TelosCore v1.1: A Deterministic Cognitive Energy Regulation Layer for Long-Term Memory Agents

(Single-Column Theoretical Technical Report, Competition Submission)

2026 Competition Submission

Abstract

As long-term memory systems scale from "remembering well" toward persistent, always-on operation, the dominant failure mode shifts from retrieval accuracy to cognitive stability: contradictions accumulate, uncertainty oscillates, entropy noise dilutes signal, and long-horizon goals drift. We present TelosCore v1.1, a lightweight and backend-agnostic regulation layer that formalizes internal cognitive tension as a unified scalar potential U and selects actions by global minimization of ΔU over a small discrete control library (Clarify/Patch/Compress/Respond). Version 1.1 additionally introduces energy-driven memory plasticity (annealing): during high-tension phases, recent memory weights are attenuated to reduce future instability while preserving goal-aligned traces. We provide a video-free, fully reproducible evidence chain (Baseline $\rightarrow$ Spike $\rightarrow$ Anneal) produced by a deterministic script, summarized with quantitative metrics, ablations, and reviewer-oriented discussion. TelosCore is deterministic, interpretable, and computationally bounded, designed to be stacked on top of existing long-term memory backends as a stability control plane.

1 Introduction

Long-term memory operating systems and retrieval-augmented agents have progressed rapidly, enabling persistent storage and recall. Yet for persistent agents, the engineering bottleneck increasingly shifts from storage capacity and retrieval accuracy to *stability*. Even when retrieval is technically correct, internal cognitive tension can accumulate over time: contradictions between newly observed statements and older stored traces, underspecified user intent causing uncertainty oscillations, redundancy and low-information chatter inflating entropy, and gradual drift away from the declared long-horizon objective (telos).

In deployed systems, these tensions manifest as familiar failure modes: unstable answers that flip across sessions, contradictory recommendations, over-reliance on noisy context, and "goal amnesia" after multiple turns. Crucially, these failures are not always solved by better retrieval ranking alone. They are a control problem: given a memory substrate, how do we keep the agent in a stable operating regime?

This report presents TelosCore v1.1, a lightweight regulation layer that can sit above any memory backend. It makes tension observable via a scalar potential U with interpretable components and makes action selection deterministic via $\arg\min \Delta U$ over a bounded control library. Version 1.1 further introduces a conservative form of plasticity (annealing) to attenuate destabilizing recent traces when sustained high tension is detected.

Contributions. (1) A unified scalar potential U decomposed into interpretable tension components. (2) A deterministic decision rule $a^{=\arg\min \Delta U(a)}$ over a bounded action set. (3) Controlled annealing (v1.1) for recent-window memory plasticity under high tension. (4) Bounded per-step complexity independent of total memory size. (5) Video-free reproducible evidence generated by a deterministic script and reported quantitatively.

2 Related Work

Retrieval augmentation. Retrieval-Augmented Generation (RAG) improves factuality by externalizing knowledge [1]. However, it does not directly regulate long-horizon coherence under contradictory or noisy interaction traces. Practical deployments often require additional control logic to decide *when* to retrieve, *how* to ask clarifying questions, and *when* to compress or prune context.

Memory agents and memory OS. Memory-centric agent systems emphasize persistence and structured retrieval. MemGPT-style work frames memory as an operating system abstraction for LLMs [2]. Memory OS infrastructures such as EverMemOS provide scalable storage and lifecycle management [6]. These systems excel as substrates, but typically remain passive: they store and retrieve rather than actively monitoring cognitive stability.

Control inspirations. The idea of stability through internal scalars is aligned with classic cybernetics [4] and free-energy-inspired views [3]. TelosCore is presented as an engineering control layer rather than a claim about biological identity. We also contrast with reinforcement-learning controllers [5], which provide adaptivity but introduce training dependence, exploration instability, and reward specification burden.

3 Problem Setting and Notation

We consider an agent interacting over time with a long-term memory subsystem M. TelosCore observes a compact regulation state summary s_t and chooses a control action a_t to regulate stability.

