (phase difference ~0) in an important pattern, the system could strengthen the coupling between A and B’s oscillators or adjust their frequencies to naturally sync. Conversely, if a mode reveals a stable anti-phase relation (phase difference ~π meaning they conflict), that could be fed into a **ψ-SyncMonitor rule** to ensure the concepts are marked as potentially contradictory unless new info resolves the conflict.
* **Learning Algorithm:** We can implement KCL using algorithms like online DMD or streaming PCA on the Hankel matrix of observations. With each new chunk, compute incremental SVD updates to get new eigenfunctions. Then apply a sparsification: e.g., solve a Lasso problem to approximate the new eigenfunction as combination of stored ones. If error is high, include the new one. Optionally, periodically run a **merge step**: check pairwise correlations between stored eigenfunctions and merge ones that are nearly duplicates (this could be done via a category-theoretic merge in the concept graph if the eigenfunctions correspond to concept clusters).
* **Outcome:** The KCL yields a **set of active ψ-modes** and a **Koopman operator** that predicts short-term concept state evolution. This not only helps with memory (by remembering recurring patterns) but also with **planning** – a stable Koopman model means TORI can project forward the consequences of a line of reasoning or code execution in a way that is **Lyapunov-stable and convergent**file-7msf4dlgfjylvyotqb8xbg. In fact, the Koopman eigenvalues can be monitored: if all have magnitude ≤1 (or real parts negative in continuous-time), the system is in a stable regime. If some eigenvalues approach the unstable zone, that signals a potential divergent thought pattern – KCL could alert or invoke consolidation to bring the system back to stability (e.g., through the consolidator or by adjusting gains, analogous to a control system regulator).

*Implementation notes:* There are Python libraries for Koopman analysis and DMD. We can start with a simpler dynamic mode decomposition on simulated concept activation sequences (which we can get from the oscillator matrix during TORI’s normal operation). As we refine it, incorporate more advanced techniques like extended DMD (with nonlinear observables, possibly using neural nets to learn eigenfunctions). Ensuring it runs continually means we need it to be efficient – incremental SVD methods or random projection could be used. The sparsity step can use existing solvers (CVX or scikit-learn Lasso). The largest engineering challenge is likely aligning the math with the actual TORI data structures (concept graph and oscillation data), but once that interface is built, the rest leverages well-known linear algebra routines.

**(c) Category-Typed Concept Merge Utility**

Over time, TORI’s LCN may accumulate duplicate or overlapping concepts (e.g., Concept: Car vs Concept: Automobile) or overly granular distinctions that aren’t needed. The **Concept Merge Utility** uses category theory principles to **unify and prune symbolic knowledge** in a principled way.

