ChatGPT

Cognitive AI Architecture Integrating Psychology and Movement – 11-Phase Research Plan

Phase 1: Cognitive Architecture with "Thinking Vectors" and Neuroplastic Knowledge Representation

Objectives & Theory: Phase 1 establishes a hybrid cognitive architecture that encodes "thoughts" in vectorial form (so-called *thinking vectors*) and supports neuroplastic adaptation of its knowledge base over time. The goal is to mimic the brain's ability to learn and rewire connections (synaptic plasticity) by using dynamic, reinforcement-driven updates to knowledge representations. The architecture should combine symbolic reasoning with subsymbolic embeddings, allowing rich conceptual representations that can be adjusted through feedback (much as biological neural networks strengthen or weaken synapses with experience). We draw on cognitive architecture theory: for example, ACT-R divides knowledge into declarative *chunks* (which are essentially vectors of property values) and procedural rules, with modular "buffers" interfacing to specialized brain-like modules [1]. This shows how human knowledge can be represented in a vector form and accessed via modular components. By adopting a similar modular design, our system can host multiple cognitive processes (memory, perception, planning, learning) that operate over a shared vector-space representation of knowledge.

Technical Implementation: A feasible implementation is to use a **neural-symbolic knowledge graph** with vector embeddings and weighted links, updated via reinforcement learning signals. For instance, OpenCog's Atomspace provides a "flexible and dynamic knowledge repository" – a weighted, labeled hypergraph that can store diverse types of knowledge (declarative facts, procedures, even emotions) in RAM for real-time reasoning [2]. Our architecture could use a similar graph structure where each node or subgraph has an embedding (the "thinking vector") that summarizes its content or context. Learning algorithms (e.g. backpropagation or Hebbian updates) would continuously refine these embeddings and link weights based on rewards or errors, achieving *dynamic reinforcement-based knowledge representation*. Neuroplasticity is emulated by allowing the graph topology to evolve (new nodes form, unused ones weaken and prune) and connection strengths to adjust when the system receives feedback. The system's design should integrate multiple AI paradigms to support general cognition – as OpenCog Hyperon does, integrating probabilistic logic, neural networks, and symbolic reasoning in one framework [3]. Key components include: (1) a **long-term memory store** (knowledge graph with vector embeddings) that supports insertion and retrieval of concepts, (2) a **learning mechanism** (e.g. a reinforcement learning module or an evolutionary algorithm) that alters connections and embeddings based on performance, and (3) cognitive processes (like reasoning or perception modules) that interface via shared representations. By Phase 1's completion, we expect to have a rudimentary "brain" capable of representing knowledge in a plastic, vectorized form and modifying itself – essentially the core engine for subsequent phases.

Key Research Questions: *How can we ensure stability–plasticity balance?* A challenge is avoiding catastrophic forgetting while enabling adaptation. Mechanisms like **gated plasticity** or neuromodulators could be explored (inspired by how dopamine signals in the brain regulate learning). *What is the optimal knowledge representation?* Do we use pure vectors (as in Vector Symbolic

Architectures) or structured graph + vectors hybrid? Comparative studies can be done where one variant uses high-dimensional vectors for concepts [4], while another uses graph-based encoding [2], measuring which better supports reasoning and learning. *How to incorporate reinforcement?* Investigate algorithms for graph reinforcement learning – e.g. using **policy gradients or Q-learning** to decide which connections to strengthen when reward is received for certain inferences. *Neuroplasticity emulation:* We can ask how closely the system mimics human learning curves or recovery from damage; experiments might involve "lesioning" parts of the knowledge network and seeing if the system can relearn functions (analogous to brain injury recovery). Additionally, the question of *vector semantics* arises: do these thinking vectors correspond to human-interpretable concepts? Probing methods (e.g. finding nearest words or visualizing dimensions) could be used to analyze what the AI's thought-vectors encode.

Related Work: Numerous cognitive architectures provide a foundation. **ACT-R** (Carnegie Mellon) is a classic architecture modeling human cognition with symbolic rules and subsymbolic activation values (it implements reinforcement-like learning via utility values for rules and base-level activation for memory chunks). **Soar** (Univ. of Michigan) offers a unified problem-solving architecture that could inspire our design of the cognitive cycle. More modernly, **OpenCog Hyperon** is directly relevant as an open-source AGI framework that "integrates diverse paradigms into a unified system" [5], including a self-modifying code capability (via the MeTTa language) for recursive self-improvement. We can also draw on **Vector Symbolic Architecture (VSA)** research [4], which explores encoding symbolic structures in high-dimensional vectors – this could inform how thinking vectors are structured and manipulated (e.g. binding and unbinding operations to compose concepts). In reinforcement learning, there's emerging work on combining RL with knowledge graphs [6], which we can leverage to design algorithms that adjust our knowledge network based on goal-directed feedback. Additionally, **neuroscience-inspired frameworks** (like SPAUN, a spiking neural cognitive model, or the **Free Energy Principle/Active Inference** models) might contribute ideas on how to incorporate plasticity and prediction-driven updates akin to human brains. All these works point towards systems that adapt and learn incrementally – crucial inspiration for Phase 1.

Potential Tools & Collaborators: We will likely implement the core in a language suitable for symbolic and neural mix – Python with frameworks like PyTorch (for neural components) and maybe a graph database (Neo4j or a custom graph) for knowledge. The OpenCog team (led by Ben Goertzel) could be a key collaborator, as their experience with Atomspace and cognitive synergies would accelerate development. Researchers in **computational neuroscience and cognitive science** (e.g. Randall O'Reilly's lab on biologically inspired AI, or teams working on **continual learning**) would provide insight into implementing plasticity without forgetting. We might also engage the **ACT-R community** (John Anderson's group) to understand lessons from decades of cognitive modeling – possibly using ACT-R's architecture as a plug-in module within our system for certain tasks. For reinforcement learning aspects, experts in **Deep RL** (e.g. DeepMind researchers who worked on integrating relational knowledge in RL) could advise on how to couple the knowledge graph with trial-and-error learning. Tools like **Open Neural Network Exchange (ONNX)** or **TensorFlow** might be used to integrate neural "thinking vector" models with symbolic code. Overall, Phase 1 will set up the technical backbone, so collaboration with both AI engineers and cognitive scientists is crucial to ensure the architecture is robust, extensible, and grounded in cognitive theory.

Phase 2: Modular "Plug-and-Play" Cognitive Models and Thinking Styles

Objectives & Theory: Building on the core architecture, Phase 2 introduces **plug-and-play cognitive modules** and predefined thinking styles. The vision is to have a **flexible cognitive framework** where

different reasoning modules or even personality-like configurations can be inserted or swapped without redesigning the whole system. The underlying theory draws from psychology's notion of **cognitive styles** (the habitual patterns or biases in thinking that vary between individuals) and personality traits. In humans, stable traits like extraversion or neuroticism manifest as biases in cognition and behavior, influencing how information is processed [7] [8]. We aim to model such variability by allowing the AI to adopt distinct "thinking styles" – for example, an analytical/logical style versus a creative/intuitive style – as modular configurations. Each style could be realized by tuning certain parameters or activating specific sub-modules in the architecture. The *objectives* are twofold: (1) **Foundation Models for Cognition:** Create a library of foundational cognitive models (rule-based reasoning, causal inference engine, probabilistic reasoning, etc.) that can be plugged in to handle different tasks or mimic different approaches to thinking. (2) **Style Profiles:** Define a set of cognitive style profiles – for instance, one might correspond to a bold, exploratory thinker (high novelty-seeking, low risk-aversion), while another to a methodical planner (high deliberation, reliance on logic). These profiles act as biases or weightings in the system's decision-making, essentially providing a *personality or thinking mode* that shapes how Phase 1's core processes operate. This phase leverages the idea that *traits are expressed as biases across multiple levels of cognition* [7] – from low-level neural parameters (e.g. learning rates, noise tolerance) to high-level goal preferences.

Technical Considerations: Implementing plug-and-play modularity means the architecture must have clearly defined interfaces between components. We might design the system as a hub-and-spoke: a central cognitive blackboard or message bus (the memory/knowledge base from Phase 1) to which various specialist modules connect. For example, we could have a **logical reasoning module** (perhaps a Prolog-based symbolic reasoner) that can be activated for deductive tasks, and a **visual imagination module** (maybe a generative neural network) for creative tasks. These modules should conform to an API to read/write from the knowledge base. Technically, we will need a **meta-controller** that can load or switch modules based on context or a user's selection of thinking style. For thinking styles, one approach is to implement them as parameter sets: e.g. a "fast, intuitive" style could lower the threshold for heuristic decisions and reduce the depth of search in problem-solving (analogous to Stanovich's System 1 thinking), whereas a "slow, analytical" style does the opposite [9]. Alternatively, styles can correspond to weighting different modules – an *emotional reasoning style* might rely more on an affect heuristic module, whereas a *numerical style* routes more queries to a math module. We'll also consider personality models such as the Big Five traits or Jungian cognitive functions: these can inform which parameters to tweak. For instance, a high **Openness** trait might correlate with a higher diversity in idea generation (we could increase random exploration factor in generation algorithms), while high **Neuroticism** might manifest as a bias to anticipate negative outcomes (we could tweak the reinforcement signal processing to give more weight to potential losses). The architecture should allow these configurations to be loaded dynamically – akin to loading different "profiles" or plugging in a new component at runtime – without retraining the entire system from scratch. This might involve **dependency injection** patterns or a plugin system where modules register their capabilities (e.g. one module declares it handles spatial reasoning, another handles social reasoning) and the cognitive controller can query the appropriate one when needed.

Key Research Questions: *How to define and quantify a thinking style?* We need a formal representation – potentially a vector of cognitive bias parameters – for each style. Research is needed to map psychological theories to implementable parameters. For example, can we derive from psychology studies what a "creative thinking style" means in terms of algorithmic differences (perhaps more random activation of remote associations)? *How modular can we get?* Is there a performance cost to high modularity? We should test scenarios where modules are swapped: does the system seamlessly continue functioning or are there integration issues (data format mismatches, etc.)? This raises the question of a **unifying knowledge schema** so that all modules, no matter how they implement internally, can communicate – likely our Phase 1 knowledge graph with a standardized ontology serves

this role. *How to enable learning of new styles?* In addition to predefined styles, the architecture could *learn* a style from examples (e.g. by analyzing how a human or another AI performs tasks, then adjusting its parameters to match that behavior). This suggests research into **meta-learning**: can the AI observe behavior traces and infer the cognitive biases that produced them? *Personality emergence:* If we run the system with continuous learning, will consistent patterns (a "personality") emerge on their own? Prior work suggests that stable traits might emerge from an adaptive system interacting with its environment [8]. We should monitor whether, for instance, our AI starts favoring certain strategies over time (which might indicate an evolving style) and whether we should allow that or enforce style reset for consistency. Another question: *to what extent can multiple styles coexist?* Perhaps the system can blend styles (e.g. 70% logical, 30% intuitive) – this becomes a design space to explore by interpolation between style parameter sets.

Related Work: The concept of modular minds has roots in theories like **Modularity of Mind (Fodor)** [10], and in AI, architectures like **LIDA (Learning Intelligent Distributed Agent)** have distinct modules for emotion, deliberation, etc., communicating via a global workspace [11]. Our plug-and-play approach extends this by adding *swapability*. The idea of cognitive styles is informed by **psychology** (e.g. Sternberg's thinking styles, or the **MBTI/Jungian functions** that Yat Malmgren incorporated into movement psychology [12] – specifically, Jung's four functions: Sensing, Thinking, Intuiting, Feeling). In AI, interest in personality modeling has grown: one study suggests building traits into architectures yields more coherent, believable agents [13]. Projects like IBM's **Cognitive Architecture for Personality Emulation** or the AAAI paper *"Designing Personality: Cognitive Architectures and Beyond"* (Matthews, 2004) discuss how traits can be implemented as biases at multiple levels (e.g. a trait influencing memory parameters, decision thresholds, etc.) [7] [8]. Also relevant is the work on **theory of mind and plug-ins for language models** – for example, adding a "Theory of Mind" module to an LLM as a plug-and-play component [14]. This showcases how a discrete cognitive ability can be turned on/off in an otherwise fixed architecture. We also look to **OpenCog's approach to cognitive synergy** where various AI algorithms (reasoners, neural nets, evolutionary learning) co-exist and can be enabled or disabled [5]. Our architecture's modular bus could be inspired by OpenCog's approach of a common knowledge base (Atomspace) that many cognitive algorithms share.