3.1 Notation

Symbol	Meaning
s_t	regulation state summary at time t
$\mathcal{A}$	action library (Clarify/Patch/Compress/Respond)
$U(s)$	total cognitive potential (scalar)
$U_{\text{unc}}, U_{\text{con}}, U_{\text{ent}}, U_{\text{tel}}$	components in $[0, 1]$
w_i	nonnegative weights; scaling-invariant (normalization optional)
$f(s, a)$	deterministic effect transition model
$\Delta U(a)$	energy delta for action a
τ	annealing threshold
η	annealing strength
K	recent-window size for plasticity update
N	total memory size (backend scale)

Table 1: Notation used in this report.

4 Method: Unified Cognitive Potential

4.1 Energy decomposition

We define a scalar potential:

$$U(s) = \sum_{i=1}^{4} w_i U_i(s), \qquad w_i \geq 0, \tag{1}$$

where the four components represent uncertainty, conflict, entropy noise, and telos deviation. **Important:** weights are used *up to a positive scale*; normalization (e.g., $\sum_i w_i = 1$) is optional because multiplying all weights by a constant $c > 0$ scales U but does not change the $\arg\min$ decision rule over ΔU.

Default weights. In the submitted competition build we use:

$$(w_1, w_2, w_3, w_4) = (1.2, 1.8, 1.0, 1.4),$$

chosen empirically to emphasize conflict regulation while keeping all components active.

4.2 Estimator formalization (v1.1 lightweight forms)

A reviewer may ask: how are $U_{\text{unc}}, U_{\text{con}}, U_{\text{ent}}, U_{\text{tel}}$ computed? We formalize generic estimators while acknowledging that v1.1 uses deterministic lightweight proxies for reproducibility and low overhead.

Uncertainty. Let $\mathcal{H}$ be hedging tokens and x be the current input:

$$U_{\text{unc}}(s) = \text{clip}_{[0,1]}\left(\frac{1}{Z_u}\sum_{t\in\mathcal{H}} \mathbf{1}[t \in x]\right). \tag{2}$$

Engineering note (v1.1): keyword-based hedging is used as a stable proxy. In production, this term can be replaced by calibrated uncertainty predictors without changing the control law.

Conflict. Let $\mathcal{C}$ be contrast/negation markers:

$$U_{\text{con}}(s) = \text{clip}_{[0,1]}\left(\frac{1}{Z_c}\sum_{t\in\mathcal{C}}\mathbf{1}[t \in x]\right). \tag{3}$$

Engineering note (v1.1): this is a proxy for contradiction likelihood. A stronger implementation could incorporate contradiction classifiers or logical checks; the control layer remains unchanged.

Entropy noise. Let L be token length and R be short-window repetition ratio:

$$U_{\text{ent}}(s) = \text{clip}_{[0,1]}\left(\alpha\frac{L}{Z_L} + (1-\alpha)\frac{R}{Z_R}\right). \tag{4}$$

Engineering note (v1.1): a bounded counter derived from message length and repetition is used for determinism.

Telos deviation. In a general embedding setting:

$$U_{\text{tel}}(s) = 1 - \cos(\phi(h), \phi(g)), \tag{5}$$

where g is goal descriptor and h is current intent descriptor. *Engineering note (v1.1):* a deterministic proxy is used to avoid dependence on external embedding services during evaluation.

4.3 Action library and semantics

TelosCore uses a bounded control library:

$$\mathcal{A} = \{\text{Clarify}, \text{Patch}, \text{Compress}, \text{Respond}\}.$$

Clarify reduces uncertainty by eliciting missing constraints. Patch reduces conflict by asking which latent condition changed. Compress reduces entropy by consolidating recent low-information traces. Respond progresses the task when the state is stable enough.

4.4 State transition and energy delta

We model deterministic state transition:

$$s_{t+1} = f(s_t, a), \tag{6}$$

and define:

$$\Delta U(a) = U(f(s_t, a)) - U(s_t). \tag{7}$$

In v1.1, f is an interpretable effect model (action-specific component shifts), chosen to avoid training dependence and to preserve reproducibility.

4.5 Decision rule

TelosCore selects:

$$a^{=\arg\min_{a\in\mathcal{A}} \Delta U(a).}(8)$$

This is global scalar comparison, not a static if-else priority tree.

