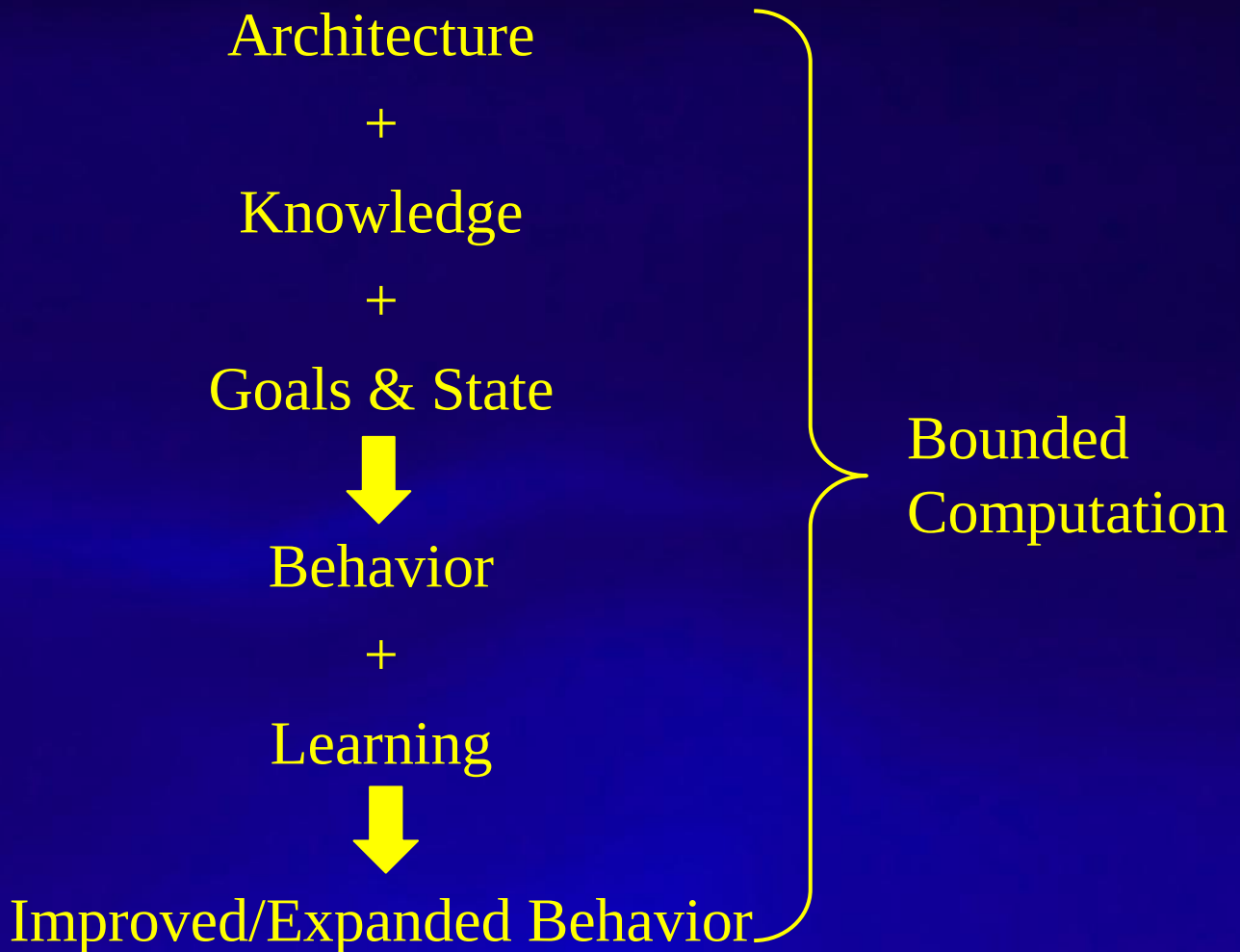
Soar Nuggets
(and Coal)

