ChatGPT

Advancing the PPP and HAOS Paradigm: From Visionary Concept to Prototype

Relevant Theoretical Foundations and Research Areas

High-Dimensional Geometric Cognition: A core idea behind Polytopal Projection Processing (PPP) is likely the representation of knowledge in high-dimensional geometric structures (polytopes) and using projections to recognize patterns. This aligns with the theory of **conceptual spaces**, where concepts are represented as regions in a geometric space [1]. In such frameworks, *instances* of knowledge are points in a similarity space and *concepts* are convex regions, offering a way to encode meaning geometrically rather than with strings or syntax [1]. Similarly, the field of **hyperdimensional computing** (or Vector Symbolic Architectures) treats information as large, high-dimensional vectors. In these models, all entities (features, roles, etc.) are encoded as high-dimensional vectors, and operations like binding and superposition are used to combine information [2]. Such approaches address the limitations of purely symbolic AI by exploiting geometry for cognition, providing PPP a rich mathematical foundation for *a-syntactic* (non-linguistic) pattern recognition. Essentially, rather than manipulating symbols with formal grammar, PPP's high-dimensional projections could recognize patterns by geometric *shape* in state-space – akin to how a visual system grasps an image holistically.

A-Syntactic Pattern Recognition (System-1 Style): PPP's emphasis on pattern recognition without explicit syntax relates to the intuitive, fast thinking described in dual-process cognitive theories. In cognitive science, **Dual Process Theory** distinguishes a quick, pattern-based "System 1" from a slower, logical "System 2." AI researchers draw parallels: deep neural networks and pattern recognizers act as a "System 1," while symbolic reasoning engines act as "System 2" [3] [4]. Pure statistical learners (like today's deep nets) excel at a-syntactic tasks (image recognition, speech) but struggle with logic, whereas symbolic systems do the opposite [5] [6]. **Neuro-symbolic AI** aims to bridge this gap by integrating both approaches [7]. For example, one opinion paper proposes using dual-process cognitive models to architect AI with a sub-symbolic intuitive system and a symbolic deliberative system working in tandem [6]. This suggests that PPP's pattern-recognition could serve as the sub-symbolic half, identifying abstract "shapes" in data, while higher-level symbolic processes refine or validate these patterns. Notably, some researchers argue that cognition isn't strictly binary – *"knowledge at any level of abstraction involves both symbolic and subsymbolic information"*, rather than a clean split [8] [9]. This perspective supports PPP's development alongside symbolic methods: even high-level abstract states may be processed via geometric projections, complemented by symbolic interpretation.

Machine Vision for Abstract States: The notion of using *machine vision* techniques on abstract state representations is cutting-edge. It implies treating internal data (say, high-dimensional state vectors or graphs of a system) as if they were images or spatial structures that a vision system can interpret. This resonates with research showing that neural networks can sometimes solve abstract tasks by *visualizing* them. For instance, conceptual-space models and attention schemas create spatial mappings of non-visual information so that pattern-based "visual" processing can occur [10] [11]. A concrete example is the **Neuro-Symbolic Concept Learner**, which learns visual concepts and language simultaneously: it turns scenes into an *object-based scene representation* and even translates words into symbolic programs, marrying perception with reasoning [12] [13]. That model's success in visual question-answering suggests that abstract problems can be encoded into a form where vision-like pattern recognition solves part of the task. For PPP, this means you could project an *abstract state* (like a complex system status or a high-dimensional vector from an LLM) into a geometric structure (maybe an image or spatial graph) and then apply computer vision or spatial algorithms to detect patterns or anomalies. This idea dovetails with emerging research on **geometric deep learning**, where neural networks operate on manifolds and graphs – effectively treating relational or high-dimension data in a spatial manner. It also aligns with cognitive theories that humans use spatial metaphors for abstract thinking (e.g., mental number lines, spatial memory of concepts). In practice, PPP could leverage convolutional neural networks or pattern-matching on visual encodings of, say, an agent's internal state matrix.

Hybrid and Hierarchical Agent Architectures: The **Hierarchical Agentic Operating System (HAOS)** paradigm envisions an operating framework composed of multiple agents with different roles and layers. This is in line with *hybrid cognitive architectures* and *hierarchical multi-agent systems*. In AI, hierarchical agent systems organize agents in layers of control, much like a company with executives, managers, and workers [14]. Higher-level agents handle abstract goals and planning, while lower-level agents execute specialized tasks. Such structuring is not only conceptually clean but has precedent: **Hierarchical Reinforcement Learning** (HRL) uses a high-level policy to choose sub-policies ("options") for low-level actions. Similarly, multi-agent frameworks often employ a *meta-controller* agent overseeing sub-agents. This paradigm has roots in cognitive architectures like **Soar** and **CLARION**. Soar's design, for example, is modular and hierarchical, attempting to replicate the mind's problem-solving by using a universal subgoal mechanism [15]. CLARION explicitly separates a subconscious *connectionist* component and a conscious *symbolic* component, merging their strengths [16]. These architectures show how a system can be both hierarchical and hybrid – exactly the kind of blueprint HAOS could draw upon. In essence, HAOS can be thought of as a cognitive architecture or **operating system for AI agents**, where each module is an agent and their interactions emulate a mind's sections (memory, vision, decision-making, etc.). *High-dimensional geometric cognition (PPP)* could be one layer in this OS (a perceptual or pattern-recognition layer), feeding into symbolic reasoning agents in another layer, creating an integrated whole.

