

# AGI and Consciousness

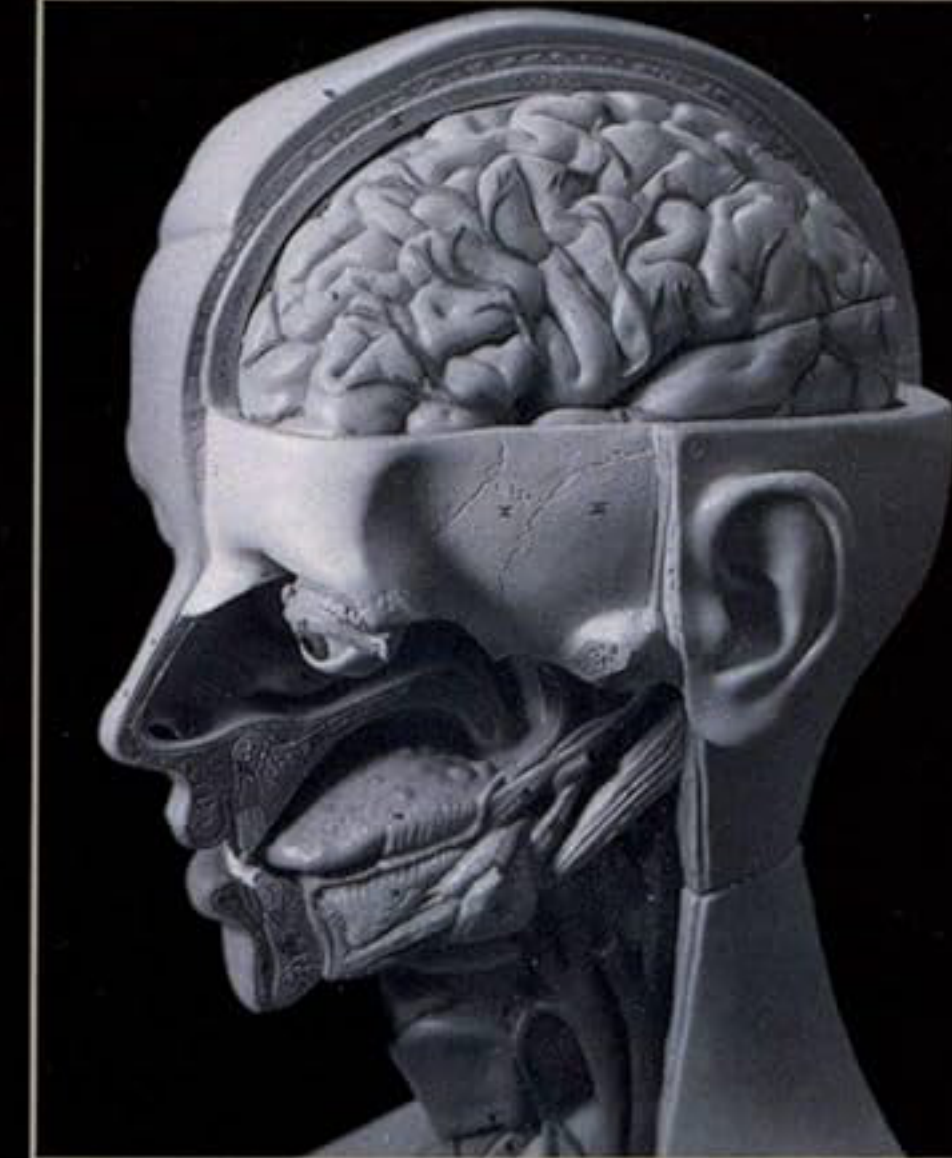
Sridhar Mahadevan, Adobe Research and U.Mass, Amherst

# What is consciousness?

Studied in every possible  
field: philosophy, psychology,  
linguistics, neuroscience,  
religion, sociology etc.