* **Typed Concepts and Morphisms:** Each concept node in TORI has a type (from an ontology or schema). For example, in the IDE domain, types might include *Function*, *Class*, *Requirement*; in general knowledge, types like *Entity*, *Event*, etc. We formalize a concept as an **object in a category**, and a relation (edge) as a **morphism**. For instance, a relation "Function A calls Function B" can be a morphism $A \to B$ of type *calls*. The type system constrains which morphisms are allowed (perhaps via a category of types). This typing ensures that only meaningful merges happen (we wouldn’t merge a Function concept with a Requirement concept, for example, as they are different types).
* **Duplicate Detection:** Using heuristics or string similarity or ψ-phase behavior, we identify candidate duplicates/aliases. Suppose concept X and Y are both objects of type *Entity* and share many of the same neighbors (or always phase-sync). The utility will attempt to merge Y into X (or into a new concept Z subsuming both). In categorical terms, we look for a **cone** that points to both X and Y – effectively a generalized concept that has morphisms into X and Y, capturing their common information. If a suitable existing concept serves this role (e.g., a parent class in ontology that both inherit from), maybe the merge is not needed or can be done at that level. If not, we create a new merged node.
* **Pushout for Merge:** The merge operation can be executed as a **pushout in the category of graphs** (more specifically, a pushout of C-sets, following Aguinaldo et al.’s approach[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=concise%20action%20models,This)). Concretely, we identify an equivalence between X and Y and quotient them out, creating a single node. All incoming/outgoing edges of X and Y that are compatible with the merge are unified. For any relation where both X and Y participated, that relation now points to the merged node. If there were relations that only X had or only Y had, the merged node will have both (union of relations), unless some are semantically contradictory – those require conflict resolution (could prompt a review or rely on domain rules to decide which to keep). Throughout, the **ontology constraints are checked**: we preserve semantics as per the user-provided ontology[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=,paper%2C%20we%20conceptually%20compare%20our). For example, if X had a property that Y didn’t, but the ontology says that property’s domain is type *T* which X/Y are, then the merged node of type *T* can have it – that’s fine. If something doesn’t fit merged type’s domain, that might indicate a type change or an error in merging (the tool should flag it rather than forcing it).
* **Category-Theoretic Consistency:** By treating the ontology + graph as a categorical structure, we ensure **no information is lost or misassigned** during merge unless explicitly intended. In category theory, the pushout of two objects identifies them along a span (an equivalence relation). All morphisms rewire accordingly. This formal underpinning prevents, say, a relation from dangling to a now-nonexistent node or a type violation. In effect, it’s like performing database schema merge with guaranteed constraints satisfaction[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=during%20planning,states%20and%20up%02dates%20in%20planning).
* **Concept Pruning:** A simpler case is when a concept is deemed obsolete or irrelevant. This too can be seen categorically: we have a subcategory (the concept and its incident relations) that needs to be *removed*. We can remove it by essentially taking a subgraph complement or by morphisms to a terminal object. But simpler: if a concept is unused (no recent activation, no unique info), the utility can remove it after backing it up. Alternatively, if it has some important historical info, we might mark it as *archived* (so it doesn’t actively take part in reasoning oscillations, but could be retrieved if needed – analogous to our long-term memory that you don’t always recall until prompted).
* **User/Governance Oversight:** Because automatic merging or deletion can be risky in terms of knowledge integrity, the utility could operate in an **interactive or logged mode**, where it suggests merges/prunes and either gets user approval or at least records them for review. This ties into TORI’s emphasis on transparency and auditability.
* **Outcome:** The concept merge utility ensures the LCN remains **lean, well-structured, and semantically consistent**. After its operations, it would likely trigger a re-indexing of the concept graph and possibly notify the oscillator core that topology changed (so it can, for instance, recompute some coupling strengths). Over time, this prevents clutter (which could slow down reasoning and even confuse the phase alignment logic with duplicate nodes) and implements a form of **structural regularization of memory**.

*Implementation notes:* We can leverage libraries like Catlab.jl (in Julia) orNetworkX (Python) with custom code to do pushouts. Initially, manual merge rules (if name matches or high phase synchrony) can drive it, and we ensure type consistency by referencing a central ontology definition (maybe in OWL or a custom schema file). The heavy theory can be abstracted – we don’t necessarily need a full category theory library; we can implement the pushout logic (merging nodes, unifying edges) with careful checks.

**Integration into ALAN’s Cognitive Loop (UML-Style Architecture Map)**

TORI’s ALAN engine operates in a loop of **perception – reasoning – learning – memory consolidation – planning/action**. Below is an integration map describing where each proposed component fits into this loop (conceptually akin to a UML component diagram):

