Project Aegis: A Spectrally-Guided Hierarchical Control System for Behavioral Self-Regulation

Blake Kell
Macalester College

January 22, 2026

Abstract

We present Project Aegis, a cognitive architecture for modeling behavioral self-regulation as navigation through a learned action manifold. The system addresses a fundamental challenge in behavioral modeling: agents can *know* what is good for them while systematically failing to *do* it. Our architecture separates this problem into three components: (1) a **Plant** that enforces thermodynamic constraints on physiological resources, (2) an **Emissary** that performs tactical action selection via Model Predictive Control, and (3) a **Master** that sets strategic objectives. The action space is constructed via spectral embedding of behavioral data, producing a manifold where geometric proximity encodes behavioral similarity. We demonstrate a working implementation where regression-based state transition models enforce realistic recovery dynamics, and where willpower functions as a depletable resource that modulates the trade-off between immediate needs and long-term goals. The system includes an Observer that tracks prediction errors via temporal difference learning, enabling surprise-driven adaptation. We report results from 81 days of personal behavioral data, showing emergent constraint satisfaction without hard-coded rules.

1 Introduction

The challenge of behavioral self-regulation—choosing actions aligned with long-term goals despite short-term physiological pressures—is poorly captured by standard reinforcement learning frameworks. Traditional RL assumes an agent whose preferences are aligned with outcomes: the reward function encodes what the agent *should* want, and learning consists of discovering how to achieve it. But human behavior exhibits a systematic divergence between *wanting* and *doing*, between *knowing better* and *acting better*.

We propose that this divergence arises from a structural asymmetry between two matrices that govern behavior:

- **Affordance** (W): A learned, correlation-based mapping from internal state to behavioral preferences. This encodes "what feels appealing given this state."
- **Feedback** (F): A causal mapping from behavior to physiological consequences. This encodes "what the action actually does to the body."

When W and F diverge, maladaptive behaviors emerge: the agent *wants* actions that *harm* it. Addiction, procrastination, and self-sabotage can be understood as consequences of this divergence.

Project Aegis formalizes this insight into a hierarchical control architecture where:

1. The **Master** sets strategic objectives (what to want)
2. The **Emissary** executes tactical actions (how to get it safely)

3. The **Observer** maintains a world model and detects surprise
4. The **Plant** enforces physiological constraints (what is possible)

The action space is not hand-designed but *learned* from behavioral data via spectral embedding of a similarity graph. This produces a manifold where clusters represent behavioral archetypes, and navigation through the manifold corresponds to behavioral choice.

1.1 Contributions

1. A formal separation of learned affordance (W) from causal feedback (F), explaining maladaptive behavior as matrix divergence
2. A working implementation of the Emissary layer with regression-based state transitions trained on real behavioral data
3. A willpower-as-resource model with fatigue dynamics that produces emergent recovery behavior
4. An Observer with TD-error tracking that provides the foundation for surprise-driven plasticity
5. Empirical demonstration on 81 days of personal behavioral logs

2 Related Work

2.1 Hierarchical Reinforcement Learning

The Master/Emissary separation draws from Options framework [2] and feudal reinforcement learning [3], where high-level controllers set subgoals for low-level executors. Our contribution is the use of Model Predictive Control (rather than learned policies) for the Emissary, providing formal safety guarantees.

2.2 Spectral Graph Theory in Machine Learning

Spectral clustering [4] and Laplacian Eigenmaps [1] use eigenvectors of the graph Laplacian for dimensionality reduction. We extend this by interpreting eigenvalues as "mode costs" and using spectral coordinates for action abstraction.

2.3 Computational Models of Self-Regulation

Ego depletion models [5] treat willpower as a limited resource. While the original findings are contested, the computational abstraction remains useful. Our willpower fatigue mechanism operationalizes this as an exponential moving average of override magnitude.