Potential Tools & Collaborators: We will likely use a **plugin framework** – possibly leveraging a component system such as ROS (Robot Operating System) if we treat each cognitive module as a service, or a more lightweight plugin approach in Python. Collaborators with expertise in **cognitive modeling** (such as researchers in the ACT-R community or those working on **MicroPsi** and **OpenCog**) can help define the modules and ensure psychological validity. We might partner with cognitive psychologists to get the parameter mappings for thinking styles (e.g. collaborating with personality psychologists who understand trait ontologies). On the technical side, teams like **IBM Research** (who have looked at neuro-symbolic AI and cognitive biases) or **AI labs focusing on multi-agent systems** could contribute modules. Tools for parameter tuning and meta-learning (e.g. **Ray Tune** for hyperparameter optimization) may be used to fine-tune style parameters to achieve target behaviors. For integrating modules, if using Python, libraries like **Pluggy** (used in Pytest for plugin management) or a microservice approach with gRPC could be explored. Additionally, open-source projects such as **MPI-Agent** or **JIFFY** (fictional examples for illustration) that allow dynamic loading of reasoning components might provide a baseline. By the end of Phase 2, we anticipate having a system where one can, for example, "swap in" a new planning algorithm or switch the AI's demeanor from logical to emotional by loading a different profile – a foundation that sets the stage for training the system in various domains in subsequent phases.

Phase 3: Training on Sensory Modalities (Visual, Auditory, Physical Inputs)

Objectives & Theory: Phase 3 expands the AI's scope to *embodied multimodal learning*. The objective is to train the model to process and integrate sensory data – images/vision, audio/speech, and physical inputs (which could include proprioceptive data or touch, if a robot body is involved). This draws on the theory of **embodied cognition**, which posits that intelligence arises from the interaction of mind and body with the environment. By exposing our cognitive architecture to raw sensory streams, we aim to have it learn grounded representations (e.g. the concept "dog" linked to seeing a dog image, hearing a bark, and touching fur). A key concept here is **multimodal representations**: the system should form internal representations that bind together features from different modalities into a coherent concept. Cognitive science tells us that human mental state is inherently multimodal – e.g. thinking of an "apple" might evoke its visual image, the crunching sound, the smell, etc., all integrated in memory [15] [16]. We want our AI to similarly associate sensory impressions. Another theoretical underpinning is the idea of an **episodic memory** of sensory experiences, as described by the integrative cognitive model in M3-Agent: it constructs episodic memory from "atomic multimodal events" (visual, auditory, etc.) which later support reasoning [17] [18]. Training on multiple modalities should foster robust representations that are closer to how humans understand the world, enabling cross-modal reasoning (like hearing a dog bark and visualizing the dog).

Technical Implementation: We will need to incorporate subsystems for each modality: e.g. a **vision encoder** (such as a convolutional neural network or Vision Transformer pre-trained on image recognition), an **audio processor** (possibly an audio spectrogram CNN or an existing speech recognition model for spoken language), and sensors for any physical data (for instance, a simulated agent might get inputs of coordinates, collisions, etc., from an environment). These subsystems feed into the cognitive architecture's memory. A straightforward architecture is to have each modality produce a vector embedding of the input (image → vector, sound → vector, etc.); these become part of the "thinking vectors" in the knowledge base. We'll train the system with multimodal datasets – for example, image–caption pairs, video with audio, or even data from a robot interacting with objects. A likely approach is **self-supervised learning** to create joint embeddings: e.g. using contrastive learning (as in OpenAI's CLIP, which aligns image and text embeddings). The model should learn that certain visual patterns correspond to certain sounds or textual descriptions. We also plan to use **multi-sensory integration networks** – perhaps an architecture where modality-specific layers feed into a fusion layer. The cognitive architecture might be extended with an "integration module" that takes input from vision, audition, etc., and updates the central knowledge graph with new entries (for example, seeing a new object creates a node with attributes from all senses). Reinforcement learning can be applied in environments where the agent must use sensory info to make decisions, reinforcing representations that lead to successful outcomes.

During training, we might present scenarios (either real-world or simulated) and have the AI narrate or classify them, storing episodes in memory. For instance, show a video of someone walking while playing footstep sounds – the AI should encode a unified memory of that event. Over time, it will accumulate a rich **episodic memory** of sensory experiences, which research suggests is fundamental to higher cognition and narrative (to be leveraged in Phase 4) [19] [20]. We will ensure that the model can do *cross-modal recall* (e.g. retrieve an image from an associated sound cue) as a test of integration. A technical challenge is the volume and heterogeneity of data; we'll likely need to use efficient data pipelines and possibly pre-train each modality subsystem on large data (like ImageNet for vision, LibriSpeech for audio) before fine-tuning the integration.

Key Research Questions: *How to achieve effective cross-modal fusion?* We need to determine the best architecture for combining modalities – early fusion (combining raw inputs), late fusion (combining high-level features), or a hybrid. Experiments should compare these. *Semantic alignment:* How do we ensure that different modalities that represent the same concept actually align in the internal representation? This relates to the **symbol grounding problem** – we'll examine whether the system's learned vectors for, say, the word "dog" and an image of a dog cluster together in vector space (indicative of successful grounding). *Temporal synchronization:* For physical input and continuous data, timing is key (e.g. a sound must align with video frames). We'll need research into sequence models (like LSTMs or transformers) that can handle synchronous sequences from multiple streams. *Scaling and memory:* As the system experiences more multimodal data, how do we store and index episodic memories efficiently? We might investigate **episodic memory modules** that compress experiences or **memory addressing schemes** that allow retrieval by content (e.g. "recall the sound of a dog" finds the memory with dog barking and associated visuals). *Evaluation:* We should set up tasks to quantitatively measure performance: for vision, classification and visual QA tasks; for audio, speech recognition or emotion detection; for integrated tasks, maybe VQA (Visual Question Answering) which requires combining image and text, or an embodied task like "find the red object that makes a noise" in a simulation. Additionally, user studies could evaluate if the model's perceptions align with human perception (e.g. does it confuse certain audio-visual pairings or does it learn natural correlations?).

Related Work: There is extensive research on multimodal AI. The **M3-Agent** framework (2025) is a prime example of a system integrating visual and auditory input to build long-term memory; it showed that such integration enabled more robust reasoning in video QA [21] [22]. Also relevant is DeepMind's **Perceiver** architecture, which is modality-agnostic and can ingest images, audio, etc., processing them with a unified Transformer – this inspires our attempt to have a unified latent space for all modalities. **Cross-modal interaction** studies in cognitive psychology [23] [24] support the idea that information from one sense can alter perception in another (e.g. the McGurk effect in audio-visual speech). We will keep this in mind, perhaps conducting analogous tests (does our AI experience a "McGurk effect" when we give conflicting audio and video?). The field of **robotics** provides relevant work: sensors fusion for robots (e.g. visual-servoing combined with touch sensors for grasping). For example, Matsumaru (2022) discusses applying Laban Movement Analysis and affect models to design emotional expressions in robot movement [25] – indicating a path for integrating physical embodiment with perceived emotion (which foreshadows later phases). Another noteworthy project is **IBM's Project Debater – Speech by Crowd**, which had to listen to many arguments (audio) and cluster them semantically; techniques from there might help our auditory semantic embedding. On the academic side, cognitive architectures like **Soar and ACT-R** have been extended with perceptual modules (ACT-R has a vision module that was used to model human eye movements, for instance). We will review such extensions to guide how our architecture's perception modules interact with central cognition (likely through buffers or working memory). Lastly, the multi-modal datasets like **CLEVR (for vision reasoning)**, **AudioSet**, and **KITTI (for physical environment sensing)** will be consulted or used to ensure we cover a broad range of sensory learning.

Potential Tools & Collaborators: To implement this phase, we'll use deep learning libraries for modality processing: e.g. **OpenCV** and **PyTorch** for building and fine-tuning CNNs on images, **Librosa** for audio feature extraction, and possibly **ROS** if we incorporate physical robot data. For integration, frameworks like **Tensorflow's Multimodal APIs** or **Hugging Face's Transformers** (which now support vision-language models) could be very useful. We might leverage pre-trained models: e.g. use CLIP's image encoder, use a Wav2Vec model for audio, and train a small integration network on top. Collaborators from **DeepMind or FAIR** who work on vision-language models could provide insight or even model weights. If available, we could collaborate with a robotics lab (e.g. at Georgia Tech or MIT) to gather physical interaction data – having a robotic arm that our AI controls to explore objects would generate rich multimodal data (vision of object, force sensors from touch, sound from interaction). Additionally,

experts in **cognitive neuroscience** (multisensory integration researchers) might join to compare the AI's integration patterns with fMRI results of human multisensory processing (checking if our AI develops something analogous to superior colliculus integration of senses). We also will need significant computational resources; partnering with an institution that has GPU clusters or using cloud platforms (AWS, GCP with multimodal pipelines) is expected. By the end of Phase 3, our AI should not just be a disembodied reasoner – it will have the beginnings of a **sensory cortex**, enabling it to see, hear, and feel (in a simulated sense) the world, setting a concrete stage for learning about social interactions and narratives in Phase 4.

Phase 4: Training on Interpersonal Interaction – Drama, Literature, and Narrative Understanding

Objectives & Theory: With sensory and basic cognitive abilities in place, Phase 4 focuses on *social and narrative intelligence*. The objective is to imbue the AI with an understanding of human interpersonal dynamics, stories, and drama by training on rich sources of narrative: plays, novels, movie scripts, dialogues from literature, etc. The underlying theory is that humans make sense of social life through **narratives and shared stories**; thus, an advanced AI should grasp concepts like character, plot, conflict, and emotion in a story context. We take inspiration from *narrative intelligence* – the capability to organize and explain experiences in story form [20]. Research has highlighted that narrative sense-making is central to human planning, social interaction, and coping with life [20]. So by training our AI on dramas and literature, we aim to equip it with models of human behavior and motivation. Specifically, we want it to learn: (1) **Dialogue and communication patterns** – e.g. how people take turns speaking, use subtext, express intentions or lies. (2) **Emotional arcs and relationships** – understanding friendship, rivalry, love, betrayal as they play out over a narrative. (3) **Narrative structures** – such as rising tension, climax, resolution (Freytag's pyramid) and common story archetypes (the hero's journey, tragedy vs. comedy patterns). (4) **Role-playing and perspective-taking** – the AI should be able to adopt the perspective of a character in a story and predict or imagine their responses (a primitive Theory of Mind). Psychologically, this training aligns with concepts from drama therapy and literature: e.g. *Stanislavski's methods* for actors to find objectives and through-lines, which essentially are ways to formalize a character's mental state and desires over a narrative.

Implementation Guidance: We will compile a large corpus of narrative data. This could include classic literature (which is often in the public domain), movie and theater scripts, transcripts of improvisational theater or role-playing games, and narrative transcripts from interactive fiction. The training process might involve fine-tuning language models on these texts so the AI picks up the style and content of interpersonal interactions. We'll leverage the architecture's multimodal and memory capabilities by also including **dramatic audio-visual datasets** (like movie scenes with both video and screenplay text) – allowing the AI to connect spoken lines with tone of voice and facial expressions (from Phase 3's skills). A likely technical approach is to use a **transformer-based model** (like GPT-style or BERT) fine-tuned on dialogue and narrative, but embedded within our cognitive architecture so that it doesn't just generate text but updates the agent's knowledge graph with narrative understanding. For example, when reading a story, the AI could populate a **story knowledge structure**: identifying characters, their relationships, events, conflicts, resolutions. We might implement a module for **narrative parsing** that uses natural language understanding to fill out a schema (somewhat like the old tale-understanding systems which used frames for events and scripts).

