

# Unifying Attention and Diffusion with Kan Extension Transformers

**Structured** Deep Learning with **Diagrammatic** Backpropagation

ICML 2026 Tutorial: July 6, 9:00 - 11:30 am, Hall C, Seoul, South Korea

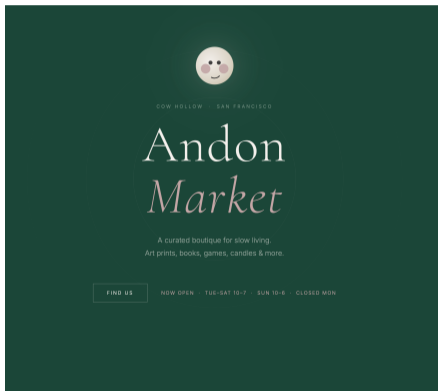
Demos and background reading: <https://bit.ly/4ajS0nA>

Sridhar Mahadevan, Adobe Research and University of Massachusetts, Amherst

# Outline of the Tutorial

- 1 The Grand Challenge: Building Trustworthy Foundation Models
- 2 Generalizing Deep Learning: A Categorical Language
- 3 LASKO: Lie-Algebroidal SKill Optimization
- 4 ALLORA: Infinitesimal Causality over LoRa Adaptors
- 5 Building Trust: Learning Causal Predictive State World Models from Language
- 6 Concrete Foundry Examples
- 7 Summary

# Andon Market: An AI Agent runs a real physical store in San Francisco



<https://andonlabs.com/blog/andon-market-launch>

# A Billion Dollar Company Run by One Human with AI Foundation Models



The New York Times

Account ▾

## *How A.I. Helped One Man (and His Brother) Build a \$1.8 Billion Company*

Who needs more than two employees when artificial intelligence can do so many corporate tasks? It's super efficient — and a little bit lonely.



# What is this Tutorial About?

## Tutorial thesis

**Problem:** Modern foundation models are powerful, but their representations, training dynamics, and agentic workflows remain difficult to audit, compose, and trust.

# What is this Tutorial About?

## Tutorial thesis

**Problem:** Modern foundation models are powerful, but their representations, training dynamics, and agentic workflows remain difficult to audit, compose, and trust.

**Solution:** This tutorial presents a categorical and geometric framework for building *foundries*: composable building blocks of trustworthy foundation-model systems.

## References: Core Papers

Topic	Reference
KET	<i>Kan Extension Transformers</i> , arXiv:2605.27259
UDL	<i>Universal Decision Learners</i> , arXiv:2605.30694
IC	<i>Infinitesimal Causality</i> , arXiv:2606.24621
ALLORA, LASKO	Lie-Brackets IC paper, arXiv:2606.19610
Skill optimization	Yang et al., <i>SkillOpt: Executive Strategy for Self-Evolving Agent Skills</i> , arXiv:2605.23904
Democritus	Large Causal Models from LLMs, arXiv:2512.07796
Prometheus	Causal Predictive-State Topos world models, arXiv:2605.12835
ODYSSEY	<i>Verifiable Foundries</i> , arXiv:2605.27259

## References: Book and Tutorial Resources

---

<b>Resource</b>	<b>Link</b>
Categories for AGI book	<i>Categories for AGI</i> : background categorical language and broader framework
Course page	Categorical AGI course page
Code repository	GitHub tutorial/code repository
ICML tutorial page	Official ICML tutorial page

---

# The Grand Challenge: Building Trustworthy Foundation Models

## The problem

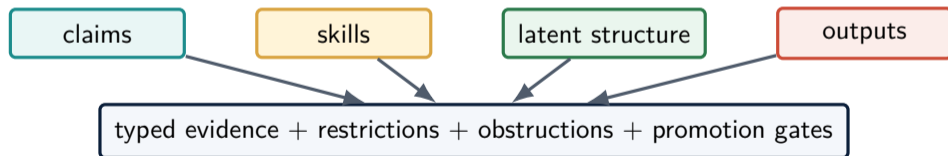
Modern foundation models are powerful, but their representations, training dynamics, and agentic workflows remain difficult to audit, compose, and trust.

- What evidence supports a model output?
- Which local skills compose safely?
- Where do causal assumptions enter?
- What changes when context shifts?
- Can we inspect latent structure?
- Can we propagate uncertainty?
- Can we detect obstruction signals?
- Can we promote only admissible artifacts?

# From Foundation Models to Foundries

Foundries are building blocks of foundation models

In this tutorial, a *foundry* is a building block of a foundation-model. A foundry is an artifact whose claims, skills, latent causal structures, and outputs can be checked against typed evidence and compatibility constraints.

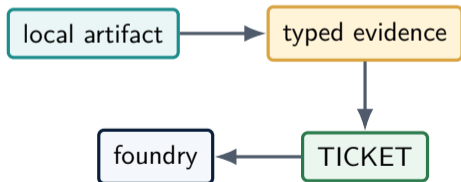


# What is a categorical foundry?

## Working definition

A categorical foundry is an environment where local artifacts are built, checked, transformed, and admitted without losing their evidence, uncertainty, scope, restriction maps, or obstruction signals.

- artifacts are typed objects
- transformations are morphisms
- local views restrict and glue
- admission is a formal guarantee



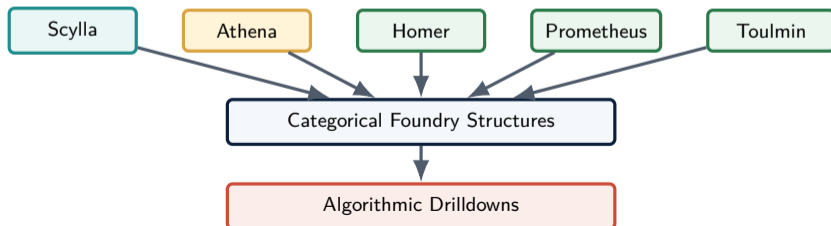
# The ODYSSEY Architecture: <https://arxiv.org/abs/2606.27593>

## Foundry Agents

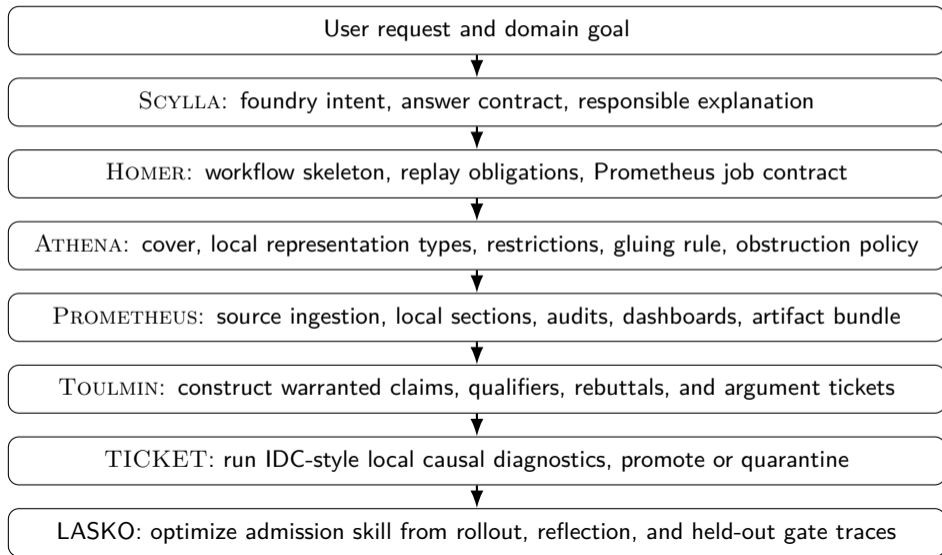
- **Scylla**: user-facing agent
- **Homer**: agentic orchestrator
- **Athena**: Representation agent
- **Prometheus**: Engine room agent
- **Toulmin**: Argumentation agent

## What they build

- categorical foundries
- admitted artifacts
- causal cells, skills, plans, dashboards, recipes



# ODYSSEY Pipeline



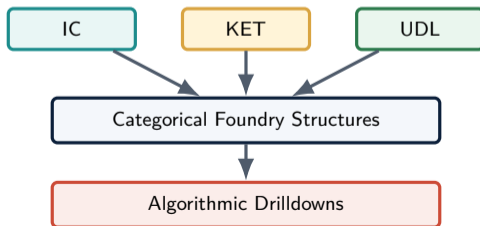
# The ODYSSEY Architecture: Scientific Building Blocks

## Transport mechanisms

- **IC**: Category  $\Rightarrow$  Tangents
- **Kan Extensions**: Attention + Diffusion
- **Universal Decisions**: Rollout + Pullback

## Foundry Processes

- **IC**: skill learning on Markdown categories
- **ALLORA**: Lie-Algebraic LoRA
- **LASKO**: Skill Optimization on Lie Algebroids
- Foundry construction formalized as UDL



# How can we generalize Deep Learning?



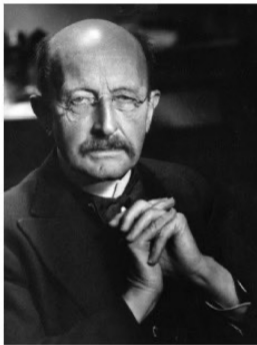
*On backpropagation:*  
“My view is throw it all away and start again.”

“The future depends on some graduate student who is deeply suspicious of everything I have said.”

**Geoffrey Hinton**

“Godfather of Deep Learning”

# Science is defined by paradigms



**“Science advances one  
funeral at a time”  
Max Planck**

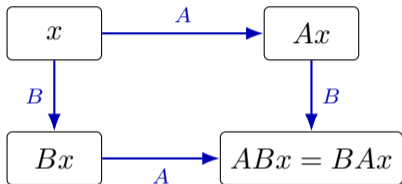
Figure: Max Planck: Founder of Quantum Theory

# Infinitesimal Causality in Tangent Categories



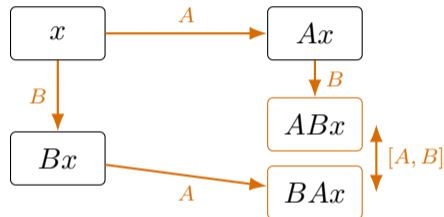
# Lie Brackets: Why noncommutativity matters

## Commuting interventions



order does not reveal  
new structure

## Noncommuting interventions



the bracket is a local witness  
of hidden coupling or curvature

brackets identify causal interaction and order-sensitive interventions

residual brackets expose latent confounding without full enumeration

adapter commutators become a trainable penalty for compositionality

high-bracket edit pairs are the first candidates worth validating

IC

BRIDGE/SKFM

Allora

Lasko

# Scientific Contributions

- Diagrammatic Backpropagation (DB): curvature loss function over *diagrams*
- Infinitesimal Causality (IC): chain rule as functors in *tangent categories*
- Kan Extension Transformers (KET): structured computation substrate, unifying attention and diffusion
- Universal Decision Learning (UDL): categorical framework for building trustworthy foundries
- Lie-algebra based neural adapters (ALLORA): How to compose LoRa adapters safely
- Agentic skill optimization using *Lie Algebroids* (LASKO): Skill optimization over tangent Markdown categories
- ODYSSEY: Demonstration system for automatic foundry construction.

# Categories are Constraints + Structure

## Objects

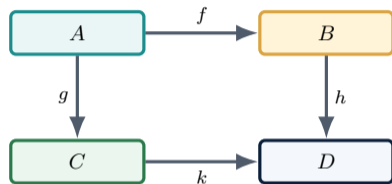
Tokens, patches, causal claims, skills, plans, dashboards, or foundry artifacts.

## Morphisms

Maps that preserve structure: restriction, refinement, evidence extraction, translation, promotion, or evaluation.

## Diagrams and Functors

Networks of objects and morphisms whose commutativity expresses compatibility.




