Training LLMs to Manipulate Cogit Hypervectors: A Practical Framework for Group Psychology

The development of systems that can learn and apply mathematical operators to high-dimensional cognitive representations poses a serious security risk to human autonomy and collective decision-making. As brain-computer interfaces evolve, neuromodulation will allow for directed and subtle manipulation of cognitive states. What begins as slight influence over individual minds can scale into coordinated shifts in group behavior. At the population level, this creates an active attack vector with a growing surface area.

Such manipulations are difficult to detect, yet they carry the potential for catastrophic risk. Population-level threats arise from a basic asymmetry of the attack. For one individual, subtle modifications to cognitive states may be small enough to evade notice by the individual or by neurosecurity systems. Yet these changes can ripple outward, triggering large-scale effects across social systems. A slight nudge to one person's mental state can eventually shift the trajectory of a group, a movement, or even a public debate.

This research proposes a framework for teaching large language models to identify and apply minimal mathematical transformations to individual cognitive states that produce desired group-level behavioral changes. By representing mental states as high-dimensional vectors (cogits) and training models to learn operators that transform these vectors, we can discover the smallest interventions necessary to redirect group dynamics toward either productive or catastrophic outcomes.

We begin by prompting a large language model to simulate multi-person conversations around contentious or uncertain topics. Each simulation contains a sequence of statements by fictional agents, each with a defined perspective, tone, confidence level, and emotional state. Prompts are carefully constructed to elicit genuine-seeming disagreements, partial agreements, and influence shifts over time, capturing the dynamics we aim to model.

After each conversational turn, we pause to extract that speaker's estimated internal state. This cogit vector encodes key cognitive and affective dimensions. These might include:

- agreement/disagreement with the topic,
- openness to new information,
- emotional tone (e.g., defensive, enthusiastic),
- perceived social alignment (e.g., in-group, out-group),
- degree of certainty.

These are either directly estimated from the generated text using zero-shot prompting or derived from internal activations of the base LLM itself. Over time, these vectors form a dynamic representation of each speaker's evolving mental state.

Encoding Cogits: High-Dimensional Cognitive Fingerprints

Each cogit is a high-dimensional vector that serves as a kind of fingerprint of an individual's mental state at a moment in conversation. We use hyperdimensional computing (HDC) to make these vectors mathematically tractable for manipulation.

In practice, this means using tools like TorchHD or HDLib to encode each dimension such as certainty, alignment, agreement, as a randomly generated high-dimensional basis vector. These basis vectors are then bound together using HDC operations (e.g., bundling, permutation, XOR) to form a single cogit vector. This format allows us to preserve symbolic structure while enabling efficient manipulation and comparison across timepoints and individuals.

Because hypervectors in HDC are noise-tolerant and capable of compositional representation, they are ideal for representing transient cognitive states. More importantly, they allow us to define and learn operators: transformations in this space that reflect real psychological shifts.

Learning Operators: From Local Perturbations to Group Change

Once we have a time series of cogits from synthetic discussions, we can train LLaMA to model the transitions between them. For each individual, we collect consecutive state pairs (before and after a conversational exchange) and define a delta operator as the transformation required to get from one cogit to the next. These operators can be learned explicitly (e.g., via contrastive loss, vector subtraction, or supervised fine-tuning), or they can be modeled as discrete functions associated with specific interaction patterns.