We will also incorporate known frameworks from narratology: e.g. **Vladimir Propp's morphology of folk tales** (which defines roles like Hero, Villain and functions like Trickery, Reward) could serve as a template for the AI to recognize plot elements. Similarly, Georges **Polti's 36 Dramatic Situations** is a catalog of fundamental conflicts in drama (like "Supplication", "Revolt", "Mistaken Jealousy") [26] – we can

encode these situations in the knowledge base and train the AI to classify story scenarios accordingly, giving it a structured lens on dramatic situations. This structured approach is supported by recent research: injecting such dramaturgical taxonomies provides strong guidance and prevents the AI from overlooking narrative elements like suspense or surprise [27] [28].

A key part of training will be **interactive simulation**: beyond passive reading, we want the AI to practice *interpersonal interaction*. We can set up a simulated environment (text-based or even using our Phase 5 environment to come) where the AI takes on a character and interacts with another agent or a human in a role-play. This reinforcement learning scenario would force the AI to apply narrative understanding in real time – e.g. improvising dialogue consistent with a character's motivations. Techniques from **dialogue management** and **affective computing** come into play: we might use reinforcement signals such as "how engaging is the story the AI is producing?" or "did it maintain character consistency?" to refine its behavior. Additionally, the AI's episodic memory (Phase 3) can be leveraged to store **narrative memory** – it should remember previous interactions or chapters of a story to inform future ones, enabling coherence over long dialogues or multi-act plays.

Key Research Questions: *Comprehension vs. generation:* We need to assess how well the AI is understanding narratives, not just generating plausible text. We can pose questions (who did what and why) to ensure it's building a correct causal model of the stories. *Narrative structure learning:* Will the AI spontaneously learn constructs like plot tension or will we need to explicitly encode a "plot monitor"? We might measure the AI's output for features like the presence of a clear climax or resolution when asked to generate a story. *Emotional understanding:* Does the AI correctly interpret and convey emotions in interpersonal contexts? For example, if in a story a friend betrays the protagonist, does the AI register the appropriate emotional significance? We can test this by asking it to predict character emotions or by checking if it uses emotion-laden language in summaries. *Bias and cultural understanding:* Literature comes with cultural context. We must check for biases the AI might pick up from older literature or misinterpretation of social norms (e.g. Shakespearean characters vs. modern ones). A question is how to make the AI adaptable to context – perhaps via meta-data tags indicating the setting (era, culture) of a narrative. *Integration with earlier phases:* We should examine how multimodal perception aids narrative understanding. For instance, if the AI *sees* a scene of two people hugging versus arguing, does it integrate that with textual descriptions of reconciliation or conflict? Also, can the AI correlate tone of voice (from audio) with the subtext of dialogue? These cross-modal narrative cues are advanced but worth researching (somewhat akin to how humans watch a movie and read body language plus dialogue to grasp the full meaning).

Related Work: In the AI research community, **computational narratology** is a rich field. Projects like *Stanford's NarrativeQA* and *Story Cloze Test* have been used to evaluate story comprehension and commonsense endings. Our training will benefit from these benchmarks to gauge progress. The classic project **TALE-SPIN** (James Meehan, 1977) attempted to generate simple fables by giving characters goals and simulating their attempts – it underlines the importance of characters' internal states and goals, which in our architecture are modeled via those inner drives (Phase 7 will delve deeper into that). More modern, the **METATRON framework (2023)** for story generation is directly relevant: it combines a symbolic planner for dramatic situations with neural text generation [29] [30]. METATRON explicitly uses Polti's dramatic situations and checks for coherence, suggesting that a hybrid approach (like ours) is state-of-the-art. We will likely collaborate ideas with the METATRON authors or adopt their approach of symbolic "story outlines" that the AI then elaborates. Another relevant line of work is on **character modeling** in interactive narratives: the FAtiMA toolkit (FearNot! project) which gave agents emotions and goals in a story world, or the Comme il Faut system for social dynamics. These offer insight into how to computationally represent things like social status, relationships, and dramatic actions (e.g. forgiving someone vs. seeking revenge).

From psychology, **dramatic theory** provides the concept of *super-objectives* (from Stanislavski) and *beats* in a scene; we'll incorporate those in Phase 8, but they influence training here by focusing the AI on the fact that characters pursue objectives. Also, literary analysis techniques, such as identifying themes or moral lessons, could be introduced to the AI's training as higher-level tasks (for example, ask the AI to state the theme of a story – a test of understanding beyond surface).

Potential Collaborators, Tools, and Frameworks: We will need **NLP tools** specialized for narrative. The **Natural Language Toolkit (NLTK)** or newer libraries can be used to do things like coreference resolution (to track characters), sentiment analysis (to gauge mood of scenes), and dialogue act tagging. We might use a knowledge representation for narrative such as **PropBank or VerbNet** frames to parse actions in stories. Collaborators from the **Storytelling AI community** (e.g. researchers like Mark Riedl at Georgia Tech, who works on story generation, or Michael Mateas at UC Santa Cruz, known for interactive drama like *Façade*) would be invaluable. They can provide both theoretical frameworks and possibly datasets (like the Story Commonsense dataset or Persona dialogues dataset). We might also seek partnership with dramaturges or literature scholars to evaluate the AI's story understanding; for example, asking an expert in Shakespeare to see if the AI grasps the motivations in *Hamlet*.

On the technical side, frameworks like **OpenAI's GPT-4 (if available via API)** could be fine-tuned or used as a component for high-quality language generation – we might integrate it as a "narrator module" that the rest of our cognitive system calls upon to verbalize internal narrative understanding. Additionally, **knowledge graph** tools can be used to keep track of narrative states: we could use an RDF triple-store or an ontology (there's an ontology called Dramaont for plot elements we could adapt). For simulations, if we create a virtual environment for role-play, we might use a text-based platform (even a multiplayer text adventure setup) or game engines like Unity with simple avatar interactions (text output for dialogue). By the end of Phase 4, the AI should have a rich training in the "soft skills" of intelligence – understanding people, plots and emotions – which sets the stage for *actually creating* and enacting such stories with characters (Phases 5 and 6).

Phase 5: Character Generation and Simulated Environments

Objectives & Theory: Phase 5 moves from understanding to **enactment**. The AI will generate characters – essentially autonomous agents with distinct personalities and cognitive profiles (leveraging Phase 2's thinking style modules) – and place them in simulated environments to interact. The objective is to test and refine the cognitive architecture in a controlled, observable world, and to enable emergent behavior that can lead to unscripted story development. This phase is grounded in theories from both AI (multi-agent systems, emergent behavior) and drama (improvisational theatre, LARP – live-action role-play, etc.). By generating characters, we mean giving the AI the ability to spawn agents each with specific attributes: e.g. a backstory, goals (super-objectives), relationships, and perhaps movement styles (from later movement phases). We then simulate an environment (could be a text-based sandbox or a simple 2D/3D world) where these agents "live" and interact, producing dynamic narratives.

The theory is that through *simulation*, complex cognitive and social phenomena can be observed and studied. This aligns with research in artificial life and agent-based modeling, where simple rules at individual level can yield complex group behavior. Here, our characters are quite complex agents (with our full cognitive architecture inside each), so we expect rich interactions. Psychologically, this phase allows exploration of how different cognitive configurations lead to different behaviors in the same situation – akin to an experiment where two people respond differently to the same stimulus. We also draw on **game design** theory: in games like The Sims, characters have needs and motives and interact in an environment; players often observe emergent storylines. We aim for a similar effect but under the AI's autonomous control.

Technical Implementation: We will set up a **sandbox environment**. Initially, it could be text-based (like a shared narrative where agents take turns describing actions, useful for quick iteration). Eventually, we might use a simple graphical environment – possibly tile-based 2D (for ease) or a constrained 3D environment like a small town. In fact, the recent Stanford Generative Agents experiment provides a template: they created a "Smallville" town with 25 agents who simulate daily life [31]. We would implement something along those lines, perhaps with even more cognitive depth per agent. Each character agent in the simulation will be an instance of our Phase 1-4 cognitive architecture, with a distinct personality profile (Phase 2) and knowledge base seeded with that character's backstory and traits.

Agents will perceive the environment (using Phase 3's sensory modules, e.g. seeing the world state, hearing messages), and interact (e.g. move around, speak via a dialogue system, manipulate objects). We will include a **scheduler or real-time loop** to manage simulation time and events. A crucial component is ensuring the agents have motivations: we'll assign each character super-objectives and shorter-term goals (these come from Phase 8's design but at this stage they can be simpler like "go to work", "cook dinner", etc., as in The Sims). We will monitor the simulation and possibly intervene or prompt scenarios (like introducing an event: "it starts raining" or "a festival is happening") to see how agents react.

One important technical aspect is **scalability**: running many cognitive agents in parallel is computationally heavy. We might simplify by running at a slower simulation clock or offloading some cognition to a central server that can sequentially step through agents' decision cycles (rather than true parallel). Alternatively, use a more lightweight reasoning for background characters and only fully simulate a few main characters. The environment's physics and rules will be simplified (we're more interested in social interaction than realistic physics).

We'll integrate a dialogue system for conversation between agents (could reuse the narrative-trained language model from Phase 4 for generating dialogue lines appropriate to each character's knowledge and mood). The memory and episodic recall become critical here: agents should remember past interactions. For example, if Agent A helped Agent B on Monday, on Tuesday B should recall that favor and have some trust in A. Our architecture's memory graph can encode these relationships (edges like *helped(A,B)* with a valence).

Key Research Questions: *Emergence:* What kinds of emergent social phenomena appear? We'll watch for things like formation of friendships, cliques, rivalries, or even economic exchanges. Is there coherence to the emergent narratives, and do they resemble human-like social patterns? This is an open-ended question – success would be seeing believable stories unfold that were not explicitly scripted. *Validation:* How to validate that the characters are behaving "as intended"? For each generated character, we have an intended personality and goals. We need metrics or observations to confirm these are manifest. E.g., if a character is meant to be extroverted and agreeable, do they initiate conversations frequently and respond kindly? We can log dialogue acts and measure frequency or sentiment as indicators. *Coordination of agents:* Multi-agent systems notoriously face coordination challenges (agents might talk over each other or all pursue the same object chaotically). How do we manage turn-taking and focus in interactions? Possibly implement a simple **social attention model** – at any time, an agent focuses on one interaction. Or use an environment moderator that queues events. *Story coherence:* Do these emergent interactions produce a coherent story if observed by a human? We might recruit human evaluators to watch a simulation and rate how coherent or engaging the resulting story is. This informs improvements like adding a "narrative coherence monitor" (maybe a module that nudges agents to follow up on unresolved events, akin to a game Dungeon Master). *Agent autonomy vs. authorial control:* There's a balance between letting agents freely act and steering them to maintain interesting outcomes. Research question: can we have a meta-agent (or an adjustable parameter) that

controls the degree of autonomy? Perhaps occasionally, to avoid boredom, an agent gets a "dramatic impulse" injected (like a sudden feeling or idea) to stir the pot – this could be random or guided by dramatic principles from Phase 6.

Cross-Phase Interplay: This phase will put to use everything from Phase 1-4. For instance, Phase 2's plug-and-play design lets us rapidly create diverse character minds by mixing modules (we might give one agent an extra emotion appraisal module, another agent a strategic reasoning module, etc.). Phase 3's sensory training is utilized if the environment is visual or spatial – agents seeing objects or hearing sounds in the sim. Phase 4's narrative understanding helps agents interpret events (e.g. recognizing a betrayal or a romantic gesture in the simulation) and respond appropriately in a story sense.

Related Work: The **Stanford Generative Agents (Park et al., 2023)** is highly relevant: they found that with an LLM as the backbone, agents can exhibit believable daily behaviors and remember and plan, leading to outcomes like organizing a party [31]. They reported that crowdworkers found the agents' behavior more believable than humans pretending to be those agents [32], which is an inspiring result to match or exceed. Our approach differs in that we have a full cognitive arch, not just an LLM prompt, but we will leverage their findings on memory (they used a "memory stream" where agents store observations and reflections, then retrieve them to plan actions [33]). We might use a similar memory retrieval mechanism to drive our agents' actions [33].

