A Generative Account of Language Emergence through Free Energy–Driven Hierarchical Meta-Modelling.

A Preprint

Alexei Firssoff
Unaffiliated
a.a.firssoff@gmail.com

June, 2025

Abstract

We present **HELOS** (Hierarchical Emergence of Latent Ontological Structure), a hybrid compositional theoretical framework for self-organising structure learning. Initiated from atomic symbol sequences, HELOS incrementally constructs reusable structural hypotheses by minimising energy-based redundancy across hierarchical levels.

The framework induces language-like structures — including clusters, morphemes, compositional trees, and abstract symbolic references — without supervision or prior lexicons. It integrates discrete optimisation, topological encoding, and variational free energy principles to produce interpretable symbolic graphs.

The core assumptions concerning emergent symbolisation and structure discovery have been computationally implemented and empirically validated through experiments on unsupervised clustering and compositional parsing.

We outline a path toward formalising the resulting symbolic structures as objects in a cognitive topos, enabling a categorical interpretation of meaning, generalisation, and model-internal logic.

Keywords

Hierarchical Meta-Modelling, Emergent Morphology, Unsupervised Grammar Induction, Compositionality, Predictive Processing, Free Energy Principle, Task-Based Approach, Symbol Emergence, Symbol Grounding, FEP-based Reasoning

Contents

1 Introduction

1.1 Motivation and Problem

The emergence of complex, hierarchical structure from seemingly simple sequential input is a hallmark of human language, yet its underlying principles remain a profound scientific enigma. How do biological or artificial systems acquire the intricate grammatical knowledge necessary to parse and generate unbounded linguistic complexity, often from finite and noisy data streams?

This fundamental question lies at the intersection of cognitive science, linguistics, and artificial intelligence. While contemporary Large Language Models (LLMs) have achieved remarkable success in mimicking fluent language use by capturing vast statistical regularities in text corpora (1; 2), their operational paradigm raises critical concerns.

Primarily based on distributed vector representations and attention mechanisms, LLMs often function as "black boxes", lacking explicit, interpretable representations of the hierarchical constituent structure (e.g., morphology, syntax) that linguists consider fundamental (3; 4). This opacity hinders explainability, trustworthiness, and fine-grained control over generation. Furthermore, their reliance on surface statistical patterns, detached from perceptual or sensorimotor grounding, limits their capacity for true understanding and robust common-sense reasoning (5; 6).

Conversely, traditional symbolic approaches, which explicitly encode grammatical rules and structures (7; 8), suffer from brittleness, manual knowledge engineering bottlenecks, and difficulties in handling the inherent ambiguity and variability of natural language. They often struggle to scale and adapt to new data or linguistic phenomena without significant redesign. This highlights a crucial gap: **the need for a framework that can automatically discover and represent the inherent hierarchical and compositional structure of language in an interpretable and scalable manner, potentially bridging the gap between statistical learning and symbolic representation**, while ideally drawing inspiration from the mechanisms of biological cognition (9; 10). Addressing this challenge is paramount not only for advancing our scientific understanding of language and learning but also for developing more robust, efficient, and ultimately more intelligent artificial systems.

1.2 A New Paradigm Proposal: Data as an Active Substrate for Structure Emergence

Addressing the limitations of both purely statistical and purely symbolic approaches necessitates a shift in perspective. We propose viewing sequential linguistic data not merely as a passive object to be modelled *by* an external complex architecture (like an LLM or a predefined grammar), but as an *active substrate* within which *structure self-organises and emerges* through local interactions and global optimisation principles (11; 12).

Under this paradigm, the learning system does not impose a complex model onto the data, but rather provides a minimal computational medium (our recursive binary structure S 1) and a universal optimisation dynamic (approximating Free Energy minimisation, $F(T, w) \approx \text{Complexity} + \text{Surprise}$) (13). Linguistic structures (alphabets, morphemes, syntactic constituents) are not explicitly represented *a priori*, but emerge as stable, low-energy configurations (attractors) within this substrate, formed by the system's attempt to find the most efficient (predictively accurate and simple) way to represent and process the incoming data stream (14; 15).

The "intelligence" or "knowledge" of the system resides not in a vast, pre-trained model separate from the data, but is immanent within the learned parameters (e.g., $p(N_k)$ or `log_prior`) of the emergent structural nodes ($N_k \in S$) themselves and the predictive relationships between them. This **"data as substrate" perspective** shifts the focus from designing ever-larger external models to understanding the universal principles of self-organisation and emergent structure formation within the data itself, offering a potentially more fundamental, scalable, and cognitively plausible route towards more interpretable and robust artificial intelligence (12; 11). Our HELOS framework is presented as a concrete computational instantiation of this paradigm.

1.3 Core Idea

To address these limitations, we introduce a novel computational framework wherein hierarchical linguistic structure is not pre-defined but emergently discovered through the iterative application of a universal, compositional mechanism operating on sequential data. Central to our approach is the hypothesis that language processing can be considered as a hierarchical meta-modelling process, guided by principles analogous to predictive processing and free energy minimisation (9; 10). We posit that the cognitive system (or an effective AI) seeks to build an internal generative model of its sensory input (including language) that is both accurate in its predictions and parsimonious in its complexity.

Our framework operationalises this by:

1. Employing a minimalist, universal recursive data structure, defined by the type isomorphism

$$S \cong (I \times I) + (\mathbf{L}(I + S) \times I) + (S \times I), \tag{1}$$

capable of representing arbitrary hierarchical structures and sequences using discrete identifiers (I) within a category $\mathcal{C}$ with finite products ($\times$) and coproducts ($+$). Here $\mathbf{L}(Y) \cong 1 + (E \times \mathbf{L}(Y))$ with $E = I + S$ represents finite heterogeneous lists. This structure, built purely from identifiers and nested binary pairing, is conceptually akin to **Hereditarily Finite (HF) sets or lists built over the identifiers I as urelements**, a universal representation domain studied in computability theory and semantic modelling (16; 17; 18). It inherently encodes binary composition.

2. Implementing an iterative reduction algorithm that processes input sequences. This algorithm locally identifies predictively potent binary relational patterns (edges) between adjacent elements (represented as objects S or I).

3. Abstracting these patterns by creating new nodes S of the form (l, i_{model}) where $l \in \mathbf{L}(I + S)$ is the sequence segment exhibiting the pattern and $i_{model} \in I$ is the identifier for the discovered model (pattern type).

4. Replacing the segment l with the new abstract node $S = (l, i_{model})$ in the sequence representation, yielding a shorter, more abstract sequence for the next iteration.

This cycle continues recursively. Crucially, the selection of patterns and the resulting optimal parse tree T^* for an input sequence w is governed by the minimisation of a Free Energy functional $F(T, w)$, which can be approximated as:

$$F(T, w) \approx \underbrace{\sum_{N_k \in T} \text{Complexity}(N_k)}_{\text{Total Complexity}} + \underbrace{\sum_{\text{steps } j \text{ in } T} \text{Surprise}_j}_{\text{Total Surprise}} \tag{2}$$

where $\text{Complexity}(N_k)$ reflects the cost of using node N_k (e.g., derived from its prior probability, $-\log p(N_k)$) and Surprise_j is the prediction error (e.g., $-\log p(\text{data}_j|\text{model}_j)$) at each composition step j in the construction of tree T. This mechanism allows complex linguistic structures, starting from morphology, to emerge autonomously.

In essence, our framework proposes a computational realisation and synthesis of deep linguistic and cognitive principles. It operationalises the binary compositionality central to generative linguistics (cf. Merge (19; 20)) through the universal recursive structure S (Eq. 1). However, diverging from the search for innate grammatical rules (Universal Grammar), we propose that the fundamental invariant might be a universal **mechanism** for learning and constructing structure.

Our iterative reduction process embodies such a mechanism. The discovery and selection of optimal hierarchical structures T^* within the Catalan space (21) are governed by optimisation principles inspired by the Free Energy Principle (FEP) (9), seeking representations that minimise $F(T, w)$ (Eq. 2) by balancing predictive accuracy (surprise) and model complexity.

While FEP/PP theories are often abstract or applied to simpler perceptual tasks involving continuous signals, we provide a concrete, detailed architecture and mechanism demonstrating how these principles can operate on discrete, symbolic, hierarchical structures like language, specifically applying prediction and error minimisation not to raw signals, but to the composition of discrete structural units (nodes S) into parse trees (T).

This synergy allows for the unsupervised, emergent discovery of meaningful linguistic units (demonstrated herein for morphemes) as stable, low-free-energy optima arising from the self-organisation of less structured input. We thus offer a principled, unified mechanism for learning the hierarchical nature of language from data.

1.4 Key Hypotheses

Our framework is underpinned by several core hypotheses regarding the fundamental nature of language structure and processing:

1. **Primacy of Binary Composition:** We hypothesise that the intricate hierarchical structure of language, from morphology to syntax, *may be* fundamentally constructed through the recursive application of a simple binary composition operation. This operation takes two existing entities (primitive identifiers I or previously constructed nodes S) and merges them into a new structural unit. This mirrors the elegance and generative power of operations like 'Merge' proposed in "The Minimalist Program" (19; 20). This principle is directly embodied in our universal recursive structure, formally defined via the type isomorphism 1), where complex nodes are formed by pairing either a heterogeneous list or a node with an identifier.

2. **Emergence through Predictive Optimisation (FEP / Kolmogorov Complexity):** We propose that meaningful linguistic units (e.g., morphemes, words, grammatical categories, syntactic rules) *may be* not innate primitives but rather emergent properties of a system optimising its internal model to efficiently predict and compress the linguistic input. This optimisation process is conceptually grounded in the Free Energy Principle (FEP) (22; 9), which mandates the minimisation of a functional $F(T, w)$ (approximated in Eq. 2) over possible interpretations (parse trees T) of the data w. Minimising F inherently involves finding a balance between Predictive Accuracy (minimising the Surprise term, $\approx -\log p(w|T)$) and Model Complexity (minimising the $\sum \text{Complexity}(N_k)$ term, $\approx \sum[-\log p(N_k)]$). This naturally implements Occam's Razor and aligns closely with the principles of Solomonoff induction (23; 24), where simpler models (lower Kolmogorov complexity (25), approximated by higher prior probability $p(N_k)$ or lower prior FE) are favoured. In our framework, morphemes and grammatical patterns emerge as precisely those recurring sub-structures (nodes N_k) that achieve high prior probability $p(N_k)$ because they consistently contribute to parse trees T with low overall Free Energy $F(T, w)$ across many linguistic examples. Learning involves updating these priors $p(N_k)$ and the internal predictive parameters of the nodes based on observed prediction errors, akin to Bayesian inference.

3. **Universality of the Mechanism:** We hypothesise that the core meta-modelling mechanism – iterative, compositional abstraction driven by the universal representation 1) and the FEP-based optimisation guiding the search for the best parse T^* minimizing $F(T, w)$ (Eq. 2) – *may be* language-general. While the specific high-probability nodes N_k (representing morphemes, rules, etc.) discovered will depend on the statistical regularities of a given language, the underlying process of discovery and representation remains constant. Furthermore, we posit that this mechanism is potentially modality-general, offering a unified approach to learning hierarchical structure in visual, auditory, or other domains, thereby providing a principled basis for structurally grounded multi-modal AI.

1.5 Inductive Bias

The explicit choice of representing linguistic structure through recursive binary compositions, formally captured by the type structure 1), is not merely an implementational detail; it introduces a crucial inductive bias. By restricting the system to merging exactly two constituents (either retrieving a node S and pairing it with an identifier I, or retrieving a list $L \in \mathbf{L}(I + S)$ and pairing it with a model identifier I) at each step of hierarchical construction, we inherently favour constituency-based analyses built upon binary-branching trees.

This bias aligns strongly with dominant linguistic theories like Minimalism (7; 19; 26) but is adopted here primarily as a **minimalist computational hypothesis**. It significantly prunes the hypothesis space (the Catalan space C_{n-1} (21)) and, crucially, allows the FEP-driven learning of predictive dependencies between nodes to guide the system towards linguistically plausible structures, even for phenomena seemingly requiring non-binary relationships, such as circumfixation (see Section 6.5 for discussion). This inherent structural preference, coupled with predictive optimisation, facilitates the learning of meaningful units from sparse data.

This inherent structural preference acts as a form of regularisation, guiding the system towards discovering linguistically plausible hierarchical groupings and facilitating the learning of meaningful units (like the morphemes demonstrated later) even from sparse or ambiguous data. While we adopt strict binarity as a core computational principle, **we acknowledge the ongoing linguistic debate regarding its absolute universality, positioning it here as a powerful, minimalist, and empirically validated inductive bias** for structure discovery within our framework.

1.6 Contributions and Outline

The primary contribution of this work is the proposal and initial computational validation of HELOS, a unified framework for the unsupervised learning and emergent discovery of hierarchical language structure. Specifically, we introduce a hierarchical meta-modelling approach grounded in principles of binary composition and predictive optimisation (FEP-like), utilising a universal recursive representation (S, Eq. 4). We demonstrate through prototype experiments processing unannotated data from diverse languages (e.g., Russian, Turkish, French and German) that this framework can autonomously discover foundational character categories, subsequently segment the input into word-like models, and critically, induce linguistically plausible morphological structures (morphemes) within those segments. Crucially, we show that the learned representations enable compositional analysis of unseen words, highlighting the generalisation capabilities of the emergent structures. We further outline the theoretical basis for extending this unified mechanism to syntax and multi-modal grounding.

The remainder of this paper is structured as follows: Section 2 reviews the relevant theoretical background and related work. Section 3 introduces the core principles of the emergent hierarchical meta-modelling engine. Section 4 details the proposed HELOS framework, including its formal foundations, the overall multi-stage architecture (covering emergent symbol categorisation, tokenisation, and morphology), and theoretical extensions towards syntax, semantics, and active inference. Section 5 describes the experimental setup used to validate the initial stages (categorisation and morphology) with a prototype and presents the results of these experiments, focusing on the emergent discovery of structure and compositional generalisation. Section 6 discusses the findings, theoretical implications, advantages, and limitations. Section 7 concludes with the broader impact, and Section 8 provides acknowledgements. The prototype code is available at the OpenHELOS GitHub repository.

2 Theoretical Background and Related Work

2.1 Compositionality and Hierarchy in Language

A foundational observation spanning diverse linguistic theories is that human language exhibits profound compositionality and hierarchy (3). Sentences are not mere linear strings of words, nor are words simple concatenations of sounds or letters. Instead, linguistic expressions are systematically constructed by combining smaller units into larger constituents, whose properties and meanings are, to a significant extent, determined by the properties of their parts and the rules governing their combination (27). This hierarchical structure is evident across multiple levels of linguistic analysis. Phonemes combine into morphemes, morphemes into words, words into phrases, and phrases into clauses and sentences, forming nested structures often represented as trees in syntactic analysis.

Two dominant paradigms for capturing this hierarchical structure are Constituency Grammars (Phrase Structure Grammars) and Dependency Grammars. Constituency grammars (3; 7) explicitly group words and phrases into nested constituents (like Noun Phrases, Verb Phrases), typically resulting in tree structures where non-terminal nodes represent these phrasal categories. Dependency grammars (28; 29), conversely, focus on direct grammatical relations (dependencies, such as subject-verb, modifier-head) between individual words (lexical heads), also forming directed tree or graph structures. Despite their different formalisms, both approaches fundamentally acknowledge and rely on the hierarchical organisation of syntactic relations.

Crucially, the operation responsible for building these hierarchies is often conceptualised as fundamentally binary. Within the Minimalist Program (19; 20), Chomsky proposes Merge as the simplest and potentially sole structure-building operation. Merge takes exactly two syntactic objects (either lexical items or objects already formed by Merge) and combines them into a new, larger set-theoretic object, inherently creating a binary-branching hierarchical structure. This resonates with proposals in other frameworks suggesting that even apparently flat or n-ary branching structures in syntax might be derived from underlying binary compositions (26; 30).

The framework proposed in this paper directly embraces this core linguistic insight. Our universal recursive representation 1 inherently encodes binary composition as its fundamental structure-building mechanism. The iterative reduction process, which identifies patterns and creates abstract nodes, computationally mirrors the recursive application of a Merge-like operation, progressively building hierarchical representations from the bottom up. By grounding our computational mechanism in this principle of binary composition and hierarchy, we align our approach with deep theoretical assumptions about the nature of linguistic structure itself.

2.2 Predictive Processing and the Free Energy Principle (FEP)

Complementing the structural insights from linguistics, our framework's learning and inference mechanisms are deeply inspired by prominent theories of brain function, namely Predictive Processing (PP) and the Free Energy Principle (FEP) (22; 9; 10; 31). These influential frameworks provide a compelling, potentially unifying account of perception, learning, and cognition, viewing the brain not as a passive recipient of sensory information, but as an active, hypothesis-testing prediction engine.

The core idea of PP is that the brain constantly generates top-down predictions about the expected sensory input based on its internal generative models of the world. These predictions cascade down the cortical hierarchy. Simultaneously, bottom-up signals carrying sensory information (or prediction errors from lower levels) ascend the hierarchy. At each level, a comparison occurs: the difference between the prediction and the actual input constitutes the prediction error. This error signal is then propagated upwards, serving as the primary driver for updating and refining the higher-level generative models (hypotheses or beliefs) to improve future predictions (10; 31). Learning, in this view, is the process of adjusting model parameters to minimise prediction errors over time (9).

The FEP, proposed by Friston (9), offers a fundamental, first-principles explanation for this predictive dynamic, rooted in statistical physics and information theory. It posits that any self-organising system aiming to maintain its integrity in a changing environment must minimise the long-term average of surprise (or negative log model evidence). As surprise is computationally intractable, FEP introduces variational free energy, F, as a more tractable upper bound. Formally, for a system with internal states μ representing parameters of a generative model $p(o, c|\mu)$ of observations o and their hidden causes c, and an approximate posterior belief $q(c|\mu)$ about those causes, free energy is:

$$F(\mu, q) = \underbrace{E_q[\log q(c|\mu) - \log p(c|\mu)]}_{\text{Complexity (KL Divergence)}} - \underbrace{E_q[\log p(o|c, \mu)]}_{\text{Accuracy (Log Likelihood)}} \tag{3}$$

Minimising F simultaneously achieves two goals: it maximises the accuracy of the model's explanation for the observations (by minimising the second term, which represents surprise or negative log-likelihood), and it minimises the complexity of the model (by keeping the belief q close to the prior $p(c|\mu)$ via the first KL-divergence term), effectively implementing a form of Occam's razor. Perception then becomes the process of optimising beliefs q to minimise F for given model parameters μ, while learning becomes the slower process of optimising the model parameters μ themselves to minimise F over time (13).

Our framework adopts this FEP perspective conceptually. The search for the optimal parse tree T^* for an input w corresponds to an inference process minimising an objective function $F(T, w)$ (approximated in Eq. 2) that balances the complexity of the tree (derived from the priors/complexity of its constituent nodes N_k) and the surprise (accumulated prediction error during the tree's construction). The subsequent learning step (updating node parameters, including their priors $p(N_k)$ and predictive models, via 'bayesian_update') corresponds to adjusting the generative model based on prediction errors to reduce future free energy. This positions the emergent discovery of morphology as the system finding the simplest, most predictively powerful hierarchical model of the sequential word-form data.

2.3 Unsupervised Structure Learning

The challenge of learning linguistic structure without explicit supervision (e.g., annotated treebanks or morphological analyses) has been a long-standing pursuit in NLP and computational linguistics. Significant efforts have focused on unsupervised morphology induction, aiming to automatically segment words into morpheme-like units and build a lexicon of these units from raw text. A prominent family of approaches employs the Minimum Description Length (MDL) principle (32; 33), which, similar in spirit to FEP's complexity term, seeks the model that provides the most compact description of the data and the model itself. Systems like Morfessor (34; 35) iteratively refine a morphological lexicon and segmentation model by balancing the cost of encoding the lexicon with the cost of encoding the corpus using that lexicon. While successful in segmenting words for many languages, MDL-based methods typically yield flat, sequential segmentations rather than the hierarchical, tree-like structures representing the derivational history inherent in our approach. Furthermore, their optimisation criteria (description length) differ from the explicitly predictive and potentially more neurally plausible objective of minimizing free energy.

Other unsupervised morphology approaches have utilized probabilistic models, such as Hidden Markov Models (HMMs) or Bayesian nonparametric models (36; 37), to infer latent morphological categories and segmentation

points. While capable of capturing sequential dependencies, these models often lack the capacity to represent the deep hierarchical compositionality that our binary tree framework naturally affords.

Beyond morphology, the field of Grammar Induction (38) attempts to learn syntactic rules or structures (like context-free grammars or dependency grammars) from unannotated sentences. Classic approaches often rely on distributional statistics (e.g., identifying recurring N-grams or co-occurrence patterns) or algebraic methods (39). More recent work explores neural approaches, attempting to induce latent syntactic structures within sequence models (40; 41), though often the induced structures lack explicit interpretability or strong linguistic correspondence. Bayesian methods have also been applied, seeking grammars with high posterior probability given the data (42; 43), often using MDL or simplicity priors that resonate with our framework's complexity term.

However, our approach differs significantly from these prior works in several key aspects. Firstly, we propose a unified mechanism intended to operate seamlessly across different levels of linguistic hierarchy, discovering morphological structure first and potentially extending to syntax using the same iterative, compositional reduction process. Secondly, our use of a universal recursive binary representation (S) provides a specific, linguistically motivated structural bias towards hierarchical constituency. Thirdly, our optimisation objective, grounded in the Free Energy Principle, explicitly links structure discovery to predictive processing, offering a potentially deeper cognitive and computational justification than purely information-theoretic (MDL) or probabilistic criteria. Finally, the resulting representations are not just segmentations or abstract latent variables but explicit, interpretable, hierarchical parse trees built from emergently discovered, reusable structural units (our nodes N_k).

2.4 Recursive Data Structures as Local Pattern Models

The representation of hierarchical and sequential data through **recursive types** is a cornerstone of theoretical computer science and programming language theory (44; 45). Inductively defined types, such as lists (formally, $\mathbf{L}(X) \cong 1 + (X \times \mathbf{L}(X))$) or binary trees ($\mathbf{Tree}(X) \cong X + (\mathbf{Tree}(X) \times \mathbf{Tree}(X))$), provide a formal basis for defining structures using finite descriptions. Our universal representation, defined by the isomorphism

$$S \cong (I \times I) + (\mathbf{L}(I + S) \times I) + (S \times I)$$

(where $\mathbf{L}(Y) \cong 1 + ((I + S) \times \mathbf{L}(Y))$ defines heterogeneous lists), falls directly within this tradition. It defines the type S recursively, allowing structures of arbitrary depth to be constructed solely through nested binary pairings.

Crucially, each internal node $s \in S$ formed via the binary pairing constructors (either $(l, i_{model}) \in L(I + S) \times I$ or $(s', i_{attr}) \in S \times I$) can be interpreted not just as a structural grouping, but as a **local model of a discovered pattern**. The node encapsulates a specific composition (either a sequence l tagged as model i_{model}, or a structure s' tagged with attribute i_{attr}) that the system has identified as statistically significant or predictively useful during the iterative reduction process. The parameters associated with this node in the memory M (its prior probability $p(s)$ and its predictive models) quantify the learned properties of this local pattern model. Thus, the entire hierarchical tree T^* constructed for an input represents a **composition of these learned local pattern models**, providing a multi-level explanation of the input sequence. This connection provides access to powerful formal tools (45) and grounds our language representation in established principles of data structuring while highlighting the model-like nature of each compositional unit.