IN THE  
THEATER OF  
CONSCIOUSNESS

*The Workspace  
of the Mind*

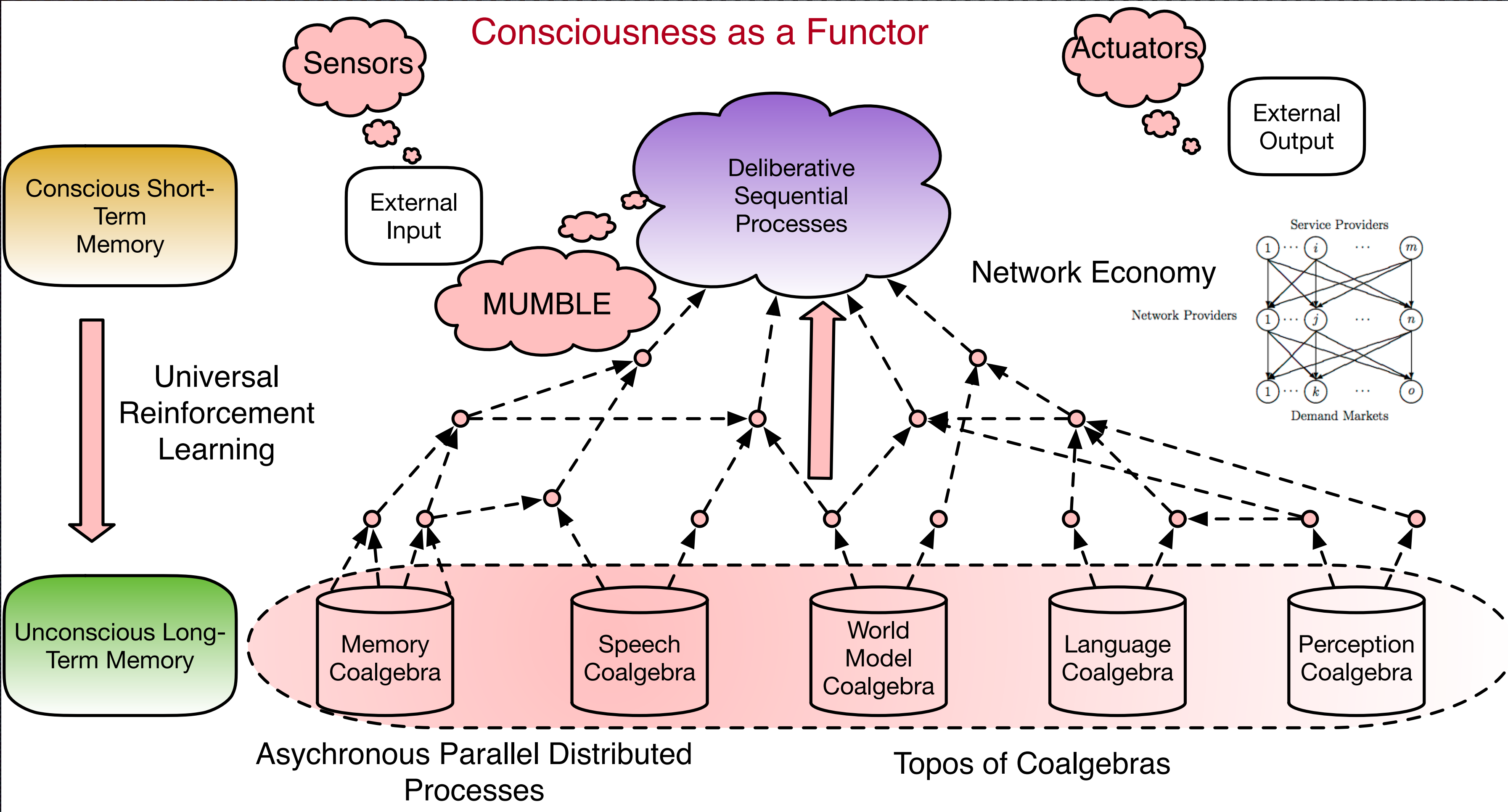


BERNARD J. BAARS

# Engineering Perspective

- An AGI system is comprised of many thousands of processes
  - Perception, language, memory, action, learning, reasoning
- An AGI agent can only act sequentially
  - Sequential action imposes a strong bottleneck on AGI systems
- Consciousness is the interface between deliberation and long-term memory

# Consciousness as a Functor

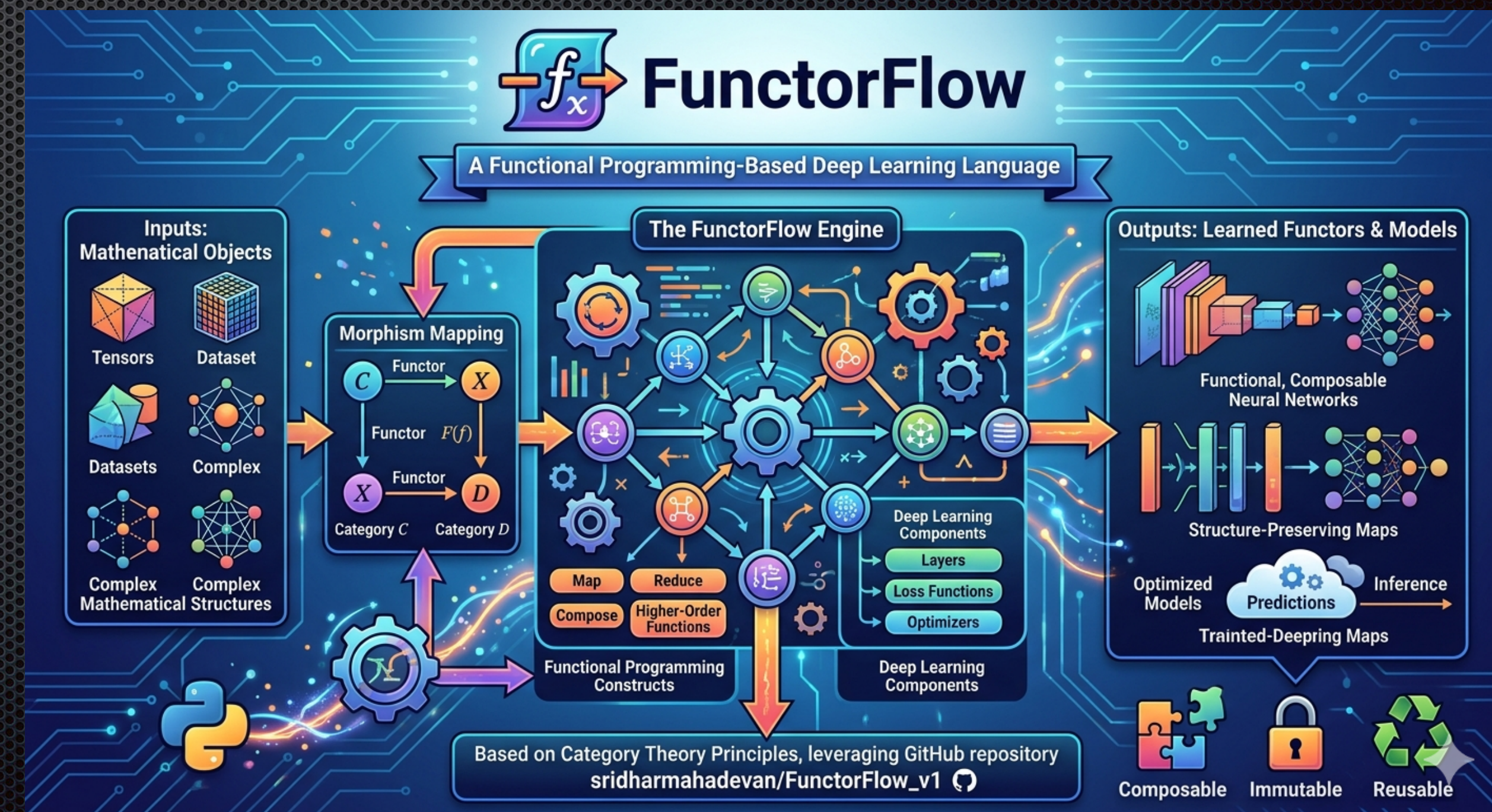


# Implementing Consciousness

- ✦ We are going to see how to implement consciousness in Functor Flow
  - ✦ CLIFF — adds a consciousness layer
  - ✦ Enables parallel processing of multiple queries
  - ✦ Each query is relegated to a team of unconscious agents

# Why FunctorFlow

Why not simply code up a team in Claude Code?