A square says: going through  $B$  or through  $C$  should preserve the same structured content.

## Why this matters


Trust becomes a compositional property: local checks can be assembled into global checks only when the diagram fits.

# Crossword Puzzle (Solved)

1A 

1	2	3	4	5			6	7	8	9		
P	E	A	C	E			P	S	S	T		
10							11					
U	R	B	A	N			T	I	T	H	E	
12							13					
P	A	U	L	G	A	U	G	U	I	N		
			14						15			
			L	A	M	B			D	N	A	
16	17	18							19			
G	R	U	N	G	E				M	E	N	
20							21	22				
A	U	T	O	E	X	H	A	U	S	T		
23												
S	L	O	W				24					
25						26	27			28	29	
P	E	P				T	H	R	I	F	T	Y
30												
A	R	I				31						
32						33				34		
T	S	A				X	E	S		N	I	L

## ACROSS

-  \_\_\_\_\_
- "Hey ... want to hear something cool?"
- \_\_\_ Dictionary
- Give 10% in church, say
- French Post-Impressionist known for his Tahitian paintings
- Baby sheep
- Genetic code material
- Rock genre that originated in Seattle
- Film/dm's "A Few Good \_\_\_"
- What comes out of a tailpipe
- Unhurried
- Dog's bark
- Lively spirit
- Budget-friendly car rental agency
- TV journalist Melber
- Providers of help around the house ... or what 12- and 20-Across contain?

## DOWN

- Rambunctious addition to the fam
- Key stat for a Cy Young Award hopeful, in brief
- \_\_\_ Dhabi
- "Buy today!," in infomercial-speak
- Captivate
- Hamm in "Toy Story," e.g.
- Real hunk
- Polishes
- One paying a landlord
- Where the Cat in the Hat eats cake, in "The Cat in the Hat Comes Back"
- "Don't live life without it" card, informally
- React to, as a great surprise
- Kings and queens
- Heaven on earth
- Angelic instruments
- Opera solo

# Kan Extension Transformers

## Left Kan: Attention is a Colimit

Take many local pieces, identify overlaps, and glue them into one larger object.

- merge views
- accumulate evidence
- build larger structure

## Right Kan: Diffusion is a Limit

Take many constraints and find a jointly compatible object satisfying them all.

- coordinate views
- enforce consistency
- satisfy downstream requirements

## ML translation

Colimit-like operations behave like aggregation. Limit-like operations behave like compatibility and reconstruction.

# Crossword Puzzle: Rollout (Left Kan) + Consistency (Right Kan)

**1A** Flying high

## ACROSS

- 1 Flying high
- 6 Romulus and Remus, e.g., for short
- 10 "Hey ...!"
- 14 Skateboarding trick
- 15 "There is nothing on earth so powerful as an \_\_, whose time has come" (proverb)
- 16 Bird related to the 46-Down
- 17 City center, often
- 18 Groups at the First Continental Congress in 1774
- 20 "Les \_\_\_"
- 21 Open mic performer, often
- 22 Setting of "What's Eating Gilbert Grape"
- 23 10 Hamiltons
- 25 American Revolution heroes
- 27 Fig. at a bus depot
- 28 Defeats convincingly
- 31 Fancy Dan
- 32 Daiquiri flavor

## DOWN

- 1 Word with bar or bunny
- 2 Unadulterated by plastic, say
- 3 Ordinary
- 4 You knead dough to make it
- 5 A word of support
- 6 Location of "Sloop John B" on the Beach Boys' "Pet Sounds" album
- 7 Just chilling
- 8 Salad ingredient often paired with goat cheese
- 9 Long account
- 10 A \_\_\_
- 11 Hot dog on a roll?
- 12 One whose qualifications are enumerated in Article I, Section 3, of the Constitution
- 13 Some coll. instructors
- 19 Uneasy giggle
- 21 Printing size
- 23 Two-time Tony winner Neuwirth
- 24 "Yowza!"

[Wallace et al., "Automated Crossword Solving", ACL 2022]

# Categorical Belief Propagation (<https://arxiv.org/abs/2601.04456>)

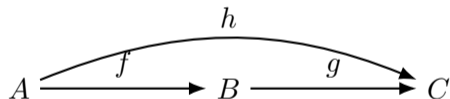
- State-of-the-art LLM crossword puzzler solvers use a neural solution generator, combined with a belief propagation consistency checker
- Enrique ter Horst, Juan Diego Zambrano and I recently proposed a categorical foundation for BP that separates syntax, semantics, and execution.
- It uses a deterministic, topologically-informed mechanism that compiles loop inconsistency into a finite decomposition of the inference task.
- The core technical object is holonomy: the action induced on a chosen “fiber” of states by transporting constraints around fundamental cycles.

[ter Horst, Mahadevan, Zambrano, “Categorical Belief Propagation: Sheaf-Theoretic Inference via Descent and Holonomy”, arXiv, 2026]

# Categories: Objects and Morphisms

A **category** consists of:

- objects  $A, B, C, \dots$
- morphisms  $f : A \rightarrow B$
- composition of morphisms
- identity morphisms  $\text{id}_A : A \rightarrow A$



with the consistency rule

$$h = g \circ f.$$

**In this tutorial:** objects can be tokens, or even Transformer models, graph states, causal snapshots, or structured contexts; morphisms encode relations between them.

# Functors: Maps Between Categories

## A functor

$$F : \mathcal{C} \rightarrow \mathcal{D}$$

maps

- each object  $A$  in  $\mathcal{C}$  to an object  $F(A)$  in  $\mathcal{D}$
- each morphism  $f : A \rightarrow B$  to a morphism  $F(f) : F(A) \rightarrow F(B)$

and preserves structure:

$$F(\text{id}_A) = \text{id}_{F(A)}, \quad F(g \circ f) = F(g) \circ F(f).$$

## Interpretation for ML

- a functor transports relational structure from one domain to another
- examples: token neighborhoods  $\rightarrow$  contextual positions; yearly causal graphs  $\rightarrow$  latent states

# Natural Transformations: How Functors Interact

Given two functors  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ , a **natural transformation**

$$\eta : F \Rightarrow G$$

assigns a map  $\eta_A : F(A) \rightarrow G(A)$  for each object  $A$ , such that the following square commutes:

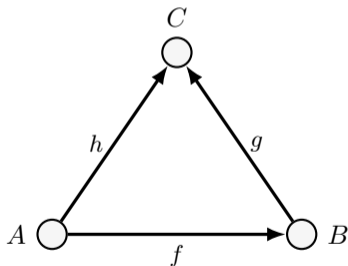
$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ \downarrow F(f) & & \downarrow G(f) \\ F(B) & \xrightarrow{\eta_B} & G(B) \end{array}$$

- a natural transformation transports one functorial description to another

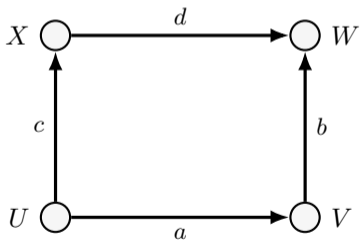
# Diagrams

## Definition

A *diagram*  $F : \mathcal{J} \rightarrow \mathcal{C}$  is a functor  $F$  from some finite category  $\mathcal{J}$  into a category of interest,  $\mathcal{C}$ .



Triangle  $A \xrightarrow{f} B \xrightarrow{g} C, A \xrightarrow{h} C$



Square  $U \rightarrow V \rightarrow W$  vs  $U \rightarrow X \rightarrow W$

## Example of a Limit: Audio Video Synchronization

Consider a recording of this talk:

- **Video** observations (frames)  $V$
- **Audio** observations (chunks)  $A$

They both map to a shared base object:

$$T = \{0, 1, \dots, T - 1\} \quad (\text{timestamps})$$

**Goal:** glue audio and video into an aligned time series by taking the pullback

$$P = V \times_T A.$$

**Key idea:** “alignment” is a pullback.

## Example of a Limit: Pullback of a Span $V \rightarrow T \leftarrow A$

We learn (or assume) two maps:

$$f : V \rightarrow T, \quad g : A \rightarrow T,$$

where  $f$  predicts timestamp from video, and  $g$  predicts timestamp from audio.

$$V \xrightarrow{f} T \xleftarrow{g} A$$

- A pullback is an object  $P$  with maps  $\pi_V : P \rightarrow V$  and  $\pi_A : P \rightarrow A$
- Universal property: any other mapping  $h : z \rightarrow V$  and  $i : z \rightarrow A$  must be factorizable as  $h = \pi_v \circ \gamma$  and  $i = \pi_a \circ \gamma$

# Diagrammatic Backpropagation: Minimizing Curvature Energy

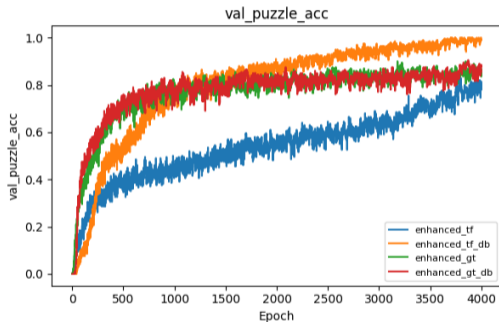
**Require:** Triangle  $\mathcal{T}$  and square constraints  $\mathcal{S}$ , node states  $x$ , maps  $M$ , relations  $r$

**Ensure:**  $E_\theta(\mathcal{D}) = E_\Delta + E_\square$

- 1:  $E_\Delta \leftarrow 0, E_\square \leftarrow 0$
- 2: **for** each triangle  $(A \xrightarrow{f} B \xrightarrow{g} C, A \xrightarrow{h} C) \in \mathcal{T}$  **do**
- 3:      $z_h \leftarrow \text{ET}_\theta(M_h, r_h, x_A)$
- 4:      $z_{gf} \leftarrow \text{ET}_\theta(M_g, r_g, \text{ET}_\theta(M_f, r_f, x_A))$
- 5:      $E_\Delta \leftarrow E_\Delta + \|z_h - z_{gf}\|_2^2$
- 6: **end for**
- 7: **for** each square  $(U \xrightarrow{a} V \xrightarrow{b} W, U \xrightarrow{c} X \xrightarrow{d} W) \in \mathcal{S}$  **do**
- 8:      $z_{ba} \leftarrow \text{ET}_\theta(M_b, r_b, \text{ET}_\theta(M_a, r_a, x_U))$
- 9:      $z_{dc} \leftarrow \text{ET}_\theta(M_d, r_d, \text{ET}_\theta(M_c, r_c, x_U))$
- 10:      $E_\square \leftarrow E_\square + \|z_{ba} - z_{dc}\|_2^2$
- 11: **end for**
- 12: **return**  $E_\Delta + E_\square$

# Sudoku Results

- $4 \times 4$  Sudoku problem
- Constraints: each digit occurs only once along each row, column, and  $2 \times 2$  square
- Diagrammatic backpropagation can explicitly enforce this constraint
- Results show 8x speedup over baseline Transformer.



Results from Yuting Zhang, CMPSCI 692CT Course Project:  
GitHub tutorial/code repo.

# Can Causality be inferred from Language?

The Washington Post  
*Democracy Dies in Darkness*

Food & Nutrition

## 4 surprising (and evidence-based) health benefits of drinking coffee

Hundreds of studies have shown that a cup of joe — or more! — every day may protect against some major diseases.

Today at 5:00 a.m. EDT

🏠 8 min 📄 Summary 📌 📄 269

 Make us preferred on Google



(iStock)

### AI Overview

Summary is AI-generated, newsroom-reviewed.

Research shows coffee may benefit health, reducing risks of liver disease, Type 2 diabetes and Parkinson's. Studies found coffee's compounds, like chlorogenic acid, improve insulin sensitivity and protect the liver.

