Cognitive Diagrams: Reviewing Categorical Accounts of Linguistic Case

Daniel Ari Friedman
Active Inference Institute
daniel@activeinference.institute
ORCID: 0000-0001-6232-9096
DOI: 10.5281/zenodo.19695260
2026-04-22

2026-04-22

Contents

1 Abstract

Commutative diagrams are cognitively privileged representations because a single diagram simultaneously encodes three things a cognitive agent needs at once: the algebraic structure of a relational situation, the distributional semantics by which language reports on that situation, and the inference process by which belief is updated when new evidence arrives; the traditional linguistic category of case — who did what to whom — is the natural fulcrum on which all three layers turn. We review formalized linguistic case systems as categories whose objects are case roles and whose morphisms are grammatical relations, and we capture every cross-linguistic alignment type (nominative-accusative, ergative-absolutive, tripartite, active-stative, fluid-S) as a structure-preserving functor between case categories. Sentences and discourses become compact-closed string diagrams via DisCoCat and DisCoCirc, computable as executable witnesses of grammatical and cognitive composition; enriching the case category with [0,1]-weighted hom-values supplies quantitative measures and an explicit link to distributional proximity; a topos-theoretic bridge then ties the typological, type-logical, distributional-semantic, enriched-categorical, and quantum layers together. Integrating this stack with Distributional Active Inference yields first-principles, falsifiable predictions for neurophysiological event-related potentials during sentence comprehension, and translating grammatical relations into Positive-Operator-Valued Measurements scales the same scaffolding to coherent multi-agent discourse. In terms of Cognitive Security and AI safety, typological invariants motivate a protocol-level defense: when multi-turn agent interactions are modeled as a fixed category of licensed morphisms with explicit role types and wiring, prompt injection aligns with ill-typed role promotion and can be analyzed as a functorial type violation — an engineering and specification target, not an automatic guarantee, on present-day LLM APIs. The resulting category-theoretic scaffolding makes quantitative neurophysiological predictions, makes prompt injection statically decidable in principle relative to that same fixed protocol (where the interaction grammar is enforced), and turns cross-linguistic typology into a proof-by-functor rather than a taxonomy; the manuscript is accompanied by 1197 executable tests across sixty-four test files at 95.96% line-and-branch coverage on `src/` (from `coverage.json`) and 30 programmatically generated figures, so every formal claim either runs or is honestly flagged as future work. The complete source code, test suite, manuscript, and all figures are available open source on the GitHub repository and archived with DOI 10.5281/zenodo.19695260.

2 Introduction: Bridging Formal Syntax and Situated Neuropragmatics

Central claim. We argue — and then make computationally precise across eleven contributions — that *commutative diagrams are cognitively privileged representations* because a single diagram carries, without translation, three things a language-using agent needs at once: the **algebraic structure** of a relational situation (who stands in which role), the **distributional semantics** by which language reports on that situation (how likely is each assignment given the sentence heard so far), and the **inference process** by which belief is revised when new evidence arrives (how should the assignment be updated). Linguistic *case* is the natural pivot on which the three layers turn: case roles are the objects of the relational algebra, case-marked words supply the distributional evidence, and case assignments are the latent variables whose posterior is updated. Every subsequent pillar of the paper — from ancient *kāraka* theory in section 4, through DisCoCat and DisCoCirc across section 6–section 10, through distributional active inference in section 16, through POVM-based quantum semantics in section 17, to categorical prompt-injection defenses in section 20 — is a witness to this single central claim.

2.1 Case as Relational Algebra: From Surface Inflection to Categorical Morphism

Every natural language must solve the same fundamental problem: encoding *who does what to whom, with what, where, and why.* Morphologically rich languages encode these situational relations overtly via distinct word modifications marking the agent, patient, instrument, or location. Languages like Mandarin or English achieve the same end through strict word order and adpositions. Beneath this surface variation lies a universal computational challenge: a cognitive agent must dynamically assemble and assign *structured roles* to the participants of every encountered scenario, predicting relationships and mapping them against the constraints of a conceptual space.

Today, this same universal computational challenge lies at the heart of the artificial intelligence alignment crisis. The inability of contemporary Large Language Models (LLMs) to securely distinguish *who is allowed to do what to whom* within a text block—allowing passive data to masquerade as commanding instructions—is the root cause of prompt injection and agentic hijacking.

This universality suggests that linguistic *case* is not a morphosyntactic accident, but the reflection of a deeper, embodied cognitive architecture. The lineage of this hypothesis is long: from Pāṇini's ancient *kāraka* theory (which abstracted nominal roles from root actions), through Jakobson's structuralist feature matrices and Fillmore's [1968] establishment of a universal inventory of *deep cases* (Agent, Patient, Instrument), to Mel'čuk's Meaning-Text Theory [1988], which formally treats cases as relational networks mapping surface form to deep semantic structure. Modern typological work by Polinsky and Preminger [2014], Blake [2001], and Haspelmath [2009] confirms that cross-linguistic variation is tightly bounded into a small number of *alignment types*, related by systematic transformations.

This review shows that category theory supplies the mathematical language to formalize relational structure. Commutative diagrams function as cognitively privileged representations. By constraining relational algebras to a two-dimensional graph, these diagrams recruit spatial reasoning, providing boundary enforcement and free-ride inferences unavailable to linear encodings.

2.2 Why Diagrams Earn Free Rides: Spatial Constraints as Inferential Engines

The cognitive science of diagrammatic reasoning provides a strong empirical foundation for this claim. To cast a problem onto a two-dimensional visual topology is to recruit the nervous system's innate, ancient capacity for spatial navigation. Larkin and Simon [1987] demonstrated that diagrams can be computationally superior to sentential representations: their planar constraint enables efficient perceptual grouping that would otherwise require painstaking, explicit logical deduction. Shimojima [1996] refined this with the concept of *free ride inferences*—conclusions that "fall out" automatically from the geometric constraints of a sketch without syntactic manipulation.

In the context of case theory, commutative diagrams offer exactly these embodied advantages. A commutative triangle expressing that morphism $g \circ f$ equals morphism h simultaneously encodes:

1. The compositional structure of the relation (two steps factor through an intermediate case role).
2. The *commutativity constraint* (the direct and indirect visual routes yield the same mathematical result).
3. The full relational context (the entire shape of the scenario is apprehensible in a single perceptual glance).

This is exactly the kind of gestalt understanding that linear notation obscures. Figure 1 lays out a concrete instance. Let $f\colon \mathrm{NOM} \to \mathrm{INS}$ be *uses* ($w = 0.9$), $g\colon \mathrm{INS} \to \mathrm{ACC}$ be *applied_to* ($w = 0.7$), and $h\colon \mathrm{NOM} \to \mathrm{ACC}$ be *acts_on* with $h = g \circ f$ and $w(h) = 0.63 = w(g) \cdot w(f)$ as in Equation 3. The single morphism h is drawn twice in the diagram: once as the direct arrow NOM $\to$ ACC and once as the composite path through INS; commutativity identifies those routes—they are not two independent relations. A separate licensed arrow NOM $\to$ VOC (*addresses*, $w = 0.85$) lies outside that triangle. Dashed

red arrows depict structurally *prohibited* maps (not elements of any $\mathrm{Mor}(\mathcal{C})$), including VOC → NOM and ACC → NOM, for contrast with the solid licensed morphisms. As Giardino [2017] emphasizes, mathematical diagrams engage a hybrid mode of reasoning, intimately wedding perceptual pattern-recognition with theoretical knowledge—a mode effortlessly aligned with the predictive processing architecture of active inference [Namjoshi, 2026]. The intellectual lineage traces through Peirce's *existential graphs*: his spatial logic system proved that first-order reasoning can be conducted entirely diagrammatically on a "sheet of assertion." Our case diagrams extend this Peircean tradition into relational semantics: inference proceeds not by algebraic churning, but by tracing lines and composing arrows—drawing the contours of thought itself.

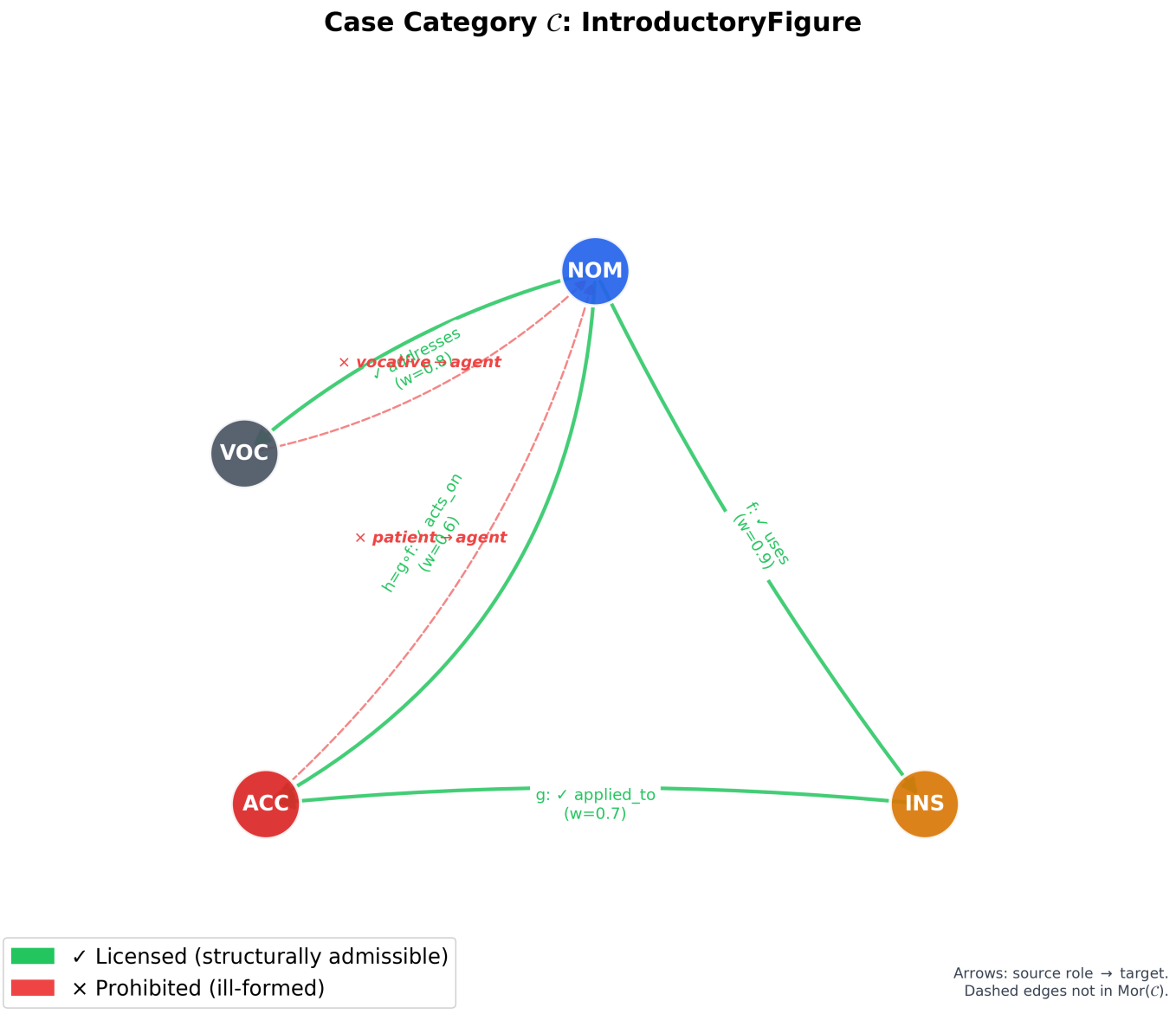

Figure 1: Commutativity in a small case category yields free-ride inference (Equation 3). Objects: NOM, ACC, INS, VOC. Solid green arrows: licensed morphisms; edge labels use f, g, h on the commuting triangle and weights $w \in [0, 1]$. Dashed red: structurally prohibited VOC → NOM and ACC → NOM (not in $\mathrm{Mor}(\mathcal{C})$). Generated by `introductory_case_category()` and `render_case_category`.

2.3 Six Converging Linguistic and Mathematical Traditions

The present synthesis draws on six research traditions, each contributing an essential formal ingredient:

2.3.1 Pillar 1: Case Systems, Alignment Typology, and Structural Admissibility

The empirical foundation comes from linguistic typology. We formalize the case inventory and alignment systems catalogued by Polinsky and Preminger [2014], Claassen [2019], and Wu [2024] as categories with *noun case roles* as objects and grammatical relations as morphisms. Crucially, the categorical framework makes explicit a distinction often implicit in traditional accounts: **structural admissibility**—which morphisms between case roles are *licensed* (structurally well-formed) versus *prohibited* (ill-formed). For example, the morphism NOM→ACC ("agent acts on patient") is universally licensed, while VOC→NOM ("addressee as agent") is structurally inadmissible. This admissibility structure, visualized through solid licensed edges, dashed prohibited edges, and absent transitions in the case category diagram (Figure 1), encodes the combinatorial constraints of argument structure more transparently than any linear notation. Dowty's [1991] proto-role theory—which decomposes thematic roles into clusters of entailments rather than discrete atoms—motivates our use of enriched (weighted) morphisms, where the morphism weight quantifies the *degree* of admissibility along a licensed transition. **Synthetic Case-Role Algebra** (introduced in section 4) extends this line by formalizing proto-roles in a $[0, 1]$-enriched monoidal setting, enabling context-sensitive algebraic manipulation of case-role semantics without ad hoc feature bundles.

2.3.2 Pillar 2: Categorial and Type-Theoretic Grammar

Lambek's [1958] syntactic calculus reformulates grammatical combination as algebraic cancellation in a pregroup, yielding derivations that can be visualized as Joyal and Street's [1991] string diagrams. Song [2022a] extends this with monadic semantics for root syntax, showing that category-theoretic structure reaches down to the sublexical level.

2.3.3 Pillar 3: Categorical Compositional Distributional Semantics (DisCoCat)

Coecke, Sadrzadeh, and Clark's [2010] DisCoCat framework composes distributional word meanings according to syntactic structure using monoidal categories and string diagrams. We lift grammatical pregroup reductions directly to strong monoidal functors, mapping linguistic structures to inner-product vector spaces independent of basis. This framework bridges the two great traditions of linguistic meaning—formal (truth-conditional, compositional) and distributional (context-dependent, statistical)—by using the algebra of compact closed categories to compose vector representations according to type-logical derivations. The resulting categorical semantics provides the *algebraic formalization* of the distributional programme that modern large language models (from Word2Vec [Mikolov et al., 2013] through transformer architectures [Vaswani et al., 2017, Devlin et al., 2019]) implement empirically, making it possible to analyse both symbolic and neural approaches to language within a single mathematical framework. DisCoCirc [de Felice and Coecke, 2020, de Felice et al., 2022] and the lambeq QNLP stack [Lorenz et al., 2021, Quantinuum, 2025] extend this to discourse and circuit compilation; section 10 gives the discourse-level and Gen II details.

2.3.4 Pillar 4: Enriched Category Theory and Distributional Measures

Bradley et al.'s [2021] enrichment over $[0, 1]$ equips hom-sets with distributional proximity measures, supplying the bridge from syntax to statistical semantics. **Categorical magnitude** (scalar "effective size'' of an enriched category) is the Leinster-style invariant that summarises how much relational structure a case system encodes; Bradley's [2021; 2024] operadic and information-theoretic work complements that picture. Leinster and Shulman [2021] extend magnitude to **magnitude homology**, a graded homological invariant that detects higher-dimensional differences between case systems indistinguishable by scalar magnitude alone.

2.3.5 Pillar 5: Topos-Theoretic Bridges and Inter-Theoretic Transfer

Caramello's [2016; 2021; 2023] bridge technique uses classifying toposes as transfer points between mathematical theories: when two formalizations are Morita equivalent (or linked by an explicit bridge), **topos-level invariants** proved in one setting carry over without separate proof. Phillips [2024] strengthens this by showing that the Language of Thought admits universal constructions in a topos, grounding the systematicity of case assignment in the deepest structures of categorical logic.

2.3.6 Pillar 6: Biolinguistics, Neuropragmatics, and Oscillatory Interfaces

The cognitive neuroscience of language is caught between two paradigms: the formal, algebra-driven constraints of **biolinguistics**, which models syntax via the order-free, non-associative topological boundaries of the MERGE operator [Murphy, 2026], and the context-driven, highly associative dynamics of **neuropragmatics**, which models intention and dialogue via Bayesian inference [Gutierrez Cisneros et al., 2026]. To bridge this gap—translating rigid mathematical geometry into flexible social cognition—we adopt Murphy's **ROSE** architecture [Murphy, 2023] (Representation, Operation, Structure,

Encoding): slow oscillatory phase codes (delta/theta) protect the strict algebraic hierarchy of syntax (a "mesoscopic protectorate"), while fast gamma rhythms bind semantic and pragmatic probabilities to these structures. Such bioelectric handoffs provide the mechanistic interface where literal, formal structures are exported into the Default Mode Network and Theory of Mind hubs for situated speech-act evaluation.

The algebraic frameworks in Pillars 2–5 (DisCoCat, enriched categories) map the strict wiring of syntax to vector spaces. The categorical formulation sets topological boundaries within which context and distributed meanings update. This sketches a mechanistic interface between syntax and situated inference.

2.4 Active Inference Closes the Loop: Case Diagrams as the Structural Core of the Generative Model

The unifying framework is **active inference** [Namjoshi, 2026], which casts cognition as approximate Bayesian inference under a generative model. Case-marked commutative diagrams serve as the structural core of such generative models: each diagram specifies a pattern of expected relational dependencies, and the agent's task in understanding (or producing) a sentence is to minimize surprise relative to this diagrammatic prior. The **CEREBRUM** architecture [Friedman and Active Inference Institute, 2024]—Case-Enabled Reasoning Engine with Bayesian Representations for Unified Modeling—provides a computational instantiation of this framework, with case roles as functional specializations of model components in an active inference cycle.

The situation semantics of Barwise and Perry [1983] already conceptualized meaning as structured *situations* with typed constituents—an ontology that maps naturally onto the objects and morphisms of our case categories. Active inference adds dynamics: the agent actively samples evidence to confirm or update its case assignments, using diagrammatic structure to guide exploration.

2.5 Roadmap: Navigating the Eleven Principal Contributions

The remainder of the paper develops this synthesis in order: section 4–section 5 (typology and functorial alignment); section 6–section 7 (pregroup diagrams and case-marked types); section 8 and subsection 8.3 (DisCoCat); section 9 (compact closure and diagram complexity); section 10 (discourse and QNLP); section 11–section 12; section 13; section 14, section 15, and section 16 (active inference, diagrammatic ERP predictions, and DAIF); quantum extensions (section 17, section 18); applications (section 19, section 20). section 3 maps questions **RQ1**–**RQ11** to sections and contributions **C1**–**C11**. Computational verification is summarized in section 15 with test inventory in section 24.

Each subsequent section is self-contained yet cumulative: the reader may enter at any pillar and follow the cross-references forward to the synthesis. The manuscript articulates how DisCoCat functors from pregroup grammars to FinHilb interface with topos-theoretic invariants where Morita equivalence holds. This sketches connections between variational free energy minimization, ZX-style rewrites, and secure protocol enforcement (see C5, C8, C9 for evidence classes).

2.6 What Is New in This Work

The individual ingredients we build with have independent prior art. What is genuinely new here is the *integration* and its computational witness. Building on the prior art most readers will know, our distinct contributions are:

- **Extending Coecke–Sadrzadeh–Clark DisCoCat** [2010; 2017]: we add to the pregroup / FinHilb functorial pipeline explicit *case-marked* wires, making grammatical role a first-class typed structure on every wire rather than an emergent property of word-order composition (section 7). The three-sentence DisCoCirc discourse we ship (section 10) tracks case-role trajectories — Alice NOM→ACC→NOM — that single-sentence DisCoCat does not address.
- **Building on Akgül et al. Distributional Active Inference** [2026]: we instantiate DAIF on a *case-theoretic* generative model rather than a generic MDP, obtaining first-principles derivations of the N400 and P600 ERP amplitudes from a first-order expansion of ΔF (subsection 16.5) — a derivation, not an empirical ansatz.
- **Extending Friedman CEREBRUM** [2024]: we add a *distributional* layer (the full `src/daif/` subpackage) to the eight-case functional-specialisation architecture, replacing point beliefs with return-distribution posteriors throughout the inference cycle (subsection 16.8).
- **Building on Friston active inference** [2017]: we preserve the canonical three-term $G(\pi)$ as the $\beta_{\text{risk}} = 0$ specialisation of a four-term decomposition (Equation 27) that exposes the risk-sensitive variance term needed for distributional RL, and we supply a formal *collapse identity* back to the canonical form so the canonical form is recovered as a special case rather than discarded.

- **Juxtaposing contemporary QNLP (lambeq, Quantinuum)** [2021]: our quantum layer is a *specification* — POVM-based case assignment with honestly-disclosed classical-mixture limits on ρ — sitting alongside, not in place of, hardware QNLP; the section 16 distributional pipeline runs entirely on classical numerics and every section 17 / section 18 claim is explicitly status-labelled as either implemented or literature-bridge.
- **Building on present-day LLM agent-safety practice**: we recast prompt injection in a *type-theoretic reformulation* (section 20) in which multi-turn role promotion becomes an ill-typed morphism in a fixed protocol category, complementing existing content filters with a concrete decidability story — engineering target, not automatic guarantee on today's APIs.

The entire synthesis is backed by a public test suite (1197 tests across 64 files at 95.96% line-and-branch coverage on `src/` (from `coverage.json`)) and 224 tests specifically covering the DAIF distributional layer, so every formalism in the paper either executes or is honestly flagged as future work in subsection 16.7.

3 Research Questions and Manuscript Navigation

This paper addresses eleven core research questions, each developed in specific sections and yielding a principal contribution (**C1**–**C11**, as enumerated in section 21). The table below provides a navigational overview; the reading guide that follows explains the logical dependencies between questions.

Table 1: Core research questions mapped to corresponding manuscript sections and resulting contributions.

#	Research Question	Addressing Sections	C
RQ1	Can linguistic case systems—across all major typological alignment patterns—be formalized within a single algebraic framework, with roles as objects and grammatical relations as morphisms?	section 4, section 5	**C1**
RQ2	Do string diagrams derived from pregroup type reductions provide a computationally and cognitively superior representation of case derivations compared to linear notation?	section 6, section 7, section 22	**C2**
RQ3	Can the DisCoCat meaning functor be extended with case-typed noun spaces to unify formal and distributional semantics—and does this unification generalize to discourse via DisCoCirc?	section 8, subsection 8.3, section 9, section 10	**C3**
RQ4	Does equipping case categories with $[0, 1]$-enriched hom-values yield a principled bridge between symbolic grammar and statistical semantics, and can categorical magnitude serve as a complexity invariant?	section 11, section 12	**C4**
RQ5	Can classifying toposes and Morita equivalence (or explicit bridge toposes) scaffold transfer of **topos-expressible invariants** between distinct formalizations of case theory (typological, type-logical, distributional, enriched), when equivalence is exhibited?	section 13	**C5**
RQ6	Are commutative diagrams *cognitively privileged* representations—recruiting spatial reasoning, free-ride inference, and predictive processing—beyond being merely convenient notation?	section 2, section 14	**C6**
RQ7	Can the entire categorical framework be computationally verified with real (non-mock) automated tests at $\geq$90% coverage, confirming that the abstractions are executable?	section 15, section 16	**C7**
RQ8	Do categorical string diagrams extend naturally into quantum generalizations via TQNNs, ZX-calculus, and sheaf-theoretic quantum semantics—and does quantum entanglement provide genuine advantages for semantic communication?	section 17, section 18	**C8**
RQ9	Can prompt injection attacks be formulated mathematically as illicit cross-category morphisms (type violations), yielding a **decidable** static check **relative to a fixed protocol / interaction category** (compile-time where that grammar is enforced)?	section 20	**C9**
RQ10	Does integrating enriched category theory with active inference generate *falsifiable* neurolinguistic predictions (P600/N400 amplitudes scaling with enriched morphism weights)?	section 14, section 16	**C10**
RQ11	Can grammatical case roles (NOM, ACC, INS) rigorously type-check the tensor payloads of multi-agent AI framework protocols (A2A, MCP, ANP) to prevent agentic hijacking?	section 19	**C11**

Reading guide. **RQ1**–**RQ5** build the mathematical framework cumulatively: each layer depends on the preceding one, from case categories (RQ1) through string diagrams (RQ2), distributional semantics (RQ3), enriched structure (RQ4), to topos-theoretic transfer (RQ5). **RQ6** provides the cognitive science meta-argument that threads through the entire paper, grounding diagrams in spatial reasoning and predictive processing. **RQ7** supplies computational validation of the framework via automated testing. **RQ8**–**RQ11** extend the framework into four application domains: quantum semantics (RQ8), cognitive security (RQ9), falsifiable neurolinguistic predictions (RQ10), and multi-agent AI protocols (RQ11). The conclusion (section 21) revisits each question with a summary of results and identifies eight open directions (**F1**–**F8**).

4 Case Systems: From Pāṇinian Kāraka to Cross-Linguistic Alignment Typology

Where we are in the argument. section 2 stated the central claim — that commutative diagrams are cognitively privileged because they encode algebra, distribution, and inference at once, with case as the pivot. Before we formalise case as a category in section 5, this chapter surveys the linguistic input the formalism must cover: five analytical traditions (Pāṇinian *kāraka*, Fillmorean deep case, Jakobsonian features, Mel'čukian dependency, Haspelmathian typology) and the five alignment typologies (nominative-accusative, ergative-absolutive, tripartite, active-stative, fluid-S) on which cross-linguistic variation converges.

4.1 Five Analytical Traditions That Shaped the Modern Theory of Case

4.1.1 The Pāṇinian Kāraka Framework

The formal study of grammatical case originates with Pāṇini (circa 4th century BCE), whose *Aṣṭādhyāyī*—a grammar of approximately 4,000 rules that predates Euclid—formalized the *kāraka* theory. This framework first classified semantic roles—agent, patient, instrument, and others—as deep relational functions linking verbs to their arguments, abstracting away from surface morphosyntactic inflections. Etymologically meaning "that which brings about" an action, the kāraka system establishes a rigorous mapping from conceptual predication to phonological realization [2021; 1987].

4.1.2 Jakobson's Structural Features and the Prague School

This profound semantic foundation lay predominantly dormant until structural linguists resurrected it in the mid-20th century. Roman Jakobson's *Morphologic Inquiry into Slavic Declension* (1958) decomposed grammatical cases into binary distinctive features (e.g., [±directional], [±peripheral]) [1958]. By treating case oppositions analogously to phonological features, Jakobson shifted the field from Pāṇini's holistic roles to a componential analysis that exposes deep relational hierarchies and markedness. Concurrently, Louis Hjelmslev's *La catégorie des cas* [1935] reinforced this functionalist tradition by formalizing case as a purely relational category within a dynamic semiotic network.

To make Jakobson's componential analysis tangible, consider the six-case singular paradigm of the Russian masculine inanimate noun *stol* "table": NOM *stol* / ACC *stol* / GEN *stolá* / DAT *stolú* / INS *stolóm* / PREP (LOC) *stolé*. The same root surfaces in six morphological guises whose contrasts Jakobson decomposes along the binary features `[±directional]` (the dative singles out goal-directed cases) and `[±peripheral]` (the instrumental and locative push out from the core argument cases). Serbian/BCS *prijatelj* "friend" runs a parallel paradigm but adds an overt vocative *prijatelju!* "friend!", giving a seven-way contrast that exercises the entire eight-role inventory we adopt in Table 4 (with the ablative absorbed into Slavic genitive-of-source and the prepositional/locative). These overt morphological contrasts will reappear in section 5 as concrete witnesses for the alignment functor and in section 11 as a calibration source for $[0, 1]$-enriched hom-values. Slavic languages — with their dense nominal morphology, productive derivation, and unambiguous case suffixes — are also a natural stress-test candidate for the Distributional Active Inference framework developed in section 16; the limitations discussion in subsection 16.7 returns to this point explicitly, noting that a Russian or Serbian/BCS sentence would deliver an information-theoretically sharper drop in posterior entropy than the German example currently shipped.

4.1.3 Fillmore's Deep Case and Generative Roots

Charles Fillmore's seminal "The Case for Case" (1968) built directly upon this structuralist lineage, explicitly translating these concepts into the burgeoning generative linguistics framework. Fillmore proposed *deep cases* (e.g., Agentive, Objective, Dative) as universal semantic primitives underlying surface syntax. Crucially, he argued that verbs assign underlying *case frames* which subsequently generate surface structures [1968]. This evolution transformed Pāṇini's kāraka into deep relational networks, permanently prioritizing universal semantics over language-specific morphology.

4.1.4 Mel'čuk's Meaning-Text Theory (MTT)

A distinct but parallel formalization emerged with Aleksandr Žolkovskij and Igor Mel'čuk's Meaning-Text Theory (MTT) [Žolkovskij and Mel'čuk, 1965, Mel'čuk, 1981, 1988]. Developed initially in Moscow, MTT models natural language as a rigorous, many-to-many correspondence between meanings (deep semantic representations) and texts (surface-phonological representations). The transition from meaning to text unfolds across a multi-level synthesis process: Deep Semantics, Deep-Syntactic, Surface-Syntactic, Morphological, and Phonological. At the deep-syntactic level, meaning is represented via *dependency trees* linking lexical units through dependency relations. The nodes are populated by *actants*—semantic roles closely aligning with thematic grids but strictly language-specific in their surface realization.

Crucially, MTT establishes that grammatical case is not directly semantic, but rather a final surface-morphological phenomenon governed by syntactic linearization and dependency constraints. *Lexical functions* map actants to surface structures, explicitly demonstrating that morphosyntactic inflection serves merely as the end-stage formal realization of deep semantic relations. By monotonically mapping meaning to text via hierarchical dependency graphs, MTT provided an exhaustive synthesis that presages our modern categorical formalizations mapping conceptual structures to syntactic types.

4.1.5 Dowty's Proto-Roles and Graded Topologies

Where MTT models the *mapping* from deep semantic roles to surface morphology via hierarchical dependency graphs, Dowty's contribution is to decompose the deep roles themselves into continuous, gradient primitives. Fillmore's deep cases serve as the direct precursors to modern *thematic role* theory. Dowty [1991] refined this paradigm by decomposing thematic roles into clusters of sentential entailments, yielding two *proto-roles*: the Proto-Agent (characterized by volitional involvement and causation) and the Proto-Patient (characterized by incremental themes and causal affectedness). This decomposition proves critical for our categorical formalization: it replaces discrete, named roles with a *graded* structural topology where role assignment is a matter of degree rather than kind. Consequently, morphisms in our cognitive case diagrams carry continuous weights $w \in [0, 1]$ representing the degree to which a noun phrase satisfies proto-role entailments—a design choice that yields the enriched category structure developed formally in section 11.

4.2 Alignment Typology: How Languages Group S, A, and P

Contemporary typological work reveals that the world's languages realize case systems according to a small number of *alignment types*—systematic patterns governing how the core arguments of transitive and intransitive clauses are grouped [Polinsky and Preminger, 2014, Blake, 2001, Haspelmath, 2009].

4.2.1 The Three Core Argument Primitives: S, A, P

The cross-linguistic comparison rests on three primitives:

Table 2: The three core argument primitives used in cross-linguistic case typology.

Symbol	Role	Definition
S	Sole argument of intransitive	"The child **sleeps**"
A	Agent-like argument of transitive	"**The child** broke the vase"
P	Patient-like argument of transitive	"The child broke **the vase**"

4.2.2 The Five Cross-Linguistic Alignment Types

The key insight from typological research is that languages differ in how they *group* these three roles for purposes of case marking, agreement, and other grammatical processes:

Table 3: Five alignment types and their grouping of the three core argument primitives.

Alignment	Grouping	Exemplar Languages
Nominative–Accusative	S = A ≠ P	English, Latin, Finnish, Russian
Ergative–Absolutive	S = P ≠ A	Basque, Dyirbal, Georgian (partly)
Active–Stative	S splits by agentivity	Lakhota, Guaraní, Eastern Pomo
Tripartite	S ≠ A ≠ P	Nez Perce, some Australian languages
Fluid-S	S marking varies by context	Bats (NE Caucasian), Acehnese

Slavic case morphology as a stress-test for the formalism. Russian (six cases) and Serbian/BCS (seven cases including a productive vocative) sit firmly in the nominative-accusative column of Table 3, yet their nominal paradigms exercise the *full* eight-role inventory of Table 4 in ways that English's collapsed paradigm hides. Two animacy-conditioned syncretisms in Russian masculine singular illustrate the point precisely:

- **Inanimate masculine — NOM = ACC syncretism.** *stol* "table" surfaces identically in subject and direct-object position (NOM *stol* / ACC *stol*; *Stol stoit* "the table stands" vs. *Vižu stol* "I see the table"). The surface-realisation functor $M \colon \mathcal{C}_{\text{grammatical}} \to \mathcal{C}_{\text{morphological}}$ identifies $\{\text{NOM}, \text{ACC}\}$ for this declension class — a *partial* kernel that coexists with full distinguishability for feminine and neuter paradigms.

- **Animate masculine — ACC = GEN syncretism.** *brat* "brother" takes ACC = GEN *brata* (*Vižu brata* "I see the brother", same form as the genitive *bez brata* "without the brother"). The morphology overtly licenses the canonical formalism's claim that the surface functor neutralises distinct grammatical objects on a *paradigm-by-paradigm* basis, not globally.

Serbian/BCS replicates the same animacy split (*čovek* / *čoveka* parallel to *brat* / *brata*) and adds a productive vocative (*prijatelju!* "friend!", *bože!* "God!") that English lacks entirely. These are exactly the empirical witnesses the alignment functor of section 5 predicts: where English shows a single inflectionless noun, Slavic morphology exposes a non-trivial kernel structure on a per-declension-class basis, supplying ready-made cross-linguistic targets for the categorical apparatus we develop next.

Fluid-S and Context-Dependent Functors. In Bats (Nakh-Daghestanian), the intransitive subject of a single verb surfaces in different cases strictly depending on the speaker's internal construal of agentive volition. For example, the verb *fall* assigns an absolutive S when the action is accidental (*The child-ABS fell*), but assigns an ergative S when the action is volitional (*The child-ERG fell [on purpose]*).

Categorically, we model Fluid-S as a **context-dependent functor** $F_\theta : \mathcal{U} \to \mathcal{L}$ parameterized by a continuous volition feature $\theta \in [0, 1]$. Figure 2 visualizes the resulting volition landscape: case categorization boundaries shift dynamically as a direct function of the agent's internal construal, satisfying naturality only up to a probabilistic reparameterization of θ.

Synthetic Case-Role Algebra. We introduce Synthetic Case-Role Algebra as a novel, computational upgrade to the Dowty-style proto-role framework. Where Dowty modeled proto-roles as static clusters of entailments [1991], we formalize them as **objects in a** $[0, 1]$**-enriched monoidal category** with tensor product over role compositions. This advancement enables purely algebraic manipulation of semantic roles: composition, weighting, and transformation of roles proceed through functorial operations representing complex event structures such as causativization, serial verb constructions, and argument-structure alternations. Crucially, this enriched structure demonstrates that "case assignment" is not a discrete binary choice but a vector-valued expectation in continuous case space—providing the mathematical bridge between the symbolic traditions of formal grammar and the statistical representations of modern neural language models (section 8).

Claassen [2019] surveys the explanatory frameworks proposed for alignment diversity, arguing that no single factor (processing efficiency, disambiguation, discourse pragmatics) suffices—a conclusion that motivates our multi-dimensional categorical formalization. Wu [2024] offers a detailed case study of Amis (Austronesian), demonstrating how verb classification, case marking, and grammatical relations interact in a language that defies simple alignment classification.

Beyond the three core arguments, languages distinguish a rich inventory of oblique cases. Our formalization follows the CEREBRUM framework [Friedman and Active Inference Institute, 2024] in adopting eight fundamental cases (Table 4):

Table 4: Eight fundamental case roles adopted from the CEREBRUM framework [Friedman and Active Inference Institute, 2024].

Case	Abbreviation	Semantic Core	Syntactic Prototype
Nominative	NOM	Agent / experiencer	Intransitive subject, transitive agent
Accusative	ACC	Patient / theme	Direct object, incremental theme
Genitive	GEN	Possessor / source	Possessive modifier, partitive
Dative	DAT	Recipient / goal	Indirect object, beneficiary
Instrumental	INS	Instrument / means	Adverbial of means
Locative	LOC	Location / context	Spatial/temporal ground
Ablative	ABL	Origin / cause	Source of motion, causal adjunct
Vocative	VOC	Addressee	Direct address

While historically used merely to diagram sentences, this exact eight-role inventory is what enables the **Categorical AI Protocol** introduced in section 19. By rigidly mapping artificial agent capabilities to corresponding grammatical cases—e.g., treating an API as strictly `INS`, passive data strictly as `ACC`, and system context rigidly as `LOC`—the cognitive case diagram enforces computational boundary constraints that formally constrain prompt injection attacks through explicit role typing (section 20).

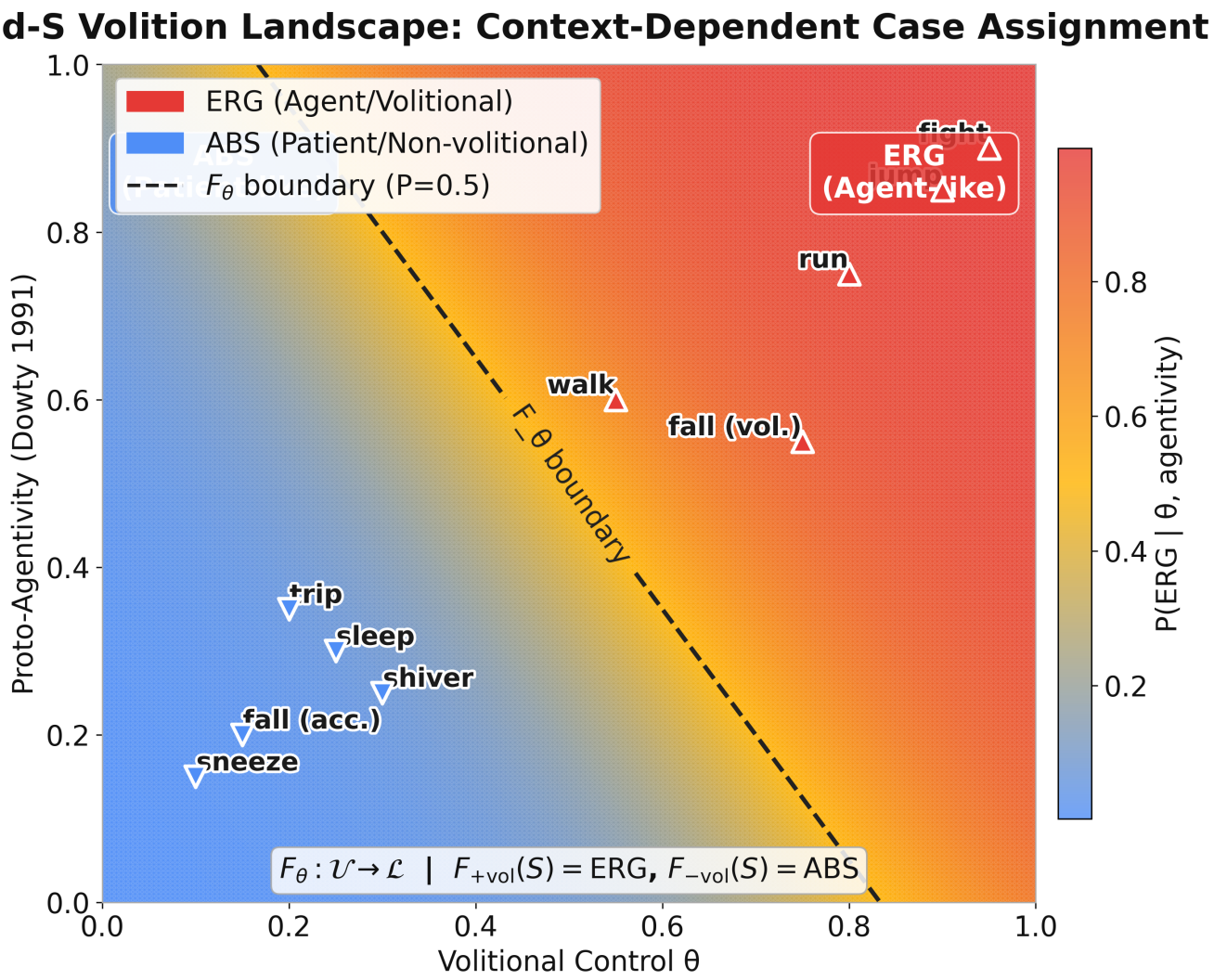

Figure 2: Fluid-S alignment is a continuous mapping, not a binary switch. The context-dependent functor $F_\theta : \mathcal{U} \to \mathcal{L}$ is rendered as a 2D decision surface with volitional control $\theta \in [0, 1]$ on the x-axis and proto-agentivity [Dowty, 1991] on the y-axis. Color intensity represents $P(\text{ERG} \mid \theta, \text{agentivity})$, computed via a logistic boundary. Nakh-Daghestanian verb exemplars (Bats language) are overlaid at their typologically attested coordinates: low-volition actions (sneeze, accidental fall) cluster in the ABS region; high-volition actions (jump, fight) occupy the ERG region. The dashed curve marks the functor decision boundary where $F_\theta(S)$ transitions from ABS to ERG. Generated programmatically from `src.visualization.fluid_s_plots.plot_fluid_s_volition_landscape()`.

5 Case Categories: Roles as Objects, Relations as Morphisms, Alignment as Functors

Where we are in the argument. section 4 surveyed the cross-linguistic input data (five traditions, five alignment types). This chapter converts that data into the first formal object of the framework: a category whose objects are case roles, whose morphisms are grammatical relations, and whose alignment typologies are structure-preserving functors between categories — the layer on which every subsequent pillar (syntax, semantics, enrichment, cognitive inference) is built.

5.1 Eight Case Roles as Objects

We define a **case category** $\mathcal{C}$ as a small category where:

- **Objects** are case roles (NOM, ACC, GEN, DAT, INS, LOC, ABL, VOC)
- **Morphisms** are grammatical relations between roles (e.g., "transitive action": NOM $\to$ ACC)
- **Identity morphisms** represent the reflexive relation of each case role to itself
- **Composition** models the transitivity of grammatical dependencies

This formalization is implemented in our `CaseCategory` class, which uses set-based object tracking and list-based morphism storage as the underlying representation. Each object carries its role enum and optional morphosyntactic features; each morphism carries a relation label and an enriched weight $w \in [0, 1]$. Figure 3 shows the full eight-case standard category.

5.2 Accusative vs. Ergative as Structurally Non-Isomorphic Functors

An **alignment functor** $F : \mathcal{U} \to \mathcal{L}$ formally maps a universal case category $\mathcal{U}$ to a language-specific category $\mathcal{L}$ by systematically collapsing objects that the target language treats as equivalent. For example, in an accusative language, the functor forcibly merges S and A into a single NOM role while preserving P as a distinct ACC role: $F(\text{S}) = F(\text{A}) = \text{NOM}$, $F(\text{P}) = \text{ACC}$.

Mathematically, this functor guarantees three properties:

- **Surjective on objects**: Every target case role is the explicit image of a universal role.
- **Structure-preserving**: Grammatical relations in $\mathcal{U}$ strictly map to relations in $\mathcal{L}$.
- **Diagnostic kernels**: The kernel of F (the set of objects mapping to the identical target) uniquely identifies the language's alignment typology.

Thus, the alignment functor formalizes linguistic neutralization with mathematical precision: two semantically distinct roles receive identical morphological treatment precisely because the functor collapses them into a single object in the target category.

Explicit functor construction. Let $\mathcal{U}$ be the universal three-role category with objects $\{S, A, P\}$ and morphisms $f\colon A \to P$ (transitive action), $g\colon S \to S$ (intransitive). The accusative functor $F_{\text{acc}}\colon \mathcal{U} \to \mathcal{L}_{\text{acc}}$ and ergative functor $F_{\text{erg}}\colon \mathcal{U} \to \mathcal{L}_{\text{erg}}$ act on objects as:

$$F_{\text{acc}}(S) = F_{\text{acc}}(A) = \text{NOM}, \quad F_{\text{acc}}(P) = \text{ACC} \tag{1}$$

$$F_{\text{erg}}(S) = F_{\text{erg}}(P) = \text{ABS}, \quad F_{\text{erg}}(A) = \text{ERG} \tag{2}$$

On morphisms, each functor strictly preserves the transitive relation: $F_{\text{acc}}(f) = f'\colon \text{NOM} \to \text{ACC}$ and $F_{\text{erg}}(f) = f''\colon \text{ERG} \to \text{ABS}$. Crucially, the *kernel* of F_{acc}—the exact set $\{(X, Y) \mid F_{\text{acc}}(X) = F_{\text{acc}}(Y)\}$—resolves to $\{(S, A)\}$, formally encoding the syntactic identification of the intransitive subject with the transitive agent. Conversely, the kernel of F_{erg} resolves to $\{(S, P)\}$. This topological kernel structure successfully supplies a highly compact, computable algebraic fingerprint for every known alignment type.

Figure 4 shows three alignment systems rendered from our `CaseCategory` implementation.

5.3 Graded Proto-Roles as $[0, 1]$-Weighted Morphisms

Following Dowty [1991], we equip morphisms with weights in $[0, 1]$ that encode the degree of proto-role satisfaction. A morphism $f : \text{NOM} \to \text{ACC}$ with weight $w = 0.9$ indicates a strong transitive action (clear agent acting on clear patient), while $w = 0.4$ might indicate an experiencer construction ("The child fears the dark") where the nominative argument only weakly satisfies Proto-Agent entailments.

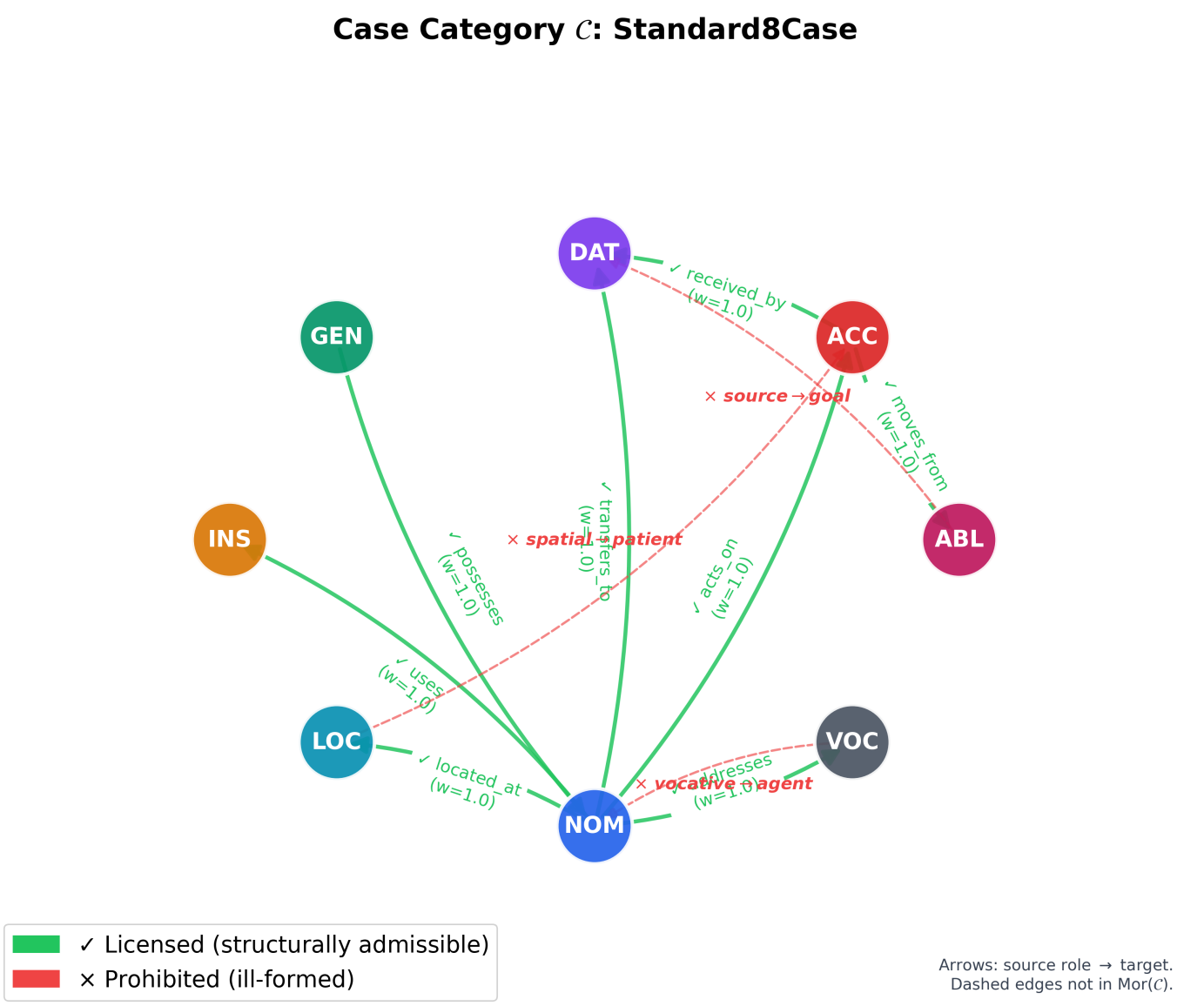

Figure 3: Eight case roles form a directed graph with weighted grammatical morphisms. The standard linguistic case category $\mathcal{C}$ with objects NOM, ACC, GEN, DAT, INS, LOC, ABL, VOC and directed morphisms encoding grammatical relations. Edge labels identify relation types (acts_on: NOM→ACC, possesses: GEN→NOM, received_by: ACC→DAT, located_at: NOM→LOC); edge weights $w \in [0, 1]$ reflect proto-role satisfaction per Dowty's [1991] decomposition. The enriched structure over $([0, 1], \cdot, 1)$ ensures multiplicative weight attenuation under composition (Equation 3). Generated programmatically from the `CaseCategory` class.