3 Architecture Overview

The system state at time t is:

$$S_t = (\mathbf{i}_t, \mathbf{b}_t, \mathcal{H}_t, \alpha_t, f_t)$$

where:

- $\mathbf{i}_t \in \mathbb{R}^4$: Internal state (sleep quality, energy, valence, arousal)
- $\mathbf{b}_t \in \mathbb{R}^9$: Behavioral feature vector

- $\mathcal{H}_t$: Habit manifold (spectral embedding of behavioral history)
- $\alpha_t \in [0, 1]$: Willpower parameter
- $f_t \in [0, 1]$: Willpower fatigue

3.1 The Plant: Physiological Substrate

The internal state $\mathbf{i}_t$ is partitioned into two geometric classes:

Resources (dimensions 0-1): Sleep quality and waking energy follow monotonic decay under effort:

$$i^{(r)}_{t+1} = i^{(r)}_t - \eta \cdot \text{effort}_t \cdot (1 + \alpha_t \cdot \text{override}_t)$$

Recovery occurs via specific state-restoring behaviors (rest, sleep).

Axes (dimensions 2-3): Valence and arousal follow Ornstein-Uhlenbeck dynamics with homeostatic mean-reversion:

$$i^{(a)}_{t+1} = i^{(a)}_t + \eta_a \cdot \text{stimulus}_t + \gamma \cdot (i^{(a)}_{\text{rest}} - i^{(a)}_t)$$

3.2 The Emissary: Tactical Controller

The Emissary selects actions via Model Predictive Control. Given the Master's willpower parameter α and target ideal $\mathbf{v}_{\text{ideal}}$, it computes utilities for each behavioral cluster k:

$$U(\boldsymbol{\mu}_k) = \alpha_{\text{eff}} \cdot (\boldsymbol{\mu}_k \cdot \mathbf{w}_{\text{ideal}}) + (1 - \alpha_{\text{eff}}) \cdot (\boldsymbol{\mu}_k \cdot \mathbf{w}_{\text{state}})$$

where:

- $\mathbf{w}_{\text{ideal}} = A \cdot \mathbf{v}_{\text{ideal}}$ (behavioral weights from ideal state)
- $\mathbf{w}_{\text{state}} = A \cdot \mathbf{i}_t$ (behavioral weights from current state)
- $\alpha_{\text{eff}} = \alpha \cdot (1 - f_t)$ (effective willpower after fatigue)
- $A \in \mathbb{R}^{9\times 4}$ is the Interaction Matrix

The Emissary selects $a^* = \arg\max_k U(\boldsymbol{\mu}_k)$, optionally subject to safety constraints on predicted resource levels.

3.2.1 Willpower Fatigue Dynamics

When the Emissary overrides need-driven behavior, fatigue accumulates:

$$\text{override}_t = \|\boldsymbol{\mu}_{\text{optimal}} - \boldsymbol{\mu}_{\text{deterministic}}\|$$

$$f_{t+1} = (1 - \beta) \cdot f_t + \beta \cdot \frac{\text{override}_t}{\text{override}_t + 1}$$

This creates burnout dynamics: sustained willpower exertion degrades the capacity for further exertion, eventually forcing the agent toward need-driven behavior.

3.3 The Decoder: Forward Model

Rather than hand-tuned physics, state transitions are predicted by trained linear regression models:

Context Evolution (behavior → sleep parameters):

$$\text{hours\_slept}_{t+1} = f_1(\boldsymbol{\mu}_{\text{selected}}, \text{NIC}, \text{CAF}) \tag{1}$$
$$\text{bedtime\_std}_{t+1} = f_2(\boldsymbol{\mu}_{\text{selected}}, \text{NIC}, \text{CAF}) \tag{2}$$

State Readout (sleep parameters → resources):

$$\text{energy}_{t+1} = g_1(\text{hours\_slept}, \text{bedtime\_std}, \text{energy}_t, \text{NIC}, \text{CAF}) \tag{3}$$
$$\text{sleep\_quality}_{t+1} = g_2(\cdots) \tag{4}$$

The regression models are trained on z-scored inputs with MinMax-scaled targets, then mapped to internal coordinates $[-2, 2]$ via:

$$z = 4 \cdot (y_{[0,1]} - 0.5)$$

3.4 The Observer: World Model and Surprise Detection

The Observer maintains a value function $V(S)$ and computes temporal difference errors:

$$\delta_t = r_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

The reward function encodes sustainable progress toward the ideal:

$$r_t = -\|\mathbf{i}_t - \mathbf{v}_{\text{ideal}}\| - \lambda_1 \cdot \text{Var}(\mathbf{i}_t) - \lambda_2 \cdot f_t$$