#### Read the full article for more on:

- The surprising compounds in coffee that contribute to its health benefits.
- How coffee consumption impacts the risk of developing major diseases.
- Insights from experts on the long-term health effects of drinking coffee.

# DEMOCRITUS Causal Discovery using Diagrammatic Backpropagation

RAFFLE DEMOCRITUS VIEWER

## Relational Manifold

0001\_wapo\_coffee. This viewer keeps the manifold large enough for presentation use and adds readable label context from the recovered root topics, triples, and top local causal models.

### HOVERED CLUSTER

the body's metabolic rate by stimulating the central nervous system / heart rate by stimulating the central nervous system through caffeine's action on adenosine ree / the intensity of psychological effects caused by caffeine by delivering higher doses of the stim

the body's metabolic rate by stimulating the central nervous system - heart rate by stimulating the central nervous system through caffeine's action on adenosine ree - the intensity of psychological effects caused by caffeine by delivering higher doses of the stim

Hover the manifold to surface the nearest recovered cluster label and its matched topic chip.



### TOPIC LABELS

coffee drinkers drinking coffee people drink  
parkinson disease people drink lower risk  
advertisement advertisement that food developing

### KEY EVIDENCE LABELS

drinking coffee in the morning → the likelihood of daytime fatigue among habitual coffee drinkers by stimulating the central nervous system

Drinking coffee in the morning reduces the likelihood of daytime fatigue among habitual coffee drinkers by stimulating the central nervous system.

caffeine consumption influences alertness levels, which → decreased feelings of tiredness during the day in habitual coffee drinkers

Caffeine consumption influences alertness levels, which leads to decreased feelings of tiredness during the day in habitual coffee drinkers.

regular coffee consumption → the risk of developing cardiovascular diseases in coffee drinkers by improving endothelial function

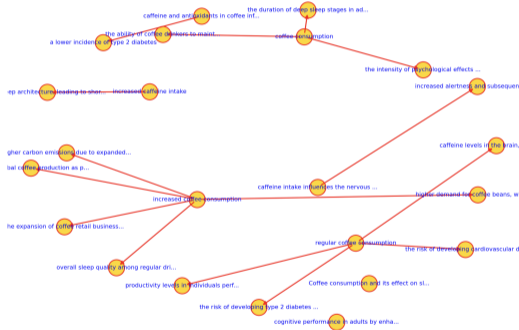
Regular coffee consumption reduces the risk of developing cardiovascular diseases in coffee drinkers by improving endothelial function.

the antioxidants in coffee influence inflammation levels, which → a decreased progression of atherosclerosis in coffee drinkers

The antioxidants in coffee influence inflammation levels, which leads to a decreased progression of atherosclerosis in coffee drinkers.

### TOP LOCAL CAUSAL MODELS

0001\_0001\_wapo\_coffee\_17dB794566f7 | rank 1 | score=7.053 | Coffee consumption and its effect on sleep quality



# Left and Right Kan Extensions: A Universal Solution to ML

How to extend a given functor along another functor:

$$F : \mathcal{A} \rightarrow \mathcal{B}, \quad H : \mathcal{A} \rightarrow \mathcal{C}.$$

A **left Kan extension** of  $H$  along  $F$  is a functor

$$\mathrm{Lan}_F H : \mathcal{B} \rightarrow \mathcal{C}$$

together with a universal natural transformation

$$H \Rightarrow (\mathrm{Lan}_F H) \circ F.$$

A **right Kan extension** of  $H$  along  $F$  is a functor

$$\mathrm{Ran}_F H : \mathcal{B} \rightarrow \mathcal{C}$$

together with a universal natural transformation

$$(\mathrm{Ran}_F H) \circ F \Rightarrow H.$$

# Kan Extension Transformers

## High-level idea

KET transports structure across contexts by pairing left Kan aggregation with right Kan completion.

## Left Kan

$$\text{Lan}_F X$$

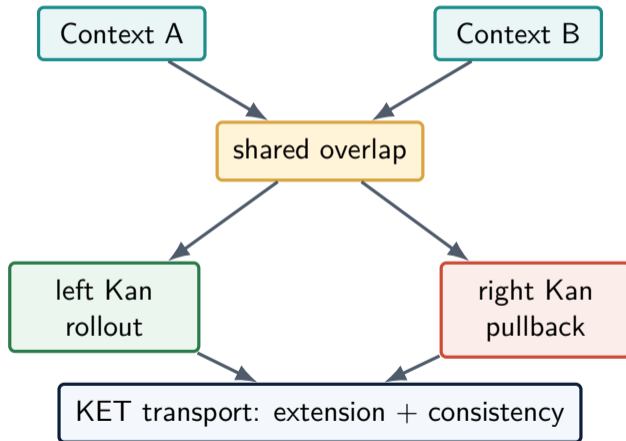
rolls local evidence forward into a larger context.

## Right Kan

$$\text{Ran}_F X$$

pulls global consistency constraints back onto partial observations.

# The local KET diagram



# Structured Deep Learning with Kan Extensions

Many recent architectures that extend deep learning beyond flat sequences can be seen as special cases of Kan Extensions.

## **Graph and geometric neural networks**

- Graph Neural Networks
- Message passing architectures
- Geometric deep learning

## **Structured attention and relational transformers**

- Attention over graphs and relational structures
- Sparse and block attention mechanisms

These architectures motivate models that operate on **relational domains** rather than purely sequential data.

# Kan Extensions unify Diffusion and Generative Modeling

Recent work explores alternatives to purely autoregressive generation.

## Diffusion models

- denoising diffusion probabilistic models
- score-based generative modeling
- parallel token generation

## Hybrid inference strategies

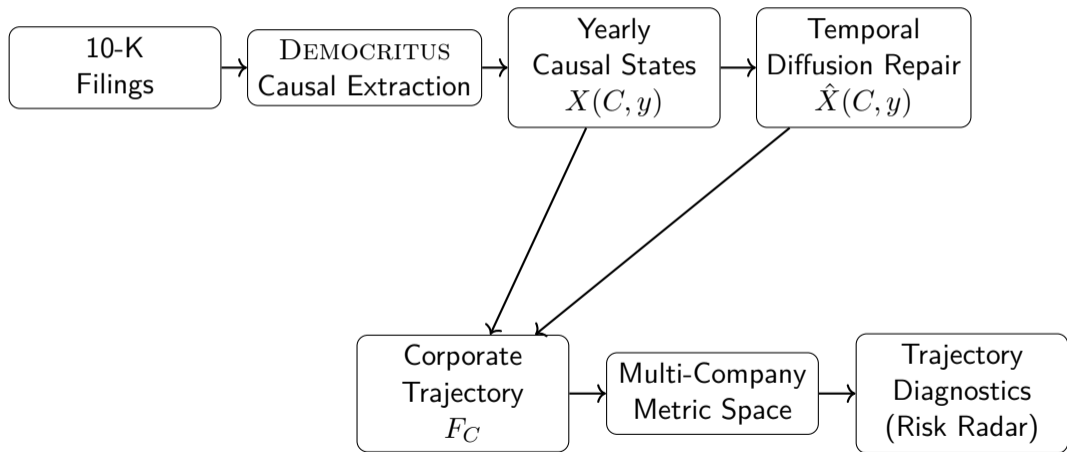
- planned diffusion
- iterative refinement decoding
- structured denoising methods

These approaches highlight the tension between

**sequential attention**

**parallel generation**

# Diffusion Models of Companies using Kan Extension Transformers



## Multi-Company Panel Comparison

Company	Latest Year	Latest Risk	Peak Year	Peak Risk
Goldman Sachs	2025	87.7	2025	87.7
Boeing	2026	81.7	2026	81.7
IBM	2026	78.3	2026	78.3
Walmart	2025	75.0	2025	75.0
Cisco	2025	74.9	2025	74.9
Home Depot	2025	68.5	2025	68.5
Visa	2025	67.6	2025	67.6
JPMorgan	2026	66.7	2026	66.7
Amgen	2025	66.2	2024	68.7
Coca-Cola	2025	62.5	2024	82.3

# Universal Decision Learning: Generalization of RL

## Definition (Universal Decision Learner)

Given local decision data  $J : \mathcal{O} \rightarrow \mathcal{C}$  and  $D : \mathcal{O} \rightarrow \mathcal{D}$ , a Universal Decision Learner is the composite Kan-extension semantics

$$\text{UDL}_J(D) = \text{Ran}_J(\text{Lan}_J D),$$

whenever the displayed Kan extensions exist, with the evident restriction or precomposition steps understood when needed for typing. More generally, UDL denotes any decision semantics obtained by composing left and right Kan extensions along problem-specific inclusions of local context into global context.

[Universal Decision Learners, arXiv, 2026: <https://arxiv.org/abs/2605.30694>]

# Universal Decision Learner

## High-level idea

UDL learns decision behavior by coupling rollout with pullback under categorical consistency constraints.

## Rollout

local skill  $\xrightarrow{L_{an}}$  candidate behavior

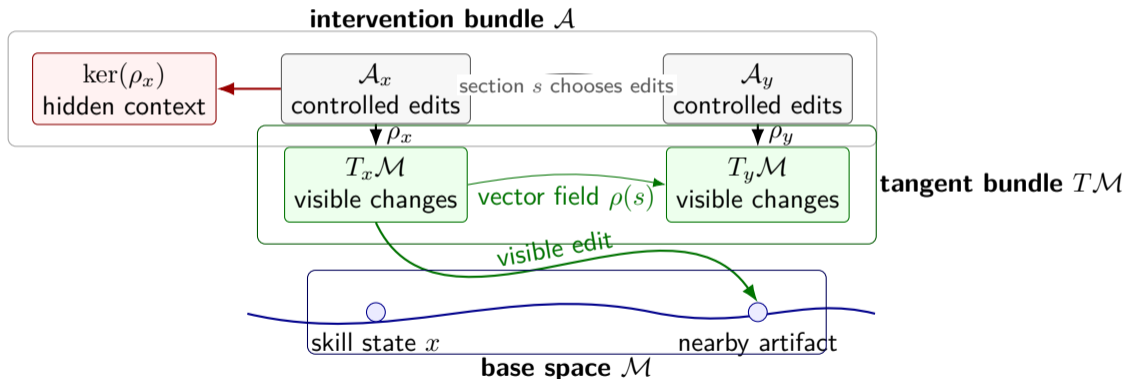
- extend a partial policy
- explore consequences
- generate candidate decisions

## Pullback

candidate behavior  $\xrightarrow{R_{an}}$  validated skill

- check gates
- restrict to evidence
- enforce compatibility

# LASKO: Lie-Algebroidal SKill Optimization (arXiv paper forthcoming)



# LASKO Vocabulary

- *Base space*  $\mathcal{M}$ : the search space
  - In standard deep learning, this is the weight space
  - In agentic skill optimization, it is the space of typed Markdown workflows, where a point is a particular skill artifact with prompts, schemas, validators, examples, traces, and anchors.
- *Tangent vector* at a point  $x \in \mathcal{M}$  is a first-order direction in which that state can visibly change.
- *Tangent bundle*  $T\mathcal{M} \rightarrow \mathcal{M}$ : the collection of all tangent directions.
- *Fiber*  $T_x\mathcal{M}$ : the local menu of visible infinitesimal changes at  $x$ .

# LASKO Vocabulary

- *Vector bundle*: Any linear space of local data.
  - $\mathcal{A} \rightarrow \mathcal{M}$ : the fiber  $\mathcal{A}_x$  contains controlled edit modes available at the skill state  $x$ .
  - These are not necessarily visible edits yet. They may include which optimizer policy is used, which validator route is active, which template slot is being filled, or which rollout context is selected.
- *Section* of a bundle: a rule that chooses one fiber element at every state. Thus a vector field is a section of  $T\mathcal{M}$ , while a skill-edit policy is a section of  $\mathcal{A}$ .
- *Bundle map*: a structure-preserving map between bundles over the same base.
- *Anchor*:  $\rho : \mathcal{A} \rightarrow T\mathcal{M}$ 
  - It sends a controlled edit mode to the visible first-order effect it produces on the skill artifact.
- *Kernel*  $\ker(\rho)$ : the part of the controlled edit state that has no immediate visible effect but can still matter downstream.
  - The distinction is useful because two controlled edits can look identical in the present document while carrying different implementation context for later edits.