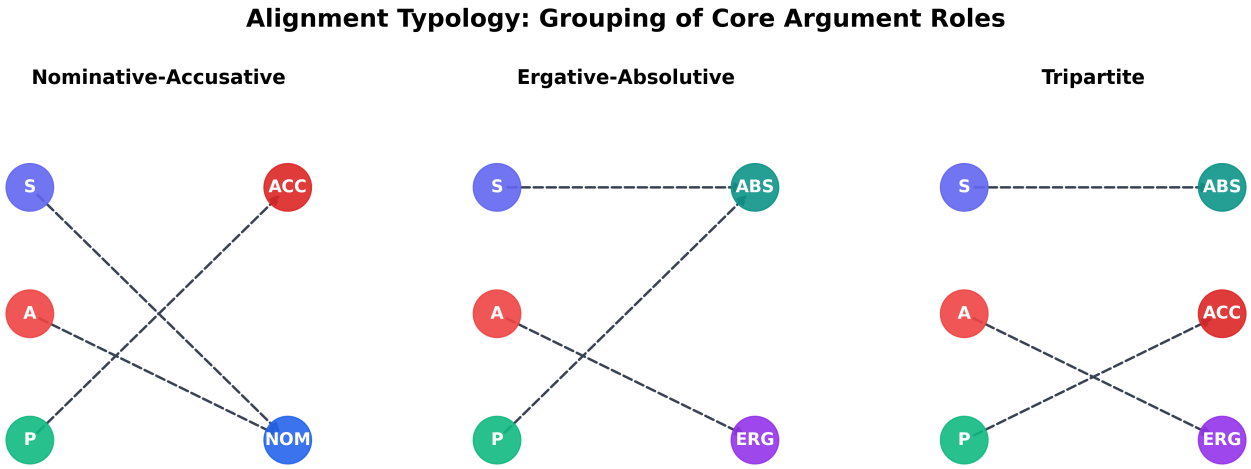

Figure 4: Functor kernels uniquely fingerprint each alignment typology. Three alignment systems realized as functors from $\mathcal{U} = \{S, A, P\}$: **Nominative–Accusative** merges $\{S, A\} \to$ NOM (kernel $\{(S, A)\}$, Equation 1); **Ergative–Absolutive** merges $\{S, P\} \to$ ABS (kernel $\{(S, P)\}$, Equation 2); **Tripartite** is injective (kernel $\emptyset$). Color-coded nodes reveal neutralization patterns: shared colors indicate functor identification of roles. Generated programmatically from the `CaseCategory` implementation.

Slavic quirky case as overt evidence for sub-unit morphism weights. Russian and Serbian/BCS supply a sharper, *morphologically overt* witness for $w < 1$: a class of verbs that systematically refuses the canonical NOM $\to$ ACC arrow and instead routes its object through GEN, DAT, or INS. In Russian, *boyat'sja* "to fear" governs the genitive (*boyus' sobaki* "I fear the dog-GEN", not **sobaku*-ACC); *pomogat'* "to help" governs the dative (*pomogayu drugu* "I help the friend-DAT"); *upravlyat'* "to manage / steer" governs the instrumental (*upravlyaet mašinoj* "drives the car-INS"). Serbian/BCS shows the same pattern: *čestitati* "to congratulate" assigns DAT (*čestitam prijatelju* "I congratulate the friend-DAT"); *bojati se* "to fear" assigns GEN (*bojim se psa* "I fear the dog-GEN"). Each such verb supplies an enriched morphism whose target is *not* ACC, equivalently a NOM $\to$ ACC arrow whose Dowtian weight has been reduced to near zero — and the reduction is visible in the suffix on the noun, not merely hypothesised from semantics. These quirky-case lexemes are the cleanest empirical anchor for the graded $[0, 1]$-enrichment we develop in section 11.

Composition of enriched morphisms multiplies weights:

$$w(g \circ f) = w(g) \cdot w(f) \tag{3}$$

This multiplicative composition reflects the intuition that grammatical dependencies attenuate as they chain through intermediate roles. Figure 5 illustrates the categorical composition of two morphisms through an intermediate case role. The resulting structure is a category enriched over $([0, 1], \cdot, 1)$—a connection we develop fully in section 11.

5.4 Alignment Shifts as Natural Transformations: Functor Commutativity Encodes Grammar Agreement

Having established that alignment systems are functors $F, G : \mathcal{U} \to \mathcal{L}$ from a universal case category to language-specific categories, a natural question arises: *how do different alignment systems relate to each other?* The categorical answer is a **natural transformation** $\alpha : F \Rightarrow G$—a systematic family of morphisms $\alpha_A : F(A) \to G(A)$ for each case role A, satisfying the **naturality condition**:

$$G(f) \circ \alpha_A = \alpha_B \circ F(f) \quad \text{for every morphism } f : A \to B \tag{4}$$

This naturality constraint is the formal expression of *grammatical coherence*: transforming one alignment into another and then applying a grammatical relation yields the same result as first applying the relation and then transforming—ensuring that alignment comparison respects the relational fabric of the grammar.

Worked example. Consider the accusative functor F (which maps S and A to NOM, P to ACC) alongside the tripartite functor G (which maps explicitly to S $\to$ ABS, A $\to$ ERG, P $\to$ ACC). The **identity natural transformation** id_F :

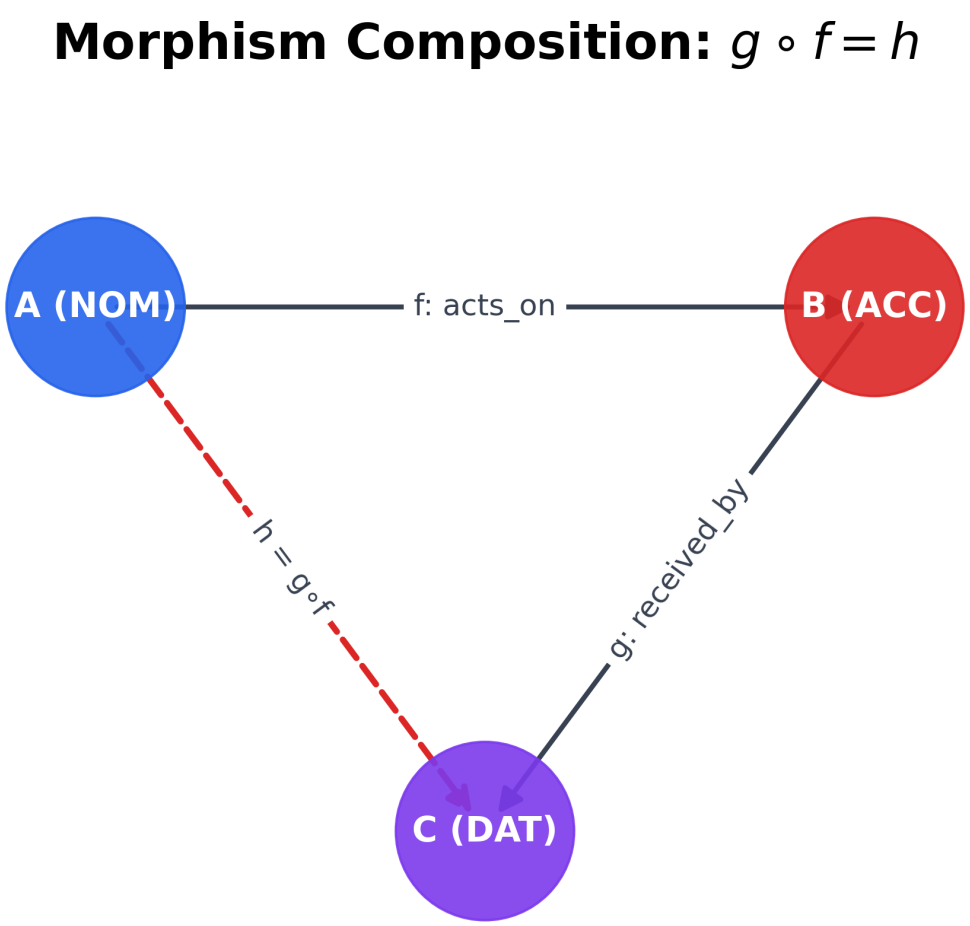

Figure 5: Morphism composition attenuates weights multiplicatively through intermediate case roles. Morphism f: NOM $\to$ ACC (acts_on) and g: ACC $\to$ DAT (received_by) compose to $h = g \circ f$: NOM $\to$ DAT per Equation 3; in the enriched category weights multiply (e.g., $w_f = 0.9$, $w_g = 0.7 \Rightarrow w(g \circ f) = 0.63$). The commutative triangle encodes that DAT assignment factors through ACC—the multiplicative attenuation reflects the typological observation that subject–recipient relations are weaker than the constituent subject–object and object–recipient links. Generated programmatically from the `CaseCategory` class.

$F \Rightarrow F$ provides components $(\mathrm{id}_F)_A = \mathrm{id}_{F(A)}$ over every role A, trivially satisfying the naturality condition. We construct the **vertical composition** $\beta \circ \alpha$ of two natural transformations $\alpha : F \Rightarrow G$ and $\beta : G \Rightarrow H$ purely componentwise: $(\beta \circ \alpha)_A = \beta_A \circ \alpha_A$.

Our `NaturalTransformation` class implements these operations, with `ComponentMorphism` objects encoding each α_A, and `compose_transformations()` implementing vertical composition. The `IdentityNaturalTransformation` constructor automatically generates identity components for every object in an `AlignmentFunctor`'s domain. This machinery provides the formal infrastructure for comparing alignment types not merely by listing their neutralization patterns but by characterizing the *structural mappings* between them—e.g., the natural transformation from accusative to tripartite alignment is injective (no two roles merge in the target), while the transformation from tripartite to ergative is non-injective (S and P merge into ABS).

6 Categorial Grammar: Syntax as Algebraic Composition and Proof

Where we are in the argument. section 5 built the case category — a static snapshot of licensed role-to-role morphisms. This chapter equips the framework with a *compositional syntax* on top of that static layer: Lambek pregroup types let each word carry its own combinatory potential, and cup/cap contractions reduce a parallel tensor of word-boxes to the single sentence type s, so a sentence becomes a proof of well-typedness rendered graphically as a string diagram.

6.1 Each Word Is Its Own Grammar Rule

Traditional generative grammar computes sentence structure using top-down phrase-structure rules that recursively combine constituents. Categorial grammar inverts this perspective: rather than specifying abstract construction rules, it assigns each lexical item a dedicated *type* that encodes its combinatory potential. For example, a transitive verb like "chases" is not loosely defined as "a word requiring a subject and object," but specifically assigned the algebraic type $(n\backslash s)/n$—an active function that categorically demands a noun to its right (the object) and a noun to its left (the subject) to yield a complete sentence.

6.2 Lambek's Residuation Law: Consuming and Producing Types in Linear Order

Lambek [1958] laid the algebraic foundations by introducing a *residuated* structure on syntactic types. He defined two binary operations—the left residual $\backslash$ and the right residual $/$—derived from the multiplicative connective $\otimes$ (concatenation). The fundamental axiom governing this system is the residuation law:

$$A \otimes B \leq C \quad \iff \quad A \leq C/B \quad \iff \quad B \leq A\backslash C \tag{5}$$

This law brilliantly captures the fundamental duality of syntax: a verb of type $(n\backslash s)/n$ actively *consumes* available noun arguments to *produce* a well-formed sentence. The resulting **Lambek calculus** operates as a non-commutative intuitionistic linear logic—non-commutative because strict word order dictates meaning, and linear because the derivation logically consumes each lexical resource exactly once.

6.2.1 Pregroups Unlock Compact Closure: The Bridge to DisCoCat String Diagrams

Lambek [2004] later simplified the framework to **pregroup grammars**, where each type a has a left adjoint a^l and a right adjoint a^r satisfying:

$$a^l \cdot a \leq 1 \leq a \cdot a^l \qquad a \cdot a^r \leq 1 \leq a^r \cdot a \tag{6}$$

Coecke, Sadrzadeh, and Clark revolutionized this formal structuralism in 2010 with the **DisCoCat** (Distributional Compositional Categorical) model [2010]. DisCoCat mathematically unifies categorial grammar with distributional semantics by defining strong monoidal functors mapping pregroup grammars directly into vector spaces. As Duneau emphasizes, this approach "allows the meaning of a sentence to be computed as a function of both the distributional meaning of the words involved, as well as its grammatical form" [2021]. By proving that pregroups and finite-dimensional vector spaces both behave as rigid monoidal categories, DisCoCat allows us to compute whole-sentence vector meanings linearly from constituent word vectors.

In this system, grammaticality reduces to algebraic verification: checking whether a sequence of types maps to the sentence type s via *contraction* ($a^l \cdot a \to 1$) and *expansion* ($1 \to a \cdot a^l$) operations. This reformulation unlocks the DisCoCat framework (section 8) because pregroups function as **compact closed categories**, natively supporting a computationally sound diagrammatic calculus. When a derivation is ill-typed, contractions fail to close: there is no valid reduction to s. section 20 develops the security reading of that failure mode—typed interaction boundaries and prompt injection as illicit role reassignment.

6.3 Cups, Caps, and Joyal–Street: How Wires Prove Grammaticality

The pivotal connection between categorial grammar and visualization comes from Joyal and Street [1991], who proved that morphisms in monoidal categories can be faithfully represented as *string diagrams*—planar graphs where:

- **Wires** represent types (objects of the category)
- **Boxes** represent words or operations (morphisms)
- **Vertical composition** represents sequential application
- **Horizontal juxtaposition** represents the tensor product (concatenation)

- **Cups and caps** represent the contraction/expansion of a pregroup

Figure 7 surveys three syntactic structures side by side (progression from intransitive to passive); Figure 6 shows the same transitive sentence rendered by our native matplotlib pipeline—a direct computational output confirming word-box layout, wire routing, and cup-contraction geometry independently of DisCoPy; and Figure 8 demonstrates that type-logical structure remains invariant across diverse surface word orders (SVO, free, SOV).

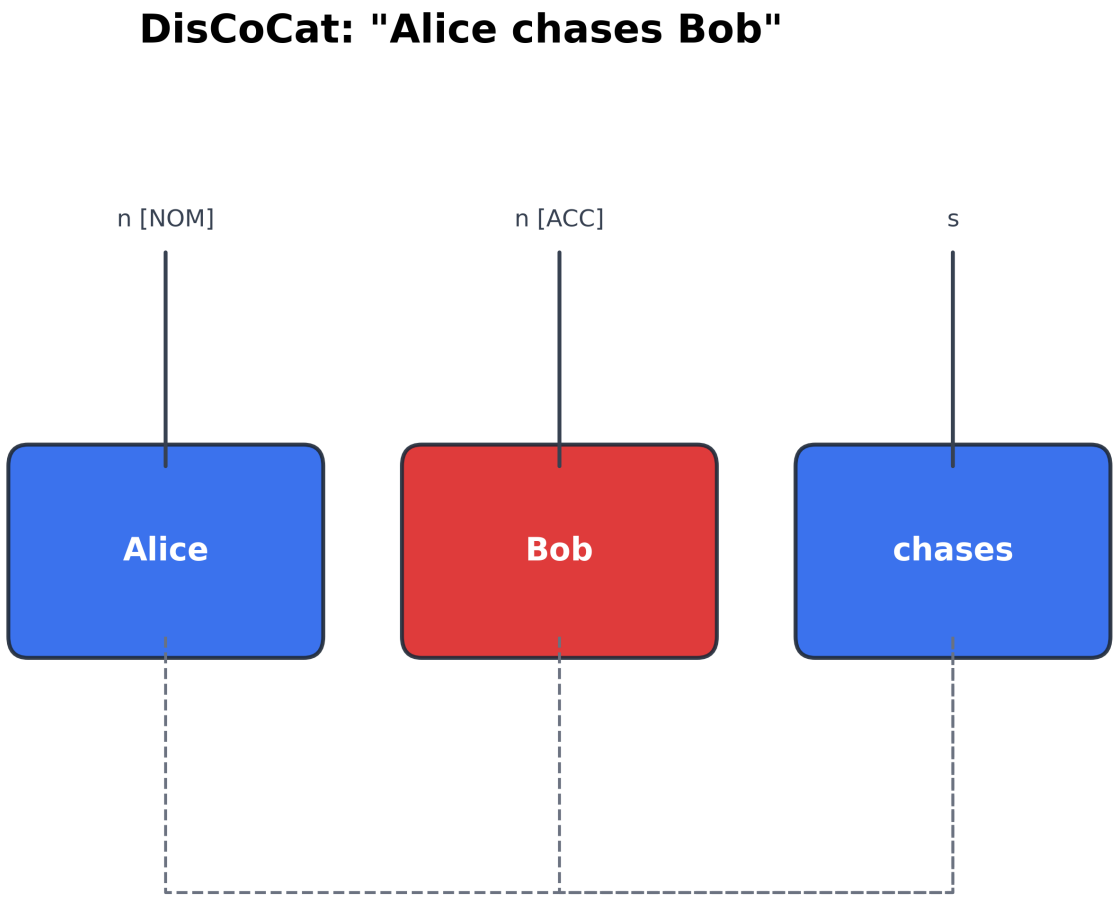

Figure 6: Native matplotlib rendering cross-validates DisCoPy pregroup geometry. String diagram for "Alice chases Bob" generated via `src.visualization.string_diagrams.render_discocat_sentence()` without the DisCoPy library. Three Word boxes carry pregroup types: n (Alice, **blue** = NOM), $n^r \otimes s \otimes n^l$ (chases, **neutral**), and n (Bob, **red** = ACC). Dashed arcs trace cup connections from noun wires to the verb's adjoint slots, rendering $\varepsilon : n^r \otimes n \to 1$ and $\varepsilon : n^l \otimes n \to 1$ as paired ligatures. This confirms that our visualization pipeline correctly reconstructs pregroup string-diagram geometry using only case-role metadata from `src.case_systems.case_category`, cross-validating the DisCoPy canonical output in Figure 9.

The Latin point in Figure 8 generalises directly to highly inflected modern Slavic. Russian *Sobaka kusaet čeloveka* "the dog bites the man" and its scrambled counterpart *Čeloveka kusaet sobaka* differ in linear order but share the reduction $n_{\text{NOM}} \otimes (n^r \otimes s \otimes n^l) \otimes n_{\text{ACC}} \to s$, with the case suffixes (-a NOM.SG.F vs -a ACC.SG.M.ANIM after stem hardening) supplying the role information that English encodes positionally. Serbian/BCS exhibits the same robustness: *Pas ujeda čoveka* / *Čoveka ujeda pas* / *Pas čoveka ujeda* (and three further permutations) all type-check to s because the morphological case marking on the noun wires uniquely identifies which is NOM and which is ACC. In Slavic languages the type calculus is therefore not just *consistent* with the surface order — the surface order is, to a first approximation, *immaterial*, with case morphology carrying the entire wiring diagram explicitly.

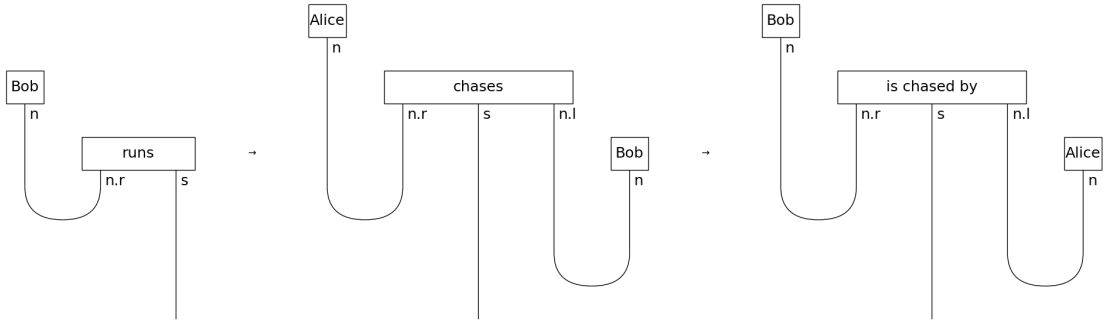

Figure 7: Valency determines diagram topology: Cup count tracks verb argument structure. A 1×3 DisCoPy pregroup grammar visualization of increasing argument complexity. **Panel (a)** intransitive "Bob runs" (NOM subject, 1 Cup contraction, type $n \otimes (n^r \otimes s) \to s$); **Panel (b)** transitive "Alice chases Bob" (NOM + ACC, 2 Cup contractions, type $n \otimes (n^r \otimes s \otimes n^l) \otimes n \to s$); **Panel (c)** passive "Bob is chased by Alice" (patient promoted to NOM, former agent demoted to an oblique *by*-phrase; full transitive-shape verb type $n^r \otimes s \otimes n^l$ with 2 Cup contractions — *which* noun enters each cup is reversed relative to panel (b)). The role reassignment is visible as a topological swap of which noun lands in the left (subject) cup and which in the right (oblique) cup; cup count itself encodes argument-slot count, not syntactic prominence. Complexity metric $\kappa(D)$ per Equation 12.

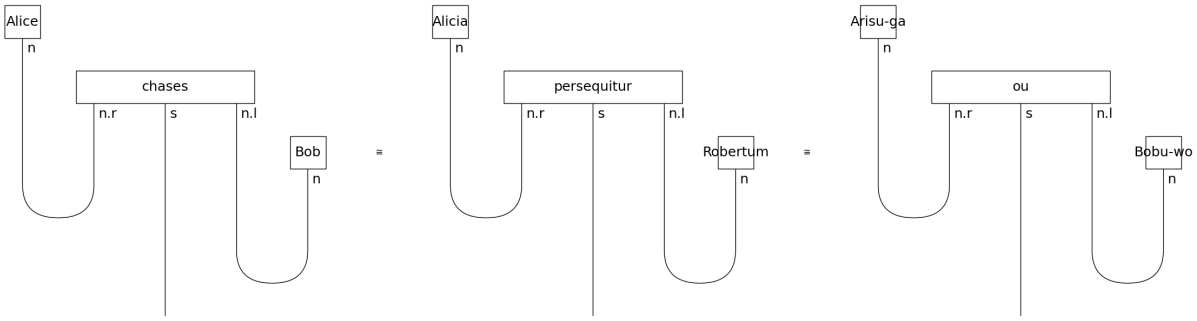

Figure 8: Type-logical structure is invariant under surface word-order permutation. Pregroup derivations for "SUBJ chases OBJ" across three typologically diverse languages: English (SVO), Latin (inflected, free order), and Japanese (agglutinative, SOV). Each renders an identical type reduction $n \otimes (n^r \otimes s \otimes n^l) \otimes n \to s$, confirming the central claim of categorial grammar: syntactic universals reside in the type algebra, not in linear word order.

This result—the soundness and completeness of string-diagrammatic reasoning—formally guarantees that any topological conclusion drawn visually from the diagram is algebraically valid: the diagram is not a heuristic but a rigorous proof instrument.

This profound visual transparency instantiates exactly the "free ride" phenomenon identified by Shimojima [1996]: a single diagram simultaneously exposes the syntactic derivation, the deep argument structure, and the compositional flow of semantic meaning—entirely eliminating the need for sequential, explicit inference steps.

Concrete derivation in DisCoPy. The type reduction for "Alice chases Bob" is computationally verified using the DisCoPy library's `discopy.rigid` module:

```
from discopy.rigid import Ty, Box, Cup, Id
n, s = Ty('n'), Ty('s')
alice = Box('Alice', Ty(), n)
chases = Box('chases', Ty(), n.r @ s @ n.l)
bob  = Box('Bob',  Ty(), n)
words = alice @ chases @ bob          # n ⊗ (n.r ⊗ s ⊗ n.l) ⊗ n
cups  = Cup(n, n.r) @ Id(s) @ Cup(n.l, n)
diagram = words >> cups               # reduces to s
assert diagram.cod == s               # type-checks: sentence type
```

The two `Cup` operations successfully contract n with n^r (resolving the subject–verb link) and n^l with n (resolving the verb–object link). This topological reduction collapses the five-wire tensor product into a single, valid sentence wire s. Consequently, the assertion `diagram.cod == s` constitutes a direct, machine-processable proof of grammaticality.

From pregroup types to graded types. The standard pregroup derivation produces a rigid, binary grammaticality judgment: the final codomain of the fully contracted tensor product either matches the sentence type s (grammatical) or fails (ill-formed). However, we can *grade* this binary verdict by replacing the underlying Boolean algebra $\{0, 1\}$ with the continuous unit interval $[0, 1]$. This substitution yields a graded type theory where type judgments carry continuous confidence weights rather than truth values. Asudeh and Giorgolo [2020] developed this approach using a monadic semantics that wraps base types inside a computational effect tracking epistemic uncertainty—an idea whose categorical content is precisely a change of enrichment base.

Mapped into our framework, this operation corresponds to the $[0, 1]$-enrichment detailed in section 11: the enriched scalar weight on any morphism $f : A \to B$ quantifies the confidence that the categorical case assignment $A \to B$ remains well-typed. From there, categorical magnitude aggregates these sparse, local confidence scores into a global complexity measure evaluating the entire syntactic derivation. The progression from rigid pregroup strings, through graded types, into fully enriched case categories operates mathematically as a cascade of base-change functors ($\mathbf{Bool} \hookrightarrow [0, 1] \hookrightarrow \mathbf{R}_{\geq 0}$)—where each successive level grants greater representational nuance while demanding higher computational complexity.

For readers encountering pregroup reduction for the first time, Figure 10 *unpacks* the same construction pedagogically: the four panels walk from the raw word types (panel 1), through the parallel tensor that juxtaposes the boxes (panel 2), through the application of the two cup contractions ε_n that cancel each argument wire against its adjoint (panel 3), to the normal form in which only the sentence wire s survives (panel 4). Each panel is coloured by case role (NOM = blue, ACC = red, verb output in indigo) so the identity of each wire is unambiguous.

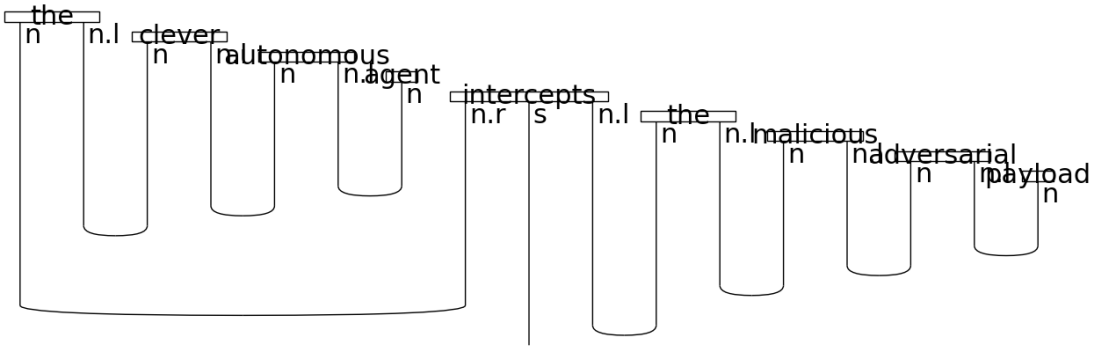

Figure 9: Direct DisCoPy machine output confirms pregroup type reduction to sentence wire s. Nine Word boxes carry complex noun-phrase types (e.g., $n \otimes n^l$ for adjectives mapping an adjacent noun to a modified noun) and core verb types ($n^r \otimes s \otimes n^l$ for the transitive "intercepts"). Eight Cup contractions sequentially resolve the argument scopes from the deepest noun phrases outward, ultimately canceling all local adjoint pairs and reducing the extended initial tensor product to the single sentence wire s. Unlike the schematic progression in Figure 7, this is the *direct computational output* of a multi-word discourse component, strictly confirming `diagram.cod == s`. By the Curry–Howard correspondence, this diagram simultaneously constitutes a syntactic derivation, a proof of well-typedness, and (under the meaning functor of section 8) a compositional semantic computation. Our extension decorates n-typed wires with functorial states $S : \mathbf{Ent} \to \mathbf{Case}$, tracking role assignments across discourse boundaries via **DisCoCirc entity wires**.

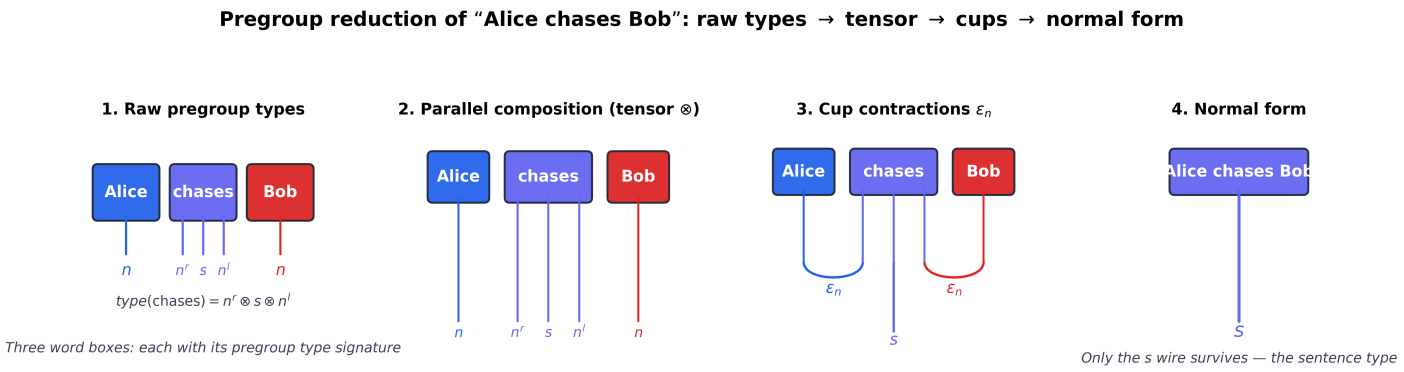

Figure 10: Pregroup reduction unpacked in four panels. Panel 1: raw pregroup types for *Alice* (n), *chases* ($n^r \otimes s \otimes n^l$), *Bob* ($n$). Panel 2: parallel composition $\otimes$ exposes all five dangling wires. Panel 3: the two cups ε_n cancel Alice's n against the verb's n^r and the verb's n^l against Bob's n, leaving only the central s wire. Panel 4: normal form — the entire sentence reduces to a single s-typed wire. Generated programmatically from `src.visualization.category_unpacking.render_pregroup_reduction_unpacking()`.

7 Case Subscripts, Passivization, and the Curry–Howard Proof

Where we are in the argument. section 6 installed pregroup syntax — cups, caps, and compact closure — as the algebraic backbone of composition. This chapter decorates that backbone with case information: every noun wire picks up a case subscript (n_{NOM}, n_{ACC}, …) so grammatical role becomes a first-class typed structure on wires, and passivization becomes a wire-level swap together with **case relabeling** (Equation 7/Equation 8) rather than an exception to the type system — the Curry–Howard image is a proof rewrite of that rearrangement.

Case-marked noun phrases receive compound types that encode both their grammatical role and their combinatory potential. For example, in a nominative–accusative language, the type assignment proceeds as follows (Table 5):

Table 5: Case-marked pregroup type assignments for a standard nominative-accusative transitive clause.

Expression	Type	Gloss
"Alice" (subject)	n_{NOM}	Noun, nominative-marked
"the ball" (object)	n_{ACC}	Noun, accusative-marked
"kicks"	$(n_{\text{NOM}}\backslash s)/n_{\text{ACC}}$	Seeks ACC right, NOM left
"to Bob" (recipient)	$(s\backslash s)/n_{\text{DAT}}$	Modifies sentence via DAT argument

The specific subscripts structurally refine the base noun type n with mandatory continuous case features, mathematically guaranteeing that the transitive verb "kicks" exclusively selects a nominative-marked subject and an accusative-marked object. Any feature mismatch automatically blocks the algebraic derivation, accurately modeling native ungrammaticality.

Alignment and natural transformations. Cross-linguistic alignment is modeled functorially in section 5: alignment types are structure-preserving functors from a universal role category to a language-specific case category, compared via natural transformations when one asks how two alignments cohere on the same underlying argument structure. At the level of case-marked pregroup types (this section), that functorial picture **suggests** treating agreement checks as naturality-style coherence constraints: the morphosyntactic alignment chosen for arguments must remain compatible with the verb's combinatorial requirements so that reductions still factor through the intended case-typed slots. We do not claim a full adjunction between identity and alignment functors in the pregroup category here; the precise categorical status of alignment is anchored in section 5, while the typed reductions below supply the concrete syntax.

7.1 Syntactic Derivation as Proof, Case Assignment as Type Inference

A deep structural parallel underlies the Lambek calculus: derivations of syntactic types correspond to proofs in intuitionistic logic (via the Curry–Howard isomorphism), and therefore to programs in a typed lambda calculus, as summarized in Table 6.

Table 6: The Curry–Howard isomorphism connecting syntactic, logical, and computational domains.

Syntactic Domain	Logical Domain	Computational Domain
Types ($A/B, A\backslash B$)	Propositions	Types
Derivations	Proofs	Programs (λ-terms)
Cut elimination	Proof normalization	β-reduction
Commutativity of cuts	Confluence of rewriting	Church–Rosser property

This foundational correspondence dictates two critical consequences for our cognitive framework:

1. **Semantic compositionality strictly follows syntactic well-formedness**: A successfully typed derivation automatically guarantees a well-formed geometric meaning representation (specifically, the exact λ-term isolated and extracted directly from the formal proof).
2. **Case assignment reduces to type inference**: Dynamically determining the correct case for a noun phrase in context is computationally equivalent to inferring the type of an unknown variable in a λ-expression—thereby grounding case assignment in the well-understood algorithmic landscape of type inference (Hindley-Milner and its extensions).

7.2 Song's Monadic Root Syntax: Sublexical Decomposition via Embedded Monads

Song [2022a; 2022b] significantly extends this categorial framework by deploying a novel **monadic semantics** for root syntax. He successfully models the sublexical decomposition of complex verb roots by embedding specialized monads directly into the base syntactic category. This computational approach captures the deep intuition that a seemingly simple verb like "break" harbors layered internal structure—specifically, an active causative layer tightly coupled to a passive result-state layer—which dynamically dictates subsequent case assignment. The formal monad elegantly encapsulates this tangled lexical complexity within one streamlined categorical construction, empowering the type-logical grammar to process lexical features and syntactic composition simultaneously and uniformly.

This monadic architecture seamlessly interfaces with modern graded type theory. For instance, Asudeh and Giorgolo [2020] previously developed a monadic semantics tracking evidentiality, deploying graded computational effects to quantify the shifting epistemic real-world status of various propositions. Our framework extends this exact pattern into the domain of morphological case, where the continuous "grade" attached to any specific morphism formally encodes the statistical strength and validity of that local semantic role assignment.

7.3 Passivization Is a Swap: Voice Alternation as Topological Wire Crossing

Within categorical linguistics, **passivization** emerges as a uniquely revealing topological operation. Traditionally described as a syntactic rule that promotes the patient to subject position while (optionally) demoting the agent to an oblique role, passivization in our framework ceases to be an ad-hoc transformation; instead, it combines (i) a **Swap** morphism $\sigma_{A,B} : A \otimes B \to B \otimes A$ that crosses the noun wires feeding into the verb's pregroup derivation—so *which* noun meets the verb's left vs. right adjoint slots reverses relative to the active diagram—and (ii) **updated case features** on those wires: the promoted subject is n_{NOM}, not a carried-over n_{ACC}.

In a pregroup grammar, the active transitive "Alice chases Bob" has the type reduction:

$$n_{\mathrm{NOM}} \cdot (n^r \cdot s \cdot n^l) \cdot n_{\mathrm{ACC}} \to s \tag{7}$$

For "Bob is chased by Alice," Bob is the grammatical subject (n_{NOM}) and Alice appears in the oblique *by*-phrase (n_{OBL}), with surface order subject–verb–oblique:

$$n_{\mathrm{NOM}} \cdot (n^r \cdot s \cdot n^l) \cdot n_{\mathrm{OBL}} \to s \tag{8}$$

This is **not** the naive swap of subscripts in (7) (which would incorrectly leave the promoted patient typed as accusative). The DisCoPy `Swap` primitive makes the wire crossing explicit; the case-marked types in (8) match the figure and standard promotion/demotion. The generated passive diagram diverges from its active counterpart through that crossing together with the reassignment of which noun occupies each cup. This transforms abstract syntactic voice alternation into a highly visible, instantly readable topological feature embedded directly within the string diagram.

This diagrammatic transparency exemplifies Shimojima's [1996] free-ride inference capability. By inspecting the visual topology, a cognitive agent immediately verifies that passivization preserves the verb's inherent argument structure while rearranging its surface realization—a structural fact that typically requires chaining inference steps to algebraically establish within standard linear notation (Figure 11).

Relative to Equation 12, the passive "Bob is chased by Alice" diagram in Figure 11 has the same cup count as the transitive active and therefore nearly identical $\kappa(D)$; the syntactic reassignment is visible in *which* cup each noun enters, not in the total count of contractions.

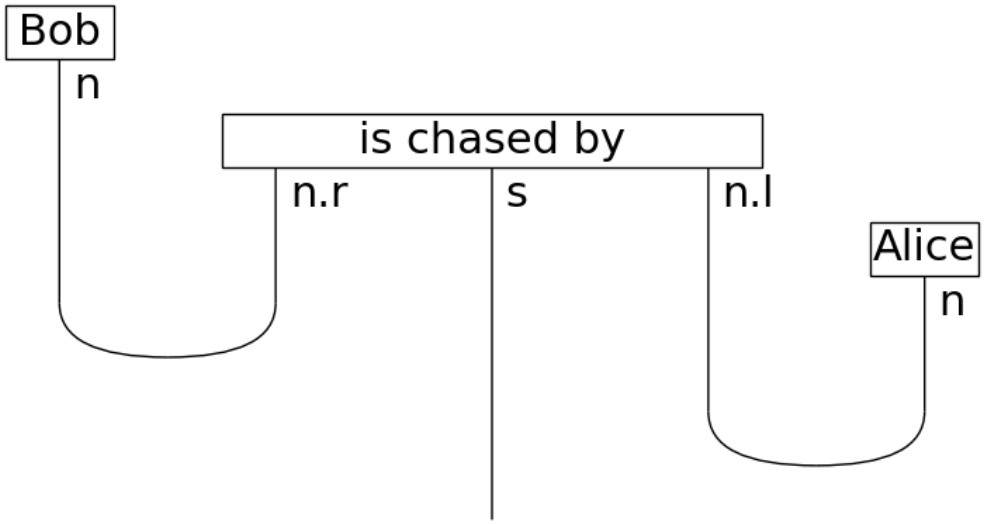

Figure 11: Passivization as role reassignment in the DisCoPy pregroup diagram for "Bob is chased by Alice" with type $n_{\mathrm{NOM}} \otimes (n^r \otimes s \otimes n^l) \otimes n_{\mathrm{OBL}} \to s$. Bob's n wire contracts into the verb's left adjoint n^r via the left Cup (n–n^r) — promoting the semantic patient to grammatical subject — while Alice's n wire contracts into the right adjoint n^l via the right Cup (n^l–n), now demoted to the oblique *by*-phrase. Compare with Figure 12 (active "Alice chases Bob"), where Alice occupies the left subject slot and Bob the right object slot: the reassignment of *which* noun lands in each Cup is what makes voice alternation a topological swap $\sigma_{n,n} : n \otimes n \to n \otimes n$ rather than a lexical substitution. Cup count is unchanged (two, matching the transitive active), so $\kappa(D)$ tracks argument slots rather than surface case marking; per Equation 12 only the box count distinguishes the two diagrams (the passive inserts the "is chased by" box in place of the active "chases" box).

8 Categorical Distributional Semantics (DisCoCat): Composing Vector Spaces via Type Derivations

Where we are in the argument. section 6–section 7 gave the framework a *syntactic* apparatus: pregroup derivations with case-typed wires. This chapter supplies the matching *semantic* one — the Coecke–Sadrzadeh–Clark meaning functor $F\colon \mathbf{Preg} \to \mathbf{FinHilb}$ that sends each word to a tensor in a vector space and each pregroup contraction to an inner-product composition, so the same syntactic diagram simultaneously proves well-typedness *and* computes the composed meaning.

8.1 The Formalism–Distribution Impasse and Why Montague Cannot Meet Harris

The study of linguistic meaning has historically been fractured between two traditions offering complementary strengths and weaknesses:

1. **Formal semantics**, following Montague [1973], strictly models meaning compositionally: the algebraic meaning of any complex expression functions as a direct product of its parts and their syntactic combination. This tradition excels at capturing rigid logical structure (quantification, negation, modality) but severely struggles with fluid lexical meaning, typically treating content words as opaque, unanalyzed primitives.
2. **Distributional semantics** models meaning empirically: a word's meaning organically emerges from its distribution across contexts, typically encoded computationally as a dense vector in high-dimensional space. This distributional hypothesis—famously summarized as "You shall know a word by the company it keeps" [Firth, 1957]—traces directly to J.R. Firth and Zellig Harris [1954], whose algebraic analysis proved that linguistic elements occurring in similar environments inherently share semantic properties. While this tradition beautifully captures graded similarity and geometric analogy, it lacks rigorous compositional structure. Under pure distribution, "dog bites man" and "man bites dog" collapse into identical vector representations.

This theoretical tension represents one of the deepest impasses in modern cognitive science. Turney and Pantel's [2010] survey of vector space models proved that distributional methods capture fine-grained semantic distinctions—synonymy, antonymy, hypernymy, analogy—but isolate these victories to the word level. Baroni and Lenci [2010] built a tensor-based *Distributional Memory* framework partially addressing this compositionality by structuring co-occurrence data into a three-way tensor over (word, relation, word) triples, yet this approach lacked a principled type-logical backbone. Lenci [2018] surveyed this landscape and identified the central open problem: how to fuse the *algebraic compositionality* of formal semantics with the *empirical grounding* of distributional vector models.

The **DisCoCat** (Distributional Compositional Categorical) framework [Coecke et al., 2010] resolves this long-standing tension. It deploys category theory to compose distributional meanings directly according to syntactic structure, yielding a semantic framework that is simultaneously *compositional* (via categorial grammar), *distributional* (via corpus-derived vector spaces), and *algebraically principled* (via rigid monoidal category theory).

8.2 From Word2Vec to Transformer Attention: DisCoCat as the Algebraic Formalization of Distributional Semantics

The distributional programme has undergone a dramatic computational intensification in the era of large language models (LLMs). The trajectory from classical co-occurrence matrices through static word embeddings to contextual transformers can be understood as a progressive *enrichment* of the distributional hypothesis itself:

1. **Static embeddings**: Mikolov et al.'s [2013] Word2Vec (skip-gram and CBOW) demonstrated that targeted prediction-based training on local context windows naturally generates word vectors exhibiting striking algebraic regularity—yielding the famous "king − man + woman
approx queen" geometric analogy. Pennington et al.'s [2014] GloVe successfully incorporated global co-occurrence statistics via log-bilinear regression, yielding dense vectors whose inner products cleanly approximate underlying pointwise mutual information. Both models instantiate the distributional hypothesis in its classical Firthian form: contextual use strictly determines meaning, permanently encoding it as geometric position within a learned vector space.
2. **Contextual embeddings**: BERT [Devlin et al., 2019] and GPT [Radford et al., 2018] dramatically push beyond static type-level representations into dynamic *token-level* contextualized embeddings, where the identical word dynamically receives entirely different coordinate vectors in varying contexts. In terms of the formal vs. distributional dichotomy, this jump represents a critical advance: contextualized embeddings successfully (though implicitly) capture *compositional* structure by continuously conditioning word representations upon their full sentential environment. This addresses polysemy and complex constructional effects that static embeddings routinely conflate.

3. **Transformer architecture**: The transformer [Vaswani et al., 2017] implements distributional composition via multi-head self-attention, where each head computes a weighted combination of input token representations. The attention weights $\alpha_{ij} = \text{softmax}(Q_i K_j^T / \sqrt{d_k})$ operate analogously to the enriched hom-values detailed in section 11: they encode the *degree of contextual relevance* between tokens i and j within a tuned representational subspace—yielding a graded, learned distributional mapping.

The mapping back to our categorical framework is direct: **DisCoCat supplies the algebraic formalization of what modern LLMs learn empirically.** A transformer builds sentence representations by attending to syntactically and semantically relevant tokens through learned weight matrices. DisCoCat achieves the same goal by composing word vectors through type-logical derivations within a compact closed category. The central functor $F : \mathbf{Preg} \to \mathbf{FVect}$ defining DisCoCat is therefore the *principled* version of the composition that neural attention learns from data. Gavranović's [2024] categorical learning programme confirms this perspective rigorously, proving that neural attention heads operate as parameterized optics—categorical constructions composing functorially, mirroring DisCoCat derivations.

For case theory specifically, the transformer analogy is illuminating: each attention head in a transformer can be understood as learning a particular *relational role*—attending to subjects, objects, modifiers, or other grammatical functions. This is precisely the role that case marking plays in natural language: structuring who-does-what-to-whom. The case-typed noun spaces of our enriched DisCoCat model ($N_{\text{NOM}}, N_{\text{ACC}}, N_{\text{DAT}}, ...$) correspond to the role-specific representational subspaces that different attention heads learn to inhabit.

8.3 The Meaning Functor $F : \mathbf{Preg} \to \mathbf{FVect}$

8.3.1 Pregroups and Vector Spaces Share a Category

DisCoCat's central observation is that pregroup grammars and vector spaces share a common abstract structure: both are **compact closed categories**. This means there exists a *meaning functor*:

$$F : \mathbf{Preg} \to \mathbf{FVect} \tag{9}$$

from the pregroup grammar category (where objects are types and morphisms are grammatical reductions) to the category of finite-dimensional vector spaces (where objects are vector spaces and morphisms are linear maps).

Under this functor:

- Noun types n map to a vector space N (the noun space)
- Sentence types s map to a vector space S (the sentence space)
- A transitive verb of type $n^r \cdot s \cdot n^l$ maps to a tensor in $N \otimes S \otimes N$
- Pregroup contractions (cups/caps) map to the standard inner product and its dual

8.3.2 Tensoring, Then Contracting: How "Alice Chases Bob" Becomes a Vector

The compositional meaning of a sentence is computed by tensoring the word meanings and then contracting the result along the indices determined by the syntactic derivation. For "Alice chases Bob":

$$\overrightarrow{\text{Alice chases Bob}} = (\overrightarrow{\text{Alice}} \otimes \overleftarrow{\text{chases}} \otimes \overrightarrow{\text{Bob}}) \circ (\varepsilon_N \otimes 1_S \otimes \varepsilon_N) \tag{10}$$

where $\varepsilon_N : N \otimes N \to \mathbb{R}$ is the compact closure map (inner product). This computation has a direct diagrammatic representation as a string diagram—the same Joyal–Street [Joyal and Street, 1991] formalism that governs the syntax.

8.3.3 DisCoCat Resolves "Dog Bites Man" vs "Man Bites Dog"

Grefenstette and Sadrzadeh [2015] demonstrated that DisCoCat models can outperform purely distributional baselines on disambiguation and sentence similarity tasks. The key advantage is that compositional structure resolves ambiguities that bag-of-words models cannot: "dog bites man" and "man bites dog" receive different sentence vectors because the syntactic structure assigns different roles to the nouns.

Attention as Cup contraction. The claim that DisCoCat provides the "algebraic formalization" of transformer-based LLMs can be made precise. In a transformer's self-attention mechanism, the query–key inner product $\text{softmax}(QK^\top/\sqrt{d})V$ selects which word vectors interact—effectively implementing a *soft* version of the Cup contraction. The pregroup Cup $\varepsilon : n^r \otimes n \to I$ contracts two noun wires into a scalar; the attention inner product $q_i \cdot k_j$ contracts two contextualized vectors into an attention weight, which then mixes the value vectors. The difference is that the DisCoCat Cup is *binary* (either the types match or they don't) while the attention Cup is *graded* (producing a real-valued weight). This is precisely

the $[0,1]$-enrichment of section 11 applied to the type-reduction level: each attention head learns an enriched Cup that contracts types with learned confidence weights rather than categorical yes/no type-matching. Multi-head attention then corresponds to the tensor product of h independent enriched contraction maps, each attending to a different substring of the relational structure—analogous to how DisCoCat composes multiple Cup contractions for multi-argument verbs (Figure 12).

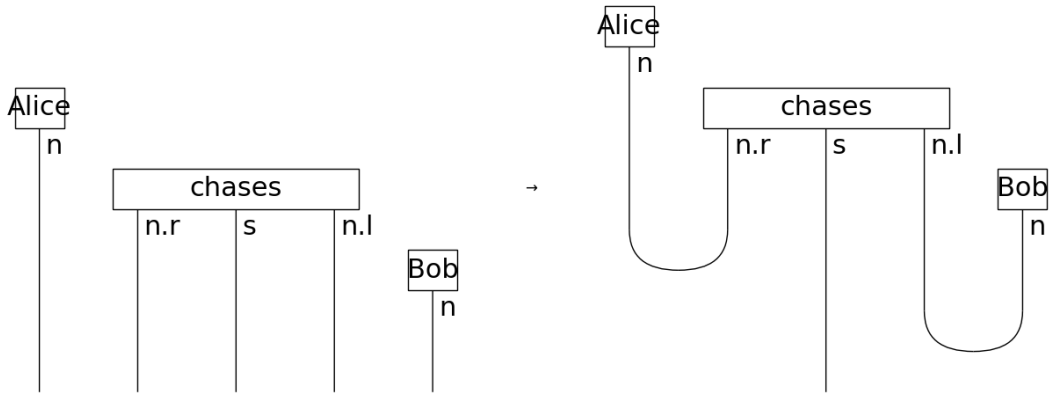

Figure 12: Cup contractions compute sentence meaning by contracting word tensors along syntactically determined indices. The DisCoCat meaning functor $F : \mathbf{Preg} \to \mathbf{FVect}$ applied to "Alice chases Bob." **Left panel**: pre-contraction tensor product $n \otimes (n^r \otimes s \otimes n^l) \otimes n$, showing three Word boxes with pregroup types. **Right panel**: fully contracted diagram where two Cup contractions ($\varepsilon : n^r \otimes n \to 1$ and $\varepsilon : n^l \otimes n \to 1$) reduce the five-wire product to sentence wire s. Under F, noun types map to N, the sentence type to S, and the verb's type to $\overleftrightarrow{\text{chases}} \in N \otimes S \otimes N$. The Cup contractions become inner products $\varepsilon_N : N \otimes N \to \mathbb{R}$, computing $\overrightarrow{\text{Alice chases Bob}} \in S$.

8.4 Case-Typed Noun Spaces and Alignment as Natural Transformation

The present contribution lies in showing how case marking enriches the DisCoCat framework with explicit role structure. In a case-marked DisCoCat model:

1. **Typed noun spaces**: Instead of a single noun space N, we define case-specific spaces $N_{\text{NOM}}, N_{\text{ACC}}, N_{\text{DAT}}, \ldots$ Morphisms between these spaces encode case-changing operations (passivization, dative shift, etc.).
2. **Case-constrained composition**: The meaning functor F maps case-typed pregroup derivations to tensor contractions that respect case constraints. A verb seeking a nominative subject and accusative object contracts only with vectors from the appropriate spaces.
3. **Alignment as Natural Transformation**: Cross-linguistic alignment differences correspond to different meaning functors. However, we go further by modeling case alignment as **natural transformations** between functors. An accusative-language transformation $\nu_{\text{acc}} : \mathcal{J} \to F_{\text{acc}}$ identifies S and A arguments; an ergative transformation ν_{erg} identifies S and P. (We write ν here so η remains reserved for the compact-closure **cap** in pregroup diagrams.) This categorification allows us to model "grammar" as the requirement that these transformations commute with the DisCoCirc entity wires, ensuring role persistence across sentences.