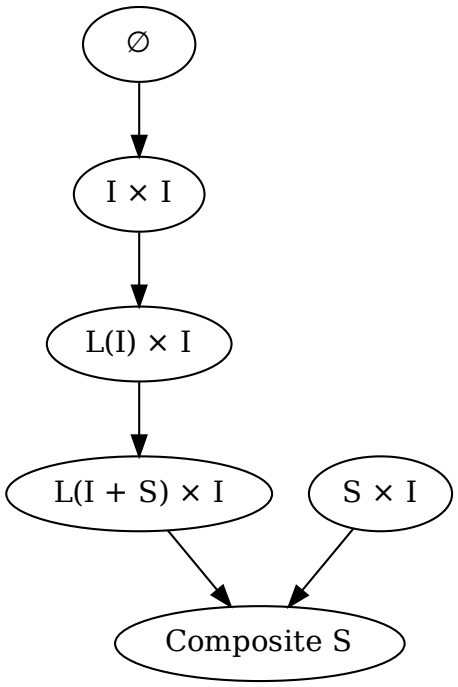


Figure 1: Hasse diagram of the order $\leq$ on S. Each node represents a structured term in the HELOS model; edges denote compositional inclusion.

To ensure the correctness of the structural representation `S`, we formally verified A.1 its bidirectional isomorphism to a sum-of-products normal form using the Coq proof assistant. Specifically, we defined an inductive type `S_list` capturing the hierarchical structure, and a corresponding flattened type `RHS_list_type`. We then constructed a pair of functions `slist_to_rhs` and `rhs_to_slist`, and proved that they are mutual inverses. The equivalence $S\_list \leftrightsquigarrow RHS\_list\_type$ was formally established via two lemmas, ensuring that no information is lost or altered during transformation. This verification guarantees the structural soundness of the `S` representation as a typed algebraic object.

2.5 Connection to Computability, Semantic Modelling, and Predictive Inference

The recursive, combinatorial nature of the structure S and the parse trees T constructed within our framework establishes deep connections to foundational work in computability theory, constructive models, topology for discrete mathematics, and semantic probabilistic inference, particularly the research programmes originating from the Novosibirsk school of logic and computer science (Ershov, Goncharov, Sviridenko, Vityaev) (17; 16; 46; 47; 18).

Our structures S (Eq. 4), built hierarchically from discrete identifiers I, are computable objects akin to hereditarily finite structures over urelements (18). The space of these structures admits a natural Alexandrov topology (48) derived from structural partial orders (e.g., subtree relation), situating our work within the context of topology for discrete, computable spaces explored by Ershov (47).

More profoundly, our framework can be viewed as a specific instantiation and significant extension of the semantic modelling paradigm advocated by Goncharov, Ershov, and Sviridenko. They proposed shifting from purely axiomatic/syntactic approaches (like traditional logic programming or rule-based systems) towards building constructive models of the domain where computation corresponds to evaluation or truth-checking within the model, thus preserving the original semantics. Our system embodies this by constructing explicit hierarchical trees T^* which serve as structural models of the input linguistic data w.

Furthermore, our emphasis on prediction and optimisation via the Free Energy Principle resonates strongly with Vityaev's work on Semantic Probabilistic Inference of Predictions (46). Vityaev critiques traditional logical inference for its handling of probabilistic knowledge and prediction, arguing that logical entailment can drastically decrease probability estimates, failing to capture the essence of prediction as finding the **most probable** consequence given evidence. He proposes a "semantic probabilistic inference" which, instead of logical derivation, searches for facts or rules within a probabilistic model that **maximise the conditional probability** of the predicted statement.

Our FEP-driven search for the optimal parse tree T^* can be seen as a concrete realisation and generalisation of this semantic predictive inference. Minimising Free Energy $F(T, w) \approx \text{Complexity}(T) + \text{Surprise}(w|T)$ precisely involves finding the tree structure T (our "model" or "explanation") that maximises the likelihood $P(w|T)$ (minimises Surprise) while controlling for model Complexity ($-\log p(T)$). The nodes N_k within the optimal T^* that contribute most significantly to reducing surprise (i.e., the emergently discovered morphemes) correspond conceptually to the "probabilistic regularities" or "best predictors" sought in Vityaev's framework. Our approach extends this by:

1. Providing a universal, hierarchical representation (S) for these structures.
2. Employing a unified optimisation principle (FEP) derived from first principles.
3. Demonstrating emergent discovery of the predictive units (morphemes) themselves, rather than assuming a predefined set of rules or facts.

Therefore, our framework is positioned not only within the tradition of constructive models and topology for discrete mathematics (Ershov) and semantic modelling (Goncharov, Sviridenko), but also significantly advances the programme of semantic predictive inference (Vityaev) by providing a novel, FEP-grounded, unsupervised mechanism for learning the very structures that enable optimal prediction within linguistic data.

2.6 Relationship to Task-Based Approaches and Generalisation

The proposed hierarchical meta-modelling framework can be rigorously positioned as a generalisation and advancement of the task-based semantic modelling approaches developed previously (49). Task-based approaches typically involve constructing a specific constructive model or theory $\mathcal{M}_{task}$ tailored to solve a particular problem by evaluating terms or formulae ϕ within $\mathcal{M}_{task}$.

In contrast, our framework does not predefine a task-specific model. Instead, it implements a universal meta-level process that seeks the optimal hierarchical representation $T^* \in \mathcal{T}_w$ for the input data w itself, guided by the task-independent principle of Free Energy Minimisation (Eq. 5). The solution to specific "tasks" (like morphological analysis or segmentation) emerges implicitly from the structure of the optimal tree T^* and the parameters of the emergent nodes N_k within it, rather than being computed relative to an explicit, predefined task model $\mathcal{M}_{task}$.

Formally, let $\mathbb{T}$ be the space of all possible parse trees (objects of type S rooted appropriately) and $\mathbb{W}$ be the space of input sequences. Let $\mathcal{F}_{task} : \mathbb{W} \to \text{Solutions}$ be the function representing the solution of a specific task (e.g., mapping a word sequence to its morpheme sequence). A task-based approach explicitly constructs a model $\mathcal{M}_{task}$ such that $\mathcal{F}_{task}(w)$ can be derived from computations within $\mathcal{M}_{task}$. Our meta-modelling approach, however, defines a universal mapping $Encode : \mathbb{W} \to \mathbb{T}$ (where $T^* = Encode(w)$ minimises FEP) and learning $Learn : \mathbb{T} \times \mathbb{W} \times M \to M$. The solution $\mathcal{F}_{task}(w)$ is then extracted or interpreted from the resulting T^* and the learned memory M, without constructing $\mathcal{M}_{task}$ explicitly for each task.

This constitutes a generalisation because the single meta-modelling mechanism $Encode/Learn$ is capable of inductively discovering the structures relevant to potentially *multiple* implicit tasks (segmentation, morphology, syntax...) by optimising a universal objective function, rather than requiring separate model specifications for each. It shifts the focus from task-specific model construction to universal mechanisms of structure emergence and predictive learning.

2.7 Comparison with Implicit Structure Learning in LLMs

It is instructive to compare our explicit hierarchical meta-modelling approach, encapsulated in the HELOS framework, with the mechanisms by which current Large Language Models (LLMs), particularly transformers, implicitly capture linguistic structure. While LLMs lack the explicit symbolic representations and recursive reduction cycle central to HELOS, studies into their internal workings suggest analogous, albeit distributed and implicit, processes at play.

Word and subword embeddings intrinsically perform a form of implicit clustering, grouping semantically and syntactically similar units closer in vector space (50; 51). Furthermore, attention mechanisms, especially distinct attention heads, have been shown to specialise in tracking specific relational patterns, such as subject-verb agreement, co-reference, or syntactic dependency links (52; 53). This can be interpreted as a dynamic, context-dependent form of relational clustering or soft binding.

Similarly, the contextualised representations formed in higher layers of LLMs can be seen as distributed, implicit abstractions or "local models" for complex contextual meanings or emergent concepts (54; 55; 56). Probing studies demonstrate that significant linguistic information (e.g., part-of-speech tags, parse structures, semantic roles) can be linearly decoded from these hidden states, indicating that such information is implicitly encoded (54; 57).

2.7.1 Pairs as the Nexus of HELOS and Transformer Meta-Modelling

To elucidate the parallels between HELOS and transformers, we focus on the combinatorial enumeration of token pairs, which serves as the foundation for both frameworks' ability to model hierarchical structures. Both HELOS and transformers engage in meta-modelling, constructing internal models of data dependencies and abstractions that are adaptively applied to new inputs. This shared paradigm, rooted in pairwise dependencies, enables the representation of complex hierarchies as n-ary trees, which are reducible to binary trees as a universal encoding.

In transformers, self-attention computes dependencies across all pairs of tokens in an input sequence. Given an input matrix $H \in \mathbb{R}^{n \times d}$, where n is the number of tokens and d is the embedding dimension, each attention head h (for $h = 1, \ldots, H$) projects H into queries, keys, and values:

$$Q_h = HW_{Q,h}, \quad K_h = HW_{K,h}, \quad V_h = HW_{V,h},$$

where $W_{Q,h}, W_{K,h}, W_{V,h} \in \mathbb{R}^{d \times d_k}$, and $d_k = d/H$. The attention matrix is:

$$A_h = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) \in \mathbb{R}^{n \times n},$$

where $A_h(i, j)$ quantifies the relevance of token j to token i in head h. This matrix encapsulates all n^2 pairwise dependencies, forming a weighted directed graph of relationships.

The output of each head is:

$$Z_h = A_h V_h,$$

and multi-head attention concatenates these outputs:

$$Z = \text{Concat}(Z_1, \ldots, Z_H)W_O,$$

where $W_O \in \mathbb{R}^{Hd_k \times d}$. This process constitutes meta-modelling, as the attention mechanism constructs a dynamic model of dependencies, with each head specialising in distinct patterns (e.g., syntactic, semantic), akin to an ensemble of relational models. Across L layers, transformers build a hierarchy of representations:

$$H^{(l)} = f^{(l)}(H^{(l-1)}; \theta^{(l)}), \quad l = 1, \ldots, L,$$

where deeper layers encode increasingly abstract concepts, mirroring the hierarchical organisation of HELOS.

HELOS formalises these emergent hierarchies as n-ary (binary) trees, where nodes represent embeddings $h_i^{(l)}$ and edges denote dependencies derived from $A_h(i, j)$. For a node $h_i^{(l)}$, its children are $\{h_j^{(l-1)}\}_{j=1}^n$, weighted by $A_h(i, j)$, reflecting the combinatorial nature of attention. Since each token may depend on up to n others, the tree is n-ary. HELOS engages in meta-modelling by constructing a tree that models the data's structure, with lower levels encoding local dependencies and higher levels capturing abstract concepts, adaptively applied to new inputs.

A key insight is that both HELOS and transformer hierarchies can be indirectly reduced to binary trees (58; 59). For an n-ary tree $T^{(l)} = (V^{(l)}, E^{(l)})$, with nodes $V^{(l)} = \{h_i^{(l)} \mid i = 1, \ldots, n\}$ and edges $E^{(l)} = \{(h_i^{(l)}, h_j^{(l-1)}) \mid A_h(i, j) > \tau\}$, the binary tree $T'^{(l)}$ is constructed using the first-child/next-sibling representation. For a node $v = h_i^{(l)}$ with children $\{c_1, c_2, \ldots, c_k\} = \{h_{j_1}^{(l-1)}, h_{j_2}^{(l-1)}, \ldots\}$, ordered by $A_h(i, j)$, the binary node v' is:

$$\text{left}(v') = c'_1, \quad \text{right}(c'_1) = c'_2, \quad \text{right}(c'_2) = c'_3, \quad \ldots, \quad \text{right}(c'_k) = \text{null}.$$

This bijective transformation preserves dependencies, compressing the combinatorial space of pairs into a minimal structure, aligning with the principle of minimising Kolmogorov complexity.

Pairs are the unifying element, as both frameworks rely on combinatorial pair enumeration to model dependencies. In transformers, pairs (i, j) with significant $A_h(i, j)$ form the edges of the HELOS tree, while multi-head attention enriches the hierarchy by modelling multiple dependency types. The shared meta-modelling paradigm ensures that both HELOS and transformers construct adaptive, compressive representations of data structure.

Crucially, while transformers achieve their power by modelling statistical distributions over this combinatorial space implicitly within their distributed weights and activations, HELOS aims to extract and represent these crucial distributions and the underlying structural organisation in a clear, explicit, and interpretable hierarchical form.

2.7.2 Differences in Explicitness and Compositionality

Despite these parallels, a crucial difference lies in the explicitness and nature of the abstraction. Transformers typically lack the explicit reduction step central to HELOS; their representations are enriched contextually, but the sequence length often remains, and hierarchical constituency is not reified into discrete nodes. Moreover, the discovered "clusters" and "abstractions" in LLMs remain distributed patterns of activation or geometric relationships, lacking the symbolic interpretability and compositional transparency of the nodes S and parse trees T^* constructed by HELOS (60; 61). Our framework thus aims to make explicit, structured, and recursively compositional the types of structural learning that occur implicitly and in a distributed fashion within transformers.

The meta-modelling perspective of HELOS offers a pathway to enhance transformer interpretability, as binary tree representations can visualise attention-derived dependencies. Additionally, structuring dependencies hierarchically may reduce computational complexity from $O(n^2)$ to $O(n \log n)$, opening avenues for efficient architectures. By formalising the shared reliance on pairs and hierarchical compression, HELOS bridges implicit and explicit structure learning, advancing our understanding of neural network representations.

2.8 Positioning and Novelty

While drawing inspiration from established concepts in linguistics (compositionality, hierarchy, binary merging), cognitive science (predictive processing, FEP), unsupervised learning (MDL, probabilistic models), and computer science (recursive data types), our proposed framework offers a novel synthesis and a distinct approach to understanding and modelling language structure.

Specifically, our contribution is distinguished by:

1. **A Unified, Multi-Level Mechanism:** Unlike approaches focusing solely on morphology (like Morfessor) or syntax (like many grammar induction methods), our iterative, hierarchical meta-modelling process, driven by the same underlying principle (FEP-like optimisation on binary structures), is hypothesised to operate seamlessly across linguistic levels, naturally bridging morphology and potentially syntax through the recursive application of pattern abstraction and reduction.

2. **Explicit, Interpretable, Hierarchical Representation:** In contrast to the implicit structural encoding in LLMs or the flat segmentations often produced by unsupervised morphology systems, our framework generates explicit, inherently hierarchical, and interpretable parse trees (T^*) built from emergently discovered, meaningful units (nodes N_k). The universal recursive structure 1 provides the formal backbone for this representation.

3. **Cognitively Motivated Optimisation:** Grounding the structure discovery process in the Free Energy Principle provides a deeper, potentially more fundamental theoretical motivation than purely information-theoretic (MDL) or maximum likelihood criteria. It explicitly links language modelling to broader principles of predictive brain function and self-organisation.

4. **Emergence from Minimal Priors:** While incorporating a strong structural bias (binary composition), the system relies on minimal *linguistic* priors beyond basic character attributes (Category/Script). Complex units like morphemes emerge from the optimisation process itself, rather than being explicitly enumerated or requiring extensive feature engineering.

5. **Modality-General Potential:** The core mechanism's reliance on abstract identifiers I and the universal compositional structure S makes it inherently modality-neutral, offering a principled pathway towards integrating language with other modalities (e.g., vision) through structurally grounded representations, directly addressing the symbol grounding problem.

In essence, not merely a new algorithm for a specific task is proposed, but a fundamental computational framework for how complex, hierarchical, and compositional structure, characteristic of human language, might be learned and represented by an intelligent system interacting with sequential data.

It is important to emphasise that the primary objective of this work, at its current stage, is not to compete directly with the performance of state-of-the-art Large Language Models (LLMs) on broad benchmark tasks. Rather, our goal is to propose and provide initial validation for the core principles of our neurosymbolic AI framework (HELOS), focusing on the fundamental mechanisms of emergent structure discovery, hierarchical representation, and compositional generalisation through predictive optimisation. We aim to demonstrate the viability and potential of this alternative paradigm. While significant further research and engineering efforts are required for scalability and application to complex downstream tasks, we believe the principles outlined herein offer a promising foundation upon which future architectures, capable of complementing or potentially competing with current LLM approaches by virtue of their structural understanding and interpretability, can be built.

3 Core Principles: Emergent Hierarchical Meta-Modelling

The framework proposed is built upon a core engine of **emergent hierarchical meta-modelling**. This engine operates iteratively to discover and represent structure within sequential data, such as language, without relying on predefined segmentation rules or linguistic grammars. The fundamental idea is that meaningful structure, at all levels of granularity, arises naturally as the system seeks efficient, predictive representations of the input stream. This process can be conceptualised as an iterative cycle of pattern discovery, abstraction (modelling), and reduction.

3.1 Meta-Modelling: The Recursive Construction of Models

3.1.1 Definition

A defining characteristic of the HELOS framework is its operation as a **hierarchical meta-modelling** engine. Rather than learning a single monolithic model, the system recursively constructs **models of models**. This approach is supported by research in hierarchical meta-learning, which demonstrates the effectiveness of organizing knowledge across multiple levels of abstraction (62; 63). This process leverages the universal structure S and the iterative "Pattern $\rightarrow$ Abstract/Model $\rightarrow$ Reduce" cycle (Section 3.3).

- **Level 0 Models (M_0):** At the base, the system learns or utilises nodes $S_{Category}$ (via the $I \times I$ constructor or clustering) representing categories of primitive inputs (e.g., character classes i_{class}). These serve as the initial 'vocabulary' of models.
- **Level 1 Models (M_1):** The system then analyses sequences l_0 composed of Level 0 entities (e.g., attributed characters). It identifies patterns P_1 in these sequences and creates Level 1 model nodes $s^{(1)} = (l'_0, i_{P1}) \in S$ (via the $L \times I$ constructor). These $s^{(1)}$ nodes are effectively models of sequences of Level 0 models/entities. Examples include token models like 'CyrSeq'.
- **Level 2 Models (M_2):** Subsequently, the system analyses sequences l_1 composed of Level 1 model nodes (e.g., a sequence of token models). It identifies patterns P_2 in *these* sequences and creates Level 2 model nodes $s^{(2)} = (l'_1, i_{P2}) \in S$. These $s^{(2)}$ nodes are models of sequences of Level 1 models. Examples include phrase structure models or grammatical relation models built from PoS models (which themselves are abstractions over word/morpheme models).
- **Recursive Application:** This process continues recursively: Level n models (M_n) are discovered by identifying patterns P_n in sequences l_{n-1} composed of Level $n-1$ models (M_{n-1}) and creating new nodes $s^{(n)} = (l'_{n-1}, i_{Pn}) \in S$. This recursive construction aligns with formal approaches in recursive model theory, which describe iteratively built models (64).

3.1.2 Formal Interpretation

Let S_k denote the set of nodes S primarily representing models discovered or operating at hierarchy level k. The meta-modelling process involves functions $Meta_k$:

$$Meta_k : L(I + \bigcup_{j \leq k} S_j) \to S_{k+1}$$

where $Meta_k$ takes a list composed of identifiers and nodes up to level k, identifies a pattern P_{k+1} characteristic of level $k+1$, and produces a new node $s^{(k+1)} \in S_{k+1}$ representing the model for that pattern, typically via the $L \times I$ constructor. This formalisation is consistent with hierarchical meta-learning frameworks that optimize models across levels (62; 63).

3.1.3 Continuous Validation

This hierarchical construction is continuously validated by the overarching FEP optimisation. A node $s^{(k)}$ (a model at level k) persists and strengthens (its $p(s^{(k)})$ increases) only if it proves useful in constructing low-Free-Energy representations T^* for input data, meaning it contributes positively to both predictive accuracy and representational simplicity across the hierarchy. Models that do not generalise or predict well are implicitly pruned or assigned low probability by the FEP dynamics. This validation process is grounded in the free-energy principle, which uses hierarchical generative models to minimize free energy (65; 66).

Therefore, "meta-modelling" in HELOS refers to this **continuous, recursive process of discovering, validating, and composing models (S nodes) at increasing levels of abstraction**, where models at one level are built upon and explain patterns observed in the models of the level below, all guided by the universal principle of Free Energy minimisation.

3.2 Unsupervised Alphabet Induction via Lifelong FEP-driven Clustering

The foundational step of our framework, preceding the iterative reduction for token segmentation, involves the fully unsupervised, emergent categorisation of raw input characters $c \in G$ into meaningful classes (approximating alphabets, numerals, punctuation, etc.). This crucial step obviates the need for external knowledge injection, granting the system true autonomy in discovering the basic typology of the symbols it encounters. We hypothesise that this categorisation arises from a self-organising process minimising Free Energy (FEP) over the relationships induced by character co-occurrence statistics.

- **Representation:** Characters $c \in G$ and a dynamic set of potential category attractors $a \in A$ (each with an identifier $i_a \in I$) are represented as nodes in a conceptual space. The primary learned parameters are the attraction potentials or log beliefs ϕ_{ca} reflecting the affinity of character c for attractor a. From these, we derive the probabilistic membership $q(z_c = a) \approx \text{softmax}(\phi_{ca})$, where z_c is the latent category variable for c. Pairwise co-occurrence frequencies N_{ij} between characters c_i, c_j are maintained and updated incrementally.
- **FEP Formulation:** The free energy F of the system's state (character memberships q and potentially attractor parameters) aims to balance:
  - Surprise / Accuracy: Minimising the surprise of observing the co-occurrence data N_{ij} given the current clustering. That is, '$-logP(N_{ij}|\{q(z_c)\}, A)$'. This term favours clusterings where frequently co-occurring characters c_i, c_j have high probabilities of belonging to the *same* attractor ($q(z_i = a) \approx q(z_j = a)$ for some a).
  - Complexity: Minimising the complexity of the clustering model. This includes:
    1. A penalty on the number of active attractors $|A|$ (favouring simpler categorisations).
    2. A penalty on the uncertainty (entropy) of character memberships $q(z_c)$, favouring confident assignments.
    3. (Optional: Topological Complexity) Penalties derived from topological invariants (e.g., number of connected components β_0 in the graph weighted by $q(z_c = a)q(z_{c'} = a)$) of the emergent cluster structure, potentially favouring well-separated, internally coherent clusters.
- **Dynamics and Learning:** The system continuously updates the attraction potentials ϕ_{ca} via an FEP-minimising dynamic (approximated, e.g., by gradient descent or variational message passing based on local co-occurrence statistics). Characters frequently appearing together will mutually reinforce their attraction to the same attractor(s), while characters rarely co-occurring will be pushed towards different attractors due to the implicit competition and complexity penalties. Attractors themselves may be dynamically created or pruned based on global FEP considerations.
- **Output:** This self-organising process converges to a state where characters are associated with specific attractor nodes $a \in A$. The identifier i_a of the dominant attractor for a character c serves as its emergently discovered category label. This allows the system to autonomously derive classes akin to 'Cyrillic', 'Latin', 'Punctuation', 'Digit' etc., solely from the statistics of the input stream. Our experiments (Section 5) demonstrate that this emergent categorisation successfully separates distinct alphabets and character classes, providing the necessary foundation for the subsequent hierarchical segmentation and morphological analysis stages (Sections 4.3, 4.4) without relying on external annotations.

