ATPs as Universal AIs

What Do AGI Architectures Suggest for ATP Research?

– A motivational, conjectural talk

– Zarathustra Amareis Goertzel (AITP’24)

Outline

- What is Universal AI?
  - And Why Do ATPs Qualify?
  - No Free Lunch and the Need for Specialization.
- Cognitive Architectures: A Brief Overview.
  - Sketching the Architecture of E and ENIGMA.
- What Would an AGITP Architecture Look Like?
  - Autonomy, Worldviews, Self-organization, Metalearning, and Autoformalization.
  - What Could this Contribute to the ATP/ITP field?
  - Could an AGITP Consume the Real World?

Universal AI

Universal AI

- Ideally, solves any *well-defined problem* (nearly) *optimally* (in the limit).
-

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
  - Given a function *f* and a value *y*, it methodically enumerates and evaluates all programs until it finds one that outputs a value *x* such that *f(x) = y*.
  - The exponential run-time depends on the length of the program, so it's constant relative to the size of the input, offering deceptively good computational complexity.

A cute video illustrating the basic idea:
https://www.youtube.com/watch?v=9ONm1od1QZo&

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
  - Given a function *f* and a value *y*, it methodically enumerates and evaluates all programs until it finds one that outputs a value *x* such that *f(x) = y*.
  - The exponential run-time depends on the length of the program, so it's constant relative to the size of the input, offering deceptively good computational complexity.
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.

Links:

- http://www.hutter1.net/ai/pfastprg.htm
- https://people.idsia.ch/~juergen/optimalsearch.html

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
  - Given a function *f* and a value *y*, it methodically enumerates and evaluates all programs until it finds one that outputs a value *x* such that $f(x) = y$.
  - The exponential run-time depends on the length of the program, so it's constant relative to the size of the input, offering deceptively good computational complexity.
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.

Links:

- https://people.idsia.ch/~juergen/oops.html
- https://people.idsia.ch/~juergen/goedelmachine.html

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.
- AIXI is a theoretical reinforcement learning agent that maximizes expected future reward weighted by all computable environments consistent with the past (a la Solomonoff Induction).
    - It's also incomputable.
    - … and subjectivity enters via the prior (the UTM).
    - … nor is it knowledge-seeking enough to be asymptotically optimal.

Links:

- http://www.hutter1.net/ai/uaibook.htm#oneline
- https://jan.leike.name/AIXI.html

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.
- AIXI is a theoretical reinforcement learning agent that maximizes expected future reward weighted by all computable environments consistent with the past (a la Solomonoff Induction).
    - It's also incomputable.
    - … and subjectivity enters via the prior (the UTM).
    - … nor is it knowledge-seeking enough to be asymptotically optimal.

    - See a Monte-Carlo Tree Search AIXI approximation play Pac-Man: https://www.youtube.com/watch?v=yfsMHtmGDKE

Links:

- http://www.hutter1.net/ai/uaibook.htm#oneline
- https://jan.leike.name/AIXI.html

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.
- AIXI is a theoretical reinforcement learning agent that maximizes expected future reward weighted by all computable environments consistent with the past (a la Solomonoff Induction).

- Refutation-Complete Automated Theorem Proving: if a (higher-order) clause set is unsatisfiable, then the ATP can derive a proof of the empty clause.
  - Constructive systems even provide programs from the proofs.
  - Can these subsume the above techniques (in theory)?

Links:

- https://abentkamp.github.io/ for HO Superposition papers

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.
- AIXI is a theoretical reinforcement learning agent that maximizes expected future reward weighted by all computable environments consistent with the past (a la Solomonoff Induction).

- Refutation-Complete Automated Theorem Proving: if a (higher-order) clause set is unsatisfiable, then the ATP can derive a proof of the empty clause.
  - Constructive systems even provide programs from the proofs.
  - Can these subsume the above techniques (in theory)?
- <u>Provocation</u>:
  - Automated theorem proving may be the most effectively developed approach to universal AI.

Universal AI

- Ideally, solves any well-defined problem (nearly) optimally (in the limit).
- Levin's Universal Search arguably solves *inversion problems* about as well as one can in general:
- Hutter's Algorithm searches through proofs of program correctness and time-boundedness to find programs computing well-defined functions, shifting the burden to an additive constant.
- Schmidhuber adds the Optimal Ordered Problem Solver to incorporate previous solutions and the Gödel Machine that also rewrites its own code when it finds proven improvements.
- AIXI is a theoretical reinforcement learning agent that maximizes expected future reward weighted by all computable environments consistent with the past (a la Solomonoff Induction).

- Refutation-Complete Automated Theorem Proving: if a (higher-order) clause set is unsatisfiable, then the ATP can derive a proof of the empty clause.
  - Constructive systems even provide programs from the proofs.
  - Can these subsume the above techniques (in theory)?
- <u>Provocation</u>:
  - Automated theorem proving may be the most effectively developed approach to universal AI.
  - So let's act like it and treat ATPs/ITPs as proto-AGI systems!

AGI Cognitive Architectures

– Core Ideas

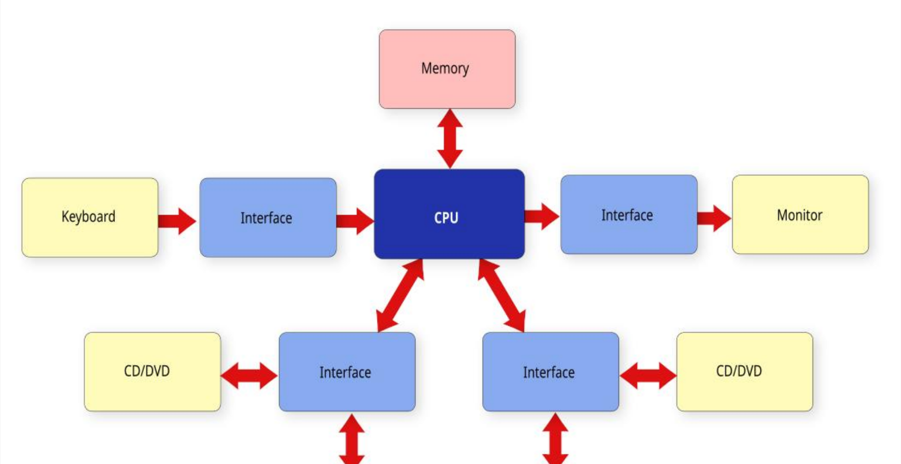
Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A *cognitive architecture* aims to do this for cognitive systems such as human minds or AI systems.

Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A *cognitive architecture* aims to do this for cognitive systems such as human minds or AI systems.

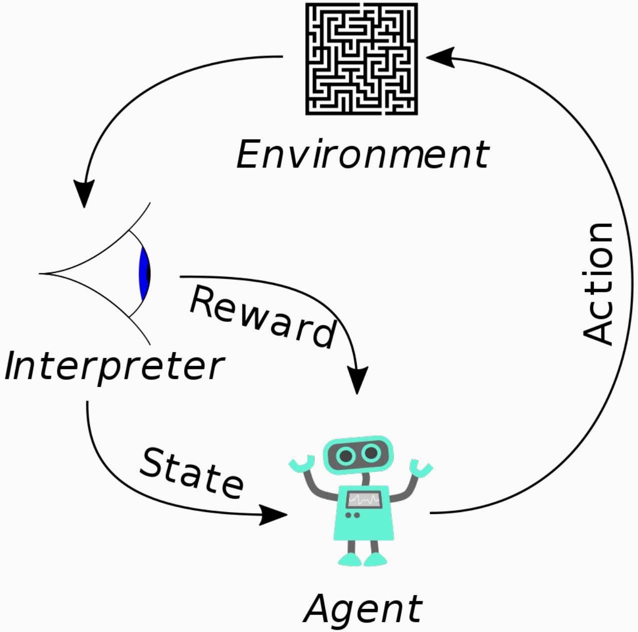
- Are architectures useful?
- E.g., if working with a computer, it's probably good to know that the *keyboard* and *CPU* are distinct functional (and physical) components to be treated separately, even if in theory it's all one quantum system.



Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A *cognitive architecture* aims to do this for cognitive systems such as human minds or AI systems.

- E.g., the RL World Architecture:
  - When is this model applicable?
  - … and when isn't it?



Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A cognitive architecture aims to do this for cognitive systems such as human minds or AI systems.

<u>Desiderata</u>:
- The cognitive architecture should capture all capacities necessary for functioning effectively in the system's environment.
- Sub-systems should be abstracted out as helpful and collapsed as they produce clutter.

Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A cognitive architecture aims to do this for cognitive systems such as human minds or AI systems.

Desiderata:

- The cognitive architecture should capture all capacities necessary for functioning effectively in the system's environment.
- Sub-systems should be abstracted out as helpful and collapsed as they produce clutter.

E.g.:

- Input/output processing often involves special operations (e.g., visual cortexes, classification, etc.)

Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A cognitive architecture aims to do this for cognitive systems such as human minds or AI systems.

Desiderata:
- The cognitive architecture should capture all capacities necessary for functioning effectively in the system's environment.
- Sub-systems should be abstracted out as helpful and collapsed as they produce clutter.

E.g.:
- Input/output processing often involves special operations (e.g., visual cortexes, classification, etc.)
- **Premise selection** is a good example.
  - Strictly speaking, it's not theoretically necessary.
  - Practically, the ATP will bloat without it.
  - Functionally, filtering background theory is very different from the deeper reasoning involved in proof search.

→ Structurally, «premise selection» should be a semi-distinct module.

Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A *cognitive architecture* aims to do this for cognitive systems such as human minds or AI systems.