John E. Laird

University of Michigan

Soar 27th Workshop

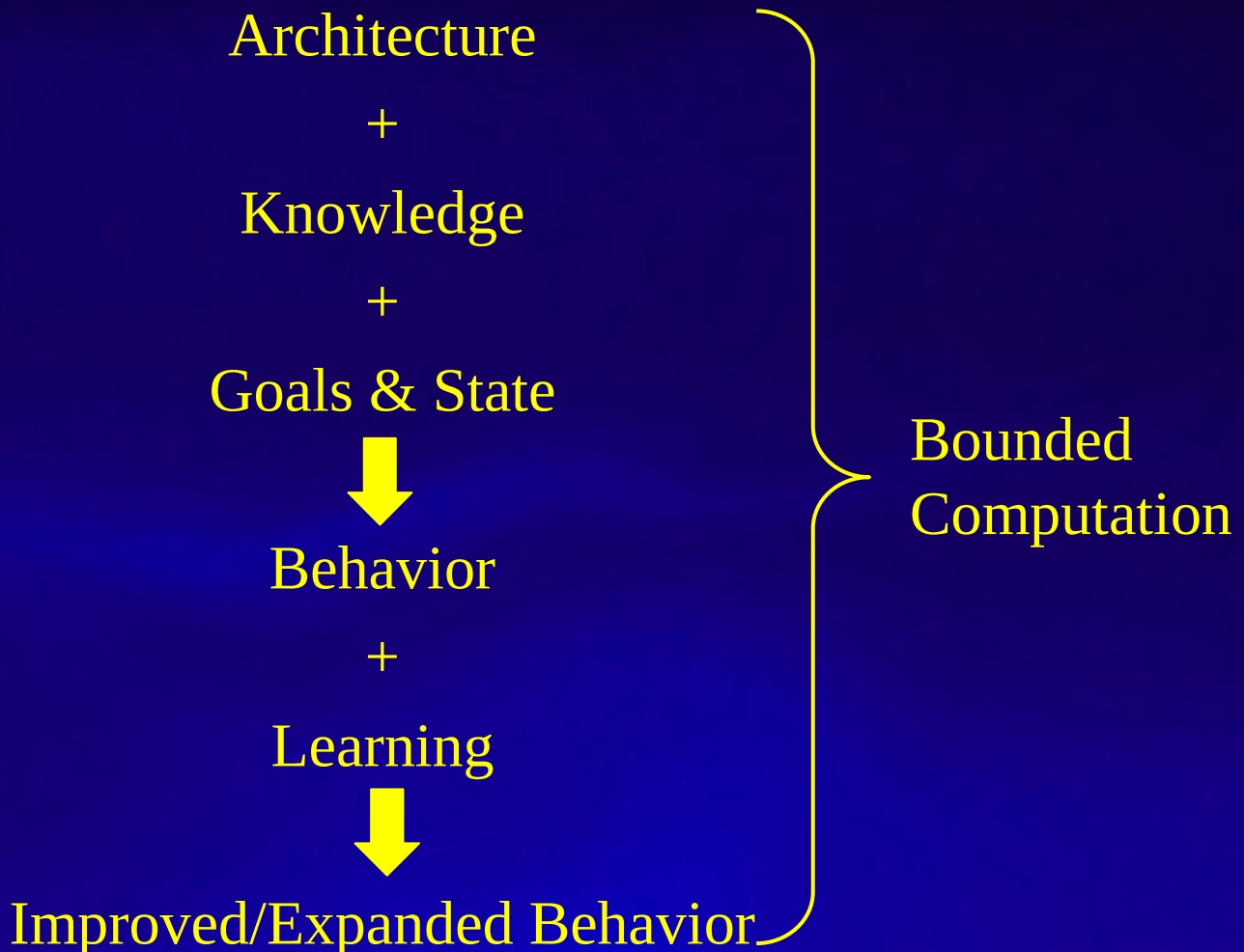
Basic Model of Cognitive Architecture



Fundamental Goals of Cognitive Architecture

1. Bring to bear all available knowledge
2. Minimal knowledge is required to produce behavior
3. Behavior is consistent with available knowledge and state
4. Learning is sufficient for all task knowledge
5. Computation is bounded