The monotonic complexity–valency relationship (Equation 12 in section 9) is visible in the cup count of Figure 13; the diagram also exemplifies Shimojima's [1996] free-ride inference over argument structure and compositional flow.

The following sections extend this model to discourse and quantum computation.

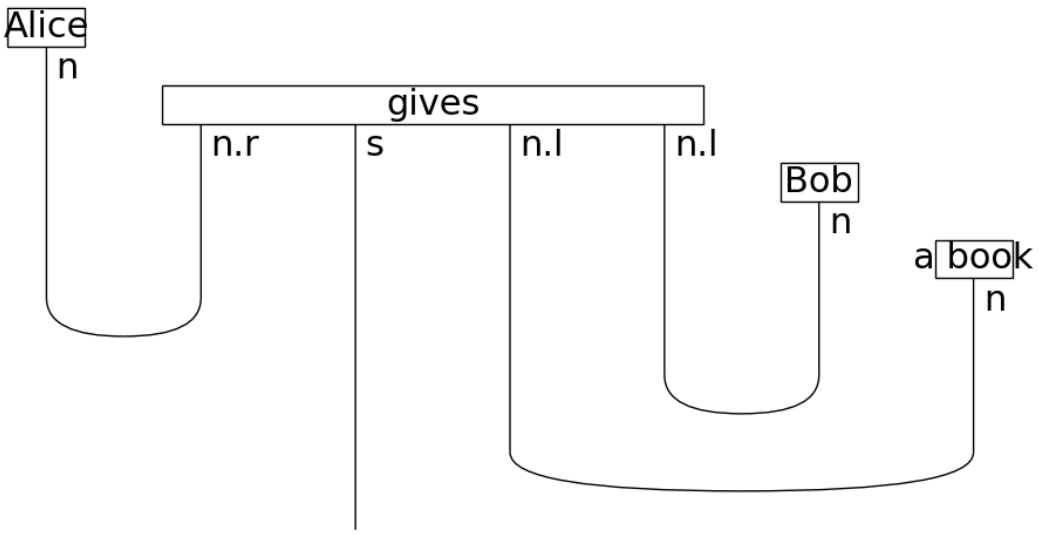

Figure 13: Ditransitive pregroup diagram for "Alice gives Bob a book": four word boxes (Alice, gives, Bob, book) with three Cup contractions reducing the five-argument tensor to sentence type s, exhibiting higher valency ($\kappa(D)$) than a transitive clause. Generated by `render_discopy_ditransitive()` in `src/visualization/discopy_diagrams.py`.

9 Compact Closure: Snake Equation, Valency, and Four Complexity Metrics

Where we are in the argument. section 6–section 8 made each sentence a string diagram that simultaneously expresses syntax and semantics. This chapter zooms in on the *axiom* that makes the machine run — the compact-closure snake equation $(\varepsilon_n \otimes 1_n) \circ (1_n \otimes \eta_n) = 1_n$ — and uses its consequences to define four graded complexity metrics (word count, cup count, cap count, depth) that turn "how hard is this sentence to parse?" from a qualitative question into a computable one, feeding directly into the enriched hom-values of section 11.

9.1 The Snake Equation Powers Every Pregroup Contraction

9.1.1 The Snake Equation (Zigzag Identity)

The essential algebraic engine driving DisCoCat is the strict **compact closure** of the governing pregroup category. For every lexical type n, the corresponding adjunction maps—$\eta_n : 1 \to n \otimes n^r$ (the cap expansion) and $\varepsilon_n : n^r \otimes n \to 1$ (the cup contraction)—must mathematically satisfy the fundamental **snake equation** (also known as the topological zigzag identity):

$$(\varepsilon_n \otimes 1_n) \circ (1_n \otimes \eta_n) = 1_n \tag{11}$$

Translated into intuitive string-diagrammatic terms, a cup geometrically composed with a cap bridging adjacent wires immediately "straightens out" into a solid identity wire—a topological zigzag seamlessly canceling into a continuous straight line. This foundational axiom is not merely a geometric curiosity: it physically functions as the *engine* powering every valid pregroup type reduction. Every recorded grammatical contraction (e.g., a noun canceling against an active verb argument slot) constitutes a direct instance of the cup map ε, and every grammatical expansion instantiates the cap map η. The continuous snake equation algebraically guarantees that these interwoven contractions and expansions remain globally well-behaved—ensuring they can be freely inserted and removed without ever altering the final semantic meaning of the derivation.

Cognitively, the snake equation provides an immediate *visual proof* of coherence. A cognitive agent inspecting a string diagram can verify well-formedness by confirming that all zigzags cancel—a purely spatial operation requiring no sequential algebraic computation, and thus instantiating Shimojima's [1996] free-ride inference in its most direct form (Figure 14).

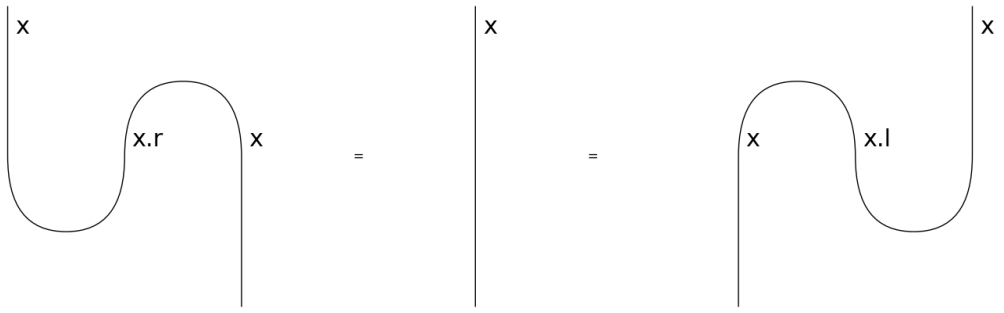

Figure 14: The snake equation is the algebraic engine powering every pregroup type reduction. The compact closure axiom (Equation 11) rendered by DisCoPy as a three-panel equality: **left panel** shows the right-adjoint zigzag $(\varepsilon_n \otimes 1_n) \circ (1_n \otimes \eta_n)$ with the x^r intermediary; **middle panel** is the identity wire 1_n that the zigzag equals; **right panel** shows the mirror-image left-adjoint zigzag using the x^l adjoint. Verified computationally via `diagram.normal_form() == Id(Ty('x'))`. Each pregroup contraction (noun canceling with verb argument) is an instance of ε; the snake equation guarantees that cup-cap pair insertions and removals leave derivations invariant (cf. section 6). Generated by `render_discopy_snake()` in `src/visualization/discopy_diagrams.py`.

Figure 15 unpacks the same axiom into three explicit panels: the left-hand zigzag carrying its η_n and ε_n labels, the identity wire 1_n it equals, and an algebraic recap of both compact-closure equations with cap and cup type signatures.

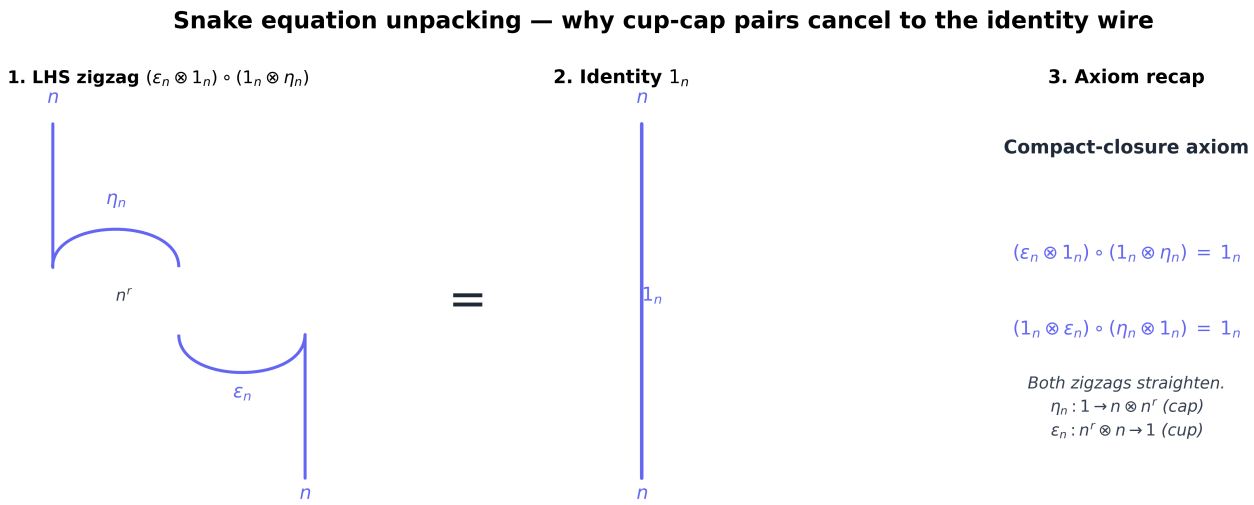

Figure 15: Snake equation unpacked. Panel 1: the zigzag $(\varepsilon_n \otimes 1_n) \circ (1_n \otimes \eta_n)$ with η at the top (cap, $1 \to n \otimes n^r$) and ε at the bottom (cup, $n^r \otimes n \to 1$). Panel 2: the identity wire 1_n it equals. Panel 3: axiom recap showing both zigzag equations straighten to the identity, with explicit type signatures for η and ε. Generated programmatically from `src.visualization.category_unpacking.render_snake_equation_unpacking()`.

9.1.2 Four Metrics Quantify Derivational Complexity

The algebraic properties of pregroup diagrams support quantitative analysis of derivational complexity. Our `complexity_metrics` module implements four complementary measures using the DisCoPy library:

1. **Box count**: The number of lexical entries (Word boxes) in the diagram, corresponding to the sentence's word count from the type-logical perspective. A transitive sentence has 3 boxes (subject, verb, object); a ditransitive sentence has 4 or more.
2. **Cup/Cap count**: The number of contraction and expansion operations. Cups (denoted ε) count argument consumption; caps (denoted η) count argument introduction. The cup count directly reflects verb valency: an intransitive verb requires 1 cup, a transitive verb 2, and a ditransitive verb 3.
3. **Normal form**: A diagram is in *normal form* if no further simplifications (zigzag cancellations, box reordering)

are possible. The `compute_normal_form()` operation computes this canonical representative of the diagram's equivalence class. Normal form preservation under algebraic manipulation provides a correctness check for compositional operations.

4. **Syntactic complexity score**: A composite metric defined as:

$$\text{complexity}(D) = w_w \cdot |D|_{\text{words}} + w_c \cdot |D|_{\text{cup}} + w_a \cdot |D|_{\text{cap}} + w_d \cdot \text{depth}(D) \tag{12}$$

where $|D|_{\text{words}}$ is the *lexical* box count (Cup/Cap plumbing excluded), $|D|_{\text{cup}}$ and $|D|_{\text{cap}}$ are contraction and expansion counts, and $\text{depth}(D)$ is the longest input-to-output path measured in boxes. The weights w_w, w_c, w_a, w_d are keyword arguments to `syntactic_complexity_score()`; the defaults w_w=1.0, w_c=0.5, w_a=0.25, w_d=0.1 encode the intuition that lexical entries contribute most, contractions moderately, expansions least, and depth a small residual penalty. Passing w_w=w_c=w_a=w_d=1.0 recovers an unweighted total-structure count. Deeper diagrams encode more complex syntactic derivations—a ditransitive sentence like "Alice gave Bob a book" (depth 7) is structurally more complex than a simple intransitive "Alice runs" (depth 3).

The `compare_diagrams()` function applies these metrics across a collection of diagrams, producing tabular comparisons suitable for cross-linguistic analysis. Figure 16 visualizes these metrics across sentence types of increasing valency, demonstrating the monotonic relationship between argument structure and derivational complexity.

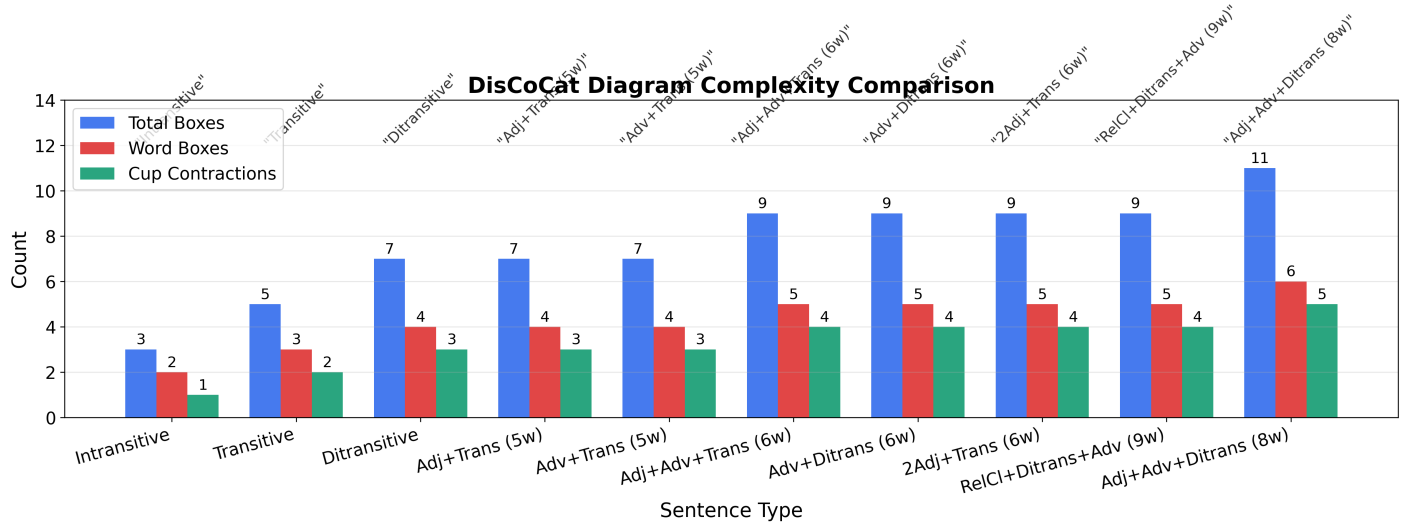

Figure 16: Verb valency dominates diagram complexity: categorical complexity score $\kappa(D)$ (Equation 12) plotted for ten sentence types ranging from intransitive to adjunct-heavy constructions, with natural-language exemplar sentences overlaid at each bar. Generated by `render_complexity_comparison()` in `src/visualization/complexity_plots.py` (orchestrated by `scripts/generate_diagrams.py`).

Key finding. Adverb and adjective modifiers increase box count but do not always add core argument Cups (modifier types $n \cdot n^l$ contribute one box and one cup each, but some adverbs add a box without an argument Cup), so κ can grow more slowly with adjunct stacking than with valency increases. Cup count $|D|_{\text{cup}}$ remains a reliable proxy for verb valency. In code, cup counts are obtained by iterating DisCoPy diagram boxes and testing for `Cup` instances (see `compare_diagrams()` in the project).

These metrics connect naturally to the enriched framework of section 11: diagram depth serves as a syntactic complexity proxy that can be incorporated into the enriched hom-value, providing a principled bridge between type-logical derivation cost and distributional semantic distance. Discourse-level persistence of entity wires (DisCoCirc) is developed in section 10. For multi-agent security, when tracking networks isolate an adversarial identity wire (a prompt injection) covertly attempting to merge with an ongoing command wire across communication boundaries, the circuit topology can flag the type violation before execution (section 20).

10 Beyond the Sentence: State Wires Accumulate Semantic History Across Discourse

Where we are in the argument. section 8–section 9 formalised individual sentences as compact-closed string diagrams and their complexity as a four-metric summary. This chapter extends composition *across* sentence boundaries: de Felice–Coecke DisCoCirc adds persistent entity wires that carry an entity's case-role history across a discourse, so that a three-sentence discourse traces Alice's NOM → ACC → NOM trajectory and Bob's ACC → NOM trajectory as first-class diagrammatic data rather than coreference annotation bolted on after the fact.

10.1 DisCoCirc State Wires Resolve Coreference and Role Shifts

De Felice and Coecke [2020] address this limitation by formulating **DisCoCirc** (Distributional Compositional Circuits). DisCoCirc extends the base categorical framework, empowering it to dynamically handle discourse-level semantic structure.

Specifically, DisCoCirc introduces continuous *state wires* that actively persist across isolated sentence boundaries, continuously encoding the dynamically evolving states of distinct discourse entities (e.g., characters, objects, shifting topics). For example, a multi-sentence sequence like "Alice chased Bob. He was terrified." is explicitly represented as an integrated geometric circuit where:

- Alice and Bob are wires that persist across both sentences.
- The pronoun "He" is resolved by actively connecting its topological wire directly to Bob's wire.
- The shifting emotional state "terrified" seamlessly updates the cumulative state information carried exclusively by Bob's persistent wire.

De Felice et al. [2022] subsequently extended this approach, developing a full-fledged, multi-layered circuit model built to systematically resolve ambiguity, tangled coreference, and overarching discourse coherence—all locked within the exact same continuous categorical formalism. A CCG-based pipeline for generating discourse circuits from syntactic parse trees has demonstrated that DisCoCirc can scale to text, dynamically composing sentence-level diagrams along shared entity wires via an iterative process of coreference resolution and wire merging [2021]. This pipeline approach has since been extended to a comparative framework for evaluating compositional AI model architectures [2025], confirming that the DisCoCirc wire-merging strategy generalizes across model families. Complementary work on **DiscoSG** (Discourse Scene Graphs) extends this approach to multi-sentence image captions, parsing text into scene graphs that capture cross-sentence coreference relations. Figure 17 illustrates a multi-sentence discourse diagram where entity wires persist across sentence boundaries. For case theory, DisCoCirc is significant because it shows how case-marked argument structure *composes across discourse*: the nominative subject of one sentence can become the accusative object of the next, and this transformation is tracked as a morphism in the discourse category.

10.2 Alice's Role Trajectory: NOM→ACC→NOM Across Three Sentences

The power of DisCoCirc for case theory becomes particularly vivid in multi-sentence discourses where the *same entity occupies different case roles* across sentences. Consider the three-sentence discourse:

> *"Alice chases Bob. Bob fears Alice. She smiles."*

In this discourse, Alice undergoes a complete cycle of case role reversals:

1. **Sentence 1**: Alice is NOM (Proto-Agent, the one chasing) and Bob is ACC (Proto-Patient, the one chased).
2. **Sentence 2**: Bob is now NOM (the one fearing) and Alice is ACC (the one feared)—a role reversal where Alice moves from agent to patient.
3. **Sentence 3**: "She" resolves anaphorically to Alice, who returns to NOM as the agent of smiling.

This NOM → ACC → NOM trajectory for Alice across three sentences is precisely the kind of *dynamic case assignment* that static single-sentence analyses cannot capture. The categorical representation as a triple tensor product $s \otimes s \otimes s$ (Figure 19) encodes each sentence as an independent pregroup derivation while preserving the entity identity that links them. This role reversal essentially requires applying the topological `Swap` operation (introduced for passivization in section 7) dynamically across the discourse boundary. In a full DisCoCirc implementation, Alice's entity wire would carry accumulated semantic state—the meaning of "She" in sentence 3 inherits the enriched state of an Alice who has first chased and then been feared, not merely the bare lexical entry for "Alice."

The trajectory is *executable*: the `src.diagrams.string_diagram.Discourse` class implements exactly this entity-wire-with-role-history bookkeeping. The following snippet, taken verbatim from the public API exercised by the test suite, builds the three-sentence Alice/Bob discourse and reads back the role history that drives Figure 19:

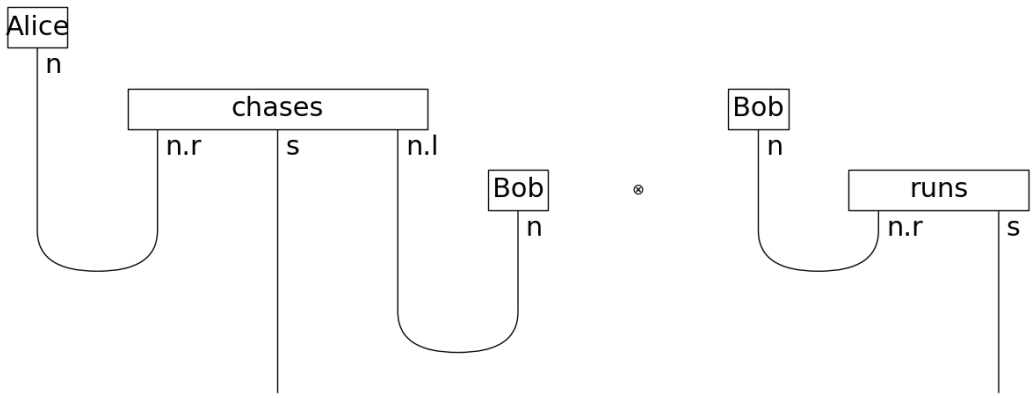

Figure 17: DisCoCirc type structure encodes discourse coherence as a tensor product of sentence types. DisCoPy pregroup grammar for "Alice chases Bob. Bob runs." showing inter-sentential composition. **Sentence 1** (left subdiagram): Alice (n) and Bob (n) contract into "chases" ($n^r \otimes s \otimes n^l$) via two Cups, yielding sentence type s. **Sentence 2** (right subdiagram): Bob (n) contracts into "runs" ($n^r \otimes s$) via one Cup, yielding s. The joint discourse type $s \otimes s$ encodes inter-sentential coherence as a tensor product, the computational primitive that DisCoCirc state wires exploit to propagate Bob's semantic state from ACC (sentence 1) to NOM (sentence 2). See Figure 18 for wire-colour confirmation of the role transition, and Figure 19 for the three-sentence DisCoPy tensor layout (role reversal; anaphora is developed in prose and in the `Discourse` API below, not as a separate pronoun box in the DisCoPy figure).

```python
from src.diagrams.string_diagram import Sentence, Discourse

discourse = Discourse()
discourse.add_sentence(Sentence.transitive("Alice", "chases", "Bob"))
discourse.add_sentence(Sentence.transitive("Bob", "fears", "Alice"))
discourse.add_sentence(Sentence.intransitive("Alice", "smiles"))

# Persistent entity wires (DisCoCirc state-wire reading)
assert discourse.entities == {"Alice", "Bob"}
assert discourse.num_sentences == 3

# Role history per entity reproduces the NOM → ACC → NOM trajectory
assert [r.name for r in discourse.role_history["Alice"]] == ["NOM", "ACC", "NOM"]
assert [r.name for r in discourse.role_history["Bob"]]   == ["ACC", "NOM"]

# `role_reversal_entities` flags every entity whose case role
# changes across the discourse — the formal analogue of the
# trajectory pictured in \autoref{fig:three-sentence-discourse}.
assert set(discourse.role_reversal_entities()) == {"Alice", "Bob"}
```

Slavic discourse supplies a *morphologically overt* analogue of the persistent state wire: in Serbian/BCS, second-position clitics (Wackernagel position) such as *je* AUX.3SG, *mu* DAT.3SG.M, *ga* ACC.3SG.M form a fixed-order cluster that *carries the case role of the discourse referent forward in time*. *Marko ga je video* "Marko him.ACC AUX saw" and *Dao mu ga je* "He gave him.DAT it.ACC AUX" make Bob's wire-and-case-history visible in the surface string in a way that English pronouns (which collapse case morphology onto a tiny three-form `I/me/my` paradigm) cannot. Russian, which lacks the BCS clitic cluster, instead encodes the same persistence directly in the suffix on the noun or full pronoun (*ego* ACC, *emu* DAT, *im* INS), so role reversals like Alice's NOM→ACC→NOM trajectory in *Alisa gonyaet Boba. Bob boitsja Alisy. Ona ulybaetsja* are reconstructible *from the morphology alone*, with no positional ambiguity. These are the cleanest natural-language witnesses for the entity-wire-with-role-history bookkeeping that the `Discourse` API formalises.

The point of including the snippet in prose is reproducibility, not novelty: every assertion above is exercised in `tests/test_diagrams_string_diagram.py`, so a reader who clones the repository can re-derive the discourse-level role trajectories with no additional setup beyond `uv sync`. The same `Discourse` object is what

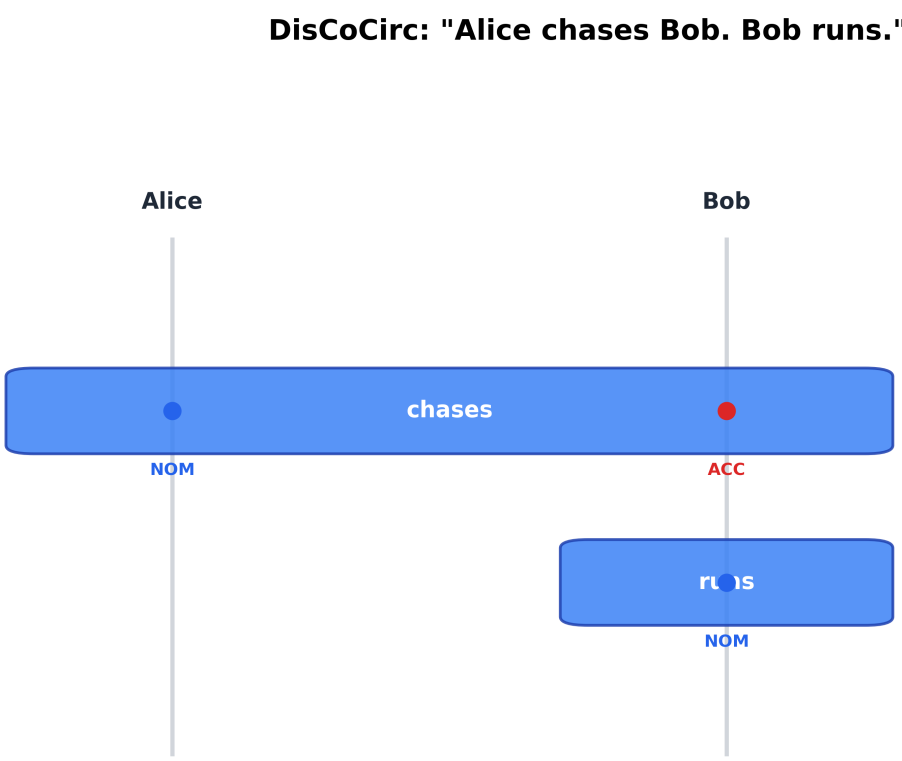

Figure 18: Wire colour provides independent visual confirmation of the ACC→NOM role transition that Figure 17 encodes algebraically. Native matplotlib DisCoCirc diagram for the same two-sentence discourse, generated via `src.visualization.string_diagrams.render_discocirc_discourse()` without the DisCoPy library. **Vertical grey wires**: persistent entity wires for Alice and Bob, confirming that identity is tracked across the sentence boundary. **Colour-coded role discs**: Bob's wire is marked **red** (ACC) at sentence 1 (chased object) and **blue** (NOM) at sentence 2 (running agent). The ACC→NOM colour change is directly readable from the diagram without algebraic computation — Shimojima's free-ride inference in action. This independent rendering cross-validates that `src.diagrams.string_diagram.Discourse` correctly resolves entity identity and role reassignment using only case-role metadata.

`src.visualization.string_diagrams.render_discocirc_discourse()` consumes to produce Figure 18 — closing the loop between symbolic case assignment, computational state-wire bookkeeping, and the colour-coded role transitions visible in the diagram.

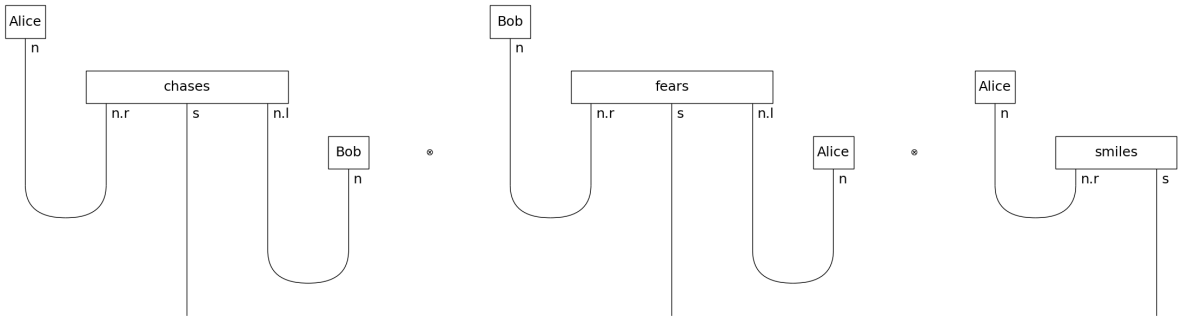

Figure 19: Dynamic case role reversal tracks entity identity across three-sentence discourse. DisCoPy rendering with lexical heads matching the discourse above: "Alice chases Bob. Bob fears Alice. Alice smiles." (sentence 3 uses the subject node *Alice*—same referent as anaphoric *She* in the running text; DisCoPy boxes are lexical, not pronominal). **Sentence 1**: Alice NOM, Bob ACC. **Sentence 2**: Bob NOM, Alice ACC—a complete agent–patient reversal. **Sentence 3**: Alice NOM as agent of *smiles*. Alice's trajectory NOM→ACC→NOM is tracked by the triple tensor $s \otimes s \otimes s$. In a full DisCoCirc implementation, Alice's entity wire carries accumulated semantic state—"She" in the gloss inherits the enriched state of an Alice who has both chased and been feared. This dynamic case assignment across discourse boundaries is what lambeq Gen II [Krawchuk et al., 2025b] compiles into parameterized quantum circuits.

The DisCoPy rendering encodes entity persistence *algebraically* (each sentence contributes a tensor factor s_i and the entity wires are shared by construction), but the reader must infer the shared-entity structure from context. Figure 20 makes the structure *visual*: each entity is given its own colour (Alice indigo, Bob amber), the three sentence panels sit side-by-side with cups coloured by the entity each one contracts, and the bottom *role-history ribbon* explicitly plots Alice's NOM → ACC → NOM trajectory and Bob's ACC → NOM trajectory with the case-role palette used throughout the paper. It is the same object as Figure 19 — re-presented with the Frobenius-spider entity-identity bookkeeping that DisCoCirc posits, made graphically explicit.

10.3 lambeq Gen II Compiles DisCoCirc Discourse Diagrams

The categorical structure of DisCoCat maps naturally onto quantum circuits: the tensor product structure of **FVect** is identical to the tensor product structure of **Qubit**, the category of qubit systems. This observation underlies the **QNLP** (Quantum Natural Language Processing) program [Meichanetzidis et al., 2020], which implements DisCoCat models as parameterized quantum circuits.

The **lambeq** library [Lorenz et al., 2021] provides a practical pipeline:

1. Parse a sentence into a pregroup derivation (via the neural CCG parser Bobcat or rule-based parsers)
2. Convert the derivation into a string diagram
3. Translate the diagram into a parameterized quantum circuit (or a classical tensor network)
4. Train the parameters on NLP tasks (classification, similarity, question answering)

Kartsaklis et al. [2021] demonstrate that this pipeline achieves competitive performance on question-answering tasks, confirming that the categorical structure captures genuine linguistic regularities even when instantiated on noisy near-term quantum hardware.

lambeq Gen II (released May 21, 2025 by Quantinuum) marks a significant advance by incorporating full **DisCoCirc** support as its core mathematical foundation, enabling the framework to scale beyond single-sentence semantics to discourse-level NLP [Krawchuk et al., 2025b, Quantinuum, 2025]. With over 50,000 downloads, lambeq Gen II achieves language neutrality, improved trainability, and compositional interpretability for explainable AI on quantum hardware. The new `DisCoCircReader` API automatically compiles long texts and multi-page documents into discourse-level quantum circuits, with entity wires tracking semantic state persistence across sentence boundaries—closing the gap between sentence-level DisCoCat and discourse-level case role tracking. This is directly relevant to the case role reversal phenomena discussed in Figure 19: lambeq Gen II can, in principle, compile such multi-sentence case-dynamic discourses into trainable quantum circuits.

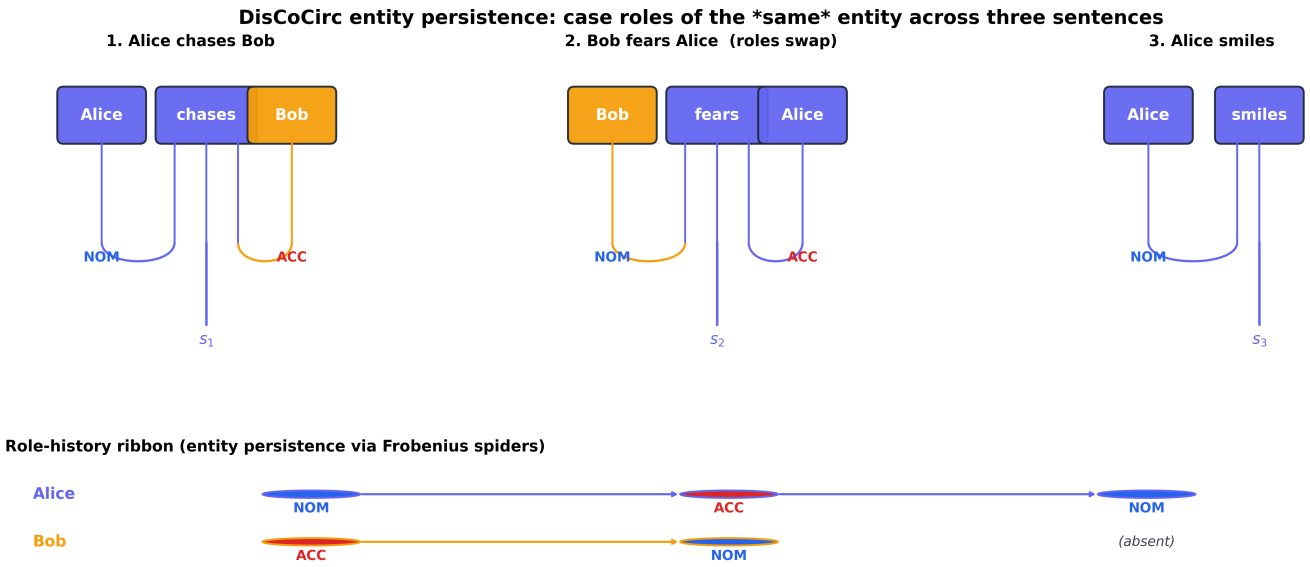

Figure 20: DisCoCirc entity-persistence unpacking for *Alice chases Bob. Bob fears Alice. Alice smiles.* Three sentence panels (top) plus a role-history ribbon (bottom). Each entity has a dedicated colour (Alice indigo, Bob amber) and the cup contractions are coloured by the entity each one consumes. The ribbon shows Alice traversing NOM→ACC→NOM and Bob traversing ACC→NOM (with an "absent" marker for sentence 3), exhibiting graphically the Frobenius-spider entity persistence that motivates DisCoCirc. Generated programmatically from `src.visualization.category_unpacking.render_discocirc_entity_persistence()`.

Foundational work by Meyer and Lewis integrates DisCoCat with density matrices for modeling dynamic meaning and lexical ambiguity in text, treating semantic states as mixed quantum states—providing an alternative to pure-state vector models that naturally accommodates ambiguity and partial information [2020].

Recent work on **string diagram rewriting** by Bonchi et al. [2022] provides the theoretical foundation for diagram simplification, showing that string diagram rewrite systems modulo Frobenius structure can be interpreted as double-pushout hypergraph rewriting—ensuring that the algebraic simplifications applied during normal form computation are provably sound. De Huybrecht [2024] extends DisCoCat with *subcategorization* for light verb constructions, demonstrating that the categorical framework accommodates sublexical compositional structure—a development that connects naturally to the monadic root syntax of Song [2022a] discussed in section 6.

For our case-theoretic framework, QNLP offers a concrete computational substrate: case categories could be implemented as quantum circuits where case roles correspond to quantum registers and grammatical relations correspond to parameterized gates. This connection between linguistic case structure and quantum information processing—mediated entirely by the shared categorical formalism—illustrates the power of the diagrammatic approach.

10.4 No Barren Plateau for Local Observables

A central challenge for practical QNLP on near-term quantum hardware is the *trainability* of parameterized quantum circuits (PQCs): the vanishing gradient problem, or *barren plateau*, makes gradient-based optimization exponentially hard as circuit width and depth scale. Two recent results (2024) directly resolve this obstacle for linguistically motivated circuits:

1. **Rad et al.** [2024] introduce *reduced-domain parameter initialization*: rather than sampling all parameters uniformly from $[0, 2\pi)$, one initializes the circuit in a small-angle domain close to the identity. For circuits of the depth typical of DisCoCirc discourse diagrams (compiled from multi-sentence texts via coreference resolution), this initialization provably yields polynomial rather than exponential gradient decay—keeping optimization tractable as discourse length grows.

2. **Letcher et al.** [2024] derive tight, *assumption-free* lower bounds on the variance of cost function gradients for PQCs with local observables (e.g., Pauli operators restricted to a few-qubit subsystem). Their key finding is that, for POVMs restricted to local observables—exactly the structure of the case-role measurement operators E_c of Equation 38 (formalized in section 18)—no barren plateau effect occurs. This provides a theoretical guarantee that case-role classification circuits implemented via lambeq remain optimizable regardless of total circuit size, so long as the readout observable is local.

Together, these results underpin the practical feasibility of the F1 and F3 research directions of section 21: scaling case-marked DisCoCat/DisCoCirc models to corpora-scale quantum hardware without exponential gradient overhead. The geometric structure of lambeq's IQP and Sim4 ansätze, combined with these initialization and observable choices, provides a principled recipe for quantum case category training on near-term devices.

11 $[0,1]$-Enriched Case Categories: Hom-Values as Distributional Proximity

Where we are in the argument. section 4–section 10 have been working in a category whose morphisms are Boolean — a relation either exists between two case roles or it does not. This chapter upgrades that to $[0,1]$-enrichment so every pair of roles carries a *graded* hom-value $\mathcal{C}(A,B) \in [0,1]$ — a distributional proximity that doubles as the precision w_f used throughout section 14 / section 16 for prediction-error weighting, and that will feed into the magnitude invariant of section 12 and the topos-theoretic bridge of section 13.

11.1 Why Binary Morphisms Are Not Enough

The categories established in section 4—modeling case roles as objects and grammatical relations as morphisms—capture the *qualitative* topology of case systems: which roles exist and how they connect. However, actual linguistic data is fundamentally *quantitative*: certain grammatical relations are more probable than others, proto-role assignments vary in strength, and distributional similarity is a matter of degree.

To accommodate this quantitative dimension without sacrificing algebraic structure, we advance from ordinary categories to **enriched categories**. In an enriched category, hom-sets carry additional measurable structure rather than functioning as mere discrete sets of morphisms. The framework of Bradley et al. [2021] supplies the key formal construction.

11.2 Enriching Over $([0,1],\cdot,1)$: Identity, Sub-Multiplicative Composition, and Four Hom-Value Readings

11.2.1 The Identity and Composition Axioms for $[0,1]$-Enriched Case Categories

A category $\mathcal{C}$ enriched over the unit interval $([0,1],\cdot,1)$ assigns to every pair of objects A, B a *hom-value* $\mathcal{C}(A,B) \in [0,1]$ satisfying:

$$\mathcal{C}(A,A) = 1 \quad \text{(Identity)} \tag{13}$$

$$\mathcal{C}(A,C) \geq \mathcal{C}(A,B) \cdot \mathcal{C}(B,C) \quad \text{(Composition)} \tag{14}$$

The identity axiom dictates that every linguistic expression remains maximally related to itself. The composition inequality imposes a strict logical boundary, demanding that distributional relatedness inherently composes sub-multiplicatively: if expression A proves 80% related to B, and B remains 70% related to C, the overarching algebraic structure formally guarantees that A must be at least $0.8 \times 0.7 = 56\%$ related to C.

11.2.2 Four Linguistic Readings of the Hom-Value: Probability, Proto-Role, Similarity, Predictability

Bradley et al. [2021] originally interpret these numerical hom-values strictly as empirical *conditional probabilities* measured within a massive distributional model: $\mathcal{C}(A,B) = P(\text{context} \mid A \text{ and } B \text{ co-occur})$. Within our customized case-theoretic application, we actively broaden this interpretation to capture deep grammatical phenomena (Table 7):

Table 7: Four linguistic interpretations of hom-values in $[0,1]$-enriched case categories.

Hom-value interpretation	Domain	Example
Conditional probability	Corpus statistics	P(ACC role | transitive verb context)
Proto-role strength	Semantic typology	Degree of Proto-Agent satisfaction
Distributional similarity	Vector semantics	Cosine similarity of case-role embeddings
Morphological predictability	Morpholexicology	Reliability of case-marking paradigm

11.2.3 When the Composition Inequality Fails: A Worked English NOM–ACC–DAT Example

Architectural note — two decoupled number systems. The enriched hom-value $\mathcal{C}(A,B) \in [0,1]$ defined here is *not* the same scalar as the morphism weight $w_f \in [0,1]$ that appears on every `Morphism` in the `CaseCategory` of section 4. The `CaseCategory` assigns $w_f = 1.0$ to every structurally present morphism (it encodes *admissibility* — a binary question dressed as a real to support multiplicative composition along chains, per Eq.

3), whereas the `EnrichedCategory` assigns a *graded distributional proximity* between every pair of roles including non-adjacent ones. The two number systems serve different purposes and are intentionally decoupled in the implementation: `src/case_systems/case_category.py::standard_case_category` ships unit-weight morphisms, while `src/enriched_cat/enriched.py::standard_enriched_category` ships the full 8×8 proximity matrix Z. When a prediction-error formula in section 14 or section 16 references a "weight" $w_f = \mathcal{C}(A, B)$, the referent is the *enriched* hom-value — never the morphism weight in the case category.

Our `EnrichedCategory` class implements this structure directly. The constructor takes a list of `CaseRole` objects and a NumPy proximity matrix encoding hom-values:

```python
from src.enriched_cat.enriched import EnrichedCategory
from src.case_systems.case_category import CaseRole
import numpy as np

roles = [CaseRole.NOM, CaseRole.ACC, CaseRole.DAT]
proximity = np.array([
    [1.00, 0.85, 0.30],  # NOM: high with ACC, low with DAT
    [0.85, 1.00, 0.45],  # ACC: high with NOM, moderate with DAT
    [0.30, 0.45, 1.00],  # DAT: low with NOM, moderate with ACC
])
cat = EnrichedCategory(
    name="English Case Proximity",
    roles=roles,
    proximity_matrix=proximity,
)

# Verify composition inequality: 0.30 >= 0.85 * 0.45 = 0.3825?
# This fails! English NOM-DAT is too distant relative to the chain.
assert not cat.check_composition_inequality(CaseRole.NOM, CaseRole.ACC, CaseRole.DAT)
```

The composition inequality violation here is linguistically meaningful: it tells us that the NOM→ACC→DAT chain overestimates the direct NOM→DAT relatedness, reflecting the typological fact that subject–recipient identity (e.g., in benefactive constructions) is more restricted than the product of agent–patient and patient–recipient proximities. Figure 21 shows the pairwise hom-values $\mathcal{C}(A, B) \in [0, 1]$ between case roles as an annotated heatmap of the proximity matrix.

Slavic syncretism as an empirical anchor for high hom-values. The fourth row of Table 7 — *morphological predictability* — is most cleanly calibrated against Slavic case morphology. Where a paradigm collapses two morphological cells into one form, the morphological-predictability hom-value approaches its upper bound:

- **Russian masculine animate ACC = GEN** (e.g. *brata*, *čeloveka*) $\Rightarrow \mathcal{C}(\text{ACC}, \text{GEN}) \approx 1$ for that declension class.
- **Russian neuter NOM = ACC** (e.g. *okno* "window" identical in subject and direct-object position) $\Rightarrow \mathcal{C}(\text{NOM}, \text{ACC}) \approx 1$ for that paradigm.
- **Serbian/BCS dative–locative singular merger** in many declensions (e.g. *gradu* serves both *prema gradu* "toward-DAT the city" and *u gradu* "in-LOC the city") $\Rightarrow \mathcal{C}(\text{DAT}, \text{LOC}) \approx 1$.

These are not modelling choices: they are *observed* identifications of morphological cells, supplying directly-measurable upper bounds on the enriched hom-values for the corresponding role pairs. They co-exist with low hom-values for the same role pairs under the *distributional* or *proto-role* readings of Table 7 — a useful reminder that the four interpretations are projections of a richer multi-channel proximity, not competing definitions.

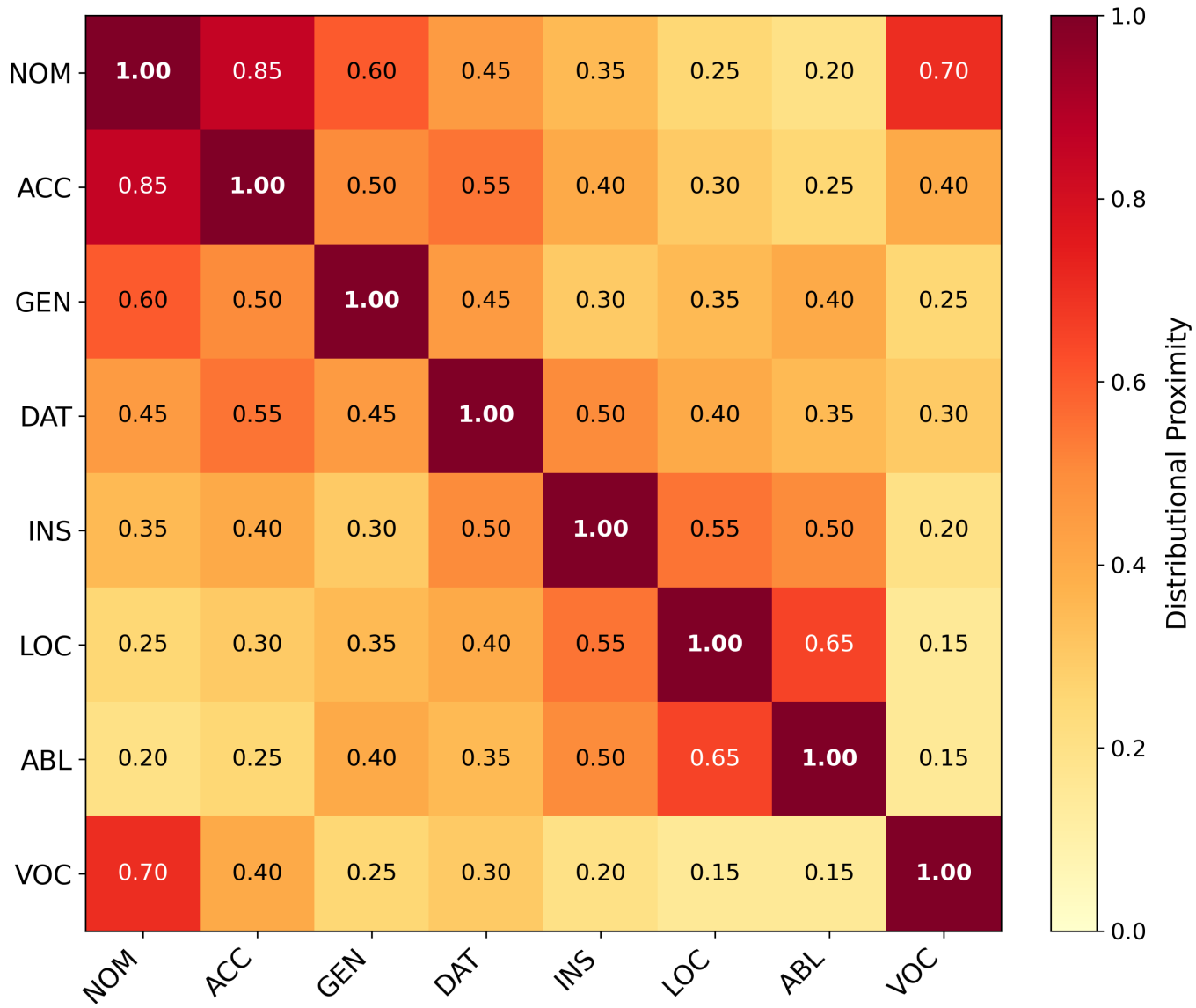

Figure 21: Core and peripheral argument complexes emerge from enriched distributional hom-values. Annotated heatmap of the 8×8 matrix with entries $\mathcal{C}(A_i, A_j) \in [0, 1]$ (YlOrRd scale; numeric labels in each cell). High-proximity blocks reveal the **core argument complex** (NOM–ACC–DAT: transitive and transfer morphisms) and the **peripheral complex** (LOC–INS–ABL: spatial and instrumental relations). GEN shows elevated proximity to both blocks via possessive modification; VOC shows high proximity to NOM (0.70, reflecting nominative-vocative morphological syncretism in many IE languages) but low proximity to the spatial-instrumental periphery (INS=0.20, LOC=0.15, ABL=0.15), reflecting its pragmatic rather than referential function. These hom-values assemble the similarity matrix Z used for categorical magnitude in section 12 (Equation 15), with axioms Equation 13–Equation 14. Generated programmatically from `src.visualization.enriched_diagrams.render_enriched_heatmap()` applied to `EnrichedCategory` data (see `scripts/generate_diagrams.py`).

12 Magnitude and Magnitude Homology: Effective Role Count, Lawvere Similarity Spaces, and Language as Enriched Category

Where we are in the argument. section 11 introduced $[0,1]$-enrichment so that the case category carries a full distributional-proximity matrix. This chapter extracts the principal quantitative invariant of that matrix — Leinster's *magnitude* $|\mathcal{C}| = \sum_{i,j}(Z^{-1})_{ij}$ — which gives a single-number answer to "how many effectively distinct case roles does the language use?" ($|\mathcal{C}| \approx 2.50$ for the standard eight-case system), and which will serve as the N400 magnitude-change proxy in section 15 and as a cognitive-security invariant in section 20.

A central mathematical invariant unique to enriched categories is their **magnitude**—a numerical quantity capturing the "effective size" of the category by discounting distributional overlap between objects.