Questions:

- Is modularized specialization going to be necessary in practical, resource-limited domains?
  - Will all effective approximations to *universal AI* involve architectures of k>1 components?
  - Note: a neural network's weights could form fuzzily distinct components even if we don't build it into the architecture

Cognitive Architectures

- A *system architecture* is a model that aims to capture the important components of a system and their relationships.
- A *cognitive architecture* aims to do this for cognitive systems such as human minds or AI systems.

Questions:

- Is modularized specialization going to be necessary in practical, resource-limited domains?
    - Will all effective approximations to *universal AI* involve architectures of k>1 components?
    - Note: a neural network's weights could form fuzzily distinct components even if we don't build it into the architecture

- "No Free Lunch"-style theorems suggest specialization will be necessary.
- We don't live in a random, uniform distribution:
    - Reality appears to be biased.

AGI Cognitive Architectures

– Brief Overview

Cognitive Architectures: SOAR

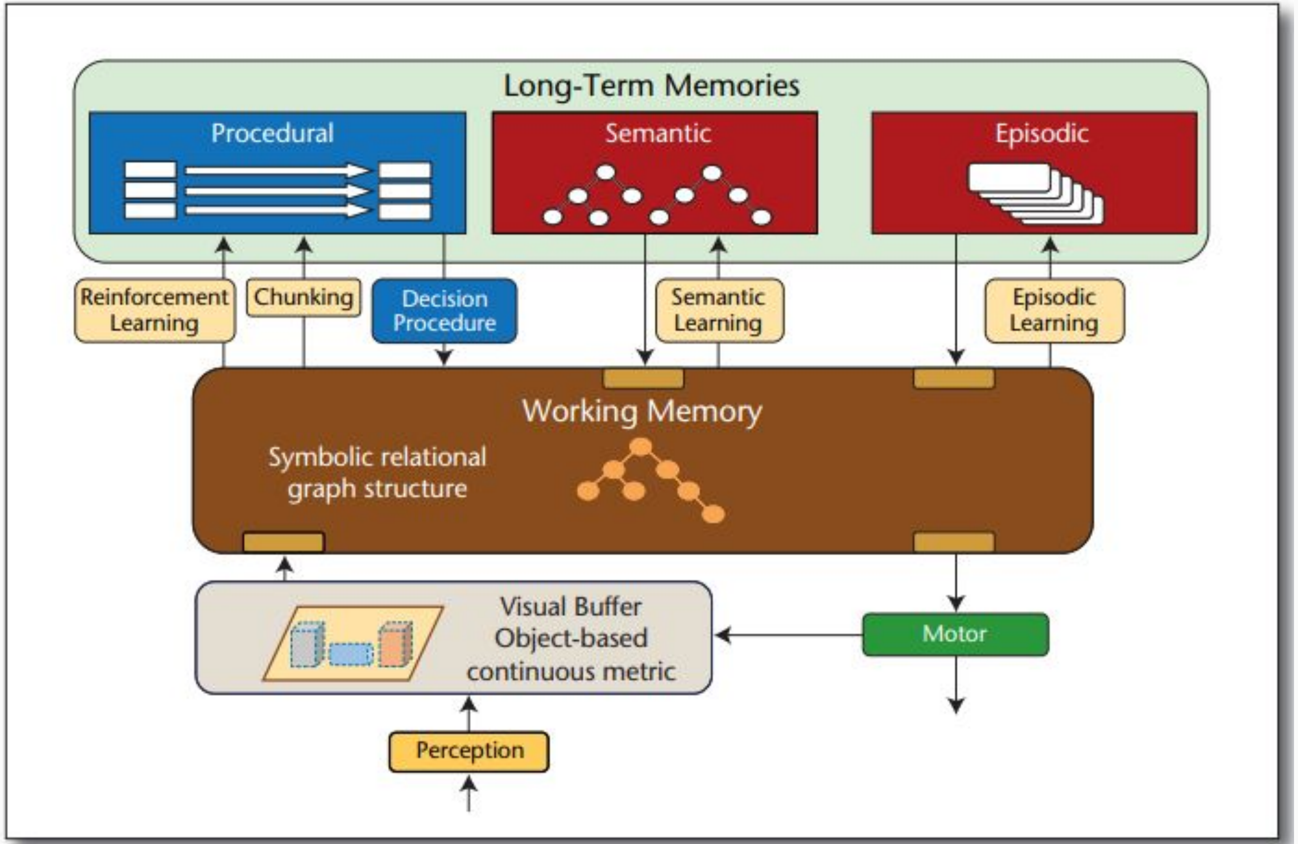
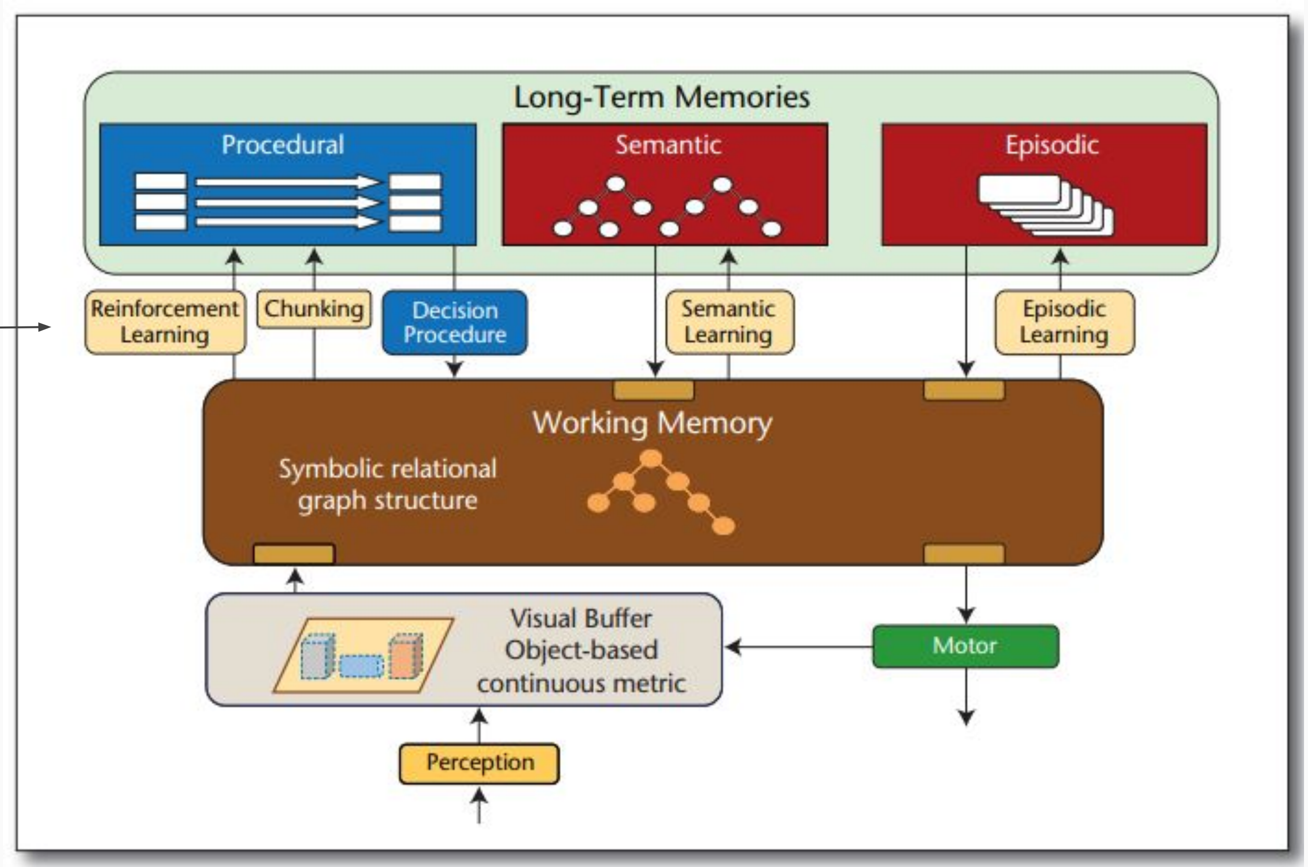


Image courtesy of "A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics" by Laird, Lebiere, and Rosenbloom.

Cognitive Architectures: SOAR

Premise Selection enters between Long-Term Memory and Working Memory!



CAs: Common Model of Cognition

- Abstracts consensus elements of many CAs.

- Note that there is no distinct component for 'goal' management.