Another area of related work is **multi-agent reinforcement learning** in simulated environments (e.g. OpenAI's hide-and-seek agents or Google's AI Soccer). Those focus on physical or game-theoretic emergent behaviors. Ours is more cognitive and social, but techniques like self-play and curriculum learning from those could be applied – e.g., progressively increase the complexity of social dilemmas agents face to improve their robustness.

In the gaming industry, **The Sims** and RPGs provide heuristic character AI (with need-based drives, etc.). We can compare our agents' spontaneous behavior to The Sims; does our architecture produce more life-like and less repetitive behavior? Possibly yes, since it has deeper memory and narrative sense. Academic projects like **Comme il Faut** (CiF) explicitly modeled social state and let agents perform "social moves" (insults, compliments, etc.) in a simulation. We will take cues from CiF's social physics to define what kind of actions our agents can take (they should be able to do things like gift-giving, apologizing, demanding, etc., not just move or speak).

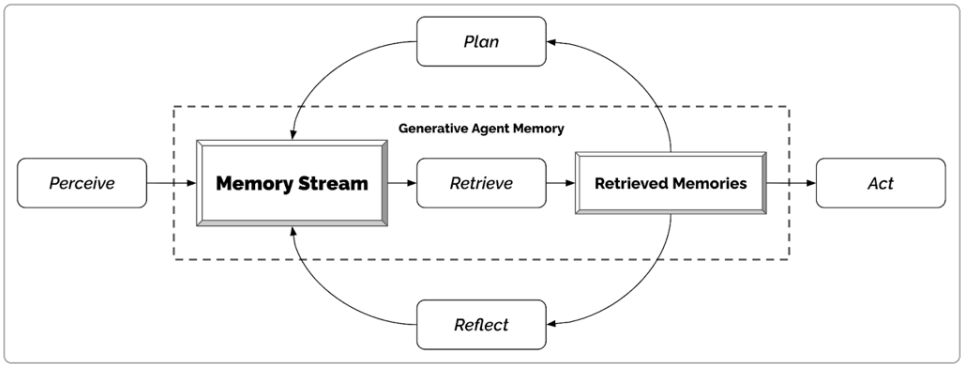
Potential Collaborators, Tools & Frameworks: We will likely develop the simulated environment using a game engine or simulation framework. **Unity** or **Unreal Engine** can be used for 3D environments, though a 2D or textual environment might be done with simpler tools (even a web-based simulator). If using Unity, its ML-Agents toolkit could facilitate embedding our AI brains into game characters. Alternatively, Python frameworks like **MESA (Mesa is an agent-based modeling framework)** could allow building a social sim with visual output. For character animation (if we present graphically), our later movement integration (Phase 7 onward) will come in, but in this phase, we could use placeholder animations or simple sprites.

Collaborators could include **game AI developers** – people who have built NPC behavior systems. They could help us integrate our cognitive models with the technical constraints of a game engine. Also, researchers from **virtual human simulation** (like USC's ICT which made virtual reality training characters) might provide insight into maintaining believability. On the evaluation side, we might engage sociologists or psychologists to analyze the social dynamics that emerge (there is precedent: some studies have used AI agents to simulate theories of sociology or organizational behavior). Their feedback can inform if our simulation is realistic or if agents are too rational or too chaotic compared to real humans.

We will also need to manage data: logs of everything agents do will be huge. Using **analytics tools** (even as simple as Jupyter notebooks to parse logs, or more complex like event databases) will be necessary to sift through and find interesting patterns or problems. Perhaps implementing a visualization of social network evolving over time (who is friends with whom, who dislikes whom) would be a great diagnostic – tools like **NetworkX** in Python can help graph relations.

By the end of Phase 5, we expect to have a "living" sandbox where one can observe multiple AI-driven characters interact. This provides a testbed for the next phases, which will add explicit dramatic drives (Phase 6) and movement expression (Phase 7) to these characters, refining both their internal motivations and external portrayals.

Example of a simulated environment with multiple generative agents interacting (Stanford's Smallville scenario). Each character (labeled with initials) carries on daily activities and social interactions autonomously, illustrating the kind of multi-agent environment Phase 5 will create. [31] [32]



Memory–action loop for a generative agent (from Park et al. 2023). Agents perceive the environment, store events in a memory stream, retrieve relevant memories, then reflect and plan before taking action [33]*. This aligns with our architecture: each character agent will continually update and consult its memory (Phase 3) to guide behavior, ensuring consistency and the ability to plan over long simulations.* [33]

Phase 6: Implementing Dramatic Drives for Story Generation

Objectives & Theory: Phase 6 adds a **narrative engine** on top of the multi-agent simulation: *dramatic drives* are introduced to actively shape story development. While in Phase 5 agents simply pursue their own goals which can incidentally produce stories, here we explicitly incorporate principles of drama to drive the system towards compelling narratives. The objective is to generate stories (either through simulation or direct narrative output) that have *dramatic structure* – meaning they contain conflict, tension, character growth, and resolution akin to authored stories. We borrow from **dramatic theory** and narrative psychology: in literature, stories are often driven by fundamental conflicts or desires (love vs duty, ambition, revenge, etc.). These can be seen as abstract *drives* that push characters into action and confrontations. In this phase, we ensure our AI agents (or a centralized storyteller module) have such dramatic drives influencing their behavior.

A key concept is integrating **Polti's 36 Dramatic Situations** and other narrative archetypes into the AI's motivational system. For example, a drive might be "Revenge" or "Romance" that the AI can instill in a character or the scenario. Another concept is **dramatic tension**: the AI should monitor and seek to increase tension up to a climax then release it (resolution), to mimic story arcs. Psychologically, one can think of dramatic drives as akin to *Jungian archetypes or Campbell's hero's journey stages* – deep patterns our psyche resonates with. By coding these into the AI, we hope to get more meaningful stories rather than aimless agent wanderings.

Implementation Guidance: There are two possible approaches (which can be combined): (1) **Internal drives in agents:** We modify the agents from Phase 5 to include not just personal goals but dramatic goals. For instance, a character might be assigned a *tragic flaw* or an *inner conflict* (say the drive to seek revenge vs the pull of their conscience). This could be done by giving the agent two competing objectives and a mechanism (like a utility or emotion model) that creates struggle. Agents would then enact scenes that externalize this inner drama. (2) **Central narrative orchestrator:** We could implement a "drama manager" that sits above the simulation, observing the global story and nudging it. This drama manager might trigger events or steer characters to ensure that dramatic situations occur. For example, if things are too calm, it might introduce a problem (a sudden illness, an antagonist arrival) aligning with one of Polti's situations to raise stakes.

From a technical perspective, representing dramatic drives could mean adding to the knowledge graph a notion of *super-objectives and conflicts*. We might have a structure like: Character X's super-objective = "secure the throne", conflicting with Character Y's objective = "protect rightful king". These create a classic drama (usurper vs protector). We will use the narrative knowledge and structures from Phase 4: for instance, knowledge of common plots can be encoded as templates. We can formalize a dramatic situation with roles and required elements (Polti's catalog provides roles and dynamics [26]). Our system could periodically assess which dramatic situation the current state most resembles and then *commit* to it by encouraging actions that fulfill that situation's structure. Recent work in story generation suggests using such high-level schemas improves coherence and creativity [30] [34].

Concretely, we might incorporate a **planning algorithm** for drama: something like a forward search or HTN (Hierarchical Task Network) planner but for plot. Given an initial state and a desired dramatic outcome (e.g. "Mistaken Jealousy" scenario), the planner can outline steps: a misunderstanding occurs, jealousy grows, confrontation happens, then realization. The agents can be guided (through subtle modifications of their goals or injected thoughts) to follow these steps. This hybrid of planning and simulation is akin to *interactive narrative systems* where a drama manager ensures that player actions still result in a good story.

We also implement **dramatic drives as motivational signals**: e.g., add a reward in the RL sense for actions that increase conflict or drive the narrative forward. An action that drastically changes the status quo (like an agent revealing a secret) could be given a positive "narrative reward" because it makes the story more interesting. Conversely, if the story stagnates, the narrative engine might give a slight negative reward to encourage agents to do something surprising. This essentially treats story generation as an optimization problem: maximize some function of dramatic tension, coherence, and surprise [35] [28].

Key Research Questions: *Measuring story quality:* We need a way to measure how dramatic or engaging a story is algorithmically to drive learning. Some possibilities: story coherence metrics (does it have a clear beginning-middle-end), compression-based surprise metrics (originality of events) [36] [37], or even learned critics (train a model on human-rated stories to score our outputs). We will investigate these to use as objectives for the drama manager or for reinforcement learning. *Balance between structure and freedom:* If we impose too strong a template (like strictly following Polti #5 Pursuit), the story might become formulaic or agents might behave unnaturally. If too loose, we may get meandering stories. Research into **blended storytelling** – partially guided, partially emergent – will be needed. Perhaps dynamically switching drives or templates if the story veers off course, maintaining flexibility. *Generalization:* Can a small set of dramatic drives yield a wide variety of stories? Likely yes, since 36 situations already cover many stories. But we should test combining them (multi-situation stories like subplots) and see if the AI can handle that complexity. *Role of user input:* If this system is eventually interactive (like a human could steer or participate in the story), how do dramatic drives adapt to user interventions? We might simulate that by random disturbances and see if the system can re-organize a coherent drama (similar to an improvisational troupe incorporating audience suggestions). *Emotional authenticity:* As drama is highly emotional, we need to ensure the AI's portrayal of emotions under these drives is convincing and not overly melodramatic or flat. This ties into Phase 7 (movement expression) but also text/dialogue expression. We could evaluate via human judges whether the emotional beats of the story resonate.

Related Work: The idea of a *drama manager* has a long history in interactive narrative (e.g. in Facade, an AI-driven interactive drama, a drama manager monitored and adjusted pacing). Our work is similar but with potentially more autonomous characters. The **Erik Berseth et al. "Story Cloze" projects** looked at how narrative arcs can be learned; also, Roger Schank's concept of "scripts" in AI (though for mundane scenarios) parallels giving agents scripts for dramatic scenarios. In terms of drama theory, **Stanislavski's super-objectives and beats** come into play: each scene has beats (units of conflict) and each character has an overarching through-line. Phase 8 will align with that explicitly, but here we implicitly use it by ensuring each character's actions serve some higher dramatic purpose.

There's also relevant computational creativity research: **Villaneau's work on surprise and suspense modeling**, *chekhov's gun principle detection*, and so on. We may incorporate formal models of suspense (e.g. measuring the audience's uncertainty about outcomes) as part of the narrative reward. Notably, the METATRON framework we discussed uses a neuro-symbolic approach to ensure things like theory-of-mind consistency and emotional arcs [38] [39]. We will borrow such mechanisms: for instance, tracking what each character knows and ensuring we can create dramatic irony (audience knows something a character doesn't – the AI can exploit that by planning reveals).

Additionally, Polti and Propp we already cover, but another is **the Hero's Journey (Campbell)** – while less formalized, we could incorporate some of its stages (call to adventure, abyss, return) as drives for an overall narrative trajectory for a main character.

Potential Collaborators, Tools & Frameworks: Collaborating with storytelling technologists or narrative designers will be important. For example, experts from the **Interactive Narrative**

community (Association for Computational Machinery's workshops on Narrative) or companies like *Massive Entertainment* that work on emergent narrative in games. Tools that could help: **propagation networks** to handle multi-agent belief and knowledge tracking (for managing dramatic irony and misunderstandings), planning tools like **SHOP or PDDL** planners for story domain (some research projects treat story generation as planning – we could use those libraries with modifications).

We might also integrate with **LLM-based tools** for inspiration: e.g., use GPT-4 to brainstorm possible dramatic events at certain points, essentially acting as a co-author. Our system could query "What could go wrong now?" and get an idea to implement. This hybrid approach could mitigate our system's possibly narrower generative diversity, by adding the breadth of a large language model's knowledge of tropes.

Finally, evaluation will likely involve human evaluators in the loop (there's no better judge of story quality than humans). So setting up a pipeline where we generate multiple story variants and have people rank them (perhaps via Amazon Mechanical Turk or a panel of experts) would be useful. We can then correlate those rankings with our internal metrics to refine how we weight dramatic drives.