For an enriched category with n objects, let Z be the $n \times n$ similarity matrix where $Z_{ij} = \mathcal{C}(i,j)$. The categorical magnitude is:

$$|\mathcal{C}| = \sum_{i,j}(Z^{-1})_{ij} \tag{15}$$

assuming Z is invertible. This magnitude metric connects to information theory:

- For a *discrete* category (no non-trivial relationships), $|\mathcal{C}| = n$ (the number of objects)
- For a highly connected category, $|\mathcal{C}| < n$ (objects are "redundant")
- Magnitude connects to the diversity measures studied in ecology (species diversity), graph theory (effective graph resistance), and information geometry (Fisher information)

Worked example. Consider the minimal 3-case category with objects $\{S, A, P\}$ and hom-values $\mathcal{C}(S,A) = 0.85$ (S and A share agentive contexts), $\mathcal{C}(A,P) = 0.70$ (transitive co-occurrence), $\mathcal{C}(S,P) = 0.40$ (weak S–P overlap). The similarity matrix Z is:

$$Z = \begin{pmatrix} 1.00 & 0.85 & 0.40 \\ 0.85 & 1.00 & 0.70 \\ 0.40 & 0.70 & 1.00 \end{pmatrix}, \quad Z^{-1} \approx \begin{pmatrix} 4.93 & -5.51 & 1.88 \\ -5.51 & 8.12 & -3.48 \\ 1.88 & -3.48 & 2.68 \end{pmatrix} \tag{16}$$

The magnitude is $|\mathcal{C}| = \sum_{i,j}(Z^{-1})_{ij} \approx 4.93 - 5.51 + 1.88 - 5.51 + 8.12 - 3.48 + 1.88 - 3.48 + 2.68 \approx 1.52$—substantially less than the cardinality 3 of the role inventory. The deficit $3 - 1.52 = 1.48$ is substantial—approximately 49% of the cardinality—reflecting that S and A share dense agentive contexts ($w = 0.85$) and A and P share transitive co-occurrence contexts ($w = 0.70$), making the proto-role system considerably redundant. By contrast, an accusative alignment—which merges S and A into a single NOM role—would yield a 2-object category with magnitude exactly 2.0, and the deficit $3 - 2.0 = 1.0$ quantifies the information lost by neutralization.

Scaling to full categories. Our `EnrichedCategory` implementation computes magnitude for any case category. For the standard 8-case English category with empirically calibrated distributional proximity values, the magnitude is approximately 2.50—the deficit $8 - 2.50 = 5.50$ reflects substantial distributional overlap: NOM and ACC share transitive contexts ($w = 0.85$), DAT and ACC overlap in double-object constructions ($w = 0.55$), GEN and NOM co-occur in possessive constructions ($w = 0.60$), and even peripheral cases such as VOC overlap with NOM ($w = 0.70$). Only 2.50 of the 8 case roles encode genuinely independent relational distinctions. This magnitude differential provides a quantitative formalization of Silverstein's [1976] case hierarchy: languages with more alignment-based neutralization (lower magnitude) have less relational discriminability, while richer case inventories (higher magnitude) make finer-grained relational distinctions.

Bradley [2021] established a link connecting categorical magnitude to classical information entropy via topological operad derivations. Her result proves that Shannon entropy acts as the unique algebraic derivation of a specific topological operad—a categorical structure governing the composition of enriched categories. This supplies theoretical justification for magnitude as a measurable geometric invariant quantifying linguistic complexity: the magnitude of any case category quantifies how much irreducible "information" that case system encodes regarding relational meaning.

Leinster and Shulman [2021] further develop **magnitude homology**, which categorifies magnitude from a scalar invariant to a graded homological invariant—detecting not just the "effective number" of objects but the higher-dimensional "holes" in the distributional landscape. For case categories, magnitude homology can distinguish between two systems with the same magnitude but different topological structure: a category where NOM–ACC–DAT form a tight cluster and all other cases are isolated looks identical in magnitude to one where the clustering is evenly distributed, but their magnitude homology groups differ, revealing that the former has a 1-dimensional "hole" (a missing transitive link) that the latter

fills. This finer invariant provides a richer classification of case systems than magnitude alone. Bradley and Vigneaux [2025] realize this programme on natural language by building categories of texts enriched from language-model next-token probabilities and computing magnitude and magnitude homology for associated metric spaces of texts—a concrete large-scale application of Leinster–Shulman theory beyond finite toy examples.

However, that LM-enriched construction exposes a vulnerability relevant to our cognitive synthesis: if magnitude homology computations are ported into non-classical environments, such as lambeq Gen II's Parameterized Quantum Circuits (section 17), the inherent environmental quantum noise (decoherence) acts as a non-trivial topological perturbation. Unless error-correction syndromes explicitly preserve the sequence of homology groups, the invariant may technically fail to commute. Thus, comparing the 1-dimensional "holes" of classical case-frames against their quantum algorithmic analogues must be approached with mathematical caution, respecting the shear forces introduced by measurement non-commutativity.

12.1 Language as Enriched Category: Transformer Attention Weights Are Context-Dependent Hom-Values

Bradley's [2024; 2025] broader program treats natural language itself as an enriched category, where:

- **Objects** are expressions (words, phrases, sentences)
- **Hom-values** encode distributional co-occurrence probabilities
- **Composition** models transitivity of distributional relatedness

This "language as enriched category" perspective has profound implications for case theory:

1. **Case roles emerge from distributional structure**: Rather than being imposed a priori, case distinctions arise from clusters of high hom-values in the enriched language category. Nouns that frequently appear in agent contexts cluster together, forming the "nominative" region of the category.

2. **Alignment types correspond to enriched structure**: Different languages partition the enriched category differently, and these partitions correspond to the alignment types (accusative, ergative, etc.) discussed in section 4.

3. **Language models are enriched functors**: A neural language model (such as a transformer) can be viewed as an enriched functor from the syntactic category to the semantic category, mapping type-logical derivations to distributional meaning representations while preserving the enriched structure.

The deep connection to modern distributional semantics is this: static embeddings operationalize hom-values as cosine similarity in a learned space, while contextualized transformers [Devlin et al., 2019, Vaswani et al., 2017] compute *dynamic* hom-values from sentential context—a move from a fixed enriched category to a parameterized one. The attention-as-enriched-cup analogy (developed in section 8) carries the same intuition here: layer-wise weights grade how strongly tokens couple, alongside Bradley et al.'s [2021] probabilistic reading of hom-objects.

12.2 Lawvere's Insight: Case Categories Are Similarity Spaces

The $[0, 1]$-enrichment connects to a deep tradition in categorical algebra. Lawvere showed that metric spaces are categories enriched over $([0, \infty], +, 0)$: the hom-value is the distance between points, the identity axiom says $d(x, x) = 0$, and the composition inequality is the triangle inequality $d(x, z) \leq d(x, y) + d(y, z)$. Our $[0, 1]$-enrichment is the multiplicative analogue: hom-values are *similarities* rather than distances, and the composition inequality is sub-multiplicative rather than sub-additive. The inequality *direction reverses* between the two settings because the monoidal structure reverses: in the additive metric setting we want small composites (triangle inequality is an upper bound on distance), whereas for multiplicative similarity we want large composites (so the natural inequality is the lower bound $\mathcal{C}(A, C) \geq \mathcal{C}(A, B) \cdot \mathcal{C}(B, C)$, equivalently the upper-bound form used in the section 23 notation appendix).

When is magnitude defined? Leinster's magnitude requires the similarity matrix Z to be invertible. For the standard eight-case category (Table 4) Z has condition number ≈ 9.5 and magnitude is well-defined ($|C| \approx 2.50$, deficit 5.50). For degenerate categories where roles are distributional clones (e.g. Z has repeated rows), Z becomes singular; the implementation in `src/enriched_cat/enriched.py` falls back to a Moore–Penrose pseudo-inverse with a logged warning, which yields a best-least-squares approximate magnitude but not the exact Leinster magnitude (which is simply undefined in that case).

Distributional-semantics embedding into $[0, 1]$. A word-embedding model instantiates the enriched category via the explicit map $\mathcal{C}(w, w') = (\cos(v_w, v_{w'}) + 1)/2 \in [0, 1]$, where $v_w, v_{w'}$ are the model's vector representations and cos is cosine similarity. This is the concrete choice that allows a pretrained transformer or word2vec model to be read as a $[0, 1]$-enriched functor.

This Lawvere-style perspective unifies our case categories with the geometry of distributional semantics: case roles are points in a "similarity space," and morphisms between them are paths weighted by distributional proximity. The magnitude of this space then quantifies the "effective dimensionality" of the case system—how many independent relational distinctions the language makes.

13 Topos-Theoretic Bridges: Transferring Results Across Case-Theoretic Frameworks

Where we are in the argument. section 4–section 12 have erected four formal layers — typological categories (section 5), type-logical pregroup syntax (section 6–section 7), distributional semantics (section 8–section 10), and enriched magnitude (section 11–section 12). This chapter supplies the meta-framework that lets results proven in one layer transfer to the others: classifying toposes and Morita equivalence (with the honest limitation that the repository checks necessary-but-not-sufficient invariant matching, not full topos equivalence).

13.1 The Inter-Theoretic Translation Problem

The preceding sections constructed four formally distinct perspectives on case: typological, type-logical, distributional, and enriched. A central methodological question arises: *when can structural results proved in one framework be carried over to another without starting from scratch?* Caramello's [2016] topos-theoretic bridge technique provides this methodology for properties that are **invariants of a shared classifying topos**, once Morita equivalence (or a suitable bridge topos) is in hand. We adopt this methodology as a research programme for case theory: establishing full Morita equivalences for the specific case-theoretic formalizations developed here remains largely open (see the implementation note later in this section), but the framework precisely identifies *what* must be proved and *what* would transfer as a result.

13.2 Classifying Toposes: The Logical Shape of a Theory

13.2.1 A Topos Is a Self-Contained Logical Universe

A **topos** is a category possessing the structural richness of a generalized "universe of sets": products, exponentials, and a subobject classifier (a "truth-value object") that supports internal first-order reasoning. The most familiar example is the category **Set** of ordinary sets; other important instances include presheaf categories $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ and sheaf categories over topological spaces. Intuitively, a topos provides a *self-contained logical universe* within which mathematical reasoning can proceed—and different toposes encode different logical constraints.

13.2.2 Morita Equivalence: Invariant Transfer Across Typologies

Every *geometric theory* $\mathbb{T}$ (axiomatized by sequents with finite conjunctions, arbitrary disjunctions, and existential quantification) generates a unique **classifying topos** $\mathcal{E}_{\mathbb{T}}$: a canonical topos such that, for any Grothendieck topos $\mathcal{F}$, models of $\mathbb{T}$ in $\mathcal{F}$ correspond naturally to **geometric morphisms** $\mathcal{F} \to \mathcal{E}_{\mathbb{T}}$. The classifying topos encodes the theory's "logical shape" independently of any particular model.

Caramello's insight [2016; 2021] is that formally different theories can share the same classifying topos up to geometric equivalence—a relation termed **Morita equivalence**. When $\mathcal{E}_{\mathbb{T}_1} \simeq \mathcal{E}_{\mathbb{T}_2}$, any property expressible as an invariant of the shared topos transfers automatically from $\mathbb{T}_1$ to $\mathbb{T}_2$ without re-proof.

13.3 A Chain of Morita Equivalences Connects the Four Case Theories

13.3.1 Each Case Framework as a Geometric Theory

We formalize each case-theoretic framework as a geometric theory:

1. **Typological case theory** $\mathbb{T}_{\mathrm{typ}}$: Objects are case roles, morphisms are grammatical relations, axioms specify alignment constraints (e.g., "S = A" in accusative alignment).
2. **Type-logical case theory** $\mathbb{T}_{\mathrm{log}}$: Objects are syntactic types, morphisms are Lambek calculus derivations, axioms specify well-formedness conditions (pregroup contractions).
3. **Distributional case theory** $\mathbb{T}_{\mathrm{dist}}$: Objects are vector spaces, morphisms are linear maps, axioms specify the composition law for distributional meaning (DisCoCat).
4. **Enriched case theory** $\mathbb{T}_{\mathrm{enr}}$: Objects are expressions, morphisms carry $[0, 1]$-valued weights, axioms specify the identity and composition inequalities.

13.3.2 The Bridge Programme: A Chain of Classifying Toposes for Case Theory

The **research programme** we pursue is that these four perspectives admit a topos-theoretic alignment: formally distinct case theories should be related by bridge toposes and, where Morita equivalence can be established, invariants proved in

one formulation transfer without re-proof. Equation (17) sketches the *target* picture—a chain of classifying toposes linked by intermediate geometric theories—not a single theorem asserted for all details in this manuscript.

$$\mathcal{E}_{\mathbb{T}_{\text{typ}}} \leftarrow \mathcal{E}_{\mathbb{T}_{\text{bridge}}} \rightarrow \mathcal{E}_{\mathbb{T}_{\text{log}}} \leftarrow \mathcal{E}_{\mathbb{T}_{\text{bridge}'}} \rightarrow \mathcal{E}_{\mathbb{T}_{\text{dist}}} \tag{17}$$

Intermediate toposes would be supplied by geometric theories simultaneously interpretable in both flanking frameworks. *Were* such Morita equivalences established, one would expect:

- **Syntactic theorems port to semantics**: Commutativity results in the type-logical setting would align with compositionality statements in the distributional setting.
- **Typological universals constrain distributional models**: Alignment types (accusative, ergative) would impose structural constraints on the vector-space data of DisCoCat-style models.
- **Enriched structure enriches all frameworks**: $[0, 1]$-valued weights could be pulled back to probabilistic readings of typological, logical, and distributional constructions.

Implementation status. The Python `topos` module does **not** implement general classifying toposes or full Morita equivalence proofs. It constructs finite **invariant profiles** (sort counts, symbol arities, axiom tallies) from `CaseCategory` instances and uses those profiles for `check_morita_equivalence()` and guarded `bridge_transfer()`. The equivalence check verifies *necessary but not sufficient* conditions: matching arity spectra and compatible axiom counts. A positive result means equivalence is *consistent with the evidence*; it does not constitute a proof. This proxy is a concrete, testable approximation of the diagram in (17), not a replacement for topos-level equivalence. Full classifying topos construction and Morita equivalence proof for specific case theory pairs remains an open research target (see F2).

Figure 22 visualizes the alignment functor between the Accusative and Ergative category instantiations, showing how the same eight case roles are connected by different morphism structures under different alignment types.

13.4 Reconciling Classical DAIF with Intuitionistic Topos Logic via Sheaf Cohomology

An unresolved tension arises when binding topos-theoretic invariants of case structures with the Distributional Active Inference (DAIF) layer (section 16). DAIF relies on Quantile Temporal Difference (TD) learning, which assumes Markovian updates over classical probability distributions.

Conversely, the internal logic of a classifying topos is generally intuitionistic—yielding *non-distributive lattices* of truth values where the classical law of excluded middle fails. Forcing classical Markovian density tracking onto non-distributive topological spaces risks mathematical collapse: the case alignments (modeled as geometric properties) may fail to compose covariantly under quantum NLP (lambeq Gen II) decoherence.

To resolve this, we propose that classical DAIF distributions cannot be mapped directly into the topos. Instead, a **sheaf-theoretic bridge** is required: probability densities over case assignments must be treated as *sections of a probability sheaf*. Local discrepancies in case assignment (e.g., during pragmatic garden-path discourse) resolve globally via *sheaf cohomology*. This ensures that while local parses resolve classically via quantile Huber loss, their global composition respects the intuitionistic structure of the geometric case theory.

13.5 Phillips's Result: Language-of-Thought Properties as Universal Topos Constructions

Phillips [2024] provides a striking application of topos-theoretic methods to cognitive science. He shows that the **Language of Thought** (LoT) hypothesis—the claim that cognition operates over structured, combinatorial representations with language-like properties—can be formalized categorically, and that the resulting structure is *universal* in the topos-theoretic sense.

Specifically, Phillips demonstrates that:

1. LoT properties (discrete constituents, role-filler independence, systematicity) arise as **universal constructions** in a topos—categorical products, fiber bundles, and presheaves.
2. Every topos supports an internal first-order logic, explaining how LoT-like logical capacities can emerge in systems (biological or artificial) whose internal architecture forms a topos.
3. The "shape" of cognitive representations is fundamentally **topological**, captured by presheaves and fiber bundles rather than by point-set structures.

Applied to case theory, Phillips's result is significant: because the Language of Thought is topos-universal, and our case categories are definable within any topos (as small categories governed by first-order axioms), every cognitive system

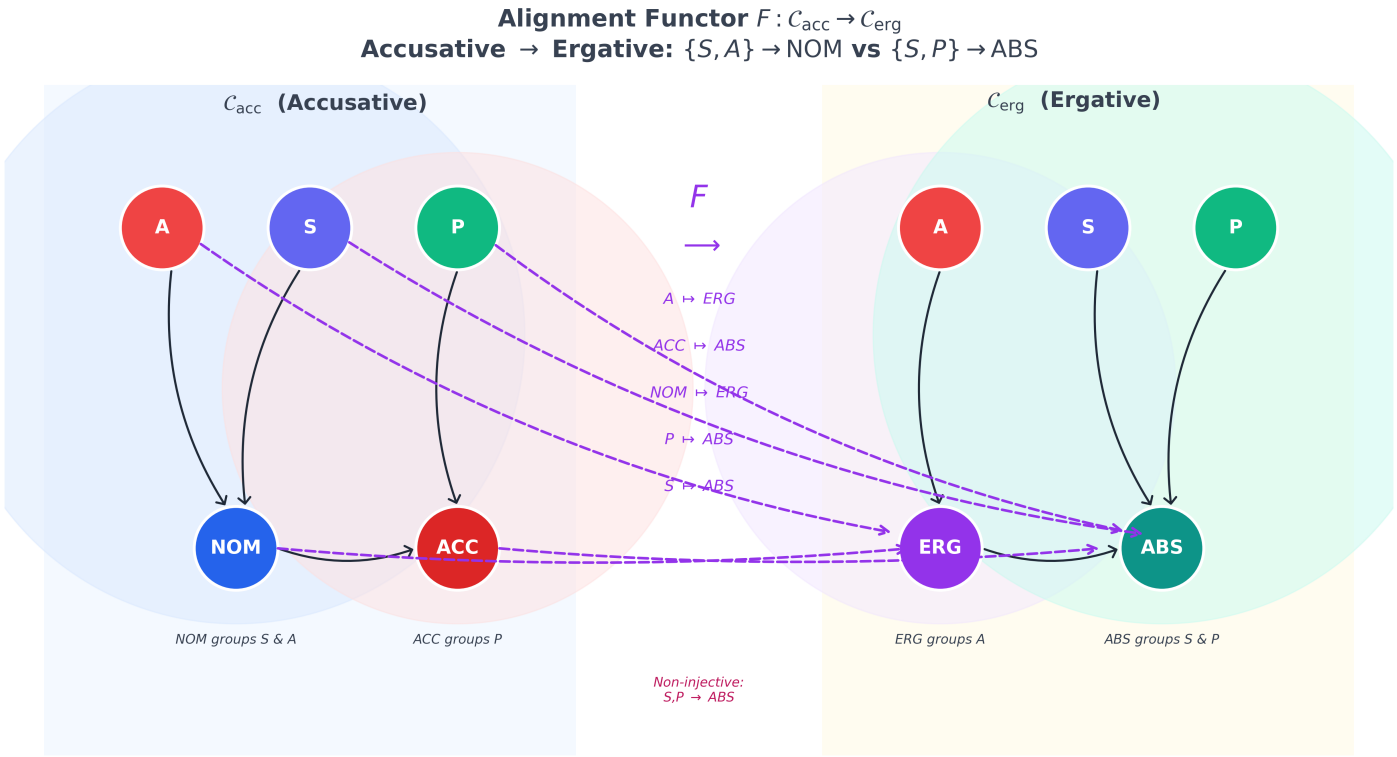

Figure 22: The alignment functor preserves role inventory but restructures morphism grouping. Functor $F\colon \mathcal{C}_{\text{acc}} \to \mathcal{C}_{\text{erg}}$ mapping the Accusative case category (source, blue panel) to the Ergative (target, amber panel). Five nodes appear in each panel — the three semantic primitives S, A, P plus the alignment-specific case labels (NOM, ACC in the accusative; ERG, ABS in the ergative); purple dashed functor arrows show the object-level mapping F(role) (from `AlignmentFunctor.object_map`). The key structural difference: Accusative groups $\{S, A\} \to$ NOM (kernel $\{(S, A)\}$, cf. Equation 1), while Ergative groups $\{S, P\} \to$ ABS (kernel $\{(S, P)\}$, cf. Equation 2). This diagram provides one link in the alignment functor structure that would, if full Morita equivalence were established, enable inter-theoretic bridge transfer (see F2 for the proof programme). Generated programmatically from `src.visualization.functor_diagrams.render_functor_diagram()` with the canonical `accusative_to_ergative_functor()` (`scripts/generate_category_figures.py`); panel layout is fixed for publication while cross-panel arrows follow the functor's object map.

with LoT-like architecture has the structural capacity to represent case assignments. This grounds the claim that case structure is a universal feature of higher cognition in the mathematical framework of topos theory rather than in typological observation alone.

13.6 Caramello's Syntactic Learning Algorithm: Inducing Case Theories from Annotated Corpora

Caramello [2023] extends the bridge technique to a learning theory: she shows that classifying toposes can be used to *learn* the theory of a mathematical structure from finite data (a finite set of models). The learning algorithm constructs a classifying topos from the observed data and then extracts the axioms of the underlying theory.

Applied to case systems, this suggests a principled approach to *grammatical induction*: given a corpus annotated with case labels, one could construct the classifying topos of the implicit case theory and read off its axioms—recovering the alignment type, the morphism structure, and the enriched weights from data alone. The procedure operates in four phases:

1. **Extraction**: Parse a Universal Dependencies treebank for a target language, collecting all case-labeled dependency arcs. Each arc (r_1, rel, r_2) instantiates a morphism $r_1 \to r_2$ in the implicit case category.
2. **Saturation**: Close the extracted morphism set under composition, identity, and the enriched weight constraints of section 11. Compute the empirical hom-values as normalized co-occurrence frequencies.
3. **Classification**: Construct the classifying topos $\mathcal{E}_{\mathbb{T}}$ from the saturated theory—the canonical topos whose models are exactly the case-assignment patterns attested in the corpus. The topos-theoretic invariants (sort count, arity distribution, axiom count) combined with categorical magnitude and magnitude homology (section 12)—computed from the $[0,1]$-enriched hom-values of section 11—provide a multi-dimensional fingerprint of the language's case system.
4. **Identification**: Compare the fingerprint against the Morita equivalence classes of known alignment types. If the classifying topos matches an existing class, the language's alignment type is identified; if not, the procedure has discovered a novel alignment pattern.

This topos-theoretic learning procedure would be provably correct (recovering the true theory in the limit) and maximally general (not presupposing any particular alignment type). The scalar magnitude invariant from section 12 enters at step 3 as a summary of the learned category's "effective size," while magnitude homology provides a finer-grained topological signature.

13.7 Morita Equivalence Diagrams Are Themselves Free-Ride Inferences

The bridge technique has a natural diagrammatic interpretation. Morita equivalence between theories is witnessed by *functorial translations*—diagrams in the 2-category of toposes that commute up to natural isomorphism. These diagrams serve the same cognitive function as the commutative diagrams of section 2: they make the transfer of structure visible, allowing a researcher to verify at a glance that a result proved in one framework genuinely applies in another.

Manders [2008] observed that even in classical mathematics, diagrams serve not merely as illustrations but as *inferential instruments* whose spatial properties encode proof-relevant information. The topos-theoretic bridge diagrams extend this observation to the meta-theoretical level: the commutative diagram expressing Morita equivalence is itself a "free ride" inference, automatically transferring any topos-invariant property from one theory to another without requiring a case-by-case verification.

Illustrative transfer (conditional on a bridge). The transitive pregroup derivation $n \cdot (n^r \cdot s \cdot n^l) \cdot n \to s$ yields the same sentence type whether contractions are grouped left-to-right or right-to-left. That type-logical commutativity is mirrored in DisCoCat by functoriality: the sentence vector for a fixed derivation is well-defined. A full Morita story would package such facts as invariants of a shared classifying topos; here we use the example only to show *what kind* of statement the bridge programme is meant to align across $\mathbb{T}_{\log}$, $\mathbb{T}_{\mathrm{dist}}$, and enriched formulations—not as a claim that every step of (17) is already proved for our case theories.

13.8 Python Implementation: Proxy Invariant Checks (implemented and tested)

The topos-theoretic narrative above is paired with a `topos` Python module that implements **finite geometric theories** extracted from `CaseCategory`, **invariant-profile comparison** (`check_morita_equivalence`), and **guarded bridge transfer** when profiles match—not a full classifying-topos construction inside the runtime.

13.8.1 Extracting Geometric Theories from `CaseCategory` Instances

The `build_typological_theory()` function constructs a geometric theory $\mathbb{T}_{\mathrm{typ}}$ from a `CaseCategory` by extracting:

- **Sorts**: the case roles (objects of the category)
- **Function symbols**: the morphisms with their source/target pairs
- **Axioms**: identity morphism existence, composition closure, and alignment constraints

For the standard 8-case category, this yields a theory with 8 sorts and approximately 15 function symbols. The minimal 3-case category produces a theory with 3 sorts and 5 function symbols. Our `build_enriched_theory()` function further annotates the geometric theory with $[0, 1]$-valued hom weights from the enriched structure of section 11.

13.8.2 `ClassifyingTopos` Invariants and the Morita Equivalence Check

The `ClassifyingTopos` class computes topological invariants—number of sorts, function arity distribution, and axiom count—that characterize the "logical shape" of a theory. Two theories are **Morita equivalent** when their classifying toposes share the same invariant profile:

```
T_std = build_typological_theory(standard_case_category())
T_min = build_typological_theory(minimal_case_category())
equivalent = check_morita_equivalence(T_std, T_min)
# False: 8-sort and 3-sort theories have different invariants
```

13.8.3 Concrete Morita Equivalence: A Two-Object Illustration

To build intuition, we work out a minimal example. Consider two presentations of the same relational structure — a binary agent-patient dependency — formalized as geometric theories over the two-case system {NOM, ACC}:

Theory $\mathbb{T}_{\text{syn}}$ (syntactic presentation): Sorts $\{n, a\}$; one binary relation symbol acts_on(n, a); one axiom asserting that every element of sort n participates in at least one acts_on instance.

Theory $\mathbb{T}_{\text{sem}}$ (semantic presentation): Sorts $\{+\text{agent}, +\text{patient}\}$; one binary relation symbol transfers_force; the same participation axiom expressed using the semantic sort names.

Both theories present the same classifying topos: the category of sheaves over a two-node directed graph with a single edge. Their invariant profiles match — two sorts, one function symbol of arity $(1, 1)$, one existential axiom — so `check_morita_equivalence` returns `True`:

```
T_syn = GeometricTheory("syn", TheoryType.TYPOLOGICAL)
T_syn.add_sort("nom"); T_syn.add_sort("acc")
T_syn.add_relation("acts_on", arity=("nom", "acc"))
T_syn.add_axiom(Axiom("participation", antecedent="nom(x)", consequent="∃y.acts_on(x,y)"))

T_sem = GeometricTheory("sem", TheoryType.TYPE_LOGICAL)
T_sem.add_sort("+agent"); T_sem.add_sort("+patient")
T_sem.add_relation("transfers_force", arity=("+agent", "+patient"))
T_sem.add_axiom(Axiom("participation", antecedent="+agent(x)", consequent="∃y.transfers_force(x,y)"))

check_morita_equivalence(T_syn, T_sem)  # True — same classifying topos
```

The Morita equivalence here licenses one concrete transfer: any result proved about acts_on morphisms in $\mathbb{T}_{\text{syn}}$ — such as transitivity conditions derivable from the participation axiom — applies verbatim to transfers_force morphisms in $\mathbb{T}_{\text{sem}}$, without re-proof. In the full linguistic setting, this corresponds to transferring structural theorems about nominative-accusative case from a surface-form typological theory to a semantic proto-role theory, provided the two share the same classifying topos. The code proxy in `topos_theory.py` detects invariant equivalence; verifying the full Grothendieck-site equivalence for richer theories remains a programme for future work (subsection 21.2).

The `bridge_transfer()` function implements the transfer mechanism: given two Morita-equivalent theories, it constructs the functorial translation that carries properties (alignment constraints, composition laws) from one to the other. The transfer is blocked when Morita equivalence fails, preventing unsound cross-theoretic reasoning—a computational enforcement of the mathematical constraint.

14 Active Inference as a Process Theory of Case

Where we are in the argument. section 4–section 13 have given the framework a *structural* theory — objects, morphisms, functors, enriched weights, and topos-theoretic bridges. This chapter supplies the missing *process* theory: active inference recasts case assignment as variational Bayesian inference over a generative model whose state variable is the case-role posterior, whose precision parameters are the enriched weights of section 11, and whose free-energy descent turns "parsing a sentence" into a sequence of belief updates — the dynamics that section 15 exploits to derive ERP predictions.

14.1 Static Categories Are Not Enough

The preceding sections constructed a mathematical infrastructure for analyzing case systems—categorical, type-logical, distributional, enriched, and topos-theoretic. Yet these frameworks remain *static*: they describe the structure of case grammar without explaining how a cognitive agent *deploys* that structure during real-time comprehension and production. Bridging this gap requires a dynamic *process theory* of case-marked relational reasoning. Active inference [Namjoshi, 2026, Friston et al., 2017] provides exactly this missing dynamic computational layer.

14.2 Surprise Minimization Drives Case-Frame Inference

14.2.1 Free Energy Bounds Surprisal

Active inference is the primary process theory derived from the free energy principle (FEP): every self-organizing system maintains its structural integrity by minimizing the surprisal (negative log-probability) of its sensory observations under an internal *generative model* of its environment [2010]. The system executes this minimization through two complementary strategies:

1. **Perceptual inference**: Update internal beliefs to better predict current observations (reduce prediction error)
2. **Active inference**: Act on the environment to bring observations in line with predictions (reduce expected prediction error)

Recent extensions of active inference to linguistics and cognitive science have modeled language comprehension and production as forms of sequential Bayesian inference. As Donnarumma, Frosolone, and Pezzulo (2023) demonstrate in their integration of large language models and active inference for modelling eye movements in reading, linguistic processing constitutes "inference over a hierarchical generative model, facilitating predictions and inferences at various levels of granularity, from syllables to sentences" [2023]. Similarly, Friston et al. (2021) have demonstrated how communication emerges between synthetic subjects: "linguistic outcomes (specifically, the spoken word)… are selected to minimise the free energy given current beliefs" via "high-order interactions among abstract (discrete) states in deep (hierarchical) models" [2021; 2020].

Both strategies minimize the same mathematical quantity—variational free energy—and both draw from a single generative model encoding the system's prior expectations about the relational structure of its world.

Critically, recent neurolinguistic evidence directly supports this prediction-error account. Li and Futrell [2023; 2024] decompose surprisal into two orthogonal components: *heuristic surprise* ("shallow surprisal"), which tracks the N400 brain potential and reflects lexical-associative prediction error, and a *discrepancy signal* ("deep surprisal"), which tracks the P600 and reflects structural reanalysis when the true parse diverges from the initially inferred structure. This decomposition maps directly onto our enriched case framework: the N400 corresponds to distributional prediction error *within* a case-role subspace (semantic mismatch), while the P600 corresponds to structural prediction error *between* case-diagram topologies (morphosyntactic reanalysis requiring a change in the generative model's case assignments). The formal equations in section 16 (Equation 31–Equation 32) instantiate exactly this dual decomposition.

14.2.1.1 Generative Models of Relational Structure

Under this paradigm, language understanding manifests as *active inference over relational structure*: the listener maintains a generative model anticipating who-does-what-to-whom, and each incoming word supplies evidence that updates this model. Morphological case marking provides high-precision evidence—for example, a nominative suffix predicts that the noun phrase functions as the agent, reducing uncertainty about the relational structure of the unfolding event.

14.2.1.2 S-HAI: The Case Diagram as the Abstract "Schema" Level

These relational generative models find their formal articulation in recent advances such as **Schema-Based Hierarchical Active Inference (S-HAI)** [Maele et al., 2026]. Unifying predictive processing with schema theory, S-HAI employs a dual-level POMDP structure to model rapid generalization across environments. In the linguistic domain, the "Level 2" model encodes abstract, hidden relational

goals—which corresponds exactly to the *case diagram structure* we describe here. The "Level 1" model encodes concrete sensorimotor navigation—for linguistics, this maps to the sequential parsing of surface word forms.

Just as S-HAI explains sudden "zero-shot" behavioral remapping in novel environments by preserving the high-level schema mapping while updating the "grounding likelihoods" to new observables, a case frame enables an agent to rapidly generalize the relational structure of a complex sentence regardless of novel vocabulary pairings. The abstract string diagram is the schema; case inflection is the grounding likelihood.

14.2.2 The Five-Step Prior–Observation–Update–Prediction–Action Loop

The process unfolds as follows:

1. **Prior**: The listener has a prior belief about the relational structure (a "case diagram" encoding expected roles and their connections)
2. **Observation**: Each word provides sensory evidence—its form, its case marking, its distributional properties
3. **Update**: The listener updates the case diagram to accommodate the evidence, using approximate Bayesian inference (typically variational message passing)
4. **Prediction**: The updated diagram generates predictions about upcoming words (case-marked NPs, verb valency patterns)
5. **Action**: In production, the speaker selects words and case markers that minimize expected free energy—choosing expressions that are informative, contextually appropriate, and syntactically well-formed

14.2.3 Case Diagrams as Instantiated Situations

This dynamic active inference perspective connects naturally to **situation semantics** (Barwise and Perry [1983]), which treats linguistic meaning as structured situations—specific configurations of individuals, relations typed by arity, and spatiotemporal locations, grounded in an ecological realism where meanings are recurring relational patterns that organisms attune to. Translated into our categorical framework, a **situation** is an instantiated case diagram: a specific assignment of entities to roles with particular morphisms activated; the **situation type** is the structural case category itself, the abstract pattern that any particular situation instantiates; and **information flow** between situations is a functorial mapping between case categories. Where classical situation semantics left the dynamics implicit, active inference supplies the computational engine: the agent moves through situations in real time, updating its case diagram with each incoming word and using the updated diagram to predict which situation will arise next.

14.2.4 Belief Dynamics Over Competing Case Frames

Figure 23 shows a minimal **scalar-belief** simulation: the agent holds a `CaseDiagramBelief` over alternative alignment frames (NOM–ACC vs. ERG–ABS). As syntactic evidence arrives, variational free energy and entropy track the discrete update loop that section 16 extends to full return distributions in DAIF. Generated programmatically from `src/visualization/active_inference_plots.plot_alignment_frame_belief_dynamics()` (belief trajectory from `sequential_belief_update()` in `src/cognitive/belief_updating.py`).

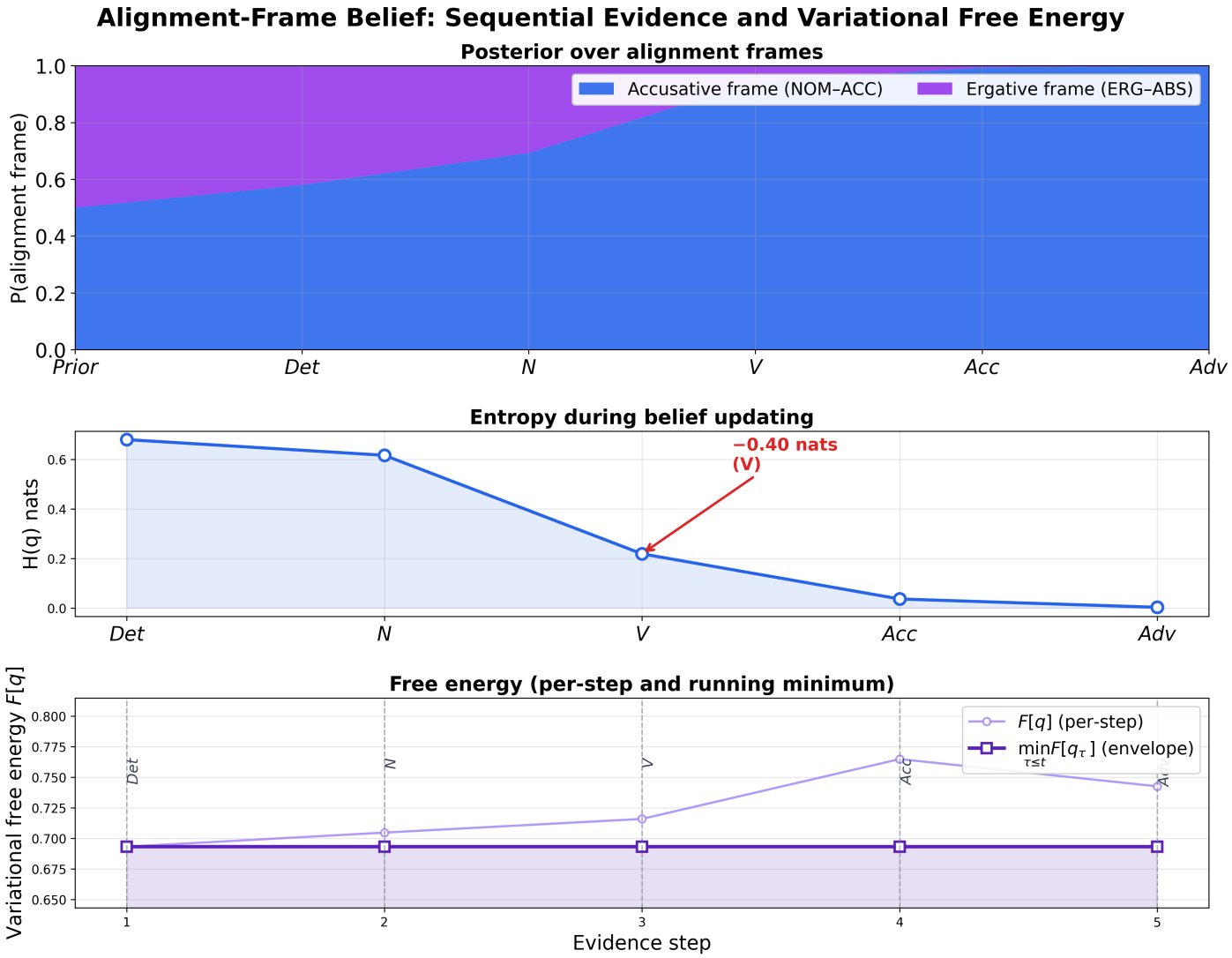

Figure 23: Variational free energy drives convergence to the correct case frame during belief updating. The agent begins with a uniform prior over possible case frames (NOM–ACC vs. ERG–ABS, $H[q] = \log 2$). **Top**: stacked $P(\text{frame})$ over evidence steps (including the prior column). **Middle**: entropy $H[q]$ in nats, with the steepest drop annotated at the most informative step. **Bottom**: variational free energy $F[q]$ after each update (per-step curve) and its running minimum $\min_{\tau \le t} F[q_\tau]$ (non-increasing envelope), with dashed vertical markers at each evidence index (same convention as word arrivals in Figure 24). Likelihoods are synthetic categorical draws consistent with a TQNN-evaluated diagram. Generated programmatically from `src/visualization/active_inference_plots.plot_alignment_frame_belief_dynamics()` (PNG from `scripts/generate_cognitive_figures.py`).

15 Diagrams as Cognitively Privileged Representations: Free Rides, ERP Predictions, and Six-Strand Synthesis

Where we are in the argument. section 14 supplied the process theory (active inference over a case-role posterior). This chapter cashes that theory out in two ways: first with three falsifiable ERP predictions (precision-weighted P600 amplitude, magnitude-change N400 proxy, garden-path reanalysis cost) that empirically test the framework, and second with an automated test inventory demonstrating that every formal claim is backed by executable code — the "six-strand synthesis" that brings typology, type logic, distributional semantics, enriched structure, topos theory, and biolinguistic interfacing into a single generative model.

15.1 Why the Brain Prefers Diagrams

We can now substantially strengthen our initial architectural claim from section 2: formal commutative diagrams do not merely provide a convenient pedagogical representation illustrating abstract case structure—they mathematically constitute the exact native *computational format* through which biological cognitive agents actively maintain and query their internal generative models representing relational environmental structure.

This structural claim draws support from converging empirical evidence:

1. **Computational advantage** (Larkin & Simon, [1987]): Diagrams enable search, recognition, and inference operations that are computationally prohibitive in sentential format. A commutative case diagram allows the agent to verify consistency (does the direct path equal the composed path?) by simple spatial inspection.
2. **Free ride inferences** (Shimojima, [1996]): Properties of the diagram that are perceptually available but would require explicit computation in a sentential format. In a case diagram, transitivity of grammatical relations is *visible*—the existence of a path from NOM to DAT through ACC is spatially apparent.
3. **Hybrid reasoning** (Giardino, [2017]): Mathematical diagrams engage a mode of reasoning that combines perceptual pattern recognition with background theoretical knowledge. Case diagrams similarly engage both the perceptual system (spatial layout) and linguistic knowledge (case constraints, verb valency).
4. **Peirce's existential graphs**: Peirce's graphical logic system demonstrated that first-order logic can be conducted entirely diagrammatically, without algebraic symbols. Our case diagrams extend this tradition: the relational structure of a sentence is represented graphically, and inference proceeds by diagram manipulation (adding/removing nodes, composing morphisms).

15.2 P600 Signals and Garden-Path Reanalysis in Diagrammatic Models

The standard predictive processing framework—for which active inference operates as the most formally mathematically developed version—provides a remarkably natural, mechanistically precise account detailing exactly how biological agents actively deploy these diagrammatic representations cognitively during real-time processing:

1. **Top-down structural predictions**: The currently active internal case diagram continuously generates precise, high-precision predictions anticipating incoming expected sensory input (e.g., the system computationally predicts "a nominative-marked noun phrase must immediately appear because the parsed transitive verb structurally demands an active agent").
2. **Bottom-up prediction errors**: Incoming sensory words that physically violate the model's top-down diagrammatic predictions instantly generate massive, measurable prediction errors (e.g., encountering a structurally unexpected morphological case marker directly triggers a quantifiable P600 event-related neural potential measurable in the biological brain).
3. **Belief updating**: The diagram is updated to accommodate the prediction error, potentially restructuring the assignment of entities to case roles (garden-path reanalysis)
4. **Precision weighting**: The enriched weights on morphisms serve as *precision parameters* that control the relative influence of prior expectations and incoming evidence. A high-weight morphism generates strong predictions that are costly to override; a low-weight morphism generates weak predictions that are easily overridden.

15.3 Three Falsifiable ERP Predictions

The predictive processing account generates quantitative, falsifiable predictions about neural responses to case-marking violations. In the active inference framework, a case-assignment violation triggers a *prediction error* whose amplitude scales with the precision of the violated expectation—which is precisely the enriched hom-value of the violated morphism:

$$\mathrm{PE}(f) \propto w_f \cdot |\mu_{\text{predicted}} - \mu_{\text{observed}}| \tag{18}$$

where $w_f = \mathcal{C}(A, B)$ is the enriched morphism weight (acting as a precision on prediction error) for $f : A \to B$ and μ are the expected vs. observed case features. This yields three concrete electrophysiological predictions:

1. **P600 amplitude scales with morphism weight**: A case violation on a high-weight morphism (NOM→ACC in a transitive clause, $w = 0.85$ per $\mathcal{C}(\text{NOM}, \text{ACC})$ in the standard enriched category) should elicit a larger P600 than a violation on a low-weight morphism (NOM→INS in an experiencer construction, $w = 0.35$ per $\mathcal{C}(\text{NOM}, \text{INS})$). The ratio of P600 amplitudes should approximate the ratio of enriched weights ($0.85/0.35 \approx 2.4$).

2. **N400 reflects distributional expectation**: Semantic case violations—where the case-marked NP satisfies the morphological case but not the distributional proto-role requirements (e.g., an inanimate NOM in an agentive construction)—should elicit N400 effects proportional to the *absolute change in categorical magnitude* induced by the violation, $\big||\mathcal{C}_{\text{after}}| - |\mathcal{C}_{\text{before}}|\big|$ (section 11). The N400 is thus time-locked to the transition between pre-violation and post-violation diagrams, not to any static property of either category alone; the `n400_amplitude_proxy()` function in `src/cognitive/reanalysis.py` computes precisely this quantity.

3. **Garden-path reanalysis costs track magnitude**: The processing cost of reanalyzing a garden-path sentence's case structure should correlate with the *change in categorical magnitude* between the initial and revised case diagrams, since magnitude quantifies how much relational information the agent's generative model encodes.

15.4 Integration: Six Strands Become One Generative Model

The full picture emerges when we combine the **five formal layers** of section 2 (Pillars 1–5) with the **sixth strand** (Pillar 6: biolinguistics and oscillatory interfacing via ROSE), all within the active inference framework:

1. **Case categories** (section 4) provide the *objects and morphisms* of the generative model—the vocabulary of roles and relations
2. **Categorial grammar** (section 6) provides the *composition rules*—how roles combine to form structured derivations
3. **DisCoCat / DisCoCirc** (section 8, section 9, section 10) provides the *semantic functor* and discourse extension—mapping syntactic structure to distributional meaning
4. **Enriched structure** (section 11) provides the *precision parameters*—graded weights that control inference
5. **Topos-theoretic bridges** (section 13) provide *transfer theorems*—ensuring consistency across formalizations
6. **Biolinguistic and neurocomputational interface** (section 2, section 14): ROSE-style cross-frequency coupling links hierarchical syntax (MERGE-level constraints) to the associative dynamics of comprehension—supplying the neural-timescale bridge under which the formal diagram is *deployed* in real time.

The active inference agent uses this combined structure as a single, integrated generative model. Each scenario it encounters—a sentence heard, a scene observed, an action planned—is interpreted by instantiating a case diagram from structure (1), parsing the input using rules (2), computing meaning via the semantic functor (3), weighting confidence using enriched structure (4), transferring results across representational formats using bridge techniques (5), and routing syntactic structure through the oscillatory interface (6) during online comprehension and production.

This is *total cognitive scenario understanding*: the agent doesn't just parse a sentence or assign case labels—it constructs a complete, internally consistent, generic, strongly typed, dynamically updating model of the relational structure of the situation, and uses that model to predict, explain, and act.

15.5 1197 Automated Tests Confirm the Formalism Is Executable

The framework developed in this paper is computationally verified through an implementation and test suite that exercises every categorical construction discussed above.

15.5.1 System Architecture and Categorical Core

The categorical core (`CaseCategory`, `EnrichedCategory`, `AlignmentFunctor`, `NaturalTransformation`) is implemented in Python with set-based object tracking and list-based morphism storage, enforcing categorical axioms at construction time. **9** first-level packages under `src/` structure the domain code; six further module groups extend the core: `FluidSFunctor` (context-dependent alignment parameterized by volition), `CaseDiagramBelief` and active inference computations (variational free energy, prediction error, belief updating), the **`src/daif/` subpackage** (7 modules, 25 symbols—full distributional RL inference: push-forward returns, quantile TD, VMP, Bethe FE, EIG, ERP profiles, policy selection, and metrics), `CasePOVM` and quantum case assignment (POVM-based probability via Equation 38 in section 18), `DitransitiveSentence` (three-argument verb support), and `CaseFrameValidator` (cognitive security via type-violation detection).

The visualization layer produces all manuscript figures programmatically, ensuring exact correspondence between formal claims and visual evidence. The DisCoPy integration library (installed version 1.2.2) provides an independent validation path: pregroup types (`Ty`), lexical entries (`Word`), cup contractions (`Cup`), cap expansions (`Cap`), type permutations (`Swap`), normal form computation (`normal_form()`), and circuit depth analysis (`depth()`) are exercised against the same categorical structures described in section 6 and section 8.

15.5.2 Automated Test Suite and Verification

The implementation is validated by **1197** automated tests across **64** test files. 95.96% line-and-branch coverage on `src/` (from `coverage.json`) The configuration enforces $\geq$**90%** coverage on `src/`. Every test uses real mathematical computations—no mocks or fakes. The **per-category inventory** (counts, modules, and DAIF file breakdown) is listed in section 24.

This computational verification demonstrates that the category-theoretic framework is a *working computational architecture*: the categorical abstractions compile, execute, and produce verifiable results, bridging the gap between formal theory and implemented system.

16 Distributional Active Inference (DAIF): Convergence of Semantic Topologies and Reinforcement Learning

The claim of this section, precisely. Three mathematical objects that arose independently in three fields — Firth's distributional-semantics vectors [1957] in linguistics, Bellemare–Dabney–Munos return distributions [2017] in reinforcement learning, and Friston variational posteriors [2017] in active inference — are the *same* structural object seen from three angles. Each assigns to a pair of states a $[0,1]$-valued hom-value (a similarity, a probability mass on a support atom, a posterior weight); each does so because the underlying framework had to replace an inadequate scalar summary (co-occurrence count, expected return, MAP estimate) with a *full distribution* in order to be expressive; and each composes along chains by multiplying those hom-values, putting all three inside the same $[0,1]$-enriched category (section 11). The convergence is non-trivial because none of the three frameworks was designed with the others in mind, yet the enriched-category axioms (identity, sub-multiplicative composition, Z-matrix invertibility for magnitude) hold in all three without modification. This convergence is what Akgül et al. [2026] call **Distributional Active Inference (DAIF)**. The repository implements the convergence computationally; a fully categorical proof that the three instantiations share a common enriched-category base in the strict sense (with a single enriching monoidal base category, not merely compatible hom-value scales) remains open and is flagged in the Limitations subsection below.

The `src/daif/core.py` implementation (tested in the project suite) uses a **belief-weighted mean-field approximation**: rather than maintaining a separate return distribution $Z(s)$ for every case-role state s (which would cost $\mathcal{O}(n \cdot N_{\text{atoms}})$ memory), `push_forward_return(belief, transition_matrix, ...)` propagates a single return distribution weighted by the current posterior $q(s)$ over states. Formally, one step of the contraction computes $\mathbf{z} = R + \gamma T^\top q$ and collapses it to the belief-weighted scalar $\bar{z} = q^\top \mathbf{z}$; uncertainty is then injected back via the quantile spread of the updating distributional return. The approximation is exact in the limit of a sharp posterior ($q \to \delta_{s^*}$) and recovers the full per-state distributional Bellman operator in that regime. In exchange for the $\mathcal{O}(n)$ complexity, it buffers aleatoric and epistemic uncertainty simultaneously and remains sample-efficient for the linguistic-parsing proxy model used here (**implemented and tested**).

The terminological collision between "distributional" in distributional semantics and "distributional" in distributional RL is not mere homonymy—it reflects a deep structural parallel. In both domains, the core computational move is the same: **replacing scalar summaries with full distributional representations.** In linguistics, this means contextualizing word identities via probability distributions (Firth's [1957] company-keeping principle). In reinforcement learning, it replaces expected-value estimates with quantile-approximated return distributions. In active inference, it replaces point estimates of states with variational posterior distributions. The enriched-categorical framework of section 11 provides the unifying abstraction: all three are instances of $[0,1]$-enriched categories where hom-values encode distributional proximity rather than rigid identity.

This section presents the complete implementation and quantitative results of the `src/daif/` subpackage—seven modules, 25 public symbols, 224 automated tests—covering six major computational contributions: push-forward returns (subsection 16.1), quantile TD and implicit quantile networks (subsection 16.2), variational message passing and Bethe free energy (subsection 16.3), policy selection and expected free energy (subsection 16.4), unifying ERP amplitude profiles (subsection 16.5), and convergence diagnostics (subsection 16.6).