* **1. Input Processing & Concept Activation:** (Unchanged) User inputs (code edits, queries) are parsed into TORI’s internal representations. New or referenced concepts are activated in the LCN, feeding into the Banksy phase-coupled oscillator networkfile-7msf4dlgfjylvyotqb8xbg. Oscillators begin to synchronize or discord based on relationships, and initial reasoning (phase dynamics) unfolds. *Output:* initial answer or suggestion to the user, plus an updated state of concept activations.
* **2. Short-Term Memory Encoding:** Any novel concept or context from this interaction is stored in a fast cache (Short-term memory module). For example, a new concept node might be created for a new API the user introduced, with high plasticity connections. The oscillator phases at conclusion of reasoning form a pattern representing the “experience”. This is analogous to hippocampal encoding of an episode.
* **3. ψ-SyncMonitor & Stability Check:** After each reasoning episode (or during, continuously), a monitor checks the oscillator network for coherence indicators. If strong out-of-phase signals (oscillatory conflicts) persist, it flags them. This could trigger an *immediate adjustment* like asking for clarification (if it’s a logical contradiction in chat) or simply logging the issue. It can also feed data to the **Koopman Continual Learner** (to update the model with any new dynamical behavior observed, including anomalies).
* **4. Continual Learning Update (Koopman):** The Koopman Continual Learner component runs either in real-time alongside reasoning or in mini-batches between user interactions. It takes the recent concept activation trajectory and updates the spectral model. For example, after a coding session where concept A and B oscillated together frequently, KCL will incorporate that. If a similar mode is found in its library, it strengthens its confidence (e.g., “A and B are typically in sync when solving X type of problem”). If it’s new, it records it as a new eigenmode (unless sparsity logic finds it negligible). The updated Koopman operator (with perhaps slight changes to eigenfunctions or eigenvalues) is then used to adjust **oscillator coupling** (ensuring ALAN’s core stays in a calibrated dynamic range) and is available for **prediction** (so if next input is similar, TORI can anticipate the concept interactions). This closes a loop where *experience immediately tunes the internal dynamics*, which will influence next perception.
* **5. Delayed Consolidation Trigger:** The system decides when to undergo a consolidation “sleep” cycle (Energy-Based Replay Consolidator). This might be time-based (e.g., nightly or after X interactions) or event-based (e.g., after a major project is finished, or when ψ-SyncMonitor indicates a drop in overall coherence, or simply when idle). At that point, the system enters a **ψ-phase replay mode** (not responding to new inputs).
* **6. Consolidation & Pruning (Sleep Cycle):** The Energy-Based Replay Consolidator kicks in. It pulls from short-term memory and long-term memory indices a set of items to replay (perhaps weighted by novelty and importance). It then iteratively runs the replay and energy minimization as described: reactivating memory patterns, adjusting weights. During this, the **Concept Merge Utility** can also be invoked: e.g., if during replay it becomes evident that two concept nodes always co-activate identically, the consolidator might call the merge utility to unify them (which will also adjust the energy model to treat them as one). The consolidation modifies the LCN (weights, edges, maybe nodes). It also likely shrinks short-term memory (transferring some to long-term, discarding others). Synaptic pruning occurs here too – if certain connections were never used in any replay, they decay further or are removed. Throughout this cycle, the oscillator core might be temporarily set to a passive mode or a *training mode* where it is driven by the replay patterns rather than external input. In UML terms, this is a separate sub-loop invoked by a controller when conditions are met.
* **7. Post-Consolidation Update:** Once consolidation is complete (could be after a fixed number of replays or a convergence criterion, e.g., minimal changes between iterations), the system “wakes up.” The LCN is now optimized: fewer nodes perhaps, stronger key links, consistent ontologies. The oscillator parameters are updated to reflect the new weights. The Koopman learner might also be informed that some modes are no longer needed (if related to pruned parts of the graph) – it can remove those from its library (this can be done by checking which eigenfunctions corresponded to now-merged concepts and recalculating those).
* **8. Resume Interaction (Planning & Action):** The next time a user query or task comes, TORI now operates on a cleaner, more stable model. The planning loop that generates multi-step solutions or dialog strategies benefits from the consolidated knowledge (e.g., the system might now recall relevant facts without confusion, due to replay, and can plan more coherently). The cycle then continues: encoding new info, continual spectral learning on the fly, periodic consolidation.