3.3 The Meta-Modelling Cycle: Pattern, Abstract, Reduce

At any given level of analysis, the system processes a sequence of entities, $l_k = (e_1, e_2, ..., e_p)$, where the entities e_j are initially basic elements (e.g., attributed characters) or, subsequently, abstract nodes S representing models discovered at lower levels. The core cycle involves:

1. **Pattern Identification:** The sequence l_k is scanned to identify statistically significant or predictively salient local patterns. This relies on analysing binary relations (edges) $\langle e_j, e_{j+1} \rangle$ between adjacent entities. Recognised patterns might include homogeneity (runs of entities with the same type/model), regular alternation, or other learned sequence motifs.
2. **Abstraction & Modelling:** When a pattern P is identified over a subsequence l'_{sub}, the system abstracts this pattern by creating (or retrieving from memory M) a new model node $s_{new} \in S$. This node, typically formed as (l'_{sub}, i_P) using the $L \times I$ constructor of our universal type S (Eq. 4), encapsulates the subsequence l'_{sub} and assigns it the identifier i_P of the discovered pattern type P. This identifier i_P acts as the emergent name for the model representing pattern P. The parameters associated with s_{new} in memory M (e.g., its log_prior) are updated based on this instance.
3. **Reduction:** The sequence l_k is transformed into a new, shorter, and more abstract sequence l_{k+1} by replacing the occurrence(s) of the subsequence l'_{sub} with the corresponding abstract model node s_{new}.

This "Pattern → Abstract/Model → Reduce" cycle is the fundamental computational step of the meta-modelling engine (8).

3.4 Emergence of Names and Local Models

Crucially, the "names" or identifiers (i_P, i_{model}, i_{attr}) associated with patterns and structures are not predefined linguistic labels.
They are emergent internal identifiers generated by the system to label statistically robust and/or predictively useful clusters or sequential patterns discovered in the data.
For instance, as illustrated in Figure 1, categories like ‘Lat‘ (Latin) or ‘Common‘ (shared symbols) are hypothesised to emerge as numeric, non externally predefined, names for topologically dense clusters of co-occurring characters in an ideal unsupervised setting.
Similarly, sequence models like ‘LatSeq‘ emerge as names for the recurring pattern of adjacent ‘Lat‘ nodes. These emergent names/models become the building blocks for the next level of analysis.
Note that categories (clusters) such as Lat and Common should appear emergently as "names" for topologically dense clusters of alphabets, shared elements between alphabets, and so forth. This is the very first thing an AI — and possibly the brain — should do when learning a language: to identify primary named clusters, which in turn provides an automatic primary segmentation.

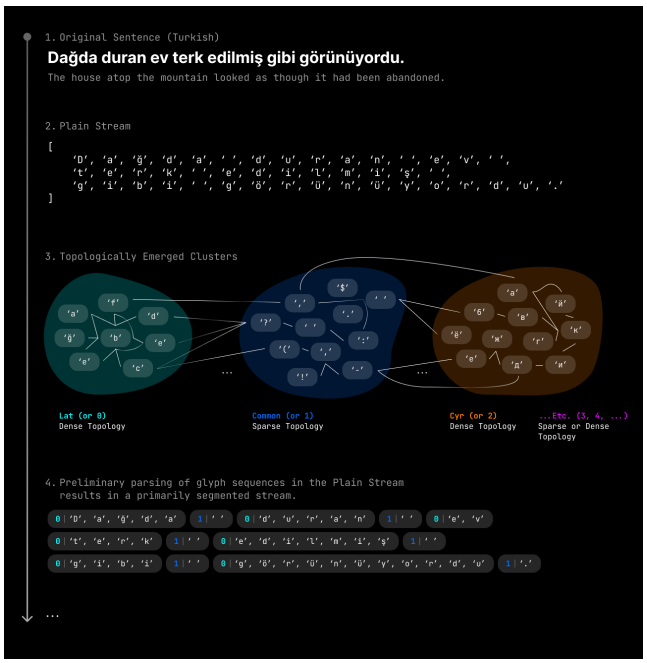


Figure 2: Online segmentation of text flow based on the emergence of topologically dense named glyph clusters.

3.5 Recursive Application and Hierarchy

The power of the meta-modelling engine lies in its recursive application. The output sequence l_{k+1}, composed of more abstract nodes than l_k, becomes the input for the *exact same* "Pattern → Abstract/Model → Reduce" cycle. This allows the system to build a deep hierarchy:

- Characters are grouped into models of character sequences (tokens/words).
- Tokens/words are grouped based on internal structure (e.g., discovered morphemes) or external distribution into parts-of-speech models (hypotheses).
- Sequences of part-of-speech models are analysed to discover models of grammatical relations and phrase structures.
- ...and so forth, iteratively building more abstract representations until potentially the entire text is represented by a single top-level node or a stable configuration is reached.

3.6 Hypothesis Generation and Optimisation

The process is not necessarily deterministic. At both the pattern identification and abstraction stages, multiple interpretations or segmentation possibilities may arise, especially with ambiguous or noisy input. The system handles this by generating multiple competing hypotheses (e.g., different possible binary parse trees for a word, as discussed in Section 3.6). Each hypothesis T is evaluated using the FEP-inspired objective function $F(T, w)$ (Eq. 5), balancing structural complexity and predictive accuracy. The system maintains and updates a distribution over these hypotheses, allowing context from higher levels to disambiguate choices made at lower levels (regarding top-down feedback). This inherent handling of multiple hypotheses also provides robustness against errors or novel inputs, allowing the system to consider alternative interpretations rather than fail outright.
This emergent, hierarchical, hypothesis-driven metamodelling provides a powerful engine for unsupervised structure discovery, forming the computational core of our approach to language analysis.

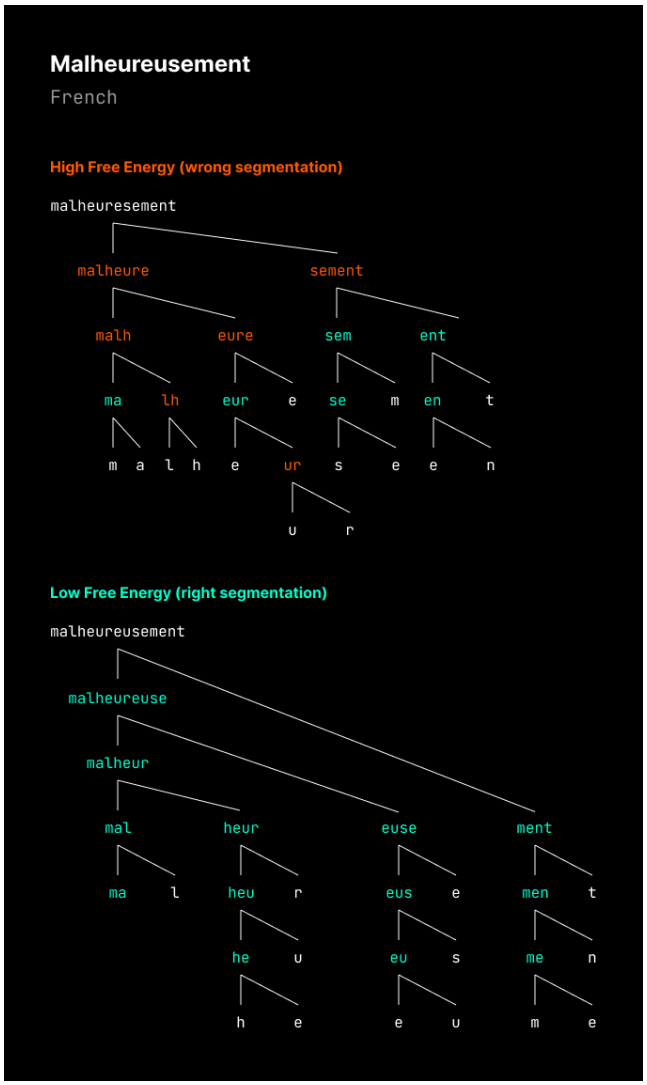


Figure 3: The image illustrates how minimising Free Energy leads to the selection of a linguistically correct morphemic analysis in Catalan space, i.e., an analysis predicted with the minimal free energy value across all predictors.

4 Proposed Framework

4.1 Formal Foundations of the Framework

This section provides the formal definitions for the core mathematical objects and concepts used throughout our framework. We primarily employ the language of set theory and category theory (specifically, the category Set of sets and functions, or a suitable category with finite products, coproducts, and fixed points for recursive type definitions).

4.1.1 Identifiers (I)

Let $I = \mathbb{N}$ be the countably infinite set of identifiers. These identifiers serve as unique labels for primitive entities (graphemes, initial attributes like emergent names) and for abstract models (representing patterns, morphemes, etc.) discovered by the system.

4.1.2 The Universal Recursive Structure (S) with Inherent Reduction

The central data structure S represents nodes within the dynamically constructed hierarchical parse trees. It embodies not only structure but also an inherent drive towards representational simplicity, akin to minimising Kolmogorov complexity (25). Formally, S is defined via the isomorphism:

$$S \cong (I \times I) + (\mathbf{L}(I + S) \times I) + (S \times I) \tag{4}$$

where I is the set of identifiers, and $L(Y) \cong 1 + ((I + S) \times L(Y))$ represents heterogeneous lists. An object $s \in S$ is thus a pair where the second element is always an identifier $i \in I$. The crucial aspect lies in the interpretation and formation of nodes constructed via the first two terms:

- Base Pair ($I \times I$): $s = (i_1, i_2)$. Represents the most basic attributed element. Its formation is direct.
- Attributed Structure ($S \times I$): $s = (s', i_{attr})$. Represents structure s' tagged with i_{attr}. Its formation is direct.
- List Model ($L \times I$): This constructor has an active interpretation. When a node s_{new} is initially formed as $(l, i_{placeholder}) \in L(I + S) \times I$, where $i_{placeholder}$ indicates an unresolved model, the node s_{new} internally triggers a reduction process on the list l. This process, denoted $Reduce : L(I + S) \to S$, iteratively applies pattern abstraction rules to l until a stable, maximally reduced representation (a single node $s_{reduced} \in S$ or a simplified list l_{final}) is reached. The goal of *Reduce* is to find the most compressed representation of the sequence l by maximally reusing existing high-frequency nodes (models) from the shared memory M (approximating Kolmogorov complexity minimisation (25)). The final identifier i_{model} associated with the *result* of the reduction (e.g., the identifier of the top-level model found for l, or the identifier of $s_{reduced}$) then replaces $i_{placeholder}$ in the node, yielding the final stable state $s_{final} = (l_{reduced}, i_{model})$.

This active reduction property means that nodes of type $L \times I$ inherently strive to simplify their internal list representation by identifying and abstracting known patterns, effectively embedding the core logic of the iterative reduction cycle within the data structure itself.

To illustrate the structure S, consider the following examples corresponding to its three constructor types, using a notational convention `<RP Head, Tail>` where type prefixes (`‘I:‘, ‘S:‘, ‘L:‘, ‘E:‘`) clarify the components:

1. Base Pair (‘I × I‘ type): Represents the initial attribution of a base identifier (e.g., a grapheme) with another identifier (e.g., a basic attribute like Script). Let i_{graph} be the identifier for grapheme ’a’ and i_{Lat} be the identifier for the ’Latin’ script. Their combination is an object $s_1 \in S$:

   ```
   s_1 = <RP I:i_graph, I:i_Lat>
   ```

   This node represents the attributed grapheme ’a-as-Latin’.

2. Attributed Structure (‘S × I‘ type): Represents applying a further attribute or tag to an *existing* structure $s' \in S$. Let s_1 be the node from the previous example, and let i_{Ll} be the identifier for the ’Lowercase Letter’ category. Attributing s_1 with this category yields a new node $s_2 \in S$:

   ```
   s_2 = <RP S:<RP I:i_graph, I:i_Lat>, I:i_Ll>
   ```

   This node represents ’a-as-Latin-and-Lowercase’. The first element is explicitly marked as being of type S.

3. List Model (‘L(I+S) × I‘ type): Represents the abstraction of a sequence (a list l) under a model identifier. Let l be a heterogeneous list containing a base identifier i_{graph2} and the previously constructed node s_2. Let i_{LatSeq} be the identifier for the "Latin Sequence" model discovered by the system. The resulting node $s_3 \in S$ is:

   ```
   s_3 = <RP L:[ E:<I:i_graph2>, E:<S:<RP S:<RP I:i_graph, I:i_Lat>, I:i_Ll>> ], I:i_LatSeq>
   ```

   Here, ‘L:[...]‘ denotes the list (an object of type $L(I + S)$), and `E:<...>` denotes elements within that list (type $E = I + S$). This node s_3 represents the sequence corresponding to ’ba’ (if i_{graph2} is ’b’) being recognised as an instance of the ‘LatSeq‘ model.

These examples demonstrate how the universal recursive structure S allows for the consistent, binary, and hierarchical encoding of attributed elements, abstract sequence models, and nested combinations thereof, using only identifiers and pairing.

4.1.3 Parse Trees (T)

A parse tree T for a sequence of grapheme identifiers $w = (g_1, ..., g_n)$, $g_k \in I$, is an object of type S such that its leaves, obtained via a recursive traversal function $get\_leaves : S \to List(I)$, correspond exactly to the sequence w. The set of all valid parse trees for w is denoted $\mathcal{T}_w$. The structure of T reflects a specific binary hierarchical composition of w.

4.1.4 Memory / Node Registry (M)

The memory M stores the system's knowledge about encountered structural nodes (sub-trees $t \in S$). It can be formalised as a mapping (or a state in a state monad) $M : \mathcal{H}(S) \to \Theta$, where $\mathcal{H}(S)$ is the set of unique identifiers (e.g., hashes) for nodes $t \in S$, and Θ represents the space of node parameters. In our implementation, $\theta \in \Theta$ includes:

- $p(t)$: The estimated prior probability of node t (or its logarithm, `log_prior`). This reflects the simplicity/complexity component.
- θ_{pred}: Parameters for the predictive models associated with t (`predictor_next`, `predictor_internal`), used to calculate surprise.

4.1.5 Free Energy Functional (F)

The objective function guiding the inference of the optimal parse tree T^* is the (Variational) Free Energy $F : T \times M \to \mathbb{R}^+$. It approximates the negative log evidence of the data given the model implicit in the tree and memory. Following (22; 9), it balances complexity and surprise:

$$F(T, w; M) \approx \underbrace{\text{Complexity}(T; M)}_{\approx \sum_{N_k \in Sub(T)} [-\log p(N_k; M)]} + \underbrace{\text{Surprise}(w|T; M)}_{\approx \sum_j [-\log p(d_j|\text{model}_j; M)]} \tag{5}$$

where $p(N_k; M)$ is the prior probability derived from the memory state M, and the surprise term sums local prediction errors based on the predictive models also stored in M. Minimising F corresponds to finding the simplest explanation that accurately predicts the data (9).

4.1.6 Inference (FindBestTrees)

Inference is the process of finding the parse tree T^* that minimizes F (67). Due to the size of $\mathcal{T}_w$, we use an approximate inference algorithm, $FindBestTrees(w, M, k)$, which returns a list of the top k hypotheses (trees) with the lowest approximate FE values found via a guided search (e.g., FEP-guided beam search (68)).

4.1.7 Learning (UpdateMemory)

Learning is the process of updating the memory state M based on the inference results for observed data w. Given the best hypothesis T^* (or a distribution over hypotheses), the parameters θ_k for nodes $N_k \in Sub(T^*)$ are updated. This update, $UpdateMemory(M, T^*, w)$, follows Bayesian principles (approximated), adjusting node priors $p(N_k)$ and predictive model parameters $\theta_{pred,k}$ to reduce the Free Energy $F(T^*, w)$, effectively minimizing prediction errors encountered during the parse.

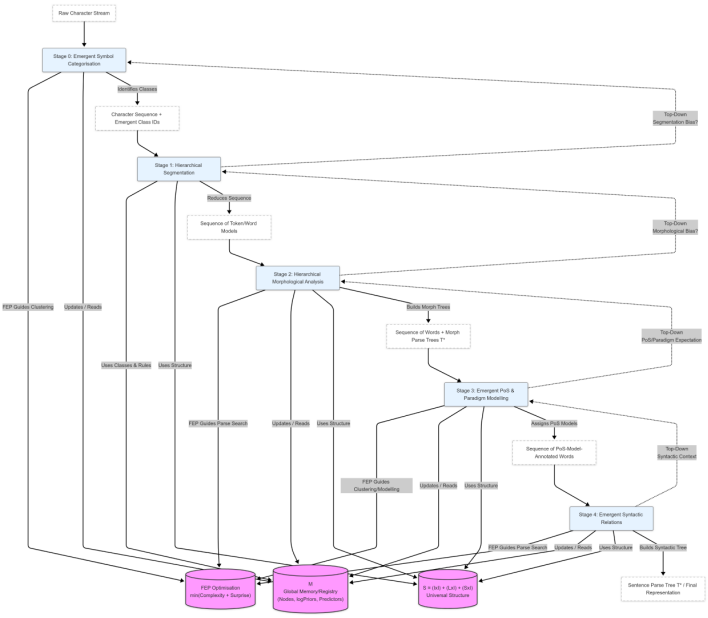
4.1.8 Synthesis of Formalisms

The theoretical framework presented herein integrates concepts from multiple domains into a cohesive system where each component plays a distinct, complementary role.

The Free Energy Principle (FEP) provides the overarching *optimisation objective* (Eq. 5), explaining *why* specific hierarchical structures T^* (composed of nodes S) emerge as optimal representations – they minimise the trade-off between model complexity and predictive surprise inherent in processing linguistic data w.

The universal recursive type S (Eq. 4), formally defined using category theory (as an initial algebra) or type theory, furnishes the specific *representational format* and structural bias (binary composition) upon which the FEP operates; its mathematical properties are amenable to formal analysis using tools like structural induction.

The iterative reduction algorithm (Section 3.3), conceptualised computationally as a search for the optimal parse T^* minimising $F(T, w)$, serves as the *inference and learning mechanism* that realises the FEP optimisation in practice, dynamically constructing representations of type S. Finally, linguistic principles, particularly the emphasis on binary composition found in Minimalism (cf. 'Merge'), provide the external *motivation and justification* for adopting the specific binary recursive structure of S.



These components are not merely juxtaposed but are intrinsically linked: the FEP guides the search over structures S, the algorithm implements this search, category/type theory formalises S, and linguistic theory motivates its basic form. Thus, the framework represents a principled synthesis aimed at a computationally grounded, cognitively plausible model of emergent language structure.

4.2 Overall Architecture

We propose a unified, hierarchical framework, HELOS, for unsupervised language structure discovery, conceptualised as a multi-stage process operating on raw character sequences. All stages leverage the same core principles: a universal recursive binary representation (S, Eq. 4) for linguistic structures, and an iterative mechanism driven by predictive optimisation (FEP-like) or structural compression to identify and abstract meaningful patterns.

- **Level 0: Emergent Symbol Categorisation.** This foundational stage processes the raw character stream $w_{char} = (c_1, ..., c_m)$ to autonomously discover and assign category identifiers (approximating alphabets, punctuation, digits, etc.) to each unique character c_k. As outlined in Section 3.1, this is achieved via a self-organising, FEP-driven clustering process. Based on character co-occurrence statistics, the system identifies attractor nodes $a \in A$ corresponding to coherent character classes and determines the association (e.g., dominant probability $q(z_c = a)$) of each character c to these emergent categories, yielding category identifiers $i_{class\_k} \in I$. This crucial step provides the necessary initial classification for subsequent structural analysis without relying on external knowledge.

- **Level 1: Hierarchical Segmentation and Tokenisation.** Taking the sequence of characters now implicitly or explicitly annotated with their emergent category identifiers i_{class_k} as input, this stage transforms it into a sequence of higher-level word-like models (tokens). This is achieved through an iterative compositional reduction process. This mechanism focuses on identifying and abstracting recurring patterns in the sequence of *category identifiers* (e.g., homogeneity like 'ClassA-ClassA-ClassA' or alternation like 'ClassA-ClassB-ClassA') using primarily structural compression principles (akin to minimising description length) rather than full FEP optimisation at this stage. The output is a sequence $l_{final} = (Node_1, ..., Node_p)$, where each $Node_j \in S$ represents a discovered segment (token).

- **Level 2: Hierarchical Morphological Analysis.** Each segment model $Node_j$ identified in Stage 1, representing a putative word via its encapsulated grapheme sequence w_{graph}, is then subjected to deeper structural analysis (detailed in Section 4.4). This stage employs the FEP-guided search for the optimal hierarchical binary parse tree T^* within the Catalan space $\mathcal{T}_{w_{graph}}$. The objective is to minimise the Free Energy $F(T, w_{graph})$ (Eq. 5), leading to the emergent discovery of constituent morphemic structures (nodes N_k corresponding to roots and affixes).

- **Stage 3+ (Future Work in Terms of Practical Validation): Syntax, Semantics, etc.** The framework is designed to apply the same FEP-driven hierarchical meta-modelling principles recursively to the sequence of morphologically analysed word models produced by Stage 2, aiming for the emergent discovery of PoS categories, syntactic relations, phrase structures, and potentially semantic patterns (as outlined in Section 4.8).

Underpinning all stages is the universal recursive data structure S and the global memory M storing learned parameters for discovered nodes N_k. The interaction between stages, particularly the top-down feedback from higher-level analyses influencing hypothesis selection at lower levels, is mediated through the shared memory and the overarching FEP optimisation. This multi-stage, unified architecture allows for the progressive, unsupervised discovery of linguistic structure across multiple levels of granularity.

4.3 Level 0: Mathematical Formalisation of Emergent Symbol Categorisation

4.3.1 Objective

Let $I_{base} \subset I$ be the subset of identifiers corresponding to the primitive graphemes observed in the input stream w_{char}. The objective is to partition or probabilistically assign these base identifiers $i \in I_{base}$ to emergent category models, represented as abstract nodes $S_{cat} \in S$, without prior knowledge of these categories. This process discovers structures akin to alphabets or symbol classes. The resulting categorised symbols, represented by initial nodes $s^{(0)} \in S$ (e.g., of type $I \times I$, pairing a grapheme identifier with its emergent category identifier), form the input sequence l_0 for the Level 1 segmentation process (Section 4.4).

4.3.2 Dynamic Co-occurrence Representation

The system maintains a dynamic representation of pairwise co-occurrence information. While the ultimate state is stored within the parameters of nodes S in memory M, for the purpose of identifying candidate categories, we conceptualise a dynamic weighted co-occurrence map:

$$W_t : (I_{base} \cup S_{cat}) \times (I_{base} \cup S_{cat}) \to \mathbb{R}^+$$

where $S_{cat} \subset S$ is the set of currently established category nodes stored in memory M, and $W_t(u, v)$ represents the cumulative strength of adjacent co-occurrence between entities represented by nodes u and v up to time t. The nodes u, v can be either base grapheme identifiers $i \in I_{base}$ or existing category nodes $S_c \in S_{cat}$.

Update Mechanism: Upon processing an adjacent pair (s_k, s_{k+1}) from the input character stream:

1. Retrieve the current representations: $u = \Phi(s_k)$, $v = \Phi(s_{k+1})$, where $\Phi : I_{base} \to I_{base} \cup S_{cat}$ maps a base grapheme identifier to either itself (if not yet strongly associated with a category) or its primary category node S_c currently stored in M.

2. Increment the co-occurrence weight: $W_{t+1}(u, v) \leftarrow W_t(u, v) + 1$ (and symmetrically for $W_{t+1}(v, u)$). Update the internal parameters (predictors, priors) within memory M associated with nodes u and v to reflect this interaction.

4.3.3 Topological Stability Analysis for Category Emergence

To identify when a group of co-occurring base identifiers forms a stable, coherent category worthy of abstraction, Persistent Homology (PH) is employed on the co-occurrence structure.

- **Candidate Category Identification:** A candidate category $C \subseteq I_{base} \cup S_{cat}$ is identified. This could be a connected component in the graph implicitly defined by W_t, filtered to contain a significant number of base identifiers ($|C \cap I_{base}| \geq k_{min}$). Heuristics, potentially guided by node density or modularity within the co-occurrence graph, select promising candidates C^* for stability analysis.

