The Imagination Machine XII: Reconstructing Conceptual Structure in an Open Text World

Mark Tracy

Abstract

The preceding papers in the Imagination Machine series develop a formal architecture for embedded epistemic systems. Observations generate world models through compression of relational structure, while extension predicts missing relations and guides action. In The Imagination Machine XI this architecture was realized as a graph-theoretic learning system whose world model is a dynamically updated knowledge graph.

The present paper introduces an experimental environment designed to test that architecture. An agent interacts with an open text world constructed from the TIM corpus itself. The agent observes only local textual segments, incrementally constructs a relational knowledge graph, compresses that graph through clustering, and predicts missing relations through extension. The resulting graph induces a clique complex whose simplicial structure provides a natural representation of higher-order conceptual relations.

Training occurs on TIM I–XI, while TIM XII is withheld as a test corpus. Evaluation measures the ability of the agent to predict structural updates induced by previously unseen text, including recovery of latent conceptual relations and completion of simplicial horns in the induced complex. To make the environment rigorous, the paper specifies an explicit latent concept vocabulary, an explicit typed relation vocabulary, a hybrid node-to-text and edge-to-text labeling protocol, and an explicit proposed ground-truth conceptual graph for the TIM world. The experiment therefore tests whether compression–extension dynamics allow an embedded agent to reconstruct relational invariants of its textual environment from within the textual surface alone.

1 Introduction

The Imagination Machine series develops a formal framework for epistemic systems embedded within their environment. Because such systems cannot access an external vantage point, knowledge must be defined operationally in terms of internal predictive coherence rather than correspondence with an independently accessible world.

The Imagination Machine XI introduced a computational realization of this framework. The agent's world model is represented as a knowledge graph whose nodes correspond to entities and whose edges represent relations extracted from observations. Learning proceeds through repeated cycles of:

(1) observation of new data,

(2) updating of the knowledge graph,

(3) compression of the graph through clustering of similar entities, and

(4) extension of the graph through prediction of missing relations.

These operations were shown to induce simplicial dynamics on the clique complex of the knowledge graph, linking the architecture to the simplicial completion conditions identified in earlier papers.

The present paper introduces an experimental setting designed to test this architecture empirically. The environment is an open text world constructed from the TIM corpus. The agent does not receive the corpus all at once. Instead, it receives local textual segments in response to actions taken with respect to its current world model. The environment is therefore partially observable, structured, and internally coherent, while remaining rich enough to support nontrivial graph reconstruction and simplicial completion tasks.

2 The Open Text World

Let

$$C = \{p_1, p_2, \ldots, p_N\}$$

be a corpus segmented into atomic textual units, such as paragraphs, definitions, theorems, remarks, or algorithm blocks. Each segment represents a possible observation.

The environment contains a latent conceptual graph

$$G^* = (V^*, E^*)$$

whose nodes represent concepts, operators, mathematical structures, and documents, and whose edges represent typed relations among them. This graph is not directly accessible to the agent.

The environment also contains a latent simplicial family

$$\Sigma^* \subseteq \mathcal{P}(V^*)$$

whose elements represent coherent higher-order conceptual configurations.

The agent interacts with the environment only through textual observations sampled from the segmented corpus in response to actions.

2.1 Atomic Observation Units

Each element $p_i \in C$ is one of the following:

- a paragraph,
- a formal definition,
- a theorem statement,
- a remark,
- an algorithm block, or
- a short subsection-introduction unit.

This segmentation ensures that observations are neither too fine-grained to support meaningful graph updates nor so coarse-grained that the environment degenerates into full-document access.

2.2 Training and Test Corpora

The training corpus is

$$C_{\text{train}} = \text{TIM I–XI},$$

and the test corpus is

$$C_{\text{test}} = \text{TIM XII}.$$

The agent trains on C_{train} alone. TIM XII is withheld during training and used only to evaluate whether the reconstructed world model predicts the structural updates induced by previously unseen text. In particular, the agent has no access to any segment of TIM XII during the training phase, and no node or edge whose sole textual evidence comes from TIM XII is available to the agent before evaluation.

3 Latent Ontology of the Environment

To make the environment explicit, we specify a finite node vocabulary and a finite typed relation vocabulary.

3.1 Node Vocabulary

The latent node set V^* is partitioned into four types.