- "Proofs as programs" merges procedural and declarative long-term memory? 😈

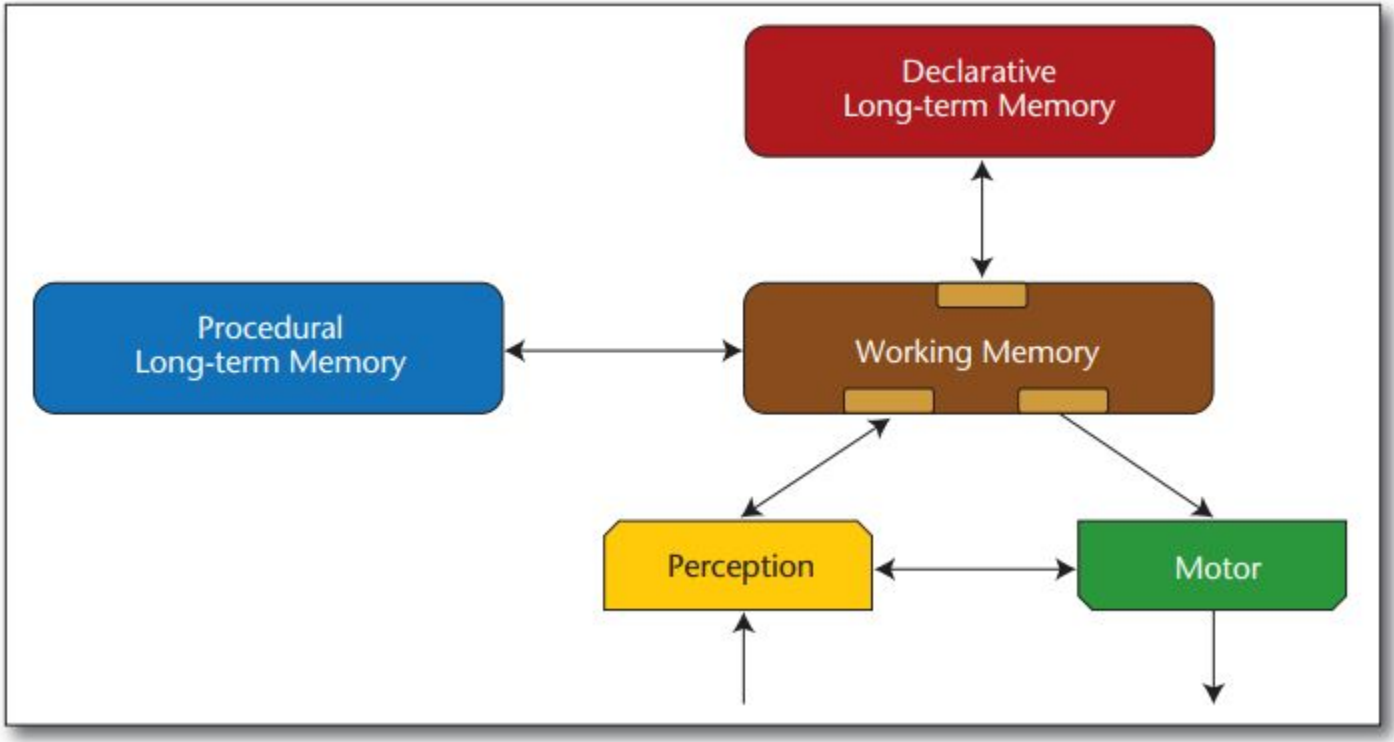


Image courtesy of "A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics" by Laird, Lebiere, and Rosenbloom.

Cognitive Architecture Examples

- An attempt to throw common CA elements together, leaning toward the supremum rather than infimum.

- Doesn't say much about the structure of the modules.

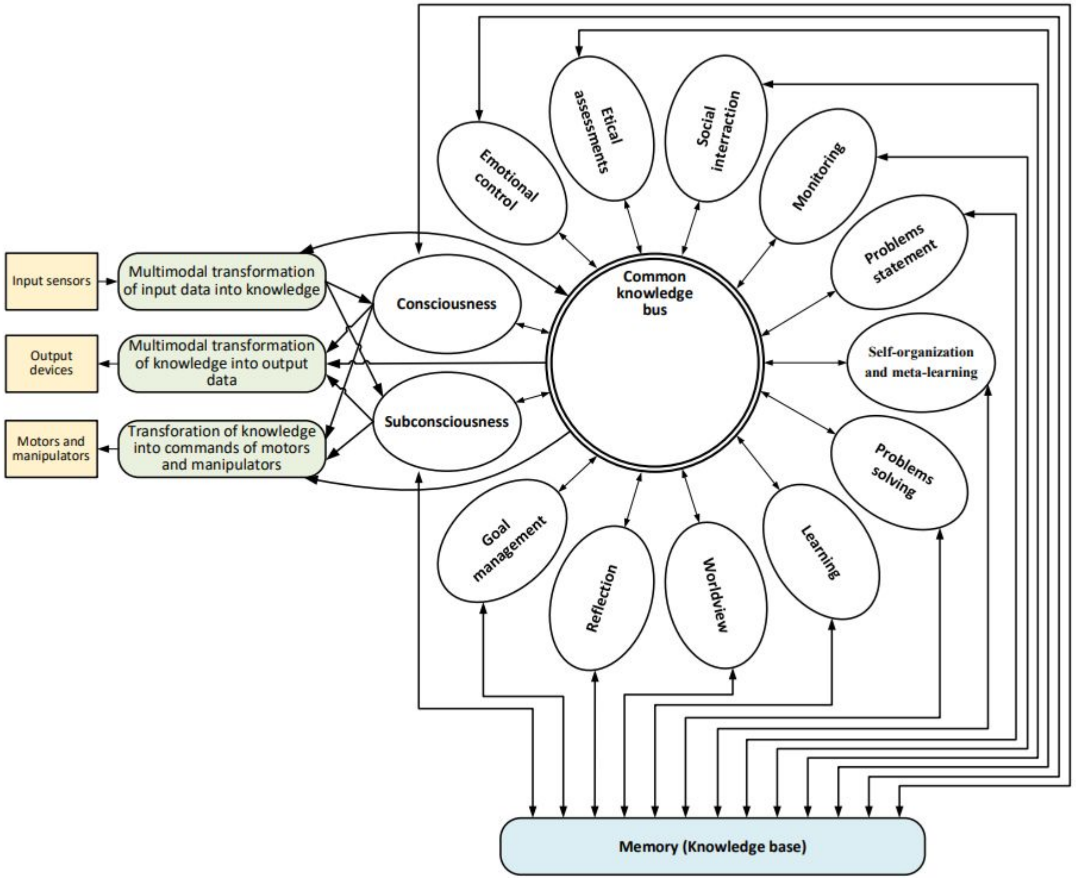


Image courtesy of "[A Universal Knowledge Model and Cognitive Architecture for Prototyping AGI]"

Cognitive Architecture Examples

- A suggestion of a "fully integrated cognitive architecture" with a knowledge-base substrate.

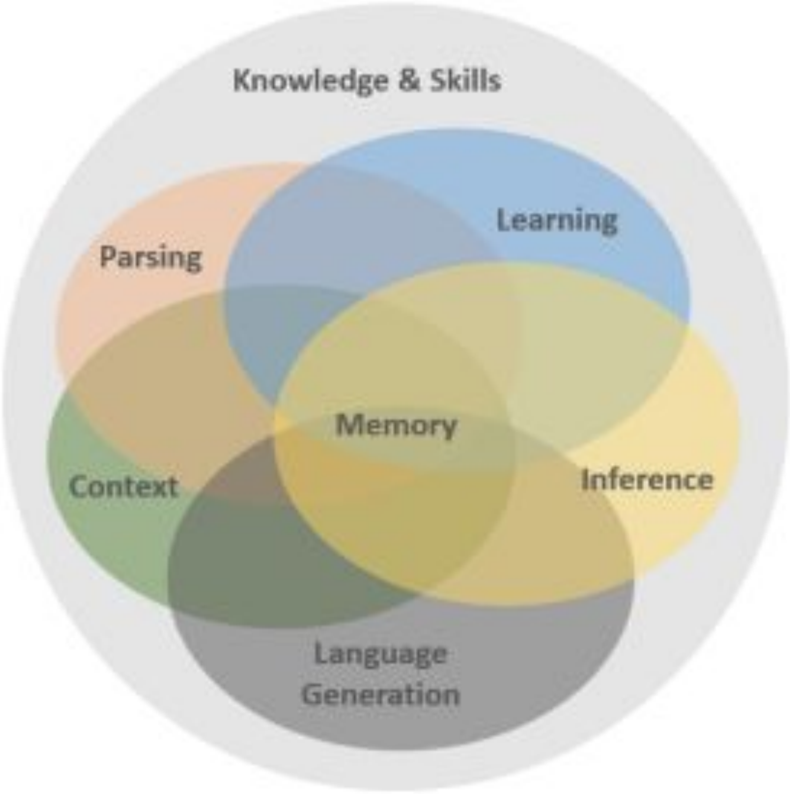


Image courtesy of "[Concepts is All You Need: A More Direct Path to AGI]" by Voss and Jovanovic

CAs: OpenCog 🤖

- An old schematic from Engineering General Intelligence, not that different from the CMC, yet also not that similar:
  - Modules for emotions, motives, language, meta-cognition, etc.