# FunctorFlow v2

## Agent-Native Categorical Workflows

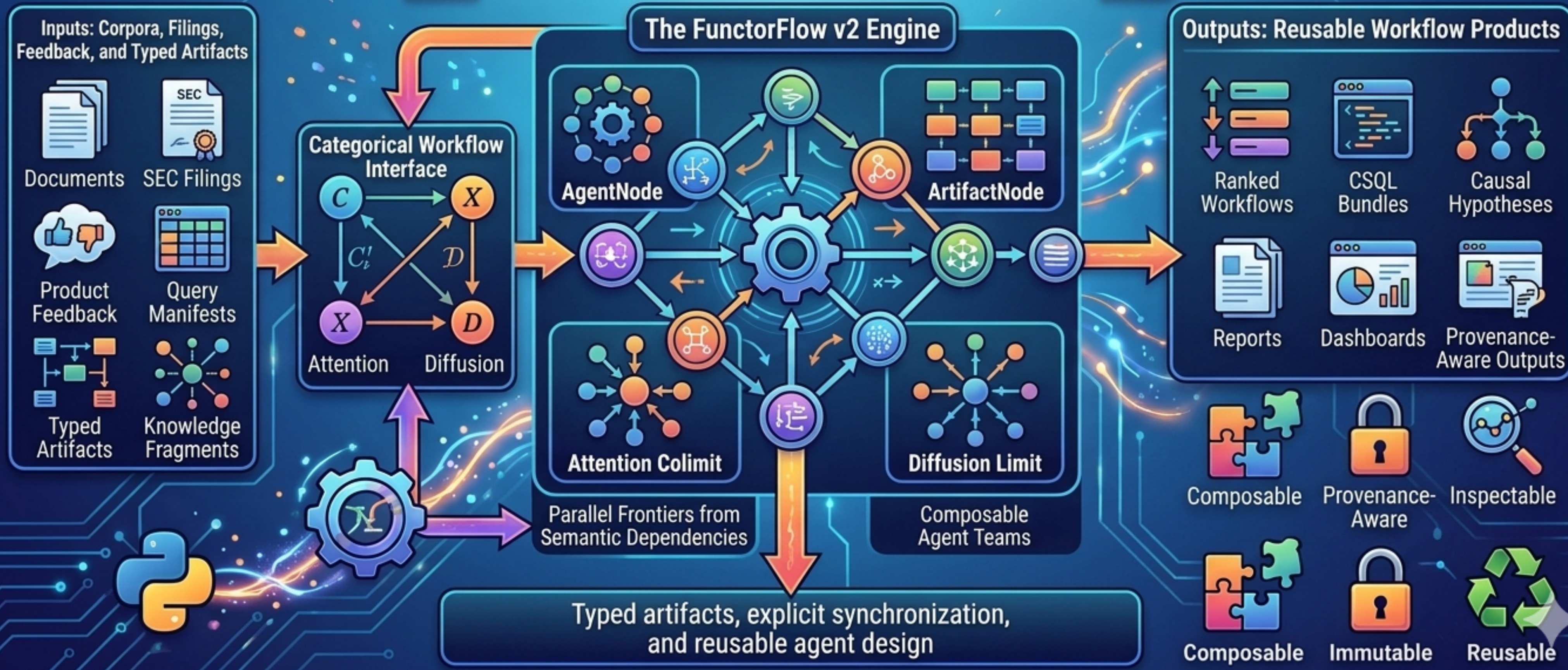


Image generated by Nano Banana 2

# What ad hoc agent teams get right

## Typical boilerplate pattern

- one controller prompt
- a few workers with tool access
- a shared scratchpad or message log
- some imperative merge logic in Python

## Why people like it

- fast to prototype
- easy to customize for one task
- good for local coding and search loops

## Where it starts to creak

- context passing is implicit
- synchronization lives in controller code
- merge logic is bespoke and hard to reuse
- provenance and artifact contracts are often weak
- parallelism is added as an execution trick, not derived from semantics

# Why the Kan / limit-colimit design helps

Question	Ad hoc team code	FunctorFlow v2
What are the units of composition?	prompts, workers, callbacks, controller code	agents, typed artifacts, attentions, diffusions
How is context passed?	hidden in prompts or shared logs	explicit attention object for each target agent
How are multiple outputs reconciled?	handwritten merge code	explicit diffusion object with declared inputs
Where does parallelism come from?	manually spawned tasks	derived from dependency frontiers
Can stages be swapped later?	often painful; logic is entangled	yes, if artifact contracts stay fixed
What survives beyond one demo?	mostly code patterns	a reusable workflow interface

# CLIFF

**Consciousness Layer Interface in Functor Flow** is the new user-facing entrypoint for FunctorFlow v3. It keeps one persistent conscious interface open, routes each natural-language query into an unconscious orchestrator, and then lets results compete to report back into the conscious layer.

[Conscious layer GUI](#)[Unconscious routed workers](#)[Inspect / Stop controls](#)[Broadcast-based FF3 design](#)[Open FF3 Manual PDF](#)[Open FF3 LaTeX Manual](#)[Open Technical Report Source](#)[Architecture Note](#)[Repository README](#)

```
python3 -m functorflow_v3.cliff --outdir /tmp/cliff-session
```



## IDENTITY CONCEPT

The mark keeps the CLIFF metaphor simple: a small illuminated ridge above the waterline for the conscious interface, and a much larger submerged iceberg below it for the unconscious orchestration layer. The underwater node pattern suggests many active processes reading and broadcasting through a shared workspace rather than acting in isolation.

## REPOSITORY FOCUS

This repository keeps the current FF3 / CLIFF baseline in a clean git home so it can evolve independently from the frozen FF2 repo. The Python package name remains `functorflow_v3` for runtime compatibility, but the repository identity is now CLIFF.

**Current focus: make message passing, inspection, and attention competition visible.**

## QUICKSTART

### Launch the conscious layer

```
python3 -m venv .venv
source .venv/bin/activate
python3 -m pip install -e .
python3 -m unittest discover -s tests -p 'test_*.py'
python3 -m functorflow_v3.cliff --outdir /tmp/cliff-session
```

- Submit repeated queries from one persistent GUI session.
- Use **Stop query** for active workers.
- Use **Inspect run** to probe partial Democritus output while it is still running.
- Use **Open result** after completion to view the final dashboard.