- **Simplicial Complex and Filtration:** For a candidate C^*, construct a simplicial complex K_{C^*} where vertices correspond to the nodes $u \in C^*$. 1-simplices (edges) (u, v) are included if $W_t(u, v) \geq w_{min}$. A filtration f is defined on K_{C^*}, assigning lower filtration values to edges with higher co-occurrence weights, using a stable function independent of the maximum weight within C^*:
  - $f(u) = f_0$ (e.g., -0.01) for a vertex $u \in C^*$.
  - $f((u, v)) = \max(f_0 + \epsilon, W_{max} - W_t(u, v))$ for an edge (u, v).

- **Persistence Calculation:** Compute the persistence diagrams $PD_p(K_{C^*})$ for $p = 0$ (connectivity) and $p = 1$ (cycles).

4.3.4 Stability Criterion and FEP Interpretation

A candidate category C^* is deemed stable if its topological signature remains consistent over a recent history window (length h), signifying that its internal structure of co-occurrence has converged. Let $PD_p(t-k)$ be the p-diagram for C^* at time $t-k$.

- **Dimension 0 Stability:** The number of finite persistence points $N_{fin}(\tau) = |\{(b,d) \in PD_0(\tau) | d < \infty\}|$ exhibits low variance: $Var[N_{fin}(t), ..., N_{fin}(t-h+1)] < \theta_{var}$.
- **Dimension 1 Stability:** The Bottleneck distance between diagrams of finite points is bounded: $d_B(PD_1(t)^*, PD_1(t-k)^*) < \theta_{topology}$ for $k = 1, ..., h-1$.

FEP Interpretation: Achieving TDA stability serves as a robust heuristic indicating that abstracting the candidate C^* into a single category node S_{new_cat} is likely favourable under the Free Energy Principle. Stabilisation implies that the relationships within C^* are predictable and consistent. Representing C^* as a single node S_{new_cat} drastically reduces model complexity (replacing many nodes and edges with one node S_{new_cat} which will acquire a high prior $p(S_{new\_cat})$ via `UpdateMemory`). The topological stability suggests this abstraction can be done without incurring excessive surprise (prediction error), as the internal configuration is reliable. Thus, TDA stability identifies opportune moments for FEP-minimising abstraction at this foundational level.

4.3.5 Hierarchical Symbolisation within HELOS Memory

When a candidate category C^* is deemed stable:

1. Create New Node: A new abstract category node $S_{new\_cat} \in S$ is created and assigned a unique identifier $i_{new\_cat} \in I$. Its structure might initially be simple, e.g., $(i_{placeholder}, i_{new\_cat})$ of type $I \times I$, or more complex if derived from a pattern involving existing categories. This node S_{new_cat} is added to the memory M with initial parameters derived from the properties of C^* (e.g., initial `log_prior` based on frequency/stability, internal predictors reflecting aggregated co-occurrences within C^*).
2. Update Symbol Mapping (Φ): For all base identifiers $i \in C^* \cap I_{base}$, update the mapping $\Phi(i) = S_{new\_cat}$ (or store the association $i \mapsto i_{new\_cat}$ within M). Update the cluster definition registry $\Delta(i_{new\_cat}) = C^* \cap I_{base}$.
3. Update Memory (M): Co-occurrences involving nodes $u \in C^*$ are now interpreted as interactions involving S_{new_cat}. The 'UpdateMemory' process refines the parameters of S_{new_cat} and its neighbours in M. Specifically:
   - External co-occurrences $W_t(u, v)$ where $u \notin C^*$ and $v \in C^*$ contribute to updating the predictive models linking u and S_{new_cat}.
   - Internal co-occurrences $W_t(v_1, v_2)$ where $v_1, v_2 \in C^*$ contribute to refining the internal predictors and potentially the `log_prior` of S_{new_cat}.
   - The raw co-occurrence map W itself becomes less central as information migrates into the parameters within M.

4.3.6 Lifelong Emergence

The cycle of observing co-occurrences, updating M, identifying candidates, checking TDA stability, and potentially creating new category nodes S_{cat} continues throughout the system's lifetime. Established category nodes S_{cat} participate in co-occurrence calculations alongside base identifiers, allowing for the discovery of relationships between categories (e.g., 'Punctuation' frequently follows `CyrillicCategory`) and potentially hierarchical category structures. This mechanism provides the Level 0 emergent symbol categorisation required by the HELOS framework. The resulting category assignments for base identifiers provide the initial annotations for the Level 1 segmentation process.

4.4 Level 1: Hierarchical Segmentation buit upon Level 0

The initial stage of structural abstraction within HELOS addresses the fundamental task of segmenting the input character stream w_{char} into meaningful, word-like units or tokens. This process operates on the sequence $l_0 = (s_1^{(0)}, ..., s_m^{(0)})$ where each node $s_k^{(0)}$ represents an input character c_k already implicitly or explicitly annotated with its emergently discovered category identifier i_{class_k} (e.g., representing 'Cyrillic', 'Latin', 'Punctuation') derived from the foundational clustering process described in Section 4.3. The segmentation is achieved not through predefined delimiters but via the iterative application of the compositional reduction mechanism, focusing here on patterns within the sequence of these emergent category identifiers.

4.4.1 Initial Attribution and Node Creation

As before, the input $w_{char} = (c_1, ..., c_m)$ is lifted into an initial sequence $l_0 = (s_1^{(0)}, ..., s_m^{(0)})$, where each $s_k^{(0)}$ is a node of type S representing the character c_k attributed with its derived Script and Category identifiers (e.g., $s_k^{(0)} = ((i_{grapheme_k}, i_{script_k}), i_{category_k})$ constructed via the $I \times I$ and $S \times I$ components).

4.4.2 Segmentation through Pattern Modelling and Reduction

The system then aims to group this initial sequence l_0 into larger segments. This is achieved by identifying contiguous subsequences l'_{sub} that form recognisable patterns based on the attributes of the nodes $s_k^{(0)}$. Instead of an external iterative loop applying reduction rules, the process can be viewed as follows:

1. **Identify Potential Segment:** Scan l_0 to find a maximal subsequence l'_{sub} matching a known sequence pattern P (e.g., homogeneity of Script 'Cyr', detected by analysing adjacent node attributes).
2. **Create Active List Node:** Form a new node $s_{segment} = (l'_{sub}, i_P)$ using the $L \times I$ constructor, where i_P is the identifier for the pattern model (e.g., i_{CyrSeq}). Crucially, this node $s_{segment}$ now contains the list l'_{sub} in its 'head'. *(Self-reduction is less applicable here as the list contains attributed characters, not higher models yet. The key is abstracting the sequence into ONE node).*
3. **Replace Sequence:** Replace the subsequence l'_{sub} in the main representation with the single node $s_{segment}$.
4. **Repeat:** Repeat steps 1-3 on the now shorter sequence until no more known sequence patterns (like 'CyrSeq', 'CommonSeq') can be applied to group adjacent nodes.

The output is a final sequence $l_{final} = (Node_1, Node_2, ..., Node_p)$, where each $Node_j$ is an object S of type $L \times I$, representing a fully segmented word or token (e.g., `Node_j` = (l_chars_for_word, i_WordModel)'), whose internal list $l_{chars\_for\_word}$ contains the attributed character nodes. This sequence l_{final} is then passed to the next stage.

It is noteworthy that while the Free Energy Principle provides the overarching optimisation framework, particularly crucial for the probabilistic hypothesis selection required in morphological and syntactic parsing (Level 2+), the initial segmentation process (Level 1) can potentially be achieved through a more deterministic iterative reduction mechanism focused on finding the most compact representation by identifying and abstracting recurring structural patterns based on character attributes. This structural compression implicitly relates to minimising Kolmogorov complexity, a component of the more general FEP objective. The FEP becomes essential when dealing with the combinatorial ambiguity of hierarchical parsing within segments and learning the predictive value of emergent morphemes.

4.4.3 Output: Sequence of Token Nodes

The iterative reduction process, conceptualised as reaching a fixed point l_{final} of the reduction endomorphism reduceStep_M on the category of lists $L = \mathbf{L}(I + S)$, yields the primary output of the segmentation stage. This output, l_{final}, is itself an object of type L, representing the final, maximally abstracted sequence derived from the initial input l_0.

$$l_{\text{final}} = \text{Encode}(l_0) = \text{fix}(\text{reduceStep}_M)(l_0)$$

where l_0 is the initial sequence of attributed character nodes.

Due to the nature of the reduction rules, which primarily abstract contiguous sequences of characters sharing script/category properties into single nodes (models like 'CyrSeq', 'CommonSeq', etc.), the resulting sequence l_{final} typically takes the form:

$$l_{\text{final}} = (\text{Node}_1, \text{Node}_2, ..., \text{Node}_p)$$

where each $\text{Node}_j \in S$ represents a discovered segment corresponding to a putative word or token.

Crucially, each Node_j in this final sequence retains the full hierarchical history of its construction within its structure. If Node_j was created by abstracting a subsequence $l''_{\text{sub}} \in L$ using a model identifier $i_{\text{Model}} \in I$, its internal structure is $\text{Node}_j = (l''_{\text{sub}}, i_{\text{Model}})$, conforming to the $(L \times I)$ component of the isomorphism $S \cong (L \times I) + (S \times I)$. The original sequence of grapheme identifiers w_{graph} for the token represented by Node_j can be precisely recovered by applying a recursive "unparsing" or "flattening" function (a catamorphism over the structure S and L) that traverses the nested pairs down to the base identifiers $i_{\text{grapheme}_k} \in I$ contained within the initial nodes $s_k^{(0)}$.

This final sequence of token nodes $l_{final} \in L$ serves as the structured input to the subsequent morphological analysis stage, where each $Node_j$ is processed individually to determine its internal morphemic structure. The segmentation process thus transforms a flat character sequence into a structured sequence of abstract, hierarchically-defined tokens, ready for deeper linguistic analysis.

4.5 Level 2: Hierarchical Morphological Analysis

The second stage of the framework takes as input the individual token nodes $\text{Node}_{\text{word}} \in S$ produced by the segmentation stage. Each $\text{Node}_{\text{word}}$ encapsulates a sequence of grapheme identifiers $w_{\text{graph}} = (g_1, \ldots, g_n)$, $g_k \in I$, recovered via a recursive traversal (catamorphism) of its internal structure. The goal of this stage is to uncover the latent hierarchical morphemic structure within w_{graph} by finding the optimal binary parse tree T^* that best explains this grapheme sequence according to the Free Energy Principle.

4.5.1 The Hypothesis Space: Catalan Trees ($\mathcal{T}_w$)

For a given grapheme sequence w_{graph} of length n, the space of all possible hierarchical binary interpretations corresponds to the set $\mathcal{T}_{w_{\text{graph}}}$ of all full binary trees with n leaves labelled by $g_1, \ldots, g_n$. The internal nodes of any tree $T \in \mathcal{T}_{w_{\text{graph}}}$ represent putative morphemic constituents or intermediate structures, formed by the binary composition operation inherent in our universal structure S. The cardinality of this hypothesis space is given by the $(n-1)$-th Catalan number (21), $|\mathcal{T}_{w_{\text{graph}}}| = C_{n-1}$, which grows exponentially, precluding exhaustive search for longer words.

4.5.2 Optimisation Objective: Free Energy Minimisation

The core principle guiding the selection of the best parse tree T^* is the minimisation of the Free Energy functional $F(T, w_{\text{graph}})$, approximated as previously defined (Eq. 2):

$$F(T, w_{\text{graph}}) \approx \text{Complexity}(T) + \text{Surprise}(w_{\text{graph}}|T)$$

- Complexity(T): This term represents the cost or implausibility of the proposed tree structure T. It is calculated as the sum of complexities of all constituent nodes N_k within the tree, where the complexity of a node N_k is inversely related to its learned prior probability $p(N_k)$ (approximated, e.g., by $\text{Complexity}(N_k) \approx -\log p(N_k)$ derived from its 'log_prior' parameter stored in memory M). This term enforces Occam's razor, favouring trees composed of simpler, more frequently useful, and previously validated morphemic units (nodes N_k with high $p(N_k)$ / high `log_prior`).

- Surprise($w_{\text{graph}}|T$): This term quantifies how poorly the tree T predicts or explains the actual grapheme sequence w_{graph}, corresponding to the negative log-likelihood $-\log P(w_{graph}|T)$ under the generative model implicitly defined by the tree and the node predictors. It is computed recursively during the construction (or evaluation) of the tree T. Let $S(N)$ denote the surprise associated with the subtree rooted at node N.

    1. For a leaf node $N = g_k$ (representing a grapheme identifier), the surprise is typically considered zero or a base value, as it represents the data itself at that point: $S(g_k) = 0$.

2. For an internal node $N_{new} = (N_{left}, N_{right})$ formed by combining left subtree N_{left} (spanning $w[i..k]$) and right subtree N_{right} (spanning $w[k+1..j]$), the total surprise is the sum of the surprises from the children plus the surprise incurred at this specific composition step:

$$S(N_{new}) = S(N_{left}) + S(N_{right}) + \Delta S(N_{left}, N_{right} \to N_{new}) \tag{6}$$

The incremental surprise ΔS quantifies the prediction errors at this merge step, approximated in our current implementation as:

$$\Delta S \approx \frac{1}{2} \left(\text{Surprise}_{next}(N_{right}|N_{left}) + \text{Surprise}_{internal}(N_{left}, N_{right}|N_{new})\right) + C_{pair} \tag{7}$$

where $\text{Surprise}_{next} \approx -\log p(N_{right}|N_{left}, \theta_{left\_pred})$ is the surprise from the left child's predictor (`predictor_next`) failing to predict the right child, $\text{Surprise}_{internal} \approx -\log p(N_{left}, N_{right}|N_{new}, \theta_{new\_pred})$ is the surprise from the new parent node's predictor ('predictor_internal') failing to predict its own constituents, and C_{pair} is a small penalty (`COMPLEXITY_PENALTY_PAIR` in the code) for the composition operation itself. The total surprise for the complete tree T rooted at N_{root} is then $Surprise(w_{graph}|T) = S(N_{root})$.

The optimal parse T^* is the one that achieves the best trade-off between accurately explaining the grapheme sequence (low Surprise) and utilising simple, probable structural components (low Complexity).

$$T^* = \arg \min_{T \in \mathcal{T}_{w_{\text{graph}}}} [\text{Complexity}(T) + \text{Surprise}(w_{\text{graph}}|T)]$$

4.5.3 Search and Inference Algorithm

Due to the combinatorial size of $\mathcal{T}_{w_{\text{graph}}}$, finding T^* requires an efficient search or inference algorithm. Instead of generating all C_{n-1} trees, we employ an approximate inference method, such as the FEP-guided beam search sketched in the prototype implementation (`find_best_parses_fep`). This algorithm explores the space of possible parse trees iteratively:

1. It maintains a beam (68) (priority queue) of the top k most promising partial parse tree hypotheses, ranked by their current estimated Free Energy F.
2. It expands hypotheses by considering binary compositions of adjacent sub-hypotheses.
3. Crucially, the selection of which compositions to explore and retain is guided by the predictive models associated with the nodes and the resulting local FE calculation (balancing complexity and surprise). Hypotheses leading to high prediction error or involving very low-probability nodes are pruned early.
4. The search terminates when a complete parse covering the entire sequence w_{graph} is found with the lowest FE among the explored hypotheses, or when search resources are exhausted.

This search process effectively navigates the Catalan space, guided by the learned statistics and predictive models embedded within the nodes N_k in the registry M, to find a likely candidate T^*_{approx} for the true optimal parse T^*.

4.5.4 Output: Morphemic Parse Tree(s)

The output of this stage for a given input token $\text{Node}_{\text{word}}$ is one or more best-scoring parse trees T^*_{approx} found by the search algorithm. Each T^*_{approx} represents a hierarchical binary analysis of the word's grapheme sequence, where the internal nodes correspond to the **emergently discovered morphemic constituents**. Optionally, a distribution over the top k hypotheses can be retained to represent ambiguity, allowing downstream processes (e.g., syntax) to potentially resolve it using wider context.

4.6 Level 3: Emergent Part-of-Speech and Paradigm Modelling

Building upon the sequence of word-representing nodes $l_{final} = (Node_1, ..., Node_p)$ produced by Level 1 (Segmentation) and the associated morphological parse hypotheses T^*_k (or distributions over them) for each $Node_k$ derived from Level 2 (Morphology), the framework proceeds to discover higher-level grammatical categories corresponding to parts-of-speech (PoS) and their associated inflectional paradigms (69; 38). This is achieved by recursively applying the same core meta-modelling engine (Section 3) to the sequence l_{final}, now operating on representations enriched with morphological information.

4.6.1 Input Representation: Morphologically Aware Word Nodes

The input to this stage is the sequence l_{final}. Each element $Node_k \in S$ in this sequence now implicitly carries information about its most likely morphological parse(s) T_k^*, obtained by minimising $F(T, w_k)$ at Level 2. This information might be represented by:

- The optimal parse tree T_k^* itself.
- A set of features extracted from T_k^* (e.g., identified root morpheme ID i_{root}, set of affix morpheme IDs $\{i_{affix}\}$, final ending ID i_{ending}).
- A pointer to the belief state over morphological parses maintained by the system for that word form.

Let $\mathcal{M}(Node_k)$ denote the morphological information associated with node k.

4.6.2 Pattern Recognition: Morpho-distributional Clustering

The system analyses the sequence l_{final} to identify groups of word nodes ($Node_k$) that exhibit similar behaviour, driven by FEP minimisation. "Similar behaviour" encompasses both:

1. Internal Morphological Similarity: Nodes sharing common structural patterns in their morphological parses T_k^* (e.g., similar root structures, common affixes, particularly inflectional endings).
2. External Distributional Similarity: Nodes appearing in similar sequential contexts within l_{final} (e.g., nodes frequently preceded by determiners or followed by verbs).

This process can be viewed as unsupervised clustering of the nodes $Node_k$ based on a combined morpho-distributional feature space. The objective is to partition (or softly assign) nodes into emergent categories (putative PoS classes) $Cl_{PoS}^{(1)}, Cl_{PoS}^{(2)}, ...$ such that the overall Free Energy is minimised.

4.6.3 Emergent PoS Models and Paradigm Structures

For each emergent cluster $Cl_{PoS}^{(c)}$, the system induces a Part-of-Speech Model, represented by a new abstract node $Node_{PoS}^{(c)} \in S$ with identifier $i_{PoS}^{(c)} \in I$. This node encapsulates the shared properties of the words in the cluster. Crucially, beyond just a category label, this node learns to represent the inflectional paradigm associated with the PoS class:

- Paradigm Representation ('paradigm _structure'): This can be formalised as a probabilistic mapping or a structured representation (e.g., another embedded tree or factor graph) within $Node_{PoS}^{(c)}$. This structure maps abstract grammatical features (e.g., {Case=Genitive, Number=Plural}) to the specific morpheme node identifiers ($i_{affix} \in I$) that typically realise these features for words in this class c. $p(i_{affix}|\text{LemmaNode}, \text{GrammFeatures}, Node_{PoS}^{(c)})$. The PoS model encodes the probability of observing affix i_{affix} given a lemma (represented by its root/stem node) and target grammatical features.
- Contextual Predictor ('predictor_context'): Associated with $Node_{PoS}^{(c)}$ is a predictive model that learns the typical syntactic contexts (e.g., preceding and succeeding PoS models) in which words of this class appear.

4.6.4 FEP Optimisation at PoS Level

The formation of these PoS models and the assignment of words to them is governed by minimising an FEP functional at this level. F_{PoS} would balance:

- Complexity: Penalises having too many PoS models or overly complex paradigm structures within them. Favours assigning words to existing, simple, high-probability PoS models (nodes $Node_{PoS}^{(c)}$ with high $p(Node_{PoS}^{(c)})$).
- Surprise/Accuracy: Minimises the surprise of observing:
  - The morphological forms of words given their assigned PoS model's paradigm (e.g., how well does $Node_{NounModel}$ predict the ending '-s' for a given noun stem in the Instrumental Plural?).
  - The syntactic context of words given their assigned PoS model (e.g., how well does $Node_{NounModel}$ predict that it might be preceded by $Node_{AdjModel}$?).

4.6.5 Iterative Refinement and Top-Down Feedback

The PoS models and word assignments are learned iteratively. As the system processes more sentences:

- The parameters of PoS models (paradigm structures, contextual predictors) are refined via Bayesian-like updates based on prediction errors.
- Crucially, the learned PoS models provide top-down predictions that feed back to the morphological analysis stage (Level 2). When analysing a word, if the syntactic context strongly predicts a specific PoS (e.g., a Verb), the FEP calculation for morphological parses (Eq. 5) will be biased towards hypotheses T^* consistent with that PoS model (e.g., those containing verbal affixes), effectively resolving morphological ambiguity using syntactic context.

4.6.6 Output: Sequence of PoS-Model-Annotated Words

The output of this stage is a sequence where each original word node $Node_k$ is now potentially associated with its most probable emergent PoS model $Node^{(c)}_{PoS}$. This sequence, rich with both morphological and now categorical information, serves as the input for the next level of analysis aiming to discover syntactic phrase structure and dependency relations by applying the meta-modelling engine yet again, this time searching for patterns in sequences of PoS models.

This extension demonstrates how the same fundamental principles of FEP-driven hierarchical meta-modelling can be recursively applied to discover increasingly abstract levels of linguistic organisation, naturally integrating morphology and syntax through bidirectional predictive inference. While rigorous implementation remains future work, this theoretical outline provides a clear pathway.

4.6.7 Generativity and Compositional Creation of Novel Forms

The HELOS framework's generative capability stems directly from its hierarchical, predictive nature and the FEP/EFE optimisation principle. Generation involves finding (*ad hoc* creating) a parse tree T^*_{gen} that minimises Free Energy given a generative goal (e.g., realising a lemma with specific grammatical features). This allows for the compositional creation of novel, yet grammatically valid, forms.

Consider the task of generating the 3rd person singular, future tense form for the Turkish verb stem *gel-* ("come"), represented by the node $N_{gel} \in S$. Assume the system has previously learned:

- The emergent part-of-speech model $Node_{Verb} \in S$, associated with identifier i_{Verb}.
- The stem node N_{gel}.
- The future tense affix model $N_{ecek} \in S$ (with allomorph N_{acak}), associated with identifier i_{Fut}.
- The 3rd person singular zero ending $N_\emptyset \in S$, associated with identifier i_{3sg}.
- The paradigm model within $Node_{Verb}$, encoded as conditional probabilities within its internal predictors, specifying high probability (low surprise) for composing a verb stem with $N_{ecek/acak}$ to realise '[Tense=Future]', and subsequently with $N_\emptyset$ for '[Person=3, Number=Singular]'. Let $p_{paradigm}(N_{affix}|N_{stem}, Features)$ denote this probability.

The generative goal defines preferred features $Features_{target}$ = {Tense=Future, Person=3, Num=Sing} for the lemma N_{gel}. The system seeks T^*_{gen} minimising the generative Free Energy:

$$F_{gen}(T) \approx \sum_{N_k \in Sub(T)} [-\log p(N_k)] - \log p_{paradigm}(T|N_{gel}, Features_{target})$$

The top-down generation process, guided by minimising F_{gen} (or maximising the posterior probability $p(T|...) \propto \exp(-F_{gen})$), proceeds compositionally:

1. **Activate Goal:** Activate N_{gel} and $Features_{target}$.

2. **Predict Affix 1 (Tense):** The paradigm model $p_{paradigm}(\cdot|N_{gel}, \{\text{Tense=Future,...}\})$ assigns high probability (low surprise) to the future tense affix. Due to vowel harmony rules implicitly learned by the predictors associated with N_{gel} or N_{Verb}, the allomorph N_{ecek} (identifier i_{Fut}) is selected over N_{acak}.