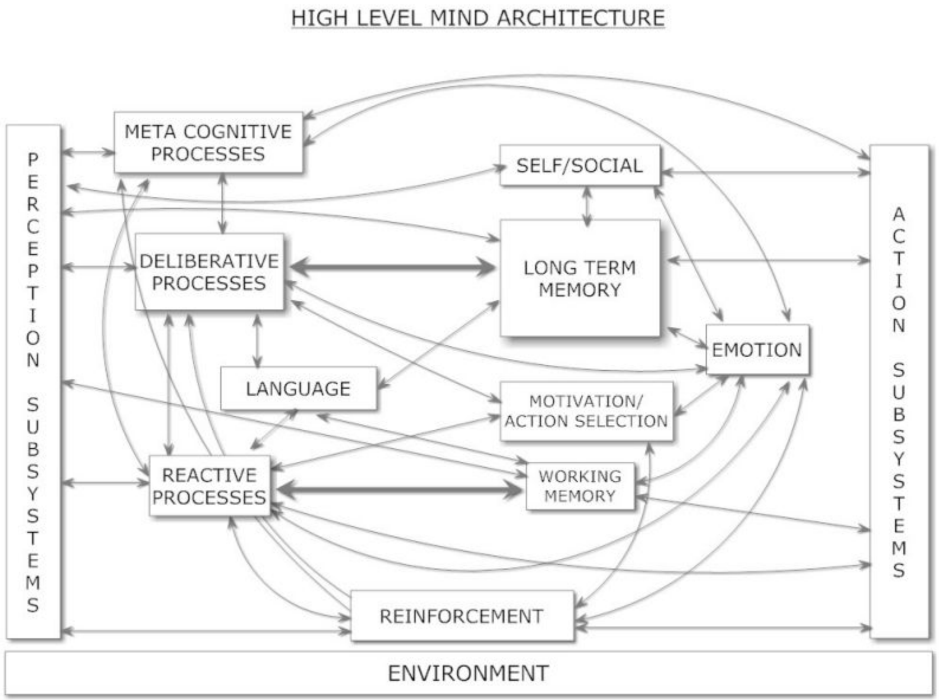
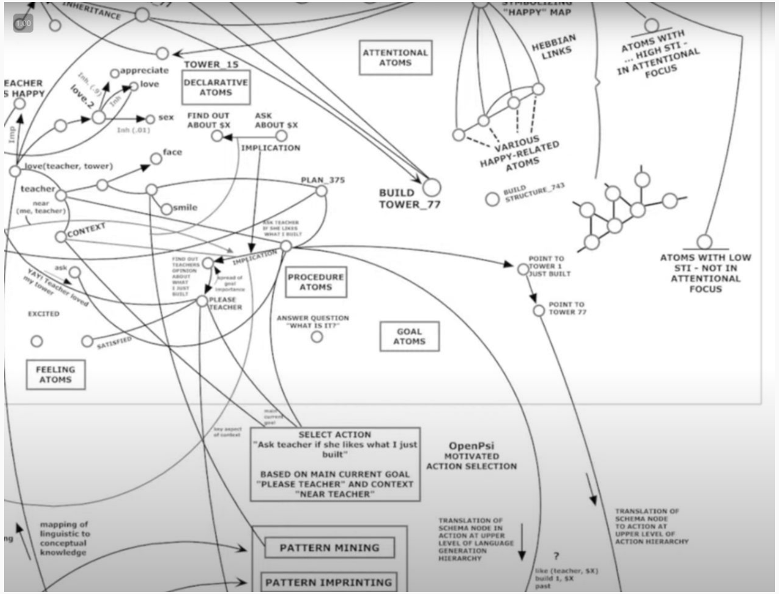


Figure 6: Illustration from Engineering General Intelligence [GPG13b] depicting the key high level components needed to support CogPrime or other similar human-like cognitive architectures. This is very similar to the basic outline of the Standard Model of Mind, basically filling in a modest amount of additional detail to Figure 16.

Cognitive Architectures: OpenCog Hyperon

- How granular should we be in including every structural component and their interrelations?
- If priors are needed, *Embodied Cognition Prior*.
- Do ATPs have a *Platonic Cognition Prior?*



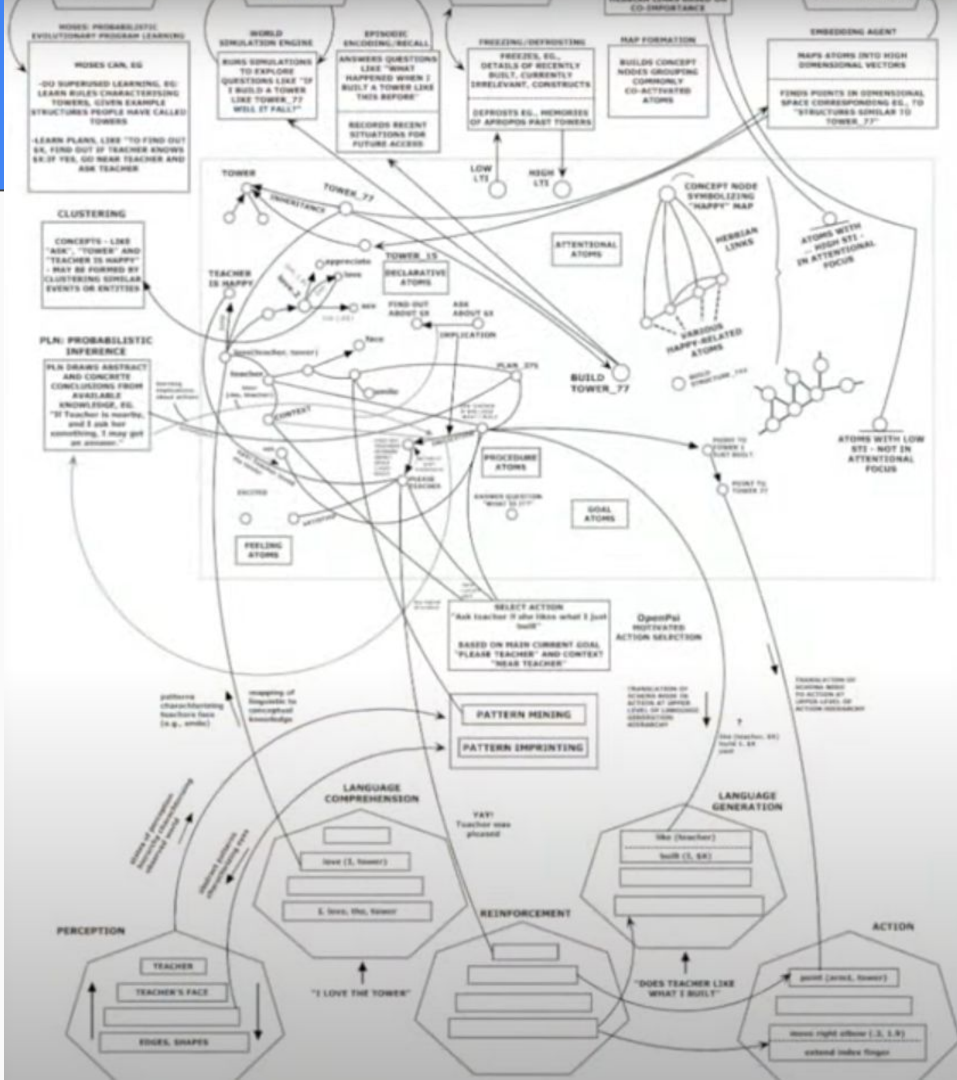
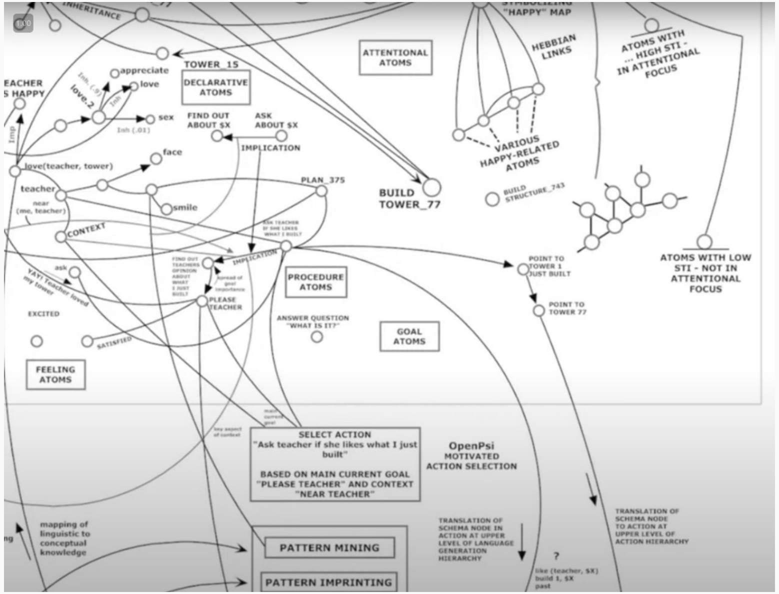


Image from Ben Goertzel's AGI-24 Keynote, "[Development of Hyperon-Based Minds: PRIMUS, Neoterics, Mind Children](#)".

Cognitive Architectures: OpenCog Hyperon

- How granular should we be in including every structural component and their interrelations?