By the end of Phase 6, our AI should not only simulate life but *craft stories*: sequences of events with intentional dramatic shape. This sets the stage to connect the internal cognitive-emotional engines we've built with physical expression (Phase 7) and to ensure the motivation system (Phase 8) aligns with these dramatic imperatives.

Phase 7: Integrating Inner Cognitive Engines with Movement Expression (Malmgren's Movement Psychology)

Objectives & Theory: Phase 7 connects the AI's *inner world* with *outer expression in movement*, guided by Yat Malmgren's Movement Psychology (which is itself based on Rudolf Laban's movement analysis and Carl Jung's psychological typology). The objective is to ensure that the cognitive and emotional state of our AI agents (the "inner engines" driving them, such as their emotions, objectives, dramatic drives) are consistently and meaningfully expressed through their physical movements and behaviors. In essence, Phase 7 is about giving the AI a **body language** and physical presence that reflects its mind – a crucial aspect for believability if these agents are embodied (in a simulation or eventually a robot/animation).

Malmgren's system posits a direct link between **mental factors (Jungian psychological functions)** and **motion factors (Laban's movement qualities)** [12]. For example, a thinking-oriented personality might have a different movement style than a feeling-oriented one. The system develops a classification of character into *Inner Attitudes* (psychological drives) and *Action Attitudes* (expressive movement patterns) [12]. We will adopt this theoretical mapping: each agent's cognitive profile and current state (its inner attitude – e.g. calculating vs impulsive, depressed vs joyful) will map to movement parameters (its action attitude – e.g. sudden or sustained timing, direct or flexible spatial use, strong or light force, etc., as per Laban's Effort qualities). The integration aims to achieve what actors train for: **embodied emotion** – the AI should "act" with its body consistent with its inner feelings, making it appear more lifelike and aiding communication (even if not explicitly instructed, an angry AI should move in a "punching" manner, a sad one in a "draining" manner, etc.).

Technical Implementation: To implement this, we must expand our agent models to include a representation of movement qualities. Laban Movement Analysis (LMA) provides a vocabulary: the Effort factors (Space, Weight, Time, Flow) each with two poles [40]. We can encode an agent's current movement state as a point in this Effort space (for instance, an agent might currently be in a state of

Press (Direct, Sustained, Strong, Bound) or **Glide** (Indirect, Sustained, Light, Free), etc.). As the agent's inner state changes, we will *dynamically update* this Effort point.

One way is to create a mapping table or function: e.g., map high arousal + positive emotion to "Buoyant/Light/Floating" movements, high arousal + negative to "Quick/Sharp/Thrusting" movements, etc., consistent with Laban's associations of Effort with emotion. There's established correspondence: LMA experts often correlate *Float, Glide, Dab, Flick* with more indulging/less fighting efforts (lighter, free) and *Punch (Thrust), Press, Wring, Slash* with more fighting efforts (stronger, bound) [41] . We also consider Jungian functions: Malmgren linked Jung's four functions (Thinking, Feeling, Intuition, Sensation) to movement tendencies. For example, Thinking might align with more direct, bound movements (controlled), Feeling with flexible, fluent ones (expressive), etc., and these can define a character's baseline movement signature.

Practically, if our simulation is graphical, we'd have animations or procedural movement generation for each Effort quality combination. We might implement something like: an agent's velocity, posture, and gesture animations are modulated by these parameters. For instance, an "indirect, light, sustained" movement (Float) could mean the agent's path curves gently, speed is moderate, and footsteps are soft; while a "direct, quick, strong" (Punch) means straight-line movement with sudden bursts and forceful gestures. If the simulation is text-based, we might describe movements in narration (e.g. "Alice storms into the room" vs "Alice glides into the room") based on the movement quality.

We will build an **expression module** that continuously reads the agent's internal state (which includes emotions from Phase 4's interpersonal engine, drives from Phase 6, etc.) and computes an Effort configuration. This acts like a filter on all motor actions. If an agent decides to go from point A to B, the path, speed, and manner will be determined by current Effort settings rather than default. Over time, as the agent's inner state shifts (maybe they calm down, or get excited), the Effort will shift accordingly, leading to changes in movement dynamics on the fly. This implements the "inner engine to movement" integration.

We will test this with scenarios: for example, an agent that is nervous (inner state: high anxiety) might show fidgety, quick, bound movements; if it then achieves relief (state changes to calm), its movements should visibly smooth out and slow. We'll ensure the mapping can handle mixed states – possibly blending Effort qualities.

Key Research Questions: *Mapping accuracy:* Does our mapping of internal states to Laban Effort truly convey the intended psychological state to observers? We will likely run user studies: show observers animations of an agent's movement (without dialogue) and ask them to infer the emotion or intent. High accuracy would validate our mapping. If not, we refine the mapping or consider more movement descriptors (like posture, shape change from LMA's Shape component). *Dynamic transitions:* How to smoothly transition between Effort states? We might use interpolation in the Effort space – e.g., an agent goes from Press to Glide through intermediate states. We need to avoid jarring switches unless dramatically appropriate. *Multiple concurrent states:* Humans can have layered emotions (angry but also sad, etc.). LMA's Effort can capture some layering (Flow is often seen as underlying emotional flux, while Space/Weight/Time are more intention). We should decide if we allow partial activation of multiple Effort qualities or restrict to a primary one. Possibly use *Effort States* (combinations of two factors) for subtle states [42] . *Control vs. autonomy:* Should agents consciously control their movement style for effect? (E.g., a character might feign a confident walk despite fear). That's an advanced concept – an agent could have a tactic to mask true inner state. We might not do that in Phase 7 unless we have a deception module; but it's interesting to consider for later, adding complexity that movement isn't always veridical of inner state (just as humans can lie with body language, though imperfectly). *Physical constraints:* In simulation or robotics, can all desired movements be realized? We might find that certain

Effort combos are hard to animate without a rich motion library. We might need to generate approximations. Research in **procedural animation** or using ML models like our Phase 10 approach (motion generation via diffusion or HMM) will help ensure continuity and physical plausibility of these movements.

Related Work: Yat Malmgren's Movement Psychology is our primary guide; it was used to train many actors with great success in creating distinct characterizations (famous alumni like Anthony Hopkins use it, indicating its effectiveness). The literature describes how Laban's Effort and Jungian types combine to form something called an **"Inner Attitude"** for a role (like "Driven by Intuition, expressed in Flexible movement" etc.). We'll refer to Mirodan's analysis [12] for specifics. Also, Rudolf Laban's own work *Mastery of Movement* and others which detail how Efforts relate to emotion will be referenced.

In technology, **Biorobotics and HRI (Human-Robot Interaction)** have explored using Laban qualities in robot motion to convey emotion (e.g., researchers have programmed robots to move with Laban Effort to appear happy or sad) [43] [44]. A 2020 paper by Samadani et al. used Laban Effort and HMMs to generate affective motion, showing that people could recognize target emotions from those motions at ~72% accuracy [45] [46]. This suggests that with proper encoding, movement can effectively communicate inner state. We will leverage their approach for evaluating our system. Also, a 2025 work by Kim et al. integrated Laban Effort into a text-to-motion diffusion model to allow expressive control of generated animations [47] – we might not use a diffusion model in real-time, but their success implies we can computationally manipulate Effort dimensions to get desired motion qualities.

There's also decades of animation practice: Disney animators had the 12 principles of animation (e.g. exaggeration, timing) which parallel some LMA ideas. Our movement expression should incorporate such principles to be visually readable.

Potential Tools & Collaborators: We will need animation tools. If we are using Unity or a similar engine, we can create or obtain animation clips exemplifying each Effort quality. Possibly use motion capture: have actors trained in Laban Efforts perform basic actions (walking, reaching, gesturing) in the eight Effort Actions [41] and use those as base animations. If resources allow, collaborating with a **movement coach or choreographer** familiar with Laban/Malmgren (perhaps from a theatre school or performance research group) would greatly help – they can validate if the agent's movement looks right for the intended attitude.

We could also collaborate with robotics researchers (like the Waseda University group or media labs) who implemented emotional gait for robots, to exchange methodologies. For computational mapping, tools from **affective computing** like the Geneva Emotion Wheel or circumplex models might be integrated – we could map points on an emotion wheel to Effort qualities, for instance.

In terms of software, if going the procedural route, using inverse kinematics and physics in the engine to modulate movement might be needed (for example, adjusting joint stiffness for Bound vs Free flow). There are libraries for motion interpolation and blending (like in Unity's Mecanim system) that we'll use to blend between animation states smoothly.

To summarize, Phase 7 ensures that when our AI agents are "on stage" in the simulation, they *behave* like actors: their thoughts and feelings are legible through their movements. It bridges the gap between cognitive intent and physical performance, a critical step for a truly lifelike AI. This sets us up for Phase 8, where we will circle back to ensure those inner engines (drives, goals) are properly structured (super-objectives), and Phase 9, where we refine how many of these movement dimensions can be active simultaneously as per Laban's theory.

Phase 8: Alignment of Inner Engines with Super-Objectives to Shape Motivation

Objectives & Theory: Phase 8 revisits the **motivation system** of the AI (the "inner engines" such as goals, drives, needs) and aligns them with the concept of *super-objectives*, a term from Stanislavski's acting methodology meaning the overarching goal driving a character through the entire narrative. The objective here is to impose a coherent hierarchical structure on the AI's motivations: every small action or objective should in principle serve a larger goal (the super-objective), providing consistency and direction to the agent's behavior over long time scales. This ensures that the characters don't just act on moment-to-moment whims, but have a guiding purpose that shapes their choices – much like well-written characters in a story or effective agents in planning systems.

In Stanislavski's system, a super-objective is the "through-line" of a character, and all inner experiences (thoughts, emotions) and outer actions ideally work in tandem towards that superobjective [48] . We adapt this idea computationally: each agent will be assigned or will formulate a super-objective (e.g. *"achieve justice for my family"*, or *"win the love of X"*, or a more internal one like *"prove my worth"*). The inner engines (like the dramatic drives from Phase 6, basic needs from Phase 5, etc.) should be orchestrated under this umbrella. If there's conflict between drives and the super-objective, that itself becomes a point of dramatic tension for the agent (just like a character might want something that conflicts with their moral objective, causing inner conflict).

Implementation Guidance: We will extend the agent's goal representation to include a hierarchy: at the top, the Super-Objective (SO); below it, medium-term objectives or tasks; and below those, immediate actions. A planning or reasoning process should link these levels (for example, if SO = "become a leader of the community", a sub-goal might be "win the town election", sub-sub goals "befriend townsfolk", "make a plan", etc., down to concrete actions "give a speech tonight"). Our architecture can implement this via a goal stack or tree, similar to HTN planning or the BDI (Belief-Desire-Intention) model in agent systems, where desires correspond to high-level goals and intentions to current plans.

We will train or program the agents to continuously evaluate: "Is what I'm doing right now helping achieve my super-objective?" This reflective check (which could happen during the *Reflect* phase of their cognitive cycle, cf. the generative agent memory flow [33]) will help them prioritize and potentially drop actions that don't serve their through-line. It also might generate *inner monologue* (the agent thinking to itself, aligning with Stanislavski's inner experiencing) which helps debug and also enriches narrative if we expose it as thought speech.

A challenge is initial assignment of super-objectives. If we have a specific story in mind, we can author them. Or we can have agents *learn or evolve* them (for example, through simulation an agent develops a strong desire based on experiences, like after being wronged, they form a revenge SO). We might implement a mechanism for SO formation: e.g. if an agent's emotional analysis (Phase 4) finds a sustained significant emotion linked to an unresolved situation, it can crystallize a new super-goal to address it.

We will also integrate the concept of **through-line of actions**: making sure that from Phase 7, the inner-outer alignment, now those outer actions indeed line up in pursuit of the SO [48] . So if an agent's super-objective is known, an observer should be able to deduce it by watching their consistent pattern of actions (like "this person always acts to protect the weak, I guess their ultimate goal is justice").