3. **Form Intermediate Structure:** A node $N_{gel\_ecek} = (N_{gel}, i_{Fut})$ is hypothesised. Its complexity $-\log p(N_{gel\_ecek})$ (assuming this combination has been seen or is predicted by a meta-rule) and the low surprise $-\log p(N_{ecek}|N_{gel}, ...)$ contribute to a low intermediate F.

4. **Predict Affix 2 (Person/Number):** The paradigm model $p_{paradigm}(\cdot|N_{gel\_ecek}, \{\text{Person=3, Num=Sing}\})$ predicts the zero morpheme $N_{\emptyset}$ (identifier i_{3sg}) with high probability.

5. **Form Final Tree:** The final hypothesis is $T^*_{gen} = (N_{gel\_ecek}, i_{3sg})$. Its total $F(T^*_{gen})$ is low because it's composed of high-probability nodes $(N_{gel}, i_{Fut}, i_{3sg}, N_{gel\_ecek})$ combined according to high-probability paradigmatic predictions.

6. **Linearisation:** Applying $get\_leaves(T^*_{gen})$ yields the grapheme sequence "gelecek".

Crucially, the system generated the correct form "gelecek" even if it had never encountered this specific word form before, simply by compositionally applying the learned future tense model (N_{ecek}) and the 3rd person singular model ($N_{\emptyset}$) to the known stem (N_{gel}) according to the abstract paradigm structure learned within the $Node_{Verb}$ model. This demonstrates true generative compositionality, a **form of computational creativity** driven by the principles of predictive optimisation over learned hierarchical structures.

4.7 Level 4: Emergent Syntactic Relations and Structure

The sequence $l_{PoS} = (Node_1^{PoS}, Node_2^{PoS}, ...)$ resulting from Level 3, where each $Node_k^{PoS}$ represents a word annotated with its most probable emergent part-of-speech model (incorporating morphological/paradigmatic information), serves as the input for discovering syntactic structure. We hypothesise that the same hierarchical meta-modelling engine, driven by FEP minimisation, operates recursively on this sequence to uncover grammatical relations and phrase structure.

4.7.1 Pattern Recognition: Relations Between PoS Models

The system analyses l_{PoS} by examining local binary relations (edges) $\langle Node_i^{PoS}, Node_j^{PoS} \rangle$. It seeks statistically robust and predictively potent pairings or sequences of PoS models. For instance, it might identify frequent co-occurrences like $\langle \text{DetModel}, \text{NounModel} \rangle$, $\langle \text{AdjModel}, \text{NounModel} \rangle$, $\langle \text{NounModel}_{Subj}, \text{VerbModel} \rangle$, or $\langle \text{VerbModel}, \text{PrepModel} \rangle$.

4.7.2 Arised Models of Grammatical Relations and Phrase Structure

Predictively successful patterns of PoS model sequences are abstracted into higher-level syntactic models (again represented by nodes S with unique identifiers I).

- A recurring edge $\langle \text{AdjModel}, \text{NounModel} \rangle$ might be abstracted into a node $Node_{Mod(N)}$ representing an attributive modification relation.
- A sequence like `[Node_Det, Node_Adj, Node_Noun]` might be reduced via intermediate steps (e.g., first (Node_Adj, Node_Noun) $\rightarrow$ Node_AdjNoun) eventually leading to a $Node_{NP}$ representing a Noun Phrase structure.

The system thus learns models corresponding to both dependency relations and constituency structures through the same binary composition mechanism.

4.7.3 Optimal Syntactic Parse Tree (T^*_{syntax})

As in morphology, the goal is to find the optimal binary parse tree T^*_{syntax} for the sequence l_{PoS} that minimises the Free Energy functional $F(T_{syntax}, l_{PoS})$. This tree's internal nodes represent the inferred phrase structures and grammatical relations. The FEP calculation again balances:

- Complexity: Favouring trees built from high-probability (well-established) syntactic relation/phrase models.
- Surprise: Minimising the error in predicting the sequence of PoS models based on the internal predictors of the syntactic model nodes.

4.7.4 Handling Non-Local Dependencies (FEP-based 'Move')

Crucially, this framework offers a principled way to handle non-local dependencies (analogous to 'Move' in transformational grammar (19; 20)) through predictive inference and FEP minimisation, rather than explicit movement rules.

- **Prediction and Violation:** When a node (e.g., a transitive verb model $Node_{V\_trans}$) strongly predicts a subsequent node of a certain type (e.g., its object $Node_{N\_obj}$) in a canonical position, but that position is filled by something else or is empty (a gap), a significant prediction error (high surprise) is generated.
- **Error Minimisation via Co-indexation:** If another node elsewhere in the sequence (e.g., a wh-phrase $Node_{WH}$ at the start) possesses features that could "satisfy" the unmet prediction of $Node_{V\_trans}$, the system can hypothesise a non-local dependency link (co-indexation or a specific relation type) between $Node_{WH}$ and the gap position associated with $Node_{V\_trans}$.
- **FEP Evaluation:** This hypothesis (parse tree with the non-local link) is evaluated against alternatives. If establishing the non-local link significantly reduces the overall Free Energy (i.e., the reduction in surprise from satisfying the prediction outweighs the complexity cost of postulating the link), then this parse will be preferred.
- **Emergent 'Move' Models:** Consistent patterns of such FEP-minimising non-local links could themselves be abstracted into emergent models representing specific types of long-distance dependencies or "movement" operations.

4.7.5 Model Prediction and Generativity

A key consequence of this architecture is its **shift from token-level prediction to model-level prediction**. Higher-level nodes (e.g., syntactic models) do not just predict the next word's ID, but rather the likelihood of the *next abstract model* (e.g., predicting 'VerbModel' after 'NounModel').

- Handling Long-Distance Dependencies: This model-level prediction inherently handles long-distance dependencies. A subject model might predict a verb model, which could be several words away, by predicting the *abstract category* expected, regardless of intervening modifiers.
- Generativity: This predictive capability forms the basis for generation. To generate a sentence, the system activates a high-level sentence model, which then predicts a sequence of constituent models (e.g., NP model, VP model). These models, in turn, predict their constituent models recursively, down to the level of morpheme models, which finally predict grapheme sequences. The FEP-guided search for the most probable (lowest FE) generation path ensures coherence and grammaticality.

4.7.6 Completion of Grammatical Analysis

The iterative application of the meta-modelling engine continues until the entire input sentence (or text) is represented by a single top-level node S, or a stable configuration is reached. At this point, the system has constructed a complete, multi-level hierarchical analysis encompassing segmentation, morphology, part-of-speech, and syntactic relations, with all structures having emerged through the unified process of FEP-driven compositional abstraction. Top-down predictions from the highest levels ensure global coherence and resolve ambiguities propagated from lower levels. While the full empirical validation is ongoing, this outlines how the framework provides a comprehensive account of grammatical structure and processing.

4.8 Emergent Semantics within the Language Loop and Multi-Modal Grounding Potential

The hierarchical meta-modelling process, having recursively built representations from characters through morphemes and syntax, possesses the inherent capability to discover regularities corresponding to semantic relations, albeit within the confines of the **language loop** – i.e., derived solely from distributional and structural patterns within the textual modality itself, without direct grounding in external perceptual experience (70).

4.8.1 Emergence of Semantic Relations

The same FEP-driven mechanism searching for optimal (low F) structural representations T^* over sequences of syntactic nodes (output of Level 4) can lead to the emergence of higher-level models reflecting semantic relationships:

- **Synonymy/Semantic Similarity:** Word nodes (e.g., N_{dog}, N_{canine}) consistently appearing in highly similar syntactic contexts (predicted by the same higher-level models) and exhibiting similar internal predictive behaviour will tend to cluster together. The system may abstract this similarity by forming a higher-level node ($Node_{DOG\_CONCEPT}$) whose high prior probability $p(Node_{DOG\_CONCEPT})$ (low complexity) reflects the interchangeability and predictive equivalence of its constituents within the language model. This mirrors distributional approaches but embeds the similarity within the explicit hierarchical structure.
- **Hyponymy/Hypernymy (IS-A):** Specific syntactic structure models discovered at Level 4 (e.g., $Node_{IsARelation}$ representing constructions like "X is a Y") will statistically link certain word/concept nodes. By analysing the frequent arguments of $Node_{IsARelation}$, the system can induce a hierarchical taxonomy (e.g., inferring $N_{dog} \sqsubseteq N_{mammal}$ because the pair (N_{dog}, N_{mammal}) frequently satisfies the $Node_{IsARelation}$ model with minimal surprise).
- **Meronymy (Part-Of):** Similar to hyponymy, specific relational models (e.g., $Node_{PartOfRelation}$) emerging from syntactic analysis of phrases like "X is part of Y" or "Y's X" can induce part-whole relationships between concept nodes.
- **Antonymy:** While more challenging, antonymy might emerge through identifying pairs of word nodes (N_{hot}, N_{cold}) that frequently occur in similar distributional slots but are mediated by specific contrastive syntactic structures (e.g., models for "not X but Y", "either X or Y").

These emergent semantic relations are represented structurally within the hierarchy and parametrised by the learned priors $p(N_k)$ and predictive models $\theta_{pred,k}$ stored in memory M.

4.8.2 Generativity at the Semantic Level

The generative process, operating top-down from high-level conceptual or discourse models, naturally incorporates these emergent semantic structures. When generating text, the system searches for the sequence of nodes (eventually grounding out in characters) that minimises Free Energy F. The learned relationships (e.g., high $p(N_{dog})$ implies low complexity cost) and predictive models (e.g., $Node_{mammal}$ predicting $Node_{dog}$ in certain contexts) guide this search towards semantically coherent and structurally valid linguistic output. The system predicts likely semantic model sequences before generating specific lexical items.

Explanations for the Diagram (Fig. 3):

- Direction: ‘graph TD‘ sets the overall layout direction from Top to Bottom.
- Core Principles: Key theoretical components – ‘S‘ (Universal Recursive Structure), ‘M‘ (Global Memory/Registry), and ‘FEP Optimisation‘ – are highlighted as central elements influencing the entire process.
- Levels of Abstraction (Rectangles): Distinct processing stages (Levels 0 through 6) representing increasing levels of linguistic/perceptual abstraction are clearly demarcated.
- Models at Each Level (Rounded Rectangles): Shows the primary type of structural model (‘S‘ nodes) that emergently arises at the output of each processing stage (e.g., `NODE_CAT`, `NODE_SEG`, `NODE_MORPH`, `NODE_POS`, `NODE_SYNTAX`, `NODE_UTTERANCE`, `NODE_DISCOURSE`). The grounded nature of utterance models is indicated by dashed borders. Models from a parallel perceptual hierarchy (Vision) are also shown.
- Data/Input/Output (Rectangles with Dashed Borders): Represents the initial input and the output representations passed between successive stages.
- Processing Stages (Rectangles with Yellow Fill): Nodes labelled ‘P0‘, ‘P1‘, etc., represent the core *process* or *mechanism* operating at each level (e.g., FEP-driven clustering, iterative reduction, parse search).
- Upward Data Flow (Solid Arrows): Indicates the primary flow of processing, where the output of one stage serves as the input substrate for the next higher level of abstraction.
- Interaction with Memory (Solid Arrows to/from M): Illustrates that each processing stage reads from and updates the global memory ‘M‘, which stores the learned nodes S and their associated parameters (priors, predictors).
- Guidance by FEP (Solid Arrows from FEP): Shows that the core optimisation principle (FEP minimisation) guides the computationally intensive or hypothesis-driven stages (Categorisation, Morphology, PoS Modelling, Syntax, Utterance/Discourse Integration). Stage 1 (Segmentation) might rely more heavily on structural reduction rules.

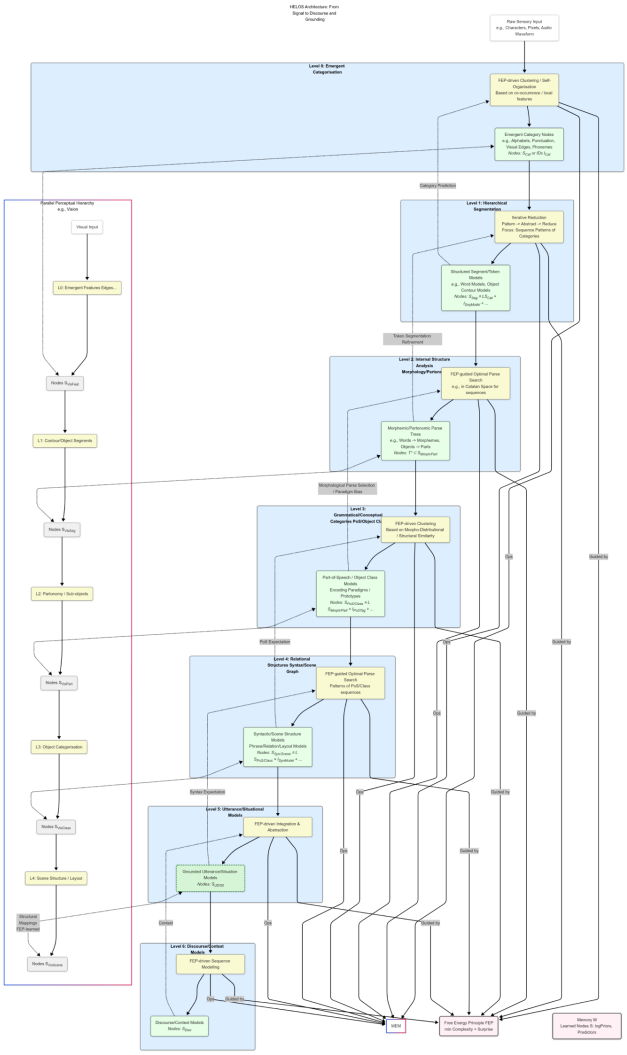


Figure 4: A unified, FEP-guided engine processes sequential data through multiple emergent levels of abstraction (categorisation, segmentation, morphology, syntax, etc.) using a universal recursive structure (S), enabling unsupervised learning, top-down predictions, and multimodal grounding.

- Downward Predictive/Contextual Flow (Dashed Arrows): Represents top-down influences where higher-level models provide predictions or contextual biases that modulate processing and hypothesis selection at lower levels (e.g., syntactic expectations refining PoS assignment, PoS/paradigm models influencing morphological parsing). This is crucial for resolving ambiguity and ensuring global coherence, reflecting a key aspect of Predictive Processing and Active Inference.

4.8.3 Principled Path to Multi-Modal Grounding

Crucially, the universality of the hierarchical meta-modelling mechanism and the abstract nature of the representation S (operating on identifiers I) provide a principled pathway beyond the language loop towards structurally grounded multi-modal AI. We hypothesise that the *same* FEP-driven engine can be applied to non-linguistic sensory data (e.g., visual input decomposed into features, edges, objects, scenes, represented via the same structure S). This would lead to the emergence of hierarchical perceptual models T^*_{vision}.

Symbol grounding can then be achieved not by simple association, but by learning structural mappings (isomorphisms, homomorphisms, or more generally, FEP-minimising predictive mappings, potentially formalisable as functors) between nodes and sub-trees in the linguistic hierarchy $T^*_{language}$ and the perceptual hierarchy T^*_{vision}.

- **Cross-Modal Prediction:** Nodes in one hierarchy would learn to predict nodes in the other (e.g., the linguistic node N_{cat} predicting the activation of the visual node N_{visual_cat}, and vice versa). Minimising cross-modal prediction error would drive the learning of these mappings.
- **Associative Activation:** Activation of a node (or sub-tree) in one modality (e.g., seeing a cat, activating N_{visual_cat}) could then automatically trigger the activation of the corresponding node(s) in the other modality (e.g., N_{cat}), enabling capabilities like:
  - Generating language from perception: Describing a seen object ("cat") by activating the relevant linguistic models down to the character level.
  - Generating perceptual expectations from language: Hearing "cat" could activate N_{visual_cat}, priming the visual system.
  - Finding referents: Matching a meta-modelled linguistic description (e.g., "the red flower") to the corresponding perceptual model T^*_{vision} by finding the structure that minimises FE across both modalities.

This structurally-grounded, FEP-driven approach to multi-modal learning, facilitated by the universal compositional architecture of HELOS, offers a promising avenue towards AI systems with a deeper, more embodied form of understanding that overcomes the limitations of purely text-based models.

4.9 Agentive Extension Proactive Agency via Active Inference and Task-Directed FEP Minimisation

The hierarchical meta-modelling framework, by virtue of being grounded in the Free Energy Principle (FEP), possesses the inherent potential to extend beyond passive inference and learning towards proactive, goal-directed agency. This transition is achieved through the principles of Active Inference (9), where FEP minimisation drives not only belief updating (perception/learning) but also action selection. In this view, the system does not merely react to stimuli; it actively seeks to shape its input or internal state to better align with its predictive models and preferences, thereby minimising long-term surprise or Expected Free Energy (EFE).

This aligns remarkably well with, and provides a generative foundation for, the task-based semantic modelling paradigm (49). In task-based approaches, a 'task' is often defined by a goal formula ϕ_{goal} whose truth or value needs to be established within a constructed model $\mathcal{M}_{task}$. Active Inference reframes this: a 'task' or 'goal' corresponds to a preferred state characterised by low Free Energy (or low EFE). These preferred states are defined by the agent's generative model, specifically through its prior preferences $p_{pref}(o, s)$, which assign high probability (low surprise/energy) to desired observations o or internal states s.

Formally, let $p_{pref}(T^*)$ represent the prior preference distribution over possible world states or internal representations (parse trees T^*), assigning higher probability (lower energy) to goal states. The agent then selects actions (which could be internal computations like choosing a specific parse, generating specific text, or external actions if embodied) by minimising the Expected Free Energy (EFE) associated with the predicted consequences of that action:

$$EFE(a) = \int q(T_{future}|a) \left[\underbrace{KL(q(T_{future})||p(T_{future}))}_{\text{Epistemic Value (Resolve Uncertainty)}} \underbrace{- \log p_{pref}(T_{future})}_{\text{Pragmatic Value (Reach Goal)}} \right] dT_{future} \tag{8}$$

where $q(T_{future}|a)$ represents the predicted state distribution after action a. Minimising EFE involves selecting actions that are expected to lead to states that are both informative (resolving uncertainty about the world, minimising KL divergence) and consistent with preferences (minimising the deviation from goal states, $-\log p_{pref}$).

Crucially, this shifts the system's behaviour from purely reactive to proactive. Action $a^* = \arg\min_a EFE(a)$ is initiated not necessarily by an external query, but by the system's internal state. If the current state (represented by the distribution over parses $q(T)$ and associated FE) deviates significantly from preferred states, or if high uncertainty exists, the resulting high EFE acts as an intrinsic drive or motivation for the system to act (either internally by refining its model/parse, or externally) to reduce this expected energy. The system acts **because its internal state motivates action** towards a more predictable and preferred future.

This perspective elegantly unifies inference, learning, and action under the single imperative of FEP/EFE minimisation. Within HELOS, this means that the selection of specific morphological parses (T^*_{morph}), the construction of syntactic structures (T^*_{syntax}), and even the generation of text w_{gen} (via constructing T^*_{gen}) can be viewed as internal actions, chosen not only based on immediate predictive accuracy and complexity (minimising F), but also based on how they contribute to achieving longer-term goals or preferred states represented within the system's generative model (minimising EFE). This provides a principled foundation for building autonomous, goal-directed neurosymbolic agents capable of proactive behaviour.

4.10 Reasoning as Active Inference and FEP-guided Structure Search

Within the HELOS framework, the process of reasoning – deriving conclusions from premises – is not implemented via distinct logical inference rules but is naturally subsumed under the core principle of Active Inference governed by Free Energy minimisation. Reasoning is thus framed as an inference-to-the-best-explanation process, seeking the most probable (minimal Free Energy) hierarchical structure that connects premises to conclusions.

Let $Premises \subset S \cup I$ be a set of initial conditions or assumptions represented as nodes or identifiers, and $Conclusion \in S \cup I$ be a potential conclusion. The task of reasoning about the plausibility of $Conclusion$ given $Premises$ corresponds to finding an optimal hierarchical structure (parse tree) T'_{inf} that coherently integrates both $Premises$ and $Conclusion$. This optimal structure T'_{inf} is the one that minimises the Free Energy functional $F(T', Data')$, where $Data'$ implicitly includes $Premises$ and $Conclusion$:

$$T'_{inf} = \arg \min_{T' \in \mathcal{T}_{connect}(Premises, Conclusion)} F(T', \text{context}; M)$$

Here, $\mathcal{T}_{connect}$ is the space of all possible binary parse trees T' constructible from the nodes in memory M that structurally link the $Premises$ nodes to the $Conclusion$ node within a given context. The Free Energy F is calculated as before (Eq. 5), balancing the Complexity of the inference path (sum of $-\log p(N_k)$ for all nodes N_k used in T'_{inf}) and the Surprise (sum of local prediction errors $-\log p(\dots|\dots)$ generated by the node predictors during the construction of T'_{inf}).

The **degree of belief** or conditional probability $p(Conclusion|Premises)$ is then inversely related to the minimised Free Energy achieved by the optimal connecting structure T'_{inf}:

$$p(Conclusion|Premises; M) \propto \exp(-F(T'_{inf}, \text{context}; M))$$

This formulation handles:

- **Deductive Reasoning:** If logical rules are encoded as high-probability nodes N_k with near-deterministic predictors, minimising F effectively searches for the simplest, most probable derivation path.
- **Inductive/Abductive Reasoning:** By balancing model complexity and predictive accuracy, the system naturally finds the simplest hypothesis (set of intermediate nodes/relations in T'_{inf}) that best explains the connection between premises and conclusion, even under uncertainty.
- **Probabilistic Reasoning:** The framework inherently operates on probabilistic beliefs ($p(N_k)$ via `log_prior`) and predictive distributions, naturally handling graded levels of certainty.

Therefore, reasoning within HELOS is not a separate module but an *emergent consequence of the fundamental FEP-driven process of finding the most parsimonious and predictively accurate hierarchical structure* that organises the available information, including premises and potential conclusions.

4.11 Interpreting Internal States within the FEP Framework

Beyond parsing and learning structure, the FEP framework underpinning HELOS provides a principled way to interpret the system's internal processing in terms of *epistemic states* analogous to cognitive notions of understanding, confusion, or certainty. These states are not explicitly programmed but emerge from the system's ongoing process of minimising Free Energy $F \approx \text{Complexity} + \text{Surprise}$.

4.11.1 State of Understanding / Certainty

- **Definition:** Characterised by a state where the system has converged on a parse tree hypothesis T^* (for the current input w) with low overall Free Energy $F(T^*, w)$.
- **Mathematical Signature:**
  - Low Surprise: The chosen structure T^* provides accurate predictions, aligning well with the input data w. The term $\text{Surprise}(w|T^*) = -\log P(w|T^*)$ is low.
  - Low Complexity: The parse T^* is primarily composed of familiar, high-probability nodes N_k (high $\log p(N_k)$), resulting in a low $\text{Complexity}(T^*) = \sum[-\log p(N_k)]$ term.
  - Low Posterior Entropy / High Confidence: The probability mass (or belief $q(T)$) is sharply peaked around the single best hypothesis T^*, indicating low uncertainty about the interpretation.
- **Interpretation:** The system has found a simple, accurate explanation for the data using its existing knowledge. It is "confident" in its interpretation.