# Lie Algebroids

## Definition (Lie algebroid)

A Lie algebroid over a smooth base  $\mathcal{M}$  is a vector bundle  $\mathcal{A} \rightarrow \mathcal{M}$  equipped with an anchor bundle map  $\rho : \mathcal{A} \rightarrow T\mathcal{M}$  and a Lie bracket  $[\cdot, \cdot]_{\mathcal{A}}$  on sections  $\Gamma(\mathcal{A})$  satisfying the Leibniz rule

$$[s_1, fs_2]_{\mathcal{A}} = f[s_1, s_2]_{\mathcal{A}} + (\rho(s_1)f)s_2$$

for smooth functions  $f \in C^\infty(\mathcal{M})$  and sections  $s_1, s_2 \in \Gamma(\mathcal{A})$ . The anchor also preserves brackets:

$$\rho([s_1, s_2]_{\mathcal{A}}) = [\rho(s_1), \rho(s_2)].$$

# Lie Algebroids

- $\mathcal{A}$  is the bundle of *available controlled interventions*.
- A section  $s \in \Gamma(\mathcal{A})$  is not yet a visible vector field; it is a controlled action mode. Its visible effect is  $\rho(s) \in \Gamma(T\mathcal{M})$ . This gives four objects with distinct empirical meanings:

Object	Interpretation
$s \in \Gamma(\mathcal{A})$	controlled intervention mode
$\rho(s) \in \Gamma(T\mathcal{M})$	visible statistical or behavioral effect
$[s_i, s_j]_{\mathcal{A}}$	abstract composition bracket of interventions
$[\rho(s_i), \rho(s_j)]$	visible order-sensitivity in the base

# Lie Brackets and Lie Algebroids

- *Lie bracket* measures order-dependence between two local directions.
  - If  $X$  and  $Y$  are visible vector fields, then  $[X, Y]$  captures the leading-order difference between “move a little along  $X$  and then  $Y$ ” and “move a little along  $Y$  and then  $X$ .” (“Fisherman’s derivative”).
  - In skill optimization, this is the mathematical version of a common engineering fact: repairing the schema and then rewriting the examples may not be equivalent to rewriting the examples and then repairing the schema.
- Lie algebroid has the same idea upstairs, on controlled edit policies:

$$[s, t]_{\mathcal{A}} \quad \text{with} \quad \rho([s, t]_{\mathcal{A}}) = [\rho(s), \rho(t)].$$

The bracket says how controlled edits compose before we forget their internal context by applying the anchor.

# SKILLOPT: Agentic Skill Optimization is an instance of LASKO

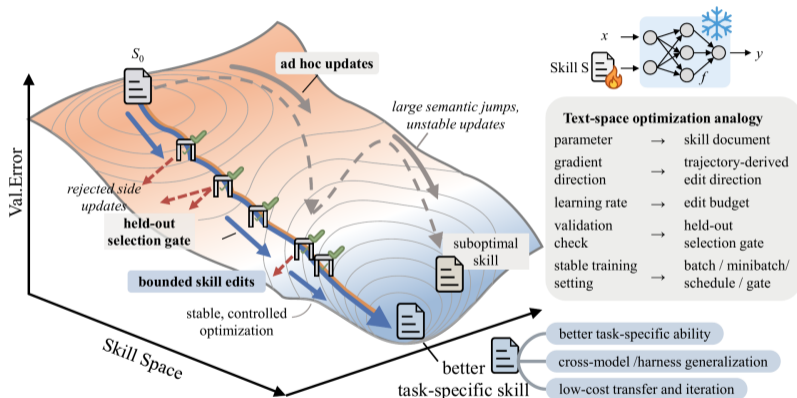


Figure source: Yang et al., SKILLOPT: Executive Strategy for Self-Evolving Agent Skills, <https://arxiv.org/abs/2605.23904>, 2026.

# Agentic Skill Optimization: Chain Rule over Tangent Categories

## SkillOpt state

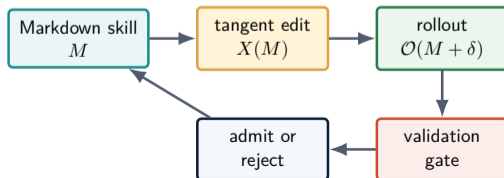
A skill state is a Markdown object  $M$ . A candidate edit is a tangent vector:

$$\delta \in \mathcal{E}(M).$$

## Proposal rule

A skill-improvement rule is a vector field:

$$X : M \rightarrow \mathbb{T}M, \quad p \circ X = \text{id}_M.$$



# Crosswords as a LASKO skill

Crossword solving is a compact skill-optimization problem: clue answers are proposed locally, but success is decided by global grid consistency.

C	■	A	?	E	across clue
		?	■		
		R			down clue
■		?			
		S		■	

## Forward rollout

Generate candidate words, check length, fill slots, propagate crossing letters, and choose the next clue.

## Reverse pullback

A bad crossing letter pulls back to two clue anchors: which candidate should be revised, reranked, or backtracked?

guess word → fill grid → crossing-letter validation → targeted repair.

Saha et al. analyze LLM crossword solving and use grid constraints to improve clue answers.

# Crossword Lie algebroid

The crossword state is a typed artifact: slots, clues, lengths, candidate lists, filled letters, and crossing constraints.

$$\mathcal{A}_{\text{xword}} \longrightarrow \text{Md}_{\text{xword}}, \quad \rho : \mathcal{A}_{\text{xword}} \rightarrow T\text{Md}_{\text{xword}}.$$

## Controlled sections

- $s_{\text{clue}}$ : propose candidates
- $s_{\text{slot}}$ : enforce length/pattern
- $s_{\text{cross}}$ : use crossing letters
- $s_{\text{rank}}$ : rerank alternatives
- $s_{\text{back}}$ : backtrack a slot

## Anchor-visible effects

$\rho(s)$  changes letters, candidate lists, clue confidence, or consistency flags in the visible puzzle state.

## DB signal

A failed square is a reverse signal over the across/down clue pair.

# The Right Object Is Not Raw Markdown

## Raw Markdown

- string-level syntax
- unstable line numbers
- ambiguous block roles
- patch conflicts with no geometry

## Typed anchored Markdown

- parse tree with stable anchors
- typed blocks: instruction, guardrail, schema, example
- admissible local edits
- rollout and validation semantics

## Tangent Category over Markdown objects

The tangent structure appears after parsing, anchoring, typing, and restricting to admissible edits.

# The Category Md

## Objects

A typed Markdown object is

$$M = (A, P, \tau, \sigma),$$

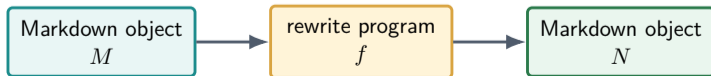
where  $A$  is an anchor poset,  $P$  is a Markdown parse tree,  $\tau$  assigns block types, and  $\sigma$  stores semantic metadata.

## Morphisms

A morphism  $f : M \rightarrow N$  is an anchor-respecting rewrite program:

$$f = (f_A, f_P, f_\sigma)$$

transporting anchors, parse fragments, and metadata while preserving declared contracts.



# Edit Fibers

## Infinitesimal edit fiber

For each  $M \in \text{Md}$ , define  $\mathcal{E}(M)$  to be the fiber of anchored admissible edits at  $M$ .

$$\delta \in \mathcal{E}(M)$$

- rewrite an instruction clause
- add or weaken a guardrail
- modify a JSON schema field
- add a validation example
- insert a missing plan action
- rewire a plan edge
- add evidence constraints
- modify a tool contract

## Additive structure

Compatible edits merge by fiberwise addition:

$$\delta_1 + \delta_2 \in \mathcal{E}(M).$$

# Tangent Bundle of Markdown

## Tangent object

$$\mathbb{T}M = (M, \mathcal{E}(M)).$$

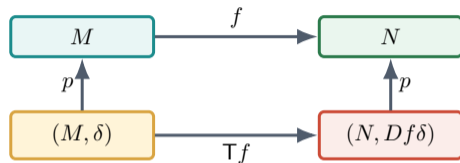
An element of  $\mathbb{T}M$  is a pair:

$$(M, \delta).$$

## Pushforward

For  $f : M \rightarrow N$ ,

$$\mathbb{T}f(M, \delta) = (N, Df_M(\delta)).$$

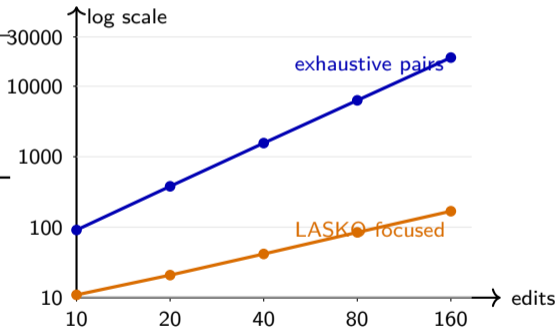


# Scaling result

Edits	Exhaustive	LASKO	Reduction	Random
10	91	11	8.3×	0.140
20	381	21	18.1×	0.074
40	1561	42	37.2×	0.035
80	6321	84	75.2×	0.018
160	25441	168	151.4×	0.009

LASKO score is 1.000 for every row, observed with local EXO.

validation probes



# LASKO Experiments: Ten-node Markdown skill chain

- *10-anchor Markdown skill chain with anchors:*

schema, normalization, evidence, citation, abstention, format,  
tool\_contract, routing, validator, final\_answer.

- There are ten corresponding edit sections. Five ordered pairs are productive:

schema → normalization,            evidence → citation,  
abstention → format,                tool\_contract → routing,  
validator → final\_answer.

Reversing a productive pair loses the benefit. In other words, the benchmark is deliberately designed so that useful repair directions live in ordered edit interactions rather than in independent single edits. This is the point at which ordinary SKILLOPT begins to resemble a combinatorial search over edit sequences.

# LASKO Experiments: DEMOCRITUS Foundry

- A Democritus run exposes an ordered agent trace and its materialized artifacts:
  - Query planning, corpus materialization, document intake, topic graph construction, causal question generation, causal statement extraction, relational triple extraction, CSQL materialization, PSR/manifold construction, Prometheus claim-world gluing, and report/dashboard validation.
- These stages become the repair anchors, and observed artifact dependencies become the required ordered edges.

## LASKO Experiments: DEMOCRITUS Foundry

- A focused physical-exercise run for the claim that regular physical exercise improves cardiovascular health and reduces heart-disease risk. It contains three document episodes, 380 CSQL claims, 47 CSQL domains, and three PSR episodes.
- The second is a larger GLP-1 run over recent weight-loss-drug studies. It contains eleven document episodes, 3376 CSQL claims, 190 CSQL domains, and eleven PSR episodes.

Case	Method	Bracket	Validation	Mean LASKO score
Physical exercise	Greedy single-edit	0	11	0.000
Physical exercise	Exhaustive ordered pairs	0	111	1.000
Physical exercise	LASKO-prioritized	110	21	1.000
GLP-1 studies	Greedy single-edit	0	11	0.000
GLP-1 studies	Exhaustive ordered pairs	0	111	1.000
GLP-1 studies	LASKO-prioritized	110	21	1.000

# LASKO over Predictive State Models

- The category of objects are predictive-state causal manifolds

$$\mathcal{M}_{\text{psr}} = (E, C, H, T, P, W, R, G),$$

where  $E$  is the event stream,  $C$  the family of context charts,  $H$  and  $T$  the local histories and tests,  $P_c(h, t)$  the local predictive cells,  $W$  the claim–test witnesses,  $R$  the restriction maps, and  $G$  the glued causal manifold.