Key Labs, Institutions, and Scholars to Watch

Developing PPP and HAOS will benefit from engaging with research groups working on *cognitive AI, neurosymbolic integration, and agent architectures*. Below are some leading contenders:

- **Neurosymbolic and Dual-Process AI Labs:** Several teams explicitly bridge neural and symbolic AI. For example, the MIT-IBM Watson AI Lab (Cambridge, MA) has pioneered neurosymbolic models like the Neuro-Symbolic Concept Learner [12]. IBM Research's groups (led by scientists like David Cox and collaborators) view neuro-symbolic AI as a path to more robust intelligence [7]. Similarly, **Mila (Montreal Institute for Learning Algorithms)** under Yoshua Bengio has explored ideas like the *Consciousness Prior* and System 2 deep learning (inspired by dual-process theory) [6]. Academic efforts, such as the one by *Stanford's AI Lab* (with Fei-Fei Li and Jiajun Wu), combine vision, language, and reasoning – Jiajun Wu's work on blending 3D scene understanding with symbolic inference is directly relevant. Another notable scholar is *Josh Tenenbaum at MIT*, whose Computational Cognitive Science group works on Bayesian concept learning and probabilistic programs that combine perception and abstract reasoning (e.g., learning physical commonsense from visual scenes).

Engaging with these labs could provide theoretical grounding and potential collaboration on PPP-like representations.

- **Cognitive Architecture and Hybrid AI Groups:** Labs that develop cognitive architectures or symbolic–neural hybrids can offer blueprints for HAOS. *Carnegie Mellon University (CMU)* hosts research on **ACT-R**, a leading cognitive architecture modeling human memory and multitasking (John Anderson's lab). *University of Michigan* (John Laird's group) maintains **Soar**, one of the oldest general cognitive architectures for intelligent agents. These groups blend insights from psychology with AI engineering. Similarly, *RPI and IBM* have researchers like **Selmer Bringsjord** and **Stuart Shapiro** working on cognitively inspired AI. On the connectionist-symbolic frontier, *professor Gary Marcus* (NYU/formerly) has been a vocal proponent of hybrid systems (though not tied to a single lab, his writings and startup, Robust.AI, reflect this approach). *Ron Sun* (at Rensselaer Polytechnic) developed CLARION, explicitly uniting neural networks with symbolic rule learning in a dual-process fashion [16]. These scholars and their labs offer a wealth of experience in building agent systems that *think* more like humans, which aligns with HAOS's long-term vision.

- **Dual-Process Cognition and Memory Research:** A number of interdisciplinary researchers are applying dual-process (System 1/System 2) ideas to AI. **Stanford's Center for Human-Centered AI (HAI)** has folks like *Noah Goodman* and *Christopher Manning* exploring language understanding with probabilistic logical back-ends to neural nets. *DeepMind* (London) should also be noted – they have published on **neural memory networks** and **Differentiable Neural Computers (DNC)** which marry algorithmic reasoning with neural learning. In fact, DeepMind's DNC (by Alex Graves et al.) is a *memory-augmented neural network* that can learn to store and retrieve complex data structures, even performing algorithmic tasks like graph traversal [17]. This work shows the power of combining sub-symbolic learning with explicit memory and could inform the "memory subsystem" of a HAOS. On the academic front, *Henry Kautz* (University of Rochester) has been evangelizing neurosymbolic AI; his 2022 AAAI Engelmore Lecture described a neurosymbolic agent that uses a neural perception module (System 1) to invoke symbolic problem-solving (System 2) when needed [10] [11]. Engaging with Kautz's work (and related groups like the University of Texas's neurosymbolic program led by *Peter Stone* or *Ray Mooney*) could yield practical frameworks for PPP–HAOS integration.

- **Institutions and Consortia:** There are also organized research programs to be aware of. **DARPA** (U.S. Defense Advanced Research Projects Agency) has in recent years funded programs on "third wave AI," which emphasize contextual reasoning and combining learning with knowledge (programs like *Lifelong Learning Machines, Machine Common Sense,* etc.). The **Allen Institute for AI (AI2)** in Seattle runs projects on common sense reasoning and has open-source tools (like Mosaic and others) that might intersect with PPP's aims of understanding abstract scenarios. In Europe, the **European Laboratory for Learning and Intelligent Systems (ELLIS)** has a focus on foundational AI – scholars like *Luc De Raedt* in KU Leuven (neurosymbolic learning) and *Kristian Kersting* (TU Darmstadt) work on statistical relational learning and could be allies in developing your paradigm. Connecting with such institutions via workshops or informal collaborations can situate the PPP/HAOS concepts among well-respected efforts and provide feedback from a broad range of experts.

In summary, look to engage with **neurosymbolic AI pioneers**, **cognitive architecture veterans**, and **dual-process cognition researchers**. They provide both the *theoretical underpinnings* (e.g. how to combine geometric pattern modules with logic) and *practical frameworks* (existing architectures, memory systems, etc.) that can accelerate PPP and HAOS from idea to implementation.

Tools and Frameworks for Building PPP/MVEP Testbeds

To move from theory to a working prototype, you'll want to assemble a *testbed environment* that lets you experiment with PPP's polytopal patterns and HAOS's agent hierarchy. Fortunately, there are many open-source tools and frameworks geared toward high-dimensional data processing, vision, and agent simulation:

- **GPU-Accelerated 3D Simulation Engines:** Since PPP might involve spatial-temporal patterns and HAOS involves multiple agents, leveraging game/simulation engines is valuable. **Unity3D with the ML-Agents Toolkit** is a prime example. Unity's ML-Agents allows you to turn any 3D scene into a playground for AI, where agents (with sensors, cameras, etc.) can learn via reinforcement learning, imitation learning, or other methods [18] [19]. Unity's advantage is real-time GPU rendering, physics, and a Python API for AI training. You could, for instance, create a **Multi-Visual Environment Platform (MVEP)** inside Unity – a sandbox world with complex visual scenes or abstract geometries – and have your PPP algorithm attempt to interpret those scenes while HAOS's hierarchical agents navigate or manipulate them. Another emerging tool is **NVIDIA Omniverse**, which is designed for scalable simulation and digital twins. Omniverse lets you *"simulate and test physical AI and robotic fleets at scale in digital twin environments"* before real-world deployment [20]. It supports multiple GPUs and real physics, which could be useful if PPP needs to process high-dimensional sensory input from multiple agents in a realistic world. Alternatively, **Unreal Engine** (with plugins like AirSim for drones or CARLA for autonomous cars) can serve similar purposes, though Unity's community and ease-of-use are often better for quick prototyping. These environments give you visually rich, physics-enabled worlds to test an agent's perceptual and cognitive capabilities – essential for demonstrating PPP's machine-vision-like abstract pattern recognition in action.