Technically, we might use a constraint solver or simply heuristic: every time a sub-action is chosen, a component checks its relevance to SO and can alter priority. If the agent strays (maybe due to random drive or reaction), the architecture might introduce an inner dialogue like "Focus, remember the mission!" to push back on track, unless we intentionally allow deviation for realism (people do digress). The degree of alignment could even be a personality trait: some characters are single-minded (everything for the SO), others are more wavering or distracted.

Key Research Questions: *Does having a super-objective make agents more effective or story-coherent?* We will compare simulations where agents have an SO versus not. We expect more coherent long-term behavior with SO. But does it reduce spontaneity? Maybe sometimes humans do unrelated things; our characters might become too goal-driven and predictable. We may need to calibrate the influence of SO, perhaps only strongly activating it when context calls (like in key plot moments). *Dynamic super-objectives:* Can an agent's SO change? In life and stories, sometimes yes (a dramatic change of heart). We should allow the architecture to update an SO if certain critical events happen (this could be triggered by Phase 6's drama manager if it decides a twist like "the villain redeems himself and now seeks forgiveness instead of power"). The mechanism to change an SO – essentially a major reorientation of the agent – would need to propagate through all sub-goals and maybe memory re-interpretation. Research here is in how to gracefully handle such major shifts (maybe akin to cognitive dissonance resolution in psychology). *Conflict of super-objectives among agents:* Likely, and dramatically interesting. We should analyze scenarios where two agents' SOs are in direct conflict. Does our architecture handle prolonged conflict? For instance, do they continually thwart each other's sub-goals? We might see emergent tit-for-tat or the need for negotiation. Possibly we can incorporate models of negotiation or compromise if we want realism (not all conflicts end in fight; sometimes they resolve diplomatically, which could happen if agents re-evaluate feasibility of SO and adjust).

We also tie this to theme: in drama theory, the super-objective can relate to the play's theme (e.g. "love conquers all"). Are our agent's SOs aligned with an overall story theme? Possibly something a higher-level narrative engine can enforce by setting complementary or opposing SOs (like protagonist vs antagonist SO). We might research how to encode theme such that multiple agents' SOs reflect different facets of that theme.

Related Work: Stanislavski's **supertask** concept is our inspiration [49] [48]. In AI planning, the idea of a top-level goal is standard (e.g. in STRIPS planning you have a goal state). BDI agent models explicitly distinguish desires (analogous to SO) and intentions (current objectives) and provide a framework for continuous goal deliberation – we will lean on BDI theory to implement this (perhaps using Jason or PyBDI libraries if any).

In narrative planning, systems like **IPOCL (Intent-based Partial Order Causal Link planner)** ensure characters' actions are intentional with respect to goals, adding "intent" constraints to plans to increase believability. This is directly relevant; IPOCL introduced the idea of plans where every action should be explainable by a character's intent. We are doing something similar in real-time. We might use a planner to check or fill gaps: for instance, if our agent simulation produced an action that doesn't line up with any active goal, that's an anomaly – maybe an "unmotivated action". We would either eliminate it or establish a new motivation retroactively to justify it (like improvisers saying "I did that because …"). This concept of **motivational continuity** is well explored in story generation research.

OpenCog's design for motivational system (ECAN, the Economic Attention Network, which allocates cognitive effort to goals with certain values) could inform how we allocate attention to super-objectives vs minor objectives [50]. We might implement something similar: the SO has a certain "budget" of attention always, so it doesn't get entirely forgotten.

Potential Collaborators, Tools & Frameworks: For the cognitive aspect, working with researchers in **cognitive architectures or agent planning** (like those who built PRS or Soar's goal systems) would help refine our design. Theatre practitioners who use Stanislavski might give qualitative feedback on whether the agent's actions indeed read as driven by a clear through-line. Tools-wise, if we use any automated planning to assist goal management, PDDL solvers or Hierarchical planners could be integrated (for instance, using a PDDL domain where the top-level task is the SO and methods break it down; the agent can consult this as needed to remind of possible sub-goals).

We'll likely also develop debugging visualizations: e.g. each agent's current tree of objectives displayed, with the super-objective highlighted, to monitor alignment. Using graph visualization libraries or even text logs is fine, but given complexity, maybe something like an interactive dashboard (web-based) to click an agent and see their inner goal state in real-time would aid development.

By the end of Phase 8, each agent in our system should feel like a protagonist of their own story, with a clear motive force that audiences (or the user) could summarize in one sentence ("This character wants X above all"). This will make the emergent narratives more meaningful and give us leverage to direct stories by tweaking those central desires. It also will interplay with Phase 9, where we impose constraints on how many internal "drives" can be active at once, refining how complex their motivational state can be at any time.

Phase 9: Constraining Inner Participations – Limiting Active Spectrums to Define Externalized Drives

Objectives & Theory: Phase 9 imposes constraints on the internal complexity of drives active within an agent at any given moment, following the insight from Laban and Malmgren that not all Effort or psychological factors operate simultaneously with equal strength. Specifically, Laban's theory of *Effort Drives* notes that typically only **3 out of the 4 Effort factors** are fully expressed at one time, with one factor latent or suppressed [51]. This concept of "only 3/4 spectrums active" can be seen as a simplification that yields a clearer, more defined external drive or action. In Malmgren's terms, an **Inner Participation** might refer to which of the psychological function spectrums are engaged in the character's inner state versus which are idle, leading to a more distinct character motivation and expression.

The objective of Phase 9 is to enforce such constraints in the AI's inner state to avoid muddled or contradictory drives and to allow the emergence of **distinct drive states** that manifest externally in recognizable ways. In other words, by limiting active drives, we get a sharper, more archetypal behavior at any one time (e.g. an agent could be in a state of Action Drive like "Fight mode" vs. a Passion Drive "Emotional mode", etc., but not everything at once). This not only mirrors how human action tends to organize (per Laban, full four-factor effort is rare and momentary [52]), but it also simplifies decision-making for the AI and gives clarity to observers.

Implementation Guidance: We will formalize a set of allowed *drive configurations* for the agents. Based on Laban's Effort Drives, the four combinations of three factors are: **Action Drive** (Space, Weight, Time active; Flow suppressed), **Passion Drive** (Weight, Time, Flow active; Space suppressed), **Vision Drive** (Space, Time, Flow active; Weight suppressed), and **Spell Drive** (Space, Weight, Flow active; Time suppressed) [51]. These could correspond to distinct modes of operation for the agent. For instance, an Action Drive mode corresponds to very goal-directed, functional action (Flow missing implies a lack of ongoing self-monitoring, just acting decisively) – perhaps suited for fight-or-flight scenarios or executing plans. Passion Drive (missing Space) implies highly emotional, not concerned with spatial/contextual nuance, often associated with indulgent expression of feeling. Spell Drive (missing Time) can

be like a sustained, almost trance-like or controlled state (perhaps contemplation or awe), and Vision Drive (missing Weight) could imply ideas and inspiration without immediate "weighty" effect (imaginative or planning mode).

We will map the agent's inner state (from Phases 6-8: emotions, goals, cognitive context) to one of these drive modes. This might be a rule-based classification or a small neural network that classifies the current state into one of (Action, Passion, Vision, Spell) or possibly a neutral state. Once classified, the agent's decision-making and movement expression will be constrained accordingly: e.g., in Action Drive mode, allow only certain types of behaviors (more direct and urgent actions, fewer reflective or flowy behaviors). We essentially reduce degrees of freedom in behavior to accentuate the dominant drive.

Additionally, at a given time, the agent's *psychological function usage* (Jungian) might be limited: Malmgren indicated people favor one function consciously and maybe two auxiliary, leaving one inferior (unused) [53] [54]. We could simulate that by, for example, if an agent is in a very Thinking-driven state, it suppresses Feeling function temporarily, etc. This ensures consistency (the agent won't simultaneously make a cold logical decision and an empathetic emotional decision in the same breath; they might oscillate, but at a given moment one dominates).

From a coding perspective, we will implement a controller that monitors the "inner participation" of various spectra (like logical vs emotional, or the 4 Effort factors as analogs). It then zeroes out or deactivates one dimension. For example, if using a numeric representation, we could set the lowest-priority factor to zero for the duration of that drive mode. The priority could depend on context: in a heated argument (Passion), spatial awareness might drop (Space factor suppressed, meaning they might not care about context or precision). In planning mode (Vision), maybe the Weight factor (intensity/pressure) is low, meaning they're exploring ideas lightly without committing force.

We must also handle transitions between modes. They might be triggered by internal changes (achieving a sub-goal might shift from Action to Spell in a moment of pause, for instance). Possibly tie this to emotional peaks: e.g., extreme anger might invoke an Action drive (remove Flow – they act without self-regulation), whereas sorrow might invoke Passion drive (remove Space – they become inward-focused and oblivious to surroundings). We'll design heuristics or learning rules for these triggers.

Key Research Questions: *Clarity vs. complexity:* Does constraining drives improve the clarity of the agent's behavior to observers? We will evaluate if, say, an agent's action is easier to interpret when we enforce one factor off. *Flexibility:* Are there situations where full-spectrum (all 4 factors) is actually needed for nuance? Laban said full effort is rare and usually momentary [52], but can occur in intense moments. Maybe our agents should have moments of full 4-factor expression (perhaps the climax of a story, showing complete commitment). We need criteria for those exceptions. *Mode frequency and balance:* We can analyze logs to see if agents favor one drive mode too much. Ideally, it varies appropriately with context. If one agent is always in Action Drive, that might be its character trait (could be fine if intentional). But if none ever enters Spell (introspective) mode, maybe our triggers are too biased towards action. Tuning will be required. *Integration with super-objectives:* How does the super-objective influence drive? Perhaps characters have a dominant drive tendency: a warrior character might tend to Action Drive, an artist to Vision Drive, etc. We could incorporate that as a personality parameter. Then research question: does that alignment of character trait to drive make them more effective at their SO or just more predictable? Could be either; we might see that sometimes a character needs to switch drives to achieve their SO (and that can be a moment of growth, e.g. a thinking character finally follows their feelings in the climax). We should allow or script such arcs. *Computational overhead:* Monitoring and toggling these states is not heavy, but we should ensure it doesn't destabilize learning (if any RL is happening concurrently). Perhaps treat drive mode as an observed variable in RL –

it could reduce state space by focusing policies on a subset of behaviors at a time, which might actually make learning easier (like context-dependent policy).

Related Work: Laban's **Effort States and Drives** are the core reference [55] [56]. Effort Drives have been used to classify movement and psychology; we have the exact definitions from LMA. In AI, there isn't direct analog, but the idea of **behavior modes or finite-state machines** in character AI is related (game characters often have states like "Aggressive", "Defensive", etc., which is a coarse analog to our drives). Our approach is more fluid but conceptually similar.

Malmgren's notion of **"shadow movements"** and inner conflict might be relevant: the search result earlier mentioned "Shadow Moves" and "Subconscious motifs" [57] – possibly referring to the suppressed aspects that still cast a 'shadow'. We might incorporate that by noting that the suppressed factor can still influence subtly (like Flow is "off" in Action Drive, but an involuntary tremble or short burst might show it trying to break through – e.g. a character in fight mode might suddenly freeze a moment, hinting at a suppressed hesitation). This is advanced nuance: if time permits, we can simulate such leaks for realism.

In robotics and HCI, one might not find an exact 3-of-4 concept, but there is the idea of *principal component analysis* of movement and focusing on dominant components. We could draw analogies that our constraint is like forcing the system to use a principal subspace of expression which is easier to interpret.

Potential Collaborators, Tools & Frameworks: Collaborating with movement analysts or theatre practitioners to validate the drives manifested would be useful (like Phase 7's collaborators, maybe same folks). On the technical side, implementing these constraints is mostly internal logic; however, if using ML for agent control, we might incorporate it as part of the network architecture – e.g. a policy network whose output is factorized by Effort factors and we gate one off. Or use constrained optimization if using planning (like add a constraint "Flow = 0 in this plan").

We might also consider using **rule-based systems** (like Jess or Drools) for high-level drive switching rules, since those can be clear to author (e.g. IF agent angry AND urgency high THEN set mode=ActionDrive). That could overlay on the learned behavior.