TD-error magnitude serves as a surprise signal. The **Cognitive Reflex** monitors $|\delta_t|$:

- Spike surprise ($|\delta_t| > \tau$, isolated): Policy may be wrong → increase exploration
- Persistent surprise ($\mathbb{E}[|\delta_t|] > \tau$ over window): Model may be wrong → trigger manifold re-embedding

3.5 The Master: Strategic Controller (Theoretical)

The Master operates on slow timescales (hours/days) to set the willpower parameter α_t. Its inputs include:

- Aggregated physiological trend $\bar{\mathbf{i}}_t$
- TD-error statistics $\mathbb{E}[|\delta_t|]$, $\text{Var}(|\delta_t|)$
- Progress toward ideal $\int_{t-T}^{t} \|\mathbf{b}_\tau - \mathbf{v}_{\text{ideal}}\| d\tau$

The Master's learning objective:

$$\max_{\pi_{\text{master}}} \mathbb{E}\left[\sum_t \gamma^t \left(r_t - \lambda \cdot \text{Var}(\mathbf{i}_t)\right)\right]$$

This maximizes progress while penalizing physiological instability, encoding "sustainable operation."

Note: The Master is currently implemented as a static parameter or simple threshold-based rules. Neural policy learning is future work.

4 The Action Manifold

4.1 Construction via Spectral Embedding

The action space is constructed from 81 days of behavioral data, each represented as a 9-dimensional feature vector (see Appendix A). The pipeline:

1. **Normalization**: MinMax scaling to $[0, 1]$
2. **Affinity Graph**: k-nearest neighbors ($k = 5$) with Euclidean distance
3. **Graph Laplacian**: Normalized Laplacian $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$
4. **Spectral Embedding**: Project onto eigenvectors corresponding to smallest non-zero eigenvalues
5. **Clustering**: DBSCAN ($\epsilon = 0.04$, min_samples$= 3$) or KMeans
6. **Centroid Extraction**: Mean position in both embedded and original feature space

4.2 Geometric Interpretation

The eigenvectors of the graph Laplacian provide coordinates that preserve local neighborhood structure. Points that are similar in 9D behavioral space remain proximate in the 2D embedding.

Clustering identifies **behavioral archetypes**—regions of the manifold that represent coherent behavioral patterns. In our data, we identified 3-4 primary archetypes:

- **High-Intensity Work**: High values on Work, Focused Learning, System Architecture
- **Rest/Recovery**: High Passive Media, low effort dimensions
- **Low-Intensity Generation**: Moderate Skill Practicing, Active Media

4.3 Voronoi Tessellation

A Voronoi diagram partitions the embedded space into regions, each associated with one archetype centroid. Any point in a region is geometrically closest to that region's archetype. This provides:

- **Action Abstraction**: The agent selects among discrete archetypes rather than continuous 9D vectors
- **Consequence Prediction**: Proximity to a centroid implies similar behavioral outcomes
- **Manifold Warping**: Internal state modulates effective distances, expanding/contracting regions

5 Implementation

The system is implemented in Python ($\sim$700 lines) using NumPy, Pandas, Scikit-learn, SciPy, and Matplotlib.

5.1 Data Pipeline

1. `phase1.py`: Raw data → Feature engineering → Normalized feature matrix
2. `action_space.py`: Extract 9 behavioral features → Separate action space
3. `action_space_embedded.py`: Spectral embedding → Clustering → Centroids
4. `run_regressions.py`: Train transition models → Export equation pack

5.2 Core Agent

`aegis_map_v2.py` implements:

- `AegisMap`: Main agent class with state, action selection, and update logic
- `Observer`: TD-error tracking, value function, cognitive reflex
- Interactive visualization with matplotlib widgets
- CLI mode for batch simulation

5.3 Key Algorithms

Algorithm 1 Emissary Action Selection

Require: Current state $\mathbf{i}_t$, ideal $\mathbf{v}_{\text{ideal}}$, willpower α, fatigue f