- **Spatial-Temporal Processing Frameworks:** For analyzing patterns over space and time (e.g., sequences of high-dimensional projections), libraries like **OpenCV** and **Open3D** are very handy. **OpenCV**, the Open Source Computer Vision Library, contains over 2,500 algorithms for image and video analysis [21]. It's optimized in C++ with Python bindings and can utilize GPU acceleration (via CUDA). OpenCV can handle tasks like object detection, feature extraction, optical flow, and more – all of which might be repurposed for PPP. For example, you could project a multidimensional state trajectory onto a 2D image (perhaps a heatmap or graph diagram) and use OpenCV edge detection or template matching to recognize recurring "shape" patterns in state transitions. If PPP involves 3D geometric data (polytopes in a mathematical sense or point clouds), **Open3D** is an open-source library for 3D data processing (point cloud segmentation, spatial partitioning, etc.). It would let you create and manipulate polytopal shapes and compute projections or intersections, which could directly test PPP's algorithms on synthetic geometric data. Additionally, for time-series or streaming data, consider frameworks like **Apache Kafka** or simply Python's async tools to feed real-time data into your system, ensuring PPP and HAOS can handle continuous inputs.

- **Machine Learning and Vision Models:** To bootstrap development, leverage pre-built models in **deep learning frameworks**. For instance, **PyTorch** (with TorchVision) and **TensorFlow** come with models for image recognition, segmentation, and even video understanding. You might integrate a **pre-trained vision transformer or CNN** to serve as a pattern-recognizer on PPP's outputs. There are also specialized models like **OpenAI's CLIP** (which connects text and image embeddings) that might inspire how to connect geometric patterns with semantic labels. If PPP aims for a novel form of "a-syntactic" perception, you could compare it against these baseline vision models on tasks (does

PPP detect an abstract pattern faster or more generally?). Moreover, **OpenCV's DNN module** allows running pretrained neural nets (from Caffe, ONNX, TensorFlow) efficiently, which could be handy if you want to incorporate neural detectors into your PPP pipeline without switching contexts.

- **Agent Simulation and Reinforcement Learning Toolkits:** For testing HAOS, consider using **OpenAI Gym/Gymnasium** environments or the newer **DeepMind Gymnasiums**. Gym provides a standard API and a suite of environments for reinforcement learning [22]. Even though Gym's tasks (like CartPole, Atari games, robotic arm control) are somewhat narrow, they can be excellent for prototyping a hierarchical agent. For example, you could take a complex Gym environment (say a maze or puzzle) and deploy a HAOS-based solution: a top-level agent decides high-level strategy (e.g., explore vs. exploit), while a low-level agent (guided by PPP's pattern recognition) handles immediate action selection. The **PettingZoo library** is another extension that supports multi-agent environments, which could allow testing multiple agents cooperating or competing. If your MVEP concept is more custom, Gym makes it straightforward to wrap your Unity or custom simulator in a Python interface for training and testing algorithms. Also noteworthy: **ROS (Robot Operating System)** if you venture into real robots or hardware-in-the-loop simulation. ROS with Gazebo simulator can integrate sensor data streams (camera, LIDAR, etc.) – PPP could ingest those high-dim sensor readings, and HAOS agents could control robot behavior, providing a very tangible demo of the paradigm.

- **Graphics and Visualization Tools:** Because PPP deals with polytopes and projections, robust visualization will be important for development (and for impressing others!). Tools like **matplotlib (Python)** or **Plotly** can plot high-dimensional data projections. For interactive 3D visuals, Unity's editor or Blender can be used to manually inspect how PPP is transforming data. If you require custom GPU computing for, say, real-time projection of 1000-dimensional vectors, consider using **CUDA or OpenCL** kernels (via libraries like CuPy in Python) to speed up linear algebra on GPUs. The good news is that many deep learning frameworks already handle high-dim tensor operations with ease – you can treat a batch of high-dim vectors as a "image" and use GPU matrix ops directly.

In summary, assembling a testbed might involve a *3D simulation engine* (Unity/Omniverse) for a realistic multi-agent environment, combined with *vision and data processing libraries* (OpenCV, PyTorch) to implement PPP, and using *RL/multi-agent frameworks* (Gym, PettingZoo, Unity ML-Agents) to realize and evaluate the HAOS control structure. This combination will let you prototype quickly: you can start an agent in a simple gridworld or visual puzzle, use PPP to analyze the state, and use a hierarchical agent from an RL library to decide actions – all with existing tools glued together.