Basic Model of Cognitive Architecture

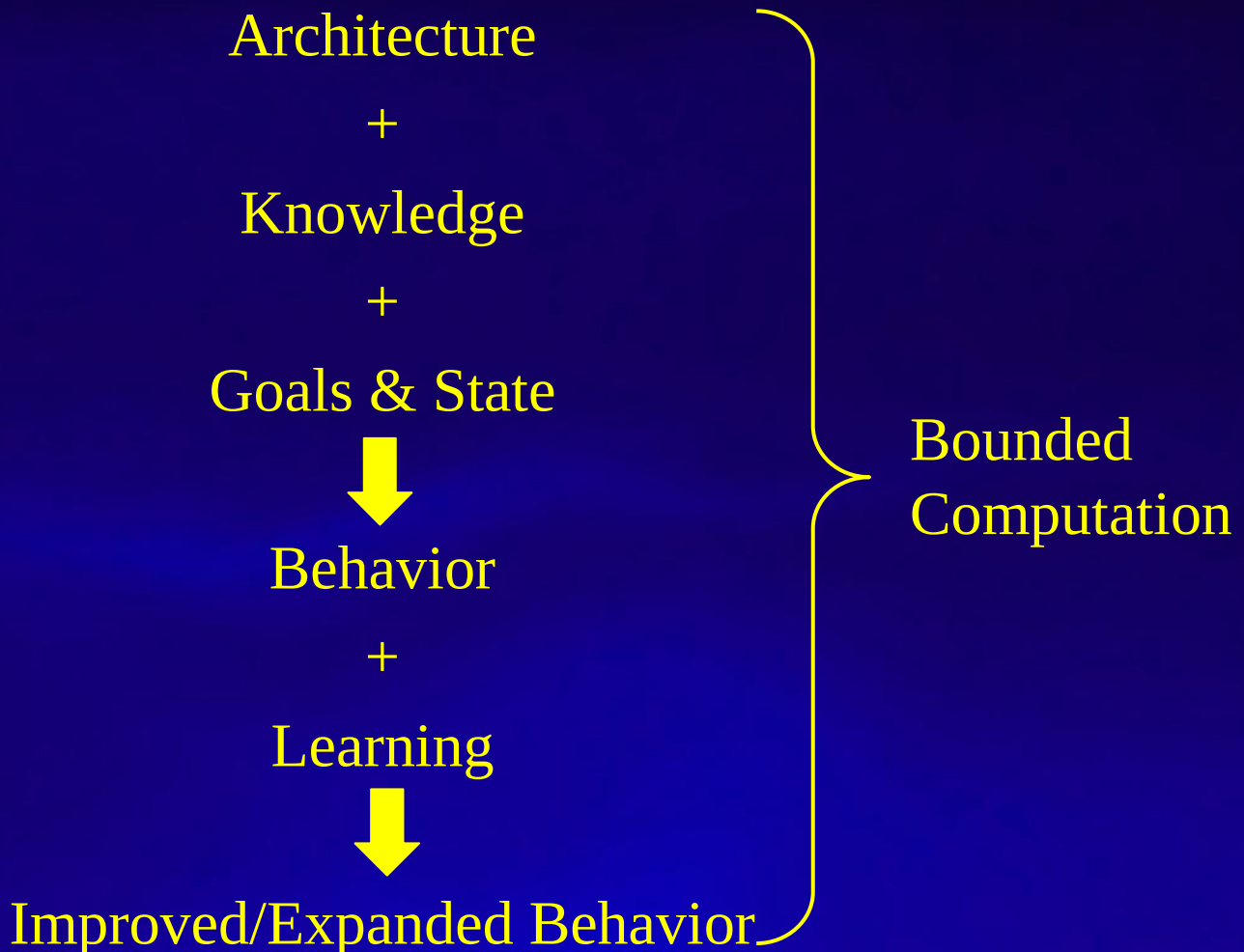


Bring to Bear All Available Knowledge

+ Fires all matching rules in parallel
  - No filtering or masking via conflict resolution
+ Keeps firing until quiescence
+ Combines preferences in decision making

– Accesses subgoal knowledge only on impasse
  - Biased toward directly available knowledge
– Accesses semantic & episodic only deliberately