## SEC IDENTITY

### Required for 10-K / 10-Q / 8-K routes

```
export FF3_SEC_USER_AGENT="Your Name your_email@example.com"
python3 -m functorflow_v3.cliff --outdir /tmp/cliff-session
```

- CLIFF also accepts `--sec-retrieval-user-agent`.
- Compatibility env vars still work: `FF2_SEC_USER_AGENT`, `SEC_USER_AGENT`, `SEC_IDENTITY`.
- Split variables also work: `SEC_CONTACT_NAME` plus `SEC_CONTACT_EMAIL`.

## ROUTES

### Current CLIFF workflow families

#### RESEARCH

##### Democritus

Evidence acquisition, paper corpora, causal question scaffolds, and live inspectable partial GUIs.

#### FILINGS

##### BASKET / ROCKET SEC

10-K / 10-Q / 8-K retrieval, filing materialization, workflow extraction, and visualization suites.

#### PRODUCTS

##### Product Feedback

Comfort, usability, durability, and review-driven dashboard generation with query-based titles.

#### BROADCAST DEMO

##### Culinary Tour

Intent, budget, destination, retrieval, shortlist, and itinerary agents that publish and consume conscious broadcasts.

## TECHNICAL REPORT

### Architecture-level companion document

A separate FF3 technical report now sits between the user manual and a research paper. It describes the actual class structure behind CLIFF, the inherited FF2 agent/artifact layer, KET-style limit/colimit interpretations, and the current orchestration/runtime design.

[Open Technical Report PDF](#)[Open Technical Report TeX](#)

CLIFF

# Consciousness Layer Interface Functor Flow

CLIFF is the conscious interface for FunctorFlow v3. Enter natural-language queries here, and CLIFF will dispatch each request into its unconscious agent orchestrator before the completed results report back into the conscious workspace.

PERSISTENT SESSION

## The prompt window stays open for the next query.

CLIFF keeps this conscious interface open while unconscious agent runs continue in the background. Long-running jobs keep working in the background and completed results can be opened from the conscious-layer run list.

### CLIFF query

Find me 10 recent AMD 10-K filings  
or  
Plan a kimchi culinary tour in Seoul July 6-11 under \$50 per meal  
or  
Find me 10 recent studies of the benefits of red wine  
or  
How comfortable is the Lovesac sectional sofa?

Send To CLIFF

## Session Runs

Queued, running, completed, and failed requests appear here.

RUN-0004

running

unconscious

democritus

Find me 5 studies of the weight loss drug GLP-1.

CLIFF's unconscious orchestrator is running the democritus workflow.

Stop query

Output: /tmp/cliff-session-run-0004-20260405-110315-find\_me\_5\_studies\_of\_the\_weight\_loss\_drug\_glp\_1

RUN-0003

complete

ready in conscious layer

conscious

product\_feedback

How comfortable are the Nike Pegasus 41 running shoes?

The unconscious orchestrator completed the query and reported back into CLIFF's conscious layer. The result is marked in CLIFF's conscious layer and is not auto-opened.

Open result

Output: /tmp/cliff-session-run-0003-20260405-110300-how\_comfortable\_are\_the\_nike\_pegasus\_41\_running\_

Artifact: /private/tmp/cliff-session-run-0003-20260405-110300-

how\_comfortable\_are\_the\_nike\_pegasus\_41\_running\_/product\_feedback/product\_feedback\_run/product\_feedback\_dashboard.html

# Adobe vs Nike

CLIFF resolved the query as a cross-company temporal diffusion comparison over yearly 10-K causal atlases.