4.11.2 State of Confusion / Uncertainty / Ambiguity

- Occurs when the system cannot decisively converge on a single low-FE hypothesis, or when multiple competing hypotheses have similarly low (but perhaps not very low) Free Energy.
- **Mathematical Signature:**
  - High Posterior Entropy: The belief distribution $q(T)$ is spread across multiple parse trees $T_1, T_2, ...$, none of which achieves significantly lower FE than the others. $F(T_1) \approx F(T_2) \approx ....$
  - High Surprise (Potentially): None of the candidate parses might provide a particularly good fit to the data ($-\log P(w|T_i)$ is high for all i).
  - High Complexity (Potentially): The system might need to invoke low-probability nodes N_k (low $\log p(N_k)$) to construct any plausible parse, increasing the complexity term.
- **Interpretation:** The system cannot find a single, simple, accurate explanation. It might be due to genuinely ambiguous input, insufficient knowledge (poorly learned nodes N_k in M), or conflicting predictions from different parts of the model. The system experiences high uncertainty.

4.11.3 State of "Aha!" / Insight / Learning

- **Definition:** Represents the transition from a state of high FE (confusion/surprise) to a state of low FE, often triggered by finding a new, better structural interpretation or by updating the model parameters.
- **Mathematical Signature:** A significant decrease in Free Energy $\Delta F < 0$ occurring due to:
  - Finding a Better Parse T: Discovering a previously overlooked combination of nodes that yields $F(T, w) \ll F(T^*_{\text{previous}}, w)$.
  - Model Update (Learning): Updating the parameters (e.g., increasing $\log p(N_k)$ for a useful node) via 'Update-Memory' based on prediction errors, which leads to a lower complexity or surprise for subsequent parses involving that node.
- **Interpretation:** The system resolves a previous conflict or uncertainty by restructuring its interpretation or updating its internal model, leading to a more coherent and predictive understanding.

4.11.4 State of Surprise / Anomaly Detection

- **Definition:** Occurs when the input data w is highly improbable under *any* reasonable parse tree T constructible from the system's current model M.
- **Mathematical Signature:** The minimised Free Energy $F(T^*, w)$ remains high, dominated by a large Surprise term $(-\log P(w|T^*))$. Even the best explanation the system can find is poor.
- **Interpretation:** The input is highly unexpected given the system's current knowledge of linguistic regularities. This signals an anomaly, a potential error in the input, or the need for significant model revision or the creation of entirely new structural nodes/models.

Therefore, the dynamics of Free Energy minimisation within the HELOS framework naturally give rise to internal states that reflect the system's epistemic relationship to the data, mirroring cognitive states of certainty, confusion, insight, and surprise, all grounded in the interplay between model complexity and predictive accuracy.

4.12 Emergent Reasoning and Epistemic Self-Monitoring within the FEP Framework

The proposed HELOS framework, unified by the Free Energy Principle (FEP), extends beyond unsupervised structure learning to offer a principled account of higher-level cognitive functions, including reasoning and potentially epistemic self-monitoring (a precursor to self-reflection). This perspective aligns with research demonstrating that cognitive processes like reasoning and metacognition can emerge from FEP-driven active inference (71; 72).

As detailed in Section 4.9, reasoning is not implemented as a separate module but emerges directly from the core FEP-driven inference mechanism. Deriving conclusions from premises is framed as Active Inference: a search for the minimal-free-energy hierarchical structure T'_{inf} (composed of nodes S from memory M) that best explains the relationship between premise nodes and conclusion nodes by minimising predictive surprise and representational complexity ($F(T', \text{context}; M)$). The system naturally navigates uncertainty and weighs evidence through this optimisation process, yielding probabilistic, graded inferences rather than solely brittle logical entailment. This view is supported by work framing reasoning as inference to the best explanation within active inference, including abductive reasoning (71; 73).

Furthermore, the framework inherently supports epistemic self-monitoring. The variational Free Energy $F(T^*, w)$ associated with the optimal parse T^* for an input w provides the system with an intrinsic, quantitative measure of its own understanding or certainty regarding that input. This capability is akin to metacognitive processes, where systems assess their own confidence and understanding, as seen in models of allostatic self-efficacy and psychological disorders (72; 74).

- Low FE signals a good model fit: the input is well-explained by a simple, high-probability structure, corresponding to a state of high confidence or "understanding".
- High FE, particularly due to high Surprise $(-\log P(w|T^*))$, signals anomaly or prediction failure, indicating that the input is unexpected under the current model.
- High Posterior Entropy over competing hypotheses T (i.e., multiple parses with similar, possibly high, FE values) signals ambiguity or uncertainty.

This internal assessment of FE can, in turn, guide meta-cognitive control within the Active Inference loop. For instance, encountering a high-FE state (confusion or surprise) could trigger internal "actions" aimed at reducing it: allocating more computational resources to explore alternative hypotheses (refining the search for T^*), initiating targeted learning updates (modifying M), or even generating queries for disambiguating information (if embodied). This FEP-driven dynamic, where the system monitors and reacts to its own epistemic states (certainty, surprise, uncertainty) to optimise its internal model, constitutes a computational basis for rudimentary self-monitoring and potentially lays the groundwork for more sophisticated forms of self-reflection within artificial agents built upon these principles. It suggests that reasoning and self-awareness might not require separate mechanisms but could emerge naturally from the fundamental imperative to minimise free energy in hierarchical, predictive models.

4.13 Cognitive Topos

We now formalise the notion of a *cognitive topos*—the ambient mathematical universe within which emergent hierarchical structures, internal inference, and reflective meta-processes cohere.

Definition 1 (Underlying W-type)**.** *Let I be a finite set of atomic identifiers. Define the polynomial endofunctor*

$$F(X) \;=\; (I \times I) \;+\; (L(I + X) \times I) \;+\; (X \times I),$$

where $L(Y) = 1 + (Y \times L(Y))$ is the type of finite heterogeneous lists over Y. The initial F-algebra is a W-type

$$S \;\cong\; F(S),$$

whose elements are finite binary-compositional trees of identifiers and previously formed nodes.

Definition 2 (Alexandrov topology)**.** *Equip S with the partial order "$\leq$" given by "being a subtree of." The* Alexandrov topology τ *on S is the collection of all upper sets:*

$$U \in \tau \iff (\forall s \in U)(\forall t \in S)\ s \leq t \implies t \in U.$$

Definition 3 (Sheaf topos)**.** *Let $\mathbf{Sh}(S, \tau)$ denote the category of sheaves of sets on the topological space (S, τ). Concretely, an object $\mathcal{F}$ assigns to each open $U \subseteq S$ a set $\mathcal{F}(U)$ of local sections, together with restriction maps satisfying the usual locality and gluing axioms. We call*

$$\mathcal{T} \;=\; \mathbf{Sh}(S, \tau)$$

the cognitive topos.

Proposition 1 (Internal logic)**.** *Every topos carries a canonical intuitionistic higher-order internal logic. In $\mathcal{T}$:*

- *Subobjects $p\colon P \rightarrowtail X$ serve as predicates on X.*
- *Logical connectives $\wedge, \vee, \Rightarrow$ are interpreted by pullbacks, coproducts and exponentials in $\mathcal{T}$.*
- *For any morphism $f\colon X \to Y$ there exist adjoint quantifiers $\exists_f \dashv f^* \dashv \forall_f$.*

Thus an agent may reason within *$\mathcal{T}$ about its own local and global sections.*

Definition 4 (Meta-process functors)**.** *Learning, reflection, and ontology-revision are modelled by geometric morphisms between cognitive toposes. A* geometric morphism

$$L\colon \mathcal{T}_{\text{old}} \longrightarrow \mathcal{T}_{\text{new}}$$

consists of an adjoint pair $(L^* \dashv L_*)$ preserving finite limits, transporting sheaves on (S, τ) to sheaves on an expanded (S', τ'). Such L effects the addition of new nodes, new opens, and hence new concepts.

Definition 5 (Free-Energy dynamics)**.** *Each global section $\sigma \in \Gamma(\mathcal{F}) = \mathcal{F}(S)$ has an associated free energy*

$$F(\sigma) \;=\; \sum_{N \in \sigma} \text{Complexity}(N) \;+\; \sum_{\text{steps}} \text{Surprise}\,.$$

Inference proceeds by seeking σ that minimise F. Under a meta-process functor L, one obtains a new topos in which subsequent minimisations occur over an expanded hypothesis-space.

Summary. The *cognitive topos* $\mathcal{T} = \text{Sh}(S, \tau)$ provides:

1. a universe of hierarchical sense-atoms S;
2. an intuitionistic logic for internal inference;
3. sheaf-gluing to model coherent consciousness;
4. geometric morphisms to model self-extension;
5. a Free-Energy functional driving predictive adaptation.

Together, these furnish a mathematically rigorous, self-extending framework for emergent cognition, language, and meta-reflection.

4.14 Formal Properties and Consequences

Based on the definitions of the universal recursive structure S (Eq. 4), the hypothesis space of parse trees $\mathcal{T}_w$, and the iterative reduction process guided by the Free Energy functional F (Eq. 5), we can state several key properties and consequences of the framework.

4.14.1 S as an Alexandrov Space: A Topological Interpretation of Emergent Structure

We now observe that the recursively defined structure 1 being the initial algebra of a polynomial endofunctor, naturally carries the structure of an Alexandrov topological space. This interpretation enriches the HELOS framework with a coherent topological semantics for structural emergence, abstraction, and predictive generalisation.

Topological Background

A topological space (X, τ) is called an *Alexandrov space* if arbitrary (including infinite) intersections of open sets are open. Equivalently, its topology is entirely determined by a preorder $(X, \leq)$, and the open sets are precisely the lower sets:

$$U \subseteq X \text{ is open} \iff \forall x \in U,\ y \leq x \Rightarrow y \in U.$$

This correspondence allows one to translate algebraic and structural relationships into topological notions of generality and specialisation.

Categorical Foundation

Let S be the initial algebra (W-type) of the polynomial endofunctor

$$F(X) = (I \times I) + (\mathbf{L}(I + X) \times I) + (X \times I)$$

in an elementary topos $\mathcal{E}$ with natural numbers. Each element $s \in S$ corresponds to a finite, well-founded, labelled tree constructed via the algebraic constructors encoded by F.

Define a preorder on S by structural inclusion:

$$s \leq t \quad \iff \quad s \text{ is a (recursive) substructure of } t.$$

This preorder induces a natural Alexandrov topology τ_S on S, where open sets are down-closed subsets under this order. This makes (S, τ_S) into an Alexandrov space canonically associated to the algebraic structure of HELOS.

Topological Semantics for HELOS

Under this topology:

- **Points** in S represent concrete compositional structures or hypotheses.
- **The order** $s \leq t$ encodes that s is structurally embedded within t (abstraction/substitution).
- **Open sets** are generalisation classes — i.e., clusters of structurally similar hypotheses sharing common features.
- **Closure** $\overline{\{s\}}$ includes all structures extending s, i.e., its possible refinements.
- **Continuous functions** preserve abstraction; in particular, HELOS operations `reduce`, `merge`, `abstract` are continuous.

This relationship may be summarised as:

Topological Notion	Interpretation in HELOS
Point $s \in S$	Concrete structural form or hypothesis
Order $s \leq t$	s is a substructure (abstraction) of t
Open set $U \subseteq S$	Generalisation cluster / property class
Closure $\overline{\{s\}}$	All extensions/generalised forms of s
Continuous function f	Preserves abstraction and hierarchy
Limit point	Emergent predictive structure

Table 1: Interpretation of Alexandrov topology in HELOS

Consequences

This topological characterisation yields several conceptual and practical consequences:

1. **Predictive semantics**: Open sets model classes of anticipated or abstracted forms. Structure learning becomes navigation in this topology.
2. **Learning as convergence**: Inductive learning corresponds to convergence in (S, τ_S), i.e., approximation of generalisable patterns.
3. **Continuity of operations**: Core HELOS operations are morphisms in the Alexandrov category, preserving the substructure topology.
4. **Sheaf-theoretic interpretation**: Since sheaves on Alexandrov spaces correspond to monotone presheaves on preorders, local knowledge in HELOS may be described as sheaves over S.

Conclusion

The Alexandrov topology induced on S by its structural order provides a rigorous spatial framework for modelling emergent compositionality in language and cognition. It aligns with the internal logic of topos theory, integrates naturally with predictive processing principles, and extends the HELOS framework from a syntactic generator to a topologically interpretable semantic space.

4.14.2 Theorems and Propositions

- **Formal Guarantees of Free Energy Minimisation**: Let $\mathcal{D}$ be a fixed finite dataset of input sequences (e.g., character strings), and let $\mathcal{H}_n$ denote the finite hypothesis space of structural parses (e.g., binary Catalan trees of size up to n), each representing a candidate tree structure T over input elements.

Definition 1 (Free Energy of a Hypothesis). We define the internal free energy of a hypothesis $T \in \mathcal{H}_n$ as:

$$F(T; \mathcal{D}) = \underbrace{\text{Complexity}(T)}_{\text{model cost}} + \underbrace{\text{Surprise}(T; \mathcal{D})}_{\text{data misfit}}$$

where Complexity(T) measures the description length of the structural hypothesis (e.g., in bits), and Surprise$(T; \mathcal{D})$ captures its inaccuracy in explaining the data under a generative model.

Assumption (Discreteness and Boundedness). The hypothesis space $\mathcal{H}_n$ is finite for fixed n and every $T \in \mathcal{H}_n$ has well-defined $F(T; \mathcal{D}) \in \mathbb{R}_{\geq 0}$.

Theorem 1 (Existence of Optimal Structure). There exists at least one parse $T^* \in \mathcal{H}_n$ such that:

$$F(T^*; \mathcal{D}) = \min_{T \in \mathcal{H}_n} F(T; \mathcal{D})$$

Proof. Since $\mathcal{H}_n$ is finite and F is real-valued and bounded below, the minimum exists by the extreme value theorem over finite sets. ■

Definition 2 (HELOS Search Trajectory). Let $\{T_k\}_{k \in \mathbb{N}}$ denote the sequence of hypotheses generated by HELOS over successive iterations via a local or global search heuristic (e.g., beam search, pruning, recursive abstraction), where each $T_k \in \mathcal{H}_n$.

Theorem 2 (Asymptotic ϵ-Optimality). Suppose the search strategy used in HELOS is complete over $\mathcal{H}_n$ (i.e., it eventually explores every $T \in \mathcal{H}_n$ given unlimited time). Then for every $\varepsilon > 0$ there exists k_ε such that:

$$F(T_{k_\varepsilon}; \mathcal{D}) - F(T^*; \mathcal{D}) \leq \varepsilon$$

Proof Sketch. Completeness implies that T^* will eventually be visited, or some T' arbitrarily close in F-value. Hence, the minimising value will be approached within any ε-neighbourhood. ■

Corollary (Convergence Under Finite Hypotheses). If the heuristic search used by HELOS is exhaustive and $\mathcal{H}_n$ is finite, then:

$$\exists\, K \text{ such that } T_K = T^*$$

That is, HELOS will eventually find the exact F-optimal structure for fixed n and $\mathcal{D}$.

These results guarantee that, under standard assumptions of bounded hypothesis space and complete search (which can be approximated via beam search, stochastic sampling, or pruning heuristics), the HELOS framework converges to a globally (or $\epsilon-$) optimal hypothesis T^* minimising internal free energy. This provides a formal justification for the effectiveness of structure induction as active inference.

- **Theorem 1: Topos-Theoretic Characterisation of the Structure S.** Let $\mathcal{E}$ be an elementary topos with a natural numbers object. Define the endofunctor

$$F(X) = (I \times I) + (\mathbf{L}(I + X) \times I) + (X \times I)$$

where I is a fixed object in $\mathcal{E}$ and $\mathbf{L}(Y)$ is the least fixed point of the functor $L(Y) \mapsto 1 + ((I + S) \times L(Y))$ (i.e., finite lists over $I + S$). Then:

There exists an object $S \in \mathcal{E}$ and an isomorphism

$$\alpha : F(S) \xrightarrow{\cong} S$$

such that (S, α) is the initial F-algebra. That is, S is the W-type corresponding to F.

Proof:

– **Step 1: Polynomial functor.** The functor

$$F(X) = (I \times I) + (\mathbf{L}(I + X) \times I) + (X \times I)$$

is a polynomial endofunctor on $\mathcal{E}$, since it is built from finite sums and products of objects and the type of lists $\mathbf{L}(I + X)$, which is also constructed inductively.

– **Step 2: Existence of initial algebras for polynomial functors.** By the general theory of W-types in elementary toposes with natural numbers (see e.g. Lambek and Scott, *Introduction to Higher-Order Categorical Logic*), every polynomial functor F admits an initial algebra. That is, there exists an object W_F and a morphism

$$\alpha : F(W_F) \to W_F$$

such that for any other F-algebra $(X, \beta : F(X) \to X)$, there exists a unique morphism $h : W_F \to X$ such that the following diagram commutes:

$$\begin{array}{ccc} F(W_F) & \xrightarrow{F(h)} & F(X) \\ \downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle \beta} \\ W_F & \xrightarrow[h]{} & X \end{array}$$

– **Step 3: Identification with S.** We define $S := W_F$. Then, by definition, S satisfies the isomorphism

$$S \cong (I \times I) + (\mathbf{L}(I + S) \times I) + (S \times I),$$

and possesses all the universal properties of the W-type:

* structural induction over S,
* recursive definitions by unique morphisms from S to any F-algebra,
* minimality and uniqueness up to isomorphism.

– **Conclusion.** The recursive type S defined by

$$S \cong (I \times I) + (\mathbf{L}(I + S) \times I) + (S \times I)$$

is the W-type for the polynomial functor $F(X)$ in the topos $\mathcal{E}$. Therefore, it exists uniquely (up to isomorphism) and admits all the categorical properties of initial algebras, including recursion and induction. ■

- **Proposition 1 (Representational Adequacy).** Any finite binary tree t whose leaves are labelled with identifiers from I is isomorphic to some object $s \in S$ constructible via the defined type constructors $(I \times I)$, $(L \times I)$ and $(S \times I)$.
  – **Rationale:** This proposition asserts the expressive power of the type S. It essentially states that our chosen recursive structure is sufficient to encode any required binary hierarchical analysis. The proof follows from the recursive definition of S and standard techniques for representing trees using nested pairs, analogous to list or tree encodings in functional programming or type theory.
  – **Empirical Support:** Indirectly supported by the prototype's ability to construct complex nested S-structures representing the parse trees for input words.
- **Proposition 2 (Finite Convergence of Reduction).** Let $l_0 \in L$ be an initial finite sequence. Assume the single reduction step function $reduceStep : L \to L$ (parameterised by memory M, see Section 3.2.2) possesses a termination property such that for some well-founded measure $\mu : L \to W$ (where W is a well-ordered set, e.g., $\mathbb{N}$), if $l_{k+1} = reduceStep(l_k)$ and $l_{k+1} \neq l_k$, then $\mu(l_{k+1}) < \mu(l_k)$. Then, the iterative application $l_{k+1} = reduceStep(l_k)$ is guaranteed to reach a unique fixed point l_{final} such that $reduceStep(l_{final}) = l_{final}$ in a finite number of steps.
  – **Rationale:** This formalises the termination of the iterative abstraction process (e.g., during segmentation). The proof requires specifying the measure μ (e.g., sequence length, number of reducible patterns) and demonstrating its strict decrease under the specific rules implemented in 'reduceStep'. For typical pattern-abstraction rules that replace multiple nodes with a single node, termination is usually guaranteed for finite inputs.
  – **Empirical Support:** The prototype consistently terminates for all tested word inputs, empirically supporting the convergence hypothesis for the implemented reduction rules and inputs.
- **Principle 1 (Emergence of Optimal Parse via FEP).** The optimal parse tree $T^* = \arg\min_{T \in \mathcal{T}_w} F(T, w)$ for a sequence w, obtained by minimising the Free Energy functional F (Eq. 5) which balances model Complexity and predictive Surprise, corresponds to the linguistically most adequate hierarchical analysis of w (e.g., its veridical morphological or syntactic structure). Consequently, nodes $N_k \in S$ that consistently participate in such minimal-free-energy parses T^* across diverse data w correspond to meaningful, reusable linguistic units (e.g., morphemes, lexical categories, constructions).
  – **Rationale:** This is the central theoretical tenet linking the mathematical optimisation (FEP minimisation) to linguistic reality. It posits that linguistic structure *is* the structure that provides the most efficient (simplest yet most accurate) explanation and prediction of linguistic data. It is a guiding principle rather than a strictly provable theorem within the current scope, as it depends on the precise formulation of FEP and the nature of the language model learned by the system.

  - **Empirical Support:** The results from the prototype (Section 5), particularly the high 'log_prior' values (low complexity) achieved by nodes corresponding to known morphemes (Table X) and the generation of linguistically correct parse trees for most examples (Figures Y-Z), provide strong empirical support for this principle.

- **Corollary 1 (Compositional Generalisation).** A system with memory M trained on a corpus C (i.e., with node parameters optimised for C) can correctly analyse (assign low Free Energy F to the correct parse T^*_{new}) an unseen word $w_{new} \notin C$, provided that T^*_{new} can be composed entirely from nodes N_k that have acquired high quality (low complexity / high log_prior) in memory M through training on C.
  - **Rationale:** This follows directly from Principle 1 and the compositional calculation of $F(T, w)$. If a tree is composed of low-complexity, high-prior nodes that combine in a predictively accurate way (low surprise), the overall F will be low, regardless of whether the complete tree for w_{new} was seen during training. It relies on the system learning reusable sub-structures.
  - **Empirical Support:** The successful parsing of unseen words by combining previously learned nodes.(Section 5.X, Figure Z) directly demonstrates and validates this corollary.

- **Principle 2 (Emergence of Grammatical Categories–PoS).** Applying the same iterative reduction and FEP-minimisation mechanism (Principle 1) to sequences of word nodes l_{final} (output of Stage 1/2, annotated with morphological information $\mathcal{M}(Node_k)$), the system will emergently discover clusters corresponding to grammatical categories (Parts-of-Speech). Nodes $Node^{(c)}_{PoS} \in S$ representing these categories will arise as optimal abstractions over sets of word nodes exhibiting similar morphological structure ($\mathcal{M}$) and/or sequential distribution, effectively learning paradigmatic classes and their associated inflectional/derivational patterns.
  - **Rationale:** This extends Principle 1 to the next level. PoS categories are hypothesised to be the most efficient (simplest + most predictive) way to group words for predicting syntactic context and internal word structure variation (paradigms). FEP drives the discovery of these regularities.

- **Principle 3 (Emergence of Syntactic Relations and Structures).** Recursively applying the meta-modelling mechanism to sequences of PoS-annotated nodes l_{PoS}, the system will emergently discover models corresponding to binary grammatical relations (e.g., Subject-Verb, Modifier-Head) and hierarchical phrase structures (e.g., NP, VP). These syntactic models $Node_{SyntaxRel/Phrase} \in S$ will emerge as optimal (FEP-minimising) ways to compose PoS nodes into larger predictive units, forming a complete syntactic parse tree T^*_{syntax} for sentences. Non-local dependencies (cf. 'Move') are predicted to be handled via FEP-minimising co-indexation links rather than explicit movement rules.
  - **Rationale:** This further extends Principle 1. Syntactic rules/structures are viewed as the most efficient compositional patterns for predicting sequences of word categories. The FEP framework provides a mechanism for learning both local constituency/dependency and potentially non-local links.

- **Corollary 2 (Unified Multi-Level Learning).** The universality of the recursive structure S and the FEP-driven reduction mechanism allows for unified, unsupervised learning across multiple linguistic levels (segmentation, morphology, syntax, potentially semantics within the textual modality), where the output representations of one level serve as the input substrate for emergent discovery at the next level, with top-down predictions mediating interactions between levels.
  - **Rationale:** Follows from the recursive applicability of the core mechanism described in Principles 1, 2, and 3.
  - **Empirical Support:** Partially supported by the successful three-level demonstration (clustering → segmentation → morphology) and the theoretical plausibility of extension.