Basic Model of Cognitive Architecture



Minimal Knowledge is Required to Produce Behavior

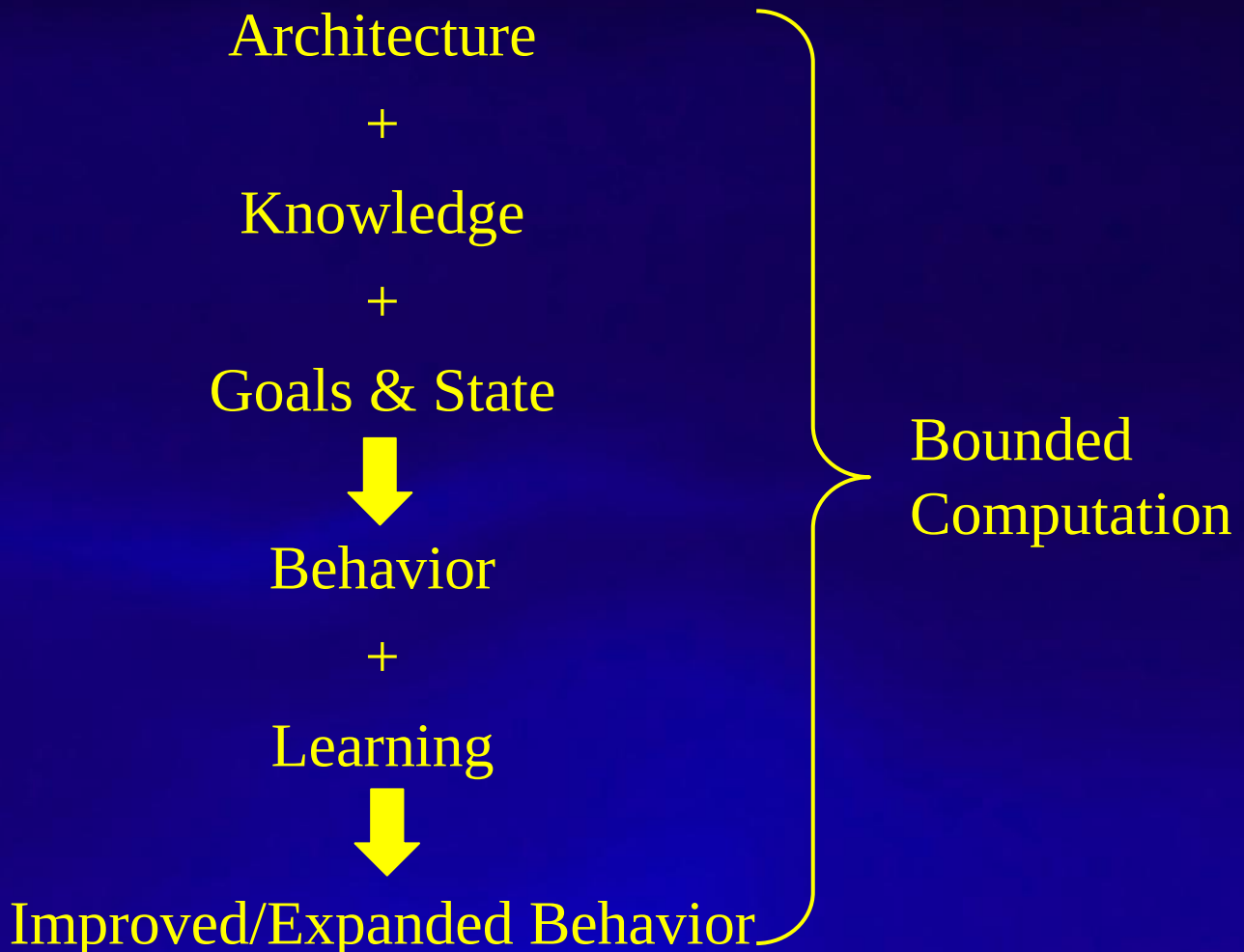
+ Can achieve new behavior with only operator proposal
  - Does not require operator maintenance or termination
+ Rules combine dynamically in sequence
  - Eliminates redundancy in duplicate conditions/actions
+ Fine grain units of knowledge: rules
  - No required structures that don't impact behavior
+ Can reuse problem spaces across subgoals (hierarchy)
+ Supports general, unconstrained interpretation
  - Easily implemented in impasse-driven subgoals

– Programmers can defeat Soar's capabilities

Alternative Operator Knowledge Categories

	Selection	Maintenance	Termination
Propose only (Soar)	P	P	-P
Propose & Terminate (Soar 7)	P		T
Propose & Maintain	P	M	-M & -P
Propose, Maintain, & Terminate	P	M	-M & -P/T