Communities, Conferences, and Open-Source Ecosystems for Collaboration

To gain feedback, collaborators, and visibility, you should immerse yourself in the communities at the intersection of AI cognition, vision, and agent systems:

- **Academic Conferences and Workshops:** Top AI conferences like **NeurIPS (Neural Information Processing Systems)**, **ICLR (Intl. Conf. on Learning Representations)**, **AAAI (Association for the Advancement of AI)**, and **IJCAI (Intl. Joint Conf. on AI)** regularly feature workshops or tracks on hybrid approaches and agent-based systems. For example, the **NeSy (Neural-Symbolic Learning**

and Reasoning) workshop series has been running since 2005 and is the premier forum for neuro-symbolic AI research [23] [24]. Attending or submitting to NeSy could plug you into the latest work on combining neural pattern recognition with symbolic reasoning (exactly PPP+HAOS territory). Similarly, at NeurIPS 2023 there were workshops on *Geometry in Neural Representations* and on *Cognitive Architectures for AI*, indicating high interest in these topics. The **AAMAS (Autonomous Agents and Multiagent Systems)** conference is ideal for the HAOS side – it's a gathering of researchers building agent societies, hierarchical teams, and multi-agent learning. Look out for workshops like "Agent Societies in Practice" or "Hierarchical Reinforcement Learning" at such venues. Additionally, interdisciplinary conferences like **CogSci (Cognitive Science Society)** or **ICCM (International Conference on Cognitive Modeling)** could be places to present a more cognitive/psychologically-inspired aspect of your paradigm, if you draw on human cognition analogies.

- **Open-Source AI Communities:** An excellent way to accelerate development and gain credibility is engaging with open-source projects. The recent surge of interest in *agentic AI* (spurred by tools like AutoGPT, BabyAGI, etc.) has led to vibrant communities on GitHub and Discord. **AutoGPT**, for instance, is an open-source project showcasing how an LLM (GPT-4) can autonomously break down goals into sub-tasks and solve them, using a loop of planning, executing, and learning [25]. While AutoGPT itself is primarily an LLM orchestrator, the *ecosystem* around it (forums, extension developers, YouTubers testing agents) can provide both inspiration and a user base for your HAOS prototype. Open-source frameworks like **LangChain** (which helps connect LLM "agents" with tools) and **Microsoft's Autogen** (a framework for multi-agent LLM systems) are evolving quickly. You could contribute to these projects or integrate them – for example, using LangChain to manage communications between sub-agents in HAOS, or plugging PPP's pattern analysis as a tool that an LLM agent can call. By sharing parts of your work (say a PPP module on GitHub that takes in data and returns a pattern description), you'll attract users and collaborators who are already interested in agentic systems. Platforms like **Hugging Face** are also expanding to multimodal and agent frameworks – you might publish an open-source package or even a demo on Hugging Face Spaces to show PPP and HAOS in action to that community.

- **Multimodal AI and Vision+Language Communities:** Since PPP touches on visual/spatial processing and HAOS could involve language (if agents communicate or explain their reasoning), it's worth engaging with multimodal AI groups. There are workshops like **Visuo-Linguistic AI** at CVPR or **Multimodal Reasoning** at EMNLP that focus on combined vision & language understanding. These could be relevant if, for example, your system needs to *explain* the geometric patterns it finds (bridging to language) or use language as a tool for higher-level agents. The **Vision-Language models** community (working on models like CLIP, Visual Question Answering, etc.) might be interested in PPP as a novel way to represent visual knowledge. In fact, you can cite on arXiv under categories like cs.AI (Artificial Intelligence), cs.CV (Computer Vision), and cs.MS (Multiagent Systems) simultaneously – *arXiv* is a great venue to publish an early "whitepaper" or prototype results, as it will be seen by people in multiple domains. ArXiv's rich categorization means your work on PPP/HAOS can be tagged under **Artificial Intelligence, Machine Learning, Computer Vision, and Multiagent Systems**, signaling the interdisciplinary nature and attracting diverse readers.

- **Niche Communities and Forums:** Don't overlook specialized communities such as **the AGI (Artificial General Intelligence) research community** – they have their own conference (AGI) and often discuss cognitive architectures and broad approaches. Your vision for HAOS as an "operating system" for general problem-solving agents would resonate there. There are also online forums like the **LessWrong/AI Alignment** community or **Effective Altruism tech circles** where forward-looking AI paradigms are discussed; presenting PPP/HAOS as an approach to more *interpretable, human-like AI* might garner interest (and even funding, which we'll touch on next). Additionally, consider organizing a **workshop or meetup** yourself. For instance, a *NeurIPS workshop on High-Dimensional Cognition for Agents* or an *AAAI spring symposium* on "Geometric Representations in Cognitive Architectures" could establish you as a thought leader bringing others together. This is a longer-term play, but starting by attending and speaking at existing workshops is the first step.

- **Conferences to Prototype/Validate:** When you have an initial prototype, venues like **NeurIPS Demonstrations track**, **ICRA (Intl. Conf. on Robotics and Automation)** if there's an embodied aspect, or **CHI (Computer-Human Interaction)** if HAOS has a user-facing element (it *is* an OS, after all) could be places to demo. Also, the **NeurIPS agentic AI hackathons** or **OpenAI/ELEUTHER AI online events** often welcome demos of new agent architectures. By sharing your progress in these circles, you not only get validation but might also find early adopters or co-developers for open-source collaboration.

In short, become a part of the **hybrid AI conversation** happening across these platforms. By contributing thoughts or code, you gain credibility. It will also help you keep tabs on parallel developments (ensuring PPP/HAOS stays state-of-the-art) and possibly recruit talent – e.g., a PhD student from a lab you met at NeSy workshop might join your effort. The goal is to position yourself and your paradigm at the *center of an emerging ecosystem* of agentic computing, neurosymbolic learning, and high-dimensional AI.