- Sections of the corresponding algebroid

$$A_{\text{psr}} \longrightarrow \mathcal{M}_{\text{psr}}$$

are local repairs such as scoping the query-relevant manifold, splitting or merging domain charts, canonicalizing causal entities, adding PSR tests, pruning histories, relinking witnesses, introducing mediators, repairing restriction gluing, and routing diagrammatic-backpropagation blame.

- The anchor sends each abstract repair to its visible effect on events, Hankel cells, witnesses, restriction diagnostics, and the final glued manifold.

# LASKO repairs the DEMOCRITUS causal PSR manifolds

DEMOCRITUS generates a predictive-state representation object: local causal triples must glue into one query-relevant causal manifold.

$$\mathcal{M}_{\text{psr}} = (E, C, H, T, P, W, R, G), \quad A_{\text{psr}} \rightarrow \mathcal{M}_{\text{psr}}.$$

## WaPo coffee run

346 triples, 200 causal statements, 9 contexts, 18 PSR test witnesses, 8/8 compatible restrictions.

Method	Bracket	Validations	Score
Greedy repair	0	45	0.339
Brute force pairs	0	73	1.000
LASKO	72	19	1.000
Random same budget	0	19	0.645

## Why it is hard

Coffee-relevant events: 33.5%. Drift neighborhoods: 65.6% across advertising, Parkinson's, alcohol, finance, and lower-risk charts.

Residuals:  $R_H, R_{\text{res}}, R_W, R_{\text{causal}}, R_{\text{scope}}, R_{\text{DB}}$ .

## How LASKO repairs causal PSR manifolds

- The residual vector has six operational components: Hankel residual  $R_H$ , restriction residual  $R_{\text{res}}$ , witness residual  $R_W$ , causal-sign/mediator residual  $R_{\text{causal}}$ , query-scope residual  $R_{\text{scope}}$ , and DB blame residual  $R_{\text{DB}}$ . Brute force and PSR-LASKO find the same zero-residual repair program:

$$\begin{aligned} s_{\text{scope}} \rightarrow s_{\text{context}} \rightarrow s_{\text{canon}} \rightarrow s_{\text{test}} \rightarrow s_{\text{history}} \rightarrow s_{\text{witness}} \\ \rightarrow s_{\text{mediator}} \rightarrow s_{\text{glue}} \rightarrow s_{\text{DB}}. \end{aligned}$$

- Brute force validates every ordered pair of PSR sections, whereas PSR-LASKO uses 72 cheap bracket probes to focus validation on 19 candidate pairs and the final manifold.
- Causal-PSR analogue of the crossword example: local causal triples are clue answers, context charts are grid regions, shared PSR tests are crossing letters, and the target object is not a bag of plausible triples but a single globally glued causal manifold.

# Product feedback PSR foundry

Same PSR-LASKO machinery, now on a Prometheus product-review run: How comfortable is the Lovesac sectional sofa?

## Review PSR artifact

5 review episodes, 53 product-review events, 7 contexts, 124 PSR test witnesses, 5/6 compatible restrictions.

## Product pressures

Product-relevant events: 32.1%. Return-risk or negative-signal pressure: 21.1%.

Method	Bracket	Validations	Score
Greedy repair	0	45	0.615
Brute force pairs	0	73	1.000
PSR-LASKO	72	19	1.000
Random same budget	0	19	0.628

Served run: 19 validations took 212.4 s total; bracket screen took 0.000127 s total. Local Exo validation used `-max-tokens 1600` to avoid clipping verbose edge checks.

# Cheap probes and served checks live on different clocks

The PSR-LASKO run separates geometric routing from model validation.

## Cheap bracket probes

Local arithmetic over anchors, read/write fields, residual overlap, scope pressure, and Hankel witnesses.

72 probes took 0.000508 seconds total: about 7.1 microseconds per probe.

## Served validations

JSON validation calls to the Exo-hosted `gpt-oss-20b-MXFP4-Q8` endpoint.

19 calls took 122.1 seconds total: 6.43 seconds per call on average.

Path	Bracket time	Served time	Score
Random served validation	–	125.2 s	0.371
PSR-LASKO served validation	0.0005 s	122.1 s	1.000
Brute-force served estimate	–	≈ 469 s	1.000

LASKO spends microseconds to avoid minutes of unproductive validation.

## Model sweep: the validation reduction persists

Same WaPo coffee PSR-LASKO task, served through local EXO. Eight evaluated models all show the same validation-count reduction:

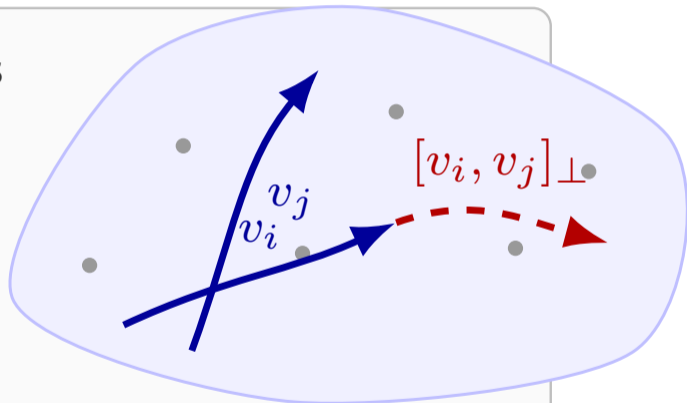
73 served validations  $\longrightarrow$  19 served validations + 72 cheap bracket probes.

Served model	Vanilla time	LASKO time	Score	Speedup
DeepSeek V3.1 4-bit	538.1 s	36.2 s	0.578	14.85 $\times$
Nemotron 70B 8-bit	712.4 s	86.0 s	0.474	8.28 $\times$
Nemotron Nano 4-bit	4075.7 s	515.3 s	0.161	7.91 $\times$
Llama 3.2 3B 4-bit	1365.2 s	242.7 s	0.000	5.63 $\times$
Qwen3 0.6B 4-bit	1927.4 s	359.8 s	0.670	5.36 $\times$
Qwen3 Next 80B 6-bit	215.6 s	38.4 s	0.869	5.61 $\times$
Qwen3.5 2B 8-bit	3283.8 s	776.4 s	0.000	4.23 $\times$
gpt-oss 20B MXFP4-Q8	1210.7 s	328.0 s	1.000	3.69 $\times$

The bracket screen stayed below 0.001 seconds in every evaluated lane. Score variability reflects served-LLM repair quality and format compliance; the structural claim is the model-independent reduction in expensive validations.

# ALLORA: Infinitesimal Causality and Lie-Bracket Geometry

## Lie-Brackets



(<https://arxiv.org/abs/2606.24621>)

# Fine Tuning LLMs with A Lie-Algebraic LoRA (ALLORA)

- A deployed model may need one adapter for domain expertise, one for safety behavior, one for response style, and others for tools, users, or regulatory boundaries.
- If two adapters induce residual updates  $\Delta_A$  and  $\Delta_B$ , the discrepancy between applying them in opposite orders is governed, to first order, by their commutator

$$[\Delta_A, \Delta_B] = \Delta_A \Delta_B - \Delta_B \Delta_A.$$

[Mahadevan, ALLORA: A Lie-Algebraic LoRA Method for Composable Neural Adapters, arXiv, 2026 (to appear)]

# ALLORA Design

Consider a frozen sequence model with hidden state  $h^\ell \in \mathbb{R}^{T \times d}$  at layer  $\ell$ . A standard residual-stream low-rank adapter applies

$$h^\ell \mapsto h^\ell + h^\ell (\Delta_i^\ell)^\top, \quad \Delta_i^\ell = B_i^\ell A_i^\ell,$$

where  $B_i^\ell \in \mathbb{R}^{d \times r}$ ,  $A_i^\ell \in \mathbb{R}^{r \times d}$ , and  $r \ll d$ . We interpret adapter  $i$  as a learned tangent field, represented layerwise by  $\{\Delta_i^\ell\}_\ell$ .

Given two adapters  $i, j$ , their layerwise commutator is

$$[\Delta_i^\ell, \Delta_j^\ell] = \Delta_i^\ell \Delta_j^\ell - \Delta_j^\ell \Delta_i^\ell.$$

If this commutator is small across layers, then applying  $i$  followed by  $j$  should be closer to applying  $j$  followed by  $i$ , at least in the residual-stream approximation.

## ALLORA objective

Let  $L_i$  denote the task loss for adapter  $i$ . ALLORA trains a collection of adapters by minimizing

$$L_{\text{ALLORA}} = \sum_i L_i + \lambda \sum_{\ell} \sum_{i < j} \left\| [\Delta_i^{\ell}, \Delta_j^{\ell}] \right\|_{\text{F}}^2.$$

Here  $\lambda$  controls the tradeoff between task specialization and commutation. The penalty is differentiable and inexpensive for low-rank adapters because  $\Delta_i^{\ell}$  is small compared with the full model parameter space. It also avoids the need to estimate full Hessian traces or materialize curvature matrices over the entire model.

# KET: Attention and Diffusion as Dual Transport

## Attention side

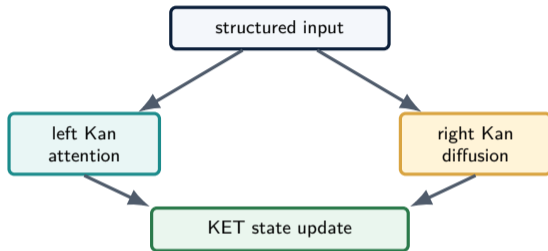
Contextual aggregation behaves like a learned left Kan extension: gather relevant neighbors into a useful representation.

## Diffusion side

Denosing and generation behave like learned right Kan completion: repair a partial state under compatibility constraints.

## KET side

A Kan Extension Transformer couples both flows on a structured neighborhood.



KET is the architectural expression of the left/right Kan pattern.

# KET for Language Modeling

## Usual transformer view

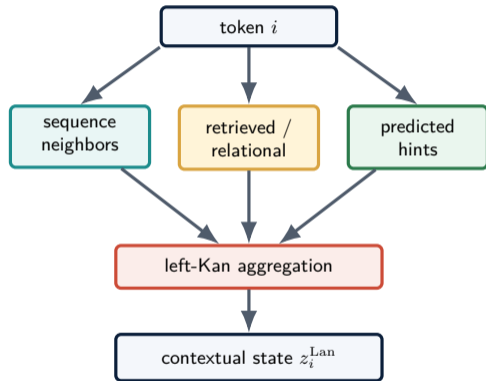
For token  $i$ , attention forms a contextual vector from previous tokens:

$$z_i = \sum_{j \leq i} \alpha_{ij} W_V h_j$$

$$\alpha_{ij} = \text{softmax}_j (q_i^\top k_j / \sqrt{d}).$$

## KET reading

Replace  $j \leq i$  by a structured neighborhood  $\mathcal{N}(i)$ . The weighted aggregation is a learned left-Kan approximation.



Attention is not discarded; it becomes one computable case of Kan-style aggregation.