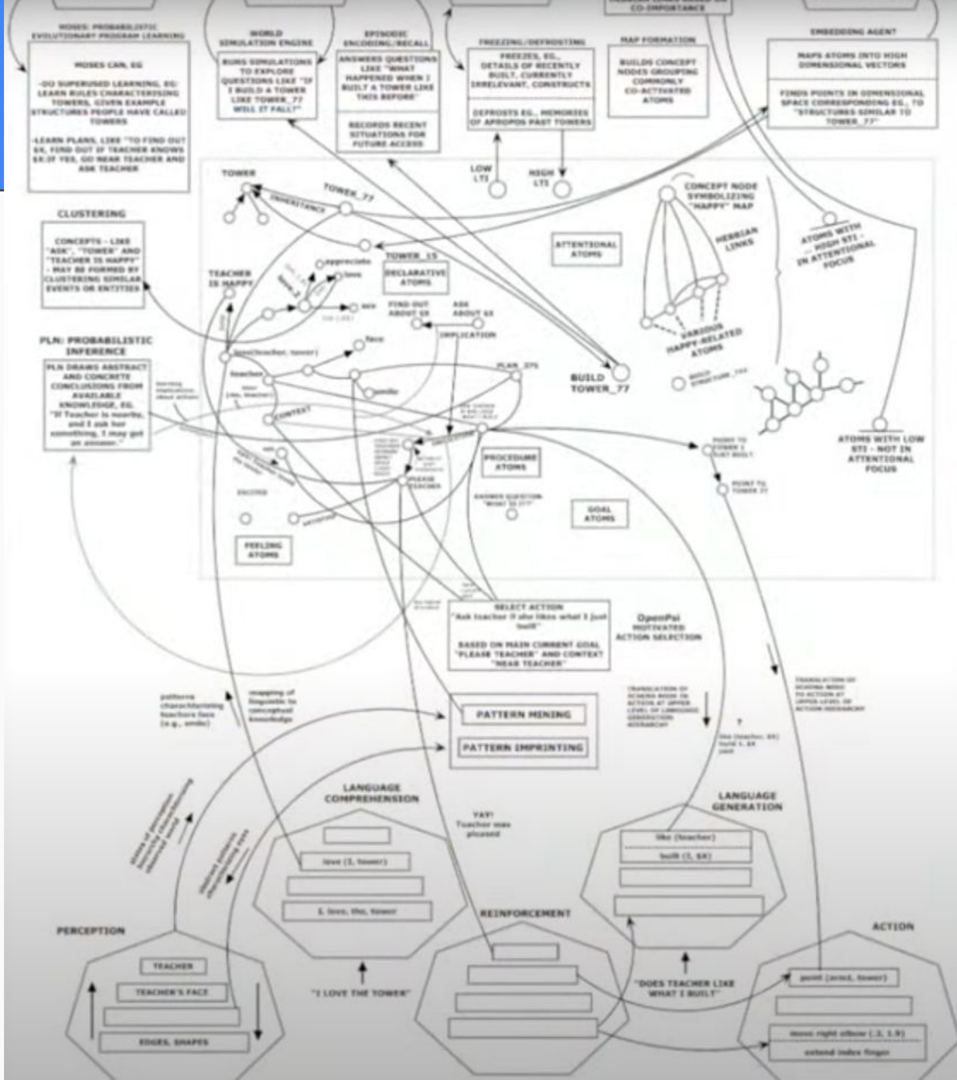


Image from Ben Goertzel's AGI-24 Keynote, "Development of Hyperon-Based Minds: PRIMUS, Neoterics, Mind Children".

Cognitive Architectures: RAISE

- What do LLMs change?

- Incorporating ‘scratchpads’ → ‘working memory’.

- LTM via database access is hidden in the ‘Tool Pool’.

- One winds up with CAs and components to synergistically integrate.

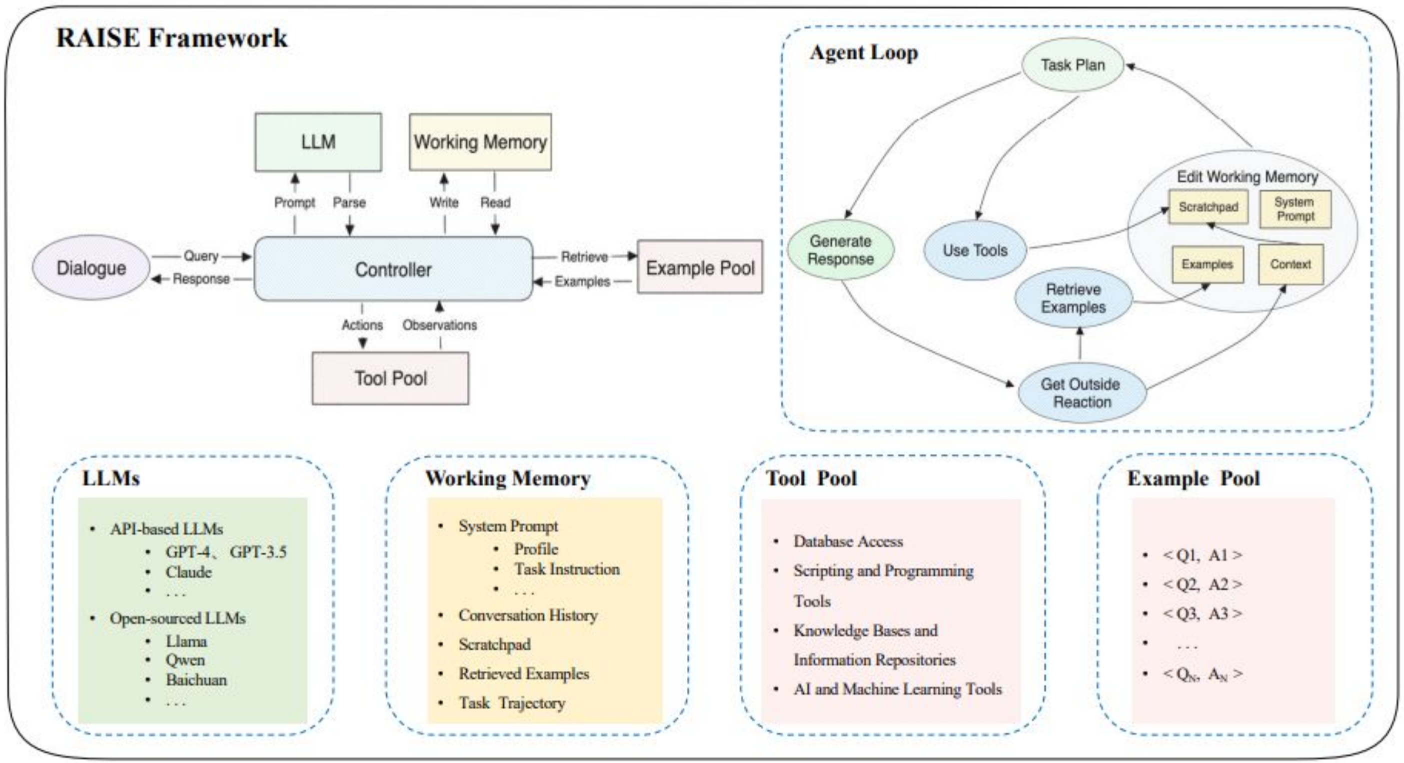
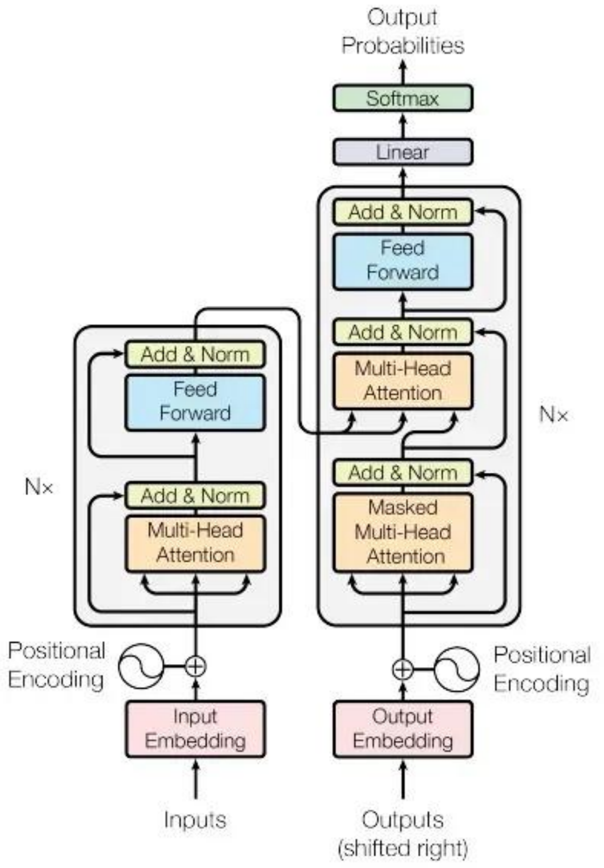


Image courtesy of “From LLM to Conversational Agent: A Memory Enhanced Architecture with Fine-Tuning of Large Language Models”.

Cognitive Architectures: Transformers?

- What is the architecture of the transformer?
- Input/Output ✔️
- Working memory via shifting context window ✔️
- Declarative memory ✖️
- Is the memory *episodic*, *procedural*, or *other*?



Cognitive Architectures: SotA of AGI

- I attended the Artificial General Intelligence conference in August.
- I witnessed surprising agreement in favor of *“hybrid neuro-symbolic approaches to AGI”*.

- LLMs are clearly amazing, yet fail at precise, reliable reasoning.
  - Chain of thought, scratchpads, and other *minimum viable reasoning/architecture* hacks don’t cut it.
  - Training is still prohibitively expensive, and online learning (vs in-context learning) doesn’t work yet.

Cognitive Architectures: SotA of AGI

- I attended the Artificial General Intelligence conference in August.
- I witnessed surprising agreement in favor of *“hybrid neuro-symbolic approaches to AGI”*.
- LLMs are clearly amazing, yet fail at precise, reliable reasoning.
    - Chain of thought, scratchpads, and other *minimum viable reasoning/architecture* hacks don’t cut it.
    - Training is still prohibitively expensive, and online learning (vs in-context learning) doesn’t work yet.

- Weak consensus on *integrating all components of an AI agent*:
    - While components may be distinct, transfer learning among them should take place.

Cognitive Architectures: SotA of AGI

- I attended the Artificial General Intelligence conference in August.
- I witnessed surprising agreement in favor of *“hybrid neuro-symbolic approaches to AGI”*.
- LLMs are clearly amazing, yet fail at precise, reliable reasoning.
    - Chain of thought, scratchpads, and other *minimum viable reasoning/architecture* hacks don’t cut it.
    - Training is still prohibitively expensive, and online learning (vs in-context learning) doesn’t work yet.
- Weak consensus on *integrating all components of an AI agent*:
    - While components may be distinct, transfer learning among them should take place.

- Thórisson and Talevi’s *theory of foundational meaning* links meaning to *autonomy*.
    - Hypothesis: in the environmental architecture of an AI system, any module relying on a human may be fundamentally limiting.
    - Without neural implants, *human-in-the-loop* components imply *low integration*.

Cognitive Architectures: SotA of AGI

- I attended the Artificial General Intelligence conference in August.
- I witnessed surprising agreement in favor of *“hybrid neuro-symbolic approaches to AGI”*.
- LLMs are clearly amazing, yet fail at precise, reliable reasoning.
    - Chain of thought, scratchpads, and other *minimum viable reasoning/architecture* hacks don’t cut it.
    - Training is still prohibitively expensive, and online learning (vs in-context learning) doesn’t work yet.
- Weak consensus on *integrating all components of an AI agent*:
    - While components may be distinct, transfer learning among them should take place.