$\alpha_{\text{eff}} \leftarrow \alpha \cdot (1 - f)$
$\mathbf{w}_{\text{state}} \leftarrow A \cdot \mathbf{i}_t$
$\mathbf{w}_{\text{ideal}} \leftarrow A \cdot \mathbf{v}_{\text{ideal}}$
for each cluster k **do**
  $\text{needs}_k \leftarrow \boldsymbol{\mu}_k \cdot \mathbf{w}_{\text{state}}$
  $\text{ideal}_k \leftarrow \boldsymbol{\mu}_k \cdot \mathbf{w}_{\text{ideal}}$
  $U_k \leftarrow \alpha_{\text{eff}} \cdot \text{ideal}_k + (1 - \alpha_{\text{eff}}) \cdot \text{needs}_k$
end for
return $\arg\max_k U_k$

6 Results

6.1 Emergent Recovery Behavior

The primary empirical result is that the regression-based Decoder enforces realistic recovery dynamics *without hard-coded rules.* When the agent sustains high-intensity work:

1. Energy and sleep quality decline (regression prediction)
2. Declining resources shift $\mathbf{w}_{\text{state}}$ toward rest-aligned behaviors
3. Eventually, even with high willpower, the utility of Rest exceeds Work
4. The agent "chooses" recovery

This resolves what we call the "always grind" problem: naive implementations where willpower can override physiological needs indefinitely.

Algorithm 2 State Update (One Day Step)

Require: Selected action $\boldsymbol{\mu}^*$, current state $\mathbf{i}_t$

// Willpower fatigue update
override $\leftarrow \|\boldsymbol{\mu}^* - \boldsymbol{\mu}_{\text{deterministic}}\|$
$f_{t+1} \leftarrow (1-\beta)f_t + \beta \cdot \text{override}/(\text{override}+1)$

// Regression-based resource prediction
$(\text{sleep}, \text{energy}) \leftarrow \text{Decoder}(\boldsymbol{\mu}^*, \text{context})$
$\mathbf{i}_{t+1}[0:2] \leftarrow (\text{sleep}, \text{energy})$

// Within-day effort drain
effort $\leftarrow \sum_{j \in \text{EFFORT}} \max(0, \mu_j^*)$
$\mathbf{i}_{t+1}[0:2] \leftarrow \mathbf{i}_{t+1}[0:2] - \eta \cdot (1 + \alpha_{\text{eff}} \cdot \text{override}) \cdot \text{effort}$

// Valence/arousal dynamics
$\mathbf{i}_{t+1}[2] \leftarrow \mathbf{i}_{t+1}[2] + \eta_v \cdot (\text{positive} - \text{negative})$
$\mathbf{i}_{t+1}[3] \leftarrow \mathbf{i}_{t+1}[3] + \eta_a \cdot (\text{stimulating} - \text{calming})$

// Homeostatic mean-reversion
$\mathbf{i}_{t+1}[2:4] \leftarrow \mathbf{i}_{t+1}[2:4] + \gamma \cdot (\mathbf{i}_{\text{rest}}[2:4] - \mathbf{i}_{t+1}[2:4])$

return $\mathbf{i}_{t+1}$

6.2 Willpower Fatigue Dynamics

Figure **??** shows a typical trajectory. As the agent overrides need-driven behavior:

- Fatigue accumulates (red line rises)
- Effective willpower decreases (gap between set and effective widens)
- Eventually the agent is "forced" toward need-aligned actions
- During low-override periods, fatigue partially recovers

6.3 TD-Error as Surprise Signal

The Observer tracks prediction errors across the simulation. High $|\delta_t|$ indicates unexpected outcomes:

- Isolated spikes: The policy produced an unusual outcome
- Persistent elevation: The world model may be stale

In testing, we observe TD-error spikes when the agent transitions between behavioral regimes (e.g., from sustained Work to forced Rest).

6.4 Manifold Warping Visualization

The weighted Voronoi diagram provides interpretable visualization of how internal state affects action selection. As energy decreases:

- The "Rest" region expands
- The "High-Intensity Work" region contracts
- Boundary shifts show which transitions become more/less likely

7 Spectral Analysis of Behavioral Structure

Beyond dimensionality reduction, the spectral properties of the behavioral graph encode deeper structural information about an individual's behavioral dynamics. This section outlines theoretical extensions that connect the manifold construction to measures of behavioral rigidity and resilience.