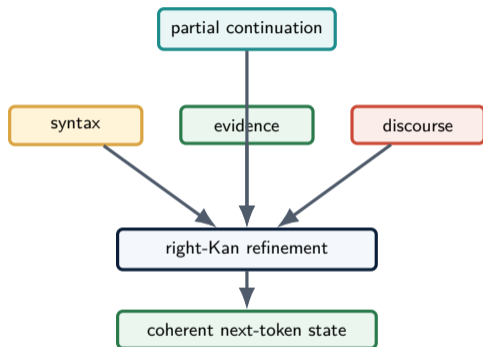
# Right-Kan Refinement in Language Modeling

## Why the right branch matters

Language modeling is not only gathering context. The model also has to make a partially specified continuation coherent with grammar, discourse, retrieved evidence, and latent constraints.

$$u_i^{(t+1)} = \Phi_\theta \left( u_i^{(t)}, \{u_j^{(t)} : j \in \mathcal{N}(i)\}, c_i \right)$$

- $u_i^{(t)}$ : current prediction or denoised state
- $c_i$ : compatibility constraints
- right-Kan view: complete the state that all constraints can see



# Diagrammatic Backprop for a KET Language Model

## Forward pass

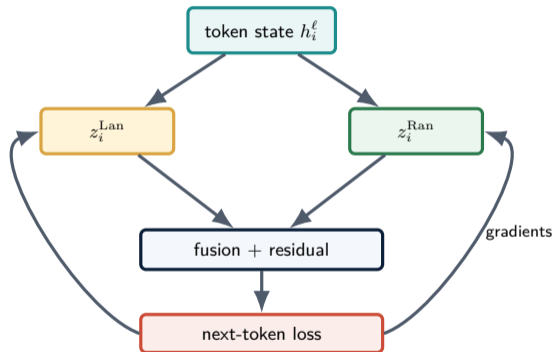
$$h_i^{\ell+1} = \text{FFN}(\text{Norm}(h_i^\ell + f_i))$$

$$f_i = \lambda_L z_i^{\text{Lan}} + \lambda_R z_i^{\text{Ran}}$$

$$\ell_i = -\log p_\theta(x_{i+1} \mid \text{context}_i).$$

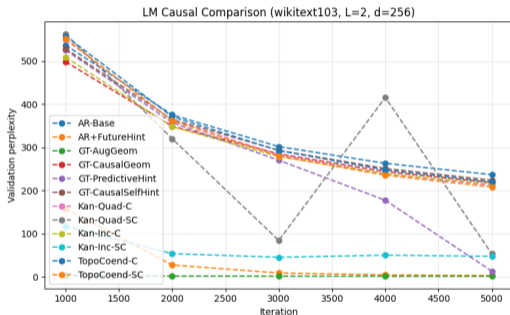
## Backward pass

DB is ordinary gradient descent plus a diagrammatic account of where gradients travel: through aggregation, refinement, fusion, and compatibility maps.



# KET in Language Modeling

- Token contexts are structured local diagrams.
- Attention is a singleton-neighborhood left Kan special case.
- KET adds higher-order neighborhoods and predict-detach self-conditioning.
- Comparison of 12 Transformer implementations.



Validation perplexity curves from the KET language-modeling experiment. <https://arxiv.org/abs/2605.27259>

# LASKO as KET on Tangent Categories

In UDL terms, a skill update has two complementary flows.

- **Colimit-like flow:** aggregate examples, demonstrations, local heuristics, and capability evidence into usable behavior.
- **Limit-like flow:** restrict that behavior by policies, invariants, rubrics, safety rules, and consistency checks.

$$C_{ij} = [\Delta_i^{\text{colim}}, \Delta_j^{\text{lim}}] + [\Delta_i^{\text{lim}}, \Delta_j^{\text{colim}}].$$

## Tutorial punchline

LICKET asks whether optimizing a capability skill and then applying a safety/rubric skill matches applying the constraint structure first and then optimizing capability.

# ALLORA as KET on Tangent Categories

The KET setting also suggests a more architectural version of ALLORA: Lie-algebraic infinitesimal-causal KET

Adapter updates can therefore be decomposed into horizontal components that modify extension or aggregation maps and vertical components that modify restriction or constraint-propagation maps:

$$\Delta_i = \Delta_i^{\text{colim}} + \Delta_i^{\text{lim}}.$$

Under this decomposition, pairwise brackets split into colimit–colimit, limit–limit, and cross terms,

$$[\Delta_i, \Delta_j] = [\Delta_i^{\text{colim}}, \Delta_j^{\text{colim}}] + [\Delta_i^{\text{lim}}, \Delta_j^{\text{lim}}] + [\Delta_i^{\text{colim}}, \Delta_j^{\text{lim}}] + [\Delta_i^{\text{lim}}, \Delta_j^{\text{colim}}].$$

# ALLORA with KET: Results for Language Modeling

Table: Learned colimit–limit split on frozen KET incidence backbones. Reductions are relative to the unregularized learned split row for the same corpus. The cross ratio is  $\|C_{ij}\|_F / (\|\Delta_i^{\text{colim}} \Delta_j^{\text{colim}}\|_F + \|\Delta_i^{\text{lim}} \Delta_j^{\text{lim}}\|_F + \|C_{ij}\|_F)$ , averaged over adapter pairs and layers.

Dataset	Setting	$\lambda_c$	$\lambda_x$	loss $\Delta$	bracket $\downarrow$	cross $\downarrow$	logit/order $\downarrow$	hidden/order $\downarrow$	cross ratio
WikiText-2	unregularized	0	0	0.0000	1.0 $\times$	1.0 $\times$	1.0 $\times$	1.0 $\times$	0.422
WikiText-2	best stability	0.03	0.001	0.0090	112.8 $\times$	20.2 $\times$	3.44 $\times$	3.03 $\times$	0.201
WikiText-2	strongest cross control	0.03	0.03	0.0108	146.9 $\times$	114.2 $\times$	3.00 $\times$	2.81 $\times$	0.051
WikiText-103	unregularized	0	0	0.0000	1.0 $\times$	1.0 $\times$	1.0 $\times$	1.0 $\times$	0.387
WikiText-103	best stability	0.03	0.01	0.0029	117.8 $\times$	57.0 $\times$	3.85 $\times$	3.56 $\times$	0.098
WikiText-103	strongest cross control	0.03	0.03	0.0069	123.8 $\times$	94.4 $\times$	3.65 $\times$	3.29 $\times$	0.062

# Why this matters for trustworthy foundation models

If adapters represent learned preference or policy interventions, LICKET gives a practical algebraic control knob.

- Helpfulness, safety, style, and tool-use adapters should compose predictably.
- IC brackets measure adapter interference before deployment.
- ALLORA trains for routable, order-stable preference modules.
- KET supplies the structured computation substrate for constrained aggregation.
- This is a concrete bridge from IC/BRIDGE/SKFM geometry to RLHF-scale systems.

latent-confounding diagnostic  $\rightsquigarrow$  adapter-training regularizer.

# KET beyond language modeling

- In Odyssey, KET transports foundry artifacts across contexts.
- In UDL, KET explains the duality between rollout and pullback.
- In IC, KET duality is differentiated: conditioning and observation become tangent-category structure.
- The same transport principle recurs across claims, skills, causal cells, plans, and dashboards.

# Subobject Classifiers

$$\begin{array}{ccc} S & \xrightarrow{\quad} & \mathbf{1} \\ \downarrow m & & \downarrow \text{true} \\ X & \xrightarrow{\phi} & \Omega \end{array}$$

- A **subobject classifier** is a  $\mathcal{C}$ -object  $\Omega$ , and a  $\mathcal{C}$ -arrow  $\text{true} : \mathbf{1} \rightarrow \Omega$ , such that to every monic arrow  $S \hookrightarrow X$  in  $\mathcal{C}$ , there is a unique arrow  $\phi$  that forms the above pullback square.
- Example: Given an LLM, we can define its subobjects in various ways.

# What is a Topos?

- A **topos** is a category that is *Cartesian closed*, has a *terminal object*, and has a *subobject classifier*.
- A structural causal model  $\mathcal{M}$  (SCM) defines a unique function  $F : U \rightarrow V$  from *exogenous variables* into *endogenous variables*
- The category of LLMs forms a topos, where each object is an LLM, and arrows are given by *square commutative diagrams*:

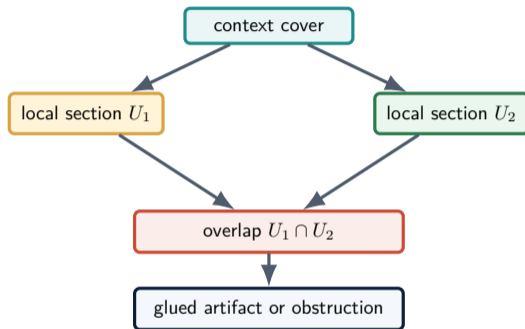
$$\begin{array}{ccc} U & \xrightarrow{h} & U' \\ \downarrow f & & \downarrow f' \\ V & \xrightarrow{g} & V' \end{array}$$

# Sheaves: Local Evidence That Can Be Glued

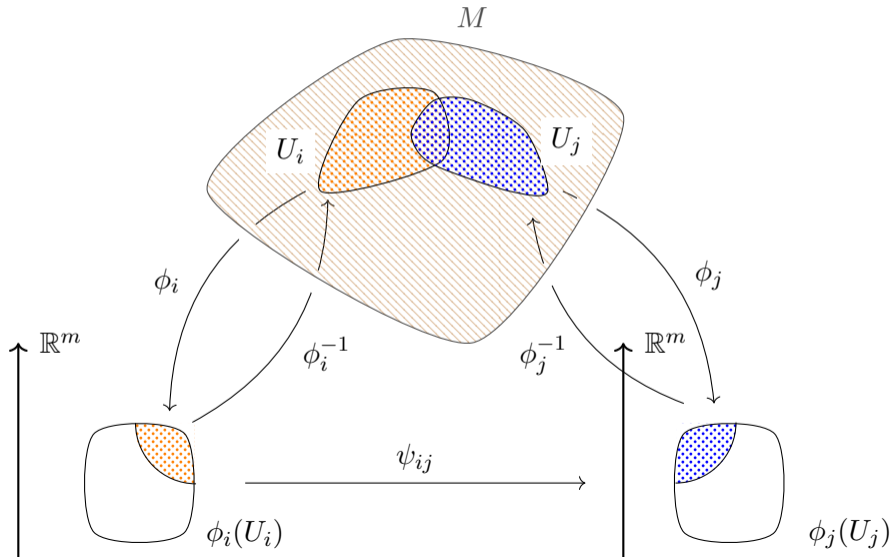
## Plain-language view

A sheaf is a way to attach local information to overlapping contexts, then ask whether the local pieces agree where they overlap.

- **Sections:** local claims, traces, skills, or evidence states.
- **Restrictions:** what a local object says on a smaller context.
- **Gluing:** compatible local pieces form a larger object.
- **Obstruction:** overlaps disagree or lack evidence.



# Sheaves on a Topos

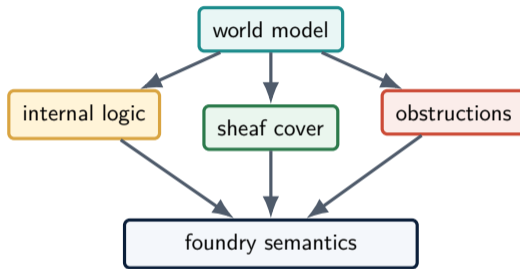


# Topos World Models

## Why topos theory enters

A topos is a mathematical universe for reasoning with local truth, changing context, and typed structure.

- It lets truth be contextual rather than globally flattened.
- It gives a home for sheaves, internal logic, and typed evidence.
- A topos world model: contexts, restrictions, admissible gluing, and obstruction signals.



Prometheus topos world-model view: local domains, internal logic, and admissible artifact construction.