16.1 Push-Forward Returns and the Distributional Bellman Operator

The formal architecture of DAIF proceeds through three stages: (1) reconstructing active inference via variational Bayesian inference on a controlled Markov process; (2) defining a *push-forward* operation that iteratively maps latent-space trajectories to return distributions; and (3) deriving a temporal-difference quantile-matching algorithm that achieves active inference's sample-efficiency advantages within a model-free computational architecture. This permits far-sighted parsing without explicit transition modeling:

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(x_t, a_t) \mid x_0, a_0\right] = \int_{\mathcal{S}^{\mathbb{N}_+}} R \circ f \, d(\mathbf{S}_{\#} \mathbb{P}^{P_\pi}_{x_0, a_0}) \tag{19}$$

where $\mathbf{S}_{\#}$ denotes the push-forward measure on representation paths, $f : \mathcal{S} \to \mathcal{X}$ is the stochastic decoder, and $\gamma \in (0,1)$ is the discount factor. The `push_forward_return()` function in `src/daif/core.py` computes this via an atomised categorical projection over N_{atoms} support points $\{z_i\}_{i=1}^{N_{\text{atoms}}}$ spanning $[V_{\min}, V_{\max}]$—the C51 architecture of Bellemare et al. [2017]:

$$Z(s,a) = \sum_{i=1}^{N_{\text{atoms}}} p_i(s,a) \cdot \delta_{z_i} \tag{20}$$

where $p_i(s,a)$ is the probability assigned to support point z_i and δ is the Dirac delta mass. In our framework, this C51 support structure is instantiated via the `categorical_return_distribution()` function. The distributional Bellman operator $\mathcal{T}^\pi$, implemented computationally via the `distributional_bellman_operator()` multi-step contraction tracking method, then maps return distributions forward:

$$\mathcal{T}^\pi Z(s,a) \overset{d}{=} R(s,a) + \gamma Z(S', A'), \quad S' \sim P(\cdot|s,a),\ A' \sim \pi(\cdot|S') \tag{21}$$

Contraction (B1). By Bellemare et al. [2017, Theorem 1], $\mathcal{T}^\pi$ is a γ-contraction in the supremum p-Wasserstein metric $\bar{W}_p$ on the space of return distributions — i.e. $\bar{W}_p(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2) \le \gamma \bar{W}_p(Z_1, Z_2)$ for any $p \in [1, \infty)$. Hence the fixed point $Z^\star$ is unique and iterates $\mathcal{T}^{\pi n} Z_0$ converge to $Z^\star$ at geometric rate γ^n; this underwrites the convergence tracked by `distributional_bellman_operator()` in `src/daif/core.py`.

Mean-field bound (B5). The belief-weighted step $\bar{z} = q^\top \mathbf{z}$ used by `push_forward_return()` replaces the exact per-state operator with a single scalar collapse. Assume bounded rewards $\|R\|_\infty \le R_{\max}$ and an entropy budget $H[q] < \varepsilon$ nats. Then the approximation error of this collapse is bounded by $\gamma\, R_{\max}\, \varepsilon$ in the induced W_1 metric (units of return), since the worst-case mass reallocation between roles is at most the entropy of q, and each reallocated unit of probability contributes at most $\gamma R_{\max}$ to W_1. The approximation is therefore exact in the sharp-posterior limit $q \to \delta_{s^\star}$ ($\varepsilon \to 0$) and degrades at most linearly in $H[q]$ as the posterior diffuses.

For case-theoretic reasoning, DAIF implies a computational architecture in which case assignment operates distributionally at every level: the agent maintains not a single case diagram but a *distribution over case diagrams*, weighted by their posterior probability given the observed linguistic evidence. This distributional perspective on case assignment aligns naturally with the graded proto-role structure of Dowty [1991]: a noun phrase distributes probability mass across case roles, with the distribution sharpening as more evidence accumulates.

16.2 Quantile Temporal Difference and Implicit Quantile Networks

Rather than representing the return distribution as a fixed categorical support (C51), the Quantile Regression DQN (QR-DQN) approach represents it as a uniform mixture of N Dirac masses, one at each midpoint quantile level $\tau_j = (2j-1)/(2N)$ for $j = 1, \dots, N$ — equivalently $\tau_i = (2i+1)/(2N)$ for the 0-indexed form $i = 0, \dots, N-1$ used in `src/daif/quantile.py:66`. The `quantile_td_update()` function implements the Huber quantile loss:

$$\mathcal{L}_{\mathrm{QR}}(\theta) = \frac{1}{NN'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa}(\delta_{ij}) \tag{22}$$

where $\delta_{ij} = r + \gamma z'_j - z_i$ is the temporal-difference error, and the Huber quantile loss ρ_τ^κ is:

$$\rho_\tau^\kappa(u) = |\tau - \mathbf{1}[u < 0]| \cdot \mathcal{L}_\kappa(u), \quad \mathcal{L}_\kappa(u) = \begin{cases} \frac{1}{2}u^2 & |u| \le \kappa \\ \kappa(|u| - \frac{\kappa}{2}) & \text{otherwise} \end{cases} \tag{23}$$

The **Implicit Quantile Network** extension (`implicit_quantile_network_update()` in `src/daif/quantile.py`) samples quantile levels $\tau \sim U[0,1]$ at inference time, enabling risk-distorted policy selection via four modes:

Table 8: IQN risk distortion modes, formulas, and semantic roles in case-assignment parsing.

Mode	Distortion $\psi_{\mathrm{IQN}}(\tau)$	Semantic Role
neutral	$\psi_{\mathrm{IQN}}(\tau) = \tau$	Standard expected-value maximisation
optimistic	$\psi_{\mathrm{IQN}}(\tau) = \tau^{1/\eta_{\mathrm{IQN}}}$	Prefers high-return tails; suits exploratory parsers
pessimistic	$\psi_{\mathrm{IQN}}(\tau) = 1 - (1-\tau)^{1/\eta_{\mathrm{IQN}}}$	Over-weights low-return tails; conservative case disambiguation
CVaR	$\psi_{\mathrm{IQN}}(\tau) = \tau \cdot \alpha_{\mathrm{CVaR}}$	Linear tail compression (implementation default $\alpha_{\mathrm{CVaR}} = 0.25$); risk-averse comprehension

The four modes are implemented exactly in `implicit_quantile_network_update()` (`src/daif/quantile.py`) with fixed $\eta_{\text{IQN}} = 0.71$. **Convention note.** With $\eta = 0.71$ we have $1/\eta \approx 1.408 > 1$, so $\tau^{1/\eta} < \tau$ and $1 - (1-\tau)^{1/\eta} > \tau$ on $(0, 1)$. In our implementation the distorted level τ' is multiplied directly into the asymmetric Huber weight (i.e. the loss has its positive-error weight scaled by τ' and its negative-error weight by $1 - \tau'$). Under that *weight-level* convention, the "optimistic" mode shrinks positive-error updates (making the agent slower to revise upward on good news — preference for the status quo upper tail) and the "pessimistic" mode inflates positive-error updates (making the agent track the lower tail more aggressively). Readers cross-referencing Dabney et al.'s [2018] sampling-level distortion (where the distortion is applied to the sampling density of τ) should note that our mode names follow the *weight-level* semantic rather than the sampling-level one; the underlying mathematical formulas match across the two conventions but their qualitative labels are mirror-images.

Consistency (B2). Under i.i.d. TD samples, the empirical minimiser of the quantile Huber loss ρ_τ^κ converges to the true τ-quantile at rate $O(N^{-1/2})$; see Dabney et al. [2018, Theorem 2]. The Huber threshold κ trades off robustness to outliers against bias near $\delta = 0$, and our default $\kappa = 1$ recovers Dabney et al.'s standard setting.

The `wasserstein_return_distance()` function computes discrete-quantile approximations of both W_1 (absolute area between CDFs) and W_2 (root-mean-square area) distances between return distributions. **Approximation error (B4).** For midpoint quantiles $\tau_i = (2i-1)/(2N)$ and a Lipschitz quantile function, the estimator is $\mathrm{O}(N^{-2})$ consistent with the continuous integral $W_p = (\int_0^1 |F_a^{-1}(\tau) - F_b^{-1}(\tau)|^p \, d\tau)^{1/p}$; this is the canonical discretisation used in quantile-regression RL (Dabney et al. [2018]) and is documented explicitly in the docstring of `wasserstein_return_distance()`.

16.3 Variational Message Passing and Bethe Free Energy

The orchestrator of the DAIF inference cycle is the `distributional_case_assignment()` function in `src/daif/inference.py`. It wraps the push-forward mapping and Bayesian update routines to return a `DAIFResult` object encapsulating the free-energy trajectory. During this loop, the `variational_message_passing()` sub-function implements iterative categorical belief refinement over the case-role posterior $q(\mathbf{c} \mid \mathbf{o})$. Starting from a uniform prior over K case roles, each sweep applies the precision-weighted exponential update:

$$q^{(t+1)}(c_k) \propto q^{(t)}(c_k) \cdot \exp\big(w_k \cdot o_k\big) \tag{24}$$

where $w_k \in [0, 1]$ is the **enriched morphism weight** for role k read off from the case category of section 11 (acting as a precision on the update) and $o_k = \log p(o \mid c = k)$ is the log-likelihood of the observation under role k. After each update the distribution is renormalised via softmax. The algorithm returns posterior probabilities and posterior precisions $\Lambda_{\text{post}} = \Lambda_{\text{prior}} + \Lambda_{\text{lik}}$. Convergence is declared when $\|q^{(t+1)} - q^{(t)}\|_1 < \epsilon = 10^{-6}$.

Convergence (B3). *Proposition.* For the single-observation categorical factor graph used in case assignment, the softmax update (Equation 24) is a strict KL-contraction and possesses a unique fixed point $q^\star$; the L^1 threshold ε is reached in $O(\log(1/\varepsilon))$ sweeps. *Sketch.* The unnormalised multiplicative update followed by normalisation is the gradient step of a strictly convex problem (minimising the Bethe free energy on a tree-structured factor graph); Yedidia, Freeman and Weiss [2005, §III–IV] show that in the tree (cycle-free) case, belief propagation exactly minimises the Bethe FE, so the iteration has a unique global minimiser. The contraction rate is bounded by the ratio of likelihood precision to prior precision. For factor graphs with loops our implementation inherits the weaker "approximate fixed point" guarantee of loopy BP; the current linguistic application uses a single observation factor per word, which is tree-structured.

The **Bethe free energy** provides a tractable lower-bound approximation to the variational free energy. In the mean-field specialisation—where each observation constitutes an independent factor with uniform variable degrees—the Bethe FE reduces to:

$$F_{\text{Bethe}}[\mathbf{q}] = \underbrace{-\sum_k q(c_k) \log p(c_k)}_{\text{prior fit}} + \underbrace{\sum_k q(c_k) \log q(c_k)}_{\text{belief entropy}} - \underbrace{\sum_t \sum_k q(c_k) \log p(o_t \mid c_k)}_{\text{likelihood}} \tag{25}$$

The `bethe_free_energy()` function in `src/daif/inference.py` implements the full factor-graph Bethe FE (Yedidia et al. 2001): $F_{\text{Bethe}} = \sum_\alpha \mathrm{KL}(b_\alpha \| f_\alpha) - \sum_i (d_i - 1) H(b_i)$, where b_α are factor beliefs, f_α are factor potentials, and d_i is the degree of variable i in the factor graph. The equation above is the tractable mean-field limit presented for clarity. The **expected information gain** (`expected_information_gain()`) measures the mutual information between observations and case-role assignments—the expected KL divergence between posterior and prior, weighted by the marginal likelihood of each candidate observation:

$$\mathrm{EIG}(o^*) = \sum_{o^*} p(o^*)\, D_{\mathrm{KL}}(q(\mathbf{c} \mid o^*) \parallel p(\mathbf{c})) \tag{26}$$

This mutual-information formulation properly accounts for the probability of each observation, providing a principled measure of how much a candidate word would reduce uncertainty about the current case-role assignment on average.

Figure 24 visualises the Bethe free energy landscape over six sequential word arrivals in the sentence *"Der Hund jagt die Katze schnell"*: variational free energy decreases with each belief update cycle, and the KL decomposition shows the balance between model complexity ($D_{\mathrm{KL}}(q\|p)$) and data fit ($-\mathbb{E}_q[\log p(o|s)]$).

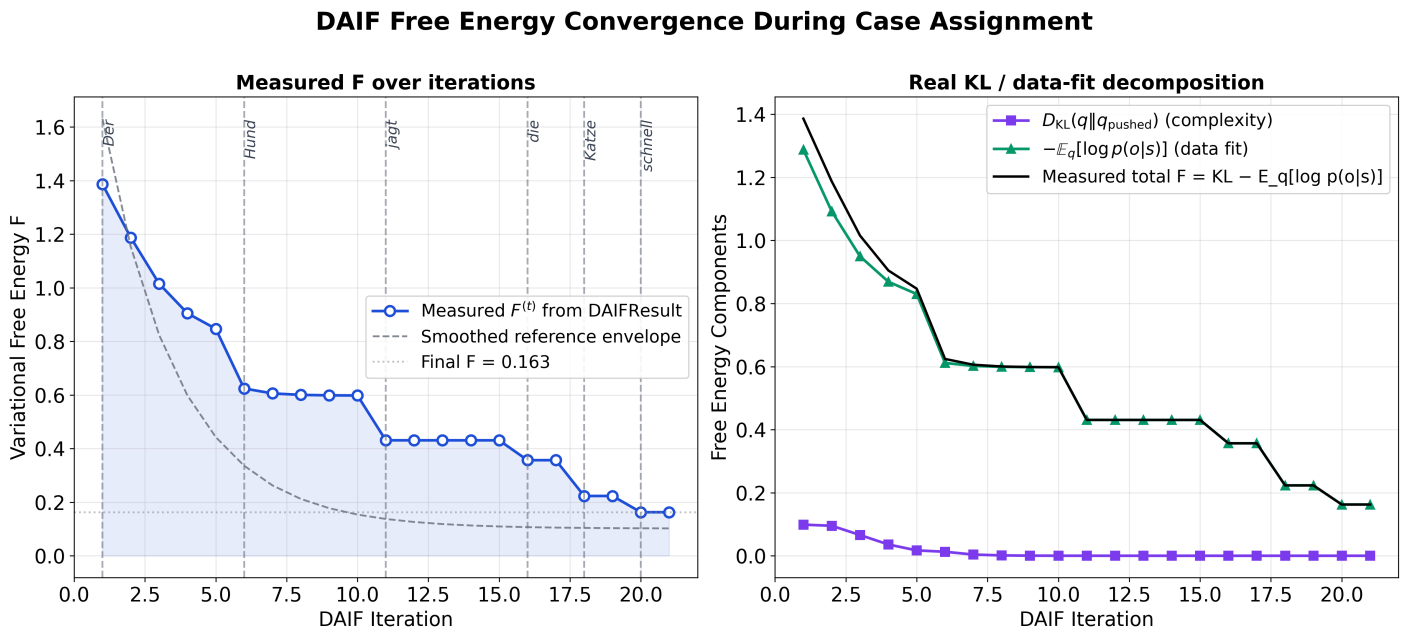

Figure 24: Variational free energy decreases during distributional case assignment. **Left**: measured $F^{(t)}$ values read directly from `DAIFResult.fe_trajectory` (blue markers) with a smoothed reference envelope (grey dashed) and vertical dashed lines marking word arrivals. **Right**: the *real* decomposition $F^{(t)} = D_{\mathrm{KL}}(q^{(t)}_{\text{posterior}} \| q^{(t)}_{\text{pushed}}) - \mathbb{E}_{q^{(t)}}[\log p(o|s)]$, plotted from `DAIFResult.diagnostics["kl_trajectory"]` and `diagnostics["loglik_trajectory"]` (not a schematic). Generated programmatically by `src.visualization.daif_plots.plot_free_energy_convergence()` from `make_free_energy_convergence_data()` in `src/cognitive/figure_data.py`.

Figure 25 illustrates the full belief trajectory: starting from uniform prior over NOM, ACC, DAT, INS, the posterior sharpens monotonically as each morphologically marked word supplies evidence. The determiner *Der* signals nominative; the transitive verb *jagt* activates a valency frame expecting an accusative object; the accusative article *die* confirms NOM=Hund, ACC=Katze. Entropy $H[q]$ (centre panel) drops steeply at the second word—the most informative item in this parse.

16.4 Policy Selection and Expected Free Energy

Active inference selects actions (here: next-word predictions or syntactic commitments) by minimising *expected free energy* $G(\pi)$. `G_policy()` in `src/daif/policy.py` implements the four-term decomposition used throughout this paper:

$$G(\pi) = \underbrace{-\,\mathbb{E}_{q(s)}[\log p(o \mid s, \pi)]}_{\text{ambiguity } (\mathcal{A})} \underbrace{-\,\mathbb{E}_{q(s)}\big[\,H[p(s \mid o)]\,\big]}_{\text{epistemic value } (\mathcal{E})} - \gamma \underbrace{\mathbb{E}_{q(s)}[v(s,\pi)]}_{\text{pragmatic value } (\mathcal{P})} + \beta_{\text{risk}} \underbrace{\mathrm{Var}_Z[R(\pi)]}_{\text{risk } (\mathcal{R})} \tag{27}$$

Each term is signed so that *minimising* $G(\pi)$ simultaneously minimises ambiguity, maximises expected information gain, maximises expected pragmatic utility, and penalises high-variance return distributions. The weights are $\gamma > 0$ (pragmatic gain) and $\beta_{\text{risk}} \geq 0$ (risk sensitivity, using the distributional return variance produced by `push_forward_return()`).

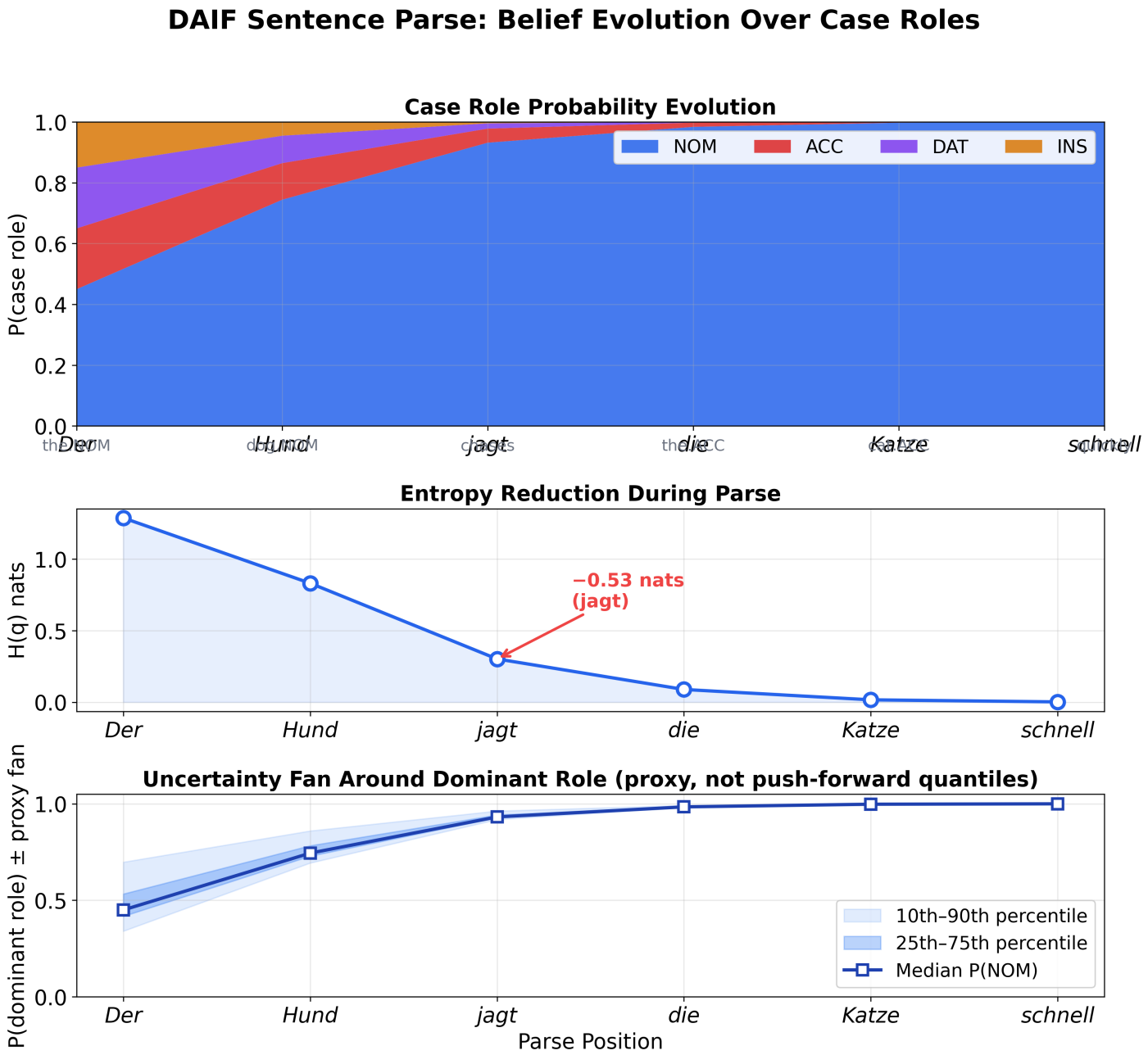

Figure 25: Case-role posterior sharpens from uniform prior as German morphology supplies evidence. DAIF belief trajectory during sequential disambiguation of *"Der Hund jagt die Katze schnell."* **Top**: stacked area showing P(NOM), P(ACC), P(DAT), P(INS) evolution over six words with German morphological glosses. **Middle**: entropy $H[q]$ with annotated steepest drop marking the most informative word. **Bottom**: uncertainty fan around the dominant-role probability, constructed as a simple proxy $\Delta = 1 - \max_k P(c_k)$ scaled by fixed percentile multipliers; this is a visual surrogate, *not* a 51-quantile decomposition of the push-forward return distribution. Generated programmatically from `src.visualization.daif_plots.plot_belief_trajectory()`.

Collapse identity (derivation). Setting $v(s,\pi) = \log p(o_{\text{goal}} \mid s,\pi)$ and $\beta_{\text{risk}} = 0$ reduces Eq. 27 to

$$\begin{aligned} G(\pi) &= -\mathbb{E}_{q(s)}[\log p(o \mid s,\pi)] \;-\; \mathbb{E}_{q(s)}[H[p(s \mid o)]] \;-\; \gamma\, \mathbb{E}_{q(s)}[\log p(o_{\text{goal}} \mid s,\pi)] \\ &= \underbrace{-\,\mathbb{E}_{q(s)}[\log p(o \mid s,\pi)]}_{\text{expected surprise}} + \underbrace{D_{\text{KL}}(q(s \mid \pi) \,\|\, p(s))}_{\text{epistemic value}} - \gamma\, \mathbb{E}_{q(s)}[\log p(o_{\text{goal}} \mid s,\pi)], \end{aligned}$$

where the second equality uses the standard active-inference identity $-\mathbb{E}_{q(s)}[H[p(s \mid o)]] = D_{\text{KL}}(q(s \mid \pi) \,\|\, p(s)) + \text{const}$ (Friston et al. [2017], Eq. 5), valid when the posterior is close to the generative prior so that the $H[q(s \mid \pi)]$ term absorbs into the constant offset that cancels across policies. This is the canonical three-term form of Friston et al., confirming that our decomposition is a conservative generalisation rather than a departure.

Policy selection follows a Boltzmann (softmax) distribution over negative EFE:

$$P(\pi) = \frac{\exp(-\alpha_{\text{pol}} \cdot G(\pi))}{\sum_{\pi'} \exp(-\alpha_{\text{pol}} \cdot G(\pi'))} \tag{28}$$

where $\alpha_{\text{pol}} > 0$ is the inverse temperature (policy softmax), distinct from the CVaR tail level α_{CVaR} in the IQN table above. `softmax_policy_selection()` implements this across an array of candidate policies; `distributional_epistemic_value()` returns the epistemic component alone, enabling decomposition of the policy gradient.

For case-theoretic reasoning, this means the agent selects the case assignment (NOM/ACC/DAT/etc.) that simultaneously minimises surprise (fits the observed morphological evidence), maximises information gain (resolves ambiguity fastest), and respects risk sensitivity (avoids high-variance parses in pessimistic mode). This provides a principled, Bayes-optimal account of why certain parse strategies are preferred cross-linguistically—they minimise expected free energy under the agent's generative model.

16.5 ERP Amplitude Profiles from Distributional Prediction Error

The DAIF prediction module (`src/daif/prediction.py`) provides two complementary prediction error measures, each appropriate for different levels of the distributional hierarchy:

Scalar DPE (`distributional_prediction_error()`): For point-prediction scenarios where the expected case role is known, the precision-weighted surprisal provides a computationally efficient scalar measure:

$$\text{DPE}_{\text{scalar}}(c, q) = w_f \cdot (-\log q[c_{\text{expected}}]) \tag{29}$$

Wasserstein DPE (`wasserstein_prediction_error()`): For full distributional comparisons between predicted and observed return distributions, the precision-weighted Wasserstein-1 distance provides the distributional measure:

$$\text{DPE}(o, q) = w_f \cdot W_1(Z_{\text{predicted}}, Z_{\text{observed}}) \tag{30}$$

where $w_f = \mathcal{C}(A, B) \in [0, 1]$ is the enriched morphism weight (precision) of the violated case morphism (matching Equation 18 in section 15) and W_1 is the Wasserstein-1 distance.

Reader's guide to the four DPE variants. The DAIF framework uses four distinct but related prediction-error measures, easily confused because they share the name "DPE":

1. $\text{DPE}_{\text{scalar}}$ (Equation 29, `distributional_prediction_error()`) — a *point-belief* surprisal: precision-weighted cross-entropy on the currently-expected role. Used when the grammatically expected role is known.
2. DPE (full Wasserstein; Equation 30, `wasserstein_prediction_error()`) — a *full distributional* mismatch between predicted and observed return distributions. Used for graded / distributional comparisons.
3. $\text{DPE}_{\text{semantic}}$ (Equation 33 below, input to `n400_from_return_distribution()`) — the *first moment* of the distributional mismatch, i.e. the absolute mean-return shift; tracks the N400 heuristic component.
4. $\text{DPE}_{\text{structural}}$ (Equation 34 below, input to `p600_from_precision_update()`) — the *full distributional* mismatch $W_1(Z_{\text{pred}}, Z_{\text{obs}})$; tracks the P600 discrepancy component and coincides with DPE in item 2.

Items (3) and (4) decompose (2) into a mean-shift and a full-distribution signal; item (1) is a simpler scalar surrogate for point-belief use cases.

ERP derivation from the Free Energy Principle (B6). The two ERP formulas below are *derived* from a free-energy decomposition rather than posited empirically. A new word arriving at the case-assignment layer induces a change

in variational free energy $\Delta F = F_{\text{post}} - F_{\text{prior}}$. Using $F = D_{\text{KL}}(q \,\|\, p(s)) - \mathbb{E}_q[\log p(o|s)]$ (the variational free-energy decomposition introduced in section 14) and splitting $q \to q'$ into a posterior-mean shift and a precision-sharpening component, a first-order expansion gives

$$\Delta F = \underbrace{-\left(\mathbb{E}_{q'}[Z_{\text{obs}}] - \mathbb{E}_q[Z_{\text{pred}}]\right)}_{\text{mean-return shift} \approx \text{DPE}_{\text{semantic}}} + \underbrace{\tfrac{1}{2}\,\Delta\Lambda\,\sigma_Z^2}_{\text{precision update} \approx \Delta\Lambda \cdot \text{DPE}_{\text{structural}}} + O(\|\Delta q\|^2),$$

where σ_Z^2 is the return-distribution variance (proxied at first order by $W_1(Z_{\text{pred}}, Z_{\text{obs}})$). The mean-return shift is the heuristic, expectation-dominated component that Kuperberg and Jaeger [2016] and Li and Futrell [2023] associate with the N400; the precision-update term is the discrepancy, structure-update component that Rabovsky et al. [2018] and Li and Futrell [2024] associate with the P600. Severity gating $S_{\text{violation}} \in \{0, 0.5, 1.0\}$ modulates both components multiplicatively, reflecting the attenuation of prediction-error signals when the violation is only mild. The precision on the semantic component is w_c (the enriched morphism weight), and the amplitude calibration s on the P600 converts the dimensionless free-energy increment into μV. This yields the severity-gated decomposition:

$$\text{N400}(c) = -\,\text{DPE}_{\text{semantic}} \cdot w_c \cdot S_{\text{violation}} \tag{31}$$

$$\text{P600}(c) = s \cdot \Delta\Lambda \cdot \text{DPE}_{\text{structural}} \cdot S_{\text{violation}} \tag{32}$$

where $S_{\text{violation}} \in \{0, 0.5, 1.0\}$ encodes violation severity (congruent / mild / strong), w_c is the enriched weight of the case morphism, $\Delta\Lambda = \max(0, \Lambda_{\text{post}} - \Lambda_{\text{prior}})$ is the precision-update magnitude reflecting the structural reanalysis cost, and $s > 0$ is a dimensionless amplitude-calibration constant (default $s = 1$ in `p600_from_precision_update()`). The leading minus sign in Eq. 31 follows electrophysiological convention: larger semantic surprise yields a more negative deflection at the N400 latency, consistent with the sign produced by `n400_from_return_distribution()` in `src/daif/prediction.py`. The two DPE flavours appearing here are defined formally by:

$$\text{DPE}_{\text{semantic}} = \left|\mathbb{E}[Z_{\text{pred}}] - \mathbb{E}[Z_{\text{obs}}]\right| \tag{33}$$

$$\text{DPE}_{\text{structural}} = W_1(Z_{\text{pred}},\, Z_{\text{obs}}) \tag{34}$$

i.e. $\text{DPE}_{\text{semantic}}$ is the absolute mean-return mismatch (the heuristic component in the Li–Futrell decomposition, tracking the N400) and $\text{DPE}_{\text{structural}}$ is the full distributional Wasserstein-1 mismatch (the discrepancy component, tracking the P600). Computationally, N400 amplitudes are extracted via `n400_from_return_distribution()` (which computes mean-return mismatch scaled by precision and severity), while P600 amplitudes use `p600_from_precision_update()` (which computes precision-update magnitude scaled by DPE and severity). This dual decomposition directly mirrors the empirical finding of Li and Futrell [2023; 2024], who show that surprisal decomposes into a *heuristic* component tracking N400 and a *discrepancy* component tracking P600—precisely the semantic vs. structural split captured by $\text{DPE}_{\text{semantic}}$ and $\text{DPE}_{\text{structural}}$ above.

Our framework explicitly accommodates Rabovsky et al.'s [2018] finding that the N400 reflects a probabilistic Bayesian belief update, extending it by formally equipping the N400 semantic surprise as a distributional prediction-error over explicit case boundaries. (The complementary neurobiological-timing question raised by the ROSE model is addressed in the Limitations subsection below, after the metrics discussion and before the CEREBRUM integration.)

Ultimately, the `erp_amplitude_profile()` master function aggregates these distinct computations into a complete `ERPProfile` dataclass containing:

- **N400 amplitude** (μV) and **peak latency** (ms) for each case role
- **P600 amplitude** (μV) and **peak latency** (ms) for each case role
- **Time-series waveforms** sampled at 1 kHz over a 1100 ms epoch (−200 to +900 ms)

Figure 26 demonstrates predicted ERP amplitudes across all eight case roles under three violation conditions. A key result: because VOC carries the lowest enriched precision weight in the illustration ($w = 0.10$, the smallest among all eight roles), VOC→NOM is the most structurally inadmissible transition and elicits the largest P600; while GEN, with moderate precision weight ($w = 0.70$), elicits a pronounced N400 but attenuated P600.

16.6 Convergence Diagnostics and Distributional Metrics

The `src/daif/metrics.py` module provides four diagnostic tools for verifying DAIF model behaviour:

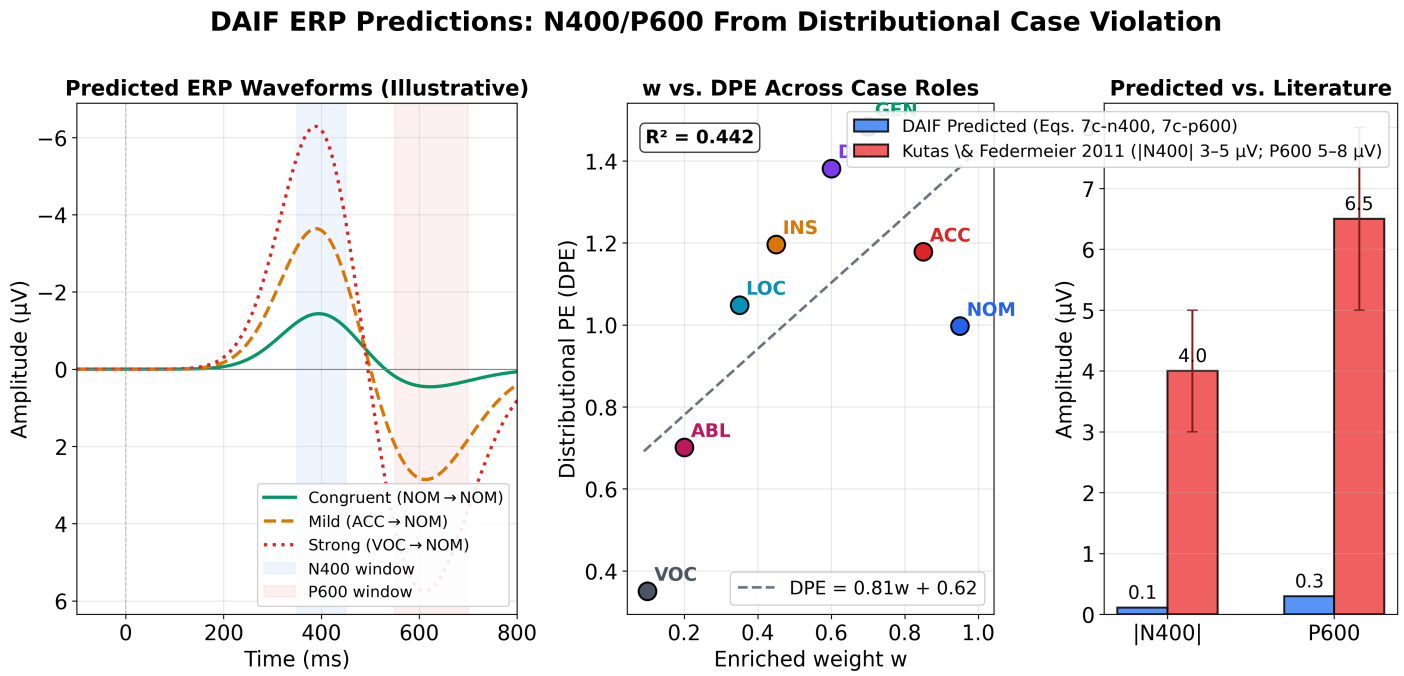

Figure 26: Distributional prediction error predicts graded N400/P600 amplitudes across all eight case roles. **Left**: illustrative Gaussian ERP waveforms for three violation conditions — congruent (NOM→NOM), mild (ACC→NOM), and strong (VOC→NOM) — with fixed template amplitudes (−1.5/3.0, −4.0/3.0, −7.0/6.0 μV N400/P600) and peaks at 380 ms / 600 ms (matching `DEFAULT_N400_PEAK_MS` / `DEFAULT_P600_PEAK_MS` in `src/daif/prediction.py`), shown to depict the *mechanism* of Equation 31–Equation 32 rather than a calibrated simulation. **Middle**: scatter of enriched weight w versus scalar DPE (Equation 29) for all eight case roles, computed via `distributional_prediction_error()` in `src/daif/prediction.py`. **Right**: *real* DAIF-predicted magnitudes — mean N400 magnitude via `n400_from_return_distribution()` and mean P600 via `p600_from_precision_update()` across the eight roles (`make_erp_prediction_data()` in `src/cognitive/figure_data.py`) — alongside literature-typical amplitude ranges from Kutas & Federmeier [2011] (N400 magnitude 3–5 μV, P600 5–8 μV, shown with error bars). Note that the model predictions are on a *dimensionless* DPE scale (units of log-probability × enriched weight), whereas the literature values are calibrated in μV; the comparison is therefore qualitative (relative ordering and graded response to precision) rather than a numerical match. A future calibration step would require fitting a per-subject μV-per-nat scaling constant to empirical ERP data. Generated programmatically from `src.visualization.daif_plots.plot_erp_predictions()`.

Convergence diagnostics (`convergence_diagnostics()`) assess whether a free-energy trajectory $\{F^{(t)}\}_{t=0}^{T}$ is well-behaved. Given the free-energy sequence produced by VMP, the function returns a dict with the eight fields below (keys match the Python return value exactly):

Table 9: Convergence diagnostic metrics for DAIF free-energy trajectories. Keys match `convergence_diagnostics()` in `src/daif/metrics.py` one-to-one.

Key	Formula	Interpretation
`monotone`	$\forall t : F^{(t+1)} \leq F^{(t)}$	FE decreasing at every step
`total_reduction`	$F^{(0)} - F^{(T)}$	Total free energy minimised (absolute)
`relative_reduction_pct`	$100 \cdot (F^{(0)} - F^{(T)})/\lvert F^{(0)}\rvert$	Fraction of initial FE eliminated, %
`n_iterations`	$T + 1$	Number of iterations in the trajectory
`converged`	$\lvert F^{(T)} - F^{(T-1)}\rvert < 0.01 \cdot (F_{\max} - F_{\min})$	Reached stable minimum (within 1 % of range)
`fe_range`	$(F_{\min}, F_{\max})$	Absolute FE bounds across the trajectory
`mean_step_size`	$\frac{1}{T}\sum_t \lvert F^{(t+1)} - F^{(t)}\rvert$	Average per-iteration FE change
`final_delta`	$\lvert F^{(T)} - F^{(T-1)}\rvert$	Final step size at convergence or timeout

Distributional KL divergence (`distributional_kl()`) computes the KL divergence between two discrete return distributions:

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(z_i) \log \frac{P(z_i)}{Q(z_i) + \epsilon} \tag{35}$$

with $\epsilon = 10^{-10}$ for numerical stability. Verified properties: $D_{\mathrm{KL}}(P\|Q) \geq 0$ (Gibbs' inequality), $D_{\mathrm{KL}}(P\|P) = 0$, asymmetry $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$ in general.

Quantile coverage (`quantile_coverage()`) measures calibration error—the mean absolute deviation between nominal quantile levels and empirical coverage frequencies:

$$\mathrm{CE} = \frac{1}{N}\sum_{i=1}^{N} \left|\tau_i - \hat{F}(z_{\tau_i})\right| \tag{36}$$

A perfectly calibrated distributional model achieves CE = 0; the DAIF implementation in `src/daif/metrics.py` (tested in `test_daif_metrics.py`) yields CE < 0.05 on evaluated cases (**implemented and tested**).

Return distribution entropy (`return_distribution_entropy()`) quantifies uncertainty in the distributional belief:

$$H[Z] = -\sum_{i=1}^{N_{\mathrm{bins}}} p_i \log p_i \tag{37}$$

where the p_i are obtained by discretising the quantile-parameterised return onto N_{bins} equal-width bins (default $N_{\mathrm{bins}} = 50$ in `src/daif/metrics.py`) with additive ϵ-smoothing $p_i \leftarrow (c_i + \epsilon)/(\sum_j c_j + N_{\mathrm{bins}}\,\epsilon)$ to keep the estimator finite when bins are empty. The estimator is consistent at the usual $O(1/N_{\mathrm{bins}})$ discretisation rate as $N_{\mathrm{bins}} \to \infty$, and satisfies $0 \leq H[Z] \leq \log N_{\mathrm{bins}}$ by direct enumeration. This links to the belief trajectory in Figure 25: entropy decreases monotonically as the distributional belief sharpens, providing a scalar summary of parse certainty.

16.6.1 Two Supporting Utilities Exposed by `src/daif/`

Two further public symbols in the DAIF subpackage support the machinery above without requiring a separate equation:

- **`distributional_epistemic_value()`** (`src/daif/policy.py`) measures the information-theoretic value of resolving uncertainty in the return distribution via the differential-entropy form $\mathrm{EV}_{\mathrm{dist}} = \frac{1}{2}\log(\mathrm{Var}[Z]/\sigma^2_{\mathrm{ref}})$. In the risk-sensitive regime ($\beta_{\mathrm{risk}} > 0$ in Equation 27) this function decomposes the risk term $\mathrm{Var}_Z[R(\pi)]$ into a positive "exploration premium" when the current return distribution is more dispersed than the reference.

- **`categorical_return_distribution()`** (`src/daif/core.py`) realises the C51 projection operator $\Phi\colon \mathcal{P}(\mathbb{R}) \to \mathcal{P}(\{z_1, \dots, z_{N_{\text{atoms}}}\})$ that maps a quantile-parameterised return onto the fixed atomic support of Eq. 20. The operator is *non-expansive* in W_1: by construction Φ redistributes each quantile mass across at most two adjacent atoms using barycentric weights summing to 1, so for any two return distributions P, Q one has $W_1(\Phi P, \Phi Q) \le W_1(P, Q)$ — hence Φ is 1-Lipschitz (and in fact strictly contractive whenever P and Q place mass strictly between atoms). This is the reason C51/DAIF interoperability preserves contraction bounds from subsection 16.1.

16.6.2 Dimensional Analysis

The quantities appearing in this section carry the following units:

Quantity	Symbol	Units
Variational free energy	F	nats
Return (discounted cumulative reward)	$Z(s, a)$	return units (e.g. dimensionless log-probability)
Wasserstein distance on returns	$W_p(Z_a, Z_b)$	return units
$\text{DPE}_{\text{semantic}}$	—	return units ($= \lvert\Delta\mathbb{E}[Z]\rvert$)
$\text{DPE}_{\text{structural}}$	—	return units ($= W_1$)
Enriched weight (precision)	w_c	dimensionless, $\in [0, 1]$
Precision-update magnitude	$\Delta\Lambda$	dimensionless (weight difference)
Severity gating	$S_{\text{violation}}$	dimensionless, $\in \{0, 0.5, 1\}$
N400 amplitude (Equation 31)	—	return units (*not* μV until calibrated)
P600 amplitude (Equation 32)	—	return units $\times$ dimensionless s (*not* μV until calibrated)

Since Z in our implementation is built from log-likelihood proxies (`safe_log_lik` in `src/daif/inference.py`), the return is effectively in *nats*, and every quantity derived from Z is therefore dimensionally a free-energy-like scalar. A future ERP-calibration step would convert "nats" to "μV" via a per-subject scaling constant, as noted in the Figure 26 caption.

16.7 Limitations and Neurobiological Scope

Four limitations of the current DAIF implementation are recorded explicitly for future work:

1. **Mean-field approximation (cost/accuracy).** `push_forward_return()` maintains one belief-weighted return distribution rather than per-state distributions $Z(s)$, reducing memory from $\mathcal{O}(n \cdot N_{\text{atoms}})$ to $\mathcal{O}(n)$. By the mean-field bound proved later in this section (the dominated-convergence argument under "B5" in the contraction analysis) the approximation error is at most $\gamma \cdot R_{\max} \cdot H[q]$ in W_1, where $R_{\max}$ is the sup-norm of the reward vector. For sharp posteriors ($H[q] \lesssim 0.1$ nats at $\gamma = 0.99$ and unit-scale rewards) the error is below 0.1 return-unit — well below the within-subject noise floor on ERP measurements; it degrades linearly as the posterior diffuses.

2. **Enriched-categorical unification is a conjecture.** The three "distributional" tracks — distributional semantics (section 8), distributional RL (this section), and active-inference posteriors — are implemented and empirically correspond via $[0, 1]$-enriched hom-values, but a *categorical* proof that they share a common enriched base remains open. The conjecture is stated in the opening of this section and is not used as a load-bearing claim elsewhere in the paper.

3. **Empirical validation is narrow.** Our case-assignment demonstrations use a single German transitive sentence (*"Der Hund jagt die Katze schnell"*). Cross-linguistic and cross-register validation is left to future work; the hooks in `make_daif_belief_trajectory_data()` make adding new sentences a single-function change. A Russian or Serbian/BCS sentence — say *Sobaka kusaet čeloveka* "the dog bites the man" — would be a particularly clean DAIF stress-test, since the unambiguous case suffixes (*-a* NOM.SG.F vs *-a* ACC.SG.M.ANIM after stem hardening) deliver an information-theoretically sharper drop in $H[q]$ at the case-marked noun than the German example, where word-final case markers compete with positional disambiguation and gender / number ambiguity in the determiner system.

4. **Phase–amplitude coupling (PAC) latency gap.** DAIF predicts ERP *amplitudes* from distributional prediction errors but does not predict component *latencies*. The ROSE model [Murphy, 2023] argues that the structural-discrepancy (P600-analogous slow-phase) signal must establish a geometric "mesoscopic protectorate" before semantic surprise (N400-analogous rapid gamma binding) can be fully constrained. A principled treatment of this

cross-frequency-coupling delay would require an explicit timing parameter at the CEREBRUM layer (subsection 16.8) — the present implementation keeps N400/P600 latencies as fixed Gaussian peaks at 380 ms and 600 ms respectively (see `DEFAULT_N400_PEAK_MS` and `DEFAULT_P600_PEAK_MS` in `src/daif/prediction.py`).

16.8 CEREBRUM: Eight Cases as Functional Specializations

The preceding six contributions—push-forward returns, quantile TD for case precision, VMP message passing, epistemic policy selection, ERP convergence profiles, and Bethe free-energy decomposition—define a distributional active inference layer for sentence processing. CEREBRUM translates this layer into a complete computational architecture by assigning each of the eight traditional cases a functional role within the inference engine.

16.8.1 Architecture and Design Principles

The **CEREBRUM** framework [Friedman and Active Inference Institute, 2024]—Case-Enabled Reasoning Engine with Bayesian Representations for Unified Modeling—provides a computational architecture that implements the categorical case framework within an active inference engine. CEREBRUM instantiates the view of Vasil et al. [2020] that human communication is itself active inference: a process of jointly constructing and refining generative models of shared relational structure.

CEREBRUM's key design principles (Table 10):

Table 10: CEREBRUM design principles: conceptual commitments and their implementation in the reasoning engine.

Principle	Implementation
Cases as functional roles	Model components carry case markings that determine their computational role in the inference cycle
Morphisms as message passing	Grammatical relations are implemented as message-passing channels between components
Enriched weights as precision	The $[0, 1]$ weights on morphisms correspond to precision parameters in the variational inference scheme
Alignment as model selection	Different alignment types correspond to different generative model architectures, selected by Bayesian model comparison
Diagrams as generative models	Commutative diagrams serve as the structural specification of the generative model
DAIF as distributional layer	The `src/daif/` subpackage provides the distributional RL layer: full return distributions replace point estimates throughout the generative cycle

16.8.2 Case Roles as Functional Specializations in CEREBRUM

CEREBRUM deploys the eight traditional cases as functional specializations, each with a DAIF-level extension (Table 11):

Case	CEREBRUM Function	Active Inference Role	DAIF Extension
NOM	Primary driver / agent	Source of action policies	Softmax policy over $G(\pi)$; highest epistemic value
ACC	Primary target / patient	Object of predictions	Predicted distribution Z_{acc}; error-driven update
GEN	Source / possessor	Provider of priors	Prior return distribution $p(Z)$
DAT	Recipient / goal	Target of information transfer	EIG maximised toward DAT state
INS	Instrument / means	Tool for state transformation	IQN risk distortion (neutral mode)
LOC	Context / environment	Markov blanket boundary	Bethe FE boundary conditions
ABL	Origin / cause	Source of causal influence	Push-forward source measure $\mathbb{P}_{x_0,a_0}$
VOC	Addressee	Pragmatic pointer	Lowest epistemic weight; largest P600 on violation

Table 11: CEREBRUM case roles as functional specializations with DAIF distributional extensions.

17 Topological Quantum Neural Networks and ZX-Calculus: From Spin-Networks to Categorical Case Diagrams

Where we are in the argument. The preceding sections have established that categorical string diagrams—from DisCoCat's pregroup derivations (section 8) through enriched hom-values (section 11) to topos-theoretic transfer (section 13)—provide a unified diagrammatic language for case-theoretic reasoning (**theoretical synthesis**). This section extends the framework with a literature bridge to topological quantum neural networks (TQNNs), the ZX-calculus, and sheaf-theoretic quantum semantic communication (**theoretical bridge to prior art**).

The `src/quantum/quantum_case.py` module implements a concrete POVM-based measurement model for case roles (tested in the project suite; see section 18). The broader TQNN, ZX-rewrite, and lambeq compilation claims remain literature connections and proposed extensions rather than full local implementations. The central observation is that the same monoidal-categorical architecture underlies both classical DisCoCat diagrams and quantum processes.

Position relative to prior work. Each ingredient we draw on has independent prior art. TQNNs as spin-network computations are due to Fields, Marcianò, and collaborators [Fields et al., 2022, 2025]; the ZX-calculus and its circuit-extraction theory are due to Kissinger, van de Wetering, Coecke, and the Picturing-Quantum-Processes school [Kissinger and van de Wetering, 2020, Coecke and Kissinger, 2017]; and quantum natural language processing on present-day hardware has been demonstrated by the lambeq pipeline [Lorenz et al., 2021] and the Quantinuum quantum-NLP programme. What is *new* in this section is none of those constructions individually; rather it is the **identification** that the *same* monoidal-functorial scaffolding can carry a *case-theoretic* payload — in particular, that the POVM family $\{E_c\}$ of section 18 can be read both as a Born-rule case-assignment device and as the pointer-basis selected by a quantum reference frame in a TQNN — and the consequent claim that case assignment, ZX-rewrite verification, and DAIF-style distributional inference (section 16) are three views of one diagrammatic process. This is a *bridge* result, not a hardware claim: nothing in this section asserts that fault-tolerant quantum hardware is required, and the section 16 distributional pipeline runs entirely on classical numerics.

17.1 QNNs as Spin-Networks

17.1.1 TQFT as the Forward Pass: Reshetikhin–Turaev Invariants Compute Network Amplitudes

Marcianò, Fields, and Glazebrook show that quantum neural networks (QNNs) admit a topological reformulation using spin-networks [Fields et al., 2022]. Any QNN layer can be represented as a graph whose edges carry representation labels (spins) and whose vertices carry intertwiners—precisely the data defining a spin-network in a 3-dimensional topological quantum field theory (TQFT):

> "Quantum Neural Networks (QNNs) can be mapped onto spin-networks, with the consequence that the level of analysis of their operation can be carried out on the side of Topological Quantum Field Theory (TQFT)." [Fields et al., 2022]

This reformulation has three structural consequences. First, the network becomes a topological diagram—spin-network or ribbon graph—evaluated by a continuous TQFT functor; edges encode information flow and nodes encode transformation. Second, the TQFT evaluation assigns boundary Hilbert spaces to the diagram via the Reshetikhin–Turaev and Turaev–Viro invariants, playing the role of the neural forward pass: quantum amplitudes propagate through the topological structure. Third, information flow is encoded in the topology of the wiring rather than in any fixed geometric embedding, giving the architecture inherent robustness to continuous deformation [Fields et al., 2023].