Basic Model of Cognitive Architecture

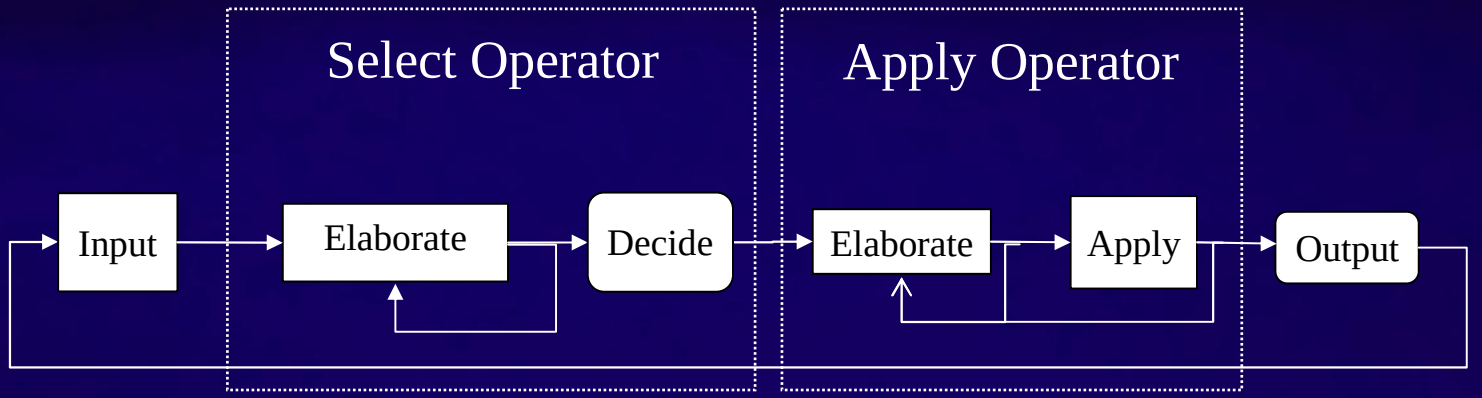


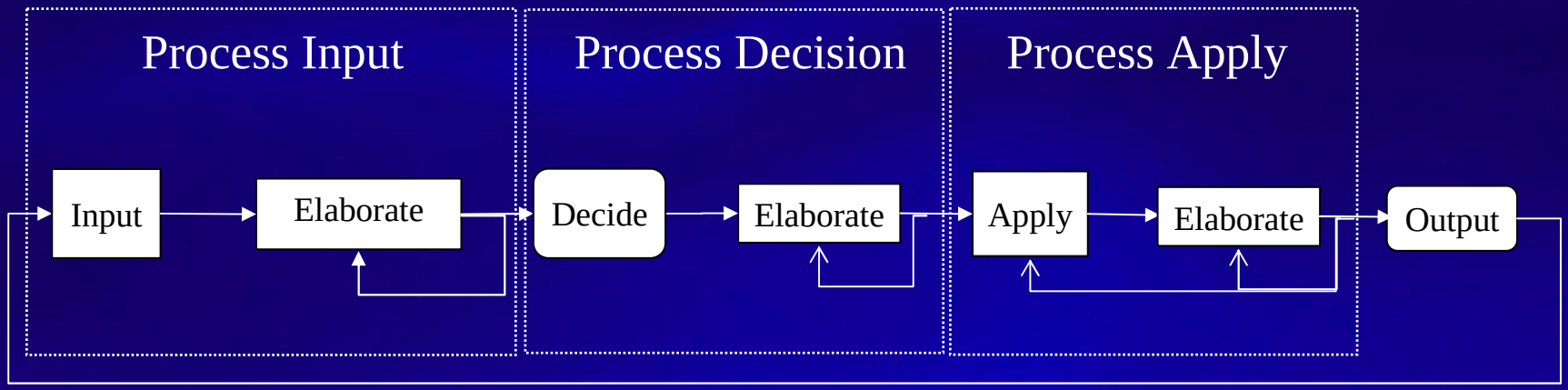
Behavior is Consistent with Available Knowledge and State

+ Automatically retracts knowledge that is no longer relevant
  - Ensures consistent processing within state
+ Only makes decisions when there is sufficient knowledge
  - Ensures consistent processing within state
+ All non-monotonic actions occur after quiescence
  - Ensures consistent processing within state
+ Substate processing from oldest to newest
  - Ensures consistent processing across substates
+ Goal dependency set (GDS) processing
  - Ensures consistent representations across substates
+ Automatic classification of justifications
  - Ensures consistent processing after chunking