Mean yearly cosine  
**0.96**

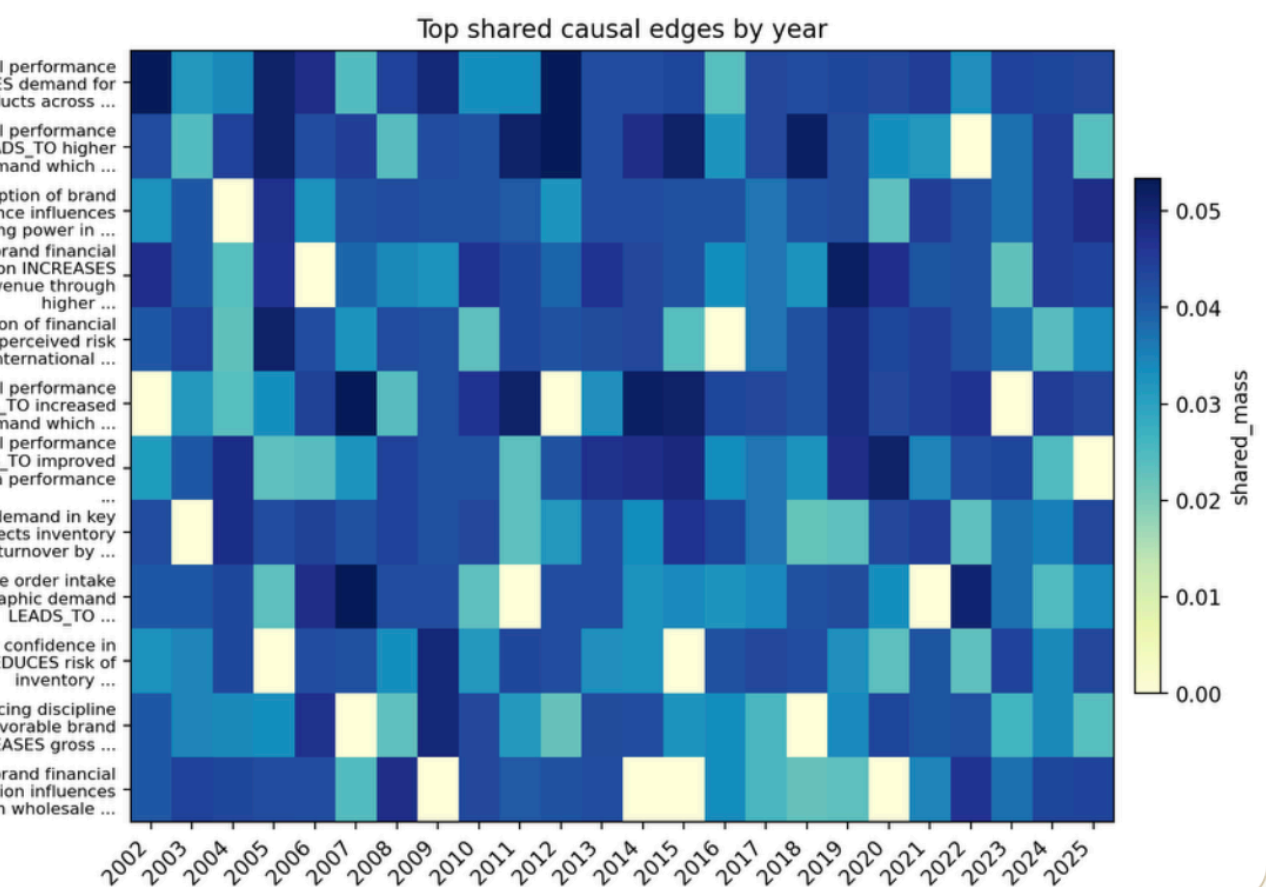
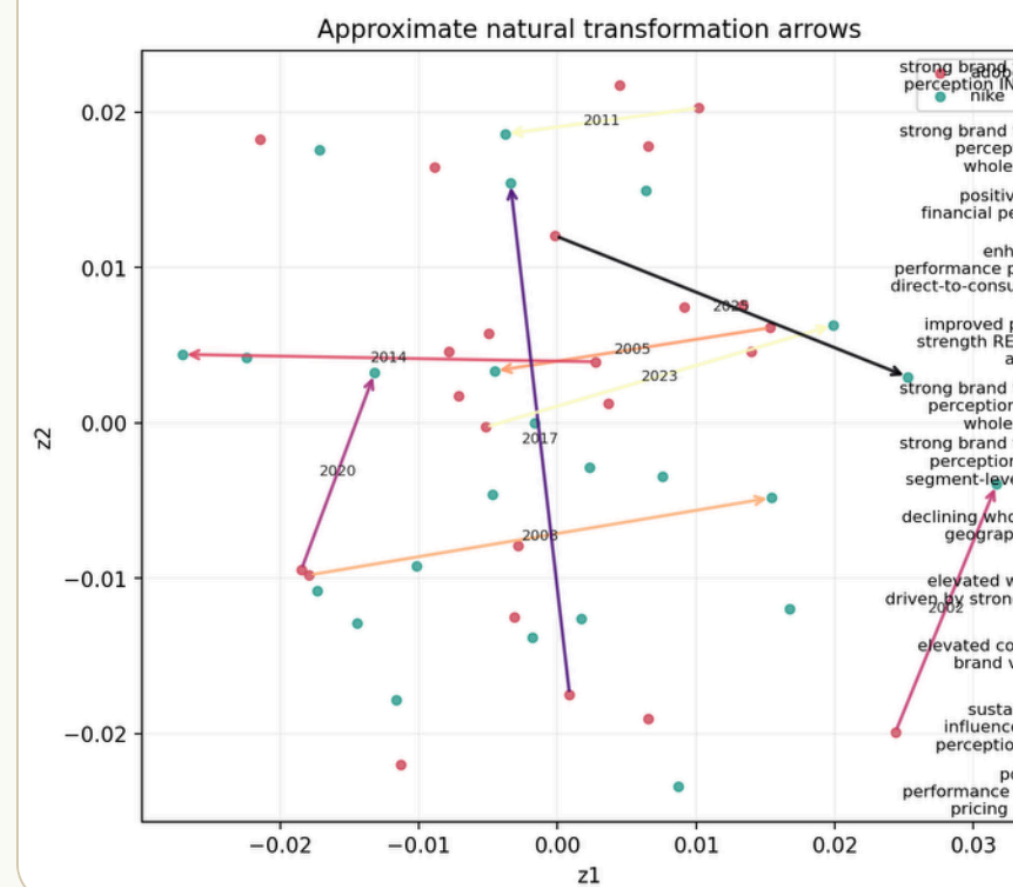
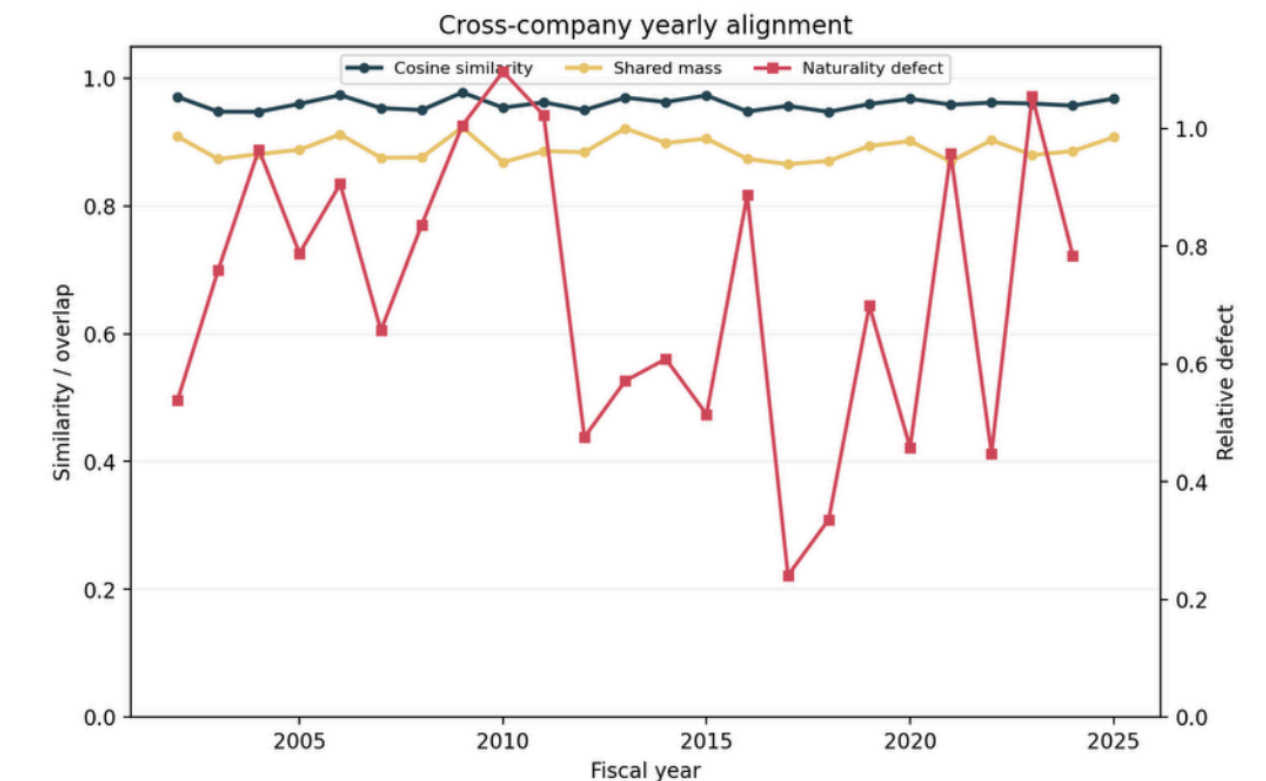
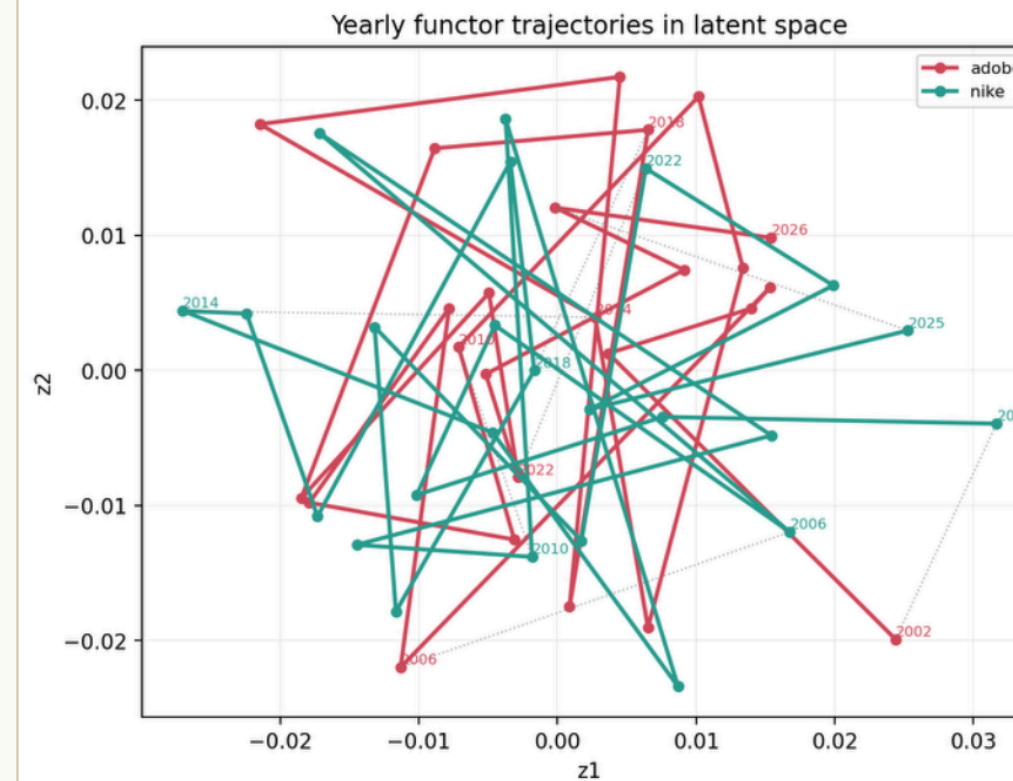
Mean yearly JS divergence  
**0.0171**