7.1 The Daily State Graph

We construct a graph $G = (V, E, W)$ where:

- **Nodes**: Each node v_t represents the complete state vector $\mathbf{x}(t)$ for day t
- **Decomposition**: The state decomposes as $\mathbf{x}(t) = \mathbf{x}_h + \mathbf{x}_p(t)$, where $\mathbf{x}_h$ is the long-term average (homogeneous component) and $\mathbf{x}_p(t)$ is the daily deviation (particular component)
- **Edges**: Edge weights encode similarity of deviation patterns via Gaussian kernel:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{x}_p(i) - \mathbf{x}_p(j)\|^2}{2\sigma^2}\right)$$

This construction means high edge weights connect days with similar *patterns of deviation from baseline*, not just similar absolute states.

7.2 The Fiedler Value: Algebraic Connectivity

The second-smallest eigenvalue of the graph Laplacian, λ_1 (the Fiedler value), quantifies how well-connected the behavioral graph is:

- **High** λ_1: The graph is well-mixed; behavioral states flow easily into one another
- **Low** λ_1: A bottleneck exists; the graph can be partitioned into weakly-connected clusters

The corresponding Fiedler vector provides the optimal bipartition—identifying which days belong to which behavioral "island."

7.3 Cheeger's Inequality: Quantifying Behavioral Ruts

The Cheeger constant $h(G)$ measures the "worst bottleneck" in the graph—the sparsest cut that partitions it:

$$h(G) = \min_{S \subset V} \frac{|\partial S|}{\min(\mathrm{vol}(S), \mathrm{vol}(\bar{S}))}$$

Cheeger's inequality connects this to the Fiedler value:

$$\frac{\lambda_1}{2} \leq h(G) \leq \sqrt{2\lambda_1}$$

Behavioral interpretation: A low Cheeger constant provides mathematical proof of a behavioral "rut"—a cluster of states (e.g., depressive episodes, procrastination cycles) that is structurally isolated from healthier states. The individual can enter this region easily but faces a "barrier" to exit.

7.4 Sobolev Inequalities: Measuring Resilience

Sobolev inequalities relate a signal's smoothness (via the Laplacian) to its overall variance. For a random walk on the behavioral graph, this bounds the convergence rate to steady-state:

$$\|f - \bar{f}\|_2^2 \leq \frac{1}{\lambda_1} \cdot f^T L f$$

Behavioral interpretation: The Sobolev constant serves as a metric for psychological resilience—the ability to return to baseline $\mathbf{x}_h$ after perturbation. A graph with good Sobolev properties is one where any initial state rapidly diffuses toward equilibrium, indicating a robust, self-correcting behavioral system.

7.5 Integration with the Control Architecture

These spectral measures inform the hierarchical control system:

1. **Rut Detection**: The Cognitive Reflex can monitor for declining λ_1, signaling that the agent is entering a structurally isolated behavioral region
2. **Bottleneck Traversal**: The Master can explicitly reward actions that cross Cheeger cuts—escaping behavioral ruts rather than optimizing within them
3. **Resilience Optimization**: Long-term Master objectives can include improving the graph's Sobolev constant, making the behavioral system more self-correcting over time
4. **Manifold Plasticity**: Hebbian updates to the interaction matrix A reshape the graph topology, potentially "filling in" bottlenecks by strengthening connections between previously isolated behavioral clusters

This spectral perspective reframes habit formation: the goal is not just to reach desired states, but to reshape the manifold topology so that desired behaviors become *attractors* with low energy barriers and high connectivity to the rest of the behavioral space.

8 Limitations and Future Work

8.1 Current Limitations

- **Coarse Action Space**: 3 clusters provide limited behavioral nuance. Higher-dimensional spectral embedding before clustering may reveal finer structure.
- **Static Master**: The willpower parameter is currently set manually or via simple thresholds. Learning the Master policy requires simulation infrastructure for sample-efficient RL.
- **Linear Decoder**: The regression models are linear. Non-linear dynamics (threshold effects, interactions) are not captured.
- **Single-Subject Data**: 81 days from one individual limits generalizability. The framework should be validated on multiple subjects.
- **Day-Level Granularity**: Each step represents one day. Intra-day dynamics (morning vs. afternoon) are not modeled.

8.2 Future Directions

8.2.1 Control Architecture Extensions

- **Master Learning**: Implement model-based RL using the Decoder as a simulator. Train the Master to set α_t optimally via policy gradient methods.
- **Intra-Day Resolution**: Separate morning reset (regression-based) from intra-day dynamics (physics-based), enabling finer-grained behavioral modeling.
- **Constraint Formalization**: Wire the MPC safety constraints into the Emissary's action selection with explicit energy floor enforcement.