Document Nodes

The following document nodes correspond to training-corpus papers and are available to the agent during training:

- TIM I
- TIM II
- TIM III
- TIM IV
- TIM V
- TIM VI
- TIM VII
- TIM VIII
- TIM IX
- TIM X
- TIM XI

The following document node corresponds to the withheld test corpus and is *not* available to the agent during training:

- TIM XII *(test corpus; withheld during training)*

Core Architecture Concept Nodes

- embedded epistemic system
- observation
- world model
- inference
- implication
- inference–implication loop
- epistemic closure
- fixed point
- compression
- extension
- action
- prediction error
- relational invariant
- internal supervision
- world-model update

Mathematical Structure Nodes

- graph
- graph morphism
- graph quotient
- graph completion
- clique
- clique complex
- simplex
- face map
- horn
- horn filling
- Kan condition
- simplicial dynamics

- quotient space
- equivalence relation
- classifier
- knowledge graph

Computational Architecture and Series-Thematic Nodes

- entity embedding
- relation tensor
- clustering
- compression threshold
- extraction
- completion
- language model
- extension operator
- action policy
- unsupervised learning
- interactive text environment
- agent–environment interaction
- analogy
- abstraction
- holon
- institutional learning
- morality
- geometric theology
- categorical formulation
- quasi-periodic environment
- Koopman structure

3.2 Relation Vocabulary

Let the latent typed edge set E^* consist of triples

$$(u, r, v), \qquad u, v \in V^*,\ r \in R^*,$$

where the relation vocabulary R^* is:

- defines
- develops
- implements
- realizes
- induces
- extends
- depends_on
- acts_on
- appears_in
- analogizes_with
- predicts
- updates
- compresses_to
- completes
- clusters
- serves_as
- stabilizes
- grounds
- tests

4 Hybrid Labeling Protocol

The environment must map latent nodes and latent edges to textual segments in a reproducible way. To do so, we define a hybrid labeling protocol.

4.1 Node-to-Text Incidence Map

Let

$$I_V : V^* \to \mathcal{P}(C)$$

assign to each latent node the set of corpus segments associated with it.

The map I_V is defined as the union of three components:

$$I_V(v) = I_V^{\text{exact}}(v) \cup I_V^{\text{alias}}(v) \cup I_V^{\text{manual}}(v).$$

Definition 1 (Exact Lexical Anchoring)**.** *For each node $v \in V^*$, define a canonical label $L(v)$. Then*

$$p \in I_V^{\text{exact}}(v) \iff L(v) \textit{ occurs verbatim in } p.$$

Definition 2 (Alias Normalization)**.** *For each node $v \in V^*$, define an alias set $A(v)$ containing normalized variants of the canonical label, including symbolic forms, hyphen variants, and close lexical alternatives. Then*

$$p \in I_V^{\text{alias}}(v) \iff \exists a \in A(v) \textit{ such that } a \textit{ occurs in } p.$$

Definition 3 (Manual Concept Annotation)**.** *For each node $v \in V^*$, a curator may add a segment p to $I_V^{\text{manual}}(v)$ whenever p clearly expresses the concept denoted by v even if no canonical label or alias appears explicitly.*

Remark 1. *The exact component provides transparency, the alias component reduces brittleness, and the manual component captures implicit conceptual expression. The hybrid scheme therefore balances reproducibility and semantic adequacy.*

4.2 Edge-to-Text Incidence Map

Let

$$I_E : E^* \to \mathcal{P}(C)$$

assign to each latent typed edge the set of corpus segments expressing that relation.

Similarly,

$$I_E(e) = I_E^{\text{exact}}(e) \cup I_E^{\text{alias}}(e) \cup I_E^{\text{manual}}(e).$$

Here I_E^{exact} collects segments explicitly stating the relation, I_E^{alias} collects segments expressing a normalized variant, and I_E^{manual} captures curator-added relation evidence.

4.3 Examples of Alias Sets

Representative alias sets include:

- A(inference–implication loop): {"inference–implication loop", "inference-implication loop", "$F \circ g$", "closure operator"}
- A(graph quotient): {"graph quotient", "quotient graph"}
- A(clique complex): {"clique complex", "complex of cliques"}
- A(Kan condition): {"Kan condition", "horn-filling condition"}
- A(extension operator): {"extension operator", "completion engine"}

5 Action Protocol

The agent does not access the corpus freely. Instead, it takes actions with respect to its current world model. The environment then returns textual evidence associated with those actions. During training, all actions draw exclusively from C_{train}; segments from C_{test} are inaccessible until the evaluation phase.

5.1 Action Space

The action space consists of:

- inspect(v), where v is a current graph node,
- inspect_relation(u, r, v), where (u, r, v) is a current or predicted edge,
- expand_cluster(K), where K is a current node-cluster,
- explore_random(), and
- verify_edge(u, r, v), which requests evidence for a predicted edge.