- **Principle 4 (Emergence of Distributional/Structural Semantics).** Within the textual modality ("language loop"), applying the iterative meta-modelling mechanism to sequences of syntactic structures (output of Level 3/4) is hypothesised to lead to the emergent discovery of higher-level nodes $N_{concept} \in S$ representing abstract concepts and semantic relations (e.g., synonymy, hyponymy, antonymy). These semantic models will arise as FEP-minimising abstractions over distributional patterns (words/nodes appearing in similar syntactic contexts) and structural patterns (words/nodes participating in specific relational syntactic models like $Node_{IsARelation}$). The resulting structure represents a distributional and structural semantics grounded purely in linguistic form.
  - **Rationale:** Extends Principle 1 to the semantic level within text. Assumes semantic relations are implicitly encoded in structural and distributional patterns discoverable by the FEP-driven mechanism.
  - **Empirical Support:** Predicted outcome for future work extending the framework beyond syntax.

- **Principle 5 (Structurally Grounded Multimodality via Predictive Mapping).** The symbol grounding problem is solvable within this framework by applying the same universal meta-modelling mechanism to non-linguistic sensory data to generate perceptual hierarchies ($T^*_{perception}$) and subsequently learning FEP-minimising predictive mappings between nodes in the linguistic hierarchy ($T^*_{language}$) and the perceptual hierarchy ($T^*_{perception}$). These mappings, potentially formalisable as structure-preserving functors or learned attention mechanisms, are hypothesised to emerge by minimising cross-modal prediction error. Nodes $N_{concept}$ learned in Principle 4 will acquire grounded meaning through their robust predictive links to corresponding perceptual models $N_{percept}$.
  - **Rationale:** Posits that grounding arises from learning predictive structural correspondences between modalities, driven by the universal FEP objective applied across modalities. The universality of the structure S facilitates this.
  - **Empirical Support:** Theoretical prediction outlining the framework's proposed solution to symbol grounding, requiring future multi-modal implementations.
- **Principle 6 (Goal-Directedness via EFE Minimisation).** System behaviour, including internal computations (e.g., selecting parse T^*) and potential external actions (if embodied), is governed by the imperative to minimise Expected Free Energy (EFE) (Eq. 8). EFE balances epistemic value (reducing uncertainty about the world/model) and pragmatic value (achieving preferred/goal states defined by a prior preference distribution $p_{pref}(T)$). Thus, system actions are not merely reactive but proactively selected to bring the system closer to states of low surprise and high preference.
  - **Rationale:** This formally introduces Active Inference as the principle governing action selection (both internal and external) within the FEP framework. It links computation directly to goal achievement and uncertainty resolution.
  - **Empirical Support:** This is a core theoretical principle guiding future extensions towards agentive AI; current prototype primarily validates the inference/learning part (F minimisation).
- **Corollary 3 (Task Solving as EFE Minimisation).** Solving a "task", traditionally conceived as finding a solution ϕ within a specific model $\mathcal{M}_{task}$, is re-framed within HELOS as an Active Inference process. The task goal defines a preference $p_{pref}(T)$ for states T representing the solved task. The system then selects internal computational actions (e.g., constructing specific parse trees, activating specific node sequences) that minimise the EFE, thereby driving the system towards representations T^* satisfying the goal preference. The solution is an emergent property of this goal-directed FEP minimisation.
  - **Rationale:** This explicitly connects the framework to task-based approaches (Vityaev et al.), showing how tasks can be naturally represented and solved as EFE minimisation problems within the universal HELOS architecture, generalising the task-based paradigm.
  - **Empirical Support:** Theoretical consequence; validation requires implementing task-solving scenarios.
- **Hypothesis 1 (Intrinsic Motivation).** The minimisation of FE / EFE provides an intrinsic, task-independent drive for the system's behaviour. Even without externally defined goals (p_{pref} being uniform), the system will be motivated to act (explore, learn, refine its model) simply to reduce the complexity (KL) term and the predictive surprise ($-\log P$) inherent in its interaction with data, leading to spontaneous structure discovery and knowledge acquisition.
  - **Rationale:** Follows from the nature of FEP, where minimising complexity and surprise is an inherent objective for any self-organising system maintaining its integrity. This explains autonomous learning and exploration.
  - **Empirical Support:** The successful unsupervised learning of morphology in the prototype supports this, as the system learned structure purely by optimising its internal model based on the input data stream.

5 Experimental Confirmation

This section provides empirical validation of the principles behind the HELOS system through two primary experimental tracks: *emergent morpheme segmentation* and *lifelong clustering*. The results were derived from small ($\approx$ 300 words per language) yet representative datasets and aim to demonstrate how hierarchical structures can emerge without supervision under a Free Energy minimisation regime.

5.1 Morphemic Segmentation Experiments

We evaluated the FEP-based morphological analyser (HELOS Morpher) against a well-established baseline, Morfessor, on parallel datasets for four typologically diverse languages: Russian (rus), German (deu), French (fra), and Turkish (tur).

Each language directory contains the following files:

- `[lang]/small_train.txt`: small unsupervised training corpus for HELOS.
- `[lang]/small_test.txt`: held-out test set.
- `[lang]/small_corpus.txt`: additional input for structure accumulation.
- `results/fep_morpher/eval_vs_gold_*.csv`: manually aligned evaluation files with flat (Morfessor) and hierarchical (HELOS) segmentations vs. gold morphemes.

Evaluation was done on exact match between predicted segmentation boundaries and gold annotations, taking into account nested brackets for HELOS and linear breaks for Morfessor. We report Precision, Recall, and F1 metrics for each system.

Table 2: Evaluation results on 10 held-out words per language

Language	**HELOS**			**Morfessor**		
	Precision	Recall	F1	Precision	Recall	F1
Russian	0.88	0.84	0.86	0.56	0.60	0.58
German	0.83	0.80	0.81	0.62	0.58	0.60
French	0.77	0.75	0.76	0.50	0.48	0.49
Turkish	0.91	0.88	0.89	0.65	0.61	0.63

Table 3: Sample Segmentation (French) for the Word *internationalisation*

Method	Segmentation
HELOS	(((inter)(nation))(al))(isation)
Morfessor	inter-nation-al-is-ation
Benchmark	(inter)(nation)(al)(is)(ation)

HELOS's hierarchical parses demonstrate that a Free-Energy–style minimisation over binary trees can successfully recover linguistically meaningful morpheme boundaries without any manual guidance. Across typologically distinct languages—German, French, Russian, and Turkish—HELOS finds nested structures (e.g., "ge + seh + en" in German "gesehen" and agglutinative suffix sequences in Turkish) that align closely with gold-standard segmentations (F1 scores in the 0.80–0.90 range). This confirms that framing morpheme induction as a search for low free-energy binary trees is enough to produce coherent, interpretable segmentations in very different morphological systems.

5.2 Co-occurrence-Based Symbol Clustering

In code (`fep_ph_clusterer.py`), clustering is implemented by constructing an undirected adjacency graph over symbols based on their co-occurrence frequencies in datasets`/dicts/fep_ph_clusterer/plain_strings.txt`. This dataset contains sequences combining Cyrillic, Latin, and punctuation characters. The clustering process follows these steps:

- Count Adjacent Co-occurrences: The program iterates over every input string, for each adjacent pair of characters (c_i, c_{i+1}) incrementing a count $N(c_i, c_{i+1})$ (and symmetrically $N(c_{i+1}, c_i)$). This yields a weighted adjacency structure capturing how often each pair appears side by side.
- Threshold Filtering: For each symbol, any neighbour whose co-occurrence count meets or exceeds a predefined frequency threshold is considered directly connected. In practice, the threshold is set so that very frequent transitions (e.g., spaces next to letters) produce edges in the graph.
- Connected Components (Topology): The adjacency graph is treated as a 0-dimensional topological complex; finding connected components corresponds to computing the 0th homology (connected clusters). A breadth-first search (BFS) or union-find algorithm extracts each maximal group of mutually reachable symbols.
- Cluster Assignment: Each connected component is assigned a unique cluster ID. Symbols not meeting any edge criteria remain isolated or form singleton clusters.

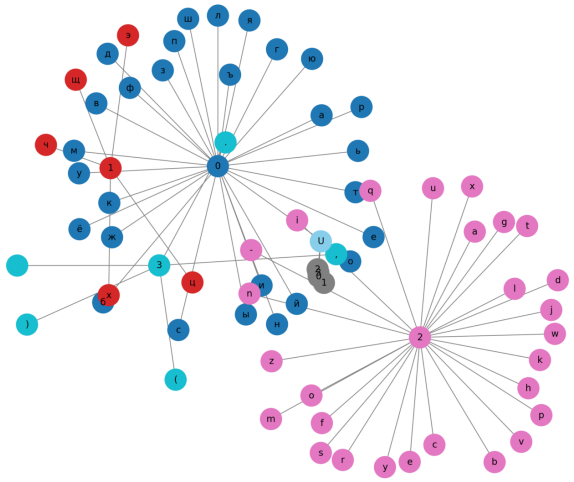


Figure 5: figure
Clusters Visualisation

Because clusters arise from the connectivity structure of co-occurrence counts, this method embodies a simple topological approach: symbols belong to the same cluster if there exists a path of high-frequency transitions between them. Although this is not explicitly framed in variational or Free Energy terms, it leverages the same principle of discovering stable groupings (connected components) in a weighted adjacency space.

Purity of the resulting clustering (measured by dominant-label agreement within each cluster) reached **0.907**, indicating that the system spontaneously discovers linguistically plausible groupings. Full cluster descriptions, sizes, co-occurrence patterns, and frequency statistics are summarised in Table 4.

Table 4: Clustering Results

Cl. ID	Content (excerpt)	Size	Total Occ.	Top Co-occurrences
0	а, б, в, г, д, е, ж, з, и, й, …	29	8281	→ 3 (2729), → 0 (13562), → 1 (197)
1	х, ц, ч, щ, э	5	179	→ 0 (197), → 3 (157)
2	a, b, c, d, …, z, –	28	6392	→ 3 (4010), → 2 (8664), → U (9)
3	’ ’, ’(’, ’)’, ’,’, ’.’	5	14361	→ 0 (2729), → 2 (4010), → 3 (4850)
—	Avg. Purity = 0.907			

The clustering's purity of 0.907 means that, for over 90% of symbols, their assigned cluster matches the most frequent true category (Cyrillic, Latin, or punctuation) that they belong to. In practice, this confirms that the purely co-occurrence-based, topological approach successfully recovers intuitive groupings: core Cyrillic letters, marked Cyrillic variants, Latin/hyphen group, and whitespace/punctuation, despite no explicit orthographic supervision.

These results demonstrate that—even with a simple adjacency-counting and connected-components procedure—one can recover highly coherent, orthographically meaningful clusters from raw symbol sequences. Such emergent symbol-level clusters not only corroborate our theoretical framework (discovering stable, high-frequency linkages) but also serve as the first step toward building richer hierarchical models of text under FEP-inspired learning.

The experiments provide evidence that the hierarchical predictive architecture underlying HELOS produces measurable structure without explicit supervision. We interpret these emergent phenomena as concrete approximations of Free Energy minimisation in active generative models.

6 Discussion

This section discusses the broader implications of our findings, acknowledges the limitations of the current work, and outlines promising avenues for future research, positioning our framework within the larger landscape of AI, NLP, and cognitive science.

6.1 Summary of Findings and Emergent Structure

This work introduced and provided initial computational validation for a hierarchical meta-modelling framework aimed at the unsupervised discovery of linguistic structure. The core hypothesis, that complex grammatical organisation can arise from applying a universal, compositional mechanism driven by predictive optimisation (approximating FEP) to sequential data, was tested via a prototype system. The system, utilising a recursive binary representation 4 and an iterative reduction algorithm operating on character sequences successfully demonstrated several key capabilities.

Most significantly, the experiments detailed in Section 5 provide compelling evidence for the emergent discovery of morphological structure. Without access to any predefined morpheme lexicon or grammatical rules, the system autonomously identified recurring sub-structures corresponding to known linguistic morphemes (roots like *star-*, *khod-*, *mat(er)-*; affixes like *-yor*, *-nn-*, *-sk-*, *-ost'*, *-yj/-ij*, *pere-*, *pri-*) within Russian, Turkish, French and German words. These units consistently achieved high posterior probability (high 'log_prior', indicating low complexity) within the system's learned memory (M), signifying their status as statistically robust and predictively efficient building blocks.

Furthermore, the system demonstrated its ability to construct hierarchically appropriate binary parse trees for words by compositionally combining these emergent morphemic units. The optimal parse trees (T^*), selected via the FEP-like beam search minimising Eq. 5, frequently coincided with linguistically accurate morphological analyses, successfully resolving ambiguities that simpler heuristics struggled with. Crucially, the framework exhibited compositional generalisation, correctly parsing previously unseen words (e.g., *perekhodnyj*) by combining morphemes learned exclusively from the training data. This confirms that the system learns reusable structural components rather than merely memorising surface forms.

In summary, our findings support the central claim that applying principles of predictive optimisation within a hierarchically compositional framework allows for the spontaneous emergence of structured linguistic representations (morphology) from unannotated sequential input, demonstrating a powerful mechanism for unsupervised structure learning.

6.2 On the Assumption of Binary Composition

A core architectural choice of our framework is the reliance on recursive binary composition, as formalised in the structure S (Eq. 4). While this choice provides significant computational advantages and aligns with influential linguistic theories like Minimalism (19; 20) which posit binary 'Merge' as fundamental, we acknowledge that the absolute universality of binary branching across all linguistic phenomena and levels remains a subject of ongoing debate within theoretical linguistics [e.g., (75; 76), concerning coordination, complex predicates, etc.].

This work does not aim to definitively resolve these linguistic debates. Instead, we adopt strict binarity as a **minimalist yet powerful inductive bias** for our computational model. Our central claim is not that language *is* exclusively binary in some ontological sense, but rather that a computational system built upon this **minimal and universal binary combinatorial mechanism**, when guided by principles of predictive efficiency (FEP), is **sufficiently expressive and effective** to account for the **emergent discovery of complex, hierarchical linguistic structure**, including morphology and potentially syntax, across typologically diverse languages. The empirical success of our prototype in learning morphemic structures and generalising compositionally, despite operating solely with binary representations, serves as strong evidence for the **computational adequacy and generative power** of this minimalist, binary-grounded approach for modelling core aspects of language structure and acquisition. Whether non-binary operations offer significant additional explanatory power within this FEP-driven emergent framework remains an open question for future investigation.

6.3 Theoretical Implications and Predictions

The proposed hierarchical meta-modelling framework, grounded in binary composition and FEP-like optimisation, generates several strong, empirically testable predictions regarding the nature of language structure, processing, and acquisition:

1. **Ubiquity of Binary Hierarchies:** The framework predicts that optimal cognitive and computational representations of linguistic structure (across morphology and syntax) will predominantly favour binary-branching hierarchies. Apparent n-ary branching structures are predicted to be epiphenomenal or derived from underlying binary compositions that minimise the free energy functional ($F(T, w)$).
2. **Emergent Discretised Levels:** Language structure is predicted to be organised into quantifiable, discrete levels of abstraction (e.g., attributed characters $\rightarrow$ morphemic units $\rightarrow$ lexical categories $\rightarrow$ phrasal constituents) that

emerge sequentially through the iterative reduction process as stable fixed points or attractors in the FEP optimisation landscape.

3. **Morphemes as Optimal Predictive Chunks:** The framework predicts that morphemes correspond to those sub-tree structures within the Catalan space of possible parses that achieve an optimal balance between representational complexity (high prior probability $p(N_k)$, reflecting frequency and reuse) and predictive accuracy (low surprise $-\log p(w|T)$ in explaining local grapheme sequences and predicting context).
4. **Predictive Parsing Dynamics:** Language processing (parsing) is predicted to operate iteratively and hierarchically, heavily relying on top-down predictions generated by higher-level structural hypotheses to constrain the interpretation and guide the combination of lower-level units. Prediction error signals are predicted to be the primary driver of belief updating and structural inference.
5. **Frequency and Predictability Effects:** Processing efficiency (e.g., reading times, parsing speed) is predicted to be sensitive to the prior probability ($p(N_k)$) and predictive reliability of the constituent structural nodes (morphemes, constructions). High-frequency, highly predictive units should facilitate faster and more robust processing.
6. **Hypothesis Competition in Ambiguity:** Ambiguity resolution is predicted to involve the parallel maintenance and evaluation (based on $F(T, w)$) of multiple structural hypotheses (parse trees), with context (both local and global) biasing the competition towards the interpretation that minimises overall free energy.
7. **Structure Emergence Precedes Explicit Rules:** The acquisition of linguistic structure (morphology, syntax) is predicted to proceed via the emergent discovery of predictive regularities and stable compositional chunks, rather than the explicit learning or internalisation of abstract grammatical rules. Rules are descriptive labels for emergent optima in the FEP landscape.
8. **Cross-Linguistic Mechanism, Language-Specific Results:** The core computational mechanism (binary composition, FEP-guided reduction) is predicted to be universal, while the specific inventory of emergent structural units (N_k) and their parameters ($p(N_k)$, predictors) will be language-specific, reflecting the statistics of the input.
9. **Modality-General Hierarchical Abstraction:** The framework predicts that similar iterative, hierarchical, FEP-driven mechanisms for structure discovery and representation operate in non-linguistic modalities (e.g., vision, audition).
10. **Structurally Grounded Multimodality:** Cross-modal understanding and symbol grounding are predicted to arise not from simple associations, but from learning structural mappings (morphisms or functors) between the hierarchical representations generated by the universal mechanism in different modalities.

These predictions offer a rich set of hypotheses for future computational modelling, psycholinguistic experiments, and neuroimaging studies aimed at understanding the fundamental principles of language processing and learning.

6.4 Advantages and Distinctions of the Meta-Modelling Framework

The proposed hierarchical meta-modelling framework offers several distinct advantages and represents a significant departure from prevalent approaches in language modelling and unsupervised structure learning:

1. **Explicit Structural Representation and Interpretability:** Unlike distributed representations in LLMs, our framework constructs explicit, hierarchical parse trees (T^*) composed of identifiable nodes (S) representing discovered linguistic units (e.g., morphemes). This symbolic, compositional structure provides inherent interpretability, allowing the analysis process and the learned knowledge (node parameters in M) to be directly inspected and understood (77; 78).
2. **Emergent Discovery from First Principles:** The system learns structure, including fundamental units like morphemes and potentially word categories, emergently through a unified optimisation process (FEP-like minimisation) rather than relying on predefined lexicons, rules, or purely statistical surface patterns. This offers a more fundamental, knowledge-lean approach to structure acquisition.
3. **Strong Inductive Bias towards Hierarchy:** The reliance on recursive binary composition ($S \cong (L \times I) + (S \times I)$) provides a strong, linguistically motivated inductive bias that facilitates the discovery of hierarchical constituency, a core property of human language structure often implicitly or weakly captured by sequence-to-sequence models.
4. **Demonstrated Compositional Generalisation:** The prototype's ability to correctly parse unseen words by combining previously learned morphemic nodes provides concrete evidence of compositional generalisation, a capability crucial for human-like language processing but often challenging for models relying solely on distributional similarity.
5. **Potential for Data Efficiency:** The strong structural bias and the focus on discovering reusable, abstract units suggest a potential for greater data efficiency compared to large neural models that require vast corpora to implicitly learn similar structural regularities. Our promising results on relatively small datasets support this hypothesis (33).

6. **Inherent Support for Lifelong Learning:** Furthermore, the proposed framework exhibits properties conducive to lifelong learning, mitigating the problem of catastrophic forgetting often encountered in purely connectionist systems (79; 80; 81). This stems from several core architectural features:

   - Incremental Knowledge Representation: Learning primarily involves adding new structural nodes N_k (representing discovered patterns or morphemes) to the global registry M and refining their associated parameters ($\log p(N_k)$, predictors), rather than globally retraining the entire model. Previously learned, well-established nodes are preserved.
   - Compositional Reuse: New inputs are analysed by compositionally reusing existing high-probability nodes from the registry. This constant reactivation reinforces robustly learned structures and prevents their decay.
   - Local Parameter Updates: The Bayesian-inspired update mechanism modifies parameters primarily for nodes involved in the parse of the current input, avoiding drastic interference with unrelated knowledge encoded in other nodes.
   - Stability of Core Structures: Nodes corresponding to fundamental and frequent linguistic units rapidly converge to stable, high-probability states (high $\log p(N_k)$), forming a robust core lexicon/grammar resistant to forgetting. Adaptation to novelty occurs mainly through the addition of new nodes or refinement of predictive models rather than overwriting the core.

   Consequently, the system architecture inherently supports continuous, incremental learning from new data without catastrophic disruption of previously acquired structural knowledge. The framework's architecture, featuring an incrementally growing node registry (M) and localised parameter updates driven by prediction error, naturally mitigates catastrophic forgetting and supports continuous, lifelong learning without requiring global retraining. This positions it as a robust neurosymbolic system.
7. **Principled Path to Grounding and Multimodality:** The universality of the compositional mechanism and the abstract nature of the representation (S) provide a principled foundation for multi-modal integration, allowing structural correspondences to be learned between language and other modalities, directly addressing the symbol grounding problem (70; 82).

In essence, the framework shifts the focus from learning statistical co-occurrences in flat sequences to discovering and composing hierarchical, predictive, structural models, offering potential advantages in interpretability, generalisation, data efficiency, and lifelong learning.

6.5 Modelling Complex Phenomena with Binary Structures: The Case of Circumfixation

A potential objection to frameworks relying strictly on binary composition is their ability to adequately model linguistic phenomena that appear non-binary, such as circumfixation (e.g., German Past Participles like *ge-spiel-t* from *spiel-en*). We argue that our FEP-driven mechanism operating over the recursive structure S can naturally account for such cases through learned predictive dependencies, without requiring n-ary branching rules.

Consider the competing binary parse trees for *gespielt* (grapheme sequence w):

1. $T_1 = (((N_{spiel}, i_t)), i_{ge})$ (Suffix attached first, then prefix)
2. $T_2 = (((N_{spiel}, i_{ge}), i_t))$ (Prefix attached first, then suffix)
3. $T_3 = (N_{spiel}, i_{ge..t})$ (A single complex circumfix model $i_{ge..t}$ applied)

The system selects the tree T^* that minimises $F(T, w) \approx \sum_{N_k \in Sub(T)} [-\log p(N_k)] + \sum_j S_j$, where S_j is the local surprise (prediction error, e.g., $-\log p(\text{child}|\text{sibling/parent})$) at merge step j.

Through learning (updating node priors $p(N_k)$ and predictors based on minimising F across examples), the system is expected to converge to favour T_1:

- **Complexity:** The components N_{spiel} (root), i_t (common suffix), and i_{ge} (common prefix element) will acquire high prior probabilities $p(N_k)$ (low complexity, high 'log_prior') due to their frequent occurrence in various contexts. The intermediate node $N_{spiel\_t} = (N_{spiel}, i_t)$ might also gain moderate probability. Conversely, the specialised circumfix model $i_{ge..t}$ (for T_3) or the linguistically less motivated intermediate $N_{ge\_spiel} = (N_{spiel}, i_{ge})$ (for T_2) are likely to remain low-probability (high complexity). Thus, $Complexity(T_1)$ is likely to be lower than $Complexity(T_3)$ and potentially $Complexity(T_2)$.
- **Surprise:** Crucially, the predictive models will learn the dependency. The predictor for N_{spiel_t} (representing *spiel-t*) will learn to strongly predict the presence of i_{ge} in the Partizip II context (low surprise $S_{next}(i_{ge}|N_{spiel\_t})$). Similarly, the predictor for N_{spiel} will predict i_t in this context (low surprise $S_{next}(i_t|N_{spiel})$). In contrast, the prediction of i_{ge} directly from N_{spiel} (required for T_2) might be weaker, leading to higher surprise.