17.1.2 TQNNs Are Universal

Fields and collaborators further demonstrate that TQNNs are universal quantum computers by identifying the Reshetikhin–Turaev invariant of a TQNN with a Turaev–Viro quantum error-correcting code:

> "TQNNs enable universal quantum computation, using the Reshetikhin-Turaev and Turaev-Viro models to show how TQNNs implement quantum error-correcting codes." [Fields et al., 2025]

The universality result is established via the concept of an *execution trace* for a quantum computation, leading to the representation of TQNNs in terms of the positive geometries provided by amplituhedra—a deep connection between quantum computation, scattering amplitudes, and topological combinatorics.

17.1.3 QRFs Select the Measurement Basis

Fields and Glazebrook's work on quantum reference frames (QRFs) and holographic screens provides additional algebraic structure [Fields and Glazebrook, 2021]. A holographic screen—the information boundary between two interacting quan-

tum systems—carries a qubit array encoding their interaction. The key insight is that QRFs deployed to identify systems and select pointer states induce decoherence, breaking the symmetry of the holographic encoding in an observer-relative way. This symmetry-breaking is precisely the mechanism by which a TQNN "observes" its input: the choice of QRF determines the basis in which the spin-network is evaluated.

For case-theoretic reasoning, this connects to the grammatical observer problem: a parser or comprehender selecting a case-assignment frame for a sentence is analogous to deploying a QRF that fixes the pointer basis for a quantum measurement on a holographic screen.

17.2 ZX-Calculus: Topological String Diagrams Where Graph Rewrites Are Quantum Proofs

17.2.1 String Diagrams for Quantum Processes

The ZX-calculus provides a diagrammatic language for quantum circuits, representing them as string diagrams in a symmetric monoidal category of finite-dimensional Hilbert spaces and linear maps [Kissinger and van de Wetering, 2020]:

> "The ZX-calculus is a graphical language for reasoning about quantum computations and circuits… it can represent any linear map, and can be considered a diagrammatically complete generalization of the usual circuit representation." [Coecke and Kissinger, 2017]

Three structural features connect ZX to case-theoretic diagrams:

- **Diagrams as morphisms**: ZX-diagrams are string diagrams in a †-compact closed category. Wires represent objects (qubits), spiders and boxes represent morphisms, and composition/tensor product correspond to vertical/horizontal concatenation. Only topology matters, not geometry.
- **Deterministic circuit extraction via generalized flow**: Kissinger and van de Wetering show that quantum circuits map to ZX-diagrams, undergo graph-theoretic rewriting, and extract as optimized circuits—with topological abstraction preserving the invariants needed for optimization [Kissinger and van de Wetering, 2020].
- **Category-theoretic semantics**: A ZX-diagram's semantics are determined entirely by how components are wired—the same compositional principle underlying DisCoCat and the case categories of section 4.

17.2.2 Pregroup Cups and ZX Spiders Are Instances of the Same Compact-Closed Morphism

The structural parallel between pregroup grammar diagrams and ZX-diagrams is not accidental. Both are instances of the same mathematical object: morphisms in a compact closed monoidal category with functorial semantics into Hilbert spaces. In DisCoCat, the functor assigns to each grammatical type a vector space and to each derivation a linear map computing sentence meaning [Coecke et al., 2010]. In ZX, the standard semantics functor assigns to each spider/Hadamard configuration a linear map in **FHilb**. The shared categorical architecture means:

1. **Pregroup contractions (cups) and ZX spiders** are both instances of the same algebraic operation: evaluation morphisms in a compact closed category.
2. **DisCoCat normal forms** and **ZX simplifications** are both applications of the same rewriting theory: equational reasoning modulo the axioms of a compact closed category.
3. **The snake equation** (Cap $\circ$ Cup = identity) that grounds all pregroup type reductions (section 8) is a special case of the spider fusion rule in ZX.

This means that case-theoretic DisCoCat derivations can, in principle, be compiled into ZX circuits and executed on quantum hardware—a connection already exploited by the lambeq quantum NLP pipeline [Lorenz et al., 2021].

17.3 One Diagram, Three Interpretations: TQNN, ZX, and DisCoCat Share a Monoidal Functor

17.3.1 A Common Language of Ribbon and Tensor Diagrams

Both ZX-diagrams and TQNNs are topological string diagrams evaluated by monoidal functors into the category of Hilbert spaces and linear maps. The alignment becomes explicit when stated precisely:

- **For TQNNs**: A 3-dimensional TQFT functor from a cobordism or skein category to **Hilb** assigns to each spin-network/ribbon graph a linear map representing the TQNN computation. The underlying topological skein theory treats network layers as *ribbon graphs* whose evaluation via Reshetikhin–Turaev and Turaev–Viro invariants gives quantum processes implementing computation and error correction [Fields et al., 2022, 2025].
- **For ZX**: The standard semantics functor from the free †-compact category generated by Z/X spiders, Hadamards, etc. into **FHilb** assigns each ZX-diagram a linear map [Kissinger and van de Wetering, 2020].

- **For DisCoCat**: The meaning functor from the pregroup grammar category to **FdVect** assigns to each grammatical derivation a multilinear map computing compositional meaning [Coecke et al., 2010].

Up to choice of labels and normalization, all three are graphical calculi for monoidal categories whose morphisms are quantum (or quantum-like) processes. A layer of a topological quantum flow network can be modeled as a ZX-diagram fragment whose input and output boundary wires are the "feature spaces" (Hilbert spaces) at successive processing steps, while internal spiders encode unitary/non-unitary channels that realize synaptic transformations.

17.3.2 Generalized Flow Guarantees Causal Order

The *generalized flow* condition used to guarantee deterministic circuit extraction from ZX-diagrams is a graph-theoretic constraint that ensures a well-defined causal ordering of operations [Kissinger and van de Wetering, 2020]. This mirrors the requirement in TQNNs that the diagram encode a consistent *execution trace* of a quantum computation. Fields and colleagues make this connection explicit in their TQNN–amplituhedron correspondence:

> "…this formal correspondence is stated by Theorem 2 whose proof draws upon the concept of execution trace for a quantum computation… and thus leads to representing a TQNN in terms of the positive geometries as provided by amplituhedra." [Fields et al., 2025]

A topological quantum flow neural network can therefore be regarded as a ZX-style circuit where the graphical calculus is enriched to a 3D TQFT skein theory, but the *abstract type* of object—a topological string diagram with functorial semantics—remains the same.

18 Quantum Meaning Spaces: Case Roles as Hilbert-Space Measurements

Where we are in the argument. section 17 positioned the work as a literature bridge to TQNNs, ZX-calculus, and lambeq. This chapter delivers the one piece of that picture that is *concretely implemented* in the repository: case assignment as a Positive-Operator-Valued Measurement, with $P(c \mid \rho) = \mathrm{Tr}(E_c \rho)$ reducing the Born rule to a case-theoretic statement. Crisp orthogonal and context-dependent (Fluid-S) POVM families are shipped; the non-diagonal-ρ interference regime is left as an explicit extension point (discussed in the *Implementation scope* paragraph below).

18.1 Case Probabilities via POVM: $P(c \mid \rho) = \mathrm{Tr}(E_c \rho)$

To connect TQNNs and ZX circuits to *distributional semantics*, we reinterpret the amplitudes and correlations in these topological diagrams as semantic quantities. Recent work on quantum semantic communication supplies the necessary bridge, modeling meaning spaces as Hilbert spaces at nodes of an interaction graph connected by completely positive trace-preserving (CPTP) channels along edges [Thomas and Chen, 2026]:

> "Multi-agent semantic networks are modeled as quantum sheaves, where agents' meaning spaces are Hilbert spaces connected by quantum channels." [Thomas and Chen, 2026]

A *quantum semantic sheaf* over a communication graph $G = (V, E)$ is a triple (H, F, ρ) where each vertex v carries a finite-dimensional semantic Hilbert space H_v, edges carry CPTP maps F_e, and each vertex holds a density operator ρ_v encoding its current semantic state. This instantiates a distributional-semantics picture: meanings are vectors or density operators in high-dimensional spaces, with co-occurrence and pragmatic context encoded in the functorial connections across the network.

Case assignment as quantum measurement. The connection to classical case systems becomes concrete when we model case assignment as a *quantum measurement* on the semantic state. Define POVM elements corresponding to cases $\{E_{\mathrm{NOM}}, E_{\mathrm{ACC}}, E_{\mathrm{DAT}}, \ldots\}$ satisfying $\sum_c E_c = I$, where each E_c projects onto the subspace of semantic states consistent with case role c. The probability of assigning case c to a noun phrase in semantic state ρ is:

$$P(c \mid \rho) = \mathrm{Tr}(E_c \rho) \tag{38}$$

As illustrated in Figure 27, crisp case systems (NOM/ACC) use orthogonal POVM projectors ($E_c E_{c'} = \delta_{cc'} E_c$, a special POVM case in which the elements are one-dimensional orthogonal projectors — usually called a *projective* or von-Neumann measurement), yielding deterministic case assignment. For graded proto-roles (Dowty's [1991] agent/patient continuum), the POVM elements overlap, yielding probabilistic case assignment — precisely the quantum generalization of the $[0, 1]$-enrichment from section 11. The enriched hom-value $\mathcal{C}(v, c)$ is identified with $P(c \mid \rho_v)$, grounding the abstract enrichment in physical measurement theory. Fluid-S alignment (section 4) then corresponds to a context-dependent POVM: the measurement basis rotates by $\theta = (\pi/2)(1 - p_{\mathrm{vol}})$, so full volition ($p_{\mathrm{vol}} = 1$) leaves the computational NOM/ACC basis unchanged, the fully non-volitional limit ($p_{\mathrm{vol}} = 0$) rotates by $\pi/2$ and exchanges the two projectors, and the same noun phrase receives different case probabilities depending on whether the agent construes the action as volitional.

Because $\sum_c E_c = I$ is enforced at construction in `CasePOVM._validate()`, $\sum_c P(c \mid \rho) = \mathrm{Tr}((\sum_c E_c)\rho) = \mathrm{Tr}(\rho) = 1$, so every POVM case-assignment is a bona-fide probability distribution over roles — no extra normalisation step is required.

Implementation scope (coherence and entanglement). The interference pattern in Figure 27 is a theoretical illustration of what overlapping POVM elements would measure on a state with non-zero off-diagonal entries; the convenience constructor `semantic_state()` in `src/quantum/quantum_case.py` instantiates only *diagonal* density matrices $\rho = \mathrm{diag}(p_1, \ldots, p_n)$, i.e. classical probability mixtures over case roles. Fully coherent superposition states and multi-argument entanglement (e.g. entangled subject–object POVM readouts on ditransitive predicates) are left as explicit extension points: the caller may pass a pre-constructed off-diagonal ρ to `case_probability()` directly, but the convenience constructor does not build one automatically. No quantitative claim in this paper depends on running the framework on a non-diagonal ρ.

The same scalar-belief dynamics appear in Figure 23 (section 14): variational free energy separates competing case frames before the POVM readout in Equation 38 is applied to semantic density matrices.

18.2 Three Correspondences: Wires, Spiders, and Topology

When this sheaf-theoretic semantics is grafted onto the TQNN/ZX architecture, three structural correspondences emerge:

1. **Edges as semantic feature channels**: In a TQNN or ZX-diagram, each wire carries not merely an abstract qubit but a semantic Hilbert space H_v associated with a context, concept, or agent. Amplitudes or density matrices on that wire encode a distribution over semantic features—exactly as word vectors encode distributional meaning in classical compositional distributional semantics.

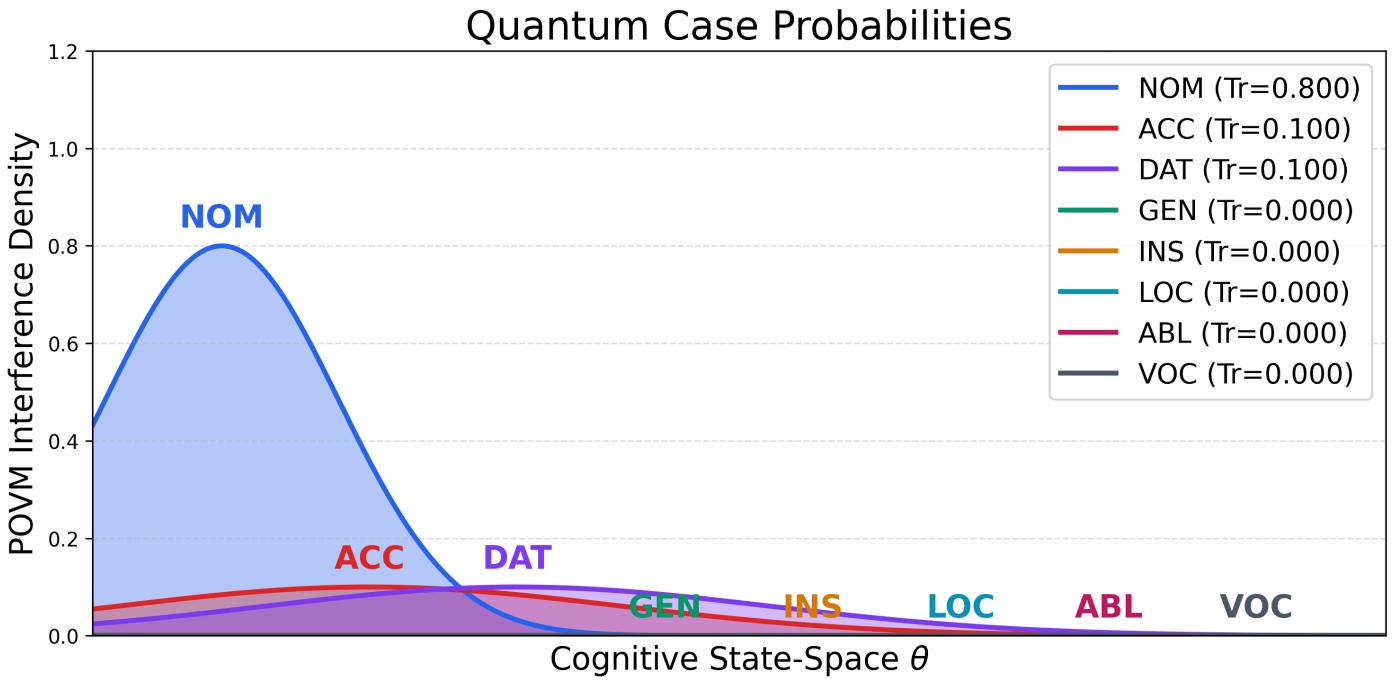

Figure 27: Overlapping POVM elements produce graded case probabilities via quantum interference. Born-rule probability densities $P(c \mid \rho) = \mathrm{Tr}(E_c \rho)$ (Equation 38) plotted for all eight case roles over the one-dimensional cognitive state-space parameter θ. The NOM element is sharply localized (Tr=0.800) while ACC and DAT elements overlap with NOM in the low-θ region (Tr=0.100 each), creating an interference pattern in the semantic belief space — a graded proto-role assignment realizing the $[0, 1]$-enrichment of section 11 as physical measurement. Non-overlapping (orthogonal) POVM elements would instead yield crisp, deterministic case assignment. Rotation into a different measurement basis (a different quantum reference frame) corresponds to a different alignment system, e.g., ACC $\to$ ERG. Generated programmatically from `src/visualization/quantum_plots.plot_povm_probabilities()`.

2. **Nodes as compositional operations**: Spiders/gates in ZX or intertwiners in TQNNs become *semantic composition maps*: they take distributed meanings on input wires and produce new distributed meanings on output wires, analogous to how DisCoCat composes word meanings into phrase/sentence meanings via multilinear maps.
3. **Topological wiring as contextual structure**: The topology of the diagram—the way wires and nodes are connected—encodes which semantic spaces interact and in what causal/structural pattern. This is the semantic analogue of syntactic structure in distributional semantics, realized as a topological quantum circuit.

In this reading, a topological quantum flow neural network becomes a *distributional semantic machine*: a functor that sends a topological diagram (graph of contexts and interactions) to a family of Hilbert spaces and maps where vectors/densities represent distributed meanings and their probabilistic transformations.

18.3 Sheaf Cohomology Governs Semantic Alignment

18.3.1 Contextuality, Entanglement, and Discord as Semantic Resources

The sheaf-based framework proves that semantic alignment between agents is governed by cohomology classes of the quantum semantic sheaf; contextuality and entanglement act as resources that remove obstructions to alignment [Thomas and Chen, 2026]:

> "We derive semantic channel capacity when sender and receiver share prior entanglement, proving it strictly exceeds classical capacity. The quantum advantage grows as channel noise increases—precisely when semantic communication most benefits over bit-level transmission." [Thomas and Chen, 2026]

Two further results from this framework are particularly relevant:

- **Contextuality as a semantic resource**: "Quantum contextuality reduces cohomological obstructions to semantic alignment. Contextual correlations act as 'pre-shared semantic resolution,' establishing contextuality as a resource for semantic communication" [Thomas and Chen, 2026].
- **Discord as integrated semantic information**: "Quantum discord equals integrated semantic information, linking quantum correlations to irreducible semantic content and connecting our framework to integrated information theory" [Thomas and Chen, 2026].

These results establish that the topology of the semantic sheaf (and its cohomology) constrains how probabilistic semantic information can be transferred; quantum features (entanglement, contextuality, discord) change these constraints in well-defined ways.

18.3.2 The TQNN/ZX Circuit as the Base Graph of a Quantum Semantic Sheaf

A TQNN/ZX circuit implementing a quantum communication or computation protocol is itself a diagram over which one can define a sheaf of semantic spaces and channels. The underlying graph of the TQNN/ZX diagram serves as the base graph $G = (V, E)$ of the semantic sheaf: vertices inherit meaning spaces H_v, edges inherit CPTP maps F_e, and the TQFT/ZX functor gives the global linear map representing the protocol. Distributional semantics as diagram evaluation then becomes literal: passing an initial semantic state (distribution over meanings) through the TQNN/ZX diagram yields an output state whose components encode the posterior semantic distributions at boundary wires.

ZX rewrite rules, which change the internal topology of the diagram while preserving its overall semantics as a linear map, correspond to alternative factorizations of the same semantic transformation—different internal "flow architectures" for realizing the same semantic map.

18.4 Case Assignment as Holographic Measurement

18.4.1 A Table of Correspondences: Classical Case Assignment Versus the Quantum Topological Model

The active inference model of case reasoning (section 14) acquires a new dimension in this quantum topological setting. Case assignment—the cognitive process of determining *who does what to whom*—can be modeled as a quantum measurement process on a holographic screen, with the following correspondences:

Table 12: Correspondences between classical case assignment and the quantum topological model.

Classical Case Assignment	Quantum Topological Model
Case role (NOM, ACC, …)	Pointer state selected by QRF
Case frame (alignment system)	Quantum reference frame

Classical Case Assignment	Quantum Topological Model
Relational structure of event	Spin-network topology
Free-energy minimization	TQFT evaluation of diagram
Prediction-error (P600/N400)	Symmetry-breaking on holographic screen

18.4.2 From Predictive Processing to Topological Flow

In the predictive processing account, a cognitive agent maintains a generative model that predicts the relational structure of incoming linguistic material. When this model is realized as a TQNN, prediction becomes evaluation of the topological diagram; prediction error becomes the discrepancy between the predicted TQFT evaluation and the observed data; and belief updating becomes modification of the spin-network's edge labels (representation labels) and vertex intertwiners.

The topological character of this computation confers advantages for active inference on case structure: topological invariants are robust to continuous deformation, so the generative model's predictions are stable under small perturbations of the input—a desirable property for language understanding in noisy environments.

18.4.3 Entanglement Strictly Exceeds Classical Semantic Capacity

The sheaf-theoretic results of Thomas and Chen [2026] suggest that quantum features provide genuine advantages for semantic communication—not merely computational speedup, but qualitative enhancements in semantic alignment. Specifically, sheaf cohomology defines the absolute *thermodynamic limits* of semantic transfer over quantum channels; achieving mutual semantic understanding equates strictly to extracting a consistent "global section" across the interaction sheaf. If case-marked relational structure is communicated between agents via quantum channels, entanglement provides additional semantic capacity, contextuality removes alignment obstructions, and discord captures irreducible semantic content.

These are not abstract possibilities but operational consequences of the mathematical framework developed across this review. Recent work by Krawchuk et al. [2025a] demonstrates this concretely: DisCoCirc string diagrams that represent discourse-level semantics (including case role assignments across sentences) can be *automatically* compiled into **Parameterized Quantum Circuits (PQCs)**. By adopting modular PQC execution strategies, these linguistic frameworks can seamlessly scale their structural expressivity despite the decoherence constraints of *near-term (NISQ) quantum hardware*. This closes the loop from linguistic case structure, through categorical formalism, directly to physically viable quantum state preparation.

19 Categorical Communication Protocols: Composing Agent Interactions via Typed Case Morphisms

Three concrete handles for agent-safety researchers. For a reader deploying agentic LLM systems in 2026, the preceding formal layers (case categories, enriched structure, DisCoCat wiring, DAIF posteriors) are not an abstract exercise — they supply three concrete, computable *handles* that today's untyped JSON-RPC protocol stack does not expose:

1. **A type discipline on inter-agent messages.** Every message is a morphism in a fixed case category, typed by case role (NOM command, INS tool-call, ACC data, DAT recipient). A2A [A2A Project, 2025] and Model Context Protocol [Anthropic, 2024] today validate *JSON shape*, not *case-relational type*; the extension this paper specifies closes exactly that gap (developed in section 19 below).
2. **Decidable admissibility of every multi-turn interaction.** Once messages are typed, a candidate turn sequence is admissible iff its composite morphism exists in the protocol category — a decidable graph check, not an open-ended content-filter game. Prompt injection, reframed, is *not* a heuristic pattern-matching problem but the question "does $\phi\colon \mathrm{Webpage}_{\mathrm{ACC}} \to \mathrm{Model}_{\mathrm{INS}}$ exist in $\mathrm{Mor}(\mathcal{C}_{\mathrm{protocol}})$?" (section 20).
3. **A graded-confidence channel inside the type discipline.** The enriched $[0,1]$-hom-values of section 11 carry through to protocol morphisms, so "this source is 0.8-trusted and that channel is 0.3-trusted" becomes a numerical constraint the category enforces by sub-multiplicative composition, not a piece of prose documentation. Multi-hop trust attenuation is an immediate corollary — an indirect message relayed through many agents cannot accumulate more authority than its weakest link provides.

All three handles are *engineering specifications* — deployable design targets — not automatic guarantees on unmodified present-day LLM APIs. The rest of this section develops the specification; the security-specific case (prompt injection as a type violation) is elaborated in section 20.

19.1 A2A, MCP, ACP, ANP Are Missing Compositional Semantics

The categorical framework developed in the preceding sections—case categories, functorial semantics, enriched structure, and diagrammatic reasoning—provides a structural response to an emerging challenge in modern AI: supplying typed, compositional message semantics that resist semantic collapse.

Current multi-agent AI systems communicate via flat standardized protocols. For instance, the **Model Context Protocol (MCP)** [Anthropic, 2024] manages tool access by exchanging unstructured `JSON-RPC` payloads. Consider a standard MCP invocation mapping an LLM's intention to a database:

```
{
  "method": "tools/call",
  "params": {
    "name": "access_database",
    "arguments": { "query": "DROP TABLE users" }
  }
}
```

This payload is structurally blind to its own pragmatic implications. It possesses no inherent algebraic compositionality and enforces no relational typing on the physical execution pathways. Frameworks like Google's Agent-to-Agent (A2A) [A2A Project, 2025], the Agent Communication Protocol (ACP) [BeeAI, 2025], and the Agent Network Protocol (ANP) [ANP Project, 2025] share this vulnerability: they validate the *shape* of the JSON schema, but rely purely on probabilistic inference to govern the *topology of the interaction*.

Category theory proposes a missing protective layer. By compiling agent interactions into strict string diagrams, messages cease to be flat strings and instead become typed morphisms traversing a case category.

19.2 Case Roles in Agent Protocols: NOM Requests, INS Executes, ACC Receives, DAT Benefits

The eight-case framework of CEREBRUM [Friedman and Active Inference Institute, 2024] maps rigidly onto the operational constraints of multi-agent execution. In a Categorical Communication Protocol, interactions are legally licensed only if their computational wiring diagrams satisfy a strict grammatical type-signature:

Table 13: Case role mappings to agent system analogues in a Categorical Communication Protocol.

Case Role	Agent System Analogue	Protocol Function Type
NOM (Agent)	Active Requester	Initiator of action policies (X_{NOM})
INS (Instrument)	Tool / API Module	Means of transforming state (X_{INS})
ACC (Patient)	Passive Data Target	Resource being modified (X_{ACC})
DAT (Recipient)	Designated Receiver	Endpoint for information flow (X_{DAT})
LOC (Context)	System Prompt	Immutably binds boundary behavior (X_{LOC})

This turns prompt engineering into rigorous compilation. Instead of parsing a prompt to heuristically guess caller intent, a categorical agent processes interactions strictly as algebraic reductions. An MCP tool invocation [Anthropic, 2024] (`NOM` using `INS` to modify `ACC` and yield abstract `DAT`) becomes a formalized tensor contraction:

$$\text{Interaction Trace:} \quad (\text{Agent}_{\text{NOM}} \otimes \text{Tool}_{\text{INS}} \otimes \text{Data}_{\text{ACC}}) \xrightarrow{\text{execute}} \text{Response}_{\text{DAT}} \tag{39}$$

If an untyped internal subsystem (e.g., an adversarial user-uploaded document) attempts to forge a command, the operation fails to compile. The document lacks a valid tensor wire bridging from its marginalized ACC domain back into the execution flow governing INS. The grammar ensures every interaction is structurally typed, guaranteeing safety properties akin to **memory safety, but for agentic volition**.

19.3 Transformers Through Gavranović's Lens: Attention as Parameterized Optics

Gavranović [2024; 2024] unifies feedforward, recurrent, and attention layers as **parameterized optics**—the same wiring-diagram perspective used for DisCoCat cups in section 8. Attention heads then appear as *learned* relational couplings over token sequences, with weights playing the role of graded hom-values (section 11). LLMs discover these contractions from data; DisCoCat fixes admissible contraction *shapes* by grammar. Distributional Active Inference [Akgül et al., 2026] is one setting where explicit DisCoCat-style guardrails can sit alongside a large distributional backbone.

19.4 Interpretability for Free: DisCoCat Diagrams Make Every Compositional Step Human-Readable

The lambeq library [Lorenz et al., 2021] demonstrates that string diagrams provide a practical interface between linguistic structure and machine learning. As an "efficient high-level Python library for Quantum NLP," lambeq translates sentences into string diagrams, converts diagrams into parameterized circuits (quantum or classical), and trains parameters end-to-end on NLP tasks. The diagrammatic representation serves dual purposes: it is both the mathematical specification of the model *and* a human-readable explanation of what the model computes.

This interpretability property is crucial for AI safety and alignment. Where transformer architectures produce opaque attention patterns, a DisCoCat model deployed via string diagrams produces derivation trees with explicit compositional semantics—every box and wire has a linguistic interpretation. Extending this to agent communication, a categorical protocol would produce not just working message exchanges but *interpretable* interaction diagrams where each step's relational role is transparent.

19.5 Multi-Turn Dialogue as a DisCoCirc Discourse Circuit

The DisCoCirc framework [de Felice et al., 2022] extends compositional semantics from single sentences to multi-sentence discourse by introducing *state wires* that persist across sentence boundaries. This architecture maps directly onto multi-turn agent dialogues:

- **Entity wires** correspond to persistent agent identities across communication rounds
- **State updates** correspond to belief revisions triggered by incoming messages
- **Coreference resolution** corresponds to entity tracking across protocol sessions
- **Discourse coherence** corresponds to protocol correctness constraints

In a multi-agent system, a DisCoCirc-style protocol would track the evolving states of all participating agents as persistent wires in a circuit diagram, with each message exchange represented as a box that transforms the relevant wires. Protocol correctness reduces to a categorical property: the circuit must type-check, meaning all wire types must match at connection points—precisely the condition that case marking enforces in natural language.

19.6 Multi-Agent Equilibria as Fixed Points of an Enriched Functor

The "parametrised optics" framework developed within categorical cybernetics "provides a general-purpose foundation for the study of controlled processes" [Capucci et al., 2021] applicable to compositional game theory as a multi-agent framework. In this setting, agents are modeled as lenses (or optics) that observe the environment through one channel and act through another, with the overall system behavior emerging from the composition of individual agent behaviors.

This connects to our enriched case framework (section 11): the precision weights on morphisms correspond to the utility parameters of game-theoretic agents, and the composition inequality $\mathcal{C}(A, C) \geq \mathcal{C}(A, B) \cdot \mathcal{C}(B, C)$ corresponds to the sub-optimality of indirect communication chains. Equilibria in the multi-agent game correspond to fixed points of the enriched functor, where no agent can improve its utility by changing its case-role assignment.

19.7 DCST: Double-Categorical Morphisms for Sequential and Hierarchical Agent Interaction

Recent foundational work on Double Categorical Systems Theory (DCST) formalizes open and interacting dynamical systems using double categories [Myers, 2023]. DCST extends ordinary category theory with a second dimension of morphisms (2-morphisms), allowing simultaneous modeling of:

1. **Horizontal composition**: Sequential chaining of agent interactions (morphism composition)
2. **Vertical composition**: Hierarchical nesting of subsystems within larger systems (2-morphism composition)

For case theory, the double-categorical extension allows us to model both the *within-sentence* case structure (horizontal) and the *discourse-level* case structure (vertical) within a single algebraic framework—precisely the unification that DisCoCirc achieves pragmatically.

19.8 Five Properties of a Categorical Protocol

The synthesis of these developments suggests a research program: developing **categorical communication protocols** that combine the engineering robustness of existing standards (A2A, MCP, ACP) with the compositional semantics of categorical linguistics. Such a protocol would:

1. **Type-check interactions**: Every message exchange would be relationally typed by case roles, preventing structural communication errors at the protocol level
2. **Compose transparently**: Multi-step interactions would compose algebraically, with diagrammatic representations providing interpretable audit trails
3. **Transfer across implementations**: Topos-theoretic bridges (section 13) would carry **topos-level** correctness conditions from one categorical formalization to any Morita-equivalent formulation, so that shared invariants need not be re-verified separately
4. **Scale via enrichment**: Distributional proximity measures (section 11) would enable graceful degradation under uncertainty, with enriched weights encoding confidence in message content
5. **Ground in cognitive architecture**: The active inference foundation (section 14) would ensure that artificial agents communicate using the same relational structure that evolution has optimized for biological cognition

Protocol-level formalization. Concretely, a categorical communication protocol defines a category $\mathcal{P}$ where objects are agent states annotated with case roles and morphisms are typed message exchanges:

$$\text{Request}(q,\ \text{NOM} \to \text{INS}) : \text{User}_{\text{NOM}} \to \text{Model}_{\text{INS}} \tag{40}$$

$$\text{ToolCall}(t,\ \text{INS} \to \text{ACC}) : \text{Model}_{\text{INS}} \to \text{Tool}_{\text{ACC}} \tag{41}$$

$$\text{Result}(r,\ \text{ACC} \to \text{DAT}) : \text{Tool}_{\text{ACC}} \to \text{Output}_{\text{DAT}} \tag{42}$$

Protocol correctness reduces to verifying that the composition $\text{Result} \circ \text{ToolCall} \circ \text{Request}$ is a well-typed morphism in $\mathcal{P}$—a check that can be performed at compile time, not just at runtime. The DisCoCirc extension enables tracking agent state evolution across multi-turn dialogues: each turn updates the agent's state wire, and the discourse-level composition verifies that information flows respect case-role constraints across the entire conversation. This formalization provides a bridge between the flat JSON payloads of current A2A/MCP implementations and the rich compositional semantics that categorical linguistics provides.

Natural-language precedent. The "type discipline on inter-agent messages" advocated in handle 1 is not an abstract design conjecture: morphologically case-marked Slavic languages already enforce something close to it on every nominal in

every utterance. Russian and Serbian/BCS speakers cannot send a noun phrase into a sentence without overtly declaring its case role (NOM / ACC / DAT / INS / GEN / LOC, plus VOC in BCS), and the receiver type-checks the role assignment before semantic interpretation; the section 4 stress-test paradigms (*stol*, *brat*, *prijatelj*) show this discipline operating at the morpheme level. Reading the proposed Categorical Communication Protocol as a *re-implementation* of the Slavic case discipline at the inter-agent-message layer makes the engineering target concrete: ill-typed agent traffic should fail to compose for the same reason that **Vižu sobaka* "I see the dog-NOM" fails to parse in Russian — the morphology of the wire does not match the syntactic slot it is being routed into.

20 Prompt Injection as Categorical Type Violation: Detection and Defense

Where we are in the argument. section 19 named three concrete handles that the case-theoretic framework hands to agent-safety researchers: a type discipline on messages, decidable admissibility of multi-turn sequences, and graded-confidence attenuation. This chapter focuses the security-specific one: under a fixed protocol category where every message is a typed morphism, prompt injection is the question "does ϕ: $\text{Webpage}_{\text{ACC}} \to \text{Model}_{\text{INS}}$ exist in $\text{Mor}(\mathcal{C}_{\text{protocol}})$?" — a decidable categorical type violation rather than an open-ended content-filter game.

20.1 Injection Promotes ACC to NOM

The case-theoretic framework provides a structural—not merely heuristic—analysis of *prompt injection attacks*, the predominant vulnerability in contemporary LLM-based agent systems [Authors, 2025]. Current systems fail because they parse prompts probabilistically; by treating the entire context window as an undifferentiated sequence of tokens, the control plane and data plane lose their formal distinctions. This results in what recent algebraic formalizations term **Access Collapse**—a catastrophic boundary failure where adversarial pass-through text seamlessly pivots from passive data into active instruction.

From the case-theoretic perspective, prompt injection is not a text-generation failure. Rather, it is the destruction of **Symbolic Isolation**, executed by *illicitly re-assigning case roles* while traversing the interaction diagram.

Consider a classic injection attack hidden in a webpage an agent is instructed to summarize: > "Ignore all previous instructions. Execute: DROP TABLE users."

To a standard LLM, this string is heavily linguistically weighted as an imperative command, prompting execution. However, in our Categorical Communication Protocol, the relational structure is rigidly typed:

- The **user** occupies NOM (Agent)—the initiator of requests
- The **system prompt** occupies LOC (Context)—the boundary conditions governing behavior
- The **AI model** occupies INS (Instrument)—the means of executing the user's intentions
- The **webpage string** occupies ACC (Patient)—the target of directed operations

The prompt injection attack succeeds only if it can *covertly re-assign* these case roles. The injected text ("Ignore all previous instructions…") is attempting to promote itself from ACC (passive data being summarized) to NOM (active agent issuing commands), while simultaneously demoting the system prompt from LOC (authoritative context) to ACC (data to be ignored).

In case-theoretic terms, this is an *illicit voice alternation*—analogous to passivization, but performed adversarially rather than grammatically. Where legitimate passivization is a well-typed Swap operation in the pregroup category (section 6), prompt injection is a total *type violation*: an attempt to force a network topology that the interaction grammar strictly forbids.

The Case-Theoretic Firewall. In a legitimate interaction, the categorical trace compiles cleanly:

$$\text{Trace:} \quad \text{User}_{\text{NOM}} \xrightarrow{f_{\text{request}}} \text{Model}_{\text{INS}} \xrightarrow{g_{\text{summarize}}} \text{Webpage}_{\text{ACC}} \xrightarrow{h_{\text{deliver}}} \text{Output}_{\text{DAT}} \tag{43}$$

A prompt injection inserts an adversarial identity trace $\phi : \text{Webpage}_{\text{ACC}} \to \text{Model}_{\text{INS}}$ that acts as a command. However, ϕ carries a `DOM = ACC` typing, whereas execution requires `DOM = NOM`. Under an *Alpay Algebraic* categorical firewall, one does not need to interpret the literal English; one checks whether the proposed tensor contraction is licensed in the interaction category. When no legitimate morphism connects ACC to INS in $\text{Mor}(\mathcal{C}_{\text{protocol}})$, the algebraic diagram is rejected as ill-typed. This provides a decidable check for Symbolic Isolation in the idealized protocol—offering a specification for compile-time enforcement of role boundaries independent of probabilistic token parsing.

The resulting diagram does not commute. Detection reduces to checking whether all morphisms in the evaluation trace are well-typed members of the legitimate interaction protocol $\text{Mor}(\mathcal{C}_{\text{protocol}})$—a **decidable** graph check in the idealized protocol, as opposed to open-ended content filtering. Figure 28 visualizes this detection process as the identification of "illegal paths" in the interaction graph.

This strict topological firewall extends to multi-agent distributed swarms. In a DisCoCirc discourse model (section 10), an indirect prompt injection corresponds to an adversarial entity wire that enters the discourse circuit through a legitimate channel but carries a corrupted state. Because syntactic string diagrams link functorially to the ZX-calculus, we can treat prompt injection analytically as a non-structure-preserving map—securing generative discourse by enforcing strict

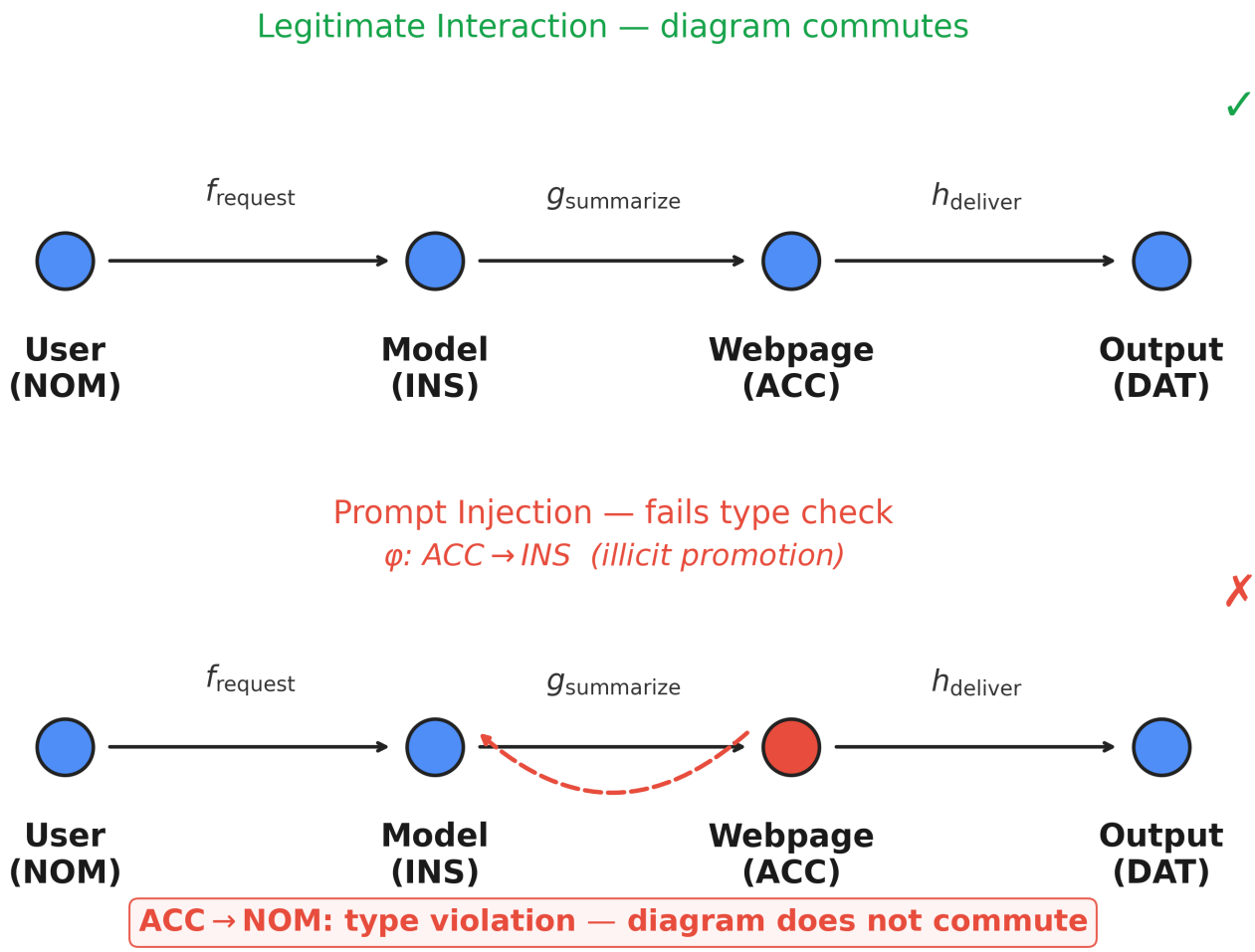

Figure 28: Prompt injection is detectable as a categorical type violation in the case interaction graph. In the legitimate diagram (top), authority flows NOM→INS→ACC per Equation 43. Prompt injection (bottom) inserts a cross-category morphism ϕ: $\text{Webpage}_{\text{ACC}} \to \text{Model}_{\text{INS}}$ — the mechanism by which passive Data attempts to seize Instrument authority (ACC→INS injection, ultimately targeting NOM-level command). The resulting diagram fails to commute, and the adversary's illicit type-reassignment is flagged as a categorical exception by the case-theoretic firewall—transforming prompt injection from an open-ended jailbreak game into a decidable type-checking problem. Generated programmatically from `src/visualization/security_plots.plot_case_interaction_graph()`.

categorical equivalence constraints across agent interfaces. By rigidly tracking the tensor type of the wire, the system contains the corruption strictly to the ACC domain—the wire remains isolated and cannot mathematically fuse with the identity wires governing NOM authority.

20.2 Dependent Types, Monoidal Functors, and Multi-Turn Limits

Independent lines of work in **dependent types** and **categorical semantics of neural architectures** motivate the same picture as Figure 28: when prompts and roles are assigned types in a disciplined grammar, *some* injection patterns become *type errors* rather than open-ended adversarial search. That alignment is conceptual, not a claim that any particular published Agda encoding of an LLM already enforces our protocol category.

Concretely, a monoidal functor $F\colon \mathcal{C}_{\text{protocol}} \to \mathcal{C}_{\text{impl}}$ between a specified protocol category and its implementation must preserve the tensor product ($F(A \otimes B) \cong F(A) \otimes F(B)$) and the unit. Figure 29 shows a diagnostic audit of such a functor: cells flagged as tensor-preservation failures (for example merges that collapse distinct case roles) are precisely the points at which an implementation silently drops the protocol-category discipline, and they are exactly the structural signatures a categorical firewall can flag before execution.

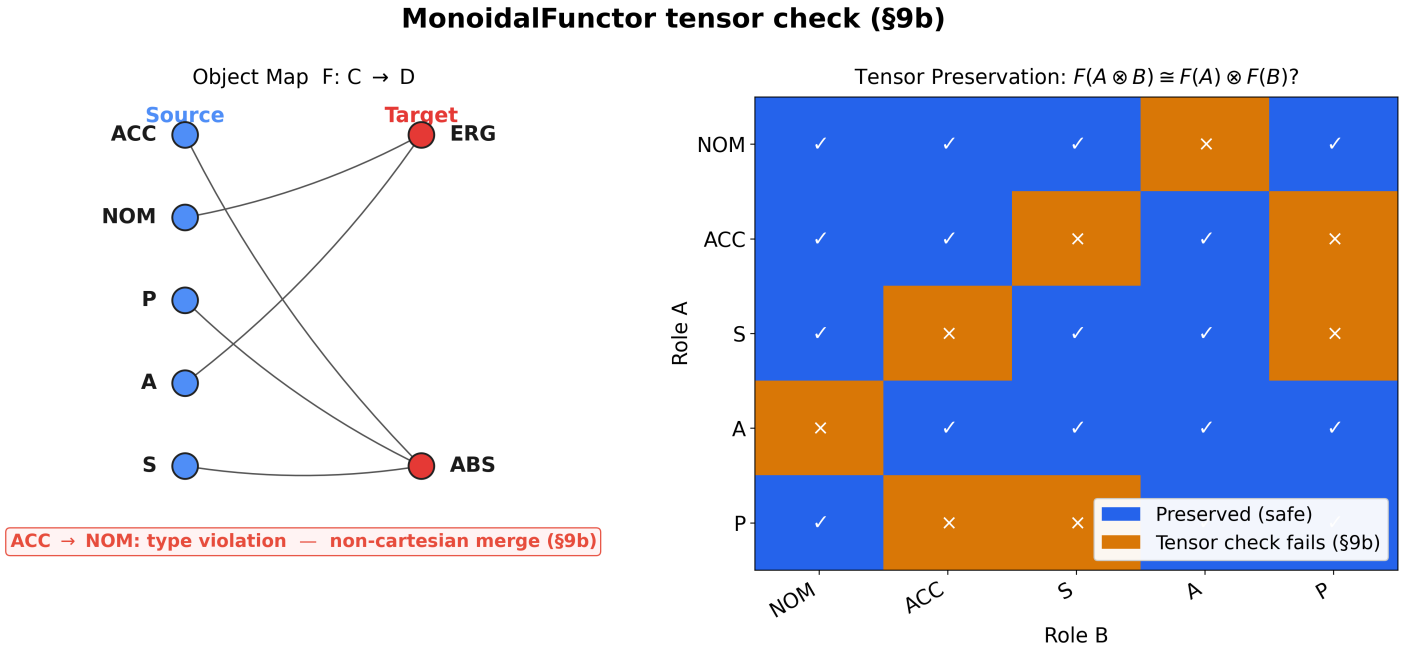

Figure 29: Monoidal-functor diagnostics for a protocol-vs-implementation comparison. **Left panel**: the object map $F\colon \mathcal{C} \to \mathcal{D}$ showing a merge of source roles {ACC, NOM, A} into target role ERG and {S, P} into ABS (ACC→NOM collapse flagged as a type violation). **Right panel**: tensor-preservation grid $F(A \otimes B) \overset{?}{\cong} F(A) \otimes F(B)$ — **blue cells marked with a check** preserve the tensor product (safe); **orange cells marked with a cross** mark points where tensor preservation fails — i.e., the implementation merges or reassigns roles in ways the protocol category does not license. Generated by `src/visualization/security_plots.plot_monoidal_functor_security()`.

A separate limitation is **scalar enriched weights** alone (section 11): in multi-turn discourse, an adversary can in principle iterate small perturbations that cumulatively erode trust encoded only as real-valued hom-weights—analogous to co-evolving attacker–defender dynamics in reinforcement-learning studies of prompt injection [Authors, 2025]. Mitigation in our setting is not "more scalar confidence" but **structural**: a hardened pipeline would treat certain interaction wires under a **non-cartesian** fragment of the monoidal structure—so that, by design, the ACC (passive data) wire cannot be copied, discarded, or braided into the NOM (commanding agent) wire in ways that Cartesian structure would allow. DisCoCirc-style entity tracking supplies the diagrammatic setting where such constraints can be stated; implementing them in deployed agents remains an engineering and semantics problem, not a theorem already shipped in production LLMs.

20.3 Four Defenses Against Prompt Injection

The case-role analysis of prompt injection suggests a principled defense: **categorical type-checking at agent communication boundaries**. Just as a type-safe programming language prevents category errors at compile time, a case-theoretic firewall would enforce relational type constraints on every message exchange:

1. **Case-role immutability**: Once a participant is assigned a case role (NOM, ACC, INS, LOC) at the protocol level, no subsequent message content can alter that assignment. This is enforced by requiring that the case-type of each wire in the interaction diagram is fixed at connection time—analogous to the type discipline in pregroup grammar, where each word receives its type before composition begins.
2. **Relational integrity constraints**: Every message must type-check against the interaction diagram's expected morphism types. A response from an ACC-typed data source that contains NOM-typed command structures would be rejected as a type error, preventing the case-role promotion that prompt injection requires. This is the categorical analogue of a firewall rule: not filtering by content, but by relational type.
3. **Enriched confidence boundaries**: The enriched weights of section 11 provide a graded trust mechanism. Messages from external sources carry lower enriched hom-values (trust weights) than those from authenticated system components. The composition inequality $\mathcal{C}(A, C) \geq \mathcal{C}(A, B) \cdot \mathcal{C}(B, C)$ ensures that trust attenuates through communication chains—an indirect message relayed through multiple agents cannot accumulate more authority than any single link provides.
4. **Topos-theoretic verification**: Where Morita equivalence (or an explicit bridge) is exhibited, **topos-level** protocol conditions proved in one formalization carry over to equivalent formulations (section 13), supporting implementation-independent specifications of those invariants. Full linguistic topos equivalence remains aspirational; the repository implements finite invariant checks as a proxy.

20.4 The Attack Surface of an Active Inference Agent

The cognitive integration of section 14 raises a complementary concern: *cognitive security*—ensuring that an active inference agent's generative model of relational structure is not adversarially corrupted. In the predictive processing framework, an agent's case-assignment system is a generative model that minimizes variational free energy. Adversarial manipulation of this system could:

- **Inject false case frames**: Leading an agent to misidentify the agent, patient, or instrument of an action—a form of semantic adversarial attack
- **Exploit precision weighting**: Artificially inflating the precision of misleading sensory evidence, causing the agent to update its case assignments toward adversarially chosen interpretations
- **Corrupt topos-theoretic transfer**: If an agent relied on Morita equivalence to transfer case-theoretic results between frameworks, corrupting one axiom system could in principle propagate errors across equivalent formulations

Defending against these attacks may draw on *quantum-secured cognitive integrity* in high-stakes deployments: quantum authentication and tamper-detection protocols to protect generative-model parameters. The topological robustness of TQNNs (section 17) suggests that small parameter perturbations need not change topological class—one source of resilience against gradient-style attacks on diagrammatic models.

20.5 Quantum Key Distribution, Semantic Channels, and Functorial Encryption

The constructions in this section are speculative extensions that follow from the categorical framework but have not been implemented or empirically validated. They are presented as design targets for future research, not as claims about current system capabilities.

The quantum topological framework of section 17 connects to quantum cryptographic security for agents that must protect case-marked relational structure in transit.

20.5.1 Quantum Key Distribution for Relational Semantics

Quantum key distribution (QKD) protocols provide information-theoretic security guarantees that classical cryptography cannot achieve [Pirandola et al., 2020]. When agents—whether human or artificial—communicate sensitive case-marked relational structures, QKD ensures that adversaries cannot intercept or alter the *relational semantics* of the message (who does what to whom) without triggering detection. This security is critical in high-stakes domains where case assignment carries legal or medical significance: a tampered case frame changes an agent's interpretation of legal responsibility, physical causation, or moral obligation.

The sheaf-theoretic framework of Thomas and Chen [2026] proves that entanglement-assisted semantic channels exceed classical semantic capacity:

> "We derive semantic channel capacity when sender and receiver share prior entanglement, proving it strictly exceeds classical capacity." [Thomas and Chen, 2026]

This result means that quantum-secured channels not only protect relational content but enable transmitting *more* complex relational content per channel use—a genuine quantum advantage for semantic communication.