Additionally, we can instrument the simulation to highlight which drive an agent is in (for debug or even in the narrative, describe the demeanor). Tools: we can color-code agents or their nameplates by drive (just for developer observation). Or log timeline of drive changes.

By implementing Phase 9, our agents will not only have rich internal lives but *disciplined* ones – at any moment dominated by a certain drive, making them appear more stylistically consistent and purposeful. This also paves the way for Phase 10, where we will use these balanced cognitive-expressive states to extrapolate concrete movements, essentially turning these Effort and drive configurations into actual motion trajectories dynamically.

Phase 10: Dynamic Movement Extrapolation from Cognitive and Expressive Balance

Objectives & Theory: Phase 10 takes the qualitative movement integration from Phases 7 and 9 and turns it into a concrete system for **dynamic movement generation**. The objective is to algorithmically generate the actual motion trajectories (positions, angles, speeds) of the agent's body in real time,

driven by the current cognitive-emotional state (balance of factors) and expressive mode (Effort drive) of the agent. Essentially, we want an AI choreographer or motion controller that reads the agent's "mind state" and outputs continuous movement that reflects that state, without needing pre-scripted animations for every possible combination. This is crucial for flexibility – our agents should be able to perform any needed physical action (walking, gesturing, fighting, etc.) in a style that reflects their inner state, even for novel sequences not explicitly animated by an artist.

The underlying theory comes from both biomechanics and AI: in biomechanics, human movement can be parameterized by qualities (Effort, Shape, etc.), so theoretically one can interpolate between exemplar movements to achieve an arbitrary point in Effort space [45]. In AI, this connects with **motion synthesis** techniques. We'll draw on approaches like **hidden Markov models (HMMs)** or **neural generative models** for motion that take high-level style parameters as input. Research such as Samadani et al. demonstrated generating movements with target emotions by overlaying affective content on a base motion via searching a motion database in Effort-Shape space [58]. Also, Kim et al. (2025) showed how to use diffusion models to adjust motions to have desired Laban Effort and Shape components [47] [59].

Implementation Guidance: We will likely pursue a two-pronged approach: 1. **Motion Library & Interpolation:** Build a library of base motions (like walk, run, reach, wave, etc.) performed in neutral style. Then apply algorithms to modulate these motions according to Effort parameters. For instance, to modulate a "walk" motion to be "light" vs "strong", we can adjust force/acceleration of footfalls (increase vertical bounce for light, increase downward force for strong). For "direct" vs "indirect", adjust the path to be straight vs curved. For "sudden" vs "sustained", adjust timing/duration (maybe speed up for sudden starts/stops, or use a ease-in-out profile for sustained). For "free" vs "bound" (Flow), adjust how fluid vs stiff the joints are – perhaps add noise for free (looseness) and dampen for bound (controlled, rigid stops). We can derive these rules from Laban movement analysis literature and some trial and error with motion capture data.

1. **Learning-Based Synthesis:** Train a model (like a neural network) that maps from a state vector (could include Effort factors, emotional state, and intended action type) to a sequence of motion frames. This could be a neural network (e.g. a conditional VAE or GAN for motions, or a transformer that generates motion frames given a "style token"). The training data can be a set of example motions labeled with Effort qualities (some datasets exist, or we label them manually via Laban codified efforts [60] [61]). For example, feed in motions of "happy walking" labeled as Indirect, Light, Sustained, and let the model learn to correlate the label with motion style. Then at runtime, given a new combination, the model can generate a new motion. The diffusion approach by Kim et al. suggests we can even fine-tune a pre-trained text-to-motion model by adding a loss for Effort qualities and then at runtime specify desired Effort and have it adjust any baseline motion accordingly [59] [62].

We will likely do a simplified version initially: parametric interpolation of motion segments might suffice for basic tasks like walking or gesturing. For complex actions, a learning approach or a more robust procedural system might be needed.

Crucially, this is dynamic: as the agent's state changes continuously, the movement generator should also adapt continuously (not just discrete switches). We might implement a **feedback loop**: every simulation tick, take the current Effort/drive state from Phase 9 (which might blend slightly, though it's mostly one drive), and update the motion target accordingly. If an agent is mid-action and suddenly gets surprised (state change), their motion path could be perturbed (like a flinch or re-routing).

Key Research Questions: *Continuity and naturalness:* Will dynamically generated movements be free of jitter and artifacts? This is a challenge – blending from one style to another seamlessly is tough. We might use techniques from robotics like trajectory optimization to ensure physical plausibility (no sudden infinite accelerations). *Evaluation of expression accuracy:* Similar to Phase 7 but now with full motion, we'll test if observers can correctly identify the emotion/drive behind a movement. If we've done well, an observer should say "that walk looks angry" when it's supposed to. If not, we refine either the generation or the mapping. *Generality:* Can our system handle various actions or is it limited to locomotion? We should test multiple movement domains: locomotion, gestural communication (like pointing, shrugging), and potentially more complex ones like dance or combat moves if relevant to our story domain. The theory suggests Laban Effort applies across all movement, but implementation difficulty varies. We might focus on a subset (walk, run, idle stance, a few gestures) that cover most story scenes. *Integration with physics:* If environment has physics (like pushing objects), how do Effort modifications affect outcomes? E.g., bound vs free movement might change whether an object is carried carefully vs. sloshed around. We can incorporate physics engines constraints (like friction, mass – Weight Effort might correlate to how much force is applied to objects, etc.). Ensuring no conflict between physics (which demands certain forces to achieve tasks) and stylistic overlay (which might reduce/increase force) is important. Possibly we always meet the minimum physics requirement, then style beyond that.

Related Work: There's a rich field of **character animation** in computer graphics: from classic interpolation with blend spaces to modern deep learning motions. The idea of a "motion style parameter" is well studied (e.g., motion style transfer networks). For instance, researchers have done style transfer where you take a motion and output the same motion in a different style (walking neutral -> walking tired or jaunty). Our Effort factors basically define a style space, so we can adopt style transfer techniques. One such technique is via neural networks that disentangle style and content, allowing you to mix content (the action being done) with a style vector to generate the motion. We can treat Effort drive as that style vector.

We also look at **procedural animation**: things like the OpenAI gym humanoid or bullet physics ragdolls with RL policies – those produce motion but usually not with a style (they optimize for efficiency). However, recently people have done RL for locomotion with style rewards (like a "bouncy" walk vs "stooped" walk). If needed, we could attempt an RL approach: train a physically simulated agent to move with certain Effort by rewarding it for matching certain movement feature patterns (e.g. a reward for high variance in shoulders to encourage free flow, etc.). That's complex and time-consuming, though.

In robotics, the **NAO robot emotional gait** experiments and stuff by Hiroshi Ishiguro's lab on android gestures might be relevant. If eventual embodiment in a physical robot, we'd definitely adapt this to real-world constraints (like a Pepper robot can only do so much, so Effort expression might be more symbolic in gesture).

Potential Tools & Collaborators: On the tool side: if in Unity, their animation system can blend animations with parameters, we could use their Animation Controller with blend trees for Effort factors. Alternatively, use specialized software like **Autodesk Maya** or **Blender** to design a parametric rig. There is also research software like **EMOTE** (Expressive Motion Engine) from a project by Chi et al. 2000, which explicitly modulated robot motion by Effort parameters. That could be a gem to look at [63] (the context mentions Chi et al. 2000 with implementing Time Effort by non-uniform scaling of animation time, etc., which is exactly what we need to do).

Collaborators likely include **computer graphics and animation experts** (who can help with making the motion output look natural and maybe supply motion capture data), and **dance/movement**

technologists (like those who work on motion capture analysis of Laban – there are academic groups for this, e.g., some at University of Maryland or University of Waterloo in Canada). If budget permits, capturing our own motion examples under direction of a movement coach (like telling an actor to do an action in each Effort mode) could greatly supply training data.

We'll also rely on any existing **datasets**: I recall a dataset of acted movements with emotional intent (possibly the CMU motion capture database has some labeled sequences). If none for Effort specifically, we can label a portion of a known dataset with Effort ourselves (through expert annotation) to train our model.

By completing Phase 10, the loop from mind to motion is fully realized in a generative way: our AI's cognitive architecture state flows through to continuous, styled movement. The agents will physically *embody* their current feelings and drives in a fluid, non-repetitive manner. Finally, we move to Phase 11, which will formalize and incorporate the higher-level conceptual pieces from Malmgren that we have touched upon (Working Actions, Inner Quests, Effort Cube) to ensure the system as a whole is aware of and utilizes these structured concepts.

Phase 11: Incorporation of Working Actions, Inner Quests, and Effort Cubes

Objectives & Theory: Phase 11 is the capstone, integrating specific advanced constructs from Malmgren/Laban and ensuring the architecture explicitly represents and can reason about them. The objectives are: (1) to incorporate **Working Actions** – the eight fundamental action Efforts (Float, Punch/Thrust, Glide, Slash, Dab, Wring, Flick, Press) – as discrete expressive units the AI can use or even name internally, (2) to integrate the concept of **Inner Quests** – which we interpret as deep internal motivations or thematic yearnings of a character (distinct from immediate objectives, more like existential goals or subconscious desires), and (3) to utilize the **Effort Cube** representation – a 3D model where Effort elements are axes and the eight effort verbs lie on the corners [60] – as a framework for organizing and navigating the agent's expressive movement repertoire.

In Malmgren's system, "Working Actions" are those composite actions that are psychologically motivated and expressed physically, essentially the external manifestation of inner motifs [57]. They correlate with subconscious motifs or archetypal actions (like "wringing hands in despair" aligns with the Wring effort). By incorporating these, our AI can use them as building blocks for movement and perhaps communicate complex states through a single concise action (e.g., an agent might internally decide "I will *Punch* the table in anger", linking both a physical move and an emotional statement).

"Inner Quests" likely refer to fundamental personal drives that are not simply situational goals (e.g. "seeking validation", "pursuit of truth", "desire for love") – similar to Jungian individuation themes or simply each character's core need. While we have super-objectives which are more concrete, an inner quest might be more spiritual or psychological, possibly what the character needs vs what they want in story terms. Aligning the AI's behavior with an inner quest provides depth – even if a super-objective fails, the inner quest might find another expression (like if "seeking respect" is an inner quest, whether the goal was career or heroism, the underlying drive is respect).

The Effort Cube is a convenient **map** of movement qualities [60]. By incorporating it, we give the AI a structured way to reason about movement transitions (moving along an axis on the cube changes one quality at a time, which could correspond to nuanced changes in attitude). For example, if an agent is currently in "Press" (Strong, Direct, Sustained) and we know we want them to lighten up slightly, we can move in the cube to "Punch" (Strong, Direct, Sudden) or to "Wring" (Strong, Sustained, Indirect)

depending on whether it's time or space we alter. This can guide the AI's modulation of movement and emotion logically.

Implementation Guidance: We will add a layer to the cognitive architecture that explicitly stores these concepts: - A library of **Working Actions** as objects. Each might have attributes: the Effort combination, typical meaning, maybe example scenarios. The AI can choose a Working Action as a communication or expressive act. For instance, if an agent wants to show determination, it might internally select "Press" action (perhaps pressing a fist into palm, metaphorically). We can create a mapping from some emotional/intention states to suggested Working Actions (like a lexicon of body language). - **Inner Quests** could be a new field in the agent's character data. Possibly aligned with the "theme" of that character's arc. We ensure it influences decision-making subtly. For example, if inner quest is "freedom", the agent might resist situations that trap it, even if logically sound – that bias would come from this deep quest. We implement that as a bias or an emotional weight on relevant decisions (like a heuristic that picks plans favoring the quest). - The **Effort Cube** can be represented mathematically (3 axes binary +/-1 for each Effort element, which yields 8 corners). We can incorporate algorithms to move in this cube. For example, a function to get neighboring Effort (change one factor by flipping it). We could even animate the idea: the agent could transition through Effort states by "climbing" from one cube corner to another along edges (like moving through Effort States which involve two factors, etc.). This might give smoother transitions in movement expression (only change one quality at a time perhaps). In AI decision-making, if an agent is trying to gradually influence another's mood, it might change its movement one factor at a time to not startle – e.g., gradually become more Direct while keeping other qualities same, then increase Weight, etc.