- Thórisson and Talevi’s *theory of foundational meaning* links meaning to *autonomy*.
    - <u>Hypothesis</u>: in the environmental architecture of an AI system, any module relying on a human may be fundamentally limiting.
    - Without neural implants, *human-in-the-loop* components imply *low integration*.

- The time for AITP to shine may be now! 🔥🤖🚀

Theorem Proving Cognitive Architectures

Cognitive Architecture: ATPs and the Cosmos

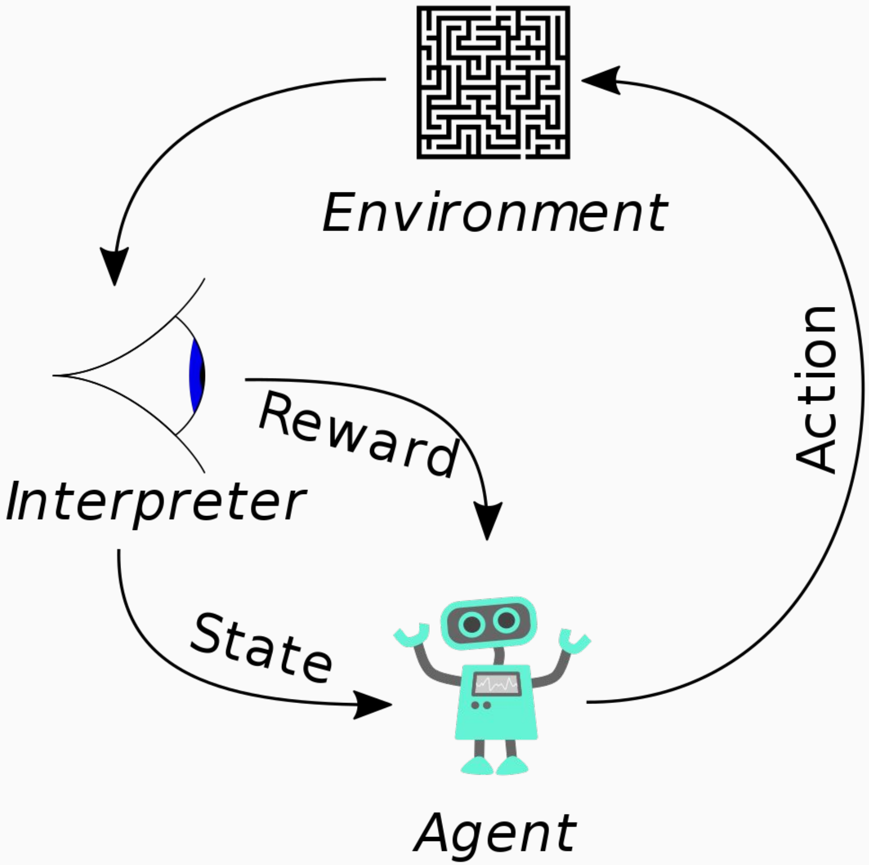
- Where do ATPs fit into the cosmos, the environment?

  - Humans or ITP systems invoke them.
    - Reward is provided to the ML components for proofs.

  - And the ITP systems are largely static KBs, growing as humans interact with them.

  - The mathematics in ITP systems often represent meaningful aspects of reality (to humans) or properties of computer software/hardware.

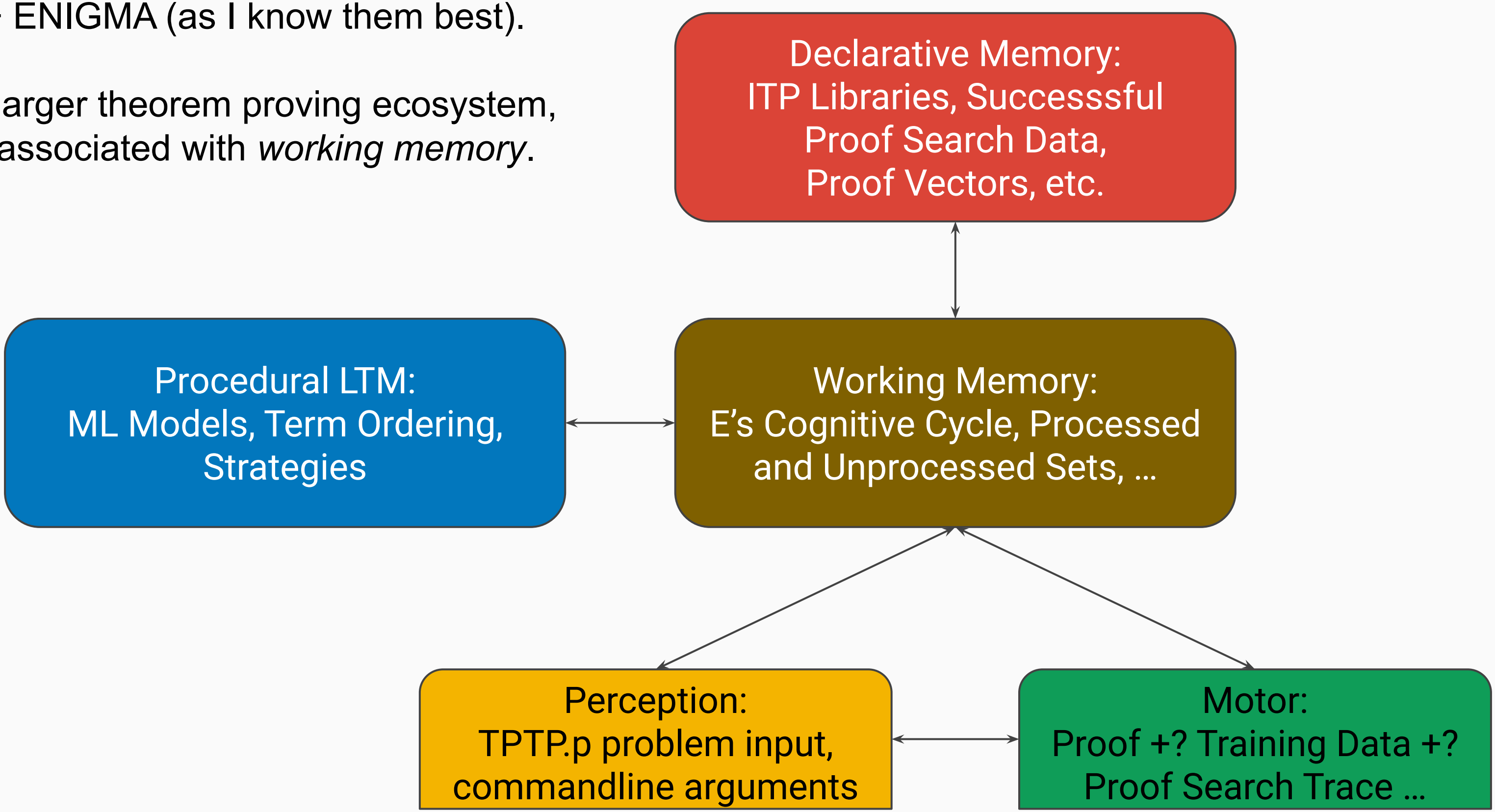
→ Humans are very much in the loop.



Cognitive Architecture: ATPs and the CMC

- I'll mainly analyze E + ENIGMA (as I know them best).

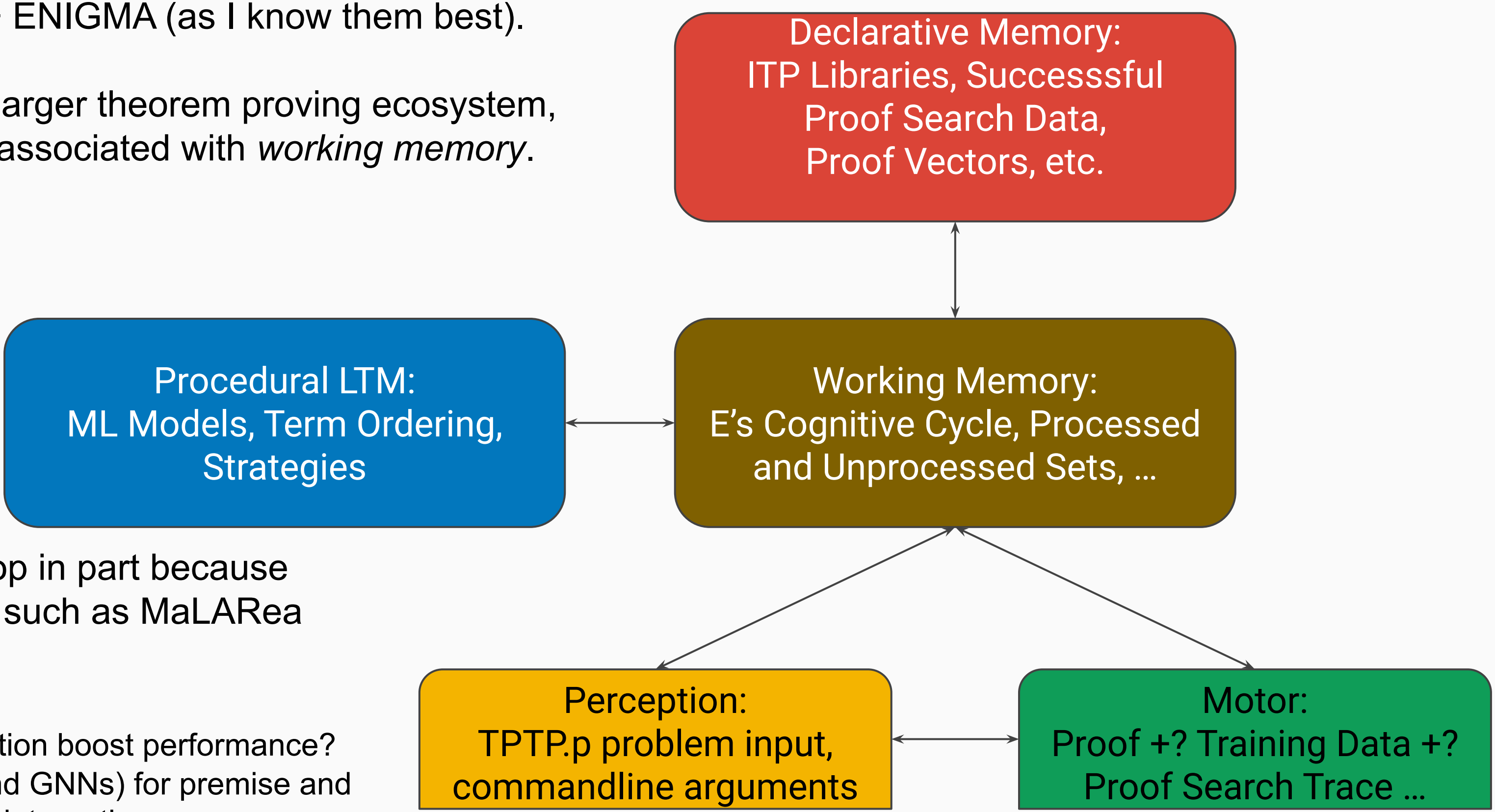
- As a component of a larger theorem proving ecosystem, the ATP (E) is mostly associated with *working memory*.