This integration ensures that **learning is woven into every stage**: immediate adjustments via spectral updates, and deeper restructuring via “sleep” consolidation. The division of responsibilities is clear in a UML-like diagram: components like **User Interface**, **Parser/Understanding**, **Phase Reasoning Core**, **Knowledge Graph (LCN)**, **Koopman Learner**, **Replay Consolidator**, **Merge Utility**, etc., each interact through well-defined interfaces (e.g., the phase core reads from and writes to the LCN; the consolidator reads the LCN and writes back updated connections; the Koopman learner subscribes to the phase core’s activity feed and sends coupling adjustments).

*(An actual UML diagram would depict these as boxes: e.g., “Phase Oscillator Engine” connected to “Koopman Learner” (updates eigenfunctions), connected to “Concept Graph” (reads structure, writes coupling), connected to “Merge/Prune Utility” (modifies graph structure), etc. A controller/timer governs transitions between normal mode and consolidation mode.)*

**Pilot Benchmark Suite for Lifelong Learning**

To validate and guide TORI’s lifelong learning capabilities, we propose a benchmark suite that tests the system on incremental tasks and measures retention, adaptability, and phase stability:

* **Split-CIFAR Continuum:** A classic vision continual learning benchmark (e.g., Split-CIFAR-100) where the system is shown sequences of image classification tasks (10 classes at a time). Although TORI is not primarily a vision system, this can be adapted by treating each class concept as a concept node and the classification as a reasoning task. The goal is to see if TORI can learn each batch of classes without forgetting previous ones. **KPI:** classification accuracy on all seen classes as tasks progress; **Recall Decay %** (the drop in accuracy on earliest classes by the end). A good lifelong learner should have near 0% recall decay or very slow decay. With TORI’s replay and dual-memory, we expect to approach that, akin to EBM approaches which show minimal forgetting without explicit task boundaries[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf#:~:text=%5BPDF%5D%20energy,incremental%20learning%20without%20using%20replay)[openreview.net](https://openreview.net/forum?id=j5d9qacxdZa#:~:text=Energy,notion%20of%20explicitly%20delineated).
* **Continual Code Parsing/Understanding:** A domain-specific benchmark for TORI IDE: present TORI with a series of increasingly complex codebases or language features in sequence. For example, start with basic Python projects, then add complexity or new languages (one after another). The system must continually update its concept graph (new APIs, patterns) and still recall how to handle the earlier ones. **Datasets:** one could use a chronologically split dataset of coding problems (from easy to hard, or old libraries to new frameworks). **KPI:** success rate on queries about older code after learning new code (measuring catastrophic forgetting in the IDE context), and **Engineering Effort** measure (maybe memory and time overhead as knowledge grows). We also measure **Concept Merge Accuracy** – after learning, check if TORI merged equivalent concepts correctly (this could be measured by seeding duplicates and seeing if it collapses them).
* **Ψ-Phase Drift Test:** A bespoke evaluation for TORI’s oscillatory core. Here we craft scenarios to measure **oscillatory coherence loss** over time. For instance, we can have a known consistent fact pattern (a set of concepts that should remain in sync). We then introduce many distractions (irrelevant concepts or tasks). We measure the **phase coherence** of the original set before and after learning the distractions – i.e., does the phase difference remain near zero (coherent) or did it drift randomly? The drift percentage or the increase in phase variance is the metric (“oscillatory coherence loss”). With robust consolidation, the coherence loss should be low – the system doesn’t forget the original relationships even as noise comes in. This test directly exercises the ψ-phase compatibility of learning algorithms: e.g., we can compare running TORI with and without the Koopman spectral stabilizer on such a test and see the difference in phase stability.
* **Memory Capacity and Pruning Stress Test:** Feed TORI a long sequence of random facts or Q&A (possibly using a dataset like **Wikipedia sequentially** or a QA dataset split into many parts). This is to see how the system scales – do we observe unbounded growth in memory, or does the pruning keep it in check? **KPIs:** number of concept nodes and edges over time (should plateau or grow sub-linearly if pruning/merging works), and query answering accuracy on both recent and very old information (to ensure old info is not lost). We can simulate an “open-ended tutor” scenario where TORI is taught a new piece of knowledge every few seconds for thousands of iterations; at the end we check a sample of early-taught knowledge recall. We also track computational metrics: e.g., **Replay Efficiency** – how many replay cycles (or how much time in consolidation) was needed versus the improvement in retention it gave. Ideally, we want high efficiency: a small amount of replay yields large retention gains[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=decay%20and%20a%20novel%20memory,memory%20impairments%20following%20neurological%20trauma).
* **Hybrid Benchmark (AI Planning with Knowledge Updates):** Since TORI is also about planning (IDE use-case, etc.), we include a scenario where TORI must maintain a plan or design over a long period with changing requirements. For example, use a **continual planning problem** (like in robotics or game AI) where new goals or constraints keep getting added. Evaluate if TORI’s concept graph and reasoning remain consistent (no catastrophic plan failures due to forgetting an earlier requirement). **KPI:** Plan coherence over time (e.g., does it still satisfy old constraints), and the frequency of **conflict alerts by ψ-SyncMonitor** (we want the system itself to catch incoherence, indicating it’s monitoring stability).