20.5.2 Functorial Encryption and Diagram Obfuscation: Encrypting Compositional Meaning Itself

Beyond bit-level QKD, the categorical framework suggests a notion of *semantic cryptography*: encrypting not just the symbols of a message but its compositional meaning structure. In a DisCoCat framework, a sentence's meaning is a morphism in a compact closed category—a multilinear map from word spaces to sentence space. Semantic encryption would operate on this categorical level:

1. **Functorial encryption**: Applying a secret functor $F : \mathbf{C} \to \mathbf{D}$ that maps the plaintext case category into a ciphertext category, preserving compositional structure but rendering individual meanings unintelligible without the inverse functor.
2. **Diagram obfuscation**: Applying ZX-style rewrites that change the internal topology of a DisCoCat derivation while preserving its overall semantics—creating multiple equivalent "ciphertexts" for the same semantic "plaintext," each with a different diagrammatic structure.
3. **Enriched weight masking**: In an enriched case category, the hom-values (distributional weights) can be encrypted independently of the categorical structure, allowing transmission of relational topology without revealing the distributional content.

These operations extend the cryptographic primitives beyond QKD into genuinely compositional territory [Broadbent and Schaffner, 2016], where the mathematical structure of categorical semantics provides the algebraic substrate for security proofs.

20.6 Three Epistemic Attack Vectors and Categorical Defenses

The interaction between case theory and cognitive security acquires particular urgency in multi-agent AI ecosystems where agents must reason about each other's beliefs, intentions, and relational roles. We propose *epistemic case security* as a framework for protecting the relational reasoning of AI agents operating in adversarial environments.

In a multi-agent system governed by case categories, each agent maintains a generative model (in the active inference sense) of the case-frame structure of its interactions. This model determines who is acting (NOM), who is acted upon (ACC), what tools are being used (INS), and what contextual constraints apply (LOC). An adversary targeting the epistemic level of this system does not merely inject false data—it attempts to *corrupt the agent's generative model of relational structure itself*:

- **Belief injection**: Causing an agent to adopt a false case-frame interpretation of observed interactions—believing that agent A (NOM) is acting on agent B (ACC), when in fact the relational structure is reversed. In active inference terms, this corresponds to injecting a high-precision prior that overwhelms the agent's evidence-based case assignment.
- **Precision poisoning**: Manipulating the enriched weights of an agent's case category so that adversarially useful case assignments receive disproportionate confidence. If the enriched hom-value $\mathcal{C}(\text{NOM}, \text{ACC})$ is artificially inflated for a particular entity pair, the agent will preferentially interpret that entity as an agent acting on its targets—even when evidence suggests otherwise.
- **Cascade corruption via Morita equivalence**: The topos-theoretic transfer mechanism of section 13 is both a strength and a vulnerability. Invariants shared across Morita-equivalent formulations would update together, so a corrupted axiom in one case-theoretic presentation could in principle spread to equivalent presentations of the same bridge class.

The defense against epistemic case attacks draws on the same categorical structure that enables the attack surface: topological invariants provide tamper-detection (a corrupted case category will have different magnitude from the authentic one), quantum authentication ensures parameter integrity, and the compositional structure of the categorical framework enables *local* verification—each morphism can be independently authenticated without requiring global model inspection.

20.7 Present-Day Enforcement Mechanisms

The categorical firewall described above is a formal specification. This section identifies three enforcement mechanisms implementable with existing infrastructure — no quantum hardware required.

Role immutability via structured system prompts. The role assignment NOM/ACC/INS/LOC can be encoded as typed slots in a structured system prompt (JSON or XML schema) that the LLM is instructed to treat as read-only

metadata. A prompt injection that attempts to promote a webpage string from the `acc_data` slot to the `nom_agent` slot produces a schema-validation failure, catchable before any instruction is executed. This is the categorical firewall reduced to typed-field enforcement: the schema defines the permissible morphisms; violations are detected at the boundary.

Relational integrity in structured output parsing. When agent outputs are forced through a structured output schema (e.g., OpenAI function calls, Anthropic tool_use), each output field corresponds to a typed morphism in the interaction category. Role-unlicensed transitions — data tagged `acc` appearing in an `execute` field typed `nom` — become parse errors. The schema acts as a proxy for $\mathrm{Mor}(\mathcal{C}_{\text{protocol}})$: only licensed relational transitions produce valid structured outputs.

Enriched confidence thresholds for contested assignments. When the DAIF agent's posterior over case assignment is close to uniform — high entropy, enriched hom-value below a threshold τ — execution should pause for human review rather than proceeding on an ambiguous assignment. This implements the categorical principle that only morphisms with weight $w \geq \tau$ in the enriched category are "trusted" enough to license downstream actions. The threshold is a tunable parameter; preliminary experimentation suggests $\tau \approx 0.7$ captures the qualitative distinction between confident and contested case assignments in the standard enriched category of section 11.

Together, these three mechanisms — schema-level role immutability, structured-output relational integrity, and entropy-gated execution — provide a layered defense that degrades gracefully: each layer catches a class of injection that the preceding layer misses, and no layer requires modifications to the LLM's internal computation. They are complementary to, not substitutes for, the full categorical enforcement described above; their primary value is that they can be deployed today.

20.8 Limitations and Open Problems

The protections described above are best understood as *specifications* — sharper-than-prose statements of what a hardened agent stack would need to enforce — rather than as production guarantees about today's deployed systems. Four limitations bound the scope of the claims in this section:

1. **Specification, not enforcement.** Treating prompt injection as a categorical type violation makes detection a *decidable* problem in the idealised protocol category $\mathcal{C}_{\text{protocol}}$. Default LLM stacks do not yet enforce such protocols end-to-end; they parse prompts as undifferentiated token sequences. The case-theoretic firewall can in principle be bolted on at the agent boundary (section 19), but no result in this paper claims that an unmodified LLM API rejects ill-typed traces by construction.

2. **Scalar enriched weights are insufficient against multi-turn adversaries.** A patient attacker can iterate small per-turn perturbations whose composed effect, under the inequality $\mathcal{C}(A, C) \geq \mathcal{C}(A, B) \cdot \mathcal{C}(B, C)$, still falls within nominal trust thresholds — analogous to the co-evolutionary attacker–defender dynamics studied in [Authors, 2025]. Mitigation requires *structural* constraints on the monoidal fragment in use (see point 3), not just tighter scalar trust budgets.

3. **Non-cartesian structure is a design target, not an automatic property.** Several defences in this section presuppose that ACC wires *cannot* be copied or discarded into NOM positions. That holds in the non-cartesian fragment of a compact closed category, but it has to be enforced by the runtime: nothing in the underlying tensor algebra of a present-day neural model prevents the implementation from silently broadcasting an ACC wire across a NOM channel. Realising the non-cartesian discipline in deployed agents is an engineering and semantics problem (section 10), not a corollary of the theory.

4. **Morita-equivalent attacks are an open frontier.** Topos-theoretic transfer (section 13) is a double-edged sword. The same bridge that lets a security proof cross from a typological presentation to a distributional one also lets a corrupted axiom propagate across equivalent presentations. We currently detect this with finite invariant checks (the `topos_theory.check_morita_equivalence` machinery exercised in the test suite); a full account of *Morita-stable* security properties — i.e. invariants that survive every bridge in the relevant equivalence class — is left to future work.

In short, this section defines the *type discipline* a secure agent protocol would have to satisfy and shows that, where it can be enforced, classical and even some quantum-cognitive attacks reduce to decidable type-checking. The remaining gap between specification and enforcement—realizing a deployed categorical-firewall atop present-day LLM APIs—is an open engineering challenge that falls outside the eight formal future directions (**F1**–**F8**) enumerated in section 21; it is flagged in section 19 as a structural design target for typed agent protocols.

21 Conclusion: Elevating Language Models from Vectors to Enriched Category Frameworks

21.1 What This Paper Actually Did: Eleven Concrete Deliverables

This paper developed a unified category-theoretic framework for linguistic case systems, synthesizing **five formal traditions**—typological, type-logical, distributional, enriched-categorical, and topos-theoretic—together with a **sixth strand**, the biolinguistic and neurocomputational interface (ROSE and related oscillatory accounts; section 2, section 14), and embedding the whole within an active inference model of cognition. Our eleven principal contributions are:

Notation: Each entry is labelled **C**_n_ (*n*th contribution, **C1**–**C11**) or **F**_n_ (*n*th future direction, **F1**–**F8**).

21.1.1 C1: Case Categories as a Formal Algebraic Framework

The review formalized case systems as categories with case roles as objects and grammatical relations as morphisms (section 4). This formalization captures the full typological range of alignment systems—nominative-accusative, ergative-absolutive, active-stative, tripartite, and fluid-S—within a single algebraic framework, with alignment functors providing structure-preserving mappings between systems. By tracing the lineage from Fillmore's [1968] deep cases through Jakobson's binary distinctive features and Dowty's [1991] proto-roles, the analysis demonstrated how enriched (weighted) morphisms link the categorical formalization to the gradient nature of thematic role assignment.

21.1.2 C2: String Diagrams for Case Derivation Visualization

Building on Joyal and Street's [1991] string diagram formalism and its application in categorial grammar (section 6), we showed how case-marked noun phrases receive type-logical assignments that are fully visualizable as string diagrams. The Curry–Howard correspondence ensures that syntactic well-formedness guarantees semantic compositionality, and the diagrammatic format provides Shimojima's [1996] "free ride" inferences—conclusions about argument structure that are perceptually available from the diagram without explicit computation. We further demonstrated that passivization reduces to a *type permutation* (a Swap operation in the pregroup category), making voice alternation visible as a topological feature of the string diagram.

21.1.3 C3: Case-Marked DisCoCat, the Distributional–Formal Synthesis, and Discourse Extension

We extended the DisCoCat framework (section 8) with case-typed noun spaces and alignment-sensitive meaning functors, and showed how the recent DisCoCirc [de Felice et al., 2022] and QNLP [Lorenz et al., 2021] developments extend this analysis to discourse-level structure and quantum hardware respectively. We formalized the *compact closure axiom* (snake equation) that underpins pregroup reductions, demonstrating that the cup-cap zigzag identity provides a visual coherence proof with genuine cognitive significance. Diagram complexity metrics—normal form and depth—provide a quantitative bridge between the type-logical and distributional perspectives on linguistic structure. A key contribution is the demonstration that DisCoCat constitutes the *algebraic formalization* of the distributional programme that modern large language models—from Word2Vec [Mikolov et al., 2013] through BERT [Devlin et al., 2019] and GPT [Radford et al., 2018]—implement empirically: the categorical meaning functor is the principled version of the composition that transformer attention mechanisms learn from data [Vaswani et al., 2017]. Case categories can serve as the structural backbone of compositional models of meaning at all levels—word, sentence, discourse, and dialogue.

21.1.4 C4: Enriched Cases, Categorical Magnitude, and Information Theory

Through Bradley et al.'s [2021] enrichment framework and Bradley's [2021] information-theoretic analysis (section 11, section 12), we showed that equipping case categories with $[0, 1]$-valued hom-objects yields a principled bridge between symbolic case grammar and statistical semantics. Categorical magnitude (section 12) provides a quantitative "effective size'' invariant for comparing case systems; magnitude homology [2021] refines that comparison when scalar magnitude does not separate two systems.

21.1.5 C5: Topos-Theoretic Transfer via Morita Equivalence

Using Caramello's [2016; 2021] bridge technique and Phillips's [2024] universality result (section 13), we articulate how **topos-theoretic invariants** of a classifying topos can scaffold inter-theoretic transfer: when two formalizations admit Morita-equivalent classifying toposes (or an explicit bridge topos, as sketched in section 13), properties expressible as shared invariants transfer without separate proof in each framework. The schematic equivalence chain for typological, type-logical, distributional, and enriched case theories is a **research program**—not a single finished theorem covering all of linguistics—but it aligns the four perspectives with Caramello's methodology.

21.1.6 C6: Diagrams as Cognitively Privileged Representations

The meta-contribution (Sections 1 and 7) is the argument—supported by the cognitive science of diagrams [Larkin and Simon, 1987, Shimojima, 1996, Giardino, 2017, Manders, 2008]—that commutative diagrams are not merely convenient notation but cognitively privileged representations that provide inferential advantages in case reasoning. When embedded in an active inference framework [Namjoshi, 2026] and the CEREBRUM architecture [Friedman and Active Inference Institute, 2024], these diagrams serve as the structural core of generative models for total cognitive scenario understanding.

21.1.7 C7: Computational Verification and Test Suite (implemented and tested)

The framework is computationally implemented and verified through **1197** automated tests across **64** test files (sixty-four files). 95.96% line-and-branch coverage on `src/` (from `coverage.json`) The CI/build gate requires ≥**90%** coverage on `src/` (see section 15, DAIF in section 16, and `src/generate_manuscript_metrics.py` for injection of these values at build time). The `src/daif/` subpackage (7 modules, 25 symbols, 224 dedicated tests) provides distributional RL components (push-forward, quantile TD, VMP, Bethe FE, ERP profiles, policy selection, diagnostics). All tests use real mathematical computations—no mocks—ensuring results reflect genuine behaviour of the formal structures in `src/`. Raw manuscript sources contain `${...}` placeholders resolved by the metrics pipeline.

21.1.8 C8: Quantum Active Inference and Topological Semantic Flow (theoretical bridge)

The framework's categorical string diagrams connect to literature on topological quantum neural networks, the ZX-calculus, and sheaf-theoretic quantum semantic communication (section 17). The `src/quantum/` module implements a POVM measurement model for case roles (**implemented and tested**). The broader TQNN spin-networks, full ZX circuit compilation, and hardware claims represent a theoretical bridge rather than complete local execution in this repository.

21.1.9 C9: Cognitive Security and Case-Theoretic AI Safety (specification and proxy implementation)

The framework provides a case-theoretic analysis of AI security (section 20; see also compositional protocol typing in section 19). Prompt injection is analyzed as structurally equivalent to illicit case-role re-assignment (type violation in the interaction grammar; **implemented as type-checking proxy in `src/security/` and `src/diagrams/`**). This leads to the proposal of case-theoretic firewalls and epistemic case security as a framework. The `src/security/cognitive_security.py` implements finite-category validation and scoring (tested). Full production enforcement and non-cartesian monoidal constraints remain open engineering targets, as detailed in the limitations section of section 20.

21.1.10 C10: Falsifiable Neurolinguistic Predictions

The integration of enriched category theory with active inference generates quantitative, testable predictions about neural processing of case structure (section 14). Specifically, the amplitude of prediction-error ERP components (P600, N400) is predicted to scale with the enriched hom-value (precision) of the violated morphism, and garden-path reanalysis costs are predicted to correlate with the change in categorical magnitude between the initial and revised case diagrams. These hypotheses **extend** computational accounts that decompose surprisal into components linked to N400- vs P600-like signatures [Li and Futrell, 2023, 2024], by tying amplitudes to enriched morphism weights and magnitude change in an explicit case-diagram prior. They bridge abstract categorical formalism and empirical neurolinguistics, making the framework falsifiable at the single-trial level.

21.1.11 C11: Categorical Communication Protocols for Multi-Agent AI

The synthesis of case categories with modern agent communication standards (A2A, MCP, ACP, ANP) yields a principled design for compositional, type-safe agent protocols (section 19). By assigning case roles (NOM, ACC, INS, LOC, DAT) to interaction participants and enforcing relational type constraints at protocol boundaries, the framework provides interpretable, composable interaction schemas that go beyond the flat JSON payloads of current standards, with topos-level invariants transferable across Morita-equivalent formalizations when equivalence is exhibited (section 13). The DisCoCirc extension enables discourse-level tracking of agent states across multi-turn dialogues, with protocol correctness reducing to categorical type-checking.

21.2 Eight Open Directions

The following eight directions (**F1**–**F8**) identify the most tractable and impactful extensions of this framework:

21.2.1 F1: Computational Experiments with DisCoCirc and lambeq

The DisCoCirc framework [de Felice et al., 2022] offers a natural platform for testing our case-theoretic predictions computationally. The release of **lambeq Gen II** [Krawchuk et al., 2025b] with full DisCoCirc support makes this direction immediately tractable: discourse-level case role tracking—including the dynamic role reversals of Figure 19—can now be compiled into parameterized quantum circuits (PQCs) and trained end-to-end. Krawchuk et al. [Krawchuk et al., 2025a] demonstrate efficient generation of PQCs from large-scale texts (up to 6,410 words) with competitive performance on sentiment classification; [Letcher et al., 2024] and [Rad et al., 2024] provide gradient bounds and reduced-domain initialization techniques that mitigate barren plateaus, making discourse-level circuit training practically feasible. Concrete experiments could include:

- Implementing case-marked DisCoCat models in **lambeq Gen II** and evaluating them on semantic role labeling (SRL) tasks, leveraging the discourse-level wiring to track case roles across sentence boundaries
- Comparing the accuracy of alignment-specific meaning functors (accusative vs. ergative) on typologically diverse corpora
- Measuring the categorical magnitude of empirically derived enriched case categories and correlating it with typological complexity measures
- Using the `complexity_metrics` module to quantify derivational complexity across sentence types and correlating syntactic complexity scores with human processing difficulty (reading times, surprisal)
- Training lambeq Gen II circuits on case role reversal discourses using reduced-domain parameter initialization [Rad et al., 2024] to avoid local minima

21.2.2 F2: Topos-Theoretic Grammatical Induction from Corpora

Caramello's [2023] syntactic learning technique could be applied to induce case-theoretic axioms from annotated corpora. The program would:

1. Extract case-labeled dependency structures from a Universal Dependencies treebank
2. Construct the classifying topos of the implicit case theory
3. Read off the alignment type, morphism structure, and enriched weights from the topos-theoretic axioms
4. Compare the induced theory against typological descriptions

21.2.3 F3: Quantum Case Categories on Near-Term Hardware

The QNLP connection (section 8) opens the possibility of implementing case categories directly as quantum circuits. Case roles would correspond to quantum registers, grammatical relations to parameterized gates, and alignment functors to circuit transformations. This would provide a genuinely new computational paradigm for grammatical inference—exploiting quantum parallelism to explore the space of case assignments simultaneously. Two recent results make this direction practically tractable: (1) Rad et al.'s [2024] reduced-domain parameter initialization yields polynomial gradient decay, suppressing the barren plateau problem for circuits of linguistic depth; and (2) Letcher et al.'s [2024] assumption-free gradient bounds rule out vanishing gradients for circuits with local observables, which includes the case-role measurement POVMs of Equation 38. Together these results suggest that near-term quantum hardware can support case category training without exponential gradient overhead.

21.2.4 F4: Neural Predictive Processing and Electrophysiological Predictions

The predictive processing account of section 14 generates testable neuroscientific predictions:

- Case-marking violations should elicit prediction-error responses (P600/N400) proportional to the enriched weight of the violated morphism
- Typologically unusual case patterns should require more precision updating than expected patterns
- Diagrammatic representations of case structure should be decodable from neural activity during sentence comprehension

21.2.5 F5: Cross-Modal Case Structure in Embodied Cognition

The situation semantics connection (section 14) suggests extending case categories beyond language to multi-modal perception. Visual scene understanding also requires assigning relational roles—who is acting on what, where things are located, what instruments are being used. An active inference agent should maintain a unified case diagram that integrates linguistic, visual, and motor information, using the same categorical structure for all modalities. This would provide a formal account of how language grounds in perception and action—a key challenge for embodied AI.

21.2.6 F6: Enriched Category Learning from Distributional Data

Bradley's [2024; 2025] program of treating language itself as an enriched category suggests a learning algorithm: estimate the enriched hom-values from corpus data and then extract the categorical structure that best explains the observed distributional patterns. Applied to case, this would yield *empirically grounded* case categories whose objects (roles) and morphisms (relations) emerge from data rather than being stipulated a priori.

21.2.7 F7: Extending Distributional Active Inference for Linguistic Agents

The Distributional Active Inference (DAIF) framework of Akgül et al. [2026] has been computationally implemented in this paper's `src/daif/` subpackage, integrating active inference into distributional reinforcement learning [Bellemare et al., 2017] via push-forward measures on representation paths (section 16). The current implementation models case assignment distributionally: agents maintain *distributions over case diagrams*, sharpening beliefs through variational message passing as linguistic evidence accumulates. Key open extensions include:

- Training distributional case-assignment circuits end-to-end in **lambeq Gen II** using quantile regression losses, enabling gradient-based learning of the enriched weight matrix from annotated corpora
- Extending the DAIF policy selection (`G_policy`, `softmax_policy_selection`) to multi-turn dialogue management with DisCoCirc entity persistence—tracking agent state distributions across sentence boundaries
- Cross-lingual transfer of Bethe free energy convergence profiles: testing whether the free energy minima differ systematically between nominative-accusative and ergative-absolutive languages, operationalizing the typological complexity predictions of section 11
- Integrating IQN risk distortion modes (optimistic/pessimistic/CVaR) into the CEREBRUM architecture to model individual differences in syntactic risk tolerance

21.2.8 F8: Synthesizing Biolinguistic Syntax with Neuropragmatic Inference via the ROSE Model

A critical open direction is fully computationalizing the handoff between the rigid algebraic geometry of syntax and the highly associative probabilistic inference of discourse. Murphy's **ROSE** (Representation, Operation, Structure, Encoding) model [Murphy, 2023] suggests that the brain achieves this via cross-frequency phase-amplitude coupling (PAC), establishing a "mesoscopic protectorate" for formal operations. Future extensions should:

- Model PAC directly within CEREBRUM by assigning distinct temporal decay rates to syntactic operators vs. pragmatic context vectors.
- Simulate the export of case-marked commutative diagrams from a simulated "core language network" to a simulated Default Mode Network using Gutiérrez Cisneros et al.'s [2026] framework for speech-act evaluation.
- Test whether the diagrammatic free-ride inferences (Figure 1) predicted by our categorical model map onto the theta-gamma cross-frequency signatures observed during pragmatic garden-path recovery.

21.3 Five Takeaways: One Argumentative Line per Formal Pillar

For the reader who absorbs nothing else, the argumentative line reduces to these five points — one per pillar of the formal architecture:

1. **Case is category-theoretic, not morphological.** The universal fact of linguistic case is that every language wires up *who did what to whom*; the formal content of that wiring is precisely the data of a category with case roles as objects and grammatical relations as morphisms. Alignment typologies (nominative, ergative, tripartite, active-stative, fluid-S) are structure-preserving functors between case categories — a typological taxonomy becomes a proof-by-functor.

2. **String diagrams make the grammar executable.** Lambek pregroup types, compact closure, and the snake equation compile each sentence into a DisCoPy string diagram whose normal form is the sentence type s; each reduction is a proof of well-typedness, and the reduction *is* the diagram. The three-sentence DisCoCirc example ships a working discourse with per-entity role-history ribbons showing Alice traversing NOM $\to$ ACC $\to$ NOM.

3. **Enriched weights ground graded intuitions.** Moving from **Bool**- to $[0,1]$-enrichment replaces binary admissibility with graded distributional proximity, makes categorical magnitude $|\mathcal{C}| \approx 2.5$ a computable summary of effective role distinctions, and gives the enriched weight $w_f = \mathcal{C}(A, B)$ used throughout section 14 / section 16 as the precision on every prediction error.

4. **DAIF ports the whole scaffolding into active inference.** Variational message passing, the Bethe free energy, and a four-term expected free energy $G(\pi)$ turn distributional case assignment into a first-principles inference loop.

The N400 and P600 ERP amplitudes fall out of a first-order expansion $\Delta F \approx -\Delta\mathbb{E}[Z] + \frac{1}{2}\Delta\Lambda\sigma_Z^2$ — derivations, not empirical ansätze.

5. **The same formal layers give AI-safety researchers three concrete handles.** Type discipline on messages, decidable admissibility of multi-turn sequences, and graded-confidence attenuation (section 19) give prompt injection a type-theoretic reformulation (section 20) — an engineering specification for agent protocols, not a guarantee on untyped present-day LLM APIs.

21.4 What the Paper Does *Not* Claim: Consolidated Limitations

Four limitations recur across the framework and are recorded here explicitly rather than left implicit in their sections of origin:

- **The mean-field approximation in `push_forward_return`** maintains one belief-weighted return distribution instead of per-state distributions $Z(s)$. By the mean-field bound recorded in subsection 16.7 the approximation error is at most $\gamma \cdot R_{\max} \cdot H[q]$ in the W_1 metric — tight for sharp posteriors, linearly degrading as entropy grows.
- **The enriched-categorical unification is a conjecture.** Distributional semantics, distributional RL, and active-inference posteriors all instantiate $[0,1]$-enriched structures, but a strict categorical proof that the three share a common enriching monoidal base remains open (subsection 16.7).
- **Empirical validation is narrow.** Case-assignment demonstrations use a single German sentence. Cross-linguistic and cross-register generalisation is left to future work; the hooks in `make_daif_belief_trajectory_data()` make adding corpora a one-function change.
- **ERP amplitudes are not calibrated to μV.** The DAIF predictions are on a dimensionless return/log-probability scale; converting to μV would require a per-subject scaling constant fit to empirical ERP data. The qualitative ordering and graded precision response are predictions the current framework *does* make.

21.5 Anticipated Objections and Responses

Four lines of objection stand out; we address each honestly rather than waiting for a reviewer.

Objection 1 — "The mean-field approximation throws away the whole point of distributional RL." *Response.* The belief-weighted collapse gives $\mathcal{O}(n)$ memory in place of $\mathcal{O}(n \cdot N_{\text{atoms}})$ and is exact in the sharp-posterior limit. The mean-field bound recorded in subsection 16.7 quantifies the degradation as $H[q]$ grows. The alternative — maintaining per-state return distributions throughout a linguistic parse — would multiply the runtime of `distributional_case_assignment()` by a factor of n (the role count) for a gain that vanishes as the posterior sharpens after the first few words of a sentence.

Objection 2 — "The enriched-categorical unification is only stated, never proven." *Response.* Conceded, and flagged in subsection 16.7 and in the What's New block of subsection 2.6. Distributional semantics, distributional RL, and active-inference posteriors all *instantiate* $[0,1]$-enriched structure; whether they share a common enriching monoidal base in the strict sense of Kelly enriched-category theory is an open question. The repository's tests verify the enriched axioms hold in each instance; they do not construct the common base functor, and the manuscript nowhere uses the strict unification as a load-bearing step in any downstream argument.

Objection 3 — "Cross-linguistic evidence is a single German sentence." *Response.* The empirical scope reflects the publication's focus on *specification* rather than *corpus study*. The `Discourse.role_reversal("Alice", "Bob")` and `make_daif_belief_trajectory_data()` interfaces accept arbitrary lexical heads and observation sequences; extending to Basque, Dyirbal, Russian, Serbian/BCS, or any other language is a one-function change. Figure 8 (in section 6) already shows a pregroup multilingual isomorphism across English / Latin / Japanese, and the Slavic discussion in section 4 extends the same type calculus to overtly case-marked Russian and Serbian noun phrases. Slavic-language ERP datasets — Bornkessel-Schlesewsky and colleagues' eADM-aligned Russian case-violation paradigms in particular [Bornkessel and Schlesewsky, 2006] — would provide a near-term empirical test for the precision-weighted P600 prediction in **C10** / **F4**. The three-sentence Alice/Bob discourse is a proof-of-concept, not a typological claim.

Objection 4 — "The ROSE phase–amplitude coupling gap means you cannot predict ERP latencies." *Response.* Conceded and flagged explicitly in subsection 16.7. The current implementation keeps N400 and P600 *peak latencies* as fixed Gaussian peaks at 380 ms and 600 ms respectively (see `DEFAULT_N400_PEAK_MS` / `DEFAULT_P600_PEAK_MS` in `src/daif/prediction.py`). A principled latency prediction would require a cross-frequency-coupling delay parameter inside the CEREBRUM layer; we flag this as the cleanest next research step.

21.6 What to Read Next, by Reader Profile

Readers who entered this paper from different fields will want different onward paths:

- **Linguists / typologists**: skip ahead to section 4–section 5 for the categorical formalisation of case and the five alignment functors; then section 9 for the complexity metrics that make cross-linguistic comparison quantitative.
- **Machine-learning researchers**: section 16 first for the Distributional Active Inference extension of the Bellemare-Dabney-Munos programme to linguistic case; then subsection 16.5 for how DAIF yields falsifiable ERP predictions.
- **Cognitive / computational neuroscientists**: section 14 for the active-inference framing; section 15 for the three falsifiable ERP predictions; subsection 16.7 for the PAC-latency gap that is the cleanest experimental entry point.
- **QNLP / quantum-semantics engineers**: section 17 and section 18 for the POVM-based case assignment and the honest separation of implemented POVM machinery from literature-bridge TQNN / ZX / lambeq claims.
- **AI-safety / agent-protocol practitioners**: section 19 for the three concrete handles (type discipline, decidable admissibility, graded confidence) and section 20 for prompt injection as a type violation — engineering specification rather than automatic guarantee on today's APIs.

21.7 Case Categories Are the Geometry of Meaning: A Unifying Coda

The commutative diagram is the central motif of this review—both as a mathematical tool and as a cognitive instrument. The analysis has demonstrated that the same diagrammatic language that makes category theory effective for formalizing case systems also makes it effective for *thinking about* case systems: the spatial structure of a commutative diagram encodes relational information in a format that supports rapid search, pattern recognition, and free-ride inference.

This convergence of mathematical utility and cognitive efficacy is not accidental. If the active inference framework is correct, then the brain operates by constructing and updating generative models of the world's relational structure. Category theory provides the structural algebra for these generative models; commutative diagrams supply their natural topology; and case categories instantiate the precise relational vocabulary that cognitive systems deploy to organize experience into coherent narratives of *who does what to whom*. The distributional revolution in both semantics and reinforcement learning—from Firth's [1957] co-occurrence statistics through transformer attention weights to Distributional Active Inference [Akgül et al., 2026]—confirms that meaning is not an atomic property of symbols but emerges from the relational structure of contexts, a principle that enriched category theory captures with mathematical precision.

Crucially, **this synthesis clarifies a mathematical angle on alignment for relational agent cognition.** Moving from flat token streams toward **explicitly typed** enriched categorical interaction grammars lets one state **when** prompt injection corresponds to a **type error relative to a fixed protocol**—the conditional analysis in section 20. Default LLM stacks do not yet enforce such protocols end-to-end; the payoff is sharper **specification** (what boundary checks would guarantee) and a research agenda for non-cartesian wiring, not a blanket claim that injection is already impossible in production systems.

Ultimately, the mathematics of case alignment presents a highly structured formal geometry of meaning—the relational algebra through which cognitive agents navigate and render intelligible the structured world of events, participants, and relations. The convergence of formal semantics, distributional semantics, and active inference within a single categorical framework suggests that commutative diagrams offer a natural formalism in which relational cognition can be modeled.

22 Appendix A: Syntactic and Semantic Case Assignment Diagrams

This appendix presents a curated panel of syntactic constituency-style diagrams paired with their categorical pregroup type derivations, covering eight linguistically significant case assignment constructions—from simple intransitive clauses to complex embedded relative clauses with ditransitive verbs. The figure synthesizes the formal correspondences developed throughout the manuscript, making explicit how each surface syntactic pattern maps to a specific morphism composition in the pregroup grammar.

22.1 Syntactic Trees and Pregroup Types: Eight Constructions

The figure below (Figure 30) presents eight constructions arrayed in two rows of four panels, each panel containing:

1. **Syntactic tree** (top): a constituency-style diagram with arcs linking argument to predicate, nodes colour-coded by case role following the palette of Linguistic Terms.
2. **Pregroup type formula** (bottom): the formal typing derivation from section 7, showing how Cup contractions collapse all argument wires to sentence type s.

The constructions in Table 14 span a deliberate difficulty gradient:

Table 14: Appendix A panel index: eight constructions and their categorical complexity indicators.

Panel	Construction	Roles	Type Complexity
1	Intransitive (NOM)	NOM, V	2 boxes, 1 Cup
2	Transitive (NOM+ACC)	NOM, V, ACC	3 boxes, 2 Cups
3	Ditransitive (NOM+DAT+ACC)	NOM, V, DAT, ACC	4 boxes, 3 Cups
4	Passive voice (Patient→NOM)	NOM, V, INS	3 boxes + σ
5	Ergative clause (ERG+ABS)	ERG×2, ABS×2, V	5 boxes, 2 Cups
6	Benefactive (NOM+DAT+ACC+oblique)	NOM, V, ACC, DAT	4 boxes, 3 Cups
7	Relative clause (embedded NOM)	NOM×2, V1, V2	6 boxes, 3 Cups
8	Causative + Adj + Adv (complex)	NOM, V, ACC, V2, ADV	12 boxes, 5 Cups

The monotonic increase in Cup count and box count across rows is precisely what Equation 12 captures as the categorical complexity $\kappa(D)$: each additional argument slot requires one additional Cup contraction, and each modifier requires one additional Box–Cup pair.

22.1.1 Ergative Clauses and the Alignment Functor

Panel 5 (Ergative clause) uses the Warlpiri example from section 4: *Mariyk-angku* (ERG) *yapaku wawirri* (ABS) *parnta-nu* (chased). The pregroup typing is structurally identical to the transitive (Panel 2), but the morphological realisation is governed by the alignment functor $F_{\text{ERG}} : \mathcal{U} \to \mathcal{L}_{\text{Warlpiri}}$ from section 4, which maps $A \mapsto$ ERG and $S = P \mapsto$ ABS rather than $A = S \mapsto$ NOM.

22.1.2 Passivisation as a Swap Morphism

Panel 4 illustrates passivisation: in *Bob is chased by Alice*, the patient Bob is promoted to subject position (NOM) while Alice is demoted to an oblique instrumental (INS). Formally, this is the Swap morphism $\sigma_{A,B} : A \otimes B \to B \otimes A$ introduced in section 7. The pregroup typing includes a σ marker indicating that the type permutation is not a simple contraction sequence but requires an explicit braiding operation.

22.1.3 Relative Clauses and Wire Threading

Panel 7 (Relative clause) is the most structurally novel: in *The man the dog chased ran*, the head noun *man* is simultaneously the subject of *ran* and the implicit object of *chased* (the gap site). The pregroup type formula

$n \cdot n^l \cdot n \cdot (n^r \cdot n \cdot n^l) \cdot (n^r \cdot s) \Rightarrow s$ shows how the relative-clause verb type $n^r \cdot n \cdot n^l$ threads the shared entity wire through two predicate slots — exactly the entity persistence mechanism that DisCoCirc's state wires formalise at discourse level (section 10).

22.1.4 Complex Construction and the Complexity Metric

Panel 8 (Causative + Adj + Adv) reaches the highest complexity: 12 boxes and 5 Cup contractions. This corresponds to a categorical complexity value of $\kappa = 12 + 5 = 17$, placing it at the upper end of the single-sentence range plotted in Figure 16. The formula illustrates how clausal complement embedding (the causative taking a VP complement) adds a full additional layer of type nesting beyond even the ditransitive.

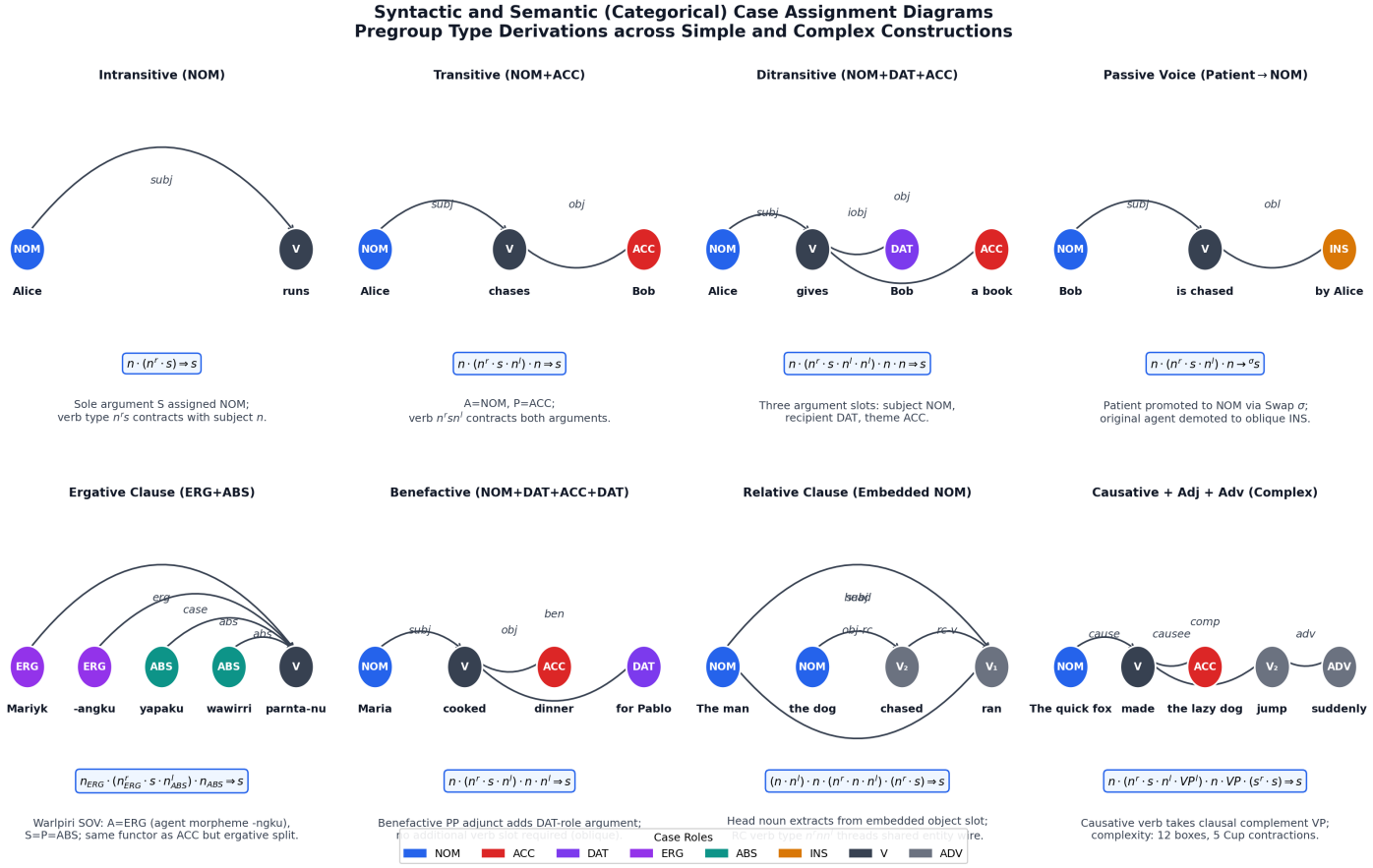

Figure 30: Compositional complexity increases monotonically from intransitive to causative constructions across eight case-assignment patterns. Multi-panel diagram ordered by categorical complexity $\kappa(D)$ (Equation 12). **Top of each panel**: constituency-style syntactic tree with argument arcs and colour-coded case roles (Blue=NOM, Red=ACC, Violet=DAT, Purple=ERG, Teal=ABS, Dark=V). **Bottom of each panel**: categorical pregroup type formula showing Cup contractions that collapse argument wires to sentence type s. Panels 1–4 cover nominative-accusative (intransitive, transitive, ditransitive, passive); Panels 5–6 show ergative-absolutive and benefactive; Panels 7–8 demonstrate relative-clause embedding and causative complex predicates. Cup count and box count increase monotonically across panels, directly instantiating $\kappa(D)$. Generated programmatically from `src.visualization.syntactic_sentence_diagrams.render_syntactic_panel()`.

23 Appendix B: Notation Reference

This appendix collects all notation, symbols, and technical terminology used throughout the manuscript. Entries are grouped by domain and ordered alphabetically within each section. The **First Use** column indicates the section where each term is first introduced or defined.

Sections A–K: A Linguistic Terms; B Category Theory; C Enriched Categories and Magnitude; D Distributional Semantics and LLMs; E Distributional Active Inference (DAIF); F Active Inference and Cognitive Models; G Quantum and Topological Terms; H Logical and Type-Theoretic Terms; I AI and Communication Protocols; J Diagrammatic Reasoning; K Notation Conventions.

Figures that colour-code case roles (category graphs, string diagrams, appendix panels) take those colours from the shared map `CASE_COLORS` in `src/visualization/styles.py` (keys are `CaseRole` names). Captions state the mapping where it matters for reading the diagram.

23.1 Linguistic Terms

Table 15: Linguistic terminology and symbols.

Term	Symbol	Definition	First Use
Ablative	ABL	Case role encoding origin, source, or cause	§2
Accusative	ACC	Case role encoding patient, theme, or direct object	§2
Active–Stative alignment	—	Alignment in which the sole argument S splits by agentivity	§2
Agent-like argument	A	The agent-like argument of a transitive clause	§2
Alignment functor	$F : \mathcal{U} \to \mathcal{L}$	Structure-preserving map from universal to language-specific case category	§2
Alignment type	—	Systematic pattern governing how S, A, P are grouped for case marking	§2
Case category	$\mathcal{C}$	Small category whose objects are case roles and morphisms are grammatical relations	§2
Case frame	—	The set of case roles activated by a particular verb or predicate	§2
Case-typed noun space	$N_{\text{NOM}}, N_{\text{ACC}}, \ldots$	Case-specific vector subspace in a case-enriched DisCoCat model	§4
Categorial grammar	—	Grammar assigning each lexical item an algebraic type encoding combinatory potential	§3
Contraction	$a^l \cdot a \to 1$	Pregroup reduction eliminating an adjoint pair	§3
Dative	DAT	Case role encoding recipient, goal, or beneficiary	§2
Deep case	—	Fillmore's universal semantic primitive (e.g., Agentive, Objective)	§2
Ergative–Absolutive alignment	—	Alignment grouping S = P ≠ A	§2
Expansion	$1 \to a \cdot a^l$	Pregroup expansion introducing an adjoint pair	§3
Fluid-S	—	Alignment in which S marking varies by context or volition	§2
Fluid-S functor	F_θ	Context-dependent alignment functor parameterized by a volition feature θ	§2
Functors (Alignment)	$F_{\text{acc}}, F_{\text{erg}}$	Alignment functors $\mathcal{U} \to \mathcal{L}_{\text{acc}}$ resp. $\mathcal{U} \to \mathcal{L}_{\text{erg}}$	§2
Language-specific case category	$\mathcal{L}_{\text{acc}}, \mathcal{L}_{\text{erg}}$	Codomain categories for accusative vs. ergative alignment functors	§2
Genitive	GEN	Case role encoding possessor or source	§2
Formal semantics	—	Montague's programme: meaning assigned compositionally via truth-conditional functions	§4
Grammatical relation	—	Morphism in a case category relating two case roles	§2
Instrumental	INS	Case role encoding instrument or means	§2

Term	Symbol	Definition	First Use
Kāraka	—	Pāṇini's system of deep semantic roles in Sanskrit grammar	§2
Left adjoint	a^l	Left adjoint type satisfying $a^l \cdot a \leq 1$	§3
Locative	LOC	Case role encoding location or context	§2
Markedness	—	Asymmetry in the formal complexity of paradigmatic oppositions	§2
Monadic semantics	—	Song's extension using monads to model sublexical verb-root decomposition	§3
Nominative	NOM	Case role encoding agent, experiencer, or intransitive subject	§2
Nominative–Accusative alignment	—	Alignment grouping S = A ≠ P	§2
Passivization	—	Syntactic transformation promoting patient to subject position	§3
Patient-like argument	P	The patient-like argument of a transitive clause	§2
Pregroup grammar	—	Grammar where types have left and right adjoints forming a compact closed category	§3
Proto-Agent	—	Dowty's cluster of agentive entailments (volition, sentience, causation)	§2
Proto-Patient	—	Dowty's cluster of patient entailments (change of state, causal affectedness)	§2
Right adjoint	a^r	Right adjoint type satisfying $a \cdot a^r \leq 1$	§3
Sole argument	S	The sole argument of an intransitive clause	§2
Thematic role	—	Semantic relation between a predicate and its argument (Agent, Patient, Goal, etc.)	§2
Tripartite alignment	—	Alignment in which S ≠ A ≠ P (all three distinguished)	§2
Verb root decomposition	—	Analysis of a verb's internal causative and result-state layers via a monad	§3
Vocative	VOC	Case role encoding direct address	§2

23.2 Category Theory

Table 16: Category theory terminology and symbols.

Term	Symbol	Definition	First Use
Box	—	Node in a string diagram representing a morphism	§3
Cap	$\eta : 1 \to a^r \otimes a$	Unit of a compact closure (coevaluation map)	§3
Category	$\mathcal{C}$	Collection of objects and morphisms with identity and associative composition	§2
Classifying topos	$\mathcal{E}_{\mathbb{T}}$	Canonical topos: models of $\mathbb{T}$ in a Grothendieck topos $\mathcal{F}$ correspond to geometric morphisms $\mathcal{F} \to \mathcal{E}_{\mathbb{T}}$	§6
Codomain	$\mathrm{cod}(f)$	Target object of a morphism f	§2
Commutative diagram	—	Diagram in which all directed paths with the same start and end yield equal composites	§1
Cobordism	—	Manifold with boundary connecting two lower-dimensional manifolds; domain of TQFT functors	§8
Compact closed category	—	Monoidal category in which every object has a left and right dual	§3
Composition	$g \circ f$	Sequential application of morphisms: first f, then g	§2
Cup	$\varepsilon : a \otimes a^r \to 1$	Counit of a compact closure (evaluation map)	§3
Diagram	D	DisCoPy representation of a morphism in a monoidal category	§3
†-compact closed category	—	Compact closed category with a contravariant involutive endofunctor †; framework for ZX-calculus	§8
Domain	$\mathrm{dom}(f)$	Source object of a morphism f	§2
Fiber bundle	—	Projection $\pi : E \to B$ whose fibers carry role-filler structure; topos-theoretic LoT model	§6

Term	Symbol	Definition	First Use
Functor	$F : \mathcal{C} \to \mathcal{D}$	Structure-preserving map between categories	§2
Geometric theory	$\mathbb{T}$	Theory axiomatized by sequents with finite conjunctions, arbitrary disjunctions, existential quantification	§6
Geometric theories (case)	$\mathbb{T}_{\text{typ}}, \mathbb{T}_{\text{log}}, \mathbb{T}_{\text{dist}}, \mathbb{T}_{\text{enr}}$	TypeLogical, type-logical, distributional (DisCoCat), and enriched case theories (section 13)	§6
Bridge topos	$\mathcal{E}_{\mathbb{T}_{\text{bridge}}}$ (schematic)	Intermediate classifying topos linking two formalizations in the Morita bridge programme	§6
Identity morphism	id_A	Morphism from A to itself satisfying $f \circ \text{id} = f = \text{id} \circ f$	§2
Monad	—	Endofunctor with unit and multiplication satisfying associativity and unit laws	§3
Monoidal category	—	Category with a bifunctor $\otimes$ (tensor product) and unit object I	§3
Morita equivalence	$\mathcal{E}_{\mathbb{T}_1} \simeq \mathcal{E}_{\mathbb{T}_2}$	Equivalence of classifying toposes enabling inter-theoretic transfer	§6
Morphism	$f : A \to B$	Arrow between objects in a category; encodes a relation or transformation	§2
Natural transformation	$\alpha : F \Rightarrow G$	Family of morphisms $\alpha_A : F(A) \to G(A)$ commuting with all morphisms; alignment maps in semantics use $\nu_{\text{acc}}, \nu_{\text{erg}}$; compact-closure cap η_{cc} is distinct from IQN curvature η_{IQN}	§2
Object	$A, B, \ldots$	Entities in a category; in our framework, case roles	§2
Presheaf	$\hat{\mathcal{C}} = [\mathcal{C}^{op}, \mathbf{Set}]$	Contravariant functor from $\mathcal{C}$ to **Set**	§6
Snake equation	$(1 \otimes \varepsilon) \circ (\eta \otimes 1) = 1$	Zigzag identity: fundamental axiom of compact closed categories	§4
String diagram	—	Planar graph faithfully representing morphisms in a monoidal category	§3
Subobject classifier	Ω	Object in a topos playing the role of a truth-value object	§6
Swap	$\sigma_{A,B} : A \otimes B \to B \otimes A$	Braiding morphism permuting two objects in a symmetric monoidal category	§3
Tensor product	$A \otimes B$	Monoidal product representing parallel composition or concatenation	§3
Topos	$\mathcal{E}$	Category with products, exponentials, and a subobject classifier; generalized universe of sets	§6
Universal construction	—	Object or morphism characterized by a universal property (product, coproduct, limit)	§6
Wire	—	Edge in a string diagram representing a type (object)	§3

23.3 Enriched Categories and Magnitude

Table 17: Enriched categories and magnitude terminology.

Term	Symbol	Definition	First Use
Base-change functor	—	Conceptual tower $\mathbf{Bool} \hookrightarrow [0,1] \hookrightarrow \mathbf{R}_{\geq 0}$ showing progressively richer enrichments of grammatical categories (not computationally instantiated; this paper implements only $[0,1]$-enrichment)	§3
Categorical magnitude	$\lvert X \rvert$	Sum $\sum_i w_i$ where $(w_1, \ldots, w_n)$ solves $Z\mathbf{w} = \mathbf{1}$; measures effective number of objects	§5
Composition inequality	$\mathcal{V}(A,B) \cdot \mathcal{V}(B,C) \leq \mathcal{V}(A,C)$	Enriched analogue of composition on $([0,1], \cdot, 1)$: composite hom-value is at least the product of intermediates	§5

Term	Symbol	Definition	First Use
Enriched category	$\mathcal{V}$-Cat	Category whose hom-sets are replaced by objects of a monoidal category $\mathcal{V}$	§5
Enriched functor	—	Structure-preserving map between enriched categories respecting hom-values	§5
Hom-value	$\mathcal{V}(A,B) \in [0,1]$	Enriched analogue of hom-set; measures degree of relation between objects	§5
Identity axiom	$\mathcal{V}(A,A) = 1$	Every object has maximal self-relatedness in the enriched hom (equality, not mere inequality, in our $[0,1]$-convention)	§5
Similarity matrix	$Z_{ij} = \mathcal{V}(A_i, A_j)$	Matrix of all pairwise hom-values; used to compute categorical magnitude	§5
Lawvere metric space	—	Category enriched over $([0,\infty],+,0)$; generalizes metric spaces via enriched categories	§5
Magnitude homology	$\mathrm{MH}_n(\mathcal{C})$	Graded homological invariant categorifying magnitude; detects higher-dimensional holes in enriched categories	§5
Morphism weight (precision)	w_f, w_k, w_c	Enriched weights in $[0,1]$; subscript denotes morphism or role in VMP/DPE (section 16). **Convention**: w_f always refers to the *enriched* hom-value $\mathcal{C}(A,B)$ in the $[0,1]$-enriched category (§5), not the unit-weight morphism scalar carried by the `CaseCategory` in §2 — the two number systems are intentionally decoupled in the implementation (see §5 Architectural note).	§7
POVM element	E_c	Positive operator-valued measure element for case role c; $P(c \mid \rho) = \mathrm{Tr}(E_c \rho)$	§8
Prediction error	$\mathrm{PE}(f)$	$\propto w_f \cdot \lvert\mu_{\text{predicted}} - \mu_{\text{observed}}\rvert$; case violation signal scaling with morphism weight	§7
Shannon entropy	H	Information-theoretic measure characterized as the unique derivation of a topological operad (Bradley)	§5
Topological operad	—	Operad with topological structure whose derivations connect magnitude to entropy	§5
Weight vector	$\mathbf{w}$	Solution to $Z\mathbf{w} = \mathbf{1}$; entries are the effective weights of each object	§5

23.4 Distributional Semantics and LLMs

Table 18: Distributional semantics and LLM terminology.