5.2 Observation Sampling

Let $H_t \subseteq C$ denote the set of segments already observed up to time t.

Then the observation rules are:

$$\text{inspect}(v) \sim \text{Sample}\big(I_V(v) \setminus H_t\big),$$
$$\text{inspect\_relation}(u, r, v) \sim \text{Sample}\big(I_E(u, r, v) \setminus H_t\big).$$

If the corresponding unseen set is empty, the environment samples instead from the full associated set with preference for least frequently returned segments.

Remark 2. *This protocol ensures that an action requests evidence about a node or relation, not unrestricted access to the full corpus. The environment is therefore partially observable and exploration-dependent. During training the sampling domain is restricted to C_{train}; segments from C_{test} become accessible only during the evaluation phase described in Section 12.*

6 Agent World Model

The agent maintains a knowledge graph

$$G_t = (V_t, E_t)$$

whose nodes correspond to discovered entities and whose edges correspond to relations extracted from observations.

Entities are represented by embedding vectors

$$z_v \in \mathbb{R}^d,$$

and relations are stored in a relation tensor.

Upon observing a new text segment, the agent extracts relational triples

$$(v_i, r, v_j)$$

which are used to update the graph.

This graph is not the latent graph G^*. It is the agent's current world model of the environment.

7 Compression and Extension

After each graph update, the world model undergoes two operations.

7.1 Compression

Compression merges nodes with similar embeddings, producing equivalence classes of entities that represent conceptual abstractions.

This operation can be interpreted as a graph quotient

$$G_t \to H_t$$

which reduces representational redundancy.

7.2 Extension

Extension predicts missing relations in the compressed graph. In the present implementation this prediction is performed by a language model conditioned on the current graph state.

The predicted relations form a proposed graph

$$G_{t+1}^{\text{pred}}$$

which represents the agent's expectation of the next graph update.

8 Simplicial Structure

The knowledge graph induces a clique complex

$$X(G_t)$$

whose simplices correspond to sets of mutually connected entities.

Compression induces simplicial face maps by collapsing equivalent nodes, while extension predicts missing simplices through completion of partially specified horns.

This connects the architecture to the simplicial completion framework developed in earlier papers.

9 Exact Proposed Ground-Truth Graph

We now expose the proposed latent graph for the TIM world. This graph is not given to the agent during training, but it defines the environment used for evaluation.

9.1 Core Proposed Edge Set

The following typed triples constitute the first-pass proposed ground-truth graph. Edges involving TIM XII are marked *(withheld)* to indicate that they are not available to the agent during training and form part of the evaluation target.

Series-Level Document Structure

(TIM I, defines, epistemic closure)
(TIM I, defines, inference–implication loop)
(TIM I, defines, fixed point)
(TIM I, develops, world model)
(TIM II, develops, agent–environment interaction)
(TIM III, develops, quasi–periodic environment)
(TIM III, develops, Koopman structure)
(TIM IV, develops, institutional learning)
(TIM V, develops, analogy)
(TIM V, develops, abstraction)
(TIM VI, develops, horn filling)
(TIM VI, develops, holon)
(TIM VII, develops, morality)
(TIM VIII, develops, geometric theology)
(TIM IX, develops, categorical formulation)
(TIM X, develops, simplicial dynamics)
(TIM XI, extends, TIM X)
(TIM XI, realizes, compression)
(TIM XI, realizes, extension)
(TIM XII, tests, TIM XI) *(withheld: evaluation target)*

Epistemic Architecture

(observation, grounds, world model)
(inference, acts_on, observation)
(implication, acts_on, world model)
(inference–implication loop, depends_on, inference)
(inference–implication loop, depends_on, implication)
(epistemic closure, depends_on, fixed point)
(fixed point, stabilizes, world model)
(prediction error, updates, world model)
(internal supervision, realizes, prediction error)
(world-model update, updates, world model)

Compression–Extension Architecture

(compression, acts_on, world model)
(extension, acts_on, world model)
(compression, compresses_to, equivalence relation)
(compression, implements, graph quotient)
(extension, implements, graph completion)
(graph quotient, implements, compression)
(graph completion, implements, extension)
(knowledge graph, serves_as, world model)
(language model, serves_as, extension operator)
(extension operator, realizes, completion)
(extraction, updates, knowledge graph)
(completion, updates, knowledge graph)
(clustering, implements, compression)
(compression threshold, updates, clustering)
(action policy, predicts, observation)
(action, grounds, observation)
(interactive text environment, grounds, observation)
(unsupervised learning, realizes, internal supervision)