Development Scaffolds, Testing Environments, and Cognitive Architecture Parallels

Moving from concept to prototype requires a solid *development scaffold* – essentially, a structure that lets you implement ideas incrementally and test them. Fortunately, parallels from cognitive architectures and modern AI toolkits provide guidance:

- **Cognitive Architecture Scaffolds:** Consider starting with an existing cognitive architecture as a foundation for HAOS. Frameworks like **SOAR** and **ACT-R** are available (Soar has a public domain C++/Python implementation, ACT-R is open-source from CMU in Common Lisp with Python bridges). These architectures come with built-in modules for memory, decision-making, etc., which could save you time. For example, Soar's problem-space planning and subgoaling mechanism might be repurposed to implement a hierarchical agent hierarchy (Soar naturally supports hierarchical goals and operators) [15]. Meanwhile, ACT-R provides a structure of "buffers" (like short-term memory stores) and production rules; one could imagine PPP acting as a perceptual module feeding into an ACT-R buffer, and HAOS logic encoded as production rules that decide when to invoke PPP or when to use symbolic reasoning. Another interesting framework is **OpenCog** (and its new incarnation OpenCog Hyperon). *OpenCog Hyperon* is an open-source cognitive architecture that explicitly aims to combine multiple AI paradigms (neural nets, logic, evolutionary algorithms) in a unified system towards AGI [26]. It provides a knowledge representation (the "Atomspace" graph database) and a cognitive loop. Integrating PPP into OpenCog's Atomspace as a geometric reasoning module, or using OpenCog's scheme to manage multiple agents' knowledge, could jump-start the HAOS implementation. The advantage of leveraging such architectures is that they embody decades of design for how to structure an intelligent agent: adopting their best features (e.g., a global working memory, or a learning mechanism for rules) will give PPP/HAOS a proven backbone. Even if you don't use one wholesale, studying their source can inform your custom scaffold.

- **Modern Agent Frameworks:** In parallel, modern "agentic" frameworks tied to LLMs can serve as a lightweight scaffold. For instance, **LangChain** (Python library) enables chaining of LLM calls and tools into coherent agent behavior. You could use LangChain to create a simple HAOS prototype: define a top-level agent (an LLM prompt that decides what subtask to do), and several tool agents (one could be a Python function for PPP that analyzes data, another could be a search tool, etc.). While LangChain's typical use-case is single-agent with tools, it can coordinate multiple calls and even has concepts of memory. **HuggingGPT** (by Microsoft) showcased how an LLM can orchestrate a set of specialized AI models (image captioning, math solver, etc.) by parsing a request and delegating tasks [27]. This is conceptually similar to HAOS: a high-level "manager" decides which specialist (vision, language, planning) to invoke. You can mimic that architecture by using an LLM (like GPT-4 via API or a local LLaMA model) to act as the planner agent and call your PPP module when a pattern-recognition task is needed. Such a scaffold lets you get something working quickly without building everything from scratch – essentially "gluing" together AI components in a hierarchical decision loop. As your own code for PPP and HAOS logic matures, you can gradually replace the placeholder LLM or tool-calls with more autonomous code.

- **Testing Environments and Benchmarks:** Having a set of concrete tasks to validate PPP/HAOS is vital. You want tasks that require both intuitive pattern recognition and higher-level reasoning. The **Abstraction and Reasoning Corpus (ARC)** by François Chollet is an excellent benchmark in this vein. It's a collection of human-like pattern puzzles (e.g., transforming input grids to output grids following an abstract rule) that *"evaluate an AI's ability to perform human-like abstract reasoning, rather than rely solely on statistical pattern recognition."* [28]. Solving ARC puzzles would be a litmus test for PPP's a-syntactic pattern recognition (since they often involve visual analogies) combined with HAOS's reasoning (figuring out the rule from few examples). You can create a test harness for ARC tasks to see if your system can generalize patterns like humans. Another domain: **games and virtual environments**. Text-based games (e.g., AI Dungeon-style or even interactive fiction) could test an agent's ability to perceive state (PPP could encode the game state as a vector) and plan (HAOS chooses actions). For spatial challenges, consider **GRID-world puzzles or maze navigation** in a simulator – these allow explicit geometric patterns (ideal for PPP) and require hierarchical planning (find keys before door, etc., ideal for HAOS). Robotics tasks, like having a robot arm stack blocks in a certain pattern, could also demonstrate PPP extracting a goal pattern and HAOS organizing sub-tasks to achieve it.

- **Memory and Knowledge Integration:** Part of developing a functional HAOS is handling memory (short-term and long-term). Cognitive architectures often have episodic memory modules or semantic knowledge bases. In a development context, you might use a **vector database** (like FAISS or Milvus) to store embeddings of experiences, essentially a modern "episodic memory" the agents can query. PPP could be used to encode experiences into high-dimensional vectors (polytopal features) and store them for later pattern matching. For a symbolic knowledge base, consider using **knowledge graphs** (Neo4j or RDF stores) to represent relations that the symbolic part of HAOS can use. Your hierarchical agents might then consist of a lower level that does pattern-matching against vector memory and a higher level that queries a knowledge graph for known facts or plans. This interplay echoes the **Global Workspace Theory** in cognitive science, where multiple specialist processes (here, PPP as one specialist, symbolic reasoner as another) share a common "blackboard" or workspace. In software, your blackboard could be a simple in-memory store or a pub-sub system where agents post observations and read others' results.

- **Iterative Development and Testing:** As you build, adopt an iterative cycle: implement a small aspect of PPP or HAOS, then test on a simple scenario. For example, first get a PPP module working that takes some multi-dimensional data and outputs a simplified representation (like clustering or dimensionality reduction) – test if it can consistently recognize a pattern (like "data points forming a line vs a curve") on synthetic data. Then embed this in a minimal agent loop: maybe a loop that generates random data and uses PPP to signal if it "recognizes" a known pattern, and log the result. Simultaneously, prototype a hierarchical agent decision process using a conditional tree or state machine (even before integrating learning). Ensure that at each stage, you have *measurable outcomes* (pattern detected or not, task solved or not) so you can iterate. This incremental approach is akin to how one would test components of a cognitive architecture – e.g., in ACT-R, one might first test the vision module independently before running a full model of a task.