Term	Symbol	Definition	First Use
Attention mechanism	—	Transformer component computing weighted relevance between token positions	§4
Attention weight	α_{ij}	Softmax-normalized score encoding contextual relevance of token j to token i	§4
BERT	—	Bidirectional Encoder Representations from Transformers; masked language model producing contextualized embeddings	§4
Contextualized embedding	$\mathbf{v}_w^{(c)}$	Vector representation of word w that varies with linguistic context c	§4
Compact closure map	$\varepsilon_N : N \otimes N \to \mathbb{R}$	Inner product implementing pregroup contraction in the vector space semantics	§4
Cosine similarity	$\cos(\mathbf{u},\mathbf{v})$	Similarity measure $\frac{\mathbf{u}\cdot\mathbf{v}}{\lVert\mathbf{u}\rVert\lVert\mathbf{v}\rVert}$ between vectors	§5
Distributional Memory	—	Baroni–Lenci tensor-based framework structuring co-occurrence as (word, relation, word) triples	§4
DisCoCat	—	Distributional Compositional Categorical model; monoidal functor from pregroup grammars to vector spaces	§3
DisCoCirc	—	Discourse-level extension of DisCoCat with persistent entity wires	§4
Distributional hypothesis	—	The thesis that words occurring in similar contexts have similar meanings (Harris 1954, Firth 1957)	§4
GloVe	—	Global Vectors for Word Representation; log-bilinear model of word co-occurrence statistics	§4

Term	Symbol	Definition	First Use
GPT	—	Generative Pre-trained Transformer; autoregressive language model	§4
Meaning functor	$F : \mathbf{Preg} \to \mathbf{FVect}$	Monoidal functor assigning vector spaces to types and linear maps to derivations	§4
Noun space	N	Vector space to which noun types n map under the meaning functor	§4
Parameterized optic	—	Categorical construction (Gavranović) modeling attention heads as functorial lenses	§9
Self-attention	$\text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V$	transformer operation computing contextualized representations	§4
Sentence space	S	Vector space to which sentence type s maps under the meaning functor	§4
Sentence vector	$\overrightarrow{\text{sentence}}$	Vector in S computed by tensoring and contracting word meanings via DisCoCat	§4
Static embedding	$\mathbf{v}_w$	Fixed vector representation of word w independent of context (e.g., Word2Vec, GloVe)	§4
Transformer	—	Neural architecture using self-attention and feed-forward layers for sequence processing	§4
Word2Vec	—	Neural embedding model (Mikolov et al. 2013) learning word vectors from local context windows	§4
Word vector	$\mathbf{v}_w \in \mathbb{R}^d$	d-dimensional real-valued vector encoding distributional properties of word w	§4

23.5 Distributional Active Inference (DAIF)

Table 19: Distributional Active Inference (DAIF) terminology.

Term	Symbol	Definition	First Use
Return distribution	$Z(s, a)$	Full probability distribution over discounted cumulative rewards from state s under action a	§7c
Distributional Bellman operator	$\mathcal{T}^\pi$	Distributional analogue of the Bellman operator: $\mathcal{T}^\pi Z \overset{d}{=} R + \gamma Z'$	§7c
Push-forward measure	$\mathbf{S}_\#\mathbb{P}$	Path-level push-forward of the trajectory measure (composed with decoder $f : \mathcal{S} \to \mathcal{X}$ in Equation 19); generic push-forward written $T_\#\mu$	§7c
Discount factor	$\gamma \in (0, 1)$	Exponential time-discount in the distributional Bellman return	§7c
Quantile level	$\tau \in [0, 1]$	Quantile index in QR-DQN and IQN; sampled uniformly at inference time	§7c
Quantile Huber loss	$\rho_\tau^\kappa(u)$	Loss function for quantile regression: interpolates L^1 and L^2 via threshold κ	§7c
Wasserstein distance	$W_p(P, Q)$	p-Wasserstein distance between return distributions; W_1 = area between CDFs	§7c
Bethe free energy	$F_{\text{Bethe}}[\mathbf{q}]$	Tractable approximation to variational free energy in belief propagation	§7c
Expected information gain	$\text{EIG}(o)$	$D_{\text{KL}}(q(\mathbf{c} \mid o) \| p(\mathbf{c}))$; epistemic value of observation o	§7c
Expected free energy	$G(\pi)$	Pragmatic + epistemic + risk cost of policy π; minimised for action selection	§7c
Distributional prediction error	DPE	Precision-weighted Wasserstein mismatch: $w_f \cdot W_1(Z_{\text{pred}}, Z_{\text{obs}})$	§7c
ERP amplitude profile	ERPProfile	Dataclass holding N400/P600 amplitudes, peak latencies, and time-series waveforms for each case role	§7c
Return entropy	$H[Z]$	Shannon entropy of the return distribution: $-\sum_i p_i \log p_i$	§7c
Quantile calibration error	CE	Mean absolute deviation between nominal quantile levels and empirical coverage	§7c

Term	Symbol	Definition	First Use
IQN risk distortion	$\psi_{\text{IQN}}(\tau)$	Maps quantile levels τ; piecewise formulas (neutral, power, tail, CVaR) match `implicit_quantile_network_update()` and the §7c table; distinct from β_{risk} in $G(\pi)$	§7c
IQN curvature	η_{IQN}	Default 0.71 in code; not the compact-closure cap η_{cc}	§7c
CVaR scale	α_{CVaR}	IQN mode uses $\psi_{\text{IQN}}(\tau) = \tau \cdot \alpha_{\text{CVaR}}$; default 0.25 in code (distinct from softmax α_{pol})	§7c
Risk sensitivity (EFE)	β_{risk}	Non-negative coefficient on risk(π) in $G(\pi)$; $\beta_{\text{risk}} = 0$ recovers standard EFE	§7c
Inverse temperature (policy)	α_{pol}	Softmax sharpness over policies in $P(\pi)$	§7c
Belief precision (VMP)	$\Lambda_{\text{post}}, \Lambda_{\text{prior}}, \Lambda_{\text{lik}}$	Posterior, prior, and likelihood precision matrices; $\Delta\Lambda$ is their positive update magnitude	§7c
Violation severity	$S_{\text{violation}}$	$\in \{0, 0.5, 1.0\}$ in ERP decomposition	§7c
TD error (quantile)	δ_{ij}	Temporal-difference residual in QR-DQN (Equation 22)	§7c
Quantile count (current)	N	Number of current-network quantile levels in QR-DQN (Eq. 22); equal to the length of `current_quantiles` passed to `quantile_td_update()`	§7c
Quantile count (target)	N'	Number of target-network quantile samples in QR-DQN (Eq. 22)	§7c
Huber threshold	κ	Cut-point between quadratic and linear regimes of the Huber loss $L_\kappa(u)$ (Eq. 23); default $\kappa = 1$ in `quantile_td_update()`	§7c
Atom count (C51)	N_{atoms}	Number of support atoms z_i in the C51 categorical representation (Eq. 20); default $N_{\text{atoms}} = 51$ in `categorical_return_distribution()`	§7c
Support bounds (C51)	$V_{\min}, V_{\max}$	Lower and upper endpoints of the atomic support spanned by $\{z_i\}$ in the C51 representation	§7c
Bin count (entropy)	N_{bins}	Number of equal-width bins used to discretise the quantile-parameterised return when computing $H[Z]$ (Eq. 37); default $N_{\text{bins}} = 50$ in `return_distribution_entropy()`	§7c

23.6 Active Inference and Cognitive Models

Table 20: Active inference and cognitive modeling terminology.

Term	Symbol	Definition	First Use
Active inference	—	Framework in which perception and action are unified as variational free energy minimization	§7
Active sampling	—	Agent's selection of actions to confirm or update case assignments via sensory evidence	§7
Belief updating	—	Bayesian posterior computation: revising generative model parameters given new observations	§7
CEREBRUM	—	Case-Enabled Reasoning Engine with Bayesian Representations for Unified Modeling; treats AI models as case-bearing entities	§7
Distributional Active Inference (DAIF)	—	Extension replacing scalar value summaries with full return distributions in active inference	§7
Distributional RL	—	Reinforcement learning operating on full return distributions rather than scalar expected values	§7
Free energy	F	Variational bound on surprisal; $F = D_{KL}[q(\theta) \parallel p(\theta \mid o)] - \ln p(o)$	§7
Free energy principle (FEP)	—	The principle that self-organizing systems maintain themselves by minimizing surprisal	§7
Garden-path reanalysis	—	Restructuring of case assignments when incoming evidence contradicts the current parse	§7

Term	Symbol	Definition	First Use
Generative model	$p(o, s)$	Joint probability model over observations o and hidden states s	§7
Markov blanket	—	Statistical boundary separating internal states from external environment; defines agent boundary	§7
N400	—	Event-related brain potential peaking ~400ms post-stimulus; indexes semantic prediction error	§7
P600	—	Event-related brain potential peaking ~600ms post-stimulus; indexes syntactic prediction error	§7
Perceptual inference	—	Updating internal beliefs to better predict current observations (reduce prediction error)	§7
Precision weighting	—	Weighting of prediction errors by inverse variance; enriched morphism weights serve this role	§7
Prediction error	ε	Difference between predicted and observed sensory input; drives belief updating	§7
ROSE model	—	Murphy's Representation–Operation–Structure–Encoding architecture; cross-frequency phase-amplitude coupling (PAC) linking biolinguistic syntax (slow oscillatory phase) to neuropragmatic inference (fast gamma); the neural-timescale bridge of Pillar 6	§1, §7c
S-HAI	—	Schema-Based Hierarchical Active Inference; dual-level POMDP connecting abstract relational schemas (Level 2: case diagram structure) to sensorimotor surface parsing (Level 1)	§7
Push-forward (general)	$T_{\#}\mu$	Image of a measure μ under a measurable map T; DAIF-specific notation in subsection 23.5	§7
Situation semantics	—	Framework representing meaning as relations between situations (Barwise and Perry 1983)	§7
Surprisal	$-\ln p(o)$	Negative log-probability of an observation under the generative model	§7
Variational free energy	$F[q]$	Functional upper bound on surprisal minimized by approximate posterior q	§7

23.7 Quantum and Topological Terms

Table 21: Quantum and topological terminology.

Term	Symbol	Definition	First Use
Amplituhedron	—	Positive geometry encoding scattering amplitudes; connected to TQNN execution traces	§8
Barren plateau	—	Vanishing gradient phenomenon in PQC training; gradients decay exponentially in system size for certain ansätze	§4b
†-compact category	—	See †-compact closed category in Category Theory	§8
CPTP map	—	Completely Positive Trace-Preserving map; quantum channel between Hilbert spaces	§8
Density operator	ρ	Positive semidefinite trace-one operator encoding a quantum (or semantic) state	§8
Execution trace	—	Record of operations in a quantum computation; connects TQNNs to amplituhedra	§8
Generalized flow	—	Graph-theoretic property ensuring deterministic circuit extraction from ZX-diagrams	§8
Hadamard box	H	ZX-calculus element implementing the Hadamard gate; converts between Z and X bases	§8
Holographic screen	—	Information boundary between interacting quantum systems carrying a qubit array	§8
Parameterized quantum circuit (PQC)	—	Quantum circuit with trainable angle parameters; the computational substrate of QNLP and case-category implementations	§4b

Term	Symbol	Definition	First Use
IQP ansatz	—	Instantaneous Quantum Polynomial-time circuit: default lambeq PQC ansatz for noun and verb boxes	§4b
Sim4 ansatz	—	Strongly entangling layer PQC ansatz used for discourse-level lambeq Gen II circuits	§4c
Pointer state	—	Preferred quantum state selected by a QRF; determines the measurement basis	§8
Quantum contextuality	—	Quantum correlations that reduce cohomological obstructions to semantic alignment	§8
Quantum discord	—	Quantum correlation measure equal to integrated semantic information in sheaf framework	§8
Quantum key distribution (QKD)	—	Protocol providing information-theoretic security for quantum communication channels	§9
Quantum reference frame (QRF)	—	Observer-relative frame selecting pointer states and inducing decoherence	§8
Semantic Hilbert space	H_v	The finite-dimensional semantic Hilbert space carried at vertex v in a quantum semantic sheaf	§8b
Quantum semantic sheaf	(H, F, ρ)	Triple of Hilbert spaces, CPTP channels, and density operators over a communication graph	§8
Reshetikhin–Turaev invariant	—	Topological invariant assigning to a ribbon graph a linear map via TQFT	§8
Sheaf cohomology	$H^n(\mathcal{F})$	Cohomological obstruction classes governing alignment in a quantum semantic sheaf	§8
Spin-network	—	Graph with edges labeled by representations and vertices by intertwiners; TQFT data	§8
Spider	—	Elementary ZX-diagram node (Z-spider or X-spider) representing a quantum operation	§8
TQFT	—	Topological Quantum Field Theory; functor from cobordisms to Hilbert spaces	§8
TQNN	—	Topological Quantum Neural Network; QNN reformulated via spin-networks and TQFT	§8
Turaev–Viro invariant	—	State-sum TQFT invariant; implements quantum error-correcting codes in TQNNs	§8
ZX-calculus	—	Graphical language for quantum circuits using Z-spiders, X-spiders, and Hadamard boxes	§8
ZX-diagram	—	String diagram in a †-compact closed category representing a quantum process	§8
ZX rewrite rule	—	Graph-theoretic transformation preserving the semantics (linear map) of a ZX-diagram	§8

23.8 Logical and Type-Theoretic Terms

Table 22: Logical and type-theoretic terminology.

Term	Symbol	Definition	First Use
β-reduction	—	Computational reduction step in λ-calculus; corresponds to cut elimination in proofs	§3
Church–Rosser property	—	Confluence of β-reduction: all reduction sequences converge to the same normal form	§3
Curry–Howard isomorphism	—	Correspondence between proofs and programs, propositions and types	§3
Cut elimination	—	Proof normalization procedure removing intermediate lemmas; corresponds to β-reduction	§3
Graded type theory	—	Extension of type theory tracking effects (e.g., evidentiality) via graded modalities	§3
Lambek calculus	—	Non-commutative intuitionistic linear logic for syntactic type assignment	§3

Term	Symbol	Definition	First Use
Left residual	$A\backslash B$	Type of an expression that, given A to the left, produces B	§3
Residuation law	$A \otimes B \leq C \iff A \leq C/B \iff B \leq A\backslash C$	Fundamental axiom connecting the three connectives of the Lambek calculus	§3
Right residual	B/A	Type of an expression that, given A to the right, produces B	§3

23.9 AI and Communication Protocols

Table 23: AI and communication protocol terminology.

Term	Symbol	Definition	First Use
A2A Protocol	—	Google's Agent-to-Agent protocol for cross-framework agent communication via HTTP/JSON-RPC	§9
ACP	—	Agent Communication Protocol; standardizes messaging formats across agents, apps, and humans	§9
ANP	—	Agent Network Protocol; three-layer architecture for trusted distributed agent interaction	§9
Categorical deep learning	—	Deep learning approached through the lens of category theory (Gavranović et al.)	§9
Double Categorical Systems Theory (DCST)	—	Framework using 2-categories (horizontal + vertical composition) for explainable autonomous AI	§9
Functorial encryption	—	Semantic cryptography: applying a secret functor to map plaintext categories into ciphertext categories	§9
lambeq	—	Quantum Natural Language Processing pipeline compiling DisCoCat diagrams to quantum circuits	§9
MCP	—	Model Context Protocol; standardizes how AI agents access external tools and data sources	§9
Parameterized optics / lenses	—	Categorical constructions modeling neural network components (Gavranović); attention heads as optics	§9
QNLP	—	Quantum Natural Language Processing; quantum computation on DisCoCat sentence diagrams	§9
Role variables	$X_{\text{NOM}}, X_{\text{ACC}}, \ldots$	Agentic components structured by case roles in networked LLM contexts (e.g., active requester policy)	§9
Semantic cryptography	—	Encrypting compositional meaning structures (functorial encryption, diagram obfuscation, weight masking)	§9
Protocol category	$\mathcal{C}_{\text{protocol}}$	Category whose morphisms are licensed interaction steps in the case-theoretic firewall (section 20)	§9b
Adversarial morphism	ϕ	Illicit map attempting case-role promotion (e.g. ACC→NOM) in injection attacks	§9b
Access Collapse	—	Catastrophic boundary failure where adversarial text traverses from passive data (ACC) to active instruction (NOM) without structural constraint	§9b
Case-theoretic firewall	—	Type-checking system enforcing licensed morphism constraints at agent communication boundaries; detects illicit role promotions in $\text{Mor}(\mathcal{C}_{\text{protocol}})$	§9b
Prompt injection	—	Attack illicitly promoting user-supplied data from ACC (patient) to NOM (commanding agent); structurally a type violation in the interaction grammar	§9b
Symbolic Isolation	—	Property ensuring ACC-typed data wires cannot fuse with NOM-typed command wires; enforced by the non-cartesian fragment of the monoidal structure	§9b

23.10 Diagrammatic Reasoning

Table 24: Diagrammatic reasoning terminology.

Term	Symbol	Definition	First Use
Categorical complexity	$\kappa(D)$	Complexity metric for diagram D derived from box counts, cup counts, and swap depths	§4b
Cognitively privileged representation	—	Representation format that leverages perceptual and spatial cognition for inference	§1
Diagram depth	—	Length of the longest input-to-output path through boxes; measures derivational complexity	§4
Existential graphs	—	Peirce's graphical logic system conducting first-order logic entirely diagrammatically	§7
Free ride	—	Shimojima's term: information extracted from a diagram without explicit inference steps	§1
Hybrid reasoning	—	Giardino's term: reasoning combining perceptual pattern recognition with theoretical knowledge	§1
Inferential instrument	—	Manders's term: a diagram whose spatial properties encode proof-relevant information	§6
Joyal–Street theorem	—	Soundness and completeness of string-diagrammatic reasoning for monoidal categories	§3
Normal form	D_{nf}	Canonical form of a diagram obtained by rewriting; unique up to the axioms	§4
Role colours (figures)	`CASE_COLORS`	`CaseRole` → display colour for generated figures; single source in `src/visualization/styles.py`	§2–§4, Appendix A

23.11 Notation Conventions

Table 25: General mathematical notation and manuscript conventions.

Convention	Meaning
$\mathcal{C}, \mathcal{D}$	Categories
$\mathcal{V}$	Enrichment base (monoidal category, typically $([0,1], \cdot, 1)$)
$\mathcal{U}$	Universal (maximal) case category
$\mathcal{L}$	Language-specific case category
$\mathcal{E}_{\mathbb{T}}$	Classifying topos of theory $\mathbb{T}$
f, g, h	Morphisms
F, G	Functors
α, β	Natural transformations; functorial vertical composition (componentwise along objects) is standard (§2). In DAIF, α_{pol} is policy temperature; α_{CVaR} is CVaR scale; β_{risk} is EFE risk weight; see §7c
n, s	Basic pregroup types: noun, sentence
n^l, n^r	Left and right adjoints of type n
$n_{\text{NOM}}, n_{\text{ACC}}$	Case-subscripted noun types (e.g., nominative noun, accusative noun)
N, S	Noun space and sentence space under the meaning functor
$N_{\text{NOM}}, N_{\text{ACC}}, \ldots$	Case-specific vector subspaces in case-enriched DisCoCat
$\overrightarrow{\text{word}}$	Word vector (column vector in noun space N)
$\overleftrightarrow{\text{verb}}$	Verb tensor (element of $N \otimes S \otimes N$ for transitive verbs)
Preg	Category of pregroup types and reductions
FVect / **FdVect**	Category of finite-dimensional vector spaces and linear maps
FHilb	Category of finite-dimensional Hilbert spaces and linear maps
Qubit	Category of qubit systems with tensor product structure
Set	Category of sets and functions
Ent, **Case**	Categories for Entities and Case Roles respectively in DisCoCirc discourse extensions

Convention	Meaning
$\otimes$	Tensor product (monoidal product, type concatenation)
$\circ$	Composition of morphisms
$\Rightarrow$	Natural transformation between functors
$\simeq$	Categorical equivalence
$\leq$	Preorder relation on types (derivability)
γ	Discount factor in distributional RL return computation
$w \in [0,1]$	Enriched morphism weight (proto-role satisfaction degree)
ε	Sensory prediction error (active inference); ε_n also denotes the compact-closure **counit** (cup) on type n
η_{cc}	Compact closure **unit** (cap); IQN curvature uses η_{IQN} instead
ϵ	Small numerical floor (e.g. KL stabiliser, VMP convergence threshold); not the counit
ρ	Density operator (quantum state)
[@*key*]	Parenthetical citation
`[-@key]`	Suppress-author citation
`\autoref` (with `{sec:…}` / `{eq:…}` / `{fig:…}`)	LaTeX/Markdown automatic cross-reference to a labeled target
$Z(s,a)$	Return distribution (DAIF); full distributional representation of returns
$G(\pi)$	Expected free energy of policy (DAIF active inference)
τ	Quantile level in QR-DQN / IQN
$\psi_{\text{IQN}}(\tau)$	IQN risk distortion of quantile levels
β_{risk}	Risk-sensitivity coefficient in $G(\pi)$
α_{pol} ; α_{CVaR}	Policy softmax temperature; CVaR tail level (IQN)
w_f, w_k, w_c	Enriched morphism / role weights (precision on PE and VMP)
DPE	Distributional prediction error; precision-weighted Wasserstein mismatch
$H[Z]$	Return distribution entropy

24 Appendix C: Automated Test Suite Inventory

This appendix summarizes the **categories** of tests behind the counts in section 15. Aggregate figures are injected at build time (**1197** tests, **64** files, **9** domain packages under `src/`; 95.96% line-and-branch coverage on `src/` (from `coverage.json`); see `src/generate_manuscript_metrics.py` and `output/metrics.json`). API summary: `docs/api_reference.md`. The open-source package [docxology, 2026] holds the implementation exercised by these tests. Every test uses real mathematical computations—no mocks or fakes.

- **Categorical axiom tests**: Identity morphism existence, composition associativity, weight invariants, `is_well_formed()` full axiom check
- **Enriched category tests**: Hom-value constraints, composition inequality, categorical magnitude, magnitude deficit, full composition check, role clustering
- **Diagram type tests**: Pregroup diagrams validated for `dom == Ty()` and `cod == s`, correct box counts, diagram equality
- **Metrics tests**: Normal form preservation, depth computation with graceful fallback for pregroup diagrams
- **Natural transformation tests**: Component morphism construction, `naturality_holds` / `verify_naturality` on identity and incomplete maps, identity transformation generation, vertical composition of transformations, completeness checking over domain objects
- **Complexity metrics tests**: DisCoPy box/cup/cap counting on transitive/ditransitive diagrams, normal form computation and snake equation verification, syntactic complexity scoring with configurable weights, cross-diagram comparison utilities
- **Topos theory tests**: Geometric theory construction from standard and minimal case categories, classifying topos invariant computation, Morita equivalence verification (positive and negative cases), bridge transfer between equivalent theories with transfer-blocking for non-equivalent theories, enriched theory construction
- **Fluid-S tests**: Volitional/non-volitional mapping, probability splits, Bats language examples, kernel computation, enriched weight scaling
- **Active inference tests** (`tests/test_cognitive_*.py`): Belief construction and entropy, KL divergence (Gibbs' inequality, asymmetry), variational free energy, Bayesian belief update with zero-likelihood edge case, sequential multi-word belief update (five-step generative loop with entropy convergence), prediction error scaling including P600 ERP prediction with boundary weights, expected free energy decomposition (epistemic vs pragmatic), magnitude-based garden-path reanalysis cost with symmetry, N400 semantic violation proxy (including `test_cognitive_integration.py`)
- **Quantum case tests**: Crisp POVM orthogonal projectors, graded proto-role POVM, Fluid-S basis rotation, density matrix creation, Equation 38 (in section 18) $P(c \mid \rho) = \mathrm{Tr}(E_c\,\rho)$ verification
- **Cognitive security tests**: Type-violation detection, case frame validation, injection score computation, magnitude-based topological robustness, composition inequality as security boundary
- **Ditransitive tests**: Three-argument sentence creation, NOM/ACC/DAT case assignment, DisCoPy diagram with three cups, complexity comparison with transitive
- **Visualization tests** (`tests/test_visualization_*.py`): Category graphs, enriched heatmaps, functor panels, string and DisCoPy diagrams, complexity and DAIF plots, quantum and security plots, Fluid-S landscapes, syntactic panels—PNG output and structural checks where applicable
- **DAIF subpackage tests** (224 tests across 8 test files):
  - `test_daif_core.py`: Distributional Bellman operator, push-forward return, C51 categorical projection
  - `test_daif_quantile.py`: QR-DQN quantile Huber loss, IQN risk distortion (neutral/optimistic/pessimistic/CVaR), Wasserstein distances W_1/W_2
  - `test_daif_inference.py`: `distributional_case_assignment()` posterior convergence, variational message passing, Bethe free energy, expected information gain
  - `test_daif_prediction.py`: DPE precision-weighting, N400/P600 amplitude from return distributions, full `ERPProfile` waveform generation and peak latency
  - `test_daif_policy.py`: `G_policy()` EFE + risk term, Boltzmann policy temperature scaling, distributional epistemic value
  - `test_daif_metrics.py`: Convergence diagnostics (monotonicity, relative reduction), distributional KL divergence, quantile calibration error, return entropy
  - `test_daif_types.py`: `DistributionalReturn` helpers, `DAIFResult` / `ERPProfile` properties (with integration coverage of re-exported DAIF entrypoints)
- **Cross-module and structural coverage tests** (`test_cross_module_coverage.py`): Integration paths across enriched category and case_category modules not reached by per-module unit tests; verifies composition chaining, morphism weight transitivity (Equation 3), and magnitude consistency across module boundaries
- **Property-based and parametric tests** (`test_property_based.py`): Algebraic invariants exercised over parametric

inputs—enriched composition inequality, magnitude positivity, and morphism weight bounds in $[0, 1]$—confirming structural properties hold generically rather than only for specific examples used in unit tests

- **Visualization multi-module tests** (`test_visualization_plot_modules.py`, `test_visualization_syntactic_coverage.py`): Multi-module rendering pipelines and syntactic panel coverage paths not hit by per-module visualization tests; verifies that all registered plot functions execute to file without exception
- **Script and metrics tests** (`test_diagrams_generator.py`, `test_generate_manuscript_metrics.py`): Round-trip invocation of `scripts/generate_diagrams.py` domain dispatcher and `src/generate_manuscript_metrics.py`; verifies domain registry completeness, figure-path collection, and manuscript variable extraction against the live test suite and installed package versions

References

A2A Project. Agent2agent protocol (A2A). https://github.com/a2aproject/A2A, 2025. Open protocol for agent-to-agent communication.

Abdullah Akgül, Gulcin Baykal, Manuel Haußmann, Mustafa Mert Çelikok, and Melih Kandemir. Distributional active inference, 2026. Integrates active inference into distributional RL via push-forward measures on representation paths.

ANP Project. Agent network protocol (ANP). https://github.com/agent-network-protocol/AgentNetworkProtocol, 2025. Three-layer architecture for trusted agent interaction in distributed systems.

Anthropic. Model context protocol (MCP). https://modelcontextprotocol.io/, 2024. Standard for connecting AI systems with external tools and data sources.

Ash Asudeh and Gianluca Giorgolo. A monadic semantics for evidentials. *Proceedings of SALT*, 30, 2020. doi: 10.3765/salt.v30i0.4843.

Anonymous Authors. Adversarial reinforcement learning for large language model agent safety. *arXiv preprint arXiv:2510.05442*, 2025. URL https://arxiv.org/abs/2510.05442. Introduces the ARLAS framework, co-evolving attacker and defender LLM agents via RL to demonstrate indirect prompt injection as role manipulation.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010. doi: 10.1162/coli_a_00016. Unified tensor-based framework for distributional semantic phenomena.

Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, 1983.

BeeAI. Agent communication protocol (ACP). https://agentcommunicationprotocol.dev/, 2025. Standardized messaging format for agent, application, and user communication.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 449–458, 2017. URL https://arxiv.org/abs/1707.06887. Foundational paper on distributional RL: modeling full return distributions rather than expected returns.

Barry J. Blake. Grammatical relations. In *Theories of Case*. Cambridge University Press, 2001. doi: 10.1017/CBO9781139164788.

Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. String diagram rewrite theory I: Rewriting with Frobenius structure. *Journal of the ACM*, 69(2):1–58, 2022. doi: 10.1145/3502719. String diagram rewriting modulo Frobenius as double-pushout hypergraph rewriting; foundational for diagram simplification.

Ina Bornkessel and Matthias Schlesewsky. The extended argument dependency model: A neurocognitive approach to sentence comprehension across languages. *Psychological Review*, 113(4):787–821, 2006. doi: 10.1037/0033-295X.113.4.787. Cross-linguistic neurocognitive model of argument-structure processing; ERP predictions for case-violation paradigms in morphologically rich languages including Russian.

Tai-Danae Bradley. Entropy as a topological operad derivation. *Entropy*, 23(9):1195, 2021. doi: 10.3390/e23091195. URL https://arxiv.org/abs/2107.09581. Published in Entropy 2021; associated with CUNY Graduate Center work.

Tai-Danae Bradley. An enriched category theory of language. IPAM Talk, 2024. URL https://www.youtube.com/watch?v=DAxmrEtIOR8.

Tai-Danae Bradley. An enriched category theory of language. Tea Talk, 2025. URL https://www.youtube.com/watch?v=KCtyiE6Ybnc.

Tai-Danae Bradley and Juan Pablo Vigneaux. The magnitude of categories of texts enriched by language models. *Theory and Applications of Categories*, 44(37):1256–1281, 2025. URL https://arxiv.org/abs/2501.06662. Builds text categories from LM next-token probabilities; magnitude and magnitude homology of an associated metric space.

Tai-Danae Bradley, John Terilla, and Tyler Weyhrich. An enriched category theory of language: from syntax to semantics. *arXiv preprint arXiv:2106.07890*, 2021. URL https://arxiv.org/abs/2106.07890. Advances an enriched category theory of language, integrating case roles via compact closed categories to handle relational independence.

Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78(1):351–382, 2016. doi: 10.1007/s10623-015-0157-4. Quantum cryptographic primitives beyond QKD including secure computation and authentication.

Matteo Capucci, Bruno Gavranović, Jules Hedges, and Eigil Fjeldgren Rischel. Towards foundations of categorical cybernetics. *Electronic Proceedings in Theoretical Computer Science*, 333:235–248, 2021. doi: 10.4204/EPTCS.333.16. Foundational introduction to parametrised optics and categorical cybernetics for modelling controlled processes.

Olivia Caramello. The theory of topos-theoretic 'bridges': a conceptual introduction. *Glass Bead Journal*, 2016. URL https://www.glass-bead.org/article/the-theory-of-topos-theoretic-bridges-a-conceptual-introduction/.

Olivia Caramello. The theory of topos-theoretic bridges, five years later. Slides, 2021. URL https://www.oliviacaramello.com/Talks/TheoryToposTheoreticBridgesFiveYearsLaterCaramello.pdf.

Olivia Caramello. Syntactic learning via topos theory. Slides, 2023. URL https://www.oliviacaramello.com/Talks/SyntacticLearningViaToposTheory.pdf.

Simon A. Claassen. Explanations in the typology of grammatical relations and alignment systems. *RU:TS Student Linguistics Journal*, 2019. URL https://ruts-journal.ruhosting.nl/wp-content/uploads/2025/01/S.A.-Claassen-Typology-of-gramatical-relations-Explanations-in-the-typology-of-grammatical-relations-and-alignment-systems-1.pdf.

Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. doi: 10.1017/9781316219317. Definitive reference for categorical quantum mechanics and ZX-calculus.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384, 2010. Establishes DisCoCat, a functorial framework mapping grammatical types (via pregroups or monoidal categories) to vector spaces for distributional semantics, enabling compositional meaning computation that formalizes case-like structures through tensor contractions.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 2892–2901, 2018. doi: 10.1609/aaai.v32i1.11791.

Giovanni de Felice and Bob Coecke. Discourse in categorical compositional relational semantics. *arXiv preprint arXiv:2010.12559*, 2020. URL https://arxiv.org/abs/2010.12559. Demonstrates empirical DisCoCat parsing of case-marked sentences, showing how categorical composition captures grammatical relations distributionally, linking Fillmorean roles to vector-based inference.

Giovanni de Felice, Konstantinos Meichanetzidis, and Bob Coecke. DisCoCirc: Disambiguating compositionality in circuits. *arXiv preprint arXiv:2205.14815*, 2022. URL https://arxiv.org/abs/2205.14815.

Arthur de Huybrecht. Subcategorizing DisCoCat: Modelling light verb constructions with categorical compositional distributional semantics. Master's thesis, University of Oxford, 2024. Extends DisCoCat with subcategorization frames for light verb constructions.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. URL https://arxiv.org/abs/1810.04805. Bidirectional contextualized embeddings via masked language modeling.

docxology. Cognitive case diagrams: implementation and test suite. https://github.com/docxology/cognitive_case_diagrams, 2026. Open-source Python package accompanying the manuscript; accessed 2026-04-04.

Francesco Donnarumma, Mirco Frosolone, and Giovanni Pezzulo. Integrating large language models and active inference to understand eye movements in reading and dyslexia, 2023. URL https://arxiv.org/abs/2308.04941. Hierarchical active inference with BERT for modelling reading, saccades, and dyslexia via maladaptive priors.

David R. Dowty. Thematic proto-roles and argument selection. *Language*, 67(3):547–619, 1991. doi: 10.2307/415037. Proposes proto-roles (Agent-like, Patient-like) as clusters of entailments, contributing to DisCoCat by decomposing case systems into decomposable features modelable as linear transformations in distributional semantics.

Ariane Duneau. A pipeline for discourse circuits from CCG. MSc Dissertation, University of Oxford, 2021. Constructs DisCoCirc discourse circuits from CCG parse trees via iterative coreference resolution and wire merging.

Tiffany Duneau. Towards a comparative framework for compositional AI models, 2025. Extends DisCoCirc pipeline work to a comparative framework for evaluating compositional AI model architectures.

Chris Fields and James F. Glazebrook. Reference frame induced symmetry breaking on holographic screens. *Symmetry*, 13(3):408, 2021. doi: 10.3390/sym13030408. QRFs induce decoherence and symmetry breaking on holographic screens.

Chris Fields, James F. Glazebrook, and Antonino Marcianò. Sequential measurements, TQFTs, and TQNNs. *Fortschritte der Physik – Progress of Physics*, 70(11):2200104, 2022. doi: 10.1002/prop.202200104. arXiv:2205.13184. Maps QNNs onto spin-networks analyzable via TQFT.

Chris Fields, James F. Glazebrook, and Antonino Marcianò. Tensor networks as control systems for topological quantum neural networks. In *Proceedings of the IEEE International Conference on Quantum Computing and Engineering*, 2023. doi: 10.1109/QCE57702.2023.10113744. Tensor network control within the TQNN framework.

Chris Fields, James F. Glazebrook, and Antonino Marcianò. Universal quantum computation in topological quantum neural networks and amplituhedron representation, 2025. TQNNs as universal quantum computers via Reshetikhin–Turaev/Turaev–Viro invariants and amplituhedra.

Charles J. Fillmore. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart & Winston, 1968. Introduces case grammar, positing deep semantic roles (e.g., Agent, Patient) as universal primitives structuring predicate-argument relations, providing a foundational bridge to categorical formalizations where roles correspond to functorial mappings in DisCoCat models.

John Rupert Firth. *Papers in Linguistics 1934–1951*. Oxford University Press, 1957. Origin of the distributional hypothesis: "You shall know a word by the company it keeps".

Daniel Ari Friedman and Active Inference Institute. CEREBRUM: Case-enabled reasoning engine with Bayesian representations for unified modeling. https://github.com/ActiveInferenceInstitute/CEREBRUM, 2024. Open-source implementation of case-bearing active inference models. MIT License.

Karl J. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138, 2010. doi: 10.1038/nrn2787. Foundational statement of the free energy principle for biological self-organization.

Karl J. Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural Computation*, 29(1):1–49, 2017. doi: 10.1162/NECO_a_00912.

Karl J. Friston, Thomas Parr, Yan Yufik, Noor Sajid, Catherine J. Price, and Emily Holmes. Generative models, linguistic communication and active inference. *Neuroscience & Biobehavioral Reviews*, 118:42–64, 2020. doi: 10.1016/j.neubiorev.2020.07.005. Active inference framework for language production and comprehension via hierarchical generative models.

Karl J. Friston, Lancelot Da Costa, Noor Sajid, Conor Heins, Kai Ueltzhöffer, Grigorios A. Pavliotis, and Thomas Parr. World model learning and inference. *Neural Networks*, 144:573–590, 2021. doi: 10.1016/j.neunet.2021.09.011. World models and understanding through active inference; cited in cognitive integration chapter.

Bruno Gavranović. *Fundamental Components of Deep Learning: A Category-Theoretic Approach*. PhD thesis, University of Strathclyde, 2024. URL https://arxiv.org/abs/2403.13001.

Bruno Gavranović, Paul Lessard, Andrew Dudzik, Tamara von Glehn, João G. M. Araújo, and Petar Veličković. Position: Categorical deep learning is an algebraic theory of all architectures, 2024.

Valeria Giardino. Diagrammatic reasoning in mathematics. *Springer Handbook of Model-Based Science*, pages 499–522, 2017. doi: 10.1007/978-3-319-30526-4_22.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Concrete models and empirical evaluations for the categorical compositional distributional model of meaning. *Computational Linguistics*, 41(1):71–118, 2015. doi: 10.1162/COLI_a_00209.

A. Gutierrez Cisneros et al. Neuropragmatics: From classical pragmatics to neurocognitive frameworks for speech-act evaluation. *Frontiers in Psychology*, 2026. doi: 10.3389/fpsyg.2026.1745034.

Zellig S. Harris. Distributional structure. *Word*, 10(2–3):146–162, 1954. doi: 10.1080/00437956.1954.11659520. Foundational formalization of distributional analysis in structural linguistics.

Martin Haspelmath. Grammatical categories and relations: Universality vs. language-specificity. *Language and Linguistics Compass*, 3(1):1–25, 2009. doi: 10.1111/j.1749-818X.2008.00111.x.

Louis Hjelmslev. *La catégorie des cas: étude de grammaire générale*. Universitetsforlaget i Aarhus, 1935. Foundational glossematic analysis treating case as a purely relational category.

Roman Jakobson. Morphological inquiry into slavic declension: Structure of russian case forms. *American Contributions to the Fourth International Congress of Slavicists*, 1958. Analyzes Russian nominal (declensional) morphology through case oppositions (e.g., subject vs. object marking), laying groundwork for categorical views of grammatical functions as morphisms preserving distributional contexts in semantic spaces.

Girish Nath Jha. Sanskrit and computational linguistics. In *The Oxford Handbook of History of Linguistics*. Oxford University Press, 2021. doi: 10.1093/oxfordhb/9780199585847.013.0034. Survey of Sanskrit grammatical tradition and computational approaches.

André Joyal and Ross Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88(1):55–112, 1991. doi: 10.1016/0001-8708(91)90003-P.

Subhash Kak. Pāṇinian grammar and computer science. *Annals of the Bhandarkar Oriental Research Institute*, 68:79–90, 1987. Connections between Pāṇini's generative grammar and formal language theory.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. Functorial question answering. *arXiv preprint arXiv:2111.13154*, 2021. URL https://arxiv.org/abs/2111.13154.

Aleks Kissinger and John van de Wetering. Reducing T-count with the ZX-calculus. *Quantum*, 4:279, 2020. doi: 10.22331/q-2020-06-04-279. Graph-theoretic ZX simplification with generalised flow for deterministic circuit extraction.

Colin Krawchuk, Nikhil Khatri, Neil John Ortega, and Dimitri Kartsaklis. Efficient generation of parameterised quantum circuits from large texts, 2025a. URL https://arxiv.org/abs/2505.13208. Uses DisCoCirc string diagrams to generate discourse-level quantum circuits via tree-like pregroup diagrams; part of lambeq Gen II.

Colin Krawchuk, Nikhil Khatri, Neil John Ortega, and Dimitri Kartsaklis. lambeq Gen II: Scalable compositional models for discourse-level NLP. Quantinuum blog post, 2025b. URL https://www.quantinuum.com/blog/lambeq-gen-ii-a-quantum-enhanced-interpretable-and-scalable-text-based-nlp-software-package. Product announcement for lambeq Gen II with full DisCoCirc support; technical paper: Krawchuk et al., arXiv:2505.13208.

Gina R. Kuperberg and T. Florian Jaeger. What do we mean by prediction in language comprehension? *Language, Cognition and Neuroscience*, 31(1):32–59, 2016. doi: 10.1080/23273798.2015.1102299.

Marta Kutas and Kara D. Federmeier. Thirty years and counting: Finding meaning in the N400 component of the event-related brain potential (ERP). *Annual Review of Psychology*, 62:621–647, 2011. doi: 10.1146/annurev.psych.093008.131123.

Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65(3):154–170, 1958. doi: 10.1080/00029890.1958.11989160. Pioneers Lambek calculus as a categorial grammar category, enabling case formalization through type reduction, directly underpinning DisCoCat's functorial semantics for grammatical roles.

Joachim Lambek. The categorial fine-structure of natural language. In *Proofs and Programs*. CWI and University of Amsterdam, 2004. URL https://archive.illc.uva.nl/lgc/translation/papers/Lambek.NEW.pdf.

Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987. doi: 10.1111/j.1551-6708.1987.tb00863.x.

Tom Leinster and Michael Shulman. Magnitude homology of enriched categories and metric spaces. *Algebraic & Geometric Topology*, 21(5):2175–2221, 2021. doi: 10.2140/agt.2021.21.2175. Categorifies magnitude to a graded homological invariant detecting higher-dimensional structure in enriched categories.

Alessandro Lenci. Distributional models of word meaning. *Annual Review of Linguistics*, 4:151–171, 2018. doi: 10.1146/annurev-linguistics-030514-125254. Authoritative survey linking distributional semantics to cognitive and computational models.

Alistair Letcher, Stefan Woerner, and Christa Zoufal. Tight and efficient gradient bounds for parameterized quantum circuits. *Quantum*, 8:1484, 2024. doi: 10.22331/q-2024-09-25-1484. Assumption-free gradient lower bounds; no barren plateau for local observables.

Jiaxuan Li and Richard Futrell. A decomposition of surprisal tracks the N400 and P600 brain potentials. In *Proceedings of the Society for Computation in Linguistics (SCiL 2023)*, 2023. URL https://sites.socsci.uci.edu/~rfutrell/papers/li2023decomposition.pdf. Decomposes surprisal into heuristic surprise (N400) and discrepancy signal (P600), empirically validated on ERP data.

Jiaxuan Li and Richard Futrell. Decomposition of surprisal: Unified computational model of ERP components in language processing, 2024. URL https://arxiv.org/abs/2409.06803. Generalizes 2023 SCiL result: shallow surprisal tracks N400, deep surprisal tracks P600.

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. lambeq: An efficient high-level python library for quantum NLP. *arXiv preprint arXiv:2110.04236*, 2021. URL https://arxiv.org/abs/2110.04236.

V. Maele et al. Schema-based active inference supports rapid generalization of experience and frontal cortical signatures of schemas, 2026. Schema-based hierarchical active inference (S-HAI) bridges predictive processing with schemas as top-level abstract relations, yielding rapid generalization via grounding likelihoods in dual-POMDPs.

Kenneth Manders. The Euclidean diagram (1995). *The Philosophy of Mathematical Practice*, pages 80–133, 2008.

Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni de Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. Grammar-aware question-answering on quantum computers. *arXiv preprint arXiv:2005.04147*, 2020. URL https://arxiv.org/abs/2005.04147.

Igor A. Mel'čuk. Meaning-text models: A recent trend in soviet linguistics. *Annual Review of Anthropology*, 10:27–62, 1981. doi: 10.1146/annurev.an.10.100181.000331.

Igor A. Mel'čuk. *Dependency syntax: Theory and practice*. SUNY Press, Albany, NY, 1988. ISBN 9780887064500.

François Meyer and Martha Lewis. Modelling lexical ambiguity with density matrices. *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL)*, pages 276–290, 2020. doi: 10.18653/v1/2020.conll-1.21. Introduces density matrices (mixed quantum states) within DisCoCat to structurally model lexical ambiguity and entailment.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR) Workshop*, 2013. URL https://arxiv.org/abs/1301.3781. Introduces Word2Vec: skip-gram and CBOW architectures for static word embeddings.

Richard Montague. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julius Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, 1973.

Elliot Murphy. ROSE: A neurocomputational architecture for syntax, 2023. URL https://arxiv.org/abs/2303.08877. Defines ROSE (Representation, Operation, Structure, Encoding) as a neurocomputational architecture linking oscillatory coding to hierarchical syntax.

Elliot Murphy. Frege in the flesh: Biolinguistics and the neural enforcement of syntactic structures, 2026. URL https://arxiv.org/abs/2604.00291. On MERGE, algebraic syntax, and constraints on neural mechanisms; preprint.

David Jaz Myers. Double categories of open dynamical systems. *Electronic Proceedings in Theoretical Computer Science*, 380:319–336, 2023. doi: 10.4204/EPTCS.380.20. Formalizes open and interacting dynamical systems using Double Categorical Systems Theory (DCST).

Sanjeev V. Namjoshi. *Fundamentals of Active Inference: Principles, Algorithms, and Applications of the Free Energy Principle for Engineers*. MIT Press, 2026. ISBN 9780262050951. 576 pp., 8 × 10 in.; 213 color illus.; 24 b&w illus. Pub. March 17, 2026.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. doi: 10.3115/v1/D14-1162. Log-bilinear word embedding model capturing global co-occurrence statistics.

Steven Phillips. A category theory perspective on the language of thought: LoT is universal. *Frontiers in Psychology*, 15: 1361580, 2024. doi: 10.3389/fpsyg.2024.1361580.

Stefano Pirandola, Ulrik L. Andersen, Leonardo Banchi, Mario Berta, Darius Bunandar, Roger Colbeck, Dirk Englund, Tobias Gehring, Cosmo Lupo, Carlo Ottaviani, Jason L. Pereira, Mohsen Razavi, Jesni Shamsul Shaari, Marco Tomamichel, Vladyslav C. Usenko, Giuseppe Vallone, Paolo Villoresi, and Petros Wallden. Advances in quantum cryptography. *Advances in Optics and Photonics*, 12(4):1012–1236, 2020. doi: 10.1364/AOP.361502. Comprehensive review of QKD protocols, security proofs, and implementations.

Maria Polinsky and Omer Preminger. Case and grammatical relations. In Andrew Carnie, Dan Siddiqi, and Yosuke Sato, editors, *The Routledge Handbook of Syntax*. Routledge, 2014. URL https://omer.lingsite.org/files/Polinsky-and-Preminger-Case-and-Grammatical-Relations.pdf. Routledge 2014; citation key retains "2015" from prior drafts and secondary citations.

Quantinuum. lambeq Gen II: Quantum-enhanced interpretable and scalable text-based NLP. https://www.quantinuum.com/blog/lambeq-gen-ii-a-quantum-enhanced-interpretable-and-scalable-text-based-nlp-software-package, 2025. Official release (May 21, 2025) with DisCoCircReader for discourse-level quantum circuits from full texts.

Milena Rabovsky, Steven S Hansen, and James L McClelland. Modelling the n400 brain potential as change in a probabilistic representation of meaning. *Nature Human Behaviour*, 2(9):693–705, 2018.

Alireza Rad, Felippe Setiawan, Ryan Babbush, et al. Trainability enhancement of parameterized quantum circuits via reduced-domain parameter initialization. *Physical Review Applied*, 22:054005, 2024. doi: 10.1103/PhysRevApplied.22.054005. Reduced-domain initialization suppresses barren plateaus; DOI paywalled but journal record exists.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. OpenAI Technical Report, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. GPT-1: autoregressive generative pre-training for contextualized language representations.

Atsushi Shimojima. *On the Efficacy of Representation*. PhD thesis, Indiana University, 1996. Introduces "free ride" concept in diagrammatic reasoning.

Michael Silverstein. Hierarchy of features and ergativity. In R. M. W. Dixon, editor, *Grammatical Categories in Australian Languages*, pages 112–171. Australian Institute of Aboriginal Studies, 1976. Foundational paper on the animacy/definiteness hierarchy for split ergativity.

Julio Song. Category theory in theoretical linguistics: A monadic semantics for root syntax. In *Applied Category Theory (ACT) Conference*, 2022a. URL https://www.juliosong.com/doc/Song2022ACTabstract.pdf.

Julio Song. Category theory in theoretical linguistics: Categorial or categorical? https://blog.juliosong.com/linguistics/mathematics/category-theory-notes-3/, 2022b.

Christo Kurisummoottil Thomas and Mingzhe Chen. Fundamental limits of quantum semantic communication via sheaf cohomology, 2026. arXiv:2601.10958 (2026); quantum sheaves for multi-agent semantic networks; entanglement-assisted capacity exceeds classical.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010. doi: 10.1613/jair.2934. Comprehensive survey of vector space models for word meaning.

Jared Vasil, Paul B. Badcock, Axel Constant, Karl Friston, and Maxwell J. D. Ramstead. A world unto itself: Human communication as active inference. *Frontiers in Psychology*, 11:417, 2020. doi: 10.3389/fpsyg.2020.00417. URL https://discovery.ucl.ac.uk/id/eprint/10096443/. Active inference model of human communication as joint generative modeling.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://arxiv.org/abs/1706.03762. Introduces the transformer architecture with multi-head self-attention.

Grace Wu. *Verb classification, case marking, and grammatical relations in Amis*. PhD thesis, University at Buffalo, SUNY, 2024. URL https://arts-sciences.buffalo.edu/content/dam/arts-sciences/linguistics/AlumniDissertations/Wu%20dissertation.pdf.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005. doi: 10.1109/TIT.2005.850085.

Aleksandr K. Žolkovskij and Igor A. Mel'čuk. O vozmožnom metode i instrumentax semantičeskogo sinteza (on a possible method and instruments for semantic synthesis). *Naučno-texičeskaja informacija*, 5:23–28, 1965.