8.2.2 Spectral Analysis Extensions

- **Spectral Mode Validation**: Test whether eigenvectors correspond to psychologically meaningful dimensions (e.g., activity/rest, cognitive/physical) through correlation with self-report measures.
- **Dynamic Cheeger Monitoring**: Track the Cheeger constant over rolling windows to detect when the agent is approaching or entering a behavioral rut.
- **Resilience Metrics**: Compute Sobolev constants from the behavioral graph and correlate with subjective measures of psychological flexibility.
- **Bottleneck-Aware Rewards**: Explicitly reward the Master for actions that traverse identified Cheeger cuts, enabling escape from maladaptive attractor states.

8.2.3 Machine Learning Integration

- **RNN for Temporal Prediction**: Use the graph Laplacian to define the state space topology that a Recurrent Neural Network learns to navigate, modeling temporal sequences of state transitions beyond the current Markov assumption.
- **Graph Neural Networks**: Replace linear spectral embedding with learned graph convolutions that can capture non-linear manifold structure.
- **Manifold Plasticity**: Implement Hebbian updates to the interaction matrix A based on experienced correlations, allowing the behavioral topology to reshape with learning.

9 Conclusion

Project Aegis demonstrates that behavioral self-regulation can be modeled as constrained navigation through a learned action manifold. The key insight—separating learned affordance from causal feedback—provides a formal account of why agents can systematically want things that harm them.

The working implementation shows:

1. Regression-based state transitions enforce realistic recovery without hard-coded rules
2. Willpower fatigue creates emergent burnout and recovery cycles
3. TD-error tracking provides a foundation for surprise-driven adaptation
4. Spectral embedding produces interpretable action abstractions

The architecture bridges control theory (MPC for the Emissary), reinforcement learning (TD-learning for the Observer), and spectral graph theory (manifold construction). While the Master learning loop remains future work, the geometric and control-theoretic foundations are in place.

The broader implication is that habit formation can be understood as reshaping a manifold—moving desired behaviors from high-energy (high-willpower-cost) modes to low-energy (habitual) modes. This provides both a computational framework for building self-regulating agents and a conceptual lens for understanding human behavioral dynamics.

References

[1] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[2] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[3] P. Dayan and G. E. Hinton, "Feudal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 5, pp. 271–278, 1993.

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 14, pp. 849–856, 2001.

[5] R. F. Baumeister, E. Bratslavsky, M. Muraven, and D. M. Tice, "Ego depletion: Is the active self a limited resource?" *Journal of Personality and Social Psychology*, vol. 74, no. 5, pp. 1252–1265, 1998.

[6] F. R. K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics, No. 92, American Mathematical Society, 1997.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proc. KDD*, pp. 226–231, 1996.

A Behavioral Feature Dictionary

The 9 behavioral features used to construct the action space:

Work A 0–5 scale of work intensity (0=No work, 5=Extremely demanding)

Focused Learning Time spent on deliberate cognitive skill-building (minutes)

Skill Practicing Time spent practicing hobbies (e.g., guitar, drawing)

Physical Endeavors Time spent on physical activity

Scrolling Time spent on social media or news feeds

Jorking Time spent on high-stimulus, low-reward activities

Passive Media Time spent consuming non-interactive content

Active Media Time spent on interactive entertainment (e.g., video games)

System Architecture Time spent on high-level cognitive tasks (planning, design)

B Internal State Dimensions

The 4-dimensional internal state vector:

Sleep Quality Subjective sleep quality, range $[-2, 2]$

Waking Energy Subjective energy level upon waking, range $[-2, 2]$

Dominant Emotion Valence dimension (negative to positive), range $[-2, 2]$

Emotional Intensity Arousal dimension (calm to activated), range $[-2, 2]$

C Interaction Matrix

The Interaction Matrix $A \in \mathbb{R}^{9\times4}$ maps internal state to behavioral preferences. Entries encode:

- $+1$: This internal dimension increases preference for this behavior
- -1: This internal dimension decreases preference for this behavior
- 0: No direct relationship

The matrix is currently hand-specified based on domain knowledge. Future work includes learning A from behavioral correlations.