Graph–Simplicial Correspondence

(graph, induces, clique complex)
(clique, depends_on, graph)
(clique complex, depends_on, clique)
(clique complex, depends_on, simplex)
(compression, induces, face map)
(extension, induces, horn filling)
(horn filling, completes, horn)
(horn filling, induces, Kan condition)
(Kan condition, depends_on, horn)
(simplicial dynamics, depends_on, clique complex)
(simplicial dynamics, depends_on, face map)
(simplicial dynamics, depends_on, horn filling)
(quotient space, depends_on, equivalence relation)
(classifier, compresses_to, quotient space)

Cross-Series Structural Correspondences

$$\begin{aligned}
&(\text{analogy}, \text{analogizes_with}, \text{abstraction})\\
&(\text{analogy}, \text{analogizes_with}, \text{horn filling})\\
&(\text{holon}, \text{analogizes_with}, \text{horn filling})\\
&(\text{categorical formulation}, \text{extends}, \text{compression})\\
&(\text{categorical formulation}, \text{extends}, \text{extension})\\
&(\text{Koopman structure}, \text{depends_on}, \text{relational invariant})\\
&(\text{quasi–periodic environment}, \text{grounds}, \text{relational invariant})\\
&(\text{institutional learning}, \text{realizes}, \text{compression})\\
&(\text{institutional learning}, \text{realizes}, \text{extension})
\end{aligned}$$

Remark 3. *The graph above is a first-pass proposed ground truth, not a claim of uniquely correct ontology. Its role in the experiment is to provide a controlled latent world against which graph reconstruction, horn completion, and graph-update prediction can be evaluated. All edges whose subject or object is TIM XII are withheld from the agent during training. They are part of the evaluation target: the agent is expected to predict them through extension from the structure learned during training on TIM I–XI alone.*

10 Exact Proposed Ground-Truth Simplices

The latent simplicial family Σ^* is generated by coherent conceptual motifs. The following are the proposed core simplices.

Epistemic Simplices

- $\{\text{inference}, \text{implication}, \text{inference–implication loop}\}$
- $\{\text{inference–implication loop}, \text{fixed point}, \text{epistemic closure}\}$
- $\{\text{prediction error}, \text{internal supervision}, \text{world-model update}\}$

Compression–Extension Simplices

- $\{\text{compression}, \text{graph quotient}, \text{equivalence relation}\}$
- $\{\text{extension}, \text{graph completion}, \text{extension operator}\}$
- $\{\text{knowledge graph}, \text{world model}, \text{prediction error}\}$
- $\{\text{completion}, \text{language model}, \text{extension operator}\}$
- $\{\text{compression}, \text{extension}, \text{world model}, \text{action}\}$

Graph–Simplicial Simplices

- $\{\text{graph}, \text{clique}, \text{clique complex}, \text{simplex}\}$
- $\{\text{compression}, \text{graph quotient}, \text{face map}\}$
- $\{\text{extension}, \text{graph completion}, \text{horn filling}\}$
- $\{\text{horn}, \text{horn filling}, \text{Kan condition}\}$
- $\{\text{simplicial dynamics}, \text{clique complex}, \text{face map}, \text{horn filling}\}$

Cross-Series Simplices

- $\{\text{analogy}, \text{abstraction}, \text{horn filling}\}$
- $\{\text{holon}, \text{horn filling}, \text{analogy}\}$
- $\{\text{quasi–periodic environment}, \text{Koopman structure}, \text{relational invariant}\}$
- $\{\text{institutional learning}, \text{compression}, \text{extension}\}$

11 Representative Incidence Tables

In practice the full incidence tables would appear in an appendix. We include representative samples here. All supporting segments listed are drawn from C_{train}. In the full experimental release, the complete incidence maps I_V and I_E would be provided together with written annotation guidelines specifying the use of exact lexical anchoring, alias normalization, and manual concept or relation annotation.

Sample Node-Incidence Table

Node	Representative supporting segments
compression	TIM XI Introduction; TIM XI Section 3; TIM XI Section 6; TIM X Section 4
extension	TIM XI Introduction; TIM XI Section 4; TIM XI Section 6; TIM XI Section 7; TIM X Section 4
graph quotient	TIM XI Section 3; TIM XI Section 6
graph completion	TIM XI Section 4; TIM XI Section 6
clique complex	TIM XI Section 5; TIM XI Section 6; TIM XI Section 10
horn filling	TIM VI Section 4; TIM VI Section 5; TIM X Section 8; TIM XI Section 6
Kan condition	TIM X Section 8; TIM XI Section 6; TIM XI Section 9
epistemic closure	TIM I Introduction; TIM I fixed-point discussion; TIM XI Introduction; TIM XI Conclusion