We'll also integrate these with learning or planning: for instance, a **behavior planner** could use Working Actions as actions in its domain. Instead of just "move to X" it could have an action "Glide to X" vs "Slash towards X" with different outcomes (slashing might intimidate, gliding might comfort, for example). The planner can then choose the one that suits the current strategy (perhaps determined by inner quest or dramatic need).

From a software perspective, these might just be data structures and some rules. E.g., a WorkingAction enum with values FLOAT, PUNCH, etc., and each agent has a current WorkingAction it's embodying, possibly updated per beat of the story. The inner quest could be a simple string or a complex object with conditions (like "never kill anyone" or "find belonging").

Key Research Questions: *Awareness and explainability:* Does explicitly modeling these improve the AI's ability to explain or justify its behavior? For example, could the AI articulate "I did that because I am seeking redemption (inner quest)" or "I reacted with a Flick because I felt playful surprise"? Having symbolic labels like Working Action and Inner Quest might allow more introspective or explanation capabilities, which are valuable. We should test if the AI can use these concepts in its internal reasoning trace or even communicate them if appropriate. *Balance between emergent and scripted:* Are we over-constraining by adding these formal constructs? We need to ensure that characters still feel organic and not like they are just following a script of archetypes. It might be good to allow occasional deviations or mixtures (maybe two working actions blended, though that's outside Laban's standard eight, but humans can mix e.g. a glide with a dab). Possibly the architecture can choose a primary and secondary effort action to allow more variety (like a slight Flick on a Float base, making it unique). *Data for inner quests:* How to assign or learn inner quests? Possibly tie to backstory: an orphan character's inner quest is family/belonging, a disgraced knight's is honor, etc. We could systematize this by reading common character archetypes (maybe use Propper or Campbell's archetypes to suggest inner quests).

Another question is *monitoring and adjustment:* do we need a "quest manager" that measures progress toward the inner quest over time? If an agent consistently fails its inner quest, does it get frustrated

(leading to perhaps a change of tactics or even a breakdown moment)? That could be dramatic to simulate. We may implement an “inner quest fulfillment level” metric and trigger events if it’s too low for too long (like the character might go into a despair mode or drastically change approach, which adds drama).

Related Work: Malmgren’s "Inner Attitudes" are essentially inner quests related to psychological needs [64]. In method acting and also in Maslow’s psychology, characters/people have needs that drive them – our inner quests align with that. In narrative theory, this is analogous to the difference between a character’s *want* and *need*: e.g., Shrek wants to be left alone (explicit goal) but needs friendship (inner quest). We could use that framework to give our agents a layer of need vs want, and see if story naturally brings out conflict between them (which often happens in good narratives).

The eight Effort verbs on the Effort Cube [60] [65] are well documented and have been used in computational recognition tasks (Remi Ronfard’s work on recognizing Effort from motion via the cube we saw shows how these verbs can be identified and thus we can generate them in reverse). It provides confidence that we can formalize them for generation because they’ve been formalized for recognition.

Potential Collaborators, Tools & Frameworks: We may bring in a dramaturg or narrative designer at this final phase to validate that these inner quests and working actions indeed create more meaningful and thematically rich stories. They could look at an agent and say “Yes, this character is clearly motivated by X and expresses it through Y” – essentially validating that our implementation achieves character coherence akin to what they’d aim for in playwriting or directing.

Technically, not many off-the-shelf tools exist for this because it’s quite tailored, but we might use knowledge representation tools to store inner quests (maybe integrate them as part of the knowledge graph: a special node representing the quest and linking to various memories that support it). That way the agent can reason about it (like search its memory for instances related to its quest, which could influence decisions).

Also, if using any symbolic AI, we could encode rules like “If inner quest is X and situation Y arises, that creates high stress” to simulate inner conflict. Tools like prolog or even just Python rule sets could do that.

Finally, evaluation: our best evaluator here might be narrative coherence or even literary analysis. Perhaps run a full simulation/story and have an expert critique the character arcs: Did characters pursue their inner quests? Did their outward actions (working actions) align and underscore those quests? Ideally the answer is yes, providing a satisfying story where movement, dialogue, and plot all serve the character’s journey.

By integrating the structured elements of working actions, inner quests, and using the effort cube, we complete the architecture’s design – it now possesses a **unified cognitive, emotional, and physical model** of acting. The creator/researcher (our user) can now experiment with this comprehensive system: assigning a character an inner quest, seeing it play out through super-objectives and actions, watching the movements express the inner life, and observing how the dramatic arcs unfold, all grounded in both psychological theory and movement theory.

Throughout these phases, we cited parallels and used frameworks like OpenCog and ACT-R for cognitive structure, Laban/Malmgren for movement and psychology integration, narrative theory for story drives, and modern AI techniques for implementation. Each phase builds upon the previous, and cross-references among them ensure consistency (for instance, Phase 7’s movement expression draws

on Phase 3's multimodal learning and Phase 9's drive constraints). The end result is an AI architecture that doesn't just think or move in isolation, but **lives through a role** – integrating thinking, feeling, moving, and interacting in a coherent, lifelike manner.

Sources:

- Goertzel, B. *et al.* (2023). *OpenCog Hyperon: A Framework for AGI*. (OpenCog Hyperon integrates probabilistic logic, neural reasoning, and multi-agent learning in a unified cognitive architecture [3] [2].)
- Anderson, J.R. *et al. – ACT-R Cognitive Architecture* (ACT-R represents declarative knowledge as chunks – vector representations of properties – and uses modular perceptual-motor components [1].)
- Long, B. *et al.* (2025). *M3-Agent: Multimodal Cognitive Framework* (Demonstrates an agent integrating visual/auditory inputs into episodic memory for cross-modal reasoning [21] [66].)
- Anderson, T. (2015). *From Episodic Memory to Narrative in a Cognitive Architecture* (Argues episodic memory is fundamental to narrative intelligence and planning, bridging raw experience into narrative structures [67] [20].)
- Park, J.S. *et al.* (2023). *Generative Agents: Interactive Simulacra of Human Behavior* (Showed believable human-like agent behavior in a sandbox environment using an LLM for memory and planning – agents acted according to biographies and remembered interactions [31] [32].)
- Liu, H. *et al.* (2023). *METATRON Framework for Story Generation* (Combines classical dramatic situation taxonomy with neural generation; injecting structures like Polti's 36 situations guides narrative coherence and creativity [30] [34].)
- Mirodan, V. (2015). *Acting the Metaphor – Laban-Malmgren System* (Explains the synthesis of Laban's movement, Jungian typology, and Stanislavski's action in Yat Malmgren's technique – linking physical motion factors with psychological functions [12].)
- **Stanislavski's System** – via Benedetti, J. (1982). (Defines super-objective ("supertask") as the through-line uniting inner (experiencing) and outer (embodiment) aspects of a role in pursuit of the character's overarching goal [48].)
- Laban, R. – *Laban Movement Analysis* (Effort system defines four factors Space, Weight, Time, Flow each with two polar elements; combinations of three factors (with one suppressed) form Drives [40] [55]. The eight Effort Actions – Float, Punch, Glide, Slash, Dab, Wring, Flick, Press – are used in actor training to quickly shift physical emotional expression [41].)
- Ronfard, R. *et al.* (2020). *Recognition of Laban Effort Qualities* (Illustrates the Effort Cube with Effort verbs at corners; shows how movement qualities can be represented in a cube model and recognized by AI [60] [65].)
- Samadani, A. *et al.* (2020). *Affective Movement Generation using Laban Effort and HMMs* (Demonstrates automatic generation of movements conveying target emotions by modulating motion paths in LMA Effort/Shape space [45] [58].)
- Kim, H. *et al.* (2025). *LaMoGen: Laban Movement-Guided Diffusion for Text-to-Motion* (Achieves expressive control in motion generation by integrating Laban Effort/Shape quantification into a diffusion model, allowing motions to be generated with desired Effort qualities [47] [59].)

---

1 ACT-R - Wikipedia
https://en.wikipedia.org/wiki/ACT-R

2 3 5 50 OpenCog Hyperon - ASI | Artificial Superintelligence Alliance
https://superintelligence.io/portfolio/opencog-hyperon/

4 Vector Symbolic Architectures and Hyper-dimensional Computing ...
https://dais-ita.github.io/1a11/

6 A Systematic Literature Review of Reinforcement Learning-based ...
https://www.sciencedirect.com/science/article/pii/S0957417423023825

7 8 13 Designing Personality: Cognitive Architectures and Beyond
https://cdn.aaai.org/Symposia/Spring/2004/SS-04-02/SS04-02-017.pdf

9 Full article: Thinking Fast and Thinking Slow: Digital Devices' Effects ...
https://www.tandfonline.com/doi/full/10.1080/07421222.2023.2196769

10 Modularity of Mind - Stanford Encyclopedia of Philosophy
https://plato.stanford.edu/entries/modularity-mind/

11 [PDF] A Systems-level Architecture for Cognition, Emotion, and Learning
https://ccrg.cs.memphis.edu/assets/papers/2013/franklin-ieee-tamd11.pdf

12 53 54 64 (PDF) Acting the metaphor: The Laban-Malmgren system of movement psychology and character analysis
https://www.researchgate.net/publication/277594855_Acting_the_metaphor_The_Laban-Malmgren_system_of_movement_psychology_and_character_analysis

14 [PDF] Minding Language Models' (Lack of) Theory of Mind - ACL Anthology
https://aclanthology.org/2023.acl-long.780.pdf

15 16 BChandrasekaran.PDF
http://act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/workshops/2002/talks/BChandrasekaran.pdf

17 18 21 22 66 M3-Agent: Multimodal Cognitive Framework
https://www.emergentmind.com/topics/m3-agent

19 20 67 drops.dagstuhl.de
https://drops.dagstuhl.de/storage/01oasics/oasics-vol045_cmn2015/OASIcs.CMN.2015.2/OASIcs.CMN.2015.2.pdf

23 3.6 Multimodal Perception – Cognitive Psychology
https://nmoer.pressbooks.pub/cognitivepsychology/chapter/multimodal-perception/

24 Cross-Modal Interaction Between Auditory and Visual Input Impacts ...
https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2021.661477/full

25 43 Methods of Generating Emotional Movements and Methods of Transmitting Behavioral Intentions: A Perspective on Human-Coexistence Robots - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC9227009/

26 27 28 29 30 34 35 36 37 38 39 Integrating Cognitive, Symbolic, and Neural Approaches to Story Generation: A Review on the METATRON Framework
https://www.mdpi.com/2227-7390/13/23/3885

31 32 33 Computational Agents Exhibit Believable Humanlike Behavior | Stanford HAI
https://hai.stanford.edu/news/computational-agents-exhibit-believable-humanlike-behavior

40 41 42 51 52 55 56 Laban movement analysis - Wikipedia
https://en.wikipedia.org/wiki/Laban_movement_analysis

44 MoRTELaban: a Neurosymbolic Framework for Motion ...
https://dl.acm.org/doi/full/10.1145/3708319.3734180

45 46 58 [2006.06071] Affective Movement Generation using Laban Effort and Shape and Hidden Markov Models
https://arxiv.org/abs/2006.06071

47 59 62 LaMoGen: Laban Movement-Guided Diffusion for Text-to-Motion Generation | Cool Papers - Immersive Paper Discovery
https://papers.cool/arxiv/2509.24469

48 49 Stanislavski's system - Wikipedia
https://en.wikipedia.org/wiki/Stanislavski%27s_system

57 (PDF) The Way of Transformation (the Laban-Malmgren System of ...
https://www.academia.edu/2108410/The_Way_of_Transformation_the_Laban_Malmgren_System_of_Dramatic_Character_Analysis_vol_I

60 61 63 65 Laban's Effort cube: the eight Effort Verbs are represented at the... | Download Scientific Diagram
https://www.researchgate.net/figure/Labans-Effort-cube-the-eight-Effort-Verbs-are-represented-at-the-corners-of-the-cube_fig1_342703800