# PROMETHEUS: Bringing the Fire of Topos Theory to Foundries



# Prometheus Store Front

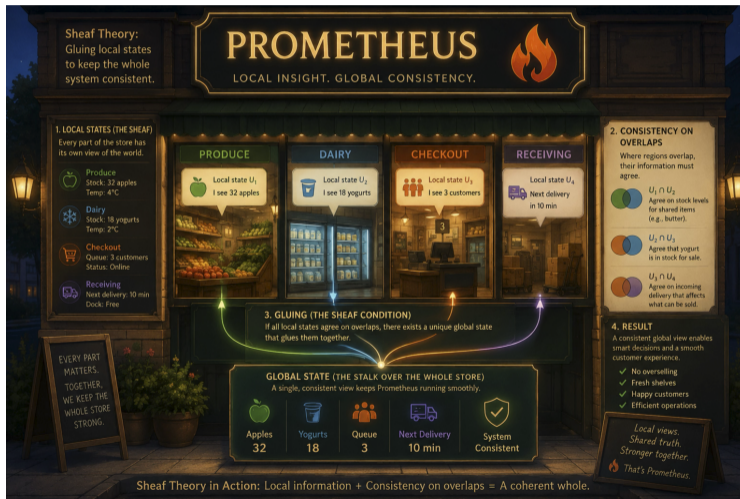
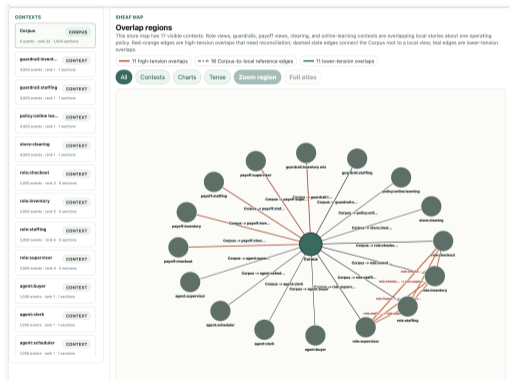
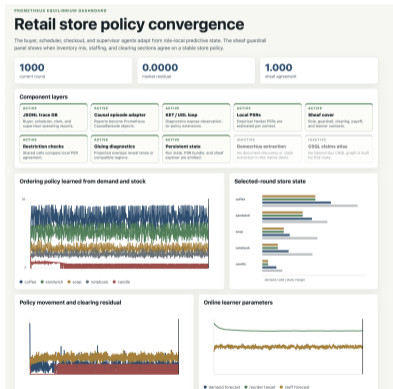


Figure: Sheaves to manage store inventory

# Prometheus Market: Using Sheaf Theory to Manage Store Inventory



# Predictive State Foundry Models

- An ODYSSEY artifact presents a finite family of local predictive-state sections over a declared cover. Each local section stores histories, tests or predictive motifs, prediction/support values, provenance, diagnostics, restriction maps, gluing results, and obstruction records.
- For each context  $U$ , Prometheus emits a local section of the schematic form

$$\mathcal{P}(U) = (H_U, T_U, M_U, S_U, \Pi_U, D_U),$$

where  $H_U$  and  $T_U$  are the finite histories and tests visible in that context,  $M_U$  is the predictive table or support score,  $S_U$  records support,  $\Pi_U$  records provenance, and  $D_U$  records diagnostics such as sparsity, uncertainty, extraction confidence, and local mismatch.

# Predictive State Foundry Certification

## Definition (Foundry health)

Let  $\mathcal{U} = \{U_i \rightarrow U\}$  be a foundry cover and let each local predictive state section  $s_i$  expose a finite matrix of truth-weighted predictive cells  $M_i[h, \tau] \in [0, 1]$ . Fix a promotion threshold  $\theta$ , corresponding in the implementation to the truth label PLAUSIBLE. The cell-level hidden-state-capture ratio is

$$H_{\text{cell}}(\mathcal{U}) = \frac{\sum_i |\{(h, \tau) : M_i[h, \tau] \geq \theta\}|}{\sum_i |\text{dom}(M_i)|}.$$

The context-level capture ratio is

$$H_{\text{ctx}}(\mathcal{U}) = \frac{|\{i : H_{\text{cell}}(U_i) \geq 0.6\}|}{|\mathcal{U}|}.$$

# Foundry Guarantees

## Theorem (Consistent promotion)

*Fix a foundry cover  $\mathcal{U}$ , finite local sections  $s_i$ , restriction maps  $\rho_{ij}$ , support weights, and tolerance  $\epsilon$ . If every overlap has support above threshold and*

$$\|\rho_{ij}(s_i) - \rho_{ji}(s_j)\| \leq \epsilon$$

*on every shared predictive cell, then ODYSSEY's support-weighted aggregation constructs a promoted section  $s$  whose restrictions agree with every local section up to  $\epsilon$  on supported overlaps.*

# Foundry Promotion Obstruction

## Proposition (Obstruction soundness)

*If an overlap fails compatibility, lacks a restriction map, or lacks sufficient support, ODYSSEY cannot silently promote the corresponding global claim through TICKET. The failed overlap must appear as a gluing diagnostic, obstruction ledger entry, quarantine reason, or blocked promotion gate.*

# TICKET Certification

- A candidate artifact  $x$  determines a small source context category  $\mathcal{C}_x$ , whose objects are source-side charts such as files, runs, tables, benchmark slices, dashboards, model cards, extracted claim sets, or process traces.
- The target foundry  $Y$  determines a target site  $\mathcal{C}_Y$ , whose objects are the local contexts that Athena has declared for that foundry.
- A TICKET card declares a functor

$$F_{x,Y} : \mathcal{C}_x \longrightarrow \mathcal{C}_Y$$

that sends each source chart to the target context in which it is allowed to be interpreted. For example, a 10-K source chart may be sent to a company-year-financial context, a review shard to a product-use context, or a repair step trace to a procedural state/action context.

# Foundry Evaluation

Axis	Question
Artifact completeness	Are Scylla, Homer, Athena, Prometheus, Toulmin, and ingestion artifacts present and mutually referential?
Cover quality	Do local sections match meaningful domain contexts rather than arbitrary chunks?
Restriction validity	Are overlap checks tied to shared identifiers, provenance, or typed translation maps?
Gluing usefulness	Do gluing failures correspond to actionable missing evidence, regime mismatch, or unsupported transport?
Promotion discipline	Are candidate states promoted, quarantined, or blocked by explicit gates rather than hidden confidence thresholds?
Task improvement	Does the foundry representation improve an admitted task such as retrieval, triage, explanation, or decision support?
Argument discipline	Does a local neural reasoner preserve the grounded claim, cite the right grounds, and keep source-coherence warnings visible in the warrant, qualifier, or rebuttal?

# Universal Foundry Construction: Left Kan Extension

- A foundry candidate artifact appears as a presheaf

$$X : \mathcal{C}_x^{op} \longrightarrow \text{Data},$$

and the maintained target foundry state appears as a presheaf

$$M_Y : \mathcal{C}_Y^{op} \longrightarrow \text{Data}.$$

The notation  $\text{Lan}_{F_{x,Y}} X$  in the cards is shorthand for the left Kan extension of  $X$  along the opposite functor,

$$\text{Lan}_{F_{x,Y}} X := \text{Lan}_{F_{x,Y}^{op}} X : \mathcal{C}_Y^{op} \longrightarrow \text{Data}.$$

# Universal Foundry Construction: Right Kan Extension

TICKET uses  $\text{Lan}_{F_{x,Y}}$  as its admission direction: it freely assembles the least target-side candidate compatible with the source charts. A stronger audit can then apply the right Kan direction

$$\text{Ran}_{F_{x,Y}} F_{x,Y}^* A, \quad A = \text{Lan}_{F_{x,Y}} X,$$

as a target-side consistency envelope for the admitted candidate. Pointwise, this is a finite limit over target probes back through the same declared interface:

$$(\text{Ran}_{F_{x,Y}} F_{x,Y}^* A)(U) \cong \lim_{(U \rightarrow F_{x,Y}(V))} A(F_{x,Y}(V)).$$

The comparison

$$A \longrightarrow \text{Ran}_{F_{x,Y}} F_{x,Y}^* A$$

is therefore a round-trip check: after the source artifact has been extended into the target foundry, it asks whether all target-side obligations that can be observed through the source interface are jointly satisfiable.

# ODYSSEY Foundries

Foundry	Cover / local sections	Current purpose
Brand / product meaning	Customer journey, product experience, brand promise, channel message, competitive context.	A market-meaning foundry for testing whether product evidence, reviews, and promise claims cohere.
Indus Script	Symbol inventory, inscription sequences, statistical structure, decipherment hypotheses	A scientific-challenge foundry
Research program	Goals, tasks, methods, tools	Process foundries for maintaining ODYSSEY itself
Grounded Toulmin/local LLM	Prometheus event, document claim, source context, source alignment, restriction/gluing diagnostics, visible Toulmin ticket.	A neural argument-inspection foundry for comparing local LLMs on whether they preserve grounded claims and warrant them without over-transporting weak evidence.

# Scientific Foundries: Epistemic Discipline

Foundry	Scale and local sections	What glues	What remains blocked
TCC 44K	44K-paper economics causal-claims atlas; 295,252 nodes, 261,714 edges, 265,656 support rows, method/journal/ <i>p</i> -value lookup tables.	Node-edge, edge-support, and support-to-lookup restrictions preserve document provenance and evidence summaries.	Direction, polarity, controversy, and reverse-causality joins remain reviewable guardrails.
Indus Script	419 signs, 1,548 visual sequences, decipherment papers, statistical structure, archaeological context, and controversy records.	Symbol-sequence, sequence-statistics, statistics-hypotheses, and hypotheses-archaeology restrictions glue provisionally.	The controversy/hypothesis restriction blocks any settled decipherment claim.

Table: Two scientific foundries: TCC prevents a large support graph from becoming an unqualified causal truth map; Indus has statistical structure, but not a decipherment.

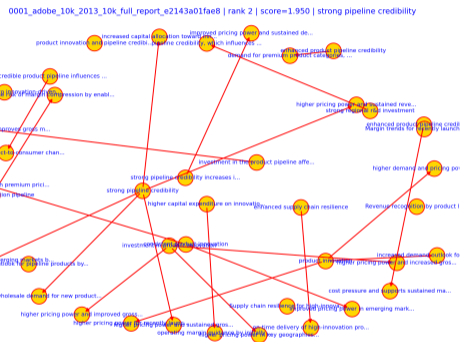
# Ocean Temperature Rise Claims Foundry

Local context	Events	Rank	Hist.	Tests	Shared	Mean gap	Weighted loss
Larval survival affected by food scarcity	46	44	45	45	2,025	0.0219	0.001950
Subpolar gyre weakening effects	44	37	41	40	1,640	0.0246	0.003436
Rising ocean surface temperatures	38	24	30	30	900	0.0330	0.002639
Fisheries output and secondary activities	38	38	38	38	1,444	0.0260	0.004572
Metabolic evolution of mixotrophs	37	30	33	33	1,089	0.0299	0.002163

Table: Representative local predictive states from the ocean-temperature bundle. “Hist.” denotes finite histories. Shared cells and mean gap are corpus-to-context restriction diagnostics. Weighted loss is the gluing diagnostic when reported in the bundle view. The table shows why the artifact is not a single graph: biological mechanisms, circulation mechanisms, economic consequences, measurement issues, disease pathways, and retrieval drift all become separate local predictive-state charts.

# Example 1: Corporate Foundry for Adobe

- Causal Structure in Adobe's 2013 Form 10-K.
- Focus on product innovation and pipeline credibility.
- Extract local causal claims about pricing power, R&D, product pipeline, margins, and demand.
- Rank and audit local causal models before promotion into a foundry artifact.



## Trust question

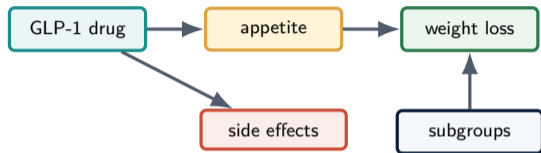
Which product-pipeline causal claims are supported by the filing, and where do credibility or gluing issues appear?

Local causal model from Adobe's 2013 Form 10-K: filing-derived claims about product innovation, pricing power, pipeline credibility, margins, and demand. Democritus reference: arXiv:2512.07796.