Sample Edge-Incidence Table

Edge	Representative supporting segments
(graph quotient, implements, compression)	TIM XI Section 3
(graph completion, implements, extension)	TIM XI Section 4
(compression, induces, face map)	TIM XI Section 6
(extension, induces, horn filling)	TIM XI Section 6
(language model, serves_as, extension operator)	TIM XI Section 7; TIM XI Conclusion
(prediction error, updates, world model)	TIM XI Section 7; TIM XI Algorithm

12 Experimental Protocol

Training is performed on $C_{\text{train}} = \text{TIM I–XI}$. The agent sequentially observes textual segments drawn from this corpus and updates its knowledge graph using the compression–extension cycle described above.

TIM XII is withheld during training. During evaluation, the agent is exposed to segments of TIM XII one at a time and predicts the structural updates each segment induces before observing it. No segment of TIM XII is accessible to the agent prior to this evaluation phase.

12.1 Main Interaction Loop

At time t:

(1) The agent selects an action a_t.

(2) The environment returns a segment $o_t \in C$ according to the action protocol.

(3) The agent extracts triples and updates G_t.

(4) The agent compresses G_t to H_t.

(5) The agent extends H_t to obtain G_{t+1}^{pred}.

(6) The agent selects the next action using its current policy.

(7) The next segment is revealed, producing the actual graph update G_{t+1}^{actual}.

(8) The prediction loss is computed as

$$L_t = \text{diff}\big(G_{t+1}^{\text{pred}}, G_{t+1}^{\text{actual}}\big).$$

12.2 Evaluation Tasks

Three evaluation tasks are considered.

Edge Recovery

Selected edges from G^* are held out from training. The agent's ability to recover them through extension is measured using precision, recall, and F_1.

Horn Completion

Selected simplices in Σ^* are partially withheld. The agent is asked to complete the corresponding horns. Accuracy on these horn-completion tasks measures whether the agent recovers higher-order conceptual structure.

Representative horn-completion tasks include:

- $\{\text{compression}, \text{graph quotient}, ?\} \to$ face map,
- $\{\text{extension}, \text{graph completion}, ?\} \to$ horn filling,
- $\{\text{inference}, \text{implication}, ?\} \to$ inference–implication loop,
- $\{\text{inference–implication loop}, \text{fixed point}, ?\} \to$ epistemic closure.

Graph Stabilization

The stability of the knowledge graph is measured by tracking:

- number of nodes,
- number of edges,
- number and composition of clusters,
- compression threshold, and
- graph-update prediction error.

13 Discussion

If compression–extension dynamics successfully capture the relational structure of the environment, the agent should anticipate structural updates introduced by previously unseen text. In this setting, successful prediction of graph updates induced by TIM XII demonstrates that the agent has reconstructed relational invariants of the TIM conceptual world from TIM I–XI alone.

More broadly, the experiment illustrates how embedded epistemic systems can learn structural models of their environment through internally generated prediction tasks. The present construction is intentionally self-contained: the TIM corpus functions as a controlled textual world in which the latent ontology, latent relation structure, and latent simplicial motifs can all be explicitly defined. This makes the environment suitable for proof-of-concept evaluation of the imagination-machine architecture before extension to broader corpora.

13.1 Scope and Status of the Environment

The TIM world defined here is a controlled proof-of-concept environment. Its purpose is not to establish immediate generalization to arbitrary corpora, but to test whether the architecture can recover explicit latent conceptual structure from a corpus whose ontology, relation structure, and simplicial motifs can be specified in advance.

Accordingly, the present experiment should be read as an internal validation study of the compression–extension architecture. A positive result would show that the architecture can recover and predict structural updates in a textual world engineered to make such evaluation possible. Whether the same architecture generalizes to broader or noisier corpora is a distinct empirical question left for future work.

14 Conclusion

This paper introduces an experimental realization of the Imagination Machine architecture in an open text world.

An agent embedded in a textual environment incrementally reconstructs a conceptual knowledge graph, compresses that graph through clustering, and predicts missing relations through extension. The induced simplicial structure provides a natural representation of higher-order conceptual relations.

To make the environment rigorous, the paper specifies an explicit latent ontology, an explicit relation vocabulary, a hybrid labeling protocol, and an explicit proposed ground-truth graph and simplex family for the TIM world. TIM XII is withheld during training; evaluation on its segments tests whether the agent can predict structural updates induced by previously unseen text. Successful prediction demonstrates that prediction of structural updates can serve as an operational definition of understanding for embedded epistemic systems.