Cognitive Architecture: ATPs and the CMC

- I'll mainly analyze E + ENIGMA (as I know them best).

- As a component of a larger theorem proving ecosystem, the ATP (E) is mostly associated with *working memory*.



- Humans are in the loop in part because autonomous systems such as MaLARea tend to plateau.

- Would successful integration boost performance?
  - The same ML (GBDT and GNNs) for premise and clause selection hints at integration….

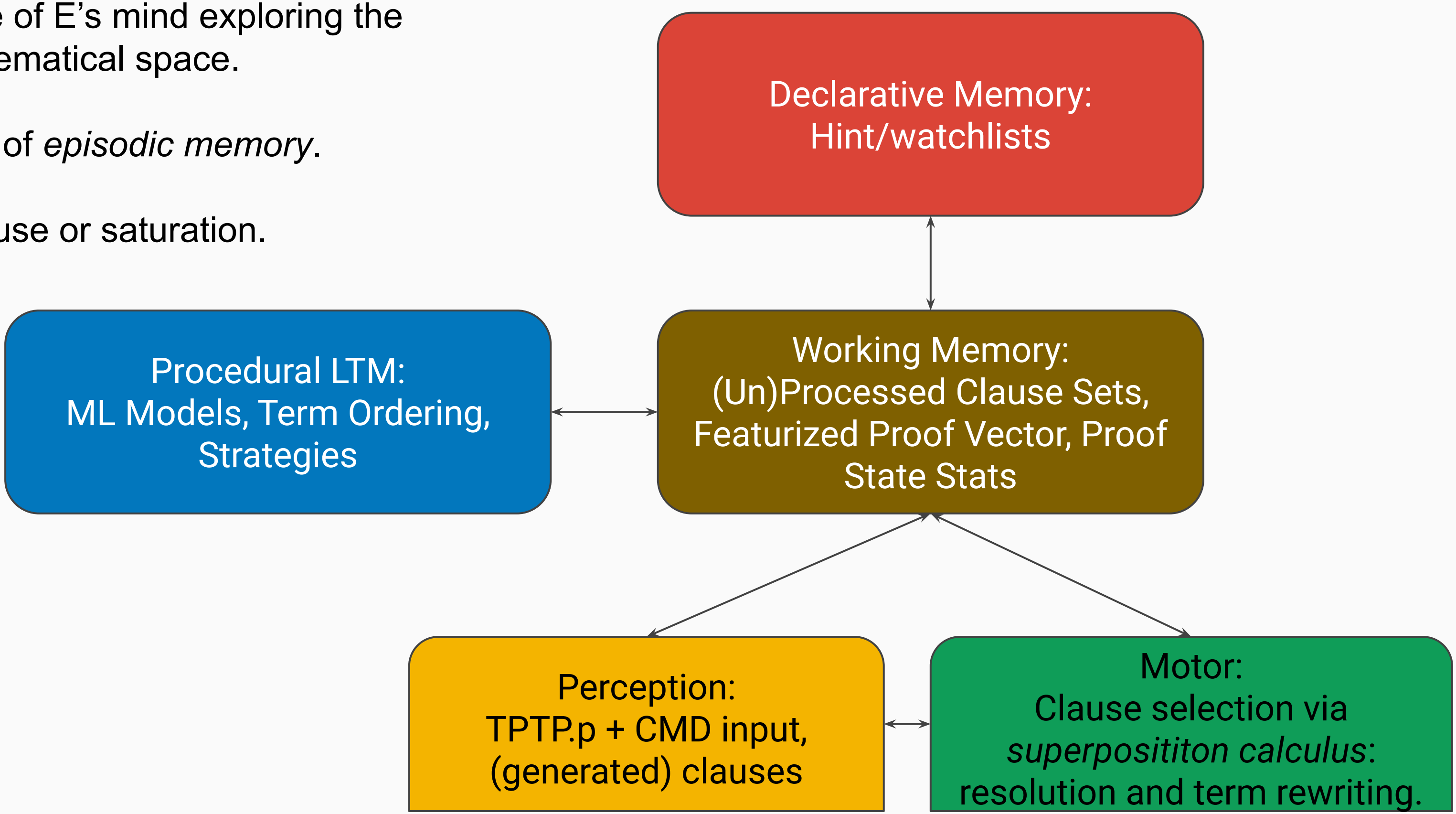
Cognitive Architecture: ATPs and the CMC

- Now let's zoom inside of E's mind exploring the enivronment of a mathematical space.

- Watchlists are a form of *episodic memory*.

- Two goals: empty clause or saturation.

- …



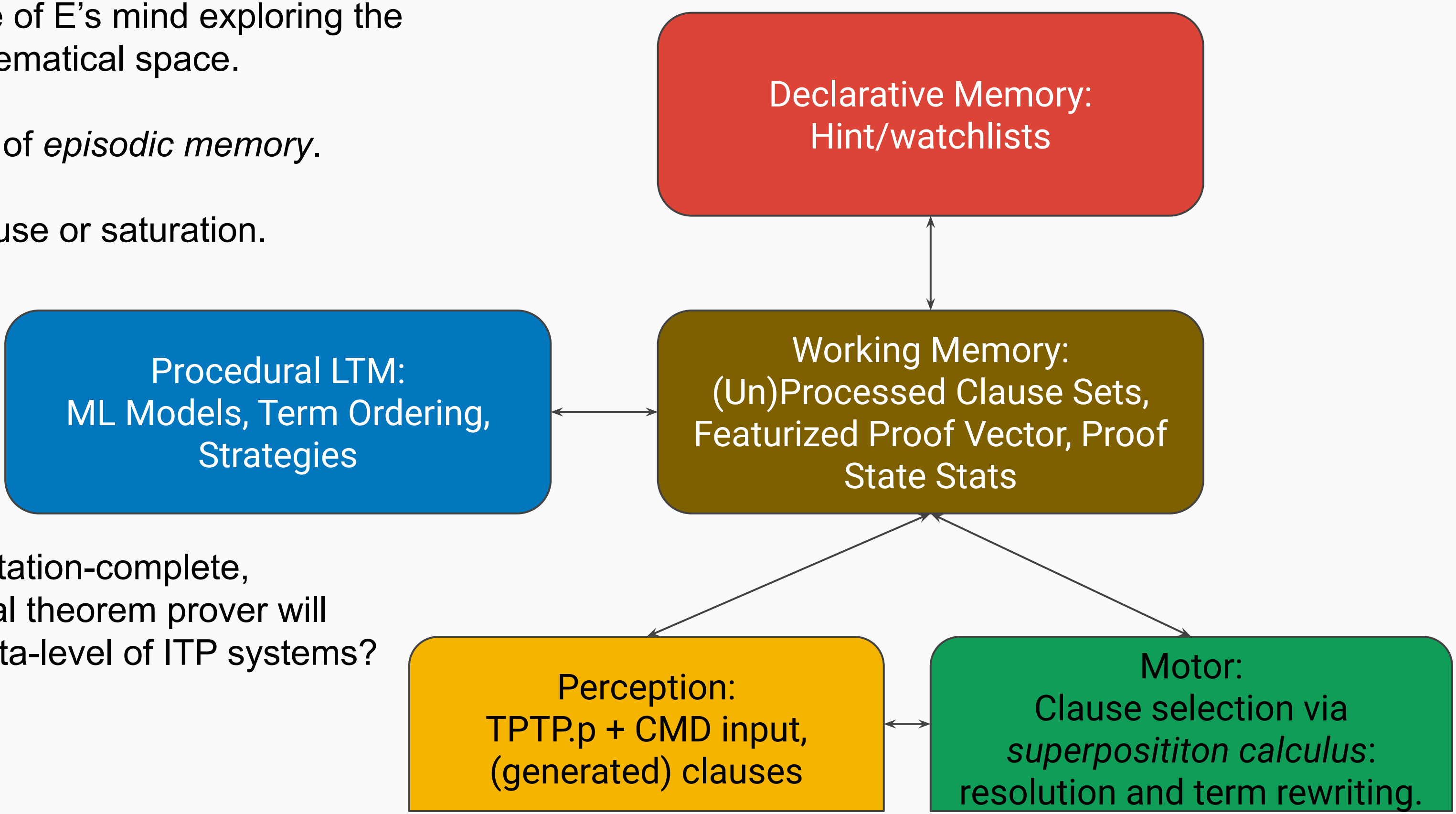
Cognitive Architecture: ATPs and the CMC

- Now let's zoom inside of E's mind exploring the enivronment of a mathematical space.