By using scaffolds from cognitive architectures and modern AI frameworks, you avoid reinventing the wheel and you ground your development in proven designs. This approach not only speeds up prototyping but also makes your system more explainable (since you can map parts of it to well-understood analogues in psychology or existing AI). It's like constructing a building: use a scaffold to shape the structure, pour the concrete (your custom algorithms), and then remove the scaffold once the structure can stand on its own. In the end, you'll have a tested, modular system – PPP modules handling perception, HAOS agents handling decisions – with the confidence that each piece has precedent in prior art and has been validated in isolation.

Funding Opportunities and Founder-Aligned Fellowships

Turning PPP and HAOS from research prototypes into a sustained project (or even a startup or research program) will likely require funding. Here are some paths to consider, spanning academic grants, industry programs, and visionary fellowships:

- **Research Grants (Academic and Institutional):** If you are in academia or partnered with an academic institution, look at grant programs supporting high-risk, high-reward AI research. For example, **NSF (National Science Foundation)** in the US has programs under its Robust Intelligence and AI divisions. NSF also runs **America's Seed Fund (SBIR/STTR)** which *"supports startups and small businesses to translate research into products and services, including AI systems"* [29] . An SBIR could fund a proof-of-concept, especially if you can pitch PPP/HAOS as having commercial potential (e.g., a new way for AI to analyze complex data with little supervision). Likewise, **DARPA** announcements (BAAs) in areas like "AI Explorations" or "Lifelong Learning" could be a fit – DARPA likes ambitious ideas like new AI architectures (you'd need a concise whitepaper and possibly a team including a university or company to bid on those). In Europe, consider **EU Horizon Europe** grants or **ERC (European Research Council)** grants if applicable – they often fund novel AI approaches (ERC Advanced Grants, for instance, give millions of Euros to single investigators pursuing high-risk ideas, which PPP/HAOS would qualify as). Also, keep an eye on **private research foundations**: the Sloan Fellowship, for example, or the **Simons Foundation** have funded AI/math researchers, and organizations like **Open Philanthropy** and the **Future of Life Institute** have issued grants for AI alignment and novel approaches to AI that could align with your vision for more human-like AI cognition.

- **Founder-Aligned Fellowships and Accelerators:** If you lean toward the entrepreneurial route (productizing the paradigm), there are accelerators and fellowships targeting AI startups and founders with visionary ideas. The **AI Grant** (started by Nat Friedman and Daniel Gross) is an example – it's an accelerator providing ~$250k funding on founder-friendly terms and cloud credits [30]. Recipients get support and computing resources, and they specifically look for ambitious, technically deep AI projects. Another example is **NVIDIA's Inception Program**, which *supports AI startups with resources, GPU credits, and networking opportunities* [31] [32]. While it doesn't give cash upfront, it provides technical support and credibility (and could offset the cost of the heavy GPU computing PPP might need). Large tech companies have their own programs too: e.g., **Google's TPU Research Cloud (TRC)** offers free compute to selected researchers, **Microsoft for Startups** provides Azure credits and mentorship for AI companies, and **Amazon AWS Activate** gives credits – all useful for training models and running experiments at scale.

- **Visionary and Innovation Prizes:** There are prestigious awards that, beyond money, confer significant recognition if you align your work with their goals. The **AAAI Squirrel AI Award for Artificial Intelligence for the Benefit of Humanity** is one – essentially the "Nobel Prize of AI" with a $1 million prize (awarded to people like Yoshua Bengio and Cynthia Rudin) [33] [34]. While that is more of a career award, aiming for such impact sets a high bar. More immediately, consider competitions like the **XPRIZE** series. In the past, there was an IBM Watson AI XPrize for AI solutions; future XPrizes might target AGI or human-machine collaboration – HAOS could be positioned as an "AI operating system" entry if such a competition arises. Additionally, **Kaggle competitions or benchmark contests** (though not huge money) could raise your profile: for instance, if you apply PPP to win a pattern recognition challenge or HAOS to win a multi-agent coordination challenge, it proves the approach. Even the **ARC Challenge** (François Chollet has mentioned a prize for any AI that can significantly crack the ARC dataset) could be in scope if PPP excels there. Check if Chollet's ARC Prize is still open – it was an open challenge for AGI-like problem solving.

- **Fellowships for Individuals:** If you as the founder/researcher are early in your career, programs like the **Thiel Fellowship** (which gives $100k over two years to young people to work on big ideas instead of college) could be an option – PPP/HAOS is certainly a bold idea that fits the mold of challenging the status quo. Another one is **Emergent Ventures** (run by Tyler Cowen at Mercatus Center), which gives fast, no-strings-attached grants to individuals with ambitious projects in science or tech; they've funded AI researchers and projects that don't fit traditional molds. Also, check **Schmidt Futures** – they have innovation fellowships and recently an **AI2050** program that funds long-term AI research aimed at AI in 2050; pitching HAOS as the "operating system for AI of the future" might resonate there. If you are a student or academic, **OpenAI, DeepMind, Microsoft Research, and Apple** all have PhD fellowship programs – a strong proposal around high-dimensional cognitive AI could secure a fellowship (which often includes tuition, stipend, and mentorship from the company).