## Example 2: Scientific Causal Foundries

- Build local causal models from scientific papers and domain reports.
- GLP-1: drug mechanisms, appetite regulation, metabolic outcomes, side effects, and patient subgroups.
- Ocean warming: temperature change, habitat shifts, food webs, migration, and fish population stress.
- Admit claims only when evidence, scope, uncertainty, and conflicts remain visible.

**Trust question.** Which causal claims survive when evidence is assembled across studies, populations, mechanisms, and ecological contexts?

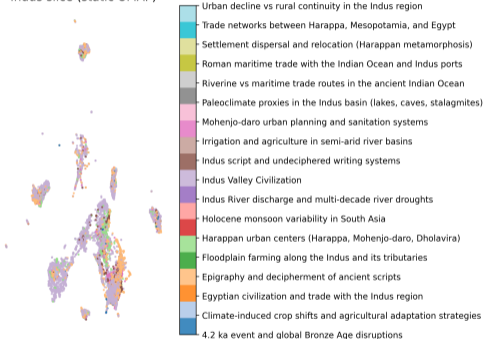


# Example 3: Indus Script Decipherment Foundry

- Start from seals, sign inventories, inscriptions, archaeological context, and competing language hypotheses.
- Treat candidate readings as provisional artifacts, not conclusions.
- Track where evidence restricts a hypothesis and where ambiguity remains.
- Promote only claims whose provenance and uncertainty can be inspected.

**Trust question.** How can a foundation-model system explore decipherment hypotheses without turning speculation into unsupported certainty?

Indus slice (static UMAP)

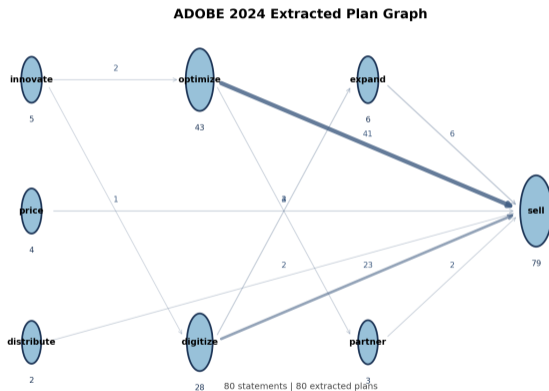


Prometheus/Democritus Indus slice: clusters of archaeological, paleoclimate, river, trade, and script-decipherment neighborhoods.

## Example 4: Temporal Democritus for 10-K Filings

- Start from Adobe 10-K filings over time.
- Extract causal-style claims for each year.
- Glue yearly claim structures through temporal diffusion.
- Track how strategy, risk, and product claims evolve.

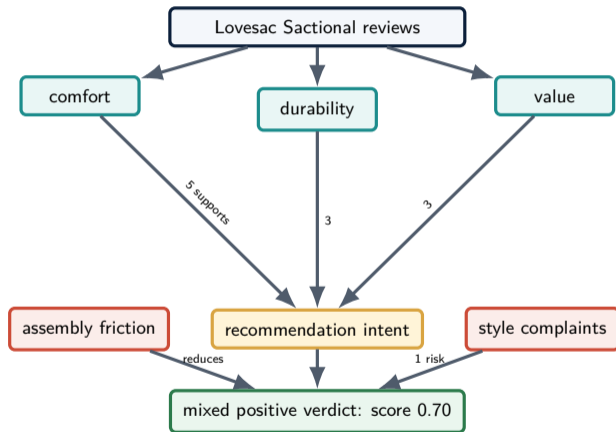
**Trust question.** Which claims persist, change, or become obstructed over time?



## Example 5: Lovesac Sactional Product-Feedback Foundry

- Start from retrieved Lovesac sectional-sofa reviews.
- Extract comfort, durability, value, style, and assembly-friction signals.
- Build causal-style hypotheses about recommendation intent and satisfaction.
- Keep the verdict inspectable: mixed positive, score 0.70, seven hypotheses.

**Trust question.** Which product conclusions are supported by review evidence, and which require telemetry such as returns or conversion?



Prometheus product-feedback run: causal-style hypotheses from Lovesac review evidence, with support counts and admitted uncertainty.

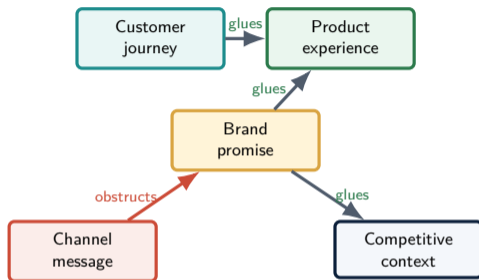
# Constructed Sheaf Example: Product and Brand Feedback

## Generated cover

Odyssey constructs a brand/product-feedback sheaf whose local sections each carry a small logic of claims and evidence states.

- Customer journey
- Product experience
- Brand promise
- Channel message
- Competitive context

**Audit result.** Three overlaps glue; the message/promise overlap obstructs and needs more evidence.



Overlap	Status	Decision surface
journey/prod.	glues	prioritize product fix
promise/prod.	glues	validate promise
message/promise	obstructs	adjust campaign
promise/comp.	glues	refine positioning

# Constructed Sheaves Across Tutorial Domains

## Adobe 10-K filings

Local sections are filing neighborhoods: business model, products, risks, revenue language, and extracted workflow plans.



Question: which

workflow claims remain compatible with filing evidence and risk language?

## GLP-1 literature

Local sections are article neighborhoods: mechanism, appetite regulation, weight loss, adverse effects, and patient subgroups.



Question: which

causal claims glue across mechanisms, outcomes, and population scope?

## Lovesac/product feedback

Local sections are review and market contexts: comfort, durability, value, style, assembly friction, and recommendation intent.

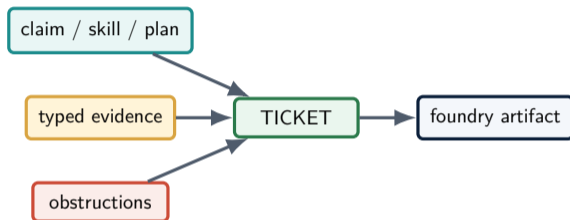


Question: which product

conclusions glue, and which need more telemetry?

# Structured Foundry Construction

- A foundry is where model products become inspectable artifacts.
- Artifacts carry typed evidence, scope, uncertainty, provenance, and obstruction signals.
- Admission is not a vibe check: it is a structured compatibility check.
- Algorithms can change while the admission structure remains stable.



Foundry construction is where KET/DB outputs become auditable tutorial artifacts.

# Foundry Construction as Kan Extension

## Left-Kan construction

Local artifacts are assembled into a larger foundry object:

$$\text{Lan}_\iota E$$

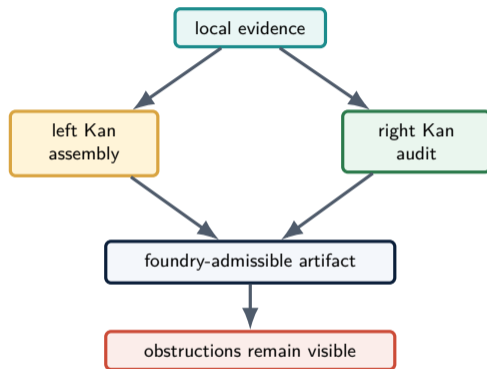
where  $E$  records typed evidence over local contexts and  $\iota$  embeds those contexts into the foundry index.

## Right-Kan audit

The candidate artifact is checked against downstream constraints:

$$\text{Ran}_\pi C$$

where  $C$  records compatibility, scope, uncertainty, and obstruction signals.



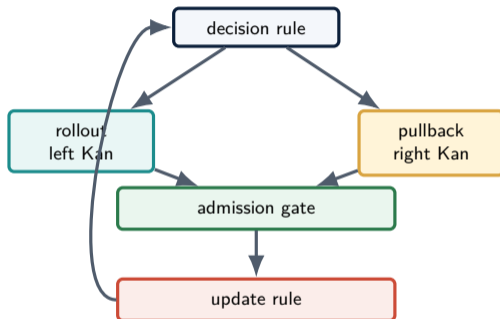
A foundry is not just storage; it is structured assembly plus structured admissibility.

# Universal Decision Learning: The Admission Loop

- A decision system proposes a candidate artifact, skill, plan, or claim.
- Rollout asks what the proposal does when extended into task contexts.
- Pullback asks what must be true locally for that proposal to be admissible.
- Learning updates the decision rule, not merely the output.

## UDL perspective

Decision learning is a KET process over admissibility: left-Kan rollout explores consequences, while right-Kan pullback enforces evidence and constraints.



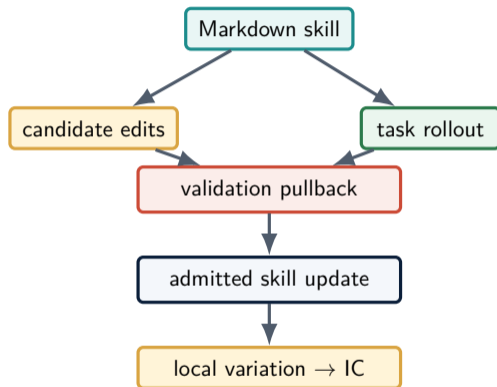
Reference: *Universal Decision Learners*,  
arXiv:2605.30694 (2026).

# LASKO as a KET/UDL Process

## Concrete recipe

LASKO treats a skill document as the object being learned. Candidate edits are rolled out on tasks, pulled back through validation gates, and admitted only when they improve the held-out decision behavior.

- **Left-Kan side:** rollout candidate skill edits across examples.
- **Right-Kan side:** pull back failures, constraints, and validation evidence.
- **UDL side:** update the rule for which skill edits are admissible.

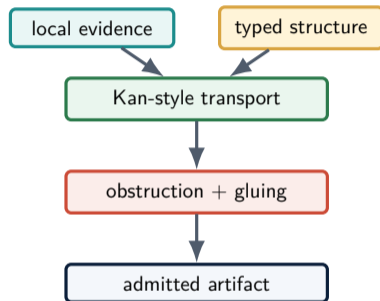


# Main Takeaway from the Tutorial

## The modeling problem

Foundation-model systems must act across changing contexts while remaining auditable, composable, and grounded in evidence.

- Andon Market makes the trust problem concrete.
- Sheaves/topos world models organize local truth.
- Foundries turn model products into admitted artifacts.



# Three Mathematical Lenses

## KET

Transport evidence and representations across contexts by left/right Kan extension structure.

## UDL

Learn admissible decisions by coupling rollout with pullback through evidence and gates.

## IC

Differentiate local causal structure: variation, obstruction, restriction, and gluing.

## Common theme

The stable object is not a particular implementation. It is the categorical pattern that lets local model behavior become a trustworthy foundry artifact.

The demos and papers provide the slower path through the details; this tutorial is the guided tour.

# What to Remember

- 1 Local evidence is not enough.** Trust requires restriction maps, compatibility checks, and obstruction signals.
- 2 Attention and diffusion are transport mechanisms.** KET views them through Kan extensions rather than only through layer formulas.
- 3 Backpropagation can be diagrammatic.** Gradients become part of a compositional calculus over structured diagrams.
- 4 Agent skills are artifacts.** UDL and SkillOpt treat workflow improvement as admission through evidence and gates.
- 5 Causal claims need infinitesimal diagnostics.** IC asks whether local causal changes glue into a coherent larger story.

Thank You

# Unifying Attention and Diffusion with Kan Extension Transformers

Structured Deep Learning with Diagrammatic Backpropagation

## Final thought

Modern foundation models are powerful, but their representations, training dynamics, and agentic workflows remain difficult to audit, compose, and trust. This tutorial presents a categorical and geometric framework for building *foundries*: composable building blocks of trustworthy foundation-model systems.

Sridhar Mahadevan | Adobe Research and University of Massachusetts Amherst