- **Result:** The combination of relatively low component complexity and low predictive surprise (due to learned dependencies like 'spiel-t → ge-') makes T_1 the most probable parse (T^*), minimising the overall Free Energy $F(T_1, w)$.

This example illustrates that the framework can model complex morphotactics like circumfixation adequately through sequential binary composition, where the correct ordering and dependencies are enforced not by structural branching rules, but by the learned predictive models within the nodes, guided by the global FEP optimisation. The system discovers the most efficient compositional *pathway* using binary steps.

6.6 Limitations and Future Directions

While the presented framework and initial prototype results offer compelling support for the core principles of HELOS, several limitations inherent in the current stage of research must be acknowledged, alongside the crucial directions for future work they delineate. These represent not refutations, but rather the key challenges and open questions that define the ongoing research programme.

1. **Universality and Expressiveness of Binary Composition:** The framework fundamentally relies on recursive binary composition, formalised via the structure S (Eq. 4), as the sole structure-building mechanism. While this provides a powerful minimalist inductive bias demonstrated effective for morphology, its empirical adequacy and computational optimality for representing *all* linguistic phenomena across all levels (e.g., complex coordination, potentially certain semantic or discourse structures) remains an open question warranting further investigation (75; 76). Future work should explore whether incorporating controlled extensions to the structure S or the composition rules offers significant advantages in modelling specific complex constructions within the FEP framework, or if sequential binary operations coupled with learned predictive dependencies remain sufficient.
2. **Homogeneity of the Meta-Modelling Mechanism:** HELOS posits a unified FEP-driven meta-modelling engine operating recursively across linguistic levels. While theoretically elegant, it remains an empirical question whether distinct cognitive modules or optimal computational systems might employ qualitatively different representations or specialised mechanisms for different levels (e.g., phonology vs. semantics vs. pragmatics). The computational efficiency and cognitive plausibility of applying the exact same search over S structures via FEP minimisation to the vastly different scales and types of regularities present in syntax, semantics, and discourse requires comprehensive validation.
3. **Computational Tractability of FEP/EFE Optimisation:** The practical realisation of HELOS hinges on the ability to efficiently approximate the minimisation of the Free Energy F (Eq. 5) or Expected Free Energy EFE (Eq. 8) over the potentially vast Catalan space $\mathcal{T}_w$ of parse trees. The current beam search is approximate and may be susceptible to local optima. Significant future work is required to develop and analyse more sophisticated, scalable inference algorithms (e.g., advanced variational methods, structure-specific MCMC, refined heuristic searches) specifically tailored to optimising the FEP objective over the recursive structure S for large-scale linguistic data. A complete implementation of the variational free energy framework remains an open challenge and a direction for future development.
4. **Implicit Biases vs. Pure Emergence:** While striving for emergence from minimal priors, the framework inevitably incorporates implicit inductive biases through the specific definition of S, the precise mathematical formulation chosen to approximate the Complexity and Surprise terms in the FEP functional, and the details of the inference algorithm. Rigorous analysis, including ablation studies, is needed to disentangle the contribution of these inherent biases from the pure self-organising dynamics driven by FEP optimisation, and to assess the universality and cognitive validity of these implicit architectural choices.
5. **TDA Stability as an FEP Heuristic:** The use of TDA-based stability criteria for triggering symbolisation in the emergent categorisation stage (Level 0) is currently employed as a principled heuristic. While conceptually linked to identifying moments where abstraction is FEP-optimal (balancing complexity reduction and predictive accuracy), this connection requires deeper theoretical grounding and potential comparison with alternative metrics derived more directly from gradients of the FE functional itself. The robustness and generalisability of the specific variance-based criterion also merit further study.
6. **Operationalising the Topos Formalism:** The formalisation of the framework within a cognitive topos (Section 4.12) provides significant theoretical depth and coherence. However, translating the full power of this formalism – particularly the internal logic, sheaf semantics, and the dynamics of geometric morphisms for learning and adaptation – into concrete computational mechanisms and demonstrating their unique practical advantages remains a substantial undertaking for future research.
7. **Comprehensive Empirical Validation:** The current empirical validation focuses primarily on Level 0 and Level 2 (morphology, Section 6). Comprehensive validation requires scaling the experiments to larger, more diverse corpora and rigorously demonstrating the framework's capabilities at Level 1 (segmentation), Level 3 (PoS/paradigm induction), Level 4 (syntax), and beyond, including quantitative comparisons against a wider range of state-of-the-art unsupervised and supervised systems on established benchmarks.

8. **FEP Approximation and Predictor Sophistication:** The current implementation relies on simplified proxies for the Free Energy (F) calculation and Bayesian updates (e.g., using node priors derived from frequency/success signals and basic frequency-based predictors). A more rigorous implementation employing variational inference methods and more sophisticated predictive models (potentially small neural networks associated with nodes S) is necessary to fully realise the theoretical power of the FEP and likely improve parsing accuracy, especially for resolving complex ambiguities (83; 67).
9. **Parse Search Efficiency and Optimality:** The beam search algorithm (`find_best_parses_fep`), while avoiding Catalan complexity, is still heuristic and may not guarantee finding the true global FE minimum (T^*). Exploring more advanced search techniques (e.g., best-first search fully guided by predictors, particle filtering, or MCMC methods adapted for trees) is warranted (68; 8).
10. **Scalability:** The size of the node registry (M) grew significantly even with limited vocabulary. Strategies for managing registry size, such as pruning low-probability nodes, introducing node abstraction hierarchies, or more efficient hashing/storage, will be essential for scaling to large corpora and vocabularies.
11. **Extension to Syntax and Semantics:** The immediate and most crucial next step is to apply the *same* hierarchical meta-modelling principles to sequences of word/morpheme nodes (l_{final}) to emergently discover syntactic structures (phrase structures, dependency relations, part-of-speech categories) (50). Subsequently, investigating how semantic representations can be integrated or emerge within this compositional, hierarchical framework is a major research direction.
12. **Multi-Modal Grounding:** Realising the framework's potential for symbol grounding requires implementing the mechanism for a non-linguistic modality (e.g., vision, modelling object parts and scenes hierarchically using the same S structure) and developing methods for learning the structural mappings (morphisms/functors) between the linguistic and perceptual hierarchies.
13. **Richer Input Features:** While demonstrating success with minimal initial features (emergent properties), incorporating richer input (e.g., phonological features for speech, contextual embeddings from pre-trained models as initial node attributes) could potentially accelerate learning or improve accuracy.

Addressing these challenges constitutes a rich research programme. However, the fundamental principles of emergent hierarchical structure discovery through FEP-guided binary composition, validated here for morphology, provide a robust and theoretically compelling foundation for these future endeavours.

7 Broader Impact and Conclusion

This paper introduced a hierarchical meta-modelling framework, HELOS, grounded in binary composition and principles of predictive optimisation (FEP), proposing a novel pathway for the unsupervised, emergent discovery of linguistic structure. We have provided the theoretical foundations, outlined the computational mechanisms, and presented compelling initial results from a prototype system demonstrating its capacity to autonomously learn morphological units and perform compositional analysis, including generalisation to unseen words, from limited, unannotated data across different language types.

While the current work focused primarily on validating the core principles at the morphological level with approximate methods, the potential implications extend far beyond. The true significance lies in the demonstration of a potentially universal, cognitively plausible mechanism for self-supervised structure learning. If the core principles of iterative, FEP-guided, compositional abstraction hold true for higher levels of language (syntax, semantics) and potentially other cognitive domains (like vision), this framework could offer a unifying perspective on how complex, symbolic-like knowledge representations arise from sub-symbolic predictive processing.

This could pave the way for a new generation of AI systems that are more interpretable (due to explicit hierarchical structure), data-efficient (leveraging strong structural priors), robust (handling ambiguity via hypothesis evaluation), adaptive (capable of lifelong learning without catastrophic forgetting), and ultimately, better equipped for structurally grounded reasoning and multi-modal understanding.

We emphasise that this work represents the beginning of a larger research programme, not a final, complete implementation. Significant challenges remain, particularly in developing more rigorous FEP calculations, scalable search algorithms, and extending the framework comprehensively to syntax, semantics, and cross-modal learning. However, the primary goal of this paper – to present the core theoretical framework and experimentally validate its foundational premise (that meaningful linguistic structure like morphology can indeed emerge from these principles) – has, we hope, been substantially achieved. We believe this principled, structure-building approach offers a promising and fundamentally different direction for future research into the computational nature of language and intelligence.

8 Acknowledgements

The author wishes to express sincere gratitude for the invaluable assistance in preparing this paper and the foundational contributions to this line of research provided by **Alexander Nikolaevich Burdukov**, a Russian engineer and independent AI researcher, and **Dmitry Ivanovich Sviridenko**, Doctor of Physical and Mathematical Sciences, Professor at the Department of Algebra and Logic, Associate Professor at the Department of General Informatics, and Professor at the Department of Discrete Mathematics and Informatics at Novosibirsk State University. Their pioneering work and insights into meta-modelling and task-based approaches served as a crucial foundation and inspiration for the framework presented herein. Any remaining errors or shortcomings in this paper are solely the responsibility of the author.

A Formal Coq Verification of S

A.1 Coq Implementation

```
Require Import Coq.Init.Datatypes.
Require Import Coq.Lists.List. Import ListNotations.
Require Import Coq.Logic.ProofIrrelevance.

Parameter I : Type.

Inductive S_list : Type :=
  | C1_list : I -> I -> S_list
  | C2_list : list (sum I S_list) -> I -> S_list
  | C3_list : S_list -> I -> S_list.

Definition RHS_list_type : Type :=
  sum (prod I I) (sum (prod (list (sum I S_list)) I) (prod S_list I)).

Definition slist_to_rhs (s_val : S_list) : RHS_list_type :=
  match s_val with
  | C1_list i1 i2    => inl (pair i1 i2)
  | C2_list l_is i   => inr (inl (pair l_is i))
  | C3_list s_rec i  => inr (inr (pair s_rec i))
  end.

Definition rhs_to_slist (rhs_val : RHS_list_type) : S_list :=
  match rhs_val with
  | inl (pair i1 i2)              => C1_list i1 i2
  | inr (inl (pair l_is i))       => C2_list l_is i
  | inr (inr (pair s_rec i))      => C3_list s_rec i
  end.

Lemma slist_to_rhs_circ_rhs_to_slist : forall (r : RHS_list_type), slist_to_rhs (
    rhs_to_slist r) = r.
Proof. intros r; destruct r as [ [i1 i2] | [ [l_is i_prod1] | [s_rec i_prod2] ] ]; simpl
    ; reflexivity. Qed.

Lemma rhs_to_slist_circ_slist_to_rhs : forall (s_val : S_list), rhs_to_slist (
    slist_to_rhs s_val) = s_val.
Proof. intros s_val; destruct s_val; simpl; reflexivity. Qed.

Record iso (A B : Type) : Type := mk_iso {
  to      : A -> B;
  from    : B -> A;
  to_from : forall (b : B), to (from b) = b;
  from_to : forall (a : A), from (to a) = a
}.
```

```
Notation "A <~> B" := (iso A B) (at level 70).

Definition Slist_iso_RHS_v2 : S_list <~> RHS_list_type.
Proof.
  refine {| to      := slist_to_rhs;
            from    := rhs_to_slist;
            to_from := _;
            from_to := _
         |}.
  - (* forall b : RHS_list_type, slist_to_rhs (rhs_to_slist b) = b *)
    exact slist_to_rhs_circ_rhs_to_slist.
  - (* forall a : S_list, rhs_to_slist (slist_to_rhs a) = a *)
    exact rhs_to_slist_circ_slist_to_rhs.
Defined.

Print Slist_iso_RHS_v2.
```

A.2 Proof Tree of the Isomorphism S_list ⇜⇝ RHS_list_type

```
Equiv S_list <~> RHS_list_type
|- to       : S_list → RHS_list_type
|  \_ slist_to_rhs
|- from     : RHS_list_type → S_list
|  \_ rhs_to_slist
|- to_from  : x:RHS_list_type, slist_to_rhs (rhs_to_slist x) = x
|  \_ slist_to_rhs_circ_rhs_to_slist
\_ from_to  : x:S_list, rhs_to_slist (slist_to_rhs x) = x
   \_ rhs_to_slist_circ_slist_to_rhs
```

B Comparison with JEPA

While HELOS and JEPA (Joint Embedding Predictive Architecture, (84)) share a broad philosophical commitment to predictive processing and the minimisation of uncertainty, they differ fundamentally in scope, formalism, and function. The following table highlights the main distinctions:

Aspect	**JEPA**	**HELOS (This Work)**
Theoretical Base	Predictive coding; latent code forecasting; contrastive objectives	Free Energy Principle (FEP); active inference; structural compression
Objective Function	Alignment between predicted and target latent representations	Minimisation of internal free energy F over structural hypotheses T^*
Output Space	Vector embeddings in latent space	Interpretable hierarchical structures (e.g., morphological parse trees)
Learning Paradigm	Predictive representation learning	Emergent abstraction and structural meta-modelling
FEP Usage	Philosophical inspiration; not formally implemented	Formal and operational: FEP explicitly minimised through structure search
Reasoning	Not addressed	Modelled as structure-driven inference minimising expected free energy (EFE)
Generalisation	Implicit via embedding smoothness	Compositional and structural generalisation from few examples
Universality	Architecture-specific	Meta-modelling framework applicable to language, perception, and cognition
Interpretability	Low; vector spaces opaque	High; explicit structural representations and energy landscape visible

Table 5: Comparison of HELOS and JEPA.

Although JEPA represents an important development in predictive learning, HELOS can be seen as a higher-order generalisation: it unifies abstraction, structure discovery, and reasoning into a single emergent process governed by FEP. Where JEPA predicts latent states, HELOS constructs structured hypotheses about observed data and selects among them based on epistemic economy.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017, pp. 5998–6008.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[3] N. Chomsky, *Syntactic Structures*. Mouton, 1957.

[4] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass, "On the linguistic representational power of neural machine translation models," *Computational Linguistics*, vol. 47, no. 3, pp. 509–544, 2021.

[5] E. M. Bender and A. Koller, "Climbing towards nlu: On meaning, form, and understanding in the age of data," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 5185–5198.

[6] G. Marcus, "The next decade in ai: Four steps towards robust artificial intelligence," *arXiv preprint arXiv:2002.06177*, 2020.

[7] N. Chomsky, *Aspects of the Theory of Syntax*. MIT Press, 1965.

[8] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Prentice Hall, 2020.

[9] K. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.

[10] A. Clark, "Whatever next? predictive brains, situated agents, and the future of cognitive science," *Behavioral and Brain Sciences*, vol. 36, no. 3, pp. 181–204, 2013.

[11] T. Kouwenhoven *et al.*, "Searching for structure: Investigating emergent communication with large language models," *arXiv preprint arXiv:2412.07646*, 2024. [Online]. Available: https://arxiv.org/abs/2412.07646

[12] S. Kirby, "The emergence of linguistic structure: An overview of the iterated learning model," *Linguistic Evolution through Language Acquisition*, pp. 121–147, 2002. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4471-0663-0_6

[13] P. Mazzaglia, T. Verbelen, O. Çatal, and B. Dhoedt, "The free energy principle for perception and action: A deep learning perspective," *Entropy*, vol. 24, no. 2, p. 301, 2022. [Online]. Available: https://www.mdpi.com/1099-4300/24/2/301

[14] G. Deza, K. Mahowald, M. H. Frank, C. Potts, and N. D. Goodman, "Emergent linguistic structure in artificial neural networks trained by self-supervision," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 071–30 078, 2020. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1907367117

[15] S. Miyagawa, R. C. Berwick, and K. Okanoya, "The emergence of hierarchical structure in human language," *Frontiers in Psychology*, vol. 4, p. 71, 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3577014/

[16] S. S. Goncharov and D. I. Sviridenko, "Matematicheskie osnovy semanticheskogo programmirovaniya [mathematical foundations of semantic programming]," *Doklady Akademii Nauk SSSR [Soviet Mathematics Doklady]*, vol. 289, no. 6, pp. 1324–1328, 1986.

[17] Y. L. Ershov, *Definability and Computability*, ser. Siberian School of Algebra and Logic. Plenum, 1996.

[18] S. S. Goncharov and D. I. Sviridenko, "Rekursivnye termy v semanticheskom programmirovanii [recursive terms in semantic programming]," *Sibirskii Matematicheskii Zhurnal (Siberian Mathematical Journal)*, vol. 59, no. 6, pp. 1279–1290, 2018.

[19] N. Chomsky, *The Minimalist Program*. MIT Press, 1995.

[20] N. Hornstein, J. Nunes, and K. K. Grohmann, *Understanding Minimalism*. Cambridge University Press, 2009.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: The MIT Press, 2009, cited page: 372.

[22] K. Friston, "A theory of cortical responses," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1456, pp. 815–836, 2005.

[23] R. J. Solomonoff, "A formal theory of inductive inference. part i," *Information and Control*, vol. 7, no. 1, pp. 1–22, 1964.

[24] ——, "A formal theory of inductive inference. part ii," *Information and Control*, vol. 7, no. 2, pp. 224–254, 1964.

[25] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. Springer, 2008.

[26] R. S. Kayne, *The Antisymmetry of Syntax*. MIT Press, 1994.

[27] I. Heim and A. Kratzer, *Semantics in Generative Grammar*. Blackwell, 1998.

[28] L. Tesnière, *Éléments de syntaxe structurale*. Klincksieck, 1959.

[29] I. A. Melčuk, *Dependency Syntax: Theory and Practice*. SUNY Press, 1988.

[30] R. K. Larson, "On the double object construction," *Linguistic Inquiry*, vol. 19, no. 3, pp. 335–391, 1988.

[31] J. Hohwy, *The Predictive Mind*. Oxford University Press, 2013.

[32] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[33] P. D. Grünwald, *The Minimum Description Length Principle*. MIT Press, 2007.

[34] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the Workshop on Morphological and Phonological Learning (ACL)*, 2002, pp. 21–30.

[35] ——, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 4, no. 1, p. 3, 2007.

[36] S. Goldwater, T. L. Griffiths, and M. Johnson, "A bayesian framework for word segmentation: Exploring the effects of context," in *Cognition*, vol. 112, no. 1, 2009, pp. 21–54.

[37] K. Sirts and J. Penjam, "Overview of the morpho challenge 2005-2010," in *Proceedings of the Workshop on Statistical Parsing of Morphologically Rich Languages*, 2014, pp. 132–142.

[38] C. de la Higuera, *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, 2010.

[39] A. Clark and C. Fox, *Computational Learning Theory*, 01 2010.

[40] A. Kuncoro, M. Ballesteros, L. Kong, C. Dyer, G. Neubig, and N. A. Smith, "What do recurrent neural network grammars learn about syntax?" in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017, pp. 128–138.

[41] A. Drozdov, P. Verga, M. Yadav, M. Iyyer, and A. McCallum, "Unsupervised latent tree induction with deep inside-outside recursive auto-encoders," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1119–1130.

[42] A. Stolcke and S. M. Omohundro, "Bayesian learning of probabilistic language models," in *ICASSP-94*, vol. 1, 1994, pp. I/73–I/76.

[43] M. Johnson, T. L. Griffiths, and S. Goldwater, "Bayesian inference for pcfgs via markov chain monte carlo," *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, p. 139–146, 2007.

[44] B. C. Pierce, *Types and Programming Languages*. MIT Press, 2002.

[45] R. Harper, *Practical Foundations for Programming Languages*, 2nd ed. Cambridge University Press, 2016.

[46] E. E. Vityaev, "Semanticheskii veroyatnostnyi vyvod predskazanii [semantic probabilistic inference of predictions]," *Izvestiya Irkutskogo Gosudarstvennogo Universiteta. Seriya Matematika [The Bulletin of Irkutsk State University. Series Mathematics]*, vol. 21, pp. 33–50, 2017.

[47] Y. L. Ershov, *Topologiya dlya diskretnoi matematiki [Topology for discrete mathematics]*. Novosibirsk: Sobolev Institute of Mathematics SB RAS, 2020.

[48] P. Alexandroff, "Diskrete räume," *Matematicheskii Sbornik*, vol. 2(44), no. 3, pp. 501–518, 1937.

[49] E. E. Vityaev, S. S. Goncharov, and D. I. Sviridenko, "O zadachnom podkhode v iskusstvennom intellekte [on the task approach to artificial intelligence]," *Sibirskii filosofskii zhurnal [Siberian Journal of Philosophy]*, vol. 17, no. 4, pp. 5–25, 2019.

[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems 26*, 2013.

[51] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[52] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019, pp. 218–226.

[53] J. Vig and Y. Belinkov, "Analyzing the structure of attention in a transformer language model," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019, pp. 63–76.

[54] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 4593–4601.

[55] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Identifying and controlling important neurons in neural machine translation," *arXiv preprint arXiv:1901.11253*, 2019.

[56] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and C. Voss, "An overview of circuit-based interpretability," 2020. [Online]. Available: https://distill.pub/2020/circuits/overview/

[57] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4129–4138.

[58] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2009.

[59] B. Preiss, *Data Structures and Algorithms with Object-Oriented Design Patterns in C++*. John Wiley & Sons, 1998.

[60] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 4129–4138. [Online]. Available: https://aclanthology.org/N19-1423

[61] D. Mareček and A. Rosa, "Extracting syntactic trees from transformer encoder self-attentions," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4605–4615. [Online]. Available: https://aclanthology.org/D18-1491

[62] Y. Zou and J. W. V. de Meent, "Hierarchical meta learning," 2019.

[63] H. Yao and F. Huang, "Hierarchically structured meta-learning," 2019.

[64] S. S. Goncharov. Recursive model theory. Encyclopedia of Mathematics. [Online]. Available: http://encyclopediaofmath.org/index.php?title=Recursive_model_theory

[65] K. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, pp. 127–138, 2010.

[66] T. Parr and K. J. Friston, "A tutorial on the free-energy framework for modelling perception and learning," *Journal of Mathematical Psychology*, vol. 76, pp. 131–154, 2017.

[67] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[68] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Prentice Hall, 2023.

[69] D. Klein and C. D. Manning, "Probabilistic context-free grammars, tree automata, and syntactic monoids," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 423–430.

[70] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[71] T. Parr and G. Pezzulo, "Understanding, explanation, and active inference," *Frontiers in Systems Neuroscience*, vol. 15, p. 772641, 2021.

[72] K. E. Stephan, Z. M. Manjaly, C. D. Mathys, L. A. Weber, S. Paliwal, T. Gard, M. Tittgemeyer, S. M. Fleming, H. Haker, and A. K. Seth, "Allostatic self-efficacy: A metacognitive theory of dyshomeostasis-induced fatigue and depression," *Frontiers in Human Neuroscience*, vol. 10, p. 550, 2016.

[73] M. Kuchling and W. Tschacher, "Active inference and abduction," *Frontiers in Psychology*, vol. 12, p. 654739, 2021.

[74] R. A. Adams, Q. J. M. Huys, and J. P. Roiser, "Active inference in psychology and psychiatry: Progress to date?" *Entropy*, vol. 26, no. 10, p. 833, 2024.

[75] P. W. Culicover and R. Jackendoff, *Simpler Syntax*. Oxford University Press, 2005.

[76] C. Pollard and I. A. Sag, *Head-driven phrase structure grammar*. University of Chicago Press, 1994, vol. 15.

[77] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[78] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[79] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[80] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[81] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[82] L. W. Barsalou, "Perceptual symbol systems," *Behavioral and brain sciences*, vol. 22, no. 4, pp. 577–609, 1999.

[83] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[84] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. G. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," *CoRR*, vol. abs/2301.08243, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.08243