- **Industry Partnerships and VC Funding:** In the longer run, if HAOS becomes something that could be commercialized (e.g., as a platform for building intelligent agents or a novel enterprise AI system), you might approach venture capital. The current climate for AI startups is very hot – VC firms have dedicated AI funds. Showcasing some prototypes and a vision of an "AI operating system" could attract seed funding. Particularly, **VCs aligned with long-term AGI** (e.g., Meta's Frontier fund, or entrepreneurs like Sam Altman who invest in moonshots) might be interested. In approaching these, it helps to have the credibility markers discussed above (publications, open-source users, perhaps a small grant already in hand). Even participating in an accelerator like **Y Combinator** (which has funded agentic AI startups recently) could be a path – YC has a Summer Fellowship that provided cloud credits and $20k grants to early-stage founders [35]. If your goal is to establish yourself as a *thought leader*, securing some funding or support from these channels will give you a platform to speak from. It shows that others believe in your paradigm's potential, which in turn can draw in more supporters and collaborators.

Next-Step Recommendations: Start by applying to a mix of opportunities that fit your situation. If you're in research mode, try an academic grant or fellowship (even a smaller one, like a university seed grant – e.g., Princeton and Johns Hopkins have internal AI seed funds for ~$25k [36] [37]). These can fund interns or compute to get a prototype done. In parallel, engage with an accelerator or mentorship program for the networking – NVIDIA Inception or AI Grant application could run concurrently. As you build momentum (say you publish a paper on arXiv and present at a workshop), use that news to approach larger funders (Open Phil, etc.), essentially saying: "This once-whitepaper idea is now a working demo with community interest; with funding X I can scale it up and change the field."

Remember that *funding is not just about money* – it's also about validation and network. Winning an innovation prize or grant places you in a community of like-minded innovators and gives you content to share (press releases, talks). As a founder or thought leader, that visibility is crucial. It will attract potential collaborators, team members, and even early users for your prototypes. Given the transformative ambition of PPP and HAOS, don't shy away from pitching it as such – many funders are specifically looking to support the next paradigm shift in AI. With a clear development roadmap and the right supporters, you can accelerate from a visionary white paper to an influential prototype that redefines how we think about intelligent systems.

Conclusion and Next Steps

Transitioning Polytopal Projection Processing and the Hierarchical Agentic OS from vision to reality is a multifaceted journey – involving deep theoretical grounding, building prototypes with the right tools, engaging communities, and securing support. In this report, we identified **key research areas** (from high-dimensional cognition to dual-process AI) that anchor your ideas in academic context, and pointed to **labs and experts** who are natural allies in this endeavor (bridging neural and symbolic, cognitive and computational). We explored practical **tools and frameworks** – from Unity simulations and OpenCV to cognitive architectures – that can serve as your experimental sandbox for PPP's geometric pattern-recognition and HAOS's agent orchestration. We also mapped out the **communities, conferences, and open-source ecosystems** where you can validate and broadcast your progress, ensuring you're plugged into the emerging agentic computing movement. In building out your system, leveraging existing **development scaffolds and cognitive frameworks** will provide a backbone, while focusing on concrete **benchmark tasks** (like ARC or multi-agent challenges) will yield demonstrable milestones. Finally, we surveyed **funding avenues** – academic grants for the foundational research, startup accelerators and fellowships for visionary builders, and even prizes – to fuel your work and enhance your credibility as a thought leader.

Actionable next steps: As you move forward, a practical sequence might be:

1. **Establish a Testbed:** Set up a simple Unity or Gym-based environment this quarter to start iterating PPP and HAOS ideas in code. For example, a grid-world where a "seer" agent (with PPP) observes

patterns and a “doer” agent (controlled by a hierarchical policy) must act. Keep it simple initially (limited dimensions, obvious patterns) and gradually increase complexity.

2. **Engage the Community:** Simultaneously, write a blog post or short arXiv paper outlining PPP and HAOS in the context of existing research (much of the content from this report can be repurposed for that). Announce what you’re building on forums (Reddit’s r/AGI or r/MachineLearning, the AI Alignment Forum, etc.) to gather initial feedback. Consider open-sourcing a preliminary module (e.g., a PPP-based pattern recognizer library). Even a rough code release can attract interest and contributors early on.

3. **Reach Out to Collaborators:** Contact a few key researchers for low-key collaboration or advisory roles. For instance, email the authors of a neurosymbolic paper you admired or a professor who works on cognitive architectures, briefly sharing your white paper and asking for a 30-minute discussion. This can lead to mentorship, or if you’re lucky, a joint project (academics might be interested in co-authoring if you have a testbed to try their theories on). Also, consider joining the mailing lists or Slack/Discord groups of workshops like NeSy – these often have ongoing discussions where you can float your ideas and find peers.

4. **Apply for Grants/Fellowships:** Draft proposals for two tracks – one academic (e.g., an NSF Early-Concept Grant or a university seed fund) and one entrepreneurial (e.g., AI Grant or NVIDIA Inception). Tailor the narrative: academic proposals should emphasize scientific questions (e.g., “investigating high-dimensional concept representations for reasoning”), whereas startup applications should highlight use-cases (e.g., “a new AI OS that can be used to build more autonomous virtual assistants or robotics”). Aim to submit at least one of each type in the coming 2-3 months. The process of writing these will also sharpen your project plan.

5. **Plan for a Milestone Demo:** Identify an event 6–12 months out where you could showcase progress – perhaps the demo track at NeurIPS or a relevant workshop. Mark a date and work backwards to a minimum viable prototype that you can present. A good demo could be: a live agent system solving a puzzle or a short video of a hierarchical agent (with visuals of its polytopal “thought process”) solving a task that traditional AI fails. This tangible milestone will focus development and also create an external buzz when achieved.