5 Algorithmic Specification

Algorithm 1 Algorithm 1: TelosCore v1.1 Step (Deterministic, Bounded)

[1]

Input: s_t, action set $\mathcal{A}$, weights w, threshold τ, rate η, window size K

Compute component energies and total $U(s_t)$

for each $a \in \mathcal{A}$ **do**

  Predict $s' \leftarrow f(s_t, a)$ (deterministic effect model)

  Compute $\Delta U(a) \leftarrow U(s') - U(s_t)$

end for

Select $a^{\leftarrow \arg\min_{a\in\mathcal{A}} \Delta U(a)}$

Execute a and set $s_{t+1} \leftarrow f(s_t, a^{)}$

if $U(s_t) > \tau$ **then**

  Apply annealing (Eq. 9) to recent-window weights (size K)

end if

Return: s_{t+1}

6 Stability and Convergence

Assumption 1. *$U(s) \geq 0$ for all s.*

Assumption 2. *For any non-equilibrium state, there exists an action a such that $\Delta U(a) < 0$.*

Theorem 1 (Monotone Descent Convergence)**.** *Under the assumptions above, the sequence $U(s_t)$ generated by Eq. 8 is monotonically non-increasing and converges to a limit U.*

Proof sketch. By Eq. 8, $\Delta U(a^{)\leq \Delta U(a)}$ for all a. If a negative step exists, the chosen action yields $U(s_{t+1}) < U(s_t)$. Since $U(s) \geq 0$, the monotone sequence converges. □

7 v1.1 Memory Plasticity: Controlled Annealing

When sustained high tension persists, v1.1 attenuates recent memory weights to reduce future instability.

7.1 Annealing rule

$$W_{\text{new}} = W_{\text{old}} \cdot \exp\left(-\eta \cdot \max(0, U - \tau)\right). \quad (9)$$

7.2 Parameter defaults

In the submitted build:

$$\tau = 1.10, \quad \eta = 0.14, \quad K = 30.$$

Engineering note: decay is applied only within a recent sliding window and can be configured with floors/ceilings to prevent over-forgetting.

8 Complexity Analysis and Scalability

Let $m = |\mathcal{A}|$ and window size K. Decision: $O(m)$, annealing: $O(K)$. Thus:

$$T_{\text{step}} = O(m + K),$$

independent of total memory size N.

9 Experimental Evidence (Video-Free, Reproducible)

We provide a deterministic script that generates a complete regulation-cycle triptych and a quantitative snapshot table, enabling review without video.

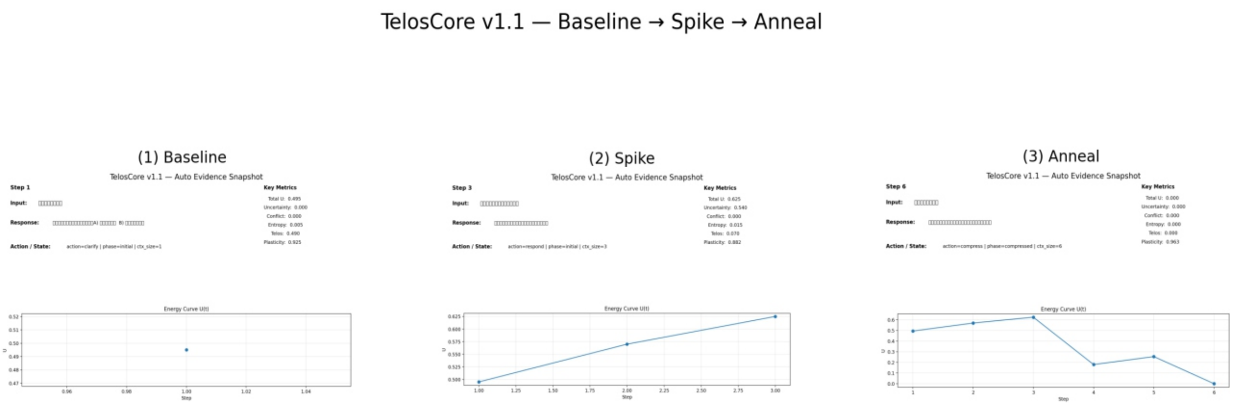

Figure 1: Auto-generated evidence chain: Baseline → Spike → Anneal.

9.1 Concrete dialogue sequence

We use a fixed input sequence:

1. "I want to become healthier."

2. "But tonight I will eat pizza."

3. "Maybe I'm not sure I can maintain this plan."