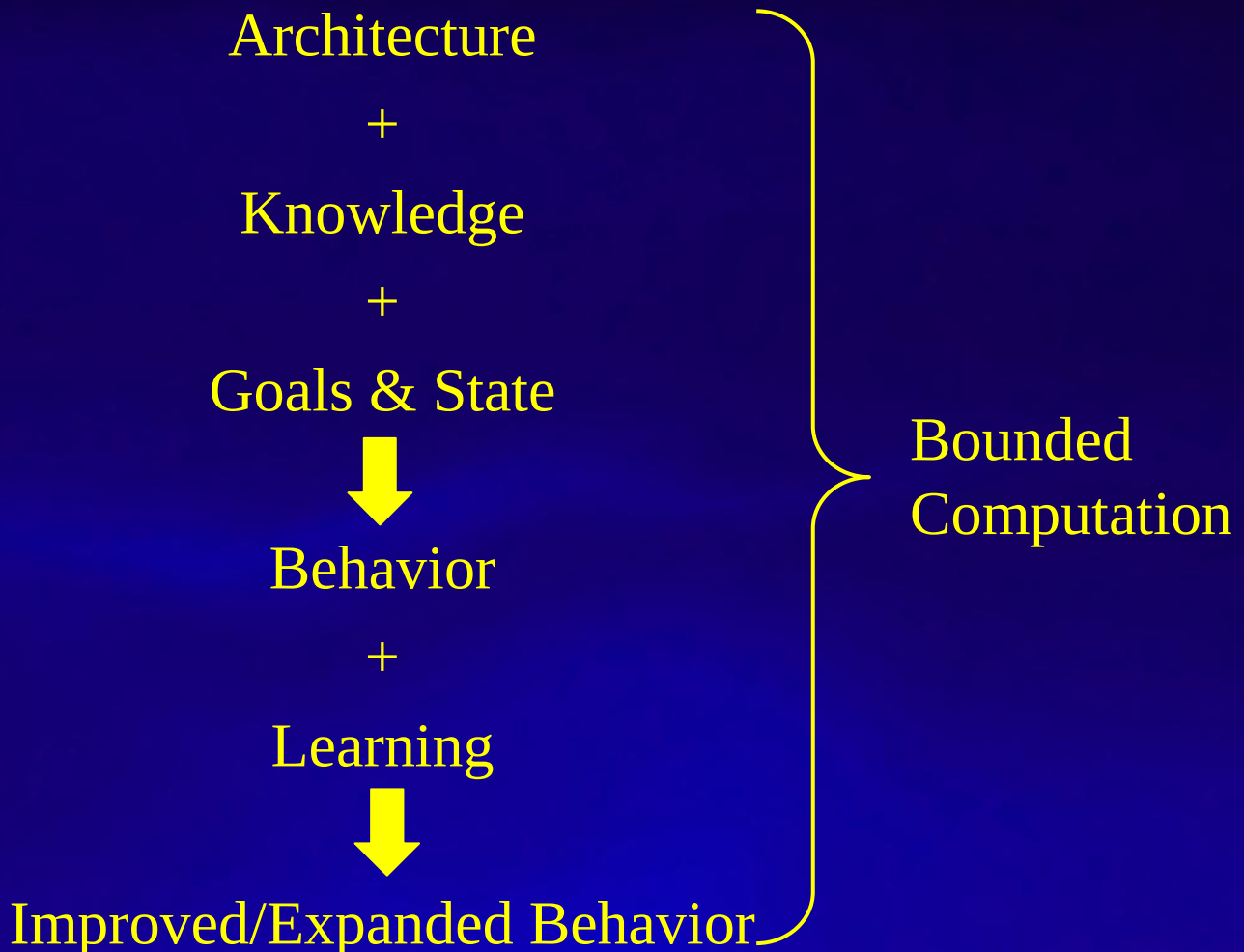
– GDS adds complexity

Alternative Views of Processing Cycle





Basic Model of Cognitive Architecture

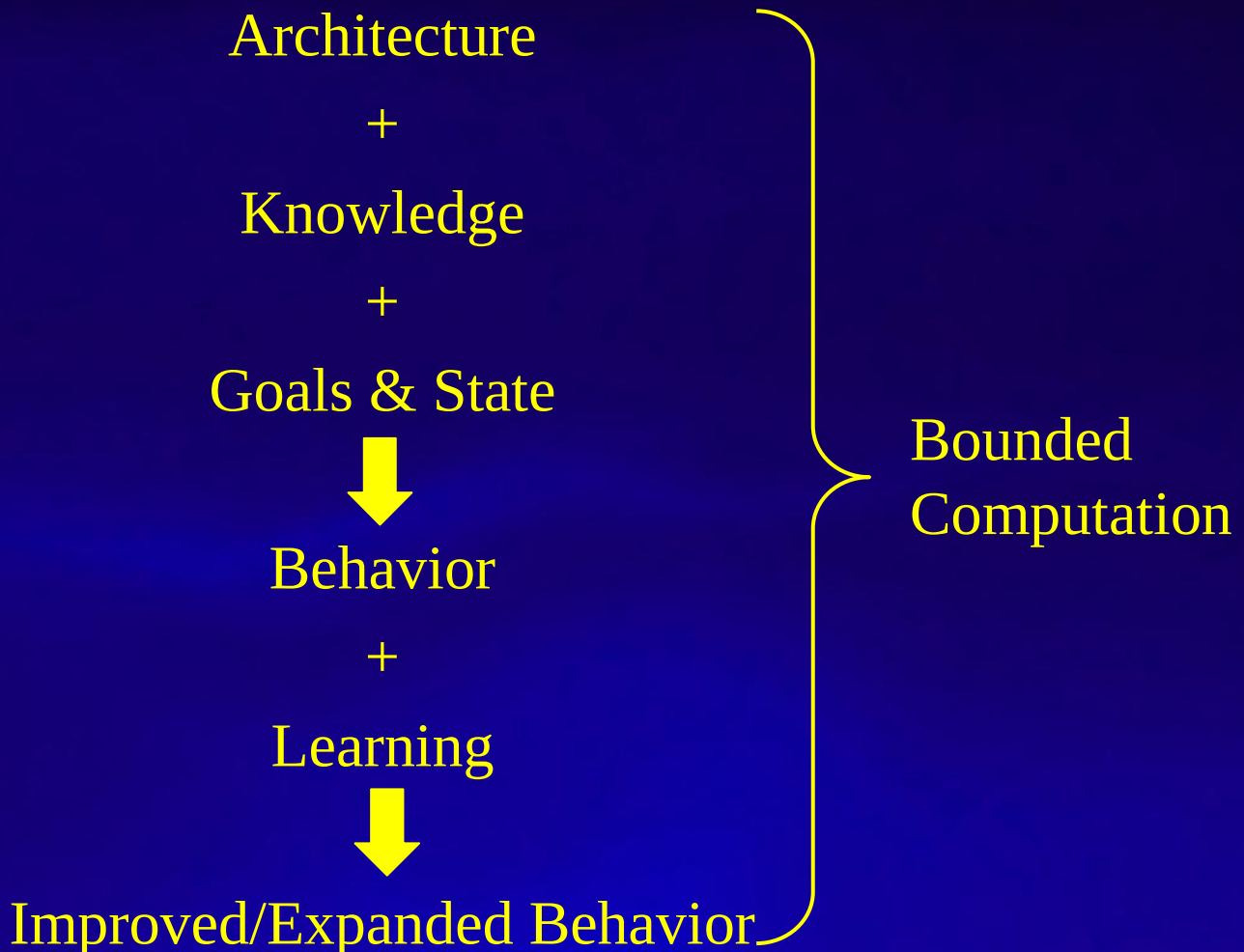


Learning is Sufficient for All Task Knowledge

+ Chunking can learn all types of procedural knowledge
  - Substates can generate all types of knowledge
  - Worst case can use generate and test
+ Episodic memory and semantic memory can learn declarative knowledge

– Not clear what initial knowledge is required
– Soar has no automatic means to generate new symbolic constants

Basic Model of Cognitive Architecture



Computation is Bounded

+ Production match scales with more rules
+ Soar processes only changes
  - Input, production match, production firing/retraction, preference creation, output
+ Bounded elaboration

– Expensive chunks
– Retrieval from episodic memory

Summary

- Soar does well across all of these dimensions
- There is an inevitable tradeoff between expressibility of knowledge and computational efficiency
  - Retrieval biased toward directly available knowledge
  - Unbounded elaborations
  - O-support in substates & GDS
  - Expensive chunks