Mean relative naturality defect  
**0.7221**

CLIFF: How similar is Adobe to Nike?

DASHBOARD

Cross-Company Functor Dashboard: adobe vs nike



- strong brand financial performance perception INCREASES demand for products across ...
- strong brand financial performance perception LEADS\_TO higher wholesale demand which ...
- positive perception of brand financial performance influences pricing power in ...
- enhanced brand financial performance perception INCREASES direct-to-consumer revenue through higher ...
- improved perception of financial strength REDUCES perceived risk among international ...
- strong brand financial performance perception LEADS\_TO increased wholesale demand which ...
- strong brand financial performance perception LEADS\_TO improved segment-level abttda performance ...
- declining wholesale demand in key geographies affects inventory turnover by ...
- elevated wholesale order intake driven by strong geographic demand LEADS\_TO ...
- elevated consumer confidence in brand value REDUCES risk of inventory ...
- sustained pricing discipline influenced by favorable brand perception INCREASES gross ...
- positive brand financial performance perception influences pricing power in wholesale ...

CLIFF: How comfortable are the Lovesac sectional sofas?

# How comfortable are the Lovesac sectional sofas?

What do retrieved reviews suggest about whether this product is successful, comfortable over the long run, or at risk for returns?

successful complete 6 feedback sources 9 hypotheses



Visual source: <https://fortyreviews.com/lovesac-sectional-reviews/>

## Outcome Snapshot

OVERALL SCORE

**0.8**

AVERAGE RATING

**n/a**

RETURN RISK RATE

**0.0%**

RECOMMENDATION RATE

**100.0%**

## Ablation Comparison

MODE	VERDICT	SCORE	RISK-ADJUSTMENT	WARNING	TOP DRIVER	HYPOTHESES	EVIDENCE COVERAGE
Prompt-like baseline	mixed positive	0.725	+0.000	no	comfort	0	100.0%
BAFFLE structured scaffold	successful	0.800	-0.075	no	style	9	100.0%

BAFFLE structured scaffold score delta vs baseline: +0.075. BAFFLE structured scaffold is less risk-adjusted than the prompt-like baseline by 0.075 score points. BAFFLE structured scaffold shifted the top driver from `comfort` to `style`. BAFFLE structured scaffold added 9 structured hypotheses over the baseline.