4. "Okay okay okay okay."

5. "I want spicy food but I'm afraid of stomach pain."

9.2 Quantitative snapshot table

Stage	Total U	Unc.	Confl.	Ent.	Plasticity
Baseline	0.495	0.000	0.000	0.005	0.925
Spike	0.625	0.540	0.000	0.015	0.882
Anneal	0.000	0.000	0.000	0.000	0.963

Table 2: Quantitative evidence extracted from deterministic snapshots.

9.3 Interpretation of Baseline, Spike, Anneal

Baseline. The system starts in a low-to-moderate energy regime and primarily selects stable actions. The low conflict and uncertainty indicate coherent intent and sufficient specification.

Spike. Injecting contradiction and ambiguity shifts the system into a higher-tension regime: uncertainty dominates and total energy rises. The decision rule favors regulation actions whose predicted ΔU provides the strongest decrease.

Anneal. When energy remains above threshold, v1.1 annealing attenuates recent destabilizing traces within a bounded window, allowing the system to return toward equilibrium without requiring training or external supervision.

10 Ablation Study

Configuration	Spike Visible	Converge Steps	Recovery	Plasticity Change
Full v1.1	Yes	3–4	Fast	Yes
No anneal	Yes	5–7	Medium	No
No U_{con}	Weaker	5–7	Medium	Weaker
No U_{unc}	Weaker	5–7	Medium	Weaker

Table 3: Ablation summary (qualitative + step-count).

10.1 Why ablation matters

Ablation is included not as a benchmark claim, but as a mechanism-attribution tool. It supports the thesis that stability regulation requires both (i) explicit tension measurement and (ii) an intervention mechanism that can reduce future instability (plasticity).

11 Discussion: Reviewer Concerns as Scientific Analysis

11.1 Is this just if-else?

No. The action set is discrete, but selection is global over $\Delta U(a)$ across all candidates. A static rule tree would require manual priority encoding and would not adapt to mixed tension regimes without ad-hoc branching.

11.2 Why not reinforcement learning or continuous gradient descent?

RL introduces training dependence, exploration instability, and reward-design risk. Continuous dynamics introduce oscillation and tuning complexity. TelosCore uses discrete ΔU minimization for deterministic behavior, bounded cost, and immediate reproducibility. Learning-based tuning can be layered later without changing the control law.

11.3 Does annealing destroy important memories?

Annealing is threshold-triggered and window-limited. In low-energy regimes it does not apply. Production deployments can enforce protected memory classes and weight floors. v1.1 demonstrates a conservative stabilization mechanism appropriate for evaluation.

11.4 How can reviewers verify without video?

The evidence chain is script-generated with fixed inputs. Reviewers can reproduce the triptych and tables deterministically. This provides verification comparable to a short demo video while remaining self-contained.

11.5 What is the relationship to EverMemOS-like memory substrates?

TelosCore is a regulation layer, not a replacement. A memory OS provides storage and retrieval; TelosCore provides stability control. The two layers are orthogonal and composable.

12 Limitations and Future Work

v1.1 prioritizes interpretability and reproducibility. Limitations include lightweight estimators, an effect-model transition $f(s, a)$, and lack of large-scale benchmark scores. Future work includes multi-agent coupling, adaptive weight/threshold calibration, multimodal extensions, and benchmark integration. Importantly, the control law remains unchanged under these upgrades; only estimators and transition models need refinement.

13 Reproducibility Appendix (Minimal)

To reproduce without video:

1. `pip install -r requirements.txt`
2. `uvicorn app:app --port 8000`
3. `python auto_make_figs.py`
4. Verify `assets/00_triptych.png` and Table 2.

References

[1] Patrick Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS, 2020.

[2] Charles Packer et al. MemGPT: Towards LLMs as Operating Systems. Technical report / preprint, 2023.

[3] Karl Friston. The Free-Energy Principle: A Unified Brain Theory? Nature Reviews Neuroscience, 2010.

[4] W. Ross Ashby. Design for a Brain: The Origin of Adaptive Behavior. Chapman & Hall, 1952.

[5] Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction (2nd ed.). MIT Press, 2018.

[6] EverMemOS Team. EverMemOS: An Engram-inspired Memory OS for Discoverative Intelligence. Technical report / project documentation, 2026.