Each benchmark should be run on multiple variants of the system (ablation testing): e.g., TORI with full Fusion-4 features vs. TORI with some components turned off (no replay, or no Koopman, etc.) to quantify the contribution of each. We expect, for instance, that disabling the energy-based consolidator would lead to higher recall decay and more oscillatory drift, whereas including it yields slower decay and stable phases.

**Key Performance Indicators Summary:**

* *Memory Retention:* measured by accuracy on old tasks (or recall rate of earlier knowledge) after new learning – target is high retention (low decay%).
* *Stability/Coherence:* measured by oscillatory coherence loss or by consistency metrics in plans – target is minimal coherence loss (phases remain aligned as expected).
* *Plasticity/Adaptability:* measured by performance on newly introduced tasks or speed of learning new info – target is that it remains high (the system can still acquire new concepts quickly, indicating that stability measures haven’t made it too rigid).
* *Efficiency:* e.g., replay efficiency (perhaps ratio of consolidation time to knowledge gained, or how many replays per item are needed), and resource usage (does memory growth slow thanks to pruning? Is compute time bounded for consolidation?).

By systematically using this suite, we can iteratively refine the TORI architecture. For example, if we find oscillatory drift is still an issue, we might strengthen the coupling adjustment from the Koopman learner or increase replay frequency. If memory footprint grows too much, we might need more aggressive merging or a smaller eigenfunction basis. The benchmarks ensure that the **roadmap for TORI’s memory architecture** is empirically validated, not just biologically or theoretically inspired.

**Conclusion**

Through this interdisciplinary survey and design exercise, we have charted a path for making TORI a **truly lifelong learning system**, combining the best of multiple worlds: energy-based models and annealing algorithms from physics to solidify memories, spectral operators and algebraic structures from mathematics to organize and stabilize knowledge, and mechanisms inspired by the brain to manage the balance between rapid learning and long-term retention. By integrating these into TORI’s ψ-phase concept network, we ensure compatibility with its unique phase-oriented reasoning core while pushing the frontier of what an AI research assistant can do. The comparative analysis and proposed implementations serve as a foundation – a **roadmap** – for engineering TORI’s memory architecture in upcoming development phases. With careful construction and testing via the benchmark suite, TORI’s ALAN engine can evolve into a **cognitive partner that never stops learning, yet never forgets the important pieces**, offering reliability and adaptiveness that mirror human cognitive prowess.