By following these steps, you’ll create momentum. The interplay of **short-term testbed viability** (getting something working soon) and **long-term strategic alignment** (framing it in the context of where AI is headed and who’s supporting that direction) will turn your visionary white paper into an *executable prototype with an audience*. Along the way, every paper you publish, demo you give, or grant you win will cement your position as a thought leader in agentic computing. The challenges are non-trivial – you are combining diverse disciplines – but as we’ve seen, the world of AI is converging toward exactly these kinds of integrations. Timing is on your side, and with careful positioning and persistent development, PPP and HAOS can go from an idea on paper to a reference architecture that others cite and build upon.

Lastly, keep in mind that while the end goal is grand (a new paradigm of AI), every step you take – be it a small experiment or a community post – is part of leading that paradigm shift. By relentlessly iterating and engaging, you’ll not only advance PPP and HAOS, but also inspire others to join in evolving the future of agentic, human-like computing.

Sources:

- Conceptual spaces and geometric knowledge representation [1] [2]
- Dual-process cognitive architectures and neurosymbolic integration [8] [7]
- Neuro-Symbolic Concept Learner (vision-language reasoning) [12] [13]
- Differentiable Neural Computer (neural network with memory for reasoning) [17]
- Hierarchical multi-agent structures and hybrid cognitive models [14] [16]
- NeSy workshop (Neural-Symbolic AI community) [23] [24]
- Unity ML-Agents and simulation environments for AI [18] [19]
- NVIDIA Omniverse for large-scale realistic simulation [20]
- OpenCV – open-source computer vision toolkit (for spatial-temporal pattern processing) [21]
- OpenAI Gym – standard API and environments for reinforcement learning agents [22]
- AutoGPT and agent ecosystems (LLM-based autonomous agents) [25]
- ARC benchmark for abstract reasoning (tests human-like pattern recognition) [28]
- RoboticsBiz overview of cognitive architectures (SOAR, ACT-R, CLARION, etc.) [16] [15]
- OpenCog Hyperon project (open-source AGI framework combining paradigms) [26]
- AI Grant accelerator and funding for AI startups [30]
- NVIDIA Inception program support for AI startups [31] [32]
- NSF/US government seed funding for innovative AI solutions [29]

---

[1] [1801.03929] Formalized Conceptual Spaces with a Geometric Representation of Correlations
https://arxiv.org/abs/1801.03929

[2] HD/VSA
https://www.hd-computing.com/

[3] [4] [8] [9] Neurosymbolic Systems of Perception and Cognition: The Role of Attention - PMC
https://pmc.ncbi.nlm.nih.gov/articles/PMC9163389/

[5] [6] Frontiers | Dual-process theories of thought as potential architectures for developing neuro-symbolic AI models
https://www.frontiersin.org/journals/cognition/articles/10.3389/fcogn.2024.1356941/full

[7] Neurosymbolic AI: Bridging Neural Networks and Symbolic Reasoning for Smarter Systems
https://www.netguru.com/blog/neurosymbolic-ai

[10] [11] Neuro[Symbolic] architecture illustrated by a mouse-maze domain. The... | Download Scientific Diagram
https://www.researchgate.net/figure/NeuroSymbolic-architecture-illustrated-by-a-mouse-maze-domain-The-System-1-agent-sees_fig15_360479310

[12] [13] The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision for ICLR 2019 - IBM Research
https://research.ibm.com/publications/the-neuro-symbolic-concept-learner-interpreting-scenes-words-and-sentences-from-natural-supervision

[14] Building Agentic Applications with LangGraph: Part 3— Hierarchical Agent Teams | by Shivansh Kaushik | Primastat | Medium
https://medium.com/primastat/building-agentic-applications-with-langgraph-part-3-hierarchical-agent-teams-3406fd02ceda

15 16 Comparing four cognitive architectures: SOAR, ACT-R, CLARION, and DUAL - RoboticsBiz
https://roboticsbiz.com/comparing-four-cognitive-architectures-soar-act-r-clarion-and-dual/

17 Differentiable neural computer - Wikipedia
https://en.wikipedia.org/wiki/Differentiable_neural_computer

18 19 ML-Agents Overview | ML Agents | 3.0.0
https://docs.unity3d.com/Packages/com.unity.ml-agents@3.0/manual/index.html

20 Omniverse Platform for OpenUSD | NVIDIA
https://www.nvidia.com/en-us/omniverse/

21 OpenCV - Open Computer Vision Library
https://opencv.org/

22 Gym Documentation
https://www.gymlibrary.dev/

23 24 NeSy2023
https://sites.google.com/view/nesy2023

25 How Multi-Agent AI is Changing the Sales Game in 2025 - Reply.io
https://reply.io/blog/multi-agent-ai/

26 OpenCog Hyperon
https://hyperon.opencog.org/

27 HuggingGPT - AI Agent Store
https://aiagentstore.ai/ai-agent/hugginggpt

28 Abstraction and Reasoning Corpus (ARC) - AI at work for all
https://shieldbase.ai/glossary/abstraction-and-reasoning-corpus-(arc)

29 35 Artificial Intelligence | NSF - National Science Foundation
https://www.nsf.gov/focus-areas/artificial-intelligence

30 31 32 36 37 Top 15 Grants for AI Startups in North America in 2025
https://www.femaleswitch.com/aiden_ai_app/tpost/nh1y5ff0b1-top-15-grants-for-ai-startups-in-north-a

33 Duke Professor Wins $1 Million Artificial Intelligence Prize, A 'New ...
https://pratt.duke.edu/news/rudin-squirrel-award/

34 INFORMS Member Wins $1 Million Artificial Intelligence Prize ...
https://www.informs.org/News-Room/INFORMS-Releases/News-Releases/INFORMS-Member-Wins-1-Million-Artificial-Intelligence-Prize-Equivalent-to-the-Nobel