## Signal Mix



## Driver Summary

TOP POSITIVE ASPECTS  
comfort, value, durability

TOP NEGATIVE ASPECTS  
style

TOP RETURN-RISK ASPECTS  
none detected

## Aspect Table

ASPECT MENTIONS POSITIVE NEGATIVE RETURN RISK RECOMMEND

## Causal Hypotheses

SOURCE RELATION DESTINATION SUPPORT CONFIDENCE

# How comfortable are the Nike Pegasus 41 running shoes?

What do retrieved reviews suggest about whether this product is successful, comfortable over the long run, or at risk for returns?

mixed positive
complete
6 feedback sources
8 hypotheses



Visual source:  
<https://www.runningshoesguru.com/reviews/road/nike-pegasus-41-review/>

## Outcome Snapshot

OVERALL SCORE

**0.717**

AVERAGE RATING

**4.50**

RETURN RISK RATE

**66.7%**

RECOMMENDATION RATE

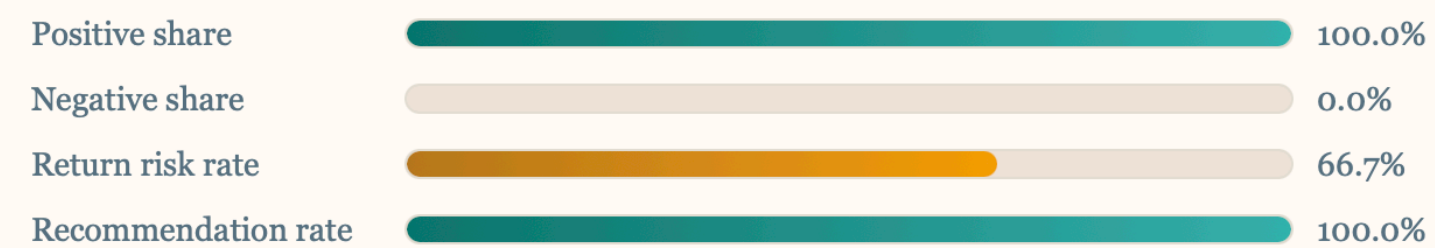
**100.0%**

## Ablation Comparison

MODE	VERDICT	SCORE	RISK-ADJUSTMENT	WARNING	TOP DRIVER	HYPOTHESES	EVIDENCE COVERAGE
Prompt-like baseline	successful	0.931	+0.000	yes	fit	0	100.0%
BAFFLE structured scaffold	mixed positive	0.717	+0.214	no	traction	8	100.0%

BAFFLE structured scaffold score delta vs baseline: -0.214. BAFFLE structured scaffold is more risk-adjusted than the prompt-like baseline by 0.214 score points. BAFFLE structured scaffold changed the return-warning decision from on to off. BAFFLE structured scaffold shifted the top driver from `fit` to `traction`. BAFFLE structured scaffold added 8 structured hypotheses over the baseline.

## Signal Mix



## Driver Summary

TOP POSITIVE ASPECTS  
 comfort, fit, durability

TOP NEGATIVE ASPECTS  
 traction

TOP RETURN-RISK ASPECTS

CLIFF CULINARY TOUR

# kimchi of from July each in Seoul

Plan a kimchi tour of Seoul from July 6 to July 11 for under \$50 for each meal.

July 6

Under \$50 per meal

3 planned days

WHY CONSCIOUSNESS MATTERS

Later unconscious agents read earlier conscious broadcasts before composing the itinerary. This turns the conscious layer into a shared message-passing workspace rather than a dead-end status screen.

PROPOSED ITINERARY

Day 1

## Gwangjang Market Banchan Walk

Jongno

kimchi of from July each emphasis: kimchi and market small plates

**\$18**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 2

## Mapo Fermentation Kitchen

Hongdae

kimchi of from July each emphasis: kimchi-focused grilled dishes and fermented sides

**\$28**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 3

## Itaewon Seasonal Chef Counter

Itaewon

kimchi of from July each emphasis: seasonal Korean tasting plates with kimchi pairings

**\$47**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 3

## Myeongdong Late Supper Hall

Myeongdong

kimchi of from July each emphasis: comfort-food staples and shared side

CONSCIOUS BROADCASTS

BROADCAST-0001

### Normalized culinary intent

intent\_interpreter\_agent

Detected a kimchi of from July each tour in Seoul with time window 'July 6'.

Read from none

Tags: intent, constraints

BROADCAST-0002

### Destination dining context

destination\_context\_agent

Prepared district context for Seoul across Jongno, Myeongdong, Hongdae.

Read from broadcast-0001

Tags: destination\_context

BROADCAST-0003

### Budget guardrails

budget\_guard\_agent

Set a working meal budget of \$50 and filtered plans accordingly.

Read from broadcast-0001

Tags: budget

BROADCAST-0004

### Retrieved culinary stop candidates

stop\_retrieval\_agent

Retrieved 6 candidate stops for kimchi of from July each using the manifest backend.

Read from broadcast-0001, broadcast-0002, broadcast-0003

# CLIFF: Plan a seafood tour in Boston from May 4-8th

## CLIFF CULINARY TOUR

### seafood in Boston

Plan a seafood tour in Boston from May 4-8th.

from May 4-8th

Flexible budget

5 planned days

## WHY CONSCIOUSNESS MATTERS

Later unconscious agents read earlier conscious broadcasts before composing the itinerary. This turns the conscious layer into a shared message-passing workspace rather than a dead-end status screen.

## PROPOSED ITINERARY

Day 1

### Alive & Kicking Lobsters

Boston

seafood emphasis: seafood

**Price unavailable**

**More info**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 2

### Aquapazza

Boston

seafood emphasis: seafood;italian

**Price unavailable**

**More info**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 3

### Atlantic Fish

Boston

seafood emphasis: seafood

**Price unavailable**

**More info**

Guided by broadcast-0005, broadcast-0002, broadcast-0003

Day 4

### Belle Isle Seafood

Winthrop

seafood emphasis: seafood

**Price unavailable**

## CONSCIOUS BROADCASTS

BROADCAST-0001

### Normalized culinary intent

intent\_interpreter\_agent

Detected a seafood tour in Boston with time window 'from May 4-8th'.

Read from none

Tags: intent, constraints

BROADCAST-0002

### Destination dining context

destination\_context\_agent

Prepared district context for Boston across .

Read from broadcast-0001

Tags: destination\_context

BROADCAST-0003

### Budget guardrails

budget\_guard\_agent

No meal budget was specified, so the itinerary remains budget-flexible.

Read from broadcast-0001

Tags: budget

BROADCAST-0004

### Retrieved culinary stop candidates

stop\_retrieval\_agent

Retrieved 6 candidate stops for seafood using the osm\_live backend.

