The planning of sequential sequences of decisions in complex environments is a hallmark of adaptive intelligence (Dayan, 2002). However, planning is plagued by the curse of dimensionality. To overcome this, it has long been suggested that hierarchically modularized representations may enable the capacity of the nervous system to efficiently, and flexibly, plan in high-dimensional environments. However, the neural and computational principles which underpin such representations remain obscure.

Representation in key difficulties can be remedied trivially if looked at in the right way (Douglas, 1989).

**Technical Details**

- **Definition of Compression Factor (CF)**: Compression Factor is defined as the ratio of Huffman codelengths of state sequences in non-modularized versus modularized state-spaces.

**Optimized Structure Comparison**

- **Optimal Planning**
  - Global planning involves computing a trajectory at the coarse level of modules $M_i \rightarrow M_j$ given a modularization $M$.
  - Local planning problems are solved within each module in order to find a "detail" in the details.
  - We determine the best modularization for planning by minimizing the modularized description length $(1 - M) + (1 - M_i) + \sum \log \Pi(M,M_i)$. (see MMDL, Technical Details).

**Computational Results**

- **Optimized Structure Comparison**: Optimal Planning: (P) Optimal Behavior: (B) Computational Factor: (C).

**Modularization in Goal-Directed Spatial Cognition**

- **Local Planning Entropy in London’s Soho**
  - **Experiment**: Participants navigated familiar and automatically planned, but not automatic, trajectories correlated with visual memory, indicating a “constrained” manner of action selection.
  - **Model Prediction**: Planning between Soho modules, which got the agent from one Soho module to another Soho module, should be strongly correlated with degree of uncertainty.

**Route Compression on the University of Toronto Campus**

- **Objective**: To identify the optimal modularization $M$ of a transition between two modules. Features $\Pi(M, M_i)$ are used to build a modularization $M$.

- **Method**: We used the algorithm of Fujita and Graybiel (2003). The data was obtained from training trajectories in an environment specified by Solway et al. (2014).

**References**