- Watchlists are a form of *episodic memory*.

- Two goals: empty clause or saturation.

- While technically refutation-complete, probably a fully general theorem prover will need to live on the meta-level of ITP systems?



What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context* – related to Thórisson and Talevi's theory of *foundation meaning*:

$$\mathcal{M}^{\mathcal{A}}_{now}(\mathcal{I}) = Pr_t(\mathcal{I}, \sigma_t, \mathcal{K}) \rightarrow G_{t'} \rightarrow Pl_{t''} \quad (1)$$

- In words, *meaning* is a process by which an *A*gent updates its predictions about the world (based on info datum *I*, situation *σ*, and knowledge *K*), its goals *G*, and its plans *Pl*.
  - If neither preditions, goals, nor plans change, then *I* is essentially meaningless to *A*.
  - E.g., a subsumed clause is *deleted* and therefore (nearly) *meaningless to E*.

What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context* – related to Thórisson and Talevi's theory of *foundation meaning*:

$$\mathcal{M}^{\mathcal{A}}_{now}(\mathcal{I}) = Pr_t(\mathcal{I}, \sigma_t, \mathcal{K}) \rightarrow G_{t'} \rightarrow Pl_{t''} \qquad (1)$$

- In words, *meaning* is a process by which an *A*gent updates its predictions about the world (based on info datum *I*, situation *σ*, and knowledge *K*), its goals *G*, and its plans *Pl*.
  - If neither preditions, goals, nor plans change, then *I* is essentially meaningless to *A*.
  - E.g., a subsumed clause is *deleted* and therefore (nearly) *meaningless to E*.

- Meaning is largely via the ordering of selected clauses*.

- There may be more transparent meaning in tactic-based AI for ITP systems [E.g., Tactician, TacticToe, etc.]:
  - Sub-goals can change, which could affect the appropriate plans.

What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context*, and *meaning*.

- Why is the ATP/ITP trying to solve *this particular conjecture right now?*

- Consider the Large Theory Batch division of CASC (The World Championship for Automated Theorem Proving):
  - The ATP has a fixed time to solve as many problems of the batch as possible.
  - Thus there is some *autonomy* to the choice of which problems to work on, when, and for how long.
  - Could solving some problems help with others, providing *meaning*?
  - Mapping out the relations among problems (and background knowledge) constitutes developing a *worldview*.

What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context*, and *meaning*.

- Why is the ATP/ITP trying to solve *this particular conjecture right now?*

- Consider the Large Theory Batch division of CASC (The World Championship for Automated Theorem Proving):
  - The ATP has a fixed time to solve as many problems of the batch as possible.
  - Thus there is some *autonomy* to the choice of which problems to work on, when, and for how long.
  - Could solving some problems help with others, providing *meaning*?
  - Mapping out the relations among problems (and background knowledge) constitutes developing a *worldview*.

- Consider *"the real world"*:
  - Why do we care about differential equations, geometry, calculus, programming language theory,...?
  - Theories *semantically* connect to real domains we care about:
    - We tend to have *models* motivating, and clarified by, the theories.

What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context*, and *meaning*.

- Why is the ATP/ITP trying to solve *this particular conjecture right now?*

- Consider *"the real world"*:
  - Why do we care about differential equations, geometry, calculus, programming language theory,...?
  - Theories *semantically* connect to real domains we care about:
    - We tend to have *models* motivating, and clarified by, the theories.

- Consider *lemmata*:
  - Salient theorems tell one about the mathematical space.
  - Inferential compressibility as an indicator of interest?
    - If L improves the proof (search time) of C, this suggests both L and C may be of interest.

What might AITPs be lacking for AGITP?

One hypothesis: *autonomy, worldviews, context*, and *meaning*.

- Why is the ATP/ITP trying to solve *this particular conjecture right now?*

- Consider *"the real world"*:
  - Why do we care about differential equations, geometry, calculus, programming language theory,...?
  - Theories *semantically* connect to real domains we care about:
    - We tend to have *models* motivating, and clarified by, the theories.

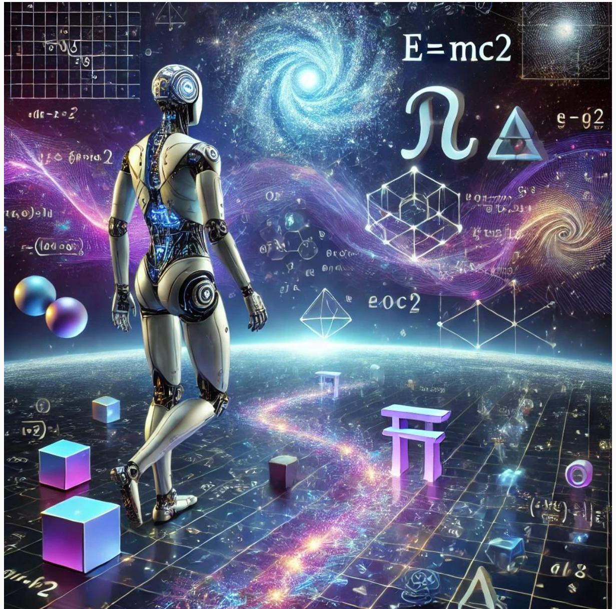
- Consider *lemmata*:
  - Salient theorems tell one about the mathematical space.
  - Inferential compressibility as an indicator of interest?
    - If L improves the proof (search time) of C, this suggests both L and C may be of interest.

→ An AGITP system should operate in the context of the available mathematical universes, autonomously choosing which domains and conjectures to explore, which will foster greater meaning for some formulas.

On the quest for *conjecturing*

My conjecture:

- The goal will remain elusive until the AI systems work with the *contexts* that provide *meaning* to the conjectures.
- Theory exploration and quality recognition should be motivated.



What might AITPs be lacking for AGITP?

Another hypothesis taking *autonomy* to the next level: the AGITP needs to be *self-organizing* and *metalearning*.

What might AITPs be lacking for AGITP?

Another hypothesis taking *autonomy* to the next level: the AGITP needs to be *self-organizing* and *metalearning*.

- The AI needs to *learn how to learn better*.
  - In my experiments with ENIGMA, even with *looping iterations* and *grid searches*, we humans carefully monitored the progress, tweaking the parameters….
  - When to tweak datasets, parameters, number of iterations, types of proofs to learn from, strategies?
  - How to analyze ATP/ITP codebases and pipelines to identify *choice points* suitable for AI insertion?
    - E.g., in *Parental Guidance*, adding an ML model to filter generated clauses based on their parents' features prior to clause selection seems help and integrate well with ML for clause selection 🤖🤝🤖🤝🤖.
  - In the ideal limit: can the ATP prove improvements to its proof calculus, term-trees, etc. like a Gödel machine?

What might AITPs be lacking for AGITP?

Another hypothesis taking *autonomy* to the next level: the AGITP needs to be *self-organizing* and *metalearning*.

- The AI needs to *learn how to learn better*.
  - In my experiments with ENIGMA, even with *looping iterations* and *grid searches*, we humans carefully monitored the progress, tweaking the parameters….
  - When to tweak datasets, parameters, number of iterations, types of proofs to learn from, strategies?
  - How to analyze ATP/ITP codebases and pipelines to identify *choice points* suitable for AI insertion?
    - E.g., in *Parental Guidance*, adding an ML model to filter generated clauses based on their parents' features prior to clause selection seems help and integrate well with ML for clause selection 🤖🤝🤖🤝🤖.
  - In the ideal limit: can the ATP prove improvements to its proof calculus, term-trees, etc. like a Gödel machine?

- This sounds hard. How do we get there?
  1) Incremental steps integrating AI components one at a time.
  2) Wild idea: set up the infrastructure/architecture to allow for metalearning self-organization in such a way that humans control the process as if *models* in the AGITP system so that their automation becomes smoother 😎.

AGITP and the “Real World”

- Ultimately, the AGI-leaning ATP system needs ways to deal with *every relevant aspect of its world*.

- The broader context of all formal math libraries contains a large amount of *informal mathematics* (physics, etc).
  - Discussions about which ITP system to use, how to update them, etc., are generally done in natural languages.

- Many domains important (to us humans) are described in natural language with multi-media additions.

AGITP and the “Real World”: Autoformalization

- Ultimately, the AGI-leaning ATP system needs ways to deal with *every relevant aspect of its world*.

- The broader context of all formal math libraries contains a large amount of *informal mathematics* (physics, etc).

- Many domains important (to us humans) are described in natural language with multimedia additions.

→ Autoformalization of math, natural language problem descriptions, and multimedia scenarios may be necessary

- Ambitious claim: in theory, most practical problems should be formally solvable (to the extent they’re solvable) provided the adequate formalization.

Concluding Food for Thought

- ATPs can be seen as practically usable universal AIs.
- Proto-AGI systems should be equipped to deal with and learn about (nearly) every aspect of their operation and environment.
  - Supportive integration among components, including neural and symbolic ones, appears to be crucial.
- AGI-ATPs should probably live on the level of ITPs (or the humans working with ITPs).
- Aiming for AGITPs may be the road to full-scale conjecturing success.
- Successful incremental integration of AI into/across components should lead to publishable results 😎.
- Claim: thinking about AITP research in the context of AGI may prove bountiful.