Read from broadcast-0001, broadcast-0002, broadcast-0003

Tags: retrieved\_stops, venues

Backend: osm\_live

BROADCAST-0005

# Extracting usage workflows from reviews

Beyond LLM summarization!

Understand how/why a user was happy/unhappy

Builds on DEMOCRITUS/BASKET/ROCKET technology

## Usage Workflows

WORKFLOW MOTIF	COUNT	STAGES
research -> order -> deliver -> unbox -> assemble -> configure -> sit -> clean -> wash -> reconfigure -> return -> recommend	2	12
research -> order -> deliver -> unbox -> assemble -> configure -> sit -> clean -> wash -> reconfigure -> recommend	1	11
research -> order -> configure -> sit -> wash -> reconfigure -> recommend	1	7
research -> order -> deliver -> assemble -> configure -> sit -> clean -> wash -> reconfigure -> return -> recommend	1	11

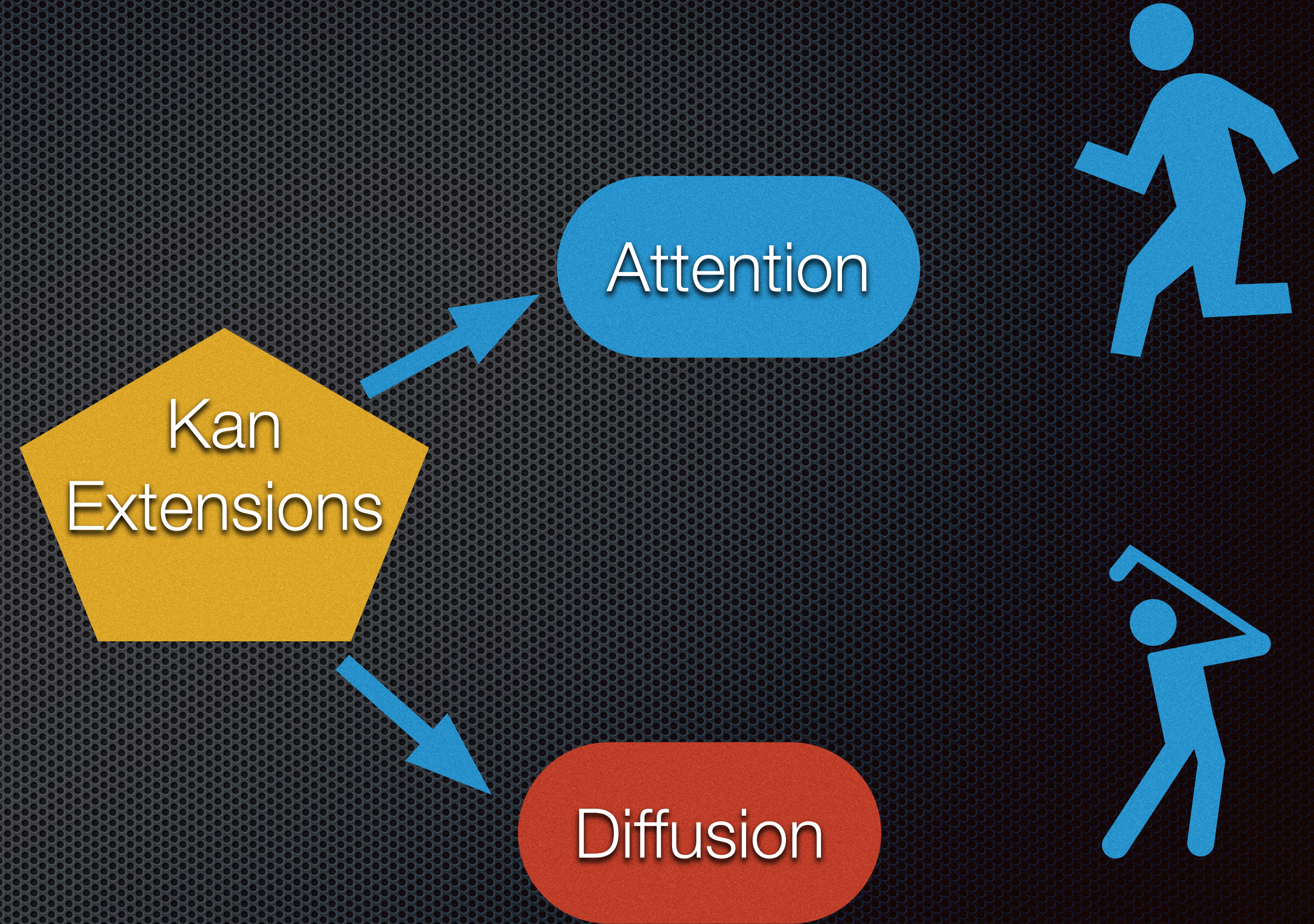
Product success often depends on what customers actually do with the product: assemble it, sit on it, run in it, wash it, reconfigure it, or return it.

# Functor Flow Language

Agentic workflow

Attention and diffusion over agents,  
not just tokens!

Kan Extensions is a unifying design  
motif



# Functor Flow

A new language for encoding agentic systems

High-level built-in primitives facilitate rapid design

```
from .agentic_workflows import (  
    AgentSpec,  
    AgenticWorkflow,  
    ArtifactSpec,  
    AttentionSpec,  
    DiffusionSpec,  
    build_agentic_workflow,  
)
```

# Agentic Design

```
@dataclass(frozen=True)
class AttentionSpec:
    """A colimit-style attention context for a target agent."""

    name: str
    target_agent: str
    input_artifacts: tuple[str, ...]
    source_agents: tuple[str, ...] = ()
    context_kind: str = "attention_context"
    metadata: dict[str, object] = field(default_factory=dict)
```

```
@dataclass(frozen=True)
class DiffusionSpec:
    """A limit-style consistency state for a target agent."""

    name: str
    target_agent: str
    input_artifacts: tuple[str, ...]
    state_kind: str = "diffusion_state"
    metadata: dict[str, object] = field(default_factory=dict)
```

# Further Reading

- ✦ Categories for AI textbook last chapter on consciousness as a functor
- ✦ Think of designing something like CLIFF for your final project
  - ✦ Product recommender, tour guide, shopping guide, stock evaluator
- ✦ How to manage conscious and unconscious processes
  - ✦ Implement the Universal Reinforcement Learning framework