Ultimately, the fusion of symbolic and subsymbolic techniques, guided by deep principles of thermodynamics (free-energy minimization), control theory (stability and Koopman spectra), and neuroscience (sleep and consolidation), will position TORI at the forefront of **biologically grounded AI** – an assistant that not only understands and reasons, but also **grows wiser over time**.

**Sources:**

* Li, S. *et al.* (2022). *Energy-Based Models for Continual Learning.* PMLR LLL (showed EBM objective reduces interference)[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf" \l ":~:text=learning%20performance%20in%20classification%20problems,new%20input%2C%20creating%20updates%20that" \t "_blank)[proceedings.mlr.press](https://proceedings.mlr.press/v199/li22a/li22a.pdf#:~:text=lead%20to%20catastrophic%20forgetting,gain%20based%20on%20the%20class).
* Singh, V. *et al.* (2024). *Wake-Sleep Energy Based Models for Continual Learning.* CVPR Workshops (introduced wake-sleep training inspired by biological sleep, improving retention)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=remains%20a%20challenge,phase%2C%20where%20the%20free%20energy)[openaccess.thecvf.com](https://openaccess.thecvf.com/content/CVPR2024W/CLVISION/papers/Singh_Wake-Sleep_Energy_Based_Models_for_Continual_Learning_CVPRW_2024_paper.pdf#:~:text=by%20long%20sleep%20phases,Our%20empirical%20evaluation).
* Redman, W. *et al.* (2025). *Koopman Learning with Episodic Memory.* Chaos 35(1) (saved local Koopman spectral patterns as episodic memory to avoid relearning, boosting prediction accuracy)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=Inspired%20by%20the%20integration%20of,with%20episodic%20memory%20we%20develop)[ar5iv.org](https://ar5iv.org/html/2311.12615v3#:~:text=the%20general%20framework%20we%20introduce,to%20better%20applications%20of%20Koopman).
* Aguinaldo, A. *et al.* (2023). *A Categorical Representation for Knowledge-Based Planning.* (used C-sets and pushout rewriting to maintain ontology-consistent world state updates)[arxiv.org](https://arxiv.org/pdf/2305.17208#:~:text=concise%20action%20models,This).
* Spens, E. & Burgess, N. (2024). *A generative model of memory construction and consolidation.* Nat. Hum. Behav. (hippocampal replay trains cortical generative model, combining unique episodic details with schemas)[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=model%20in%20which%20hippocampal%20replay,the%20use%20of%20limited%20hippocampal)[nature.com](https://www.nature.com/articles/s41562-023-01799-z#:~:text=during%20hippocampal%20replay%20,research%20into%20the%20relationship%20between).
* Shi, Q. *et al.* (2025). *Hybrid neural networks for CL inspired by corticohippocampal circuits.* Nat. Commun. 16:1272 (dual memory system mitigates catastrophic forgetting via specific+generalized memory)[nature.com](https://www.nature.com/articles/s41467-025-56405-9#:~:text=Current%20artificial%20systems%20suffer%20from,neural%20functions%20of%20both%20feedforward).
* Shaham, N. *et al.* (2022). *Stochastic consolidation of lifelong memory.* Sci. Reports 12:13107 (Hebbian RNN with synaptic decay + stochastic memory replay yields extended memory lifetimes)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=maintaining%20old%20memories%20for%20a,The%20number%20of%20retrievable)[nature.com](https://www.nature.com/articles/s41598-022-16407-9#:~:text=problem%20in%20computational%20neuroscience,graded%20retrograde%20and%20anterograde%20deficits).
* **TORI Architecture Docs:** *ALAN 2.x Banksy Core – Phase-Coupled Oscillators & Stability* (unpublished internal) – describes ψ-phase spectral coupling and Lyapunov-stable reasoning dynamicsfile-7msf4dlgfjylvyotqb8xbgfile-7msf4dlgfjylvyotqb8xbg.